

INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SECCIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

**REGULACIÓN DE VOLTAJE DE LA MÁQUINA SÍNCRONA
UTILIZANDO UNA RED NEURONAL “NEOCOGNITRÓN”**

TESIS

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN CIENCIAS CON ESPECIALIDAD
EN INGENIERÍA ELÉCTRICA

P R E S E N T A:

YORAM ASTUDILLO BAZA



MÉXICO, D.F., DICIEMBRE 2004



Resumen

Los generadores síncronos son las máquinas más usadas para la generación de energía eléctrica. En la generación de energía eléctrica dentro de un sistema interconectado se requiere que la frecuencia y el voltaje en terminales permanezcan constantes ante diferentes disturbios que puedan presentarse.

Los sistemas de potencia son sistemas dinámicos y su operación es por naturaleza estocástica. Las características del sistema son no lineales, la ganancia del sistema aumenta con la carga del generador y la potencia del sistema. Así, los parámetros de los controladores que son óptimos para un grupo de condiciones de operación pueden no ser óptimos para otros grupos de condiciones de operación.

En los controles convencionales con parámetros fijos, las ganancias y otros parámetros no pueden ajustarse idealmente en un conjunto de puntos de operación. Esto hace factible desarrollar e implementar mejores controles basados en técnicas más modernas y sofisticadas de control.

En este trabajo se presenta el análisis y la aplicación de un esquema de control para la máquina síncrona utilizando un controlador cuyos fundamentos teóricos se basan en las metodologías y conceptos de la Inteligencia Artificial. Se propone una estructura de las Redes Neuronales Artificiales llamada neocognitrón para construir un regulador neuronal con propiedades de adaptación.

El regulador de voltaje propuesto permite que el sistema de potencia sea estable sobre una amplia gama de condiciones de operación. Para la situación en que el sistema está sujeto a disturbios al azar, exhibe un funcionamiento aceptable. Estos resultados son validados por estudios de simulación por computadora.

Se utiliza un sistema de prueba máquina bus – infinito, donde el generador esta conectado a un bus infinito a través de una impedancia externa. El generador esta representado por un modelo de 5^o orden.

Finalmente, el desempeño del regulador neuronal se compara con el esquema convencional del tipo ST1 bajo las mismas condiciones de operación. Como se ve, este trabajo demuestra que la aplicación de este controlador inteligente no lineal tiene un mejor funcionamiento.



Abstract

Synchronous generators are the most utilized machines for electrical power generation. In electrical power generation inside an interconnected system it is necessary to keep both frequency and voltage constant, with different disturbances that could be present.

The conventional controls that at the moment are in operation to regulate the voltage, called automatic regulators of voltage are designed to work around a certain operation point and with constant parameters, for what they don't always provide a satisfactory voltage regulation, because in an electric system of power constant changes of the operation point are presented, besides being non lineal systems.

In this work an examination as well as the application of a control scheme for the synchronous machine using a controller-based Artificial Intelligence concepts and methodologies is presented. This controller proposes a structure of the Nets Neuronal Artificial called neocognitrón to build a regulator neuronal with properties of adaptation.

The regulator of voltage proposed allows that the system of power is stable on a wide range of operation conditions. For the situation in that the system is at random subject to disturbances, it exhibits an acceptable operation. These results are demonstrated by simulation studies by computer.

A one machine infinite-bus test system is utilized, where the generator this connected one to an infinite bus through an external impedance. The generator this represented by a model of 5^o order. A single reheat steam turbine is utilized with a governor of the electrohydraulic type.

Finally, the behavior of the regulator neural is compared with the one obtained of the conventional scheme of the type ST1 under the same operation conditions. As it will be seen, this work it is demonstrated that this intelligent controller's application not lineal has a better operation.



Índice de Contenido

Resumen	4
Abstract	5
Índice de Figuras	8
Índice de Tablas	12
Glosario de Términos	12
Agradecimientos	15
CAPÍTULO 1. Introducción	16
1.1.- Estado del Arte	16
1.2.- Objetivos de la Tesis	17
1.3.- Justificación	17
1.4.- Antecedentes del Problema	18
1.5.- Contenido de la Tesis	20
1.6.- Aportaciones	21
CAPÍTULO 2. Control Neuronal y Control Inverso	22
2.1.- Introducción	22
2.2.- Sistemas de Control	23
2.3.- Sistemas de Control con Retroalimentación	24
2.4.- Control Neuronal	25
2.5.- Control Inverso, Aprendizaje Inverso..	26
CAPÍTULO 3. El neocognitrón	31
3.1.- Introducción	31
3.2.- ¿Qué es el Neocognitrón?	32
3.3.- Arquitectura del Neocognitrón	32
3.4.- Descripción de las Funciones de los Elementos de Procesamiento ó células del Neocognitrón	33
3.4.1.- Descripción Matemática de una Célula S	34
3.4.2.- Descripción Matemática de una Célula V	35
3.4.3.- Descripción Matemática de una Célula C	35
3.5.- Pesos y Conexiones en el Neocognitrón	36
3.5.1.- Pesos Compartidos	36
3.5.2.- Pesos a y Pesos b	36
3.5.3.- Pesos c y Pesos d	36
3.6.- Aprendizaje del Neocognitrón	37
CAPÍTULO 4. Regulación de Voltaje de la Máquina Síncrona Usando la Red Neuronal Neocognitrón	39
4.1.- Introducción	39
4.2.- Esquema Básico de un Regulador Automático de Voltaje Convencional	40
4.3.- Regulación de Voltaje de la Máquina Síncrona Usando el Neocognitrón	41
4.3.1.- Estructura del Control Propuesto.....	41



4.3.2.- Procedimiento para el Control Propuesto.....	42
4.3.3. Arquitectura del Neocognitrón Utilizada para Regular el Voltaje de la Máquina Síncrona.....	43
4.3.4. Esquema de Entrenamiento del Neocognitrón.....	44
4.4. Simulación de los Reguladores de Voltaje Bajo Distintas Condiciones de Operación.....	45
4.4.1. Caso 1. $P=0.8240$. Con una falla de 6, 9 y 12 ciclos de duración.....	46
4.4.2. Análisis de resultados del caso 1.....	49
4.4.3. Caso 2. $P=0.9833$. Con una falla de 6, 9 y 12 ciclos de duración.....	51
4.4.4. Análisis de resultados del caso 2.....	54
4.4.5. Caso 3. $P=0.5284$. Con una falla de 6, 9 y 12 ciclos de duración.....	55
4.4.6. Análisis de resultados del caso 3.....	58
4.4.7. Caso 4. $P=1.1000$. Con una falla de 6, 9 y 12 ciclos de duración.....	59
4.4.8. Análisis de resultados del caso 4.....	62
4.4.9. Caso 5. $P=0.9592$. Con una falla de 6, 9 y 12 ciclos de duración.....	63
4.4.10. Análisis de resultados del caso 5.....	66
CAPÍTULO 5. Conclusiones	68
5.1.- Conclusiones	68
5.2.- Recomendaciones	69
5.3.- Trabajos Futuros	69
Referencias	70
APENDICES	
A. Modelo Matemático de la Máquina Síncrona.....	75
A.1. Modelo Matemático.....	75
A.1.1. Modelo de la Máquina Síncrona.....	75
A.1.2. Modelo de la Red.....	75
A.1.3. Modelo de la Turbina de Vapor y del Gobernador de Velocidad.....	76
A.1.4. Modelo del Sistema de Excitación y Regulador de Voltaje.....	76
A.1.5. Datos del Sistema Bajo Prueba.....	77
A.2. Determinación de las Condiciones Iniciales y de las Ecuaciones Dinámicas para el Sistema Máquina Bus Infinito.....	78
A.2.1. Red.....	78
A.2.2. Generador.....	79
A.2.3. Sistema de Excitación – Regulador de Voltaje.....	80
A.2.4. Turbina de Vapor y Gobernador de Velocidad.....	81
A.2.5. Ecuaciones Dinámicas.....	81
B. Teoría de las Redes Neuronales	83
B.1. Introducción	83
B.1.1. Capacidad de Aprendizaje	83
B.1.2. Autoorganización	83
B.1.3. Tolerancia a Fallos	83
B.1.4. Operación en Tiempo Real	84
B.1.5. Fácil Inserción en la Tecnología Existente	84
B.2. Teoría de las Redes Neuronales	84



B.2.1. Redes Neuronales	85
B.2.2. Entrenamiento de la Red	85
B.2.3. Prueba de la Red	88
C. Índice de Error de Desempeño	89
D. Programas de Simulación	91
D.1. Máquina Síncrona	91
D.2. Máquina Síncrona Bus Infinito con sistema de excitación ST1	94
D.3. Programa de Aplicación de la Máquina Síncrona con Regulador Neocognitrón	101

Índice de Figuras

Figura 2.1. Diagrama de bloques de un sistema de control retroalimentado.....	24
Figura 2.2. Diagrama de bloques para el aprendizaje inverso: (a) Bloque de la planta, (b) Fase de entrenamiento, (c) Fase de aplicación	28
Figura 2.3. Esquema del control neuronal con aprendizaje inverso en línea.....	29
Figura 2.4. Control inverso adaptable del modelo de referencia, proceso fuera de línea. (a) Entrenamiento del Identificador, (b) entrenamiento del controlador..	30
Figura 3.1. Arquitectura común de la red neuronal neocognitrón	32
Figura 3.2. Esquema de la conexión entre las células del neocognitrón.	34
Figura 3.3. Pesos y conexiones en el neocognitrón	36
Figura 4.1. Esquema básico del sistema máquina bus infinito con un RAV convencional.....	41
Figura 4.2. Estructura del control propuesto, procedimiento fuera de línea, a) Fase de identificación, b) Fase de control.....	42
Figura 4.3. Arquitectura del neocognitrón propuesta para el esquema del control neuronal.....	44
Figura 4.4. Comportamiento del voltaje en terminales ante una falla de 6 ciclos de duración, caso 1.....	46
Figura 4.5. Índice de comportamiento del voltaje en terminales, caso 1.....	46
Figura 4.6. Comportamiento del ángulo de la máquina ante una falla de 6 ciclos de duración, caso 1.....	47
Figura 4.7. Comportamiento de la potencia activa ante una falla de 6 ciclos de duración, caso 1.....	47
Figura 4.8. Comportamiento de la potencia reactiva ante una falla de 6 ciclos de duración, caso 1.....	47
Figura 4.9. Comportamiento del voltaje de campo ante una falla de 6 ciclos de duración, caso 1.....	47
Figura 4.10. Comportamiento del voltaje en terminales ante una falla de 9 ciclos de duración, caso 1.....	47
Figura 4.11. Índice de comportamiento del voltaje en terminales, caso 1.....	47
Figura 4.12. Comportamiento del ángulo de la máquina ante una falla de 9 ciclos de duración, caso 1.....	48



Figura 4.13. Comportamiento de la potencia activa ante una falla de 9 ciclos de duración, caso 1.....	48
Figura 4.14. Comportamiento de la potencia reactiva ante una falla de 9 ciclos de duración, caso 1.....	48
Figura 4.15. Comportamiento del voltaje de campo ante una falla de 9 ciclos de duración, caso 1.....	48
Figura 4.16. Comportamiento del voltaje en terminales ante una falla de 12 ciclos de duración, caso 1.....	48
Figura 4.17. Índice de comportamiento del voltaje en terminales, caso 1.....	48
Figura 4.18. Comportamiento del ángulo de la máquina ante una falla de 12 ciclos de duración, caso 1.....	49
Figura 4.19. Comportamiento de la potencia activa ante una falla de 12 ciclos de duración, caso 1.....	49
Figura 4.20. Comportamiento de la potencia reactiva ante una falla de 12 ciclos de duración, caso 1.....	49
Figura 4.21. Comportamiento del voltaje de campo ante una falla de 12 ciclos de duración, caso 1.....	49
Figura 4.22. Comportamiento del voltaje en terminales ante una falla de 6 ciclos de duración, caso 2	51
Figura 4.23. Índice de comportamiento del voltaje en terminales, caso 2.....	51
Figura 4.24. Comportamiento del ángulo de la máquina ante una falla de 6 ciclos de duración, caso 2	52
Figura 4.25. Comportamiento de la potencia activa ante una falla de 6 ciclos de duración, caso 2	52
Figura 4.26. Comportamiento de la potencia reactiva ante una falla de 6 ciclos de duración, caso 2	52
Figura 4.27. Comportamiento del voltaje de campo ante una falla de 6 ciclos de duración, caso 2	52
Figura 4.28. Comportamiento del voltaje en terminales ante una falla de 9 ciclos de duración, caso 2	52
Figura 4.29. Índice de comportamiento del voltaje en terminales, caso 2.....	52
Figura 4.30. Comportamiento del ángulo de la máquina ante una falla de 9 ciclos de duración, caso 2	53
Figura 4.31. Comportamiento de la potencia activa ante una falla de 9 ciclos de duración, caso 2	53
Figura 4.32. Comportamiento de la potencia reactiva ante una falla de 9 ciclos de duración, caso 2	53
Figura 4.33. Comportamiento del voltaje de campo ante una falla de 9 ciclos de duración, caso 2	53
Figura 4.34. Comportamiento del voltaje en terminales ante una falla de 12 ciclos de duración, caso 2	53
Figura 4.35. Índice de comportamiento del voltaje en terminales, caso 2	53
Figura 4.36. Comportamiento del ángulo de la máquina ante una falla de 12 ciclos de duración, caso 2.....	54
Figura 4.37. Comportamiento de la potencia activa ante una falla de 12 ciclos de duración, caso 2.....	54
Figura 4.38. Comportamiento de la potencia reactiva ante una falla de 12 ciclos de duración, caso 2.....	54
Figura 4.39. Comportamiento del voltaje de campo ante una falla de 12 ciclos de duración, caso 2	54



Figura 4.40. Comportamiento del voltaje en terminales ante una falla de 6 ciclos de duración, caso 3	55
Figura 4.41. Índice de comportamiento del voltaje en terminales, caso 3	55
Figura 4.42. Comportamiento del ángulo de la máquina ante una falla de 6 ciclos de duración, caso 3	56
Figura 4.43. Comportamiento de la potencia activa ante una falla de 6 ciclos de duración, caso 3	56
Figura 4.44. Comportamiento de la potencia reactiva ante una falla de 6 ciclos de duración, caso 3	56
Figura 4.45. Comportamiento del voltaje de campo ante una falla de 6 ciclos de duración, caso 3	56
Figura 4.46. Comportamiento del voltaje en terminales ante una falla de 9 ciclos de duración, caso 3	56
Figura 4.47. Índice de comportamiento del voltaje en terminales, caso 3	56
Figura 4.48. Comportamiento del ángulo de la máquina ante una falla de 9 ciclos de duración, caso 3	57
Figura 4.49. Comportamiento de la potencia activa ante una falla de 9 ciclos de duración, caso 3	57
Figura 4.50. Comportamiento de la potencia reactiva ante una falla de 9 ciclos de duración, caso 3	57
Figura 4.51. Comportamiento del voltaje de campo ante una falla de 9 ciclos de duración, caso 3	57
Figura 4.52. Comportamiento del voltaje en terminales ante una falla de 12 ciclos de duración, caso 3	57
Figura 4.53. Índice de comportamiento del voltaje en terminales, caso 3	57
Figura 4.54. Comportamiento del ángulo de la máquina ante una falla de 12 ciclos de duración, caso 3	58
Figura 4.55. Comportamiento de la potencia activa ante una falla de 12 ciclos de duración, caso 3	58
Figura 4.56. Comportamiento de la potencia reactiva ante una falla de 12 ciclos de duración, caso 3	58
Figura 4.57. Comportamiento del voltaje de campo ante una falla de 12 ciclos de duración, caso 3	58
Figura 4.58. Comportamiento del voltaje en terminales ante una falla de 6 ciclos de duración, caso 4	59
Figura 4.59. Índice de comportamiento del voltaje en terminales, caso 4	59
Figura 4.60. Comportamiento del ángulo de la máquina ante una falla de 6 ciclos de duración, caso 4	60
Figura 4.61. Comportamiento de la potencia activa ante una falla de 6 ciclos de duración, caso 4	60
Figura 4.62. Comportamiento de la potencia reactiva ante una falla de 6 ciclos de duración, caso 4	60
Figura 4.63. Comportamiento del voltaje de campo ante una falla de 6 ciclos de duración, caso 4	60
Figura 4.64. Comportamiento del voltaje en terminales ante una falla de 9 ciclos de duración, caso 4	60
Figura 4.65. Índice de comportamiento del voltaje en terminales, caso 4	60
Figura 4.66. Comportamiento del ángulo de la máquina ante una falla de 9 ciclos de duración, caso 4	61



Figura 4.67. Comportamiento de la potencia activa ante una falla de 9 ciclos de duración, caso 4	61
Figura 4.68. Comportamiento de la potencia reactiva ante una falla de 9 ciclos de duración, caso 4	61
Figura 4.69. Comportamiento del voltaje de campo ante una falla de 9 ciclos de duración, caso 4	61
Figura 4.70. Comportamiento del voltaje en terminales ante una falla de 12 ciclos de duración, caso 4	61
Figura 4.71. Índice de comportamiento del voltaje en terminales, caso 4	61
Figura 4.72. Comportamiento del ángulo de la máquina ante una falla de 12 ciclos de duración, caso 4	62
Figura 4.73. Comportamiento de la potencia activa ante una falla de 12 ciclos de duración, caso 4	62
Figura 4.74. Comportamiento de la potencia reactiva ante una falla de 12 ciclos de duración, caso 4	62
Figura 4.75. Comportamiento del voltaje de campo ante una falla de 12 ciclos de duración, caso 4	62
Figura 4.76. Comportamiento del voltaje en terminales ante una falla de 6 ciclos de duración, caso 5	63
Figura 4.77. Índice de comportamiento del voltaje en terminales, caso 5	63
Figura 4.78. Comportamiento del ángulo de la máquina ante una falla de 6 ciclos de duración, caso 5	64
Figura 4.79. Comportamiento de la potencia activa ante una falla de 6 ciclos de duración, caso 5	64
Figura 4.80. Comportamiento de la potencia reactiva ante una falla de 6 ciclos de duración, caso 5	64
Figura 4.81. Comportamiento del voltaje de campo ante una falla de 6 ciclos de duración, caso 5	64
Figura 4.82. Comportamiento del voltaje en terminales ante una falla de 9 ciclos de duración, caso 5	64
Figura 4.83. Índice de comportamiento del voltaje en terminales, caso 5	64
Figura 4.84. Comportamiento del ángulo de la máquina ante una falla de 9 ciclos de duración, caso 5	65
Figura 4.85. Comportamiento de la potencia activa ante una falla de 9 ciclos de duración, caso 5	65
Figura 4.86. Comportamiento de la potencia reactiva ante una falla de 9 ciclos de duración, caso 5	65
Figura 4.87. Comportamiento del voltaje de campo ante una falla de 9 ciclos de duración, caso 5	65
Figura 4.88. Comportamiento del voltaje en terminales ante una falla de 12 ciclos de duración, caso 5	65
Figura 4.89. Índice de comportamiento del voltaje en terminales, caso 5.....	65
Figura 4.90. Comportamiento del ángulo de la máquina ante una falla de 12 ciclos de duración, caso 5	66
Figura 4.91. Comportamiento de la potencia activa ante una falla de 12 ciclos de duración, caso 5	66
Figura 4.92. Comportamiento de la potencia reactiva ante una falla de 12 ciclos de duración, caso 5	66
Figura 4.93. Comportamiento del voltaje de campo ante una falla de 12 ciclos de duración, caso 5	66



Figura A.1. Diagrama de bloques del gobernador de velocidad mecánico – hidráulico.....	76
Figura A.2. Diagrama de bloques del regulador tipo ST1.....	77
Figura A.3. Diagrama del sistema máquina bus infinito usado en esta tesis.....	78
Figura A.4. Diagrama vectorial de la máquina síncrona, para determinar el ángulo de carga δ_r , caso subtransitorio.....	79
Figura B.1. Estructura de una red neuronal	84
Figura B.2. Detalle de una red neuronal	85
Figura B.3. Modo de trabajo con redes neuronales	86
Figura C.1. Índice de comportamiento del voltaje en terminales	90
Figura D.1. Diagrama de flujo del programa de aplicación con el regulador Neocognitrón	103

Índice de Tablas

Tabla 4.1. Casos analizados en la simulación	46
Tabla A.1. Datos del sistema bajo prueba	77

Glosario de Términos

SEP	Sistema eléctrico de potencia.
RAV	Regulador automático de voltaje.
AI	Inteligencia artificial.
ANS	Sistema neuronal artificial.
PEs	Elementos de procesamiento.
USI	Capa de células S.
UCI	Capa de células C.
UVI	Capa de células V.
l	Número de serie de una etapa del neocognitrón.
n	Posición de una célula S, V y C.
k	Número de serie del plano de una célula del neocognitrón.
r_l	Parámetro de selectividad de una célula del neocognitrón.
K_{cl-1}	Números de planos de célula en una capa del neocognitrón.
v	Posición del área de conexión de una célula del neocognitrón.
A_l, D_l	Área de conexión de una célula S, V y C.
j_l	Factor de conexión de una célula C del neocognitrón.
a_l, b_l, c_l, d_l	Pesos a, b, c y d del neocognitrón.
\hat{n}	Posición de una célula de activación del neocognitrón.
\hat{k}	Número de serie del plano de célula con la célula núcleo del neocognitrón.
q_l	Coefficiente de aprendizaje.
ANN	Red neuronal artificial
ANNS	Redes neuronales artificiales



PID	Control proporcional integral diferencial
PI	Control proporcional integral
NI	Identificador neuronal
NC	Controlador neuronal
$u(t)$	Entrada del sistema
$e_c(t)$	Error del control
$y(t)$	Salida del sistema
e	Error
Δe	Cambio del error
V_f	Voltaje de campo
δ	Ángulo del rotor
e''_d	Voltaje subtransitorio en eje d
e'_q	Voltaje transitorio en eje q
e''_q	Voltaje subtransitorio en eje q
F_{HP}	Fracción de la potencia de alta presión en la turbina
ω_o	Velocidad angular
f_o	Frecuencia base
i_d	Corriente en eje directo
i_q	Corriente en eje de cuadratura
K_A	Ganancia del regulador de voltaje
K_d	Coefficiente de amortiguamiento
K_F	Ganancia del estabilizador del sistema de excitación
M	Momento de inercia
P_{GV}	Potencia en la compuerta de la válvula
P_{HP}	Potencia de alta presión
P_m	Potencia Mecánica
r_a	Resistencia de armadura
r_e	Resistencia de la línea de transmisión
T_A	Constante del tiempo del regulador de voltaje
T_{CH}	Constante de tiempo de vapor exhausto
T_F	Constante del tiempo del sistema de control de excitación
T_m	Par mecánico
T'_{do}	Constante de tiempo transitoria en eje directo
T''_{do}	Constante de tiempo subtransitoria en eje directo
T''_{qo}	Constante de tiempo subtransitoria en eje cuadratura
T_{RH}	Constante de tiempo de recalentamiento
T_{SM}	Constante de tiempo del servomotor de la válvula
T_x	Constante de tiempo del excitador
X_d	Reactancia síncrona
X_e	Reactancia equivalente
X'_d	Reactancia transitoria en eje d
X'_q	Reactancia transitoria en eje q
X''_d	Reactancia subtransitoria en eje d
X''_q	Reactancia subtransitoria en eje q
p.u.	Por unidad
V_t	Voltaje en terminales del generador
S	Deslizamiento



T_e	Par eléctrico
V_d	Voltaje en eje directo
V_q	Voltaje en eje cuadratura
U	Entrada de control a la excitatriz
V_b	Voltaje en el bus



Agradecimientos

QUIERO AGRADECER A MIS PADRES, PEDRO Y NELYDA POR DARMER LA VIDA,
POR SU CARIÑO Y APOYO INCONDICIONAL.

A MIS HERMANOS HERIBERTA KARINA, PEDRO Y CINTHYA GUADALUPE
POR SU CARIÑO Y APOYO.

QUIERO MANIFESTAR POR ESTE CONDUCTO MI ADMIRACIÓN Y RESPETO HACIA EL
DR. DAVID ROMERO ROMERO, QUIEN CON SUS CONSEJOS Y SUGERENCIAS ME GUIÓ
PARA REALIZAR ESTE TRABAJO, EXPRESO MI GRATITUD HACIA EL.

ME ES IMPORTANTE DECIR GRACIAS A LOS PROFESORES DEL POSGRADO EN INGENIERÍA
ELÉCTRICA QUE INTERVINIERON DIRECTAMENTE O INDIRECTAMENTE
EN MI FORMACIÓN ACADÉMICA.

DR. DANIEL OLGUIN SALINAS, DR. RAUL CORTES MATEOS, DR. JAIME ROBLES GARCIA,
M. EN C. JESUS REYES GARCIA Y AL M. EN C. TOMAS IGNACIO ASIAIN OLIVARES

ASÍ MISMO DOY GRACIAS AL CONSEJO NACIONAL DE CIENCIA Y TECNOLOGÍA (CONACYT),
Y AL INSTITUTO POLITÉCNICO NACIONAL POR EL APOYO ECONÓMICO BRINDADO PARA
LLEVAR A CABO ESTE TRABAJO.

G R A C I A S

ATENTAMENTE

YORAM ASTUDILLO BAZA



Capítulo 1

Introducción

1.1. Estado del Arte.

La máquina síncrona que opera como un generador de corriente alterna impulsada por una turbina para convertir la energía mecánica en eléctrica es la principal fuente de generación de potencia eléctrica en el mundo. La conexión de las máquinas síncronas en paralelo exige que los generadores tengan concordancia de fases, la misma frecuencia y el mismo voltaje en las terminales. Para poder cumplir con estos requerimientos mencionados anteriormente se necesita equipo que controle la velocidad, la potencia y el voltaje en las terminales de los generadores.

Las grandes redes de potencia operan en una frecuencia constante (50 o 60 Hz) y estas confían casi exclusivamente en los generadores eléctricos para la provisión de energía eléctrica, una condición necesaria para la transmisión de potencia es que todos los generadores operen a la velocidad síncrona. La velocidad de una máquina síncrona esta directamente relacionada con la frecuencia y el número de pares de polo de la máquina.

El concepto de estabilidad en un sistema eléctrico de potencia relaciona la capacidad de un sistema de potencia para permanecer en sincronismo y la tendencia de un sistema para regresar y mantenerse en un punto de operación en el estado estacionario después de que ha ocurrido una falla [1]. Los sistemas eléctricos de potencia involucran generadores síncronos situados en largas estaciones de potencia y conectados a cargas que son dinámicas o estáticas. Las cargas tienen una importante influencia en la estabilidad y pueden afectar a un generador volviendolo inestable.

Se sabe que el control a través del lazo de excitación del generador síncrono es una de las formas más efectivas para estabilizar sistemas eléctricos de potencia [2]. La regulación de voltaje en el generador síncrono es de vital importancia, ya que de ello depende en gran parte la estabilidad de un sistema eléctrico de potencia. Uno de los primeros responsables de mantener la estabilidad del sistema es el regulador automático de voltaje. La función principal de un regulador de voltaje es la de ajustar continuamente el nivel de excitación del generador, en respuesta a los cambios del voltaje en las terminales del generador. El regulador de voltaje actúa precisamente para mantener el voltaje del generador deseado y un cambio en el nivel de excitación en respuesta a disturbios que se presenten.

La frecuencia y el voltaje, son las dos variables más importantes en un sistema eléctrico de potencia. El control de estas variables se realiza por medio de los controles, automático de voltaje y gobernador de la turbina. Los parámetros de estos controles son ajustados de tal forma que las respuestas a los cambios en el sistema tengan características conocidas. El control automático de voltaje se utiliza para regular el voltaje en las terminales del generador y el flujo de potencia reactiva.



Por otra parte existe un gran número de tendencias que ha causado un crecimiento del interés de la gente en el control inteligente de sistemas, basados en las técnicas de inteligencia artificial [3], debido a: algunas restricciones de los métodos convencionales debido al gran énfasis en linealizar los sistemas; el éxito que tienen los operadores para controlar un número de problemas prácticos quienes son difíciles de modelar; el análisis, diseño y simulación de los sistemas que se han desarrollado considerablemente en las últimas décadas y se han concentrado en las herramientas que ofrecen las paqueterías de software; y a la exitosa proliferación de las técnicas inteligentes en muchas aplicaciones de la ingeniería con la introducción de sistemas expertos y redes neuronales.

Los controles inteligentes, son controles que se diseñan e implementan siguiendo algunas técnicas existentes de la inteligencia artificial, se están convirtiendo en la nueva generación de controles industriales, los reguladores inteligentes pueden ser definidos con base al conocimiento que se tiene de sus componentes, con referencia de los reguladores convencionales.

Las redes neuronales artificiales son herramientas muy apropiadas para el modelado y propósitos de control. La habilidad de representar un mapa no lineal arbitrario es una característica especialmente interesante para problemas complejos de control no lineal. Las redes neuronales tienen la capacidad de aprender por el ajuste de parámetros internos a través de la experiencia. Tienen buenas propiedades de robustez con respecto a información incompleta y con ruido.

Las características más relevantes para involucrar redes neuronales al control son [4]: porque pueden representarse relaciones no lineales arbitrarias; aprendizaje y adaptación en sistemas desconocidos que pueden ser provistos por la adaptación de sus pesos; la información es transformada dentro de una representación interna (peso) seguida por la fusión de señales cuantitativas y cualitativas y debido a la arquitectura paralela para el procesamiento de datos. Estos métodos pueden ser usados en sistemas dinámicos de gran escala.

1.2. Objetivos de la Tesis.

Aplicar una herramienta de la inteligencia artificial basada en un modelo de red neuronal llamado "neocognitrón" para regular el voltaje de la máquina síncrona.

Comparar el regulador propuesto con un esquema de regulación convencional del tipo ST1, bajo las mismas condiciones de operación y falla, mediante la utilización de técnicas de simulación digital.

1.3. Justificación.

Los sistemas eléctricos de potencia son sistemas que presentan la característica de ser no lineales, variables en el tiempo, operan en diferentes puntos de operación y tienen cambios en sus estructuras. Se tiene la necesidad de mantener el voltaje de salida de un generador síncrono, evitando variaciones de voltaje sin importar las variaciones de carga. Uno de los principales equipos de control en una máquina síncrona es el regulador automático de voltaje y en la actualidad los que operan son normalmente sintonizados para un determinado punto de operación y algunas veces tienen que ser reajustados manualmente por personal altamente capacitado [4].



En la actualidad se están aplicando técnicas de la inteligencia artificial, llamados controles inteligentes en distintos campos de la ciencia para superar los límites de los métodos clásicos [5]. Los controles inteligentes se caracterizan por su habilidad de establecer una relación funcional entre sus entradas y salidas con datos empíricos, sin el recurso de modelos explícitos de los sistemas controlados. Al contrario de los controles convencionales, los controles inteligentes pueden aprender, recordar y tomar decisiones.

Los controles inteligentes pueden ser entrenados para operar efectivamente en condiciones de incertidumbre, pueden responder a situaciones imprevistas autónomamente, sin la intervención del operador del sistema.

La mayor parte de estas aplicaciones vienen esencialmente de las redes neuronales artificiales y de los algoritmos genéticos que principalmente son utilizados en la recuperación de información de los sistemas expertos. Las redes neuronales artificiales son herramientas muy apropiadas para el modelado y control.

Tienen la capacidad de aprender por el ajuste de sus parámetros internos a través de la experiencia, tienen buenas propiedades de robustez con respecto a información incompleta y con ruido, ofreciendo un acercamiento atractivo para el cálculo de procesos en el campo y para el modelado de plantas [6].

Por lo anterior, se desarrolla un sistema de control con el uso de técnicas de la inteligencia artificial, para regular el voltaje de la máquina síncrona que permita hacer un estudio detallado, confiable y satisfactorio del comportamiento de este tipo de técnicas.

1.4. Antecedentes del Problema.

Durante los últimos veinte años, se ha hecho un gran esfuerzo para conseguir nuevos desarrollos y técnicas de control no convencionales que puedan aumentar o reemplazar las técnicas de control convencionales. Un número de técnicas de control no convencionales han evolucionado, ofreciendo soluciones a muchos problemas del control en la industria. Esto es la esencia de lo que ha sido llamado el control práctico [7], es una colección de técnicas que los ingenieros de campo han encontrado eficazmente y fácilmente para usar en el campo.

El acercamiento a la base de reglas en el control inteligente ha sido introducido por Mamdani en 1975, usando lógica difusa en combinación con reglas para el control de un horno de cemento. El área del control difuso está basada en el trabajo de Zadeh (1965). La tecnología de la computación moderna lleva un reavivamiento renovado de la investigación de la inteligencia artificial y a la introducción de sistemas expertos. Estos sistemas expertos estimulan la investigación dentro de la aplicación de estos sistemas, incluyendo aplicaciones de control [8].

Recientemente, ha habido un gran resurgimiento por el interés de aplicar redes neuronales en el modelado de sistemas dinámicos y en el control de procesos en tiempo real. Ellos han sido usados con éxito en numerosas áreas, incluyendo a estas en el control de trenes, robots, aparatos electrónicos (Narendra, 1990), en sistemas de guías para vehículos (Nguyen, 1990), en el modelado y simulación de plantas (Griño & Aynsley, 1991), en el control y modelado de plantas químicas (Bhat, 1990).



Las aplicaciones de las redes neuronales en sistemas eléctricos de potencia han sido desarrolladas en diferentes áreas, como: en planeación, evaluación de la seguridad, detección de fallas, diseño, análisis, protección y control [9].

El mejorar la estabilidad de un sistema eléctrico de potencia es uno de los aspectos de mayor importancia en el control de los sistemas eléctricos de potencia. En 1996 se desarrolló un sistema de regulación con técnicas de inteligencia artificial por investigadores en la Universidad de Singapur y de la República de China, este sistema se basa en el control robusto de un modelo de máquina síncrona bus infinito linealizado [10].

Algunos antecedentes con los que cuenta la Sección de Estudios de Posgrado en el uso de inteligencia artificial aplicados al control de sistemas y otras aplicaciones son:

- M. C. Raúl Cortés Mateos. Tesis de Doctorado: "Control de Excitación Difuso de un Generador Síncrono". En 1997.
- M. C. José Alberto Gómez Hernández. Tesis de Doctorado: "Optimización de la Confiabilidad en Sistemas Eléctricos de Potencia Compuestos Utilizando Algoritmos Genéticos". En 2001.
- Ing. José Alberto Gómez Hernández. Tesis de Maestría: "Optimización de la Confiabilidad en la Transmisión en Sistemas Eléctricos de Potencia Utilizando Algoritmos Genéticos". En 1997.
- Ing. Marco Antonio Sobrevilla González. Tesis de Maestría: "Diseño de un Estabilizador del Sistema Eléctrico de Potencia Utilizando Control Difuso". En 1997.
- Ing. Marco Vinicio Magallón Valderrama. Tesis de Maestría: "Sintonización de un Estabilizador de Sistemas de Potencia por Medio de Redes Neuronales". En 1998.
- Ing. Julian Mendoza Fernández. Tesis de Maestría: "Compensación por Avance de Fase Neurodifuso para la Estabilización de Sistemas Eléctricos de Potencia". En 1998.
- Ing. Gabriel Mendoza Figueroa. Tesis de Maestría: "Control de Velocidad de un Motor de Inducción Aplicando el Mapa Auto-organizable de Kohonen". En 1998.
- Ing. Norberto Hernández Romero. Tesis de Maestría: "Control Difuso de la Velocidad de un Motor de Corriente Directa". En 2000.
- Ing. Zenón Yañez Barraza. Tesis de Maestría: "Control de Velocidad de un Motor de Corriente Directa Utilizando el Modelo Difuso Takagi Sugeno". En 2001.
- Ing. Jacob Efraín Díaz Lavariega. Tesis de Maestría: "Desarrollo de un Regulador de Voltaje para un Generador Síncrono Utilizando la Teoría de Resonancia Adaptable". En 2001.



- Ing. Jorge Landa Hernández. Tesis de Maestría: "Control Neuronal del Sistema de Excitación de un Generador Síncrono". En 2001.
- Ing. Hugo Ruiz Rosas. Tesis de Maestría: "Control Difuso Autosintonizable de voltaje por medio de la Búsqueda por Tabú para un Generador Síncrono". En 2002.
- Ing. Manlio Altamirano González. Tesis de Maestría: "Control Difuso de velocidad tipo Mamdani de un Motor de CD Autosintonizable por Algoritmos Genéticos". En 2002.
- Ing. Celso Gerardo Mendoza Hernández. Tesis de Maestría: "Control Difuso de velocidad de un Motor de Corriente Directa". En 1997.
- Ing. German Ortiz Rosas. Tesis de Maestría: "Detección y Clasificación de Fallas de Sistemas de Transmisión empleando una Red Neuronal con Retropropagación del Error". En 2000.

1.5. Contenido de la tesis.

La tesis está dividida en cinco capítulos y una sección de anexos al final de la misma.

En el capítulo 1 se presenta la introducción de la tesis que envuelve el estado del arte, se presentan los objetivos, la justificación y las aportaciones que se tienen con la realización de la tesis.

Los siguientes dos capítulos conforman el contexto teórico en donde se desarrolla la investigación. El capítulo 2 se enfoca al estudio del control neuronal y control inverso proponiendo avanzadas estrategias de aprendizaje.

El capítulo 3 está dedicado a la red neuronal neocognitrón, en él se encuentra la descripción de la red y comprende el tipo de red, su arquitectura, la descripción teórica y matemática de los elementos de procesamiento del neocognitrón, los pesos y el tipo de aprendizaje de la red.

El capítulo 4 presenta la información necesaria para la regulación del voltaje de la máquina síncrona, presenta la arquitectura de la red neocognitrón utilizada para el control, el procedimiento y el esquema de entrenamiento para realizar el control. Se muestran los resultados de las simulaciones bajo distintas condiciones de operación, aplicando una falla de corto circuito al sistema para diferentes ciclos de duración. Por último, estos resultados se comparan con aquellos obtenidos mediante un esquema de regulación convencional del tipo ST1.

Finalmente, en el capítulo 5 se muestran las conclusiones del trabajo y las recomendaciones para trabajos futuros. Por otra parte, al final de la tesis se encuentran los apéndices que proporcionan mayor información sobre el modelo matemático de la máquina síncrona, una breve introducción al mundo de las redes neuronales, información sobre el cálculo del índice de funcionamiento de un sistema, así como los programas de simulación utilizados para la simulación de esta tesis.



1.6. Aportaciones.

Las aportaciones más importantes de este trabajo son:

1. Investigación y aplicación de una red neuronal neocognitrón dentro del esquema de regulación de voltaje de la máquina síncrona.
2. Se muestra la utilización del aprendizaje inverso el cual permite, entre otras cosas, la producción de la ley de control adecuada para el control apropiado del sistema.
3. Elaboración de un algoritmo para regular el voltaje en terminales de una máquina síncrona, que obtiene resultados más robustos en condiciones de falla.

Además de que se publicaron los siguientes artículos:

- √ Astudillo, Y. & Romero D. "Control de Voltaje de un Generador Síncrono utilizando una Red Neuronal Neocognitrón". IEEE RVP-AI/2004, Julio 2004.
- √ Astudillo, Y. & Romero D. "Factibilidad de Regulación de Voltaje de un Generador Síncrono con una Red Neuronal Neocognitrón". Séptimo Congreso Nacional de Ingeniería Electromecánica y de Sistemas. IPN-SEPI ESIME, Noviembre 2003.
- √ Astudillo, Y. & Romero D. "Regulación de Voltaje de un Generador Síncrono utilizando una Red Neuronal Neocognitrón". 8º Congreso Nacional de Ingeniería Electromecánica y de Sistemas. IPN-SEPI ESIME, Noviembre 2004.



Capítulo 2

Control Neuronal y Control Inverso

2.1. Introducción.

Muchas estrategias de control se estudian en círculos académicos, en la industria las estrategias de control utilizadas son, el control tipo **PID** (control proporcional integral-diferencial) y el control tipo **PI** (control proporcional integral) debido a su desarrollo y simplicidad en algunos casos [11]. Sin embargo, la meta actual es proporcionar estrategias de control alternas que jueguen la simplicidad de los **PID** y los **PI** pero que mejoren su desempeño. En esta sección se presenta una metodología simple para desarrollar controles basados en los conceptos del control inverso adaptable. El trabajo en el control inverso adaptable fue inicializado en los años 60 por Widrow, y lentamente ha ganado popularidad con la potencia de las redes neuronales.

Un control inteligente, que incluye al control neuronal, es el resultado de aplicar inteligencia computacional al control de sistemas complejos. Esta clase de sistemas de control no convencionales difieren radicalmente de los sistemas convencionales que se basan en la teoría de control clásico. Las técnicas del control inteligente están aplicándose cada vez más a problemas de control industrial y de manufacturación proporcionando un alto mejoramiento en productividad, reducción del consumo de energía y una mejora de la calidad de un producto, estos son factores de gran importancia que se requieren hoy en día [12].

Los controles siguen una estrategia de control que debe ser aplicada a una planta o sistema para satisfacer los requerimientos específicos de un diseño. Esta acción puede ser el resultado de operaciones en un grupo preespecífico de reglas de control ó del entrenamiento de las redes neuronales artificiales **ANNS** con reglas de codificación numérica para el caso de los controles neuronales. El primer objetivo es el de generar acciones de control que emparejen a las de un operador humano experto. De esta manera, un controlador neuronal puede asistir al operador humano para mantener la planta en el estado de operación nominal además de que simultáneamente compensan la inconsistencia y la no fiabilidad provocada a través de la fatiga, fastidio y difíciles condiciones de trabajo de un operador.

Indudablemente, existe un gran resurgimiento en la última década por las **ANNS** jugando un rol creciente en la ingeniería. En los últimos años, ellas se han visto como una promesa para aplicaciones de control no lineal. Esta promesa se basa en su capacidad teórica para aproximar bien esquemas no lineales continuos [13].

La aplicación de redes neuronales para el control se hace usualmente construyendo un modelo de la planta y, en base a este modelo, diseñar una ley de control. El procesamiento de información entrada-salida, desarrollado por la red, puede ser representado como un esquema algebraico no lineal. Una capacidad básica de las redes neuronales es aproximar cualquier función continua no lineal, una extensión es la de aproximar la conducta entrada-salida de sistemas no lineales: cuyo proceso de



información se describe por ecuaciones diferenciales para tiempo continuo o ecuaciones diferenciales para tiempo discreto [13].

Se intenta familiarizarse en el campo de las redes neuronales en aplicaciones para el control no lineal, esto es, desarrollar un análisis sistemático para identificación, estimación de estado y manejo de trayectorias de sistemas no lineales por medio de redes neuronales.

Las redes neuronales han tenido un notable resurgimiento, usándose con éxito en un amplio rango de aplicaciones tales como comunicaciones, análisis y síntesis de reconocimiento de modelos, identificación de sistemas y control. Estas nuevas técnicas han revolucionado el campo de la ingeniería de control, ofreciendo nuevas esperanzas para resolver muchos de los problemas de control [12].

2.2. Sistemas de Control.

Los sistemas de potencia son sistemas dinámicos y su operación es de una naturaleza estocástica. Las características de la planta son no lineales, la ganancia de la planta aumenta con la carga del generador y la potencia del sistema. También el retraso de la fase de la planta incrementa. Así, los parámetros de los controladores que son óptimos para un grupo de condiciones de operación pueden no ser óptimos para otros grupos de condiciones de operación. La configuración del sistema guarda cambios debido a las acciones de cambio en el término corto o el perfeccionamiento del sistema en el término largo.

En los sistemas de control convencional con parámetros fijos, las ganancias y otros parámetros no pueden ajustarse idealmente al problema entero de operación. Desarrollos en la tecnología digital han hecho esto factible, para desarrollar e implementar mejores sistemas de control basados en técnicas más modernas y sofisticadas del control. Sistemas de control como estabilizadores de sistemas de potencia basados en control adaptable, con redes neuronales artificiales están siendo desarrollados [14].

La aplicación de **ANNS** a las técnicas de control parece ser la mas conveniente siempre que un bien definido objetivo de control no pueda ser especificado, o si el sistema a ser controlado es complejo o su modelo matemático no esta disponible.

Comparado con la predicción, el problema de un sistema de control esta más claramente definido, debido a que el objetivo es manejar la salida de la planta o una variable particular en un punto específico. Hay dos problemas básicos en un sistema de control [14]:

1. La identificación de la planta que envuelven el modelado y el problema de estimación de parámetros.
2. El cálculo del control que envuelve el cálculo de la entrada de control optimizando un criterio predefinido.

La estrategia del control de predicción incorpora predicciones o suposiciones de la variable del controlador que relaciona los futuros valores de los parámetros del controlador. El objetivo básico es reforzar la robustez del controlador. En general, la



robustez y velocidad de respuesta son dos demandas contradictorias en un controlador. Sin embargo, en la práctica hay mayor énfasis en la robustez, que es lograda a través del conocimiento del controlador de predicción [14].

El entendimiento y estudio de cualquier planta, requiere una representación matemática o modelo de la planta. La planta puede ser un proceso de entrada-salida o de series de tiempo. El modelado envuelve la selección de las variables del proceso que serán consideradas, la selección de la clase de modelo, selección de la estructura del modelo, estimación de los parámetros del modelo y probar la validez del modelo.

Los modelos con redes neuronales tienen etapas de jerarquía o capas, donde cada etapa incorpora elementos simples de no linealidad. Estos modelos son aplicables para procesos no lineales así como casi lineales.

2.3. Sistemas de Control con Retroalimentación.

El empleo de la retroalimentación en un sistema de control es una parte importante para que una estrategia de control tenga éxito. La presencia de la retroalimentación ofrece la posibilidad de modificar sustancialmente el comportamiento de un sistema y a menudo una posible variación en sus parámetros se muestran con resultados dramáticos [15,16,17].

La figura 2.1 es un diagrama típico de un sistema de control retroalimentado donde la planta representa el sistema dinámico a ser controlado y el controlador emplea una estrategia específica para llevar a la planta a un estado deseado. El vector $x(t)$ representa el vector de las variables de estado de la planta, usualmente gobernadas por un conjunto de ecuaciones diferenciales que caracterizan el desempeño de la planta. Muchas veces, algunas de estas variables no pueden medirse directamente. Por esa razón en ocasiones la señal de error, $e(t)$, producida por la diferencia entre la salida actual de la planta y la señal deseada, es utilizada por el controlador para llevar a la planta al estado deseado. En este caso, la variable de estado que sí es observable es la salida de la planta representado por el vector $y(t)$ la cual es igual al estado $x(t)$ tal como se muestra en la figura siguiente.

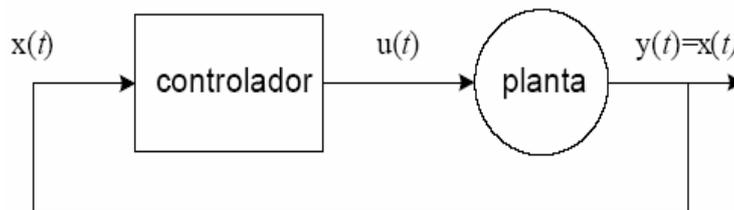


Figura 2.1. Diagrama de bloques de un sistema de control retroalimentado.

La ecuación en espacio de estado para una planta no lineal se expresa de la siguiente forma matricial:

$$y(t) = f(x(t), u(t)) \quad (3.1)$$

Donde $u(t)$ es la señal de entrada a la planta en el tiempo t . El objetivo del control es el de encontrar una función $\Phi(\cdot)$ que transforme la entrada $x(t)$ (o el error entre la



señal de referencia y la salida) de la planta en una acción de control $u(t)$, tal que $u(t)=\Phi(x(t))$, de tal forma que la salida de la planta $y(t)$ pueda seguir a una señal de referencia deseada $y_d(t)$ lo más cerca posible. Si no se conoce f , entonces es necesario identificar el sistema para encontrar un modelo apropiado de la planta. Más aun, si f es variante en el tiempo [17,18,19], es deseable hacer que $\Phi(\cdot)$ sea adaptable para que el controlador responda de una manera favorable ante las características cambiantes de la planta.

2.4. Control Neuronal.

Las ANNS exhiben la propiedad de paralelismo masivo, robustez y tolerancia a fallas, propiedades que las hacen ideales para aplicaciones de control. Además de que poseen la habilidad de aprender de la experiencia en lugar de los modelos, la habilidad de generalizar cualquier función no lineal arbitraria entre sus entradas y salidas, una arquitectura distribuida que es eminentemente ajustada para el calculo paralelo, una estabilidad inherente, la habilidad de absorber nuevo conocimiento sin destruir la existencia del conocimiento y la habilidad de aprender fuera de línea o en línea, a pesar de disturbios en el proceso [15].

Las propiedades que hacen a las **ANNS** particularmente aplicables al control son las siguientes: ser no lineales por naturaleza, ellas están eminentemente preparadas para el control de plantas no lineales; ellas son directamente aplicables al control multivariable; son inherentemente tolerantes a fallas debido a su estructura paralela y se enfrentan a nuevas situaciones teniendo la habilidad de generalizar.

Estas propiedades satisfacen los requerimientos fundamentales para su uso en los esquemas de control [12,15]. Es necesario enumerar las características básicas para el control neuronal:

- √ Sistemas no lineales: las redes neuronales artificiales son directamente aplicables al control no lineal como una consecuencia de su habilidad para reproducir cualquier esquema arbitrario de entrada-salida.
- √ Procesamiento paralelo: las redes neuronales artificiales tienen una estructura inherente, por eso permiten alta velocidad computacional. La estructura paralela implica que los controladores neuronales tienen alta fiabilidad y tolerancia a fallas que los controladores convencionales.
- √ Aprendizaje y adaptación: las redes neuronales artificiales pueden ser entrenadas con datos operacionales y pueden generalizar cuando se someten a causas que ellas no fueron entrenadas.
- √ Sistemas multivariables: las redes neuronales artificiales tienen la habilidad para procesar múltiples entradas y generar múltiples salidas simultáneamente, haciéndolas ideales para el control inteligente multivariable.

Desde el punto de vista del control, la habilidad de las redes neuronales para cubrir con fenómenos no lineales es particularmente significativa. Las **ANNS** pueden ser usadas con ventaja en el diseño de controladores no lineales para plantas no lineales, particularmente aunque el diseño sea el resultado del aprendizaje.



Las redes neuronales ofrecen algunas de las más versátiles maneras de modelar plantas no lineales de una naturaleza diversa. La relación no lineal entre entradas y salidas es modelada usando un número de bloques básicos, llamados neuronas o nodos. Los nodos están interconectados y son usualmente arreglados en múltiples capas. Cada unión de nodos internos o interconexión es un peso. En cada nodo, las entradas de los pesos son sumados junto con un umbral, y el resultado es pasado a través de una función no lineal conocida como función de activación, que forma la salida del nodo. La no linealidad asociada con cada nodo permanece fija. Los pesos en las interconexiones son estimados iterativamente por un método de optimización no lineal usando grupos conocidos de datos de entrada y salida; tal adaptación de la red es conocida como el entrenamiento o el aprendizaje de la red.

La principal característica es la arquitectura de la red, la no linealidad asociada con los nodos y el método de entrenamiento usado. El aprendizaje está basado en la minimización del error entre la salida calculada y la deseada. La regla básica de corrección del error involucra el ajuste de los pesos en las interconexiones en proporción al error entre la salida calculada y el valor deseado de cada nodo en la capa de salida.

Por otra parte, es importante mencionar que un controlador neuronal puede, en todos los esquemas, ser parametrizado a través de la estructura de un controlador convencional [12]. Los controles neuronales basados en redes neuronales multicapas han ido demostrando alto potencial en la rama no convencional para el control de procesos adaptables.

El neurocontrol es atractivo para la ingeniería de control debido a que las redes neuronales artificiales son esencialmente sistemas adaptables capaces de aprender como desarrollar tareas complejas y a que las técnicas de neurocontrol son capaces de superar muchas de las dificultades que las técnicas de control convencional sufren cuando tratan con plantas no lineales o plantas con estructuras desconocidas [20].

Los controles neuronales basados en estructuras neuronales se han propuesto para aprender la dinámica inversa de la planta de control desde la observación de la conducta de entrada-salida a través del tiempo.

2.5. Control Inverso, Aprendizaje Inverso.

El control de procesos es frecuentemente complejo, envuelve parámetros múltiples, dinámicos de plantas altamente no lineales, y combinaciones de mecanismos retroalimentados directos e indirectos. La habilidad de las redes neuronales para generalizar y aproximar ecuaciones de plantas y coeficientes proporciona la motivación para investigar su uso como controladores, particularmente cuando un modelo desconocido que describe una planta es complejo y los datos de entrenamiento son abundantes [21].

El control inverso ofrece una solución al problema de control adaptable. El controlador aproxima la inversa del modelo de la planta, en otras palabras el controlador transforma la secuencia de entrada a una secuencia la cual cuando se aplica a la entrada de la planta produce una secuencia de salida de la planta idéntica a la entrada del controlador.



El desarrollo del aprendizaje inverso, también conocido como aprendizaje general, para diseñar controles neuronales o neuro-difusos envuelve dos fases. La primera es la fase de entrenamiento, donde se utiliza la red neuronal para identificar el modelo inverso de la planta y puede ser a través de una técnica en línea o fuera de línea. Así el modelo neuronal obtenido, que representa la dinámica inversa de la planta, es entonces usado para generar las acciones de control en la fase de aplicación. Esas dos fases pueden proceder simultáneamente si se decide que el aprendizaje sea en línea [22,23].

Considerando que se cuenta con el modelo de la planta y que las variables de estado son observables, se tiene:

$$x(k+1) = f(x(k), u(k)) \quad (3.2)$$

Donde $x(k+1)$ es el estado en el tiempo $k+1$, $x(k)$ es el estado en el tiempo k y $u(k)$ es la señal de control en el tiempo k . Similarmente, el estado en el tiempo $k+2$ puede escribirse como:

$$x(k+2) = f(x(k+1), u(k+1)) = f(f(x(k), u(k)), u(k+1)) \quad (3.3)$$

En general se tiene:

$$x(k+n) = F(x(k), U) \quad (3.4)$$

La ecuación anterior señala el hecho de que dado la entrada de control u desde el instante k hasta $k+n-1$, el estado de la planta se moverá desde $x(k)$ hasta $x(k+n)$ en exactamente n veces. Más aun, si se asume que la inversa de la planta existe, entonces U puede expresarse como una función explícita de $x(k)$ y $x(k+n)$:

$$U = G(x(k), x(k+n)) \quad (3.5)$$

Esta ecuación esencialmente dice que ahí existe una única secuencia de entrada U , especificada al transformar G , que puede conducir la planta desde el estado $x(k)$ a $x(k+n)$ en n pasos de tiempo. El problema es ahora como encontrar la función G , misma que será aproximada al utilizar la red neuronal para identificarla de manera inversa [23].

La figura 2.2(a) muestra el bloque de la planta en la cual la salida de la planta $x(k+1)$ esta en función del estado previo $x(k)$ gracias al bloque de retardo Z^{-1} y la entrada $u(k)$; la figura 2.2(b) es el diagrama de bloque durante la fase de entrenamiento; la figura 2.2(c) es el diagrama de bloque durante la fase de aplicación.

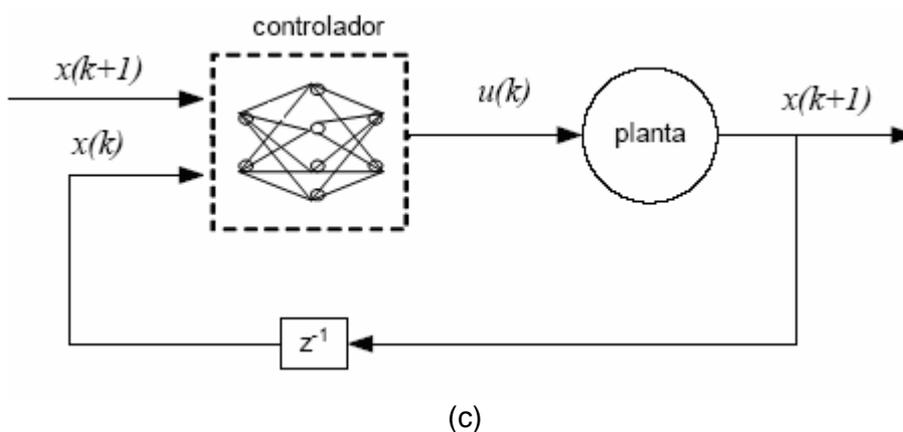
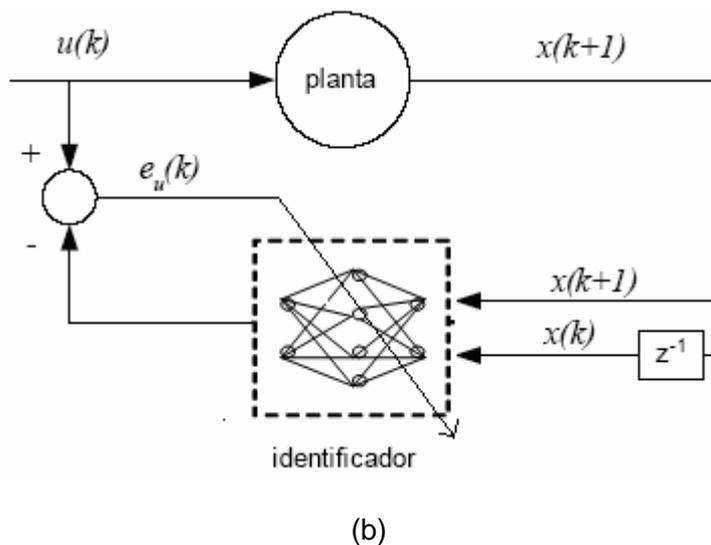
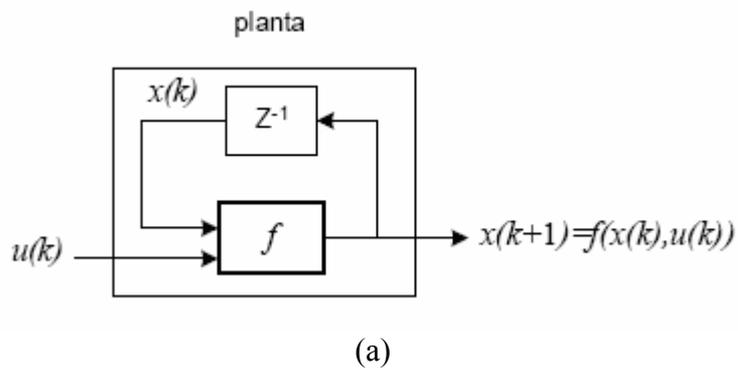


Figura 2.2. Diagrama de bloques para el método de aprendizaje inverso: (a) bloque de la planta, (b) fase de entrenamiento, (c) fase de aplicación.



La figura 2.2 muestra como una red neuronal se utiliza en el aprendizaje inverso para identificar la inversa de la planta, donde la entrada de la red es la salida de la planta. El error que se genera al comparar la salida de la red con la entrada de la planta se utiliza para estimar los parámetros de la red vía el algoritmo de aprendizaje. Una vez que la red ha identificado el modelo inverso de la planta, ahora se emplea como controlador [24]. La señal de referencia indica la salida deseada de la red y se utiliza como entrada del control neuronal así como el estado actual de la planta. La salida del control neuronal corresponde a la acción de control correspondiente. Debido a que la red ha aprendido el modelo inverso de la planta, las acciones de control la conducen al estado deseado asumiendo que la función de la red adaptable G es exactamente la misma que el modelo inverso [25]. En caso contrario la secuencia de control U no podrá llevar a la planta al estado $x(k+1)$ o bien al estado deseado.

Durante la fase de entrenamiento de la red neuronal, cuando se desarrolla fuera de línea con conocidos grupos de entrenamiento, el objetivo es establecer la relación inversa entre la salida y la entrada de la planta, el entrenamiento de la red se basa en medir el error del sistema entre la salida deseada y la salida actual del sistema. Para aplicaciones fuera de línea, tenemos que recolectar un grupo de pares de datos de entrenamiento mediante el uso de valores aleatorios como entrada y así entrenar la red adaptable observando la salida correspondiente de la red. Para aplicaciones en línea las acciones de control son generadas cada instante de tiempo mientras el aprendizaje en línea ocurre. Alternativamente, podemos generar la secuencia de control en cada paso de tiempo y aplicar solo el primer componente para la planta [23]. La figura 2.3 es un diagrama de bloques del control neuronal con aprendizaje inverso en línea, en donde los dos bloques de las estructuras neuronales son exactamente duplicados uno de otro, es decir conservan idénticas características.

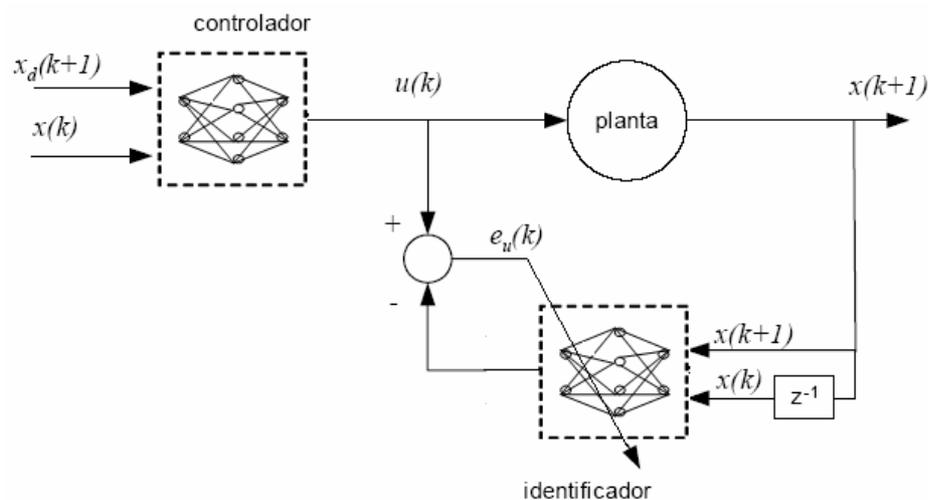


Figura 2.3. Esquema del control neuronal con aprendizaje inverso en línea.



Ahora bien, la versión de entrenamiento fuera de línea del esquema de control inverso adaptable del modelo de referencia se muestra en la figura 2.4 [26]. La entrada para el modelo de referencia es $ref(k+1)$.

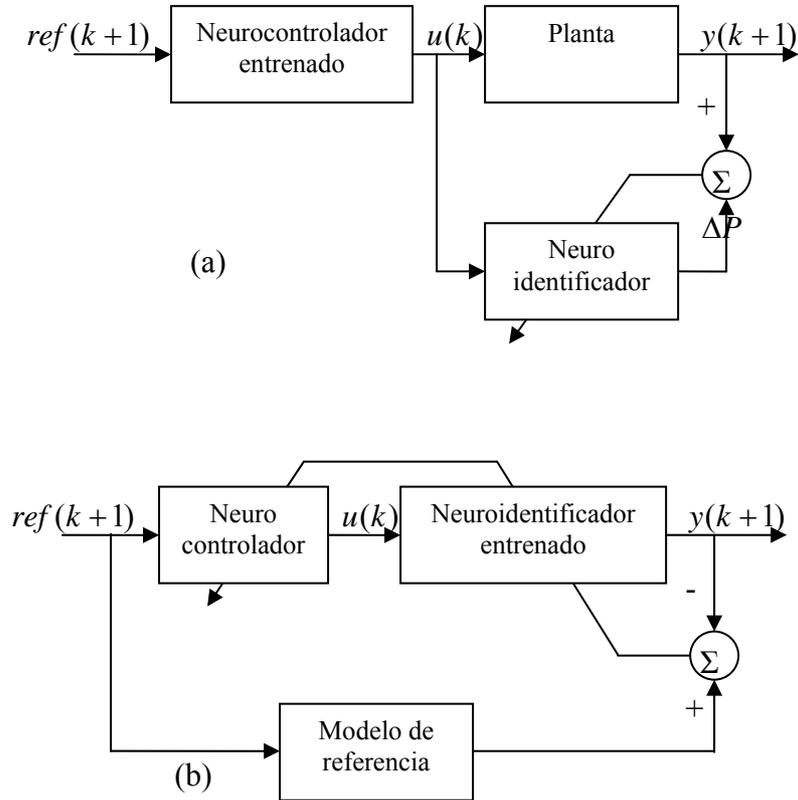


Figura 2.4. Control inverso adaptable del modelo de referencia, proceso fuera de línea. (a) entrenamiento del identificador, (b) entrenamiento del controlador.

El objetivo de este esquema de control es encontrar la acción de control $u(k)$, la diferencia entre la salida real de la planta $y(k+1)$ y la salida del modelo de referencia $ref(k+1)$ lo más pequeño posible. Aquí la planta se asume desconocida y no lineal haciendo el esquema de las dos redes necesario para actualizar el neurocontrolador. El modelo libre basado en la red neuronal es aplicado para implementar el neuroidentificador y el neurocontrolador.



Capítulo 3

El Neocognitrón

3.1. Introducción.

Las arquitecturas de un sistema neuronal artificial muestran como la retropropagación tiende a tener una aplicación general. Podemos utilizar un solo tipo de red en diferentes aplicaciones cambiando el tamaño de la red, sus parámetros y entrenando los sistemas. En contraste, el desarrollo del neocognitrón es preciso para adaptarse a una arquitectura de una aplicación específica: como lo es el reconocimiento de caracteres manuscritos y para aplicaciones con propósito de control, mostrando que en un sistema, esta tiene un gran número de aplicaciones prácticas. Kunihiko Fukushima y su compañero de trabajo Sei Miyake tuvieron interés en el desarrollo de un modelo neuronal llamado neocognitrón. Sus diseños se basaron en el trabajo de Hubel y Weisel aclarando algunas de las arquitecturas funcionales de la corteza visual [27].

La red neuronal denominada "neocognitrón" fué desarrollada por Kunihiko Fukushima con la asistencia de Sei Miyake alrededor de los años 70 y 80 [28,29]. El neocognitrón es un diseño de red multicapas, que consiste de conexiones en forma de cascada de capas de células. La información del diseño estimulado, que se da en la capa de entrada se procesa paso a paso a través de las siguientes fases de la red. Las conexiones entre las neuronas son actualizadas por un método de aprendizaje supervisado. Después del entrenamiento, las neuronas en la capa más alta tienen la tendencia a responder selectivamente a algunas características complicadas a pesar de variaciones en las mismas características presentadas.

El neocognitrón de Fukushima ha recibido atención durante la última década, como un clasificador de distorsión tolerante. Es una de las más complejas estructuras a simular de las redes neuronales artificiales, y ésta es quizás la razón principal que su desarrollo no se ha estructurado en la magnitud de otras redes neuronales artificiales, comparadamente con redes simples [30]. El neocognitrón tiene parámetros ajustables y los artículos que detallan su aplicación en los problemas prácticos, frecuentemente omiten información sobre los valores de los parámetros empleados. Esto puede llevar al investigador a asumir que el neocognitrón no es suficientemente robusto para realizar bien un desarrollo con un grupo de rangos de parámetros, pero esto no es necesariamente así.

El neocognitrón es una red neuronal con jerarquía para el reconocimiento de diseños o señales, es considerablemente robusta para distorsión y/o traducción de diseños [29]. Realiza un proceso de extracción de rasgos mediante filtrados sucesivos del modelo de entrada hacia las capas conectadas en cascada. Cada capa extrae las características desde la salida de otra capa precedida.



3.2. ¿Qué es el Neocognitrón?

El neocognitrón es una red neuronal multicapas con jerarquía, propuesta por el profesor Kunihiko Fukushima para el reconocimiento de caracteres manuscritos. Existen muchas versiones del neocognitrón. La principal ventaja del neocognitrón es la habilidad de reconocer correctamente no solo modelos de aprendizaje sino también modelos de movimiento parcial, rotación y otro tipo de distorsión [27,30].

3.3. Arquitectura del Neocognitrón.

El diseño del neocognitrón envuelve un modelo llamado cognitrón, y hay varias versiones del mismo neocognitrón [27,30]. La única que se describe tiene capas con elementos de procesamiento (PEs) llamados células, incluyendo la capa llamada capa de entrada o en algunos casos también llamada capa 0.

La estructura del neocognitrón surge de una jerarquía de extraer rasgos. La red contiene una etapa etiquetada como etapa 0, que no es usada, en contraste con las etapas más altas, para la extracción de rasgos. El número total de etapas del neocognitrón depende en la complejidad del modelo a reconocer.

La etapa 0 del neocognitrón contiene una capa de entrada propuesta para retener el modelo presentado a la red. En las etapas más altas del neocognitrón se desarrolla la extracción de rasgos. Las etapas más altas consisten de cuatro tipos de capas: la capa S compuesta por células S cuya función es la extracción de rasgos; La capa C compuesta por células C propuesta para extraer rasgos y asegurar una tolerancia de cambios de los rasgos extraídos en la capa S; La capa V compuesta por células V cuya función es obtener información acerca del área de conexión de una célula C; Y la capa de salida (es importante mencionar que está es una capa de células C) cuya función es presentar el resultado final del proceso de extracción de la red. La figura 3.1 muestra la estructura del neocognitrón y los componentes de este tipo de red.

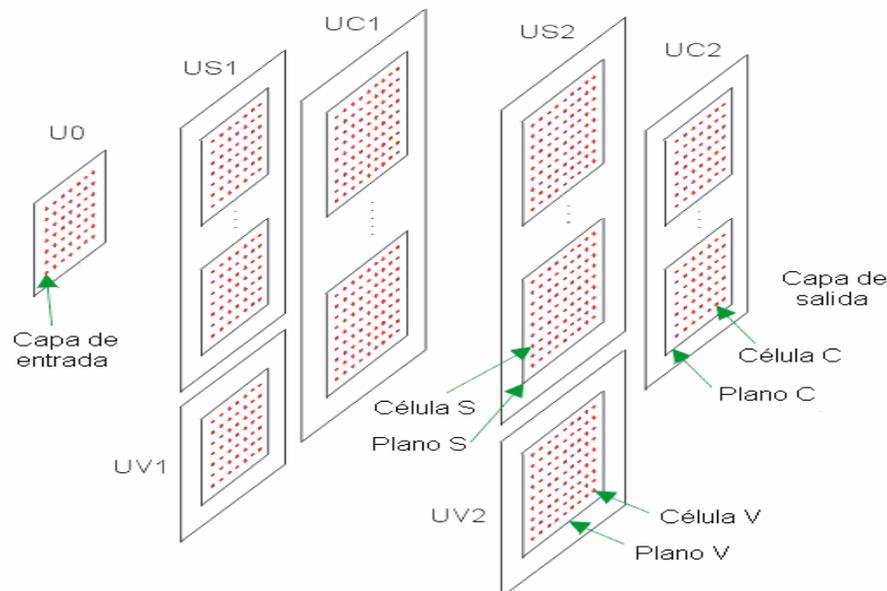


Figura 3.1. Muestra la arquitectura de la red neuronal neocognitrón.



3.4. Descripción de las Funciones de los Elementos de Procesamiento ó Células del Neocognitrón.

Los PEs del neocognitrón están organizados dentro de módulos que se pueden referir como niveles. Cada nivel consiste de dos capas: una capa de células simples o células *S*, seguida por una capa de células complejas o células *C*. Hay también PEs llamados células *V*, esos elementos pueden ser omitidos o mantenidos constantes en el procesamiento, y se puede describir la funcionalidad del sistema sin hacer referencia de ellos. Cada capa consiste de un cierto número de planos de célula del mismo tipo. La capa de entrada es la excepción de esta regla, es decir que para la capa de entrada el término plano de célula no está establecido. El número de planos de célula en cada capa *S* y *C* depende del número de rasgos a extraer en la correspondiente etapa de la red [31,32,33,34].

Las células denotan el elemento básico de operación. La célula es un elemento o neurona que procesa valores de entrada de acuerdo a una cierta regla y produce valores de salida. Como ya se mencionó hay cinco tipos de células elementales en la estructura del neocognitrón: célula receptora, células *S*, células *V*, células *C* y la célula de salida (célula *C*), que cada una difiere por su función que realiza en el proceso de extracción.

La célula receptora, célula de entrada ó célula *O* es el tipo más simple de célula, cuya función es solo retener los modelos presentados a la red.

La célula *S* ó célula simple, este tipo de célula tiene la función de extraer rasgos de la capa de entrada, el rasgo extraído por la célula *S* está determinado por los pesos de esta célula. Otra función de la célula *S* es evaluar salidas de una célula *C* desde una cierta área de conexión de una capa de células *C* anterior.

La célula *V*, que tiene la función de obtener información acerca del área de conexión de una célula *C*, este tipo de célula generalmente es omitida en el diseño de la red.

La célula *C* ó célula compleja, tiene la función de extraer y asegurar una tolerancia de cambio en los rasgos extraídos por las células *S*. Está célula también evalúa la salida de una célula *S* de una cierta área de conexión de una capa de células *S* anterior.

La célula de salida (una célula *C*), cuya función es presentar el resultado final del proceso de extracción de la red.

La figura 3.2 da una ilustración del modo en el cual las células están conectadas en el neocognitrón. Cada capa de células *S* actúa como un sistema de extracción de características o rasgos que usa la capa anterior como la capa de entrada.

En la primera capa *S*, las células en cada plano son sensibles a las características simples de la capa de entrada en este caso. Cada célula *S* en el plano simple es sensible a las mismas características, pero en diferentes ubicaciones en la capa de entrada. Las células *S* en diferentes planos responden a diferentes características.



Observe la pequeña diferencia entre la primera capa S y las capas subsiguientes S en la figura 3.2. Cada célula en el plano de la primera capa S recibe entradas desde una capa simple ó capa de entrada. En las capas subsiguientes, cada plano de célula S recibe entradas desde cada plano de células C inmediatamente precedidas. La situación es ligeramente diferente para el plano de células C. Típicamente, cada célula en un plano de células C examina una pequeña región de células S en un plano simple de células S.

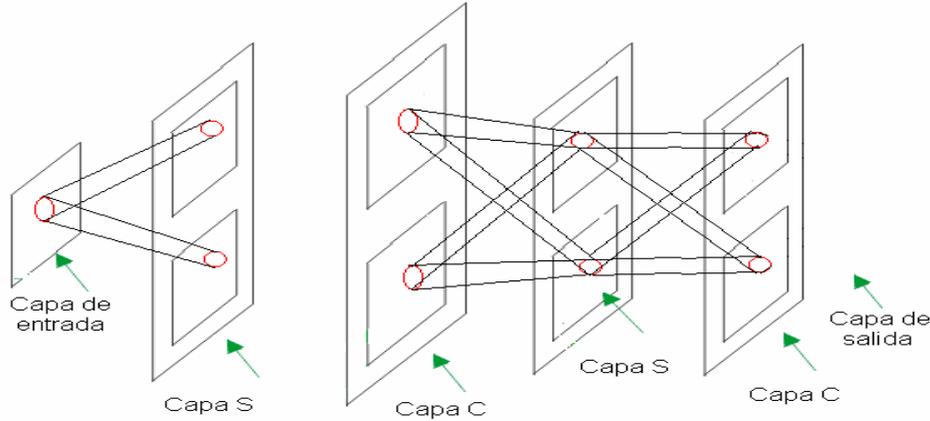


Figura 3.2. Esquema de la conexión entre las células del neocognitrón.

3.4.1. Descripción Matemática de una Célula S.

La función de una célula S es extraer un rasgo de una cierta posición en la capa de entrada [33]. Y su descripción matemática queda definida por la ecuación (3.1):

$$u_{sl}(n, k) = r_l \cdot \varphi \left[\frac{1 + \sum_{k=1}^{K_{Cl-1}} \sum_{v \in A_l} a_l(v, K, k) \cdot u_{Cl-1}(n + v, K)}{1 + \frac{r_l}{1 + r_l} \cdot b_l(k) \cdot u_{vl}(n)} - 1 \right] \quad (3.1)$$

Donde:

l	Número de serie de la etapa.
n	Posición de la célula.
k	Número de serie del plano de célula.
r_l	Selectividad.
K_{Cl-1}	Números de plano de célula en la capa C anterior.
v	Posición en el área de conexión.
A_l	Área de conexión de la célula S.
a_l, b_l	Pesos a y b (≥ 0).

$$\varphi[x] = \begin{cases} x & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$



3.4.2. Descripción Matemática de una Célula V.

El valor de salida de la célula V representa la actividad de las células en el área de conexión y se usa para inhibir la actividad de una célula C [33]. La especificación exacta de la función de una célula V se describe matemáticamente por la ecuación (3.2):

$$u_{Vl}(n) = \sqrt{\sum_{K=1}^{K_{Cl-1}} \sum_{v \in A_l} c_l(v) \cdot u_{Cl-1}^2(n+v, K)} \quad (3.2)$$

Donde:

l	Número de serie de la etapa.
n	Posición de la célula.
K_{Cl-1}	Números de plano de célula en la capa C anterior.
v	Posición en el área de conexión.
A_l	Área de conexión de la célula V.
c_l	Pesos c (≥ 0).

3.4.3. Descripción Matemática de una Célula C.

Cada célula C evalúa salidas de las células S de una cierta área de conexión [33]. La función de una célula C esta exactamente en la descripción matemática de la ecuación (3.3):

$$u_{Cl}(n, k) = \psi \left[\sum_{K=1}^{K_{Sl}} j_l(K, k) \cdot \sum_{v \in D_l} d_l(v) \cdot u_{Sl}(n+v, K) \right] \quad (3.3)$$

Donde:

l	Número de serie de la etapa.
n	Posición de la célula.
k	Número de serie del plano de célula.
K_{Cl-1}	Números de plano de célula en la capa S anterior.
j_l	Factor de conexión.
v	Posición en el área de conexión.
D_l	Área de conexión de la célula C.
d_l	Pesos d (≥ 0).

$$\psi[x] = \frac{\phi[x]}{1 + \phi[x]} \quad \phi[x] = \begin{cases} x & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$



3.5. Pesos y Conexiones en el Neocognitrón.

El neocognitrón es característico no solo por el largo número de células sino también por un largo número de conexiones. Estas conexiones sirven para transferir información entre las células en capas adjuntas. Para cada conexión hay un peso y por medio de este se puede afectar a la cantidad de información transferida entre las células.

Existen cuatro tipos de pesos en el neocognitrón (pesos *a*, pesos *b*, pesos *c* y pesos *d*). Cada uno de estos tipos de pesos es usado para conexiones entre dos capas de diferentes tipos [33,34]. Esto se muestra en la figura 3.3.

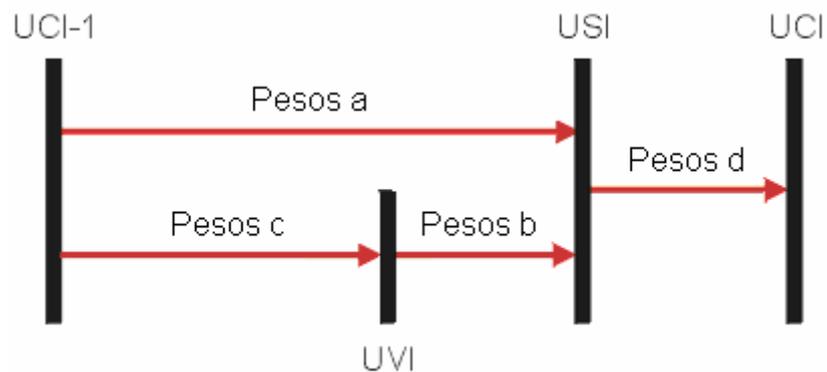


Figura 3.3. Tipos de pesos y conexiones en el neocognitrón.

3.5.1. Pesos Compartidos.

Este término se asigna al factor que todas las células en un plano de célula usan los mismos pesos para conexiones que llevan las células en sus áreas de conexión. Con los mismos pesos compartidos se garantiza que todas las células de un plano de célula siempre extrae el mismo rasgo.

3.5.2. Pesos *a* y Pesos *b*.

Los pesos *a* son el primer tipo de pesos modificados por el aprendizaje. Estos pesos son usados para conexiones entre células *S* y células *C* que pertenecen a su área de conexión. La extracción de rasgos por células *S* son codificados en estos pesos *a*. El ajuste de los pesos es desarrollado durante el aprendizaje de la red de acuerdo al modelo de entrenamiento presentado. Los pesos *b* con el segundo tipo de pesos modificados por el aprendizaje. Estos pesos son usados por conexiones entre células *S* y células *V*.

3.5.3. Pesos *c* y Pesos *d*.

Los pesos *c* son fijos y son usados para conexiones entre células *V* y células *C* en sus áreas de conexión. Los valores de los pesos *c* son determinados en la construcción de la red. Estos pesos son los más frecuentes para reducir la transferencia de información del área de conexión. Los pesos *d* también son fijos y



usados para conexiones entre células C y células S en sus áreas de conexión. Reducen la transferencia de información del área de conexión.

3.6. Aprendizaje del Neocognitrón.

El aprendizaje es el proceso durante el cual la red neuronal se adapta para resolver una tarea deseada. El aprendizaje de la red es controlado por un maestro, es decir es un aprendizaje supervisado. Esta tarea es para determinar que rasgos serán extraídos en las etapas de la red y para preparar los modelos de entrenamiento antes de comenzar el aprendizaje [33].

El aprendizaje del neocognitrón se beneficia etapa por etapa desde la etapa más baja de la red ajustando los pesos modificables (pesos a , pesos b) de acuerdo a la respuesta aprendida de la red para representar un modelo de entrenamiento [33]. La descripción matemática del aprendizaje esta definida por la ecuación (3.4):

$$\begin{aligned}\Delta a_l(v, K, \hat{k}) &= q_l \cdot c_l(v) \cdot u_{Cl-1}(\hat{n} + v, K) \\ \Delta b_l(\hat{k}) &= q_l \cdot u_{vl}(\hat{n})\end{aligned}\tag{3.4}$$

Donde:

l	Número de serie de la etapa.
\hat{n}	Posición de la célula de activación.
\hat{k}	Número de serie del plano de célula con la célula núcleo.
q_l	Coefficiente de aprendizaje.
a_l, b_l, c_l	Pesos a , b y c (≥ 0)

Hay varios métodos diferentes para el entrenamiento de los pesos en el neocognitrón. El método que se detalla aquí es un algoritmo de aprendizaje supervisado diseñado por los creadores del neocognitrón original.

Para cada plano S en la red un modelo de entrenamiento es usualmente usado y este modelo es usualmente necesario para presentar a la red. En el comienzo del aprendizaje con maestro se tiene un grupo de pesos a y pesos b en la red. La presentación del modelo de entrenamiento dado por este plano S dentro de la capa de entrada UO es el siguiente paso. Finalmente el maestro ajusta los pesos de las células de acuerdo a las ecuaciones mencionadas en la descripción matemática de aprendizaje.

En principio, un patrón de entrada es presentado en la capa de entrada y el dato es propagado a través de la red [33]. Entonces los pesos son permitidos para hacer incrementos con ajustes de acuerdo al algoritmo de optimización específico. Después la actualización de los pesos tiene que ocurrir, un nuevo patrón es presentado en la capa de entrada, y el proceso es repetido con todos los patrones en el entrenamiento dentro de la red, clasificando las propiedades de los patrones de entrada.



Con este modelo, al aplicar un patrón de entrada a la red se examina la respuesta de las células S en cada capa. Para asegurar esto cada célula S provee distintas respuestas, podemos inicializar los pesos a_i en pequeños y positivos valores de redondeo. Los pesos b_i en las conexiones inhibitorias pueden ser inicializados en cero [33,34].



Capítulo 4

Regulación de Voltaje de la Máquina Síncrona Usando la Red Neuronal Neocognitrón

4.1. Introducción.

Los sistemas de potencia son sistemas dinámicos y su operación es de una naturaleza estocástica. Las características de estos sistemas son no lineales, la ganancia de estos aumenta con la carga del generador y la potencia del sistema. Así, los parámetros de los controladores que son óptimos para un grupo de condiciones de operación pueden no ser óptimos para otros grupos de condiciones de operación. En los controladores convencionales con parámetros fijos, las ganancias y otros parámetros no pueden ajustarse idealmente al problema entero de operación. Desarrollos en la tecnología digital han hecho esto factible al desarrollar e implementar mejores controladores basados en técnicas más modernas y sofisticadas. Los estabilizadores de sistemas de potencia basados en control adaptable, redes neuronales artificiales, lógica difusa están siendo desarrollados. Cada una de estas técnicas posee características únicas.

La aplicación de las redes neuronales artificiales a las técnicas de control parece ser una técnica conveniente siempre y cuando no se tiene especificado un buen objetivo de control, el sistema a ser controlado es complejo o su modelo matemático no está disponible.

Debido a que el desarrollo del área de control ha tenido un gran resurgimiento por la necesidad de trabajar con sistemas complejos, la necesidad de cumplir con requerimientos de diseño y la necesidad de alcanzar estos requerimientos con poco conocimiento de los sistemas a controlar [7,35] y a que durante los últimos años se han hecho grandes esfuerzos para conseguir nuevas técnicas y desarrollos de control del tipo no convencional que permitan aumentar o reemplazar las técnicas de control convencional [7,35], se desarrolla una técnica de control que nos permite regular el voltaje de la máquina síncrona, esta técnica de control involucra directamente al aprendizaje inverso descrito en la sección 2.5 del capítulo 2 y a la red neuronal neocognitrón cuyas características se definen en todo el capítulo 3.

La complejidad creciente de los sistemas eléctricos de potencia y la demanda impuesta por economía y los requerimientos operacionales han dado énfasis a la necesidad para continuas mejoras en desarrollar plantas de potencia y control. Los sistemas de potencia son plantas no lineales y operan sobre un rango amplio de operación. Ellos están sujetos a cambios de carga al azar, una variedad de diferentes puntos de operación y diferentes configuraciones [36,37]. Por eso, se desea desarrollar un controlador inteligente adaptable el cual tenga la habilidad para ajustar estos parámetros. Para el propósito del control adaptable de plantas no lineales, una



fase de identificación es requerida para proporcionar mejora continua de la exactitud de identificación y la habilidad para rastrear variaciones de la planta cuando ellas ocurran. Y una fase de control que opere según lo aprendido en para ese punto de operación.

El control de excitación de un generador síncrono es uno de los aspectos más importantes para reforzar la estabilidad de un sistema de potencia y para garantizar la salida de la potencia eléctrica que este proporciona. La función de un **RAV** es ajustar continuamente el nivel de excitación de una máquina síncrona ante diferentes condiciones de operación de la máquina con el propósito de mantener el voltaje en terminales con la menor cantidad de variaciones posibles.

Cuando un **SEP** cambia su punto de operación, las máquinas síncronas deben ser reajustadas de tal forma que la energía requerida por la carga sea exactamente igual que la energía suministrada por las máquinas síncronas, esto provoca a su vez que la máquina síncrona cambie su nivel de voltaje de campo, la variación debe ser controlada por un **RAV** el cual debe cambiar el nivel de excitación para poder cumplir con la demanda de energía activa y reactiva requerida para ese nuevo punto de operación. Cuando la máquina síncrona cambia su punto de operación, el **RAV** debe ser reajustado para trabajar en ese nuevo punto de operación. Si al cambiar el punto de operación el **RAV** no es reajustado, cuando ocurren cambios abruptos de amplitud y fase de voltaje provocados por una falla, pueden repercutir en fallas parciales o totales del sistema ya que el **RAV** convencional se ve muchas veces imposibilitado para mantener la máquina síncrona dentro de un rango de operación estable [38].

La técnica inteligente de control desarrollada en esta tesis involucra que la red neuronal neocognitrón aprenda la dinámica inversa del sistema que será controlado en diferentes puntos de operación en la primera fase del control, es decir, el neocognitrón es entrenado aplicando el rango deseado de entradas de la planta, de esta forma sus correspondientes salidas pueden ser obtenidas y un grupo de modelos de entrenamiento seleccionados. Una vez entrenada la red neocognitrón, está podrá usarse para producir la correcta entrada de control como una función de la salida deseada de la planta.

4.2. Esquema Básico de un Regulador Automático de Voltaje Convencional.

El esquema básico de un sistema de potencia utilizado en este trabajo consiste de una máquina síncrona conectada a un voltaje constante (bus infinito) a través de una línea de transmisión y se muestra en la figura 4.1. El sistema es representado por un modelo no lineal incluyendo el gobernador y el regulador automático de voltaje **RAV** excitador [39].

En este esquema la función del **RAV** es ajustar continuamente el nivel de excitación de la máquina síncrona ante diferentes condiciones de operación con el fin de mantener el voltaje en terminales con la menor cantidad de variaciones posibles.

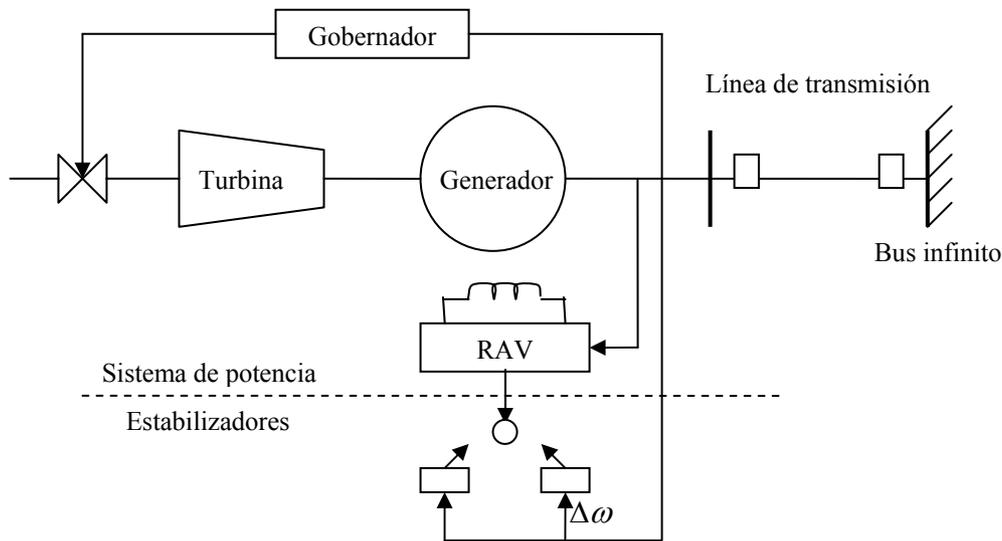


Figura 4.1. Esquema básico del sistema máquina bus infinito con un RAV convencional.

Este tipo de esquema de control presenta la característica que cuando la máquina síncrona cambia su punto de operación, el **RAV** debe ser reajustado para trabajar en ese nuevo punto de operación. Si al cambiar el punto de operación el **RAV** no es reajustado, cuando ocurren cambios abruptos de amplitud y fase de voltaje provocados por una falla, esta característica puede repercutir en fallas parciales o totales del sistema ya que el **RAV** se ve muchas veces imposibilitado para mantener la máquina síncrona dentro de un rango de operación estable. Por este motivo es necesario investigar el comportamiento del sistema incorporando un esquema de control basado en metodologías de inteligencia artificial que como ya se menciona en el capítulo 1 conservan cualidades que pueden servir a solucionar algunos problemas que los esquemas convencionales presentan.

4.3. Regulación de Voltaje de la Máquina Síncrona Usando el Neocognitrón.

En esta sección se presenta el esquema de control propuesto basado en el aprendizaje inverso que incluye a la red neuronal neocognitrón.

4.3.1. Estructura del Control Propuesto.

La estructura del control propuesto se muestra en la figura 4.2 y consiste de dos partes o fases principales: un identificador neuronal y un controlador neuronal. El identificador o estimador de estado como se menciona en el capítulo 2, sección 2.5 utiliza la red neuronal neocognitrón para construir el modelo de la planta, considerando dos casos:

La dimensión del estado de la red neuronal que coincide con un sistema no lineal, así la red neuronal se vuelve un identificador. La salida del sistema no lineal depende



linealmente de los estados. La red neuronal permite estimar el estado del sistema por medio de la implementación de un observador o identificador neuronal.

Y el controlador, basado en el modelo, implementado por el identificador neuronal o el observador, aquí se desarrolla la ley del control del mínimo local, que en cada tiempo minimiza el error con respecto a un modelo de referencia no lineal bajo la prehistoria fija de este modelo.

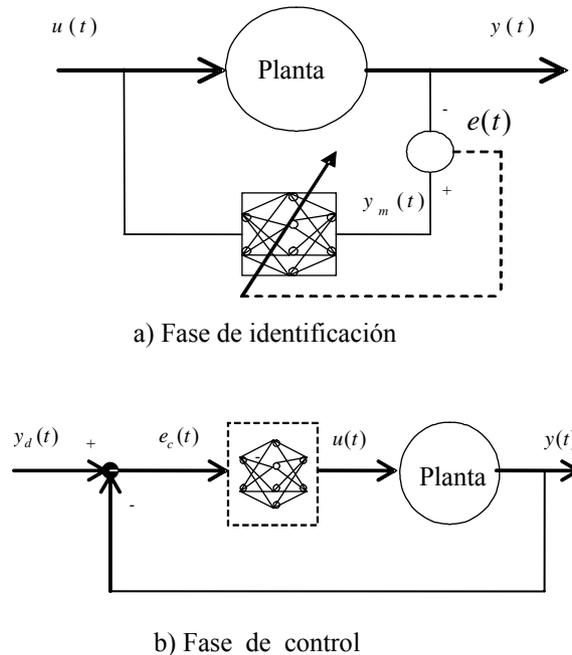


Figura 4.2. Estructura del control propuesto, procedimiento fuera de línea, a) Fase de identificación, b) Fase de control.

En la fase a) el identificador neuronal **NI** puede ser utilizado también cuando el modelo de la planta es desconocido, considerado como una caja negra, así el **NI** actúa como un modelo matemático de la planta para deducir la ley de control. La idea principal es retropropagar la trayectoria del error $e_c(t)$ a través del **NI** para evaluar el error en la entrada del control $u(t)$ y así adaptar el controlador neuronal **NC** [22].

Es importante mencionar que el control se realiza fuera de línea ya que en este tipo de control se manipula la señal de control con el error de salida de la planta para poder mantener el voltaje en terminales con la menor cantidad de variaciones posibles.

4.3.2. Procedimiento para el Control Propuesto.

Utilizando las herramientas y técnicas de control presentadas hasta esta parte de la tesis, se enlistan los pasos a seguir para llevar a cabo el control de un sistema a través de la red neuronal neocognitrón propuesta en este trabajo. Este procedimiento originalmente fue propuesto para el control con **ANNS** [22,24], el procedimiento para realizar el control neuronal se muestra a continuación:



1. La colección de datos de entrenamiento a través de la experimentación. La idea es variar la entrada o entradas del sistema $u(t)$, y observar el impacto sobre la salida, $y(t)$, figura 4.2 (a).

2. Establecer la estructura de la red neuronal en lo referente al número de entradas y salidas, número y tipo de células S y C para el caso del neocognitrón, figura 4.3.

3. Identificación del modelo inverso de la planta a través del algoritmo de aprendizaje el cual puede ser en línea con la planta o fuera de línea, como se menciona en la sección 4.3.1 es fuera de línea, figura 4.2.

4. Aplicación de la red en el esquema de control. Usando el modelo inverso de la planta, la red neocognitrón como controlador será capaz de generar las acciones de control que lleven a la planta al estado deseado.

Por lo descrito en el procedimiento anterior, el control neuronal se reduce a una identificación de procesos junto con una validación que en este caso se presenta en la fase de control. Es muy importante señalar que para asegurar que la red como controlador tenga un desempeño aceptable dentro del sistema, es necesario asegurarse que en la fase de control, el controlador opere con las mismas condiciones con las que fue entrenado, es decir que no se podrían esperar buenos resultados si el controlador tuviera señales de entrada diferentes a las que tuvo en la fase de entrenamiento, o bien que tuviera una estructura diferente.

4.3.3. Arquitectura del Neocognitrón Utilizada para Regular el Voltaje de la Máquina Síncrona.

La figura 4.3 muestra la arquitectura del neocognitrón propuesta para el control del voltaje de la máquina síncrona bajo el esquema de control neuronal mostrado en la figura 4.2.

La estructura del neocognitrón propuesta consta de una capa de células de entrada, dos capas de células S y dos capas de células C, una de estas capas C representa la capa de células de salida.

La capa de entrada como se ve en la figura 4.3 esta compuesta por dos células de entrada, una representa el error definido por la diferencia entre el voltaje de referencia y el voltaje en terminales del generador y la otra a la variación del error definido por la diferencia entre el error actual y el estado anterior de este error.

Las dos capas de células S que están compuestas por 12 células S cada una, y las dos capas de células C que una esta compuesta por 12 células C y la otra capa de células C que representa a la capa de salida compuesta por una célula C que como se ve en la figura 4.3 contiene la variable de salida representada por el voltaje de campo, ya que esta variable nos permite generar la ley de control adecuada para nuestro sistema de control.

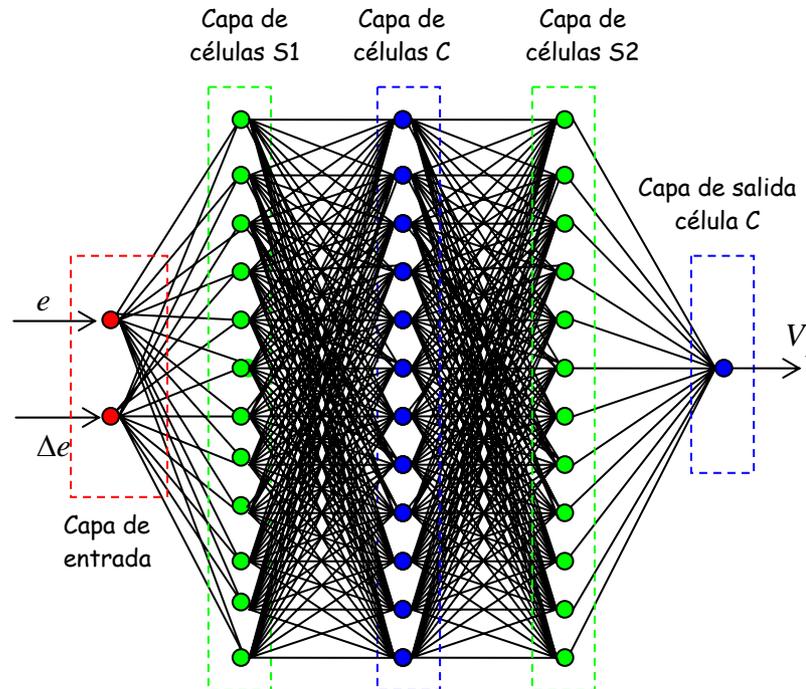


Figura 4.3. Arquitectura del neocognitrón propuesta para el esquema del control neuronal.

Como se menciona en la sección 3.3 del capítulo 3, existen varias estructuras del neocognitrón y estas pueden variar de acuerdo a los requerimientos del diseñador. En este diseño del neocognitrón la capa de células V se omite, ya que como se menciona en la sección 3.4 estas pueden ser despreciadas, no afectando el desempeño de la red.

Para el diseño del neocognitrón, la cantidad de células de entrada como la cantidad de células ocultas S , C y la cantidad de células de salida se define de forma heurística, es decir que debido a que no existe una metodología para obtener el diseño adecuado de la red neuronal para que realice una tarea específica la estructura del neocognitrón se definió a prueba y error, encontrando que la estructura del neocognitrón para efectos de la simulación mostrada en la figura 4.3 presentó un rendimiento aceptable.

4.3.4. Esquema de Entrenamiento del Neocognitrón.

El esquema de entrenamiento del neocognitrón se basa en el aprendizaje inverso descrito en la sección 2.5 del capítulo 2. Tomando en cuenta las características que presenta el neocognitrón en su esquema de aprendizaje, estas características se describen en la sección 3.6, capítulo 3.

El neocognitrón utiliza un esquema de entrenamiento del tipo supervisado, es decir, la respuesta deseada $d(t)$ es proporcionada al sistema, en este tipo de aprendizaje se proporcionan una serie de pares constituidos por entradas y salidas deseadas. Cuando cada vector de entrada $u(x)$ se proporciona al sistema, al mismo tiempo se proporciona la salida deseada $d(x)$. La señal de error definida por la



diferencia entre la salida actual $y(k)$ y la salida deseada $d(t)$ es enviada al sistema neuronal para que efectúe el ajuste o el cambio de sus pesos a través de su técnica de optimización, el neocognitrón realiza esta función mediante la ecuación 3.4 mostrada en la sección 3.6. De esta manera la salida actual del sistema es corregida para aproximarse a la salida deseada del sistema.

Para llevar á cabo el proceso de aprendizaje se tomo el procedimiento mencionado anteriormente, proporcionando como vectores de entrada a la red al error dado por la diferencia del voltaje de referencia y el voltaje en terminales de la máquina y a la variación de este error determinado por la diferencia entre el error actual y el error anterior, por lo tanto las entradas de la estructura neuronal del neocognitrón para el proceso de aprendizaje pueden resumirse como:

$$e(t) = V_{ref} - V_T(t) \quad (4.1)$$

$$\Delta e(t) = e(t) - e(t-1) \quad (4.2)$$

El vector de salida es el voltaje de campo necesario para mantener el voltaje en terminales lo más cerca del valor de referencia.

Las entradas del neocognitrón deben ser escogidas con mucho cuidado de tal forma que generen la cantidad mínima necesaria de condiciones para representar la dinámica del sistema.

4.4. Simulación de los Reguladores de Voltaje Bajo Distintas Condiciones de Operación.

Para llevar a cabo la simulación digital de los reguladores fue necesario la elaboración de un programa de cómputo en lenguaje de programación C++ para el regulador de voltaje basado en el esquema neuronal y para el regulador de voltaje convencional ST1, estos programas se pueden ver en el apéndice D.

Para efectos de la simulación el desarrollo del control se basa en un generador síncrono conectado a un bus infinito a través de una línea de transmisión equivalente, se utiliza un modelo matemático de 5º orden para la representación de la máquina síncrona, en función de los siguientes parámetros: reactancias síncronas, transitorias y subtransitorias, y constantes de tiempo transitorias y subtransitorias, ver apéndice A.

Las pruebas se realizaron con los mismos parámetros de la máquina síncrona y bajo las mismas condiciones de operación, permitiendo hacer la comparación del sistema de regulación neuronal con el sistema de regulación del tipo ST1 y así evaluar el índice de desempeño de cada regulador. Para las pruebas se grafican: el voltaje en terminales, el voltaje de campo, la potencia activa, la potencia reactiva, el ángulo de la máquina y el índice de desempeño del voltaje en terminales.

Para la aplicación de los sistemas de regulación se tomaron varios conjuntos de condiciones iniciales, se probaron para fallas de distintos tiempos de duración. El tiempo en que ocurre el corto circuito es de 1 seg., el tiempo final de la simulación es de 8 seg.



La tabla 4.1 presenta los casos analizados en esta simulación y son los siguientes:

Caso No.	Potencia activa (p.u.)	Falla (ciclos de duración)
1	0.8240	6, 9, 12
2	0.9833	6, 9, 12
3	0.5284	6, 9, 12
4	1.1000	6, 9, 12
5	0.9592	6, 9, 12

Para el caso 1. El sistema máquina bus infinito se encuentra trabajando en un punto de operación donde la potencia activa es $P=0.8240$ p.u., se muestran fallas con 6, 9 y 12 ciclos de duración para este caso.

Para el caso 2. Donde el sistema máquina bus infinito se encuentra trabajando en un punto de operación donde la potencia activa es $P=0.9833$ p.u., para este caso se analizaron fallas de 6, 9 y 12 ciclos de duración.

En el caso 3. Donde el sistema máquina bus infinito se encuentra trabajando en un punto de operación donde la potencia activa es $P=0.5284$ p.u., se muestran fallas con 6, 9 y 12 ciclos de duración para este caso.

Para el caso 4. El sistema máquina bus infinito se encuentra trabajando en un punto de operación donde la potencia activa es $P=1.1000$ p.u., se muestran fallas con 6, 9 y 12 ciclos de duración para este caso.

Para el caso 5. El sistema máquina bus infinito se encuentra trabajando en un punto de operación donde la potencia activa es $P=0.9592$ p.u., para este caso se analizaron fallas de 6, 9 y 12 ciclos de duración.

4.4.1. Caso 1. $P=0.8240$. Con una falla de 6, 9 y 12 ciclos de duración.

Las siguientes simulaciones son realizadas con una potencia activa igual a $P=0.8240$ p.u., con una falla de 6, 9 y 12 ciclos de duración, obteniéndose los resultados siguientes:

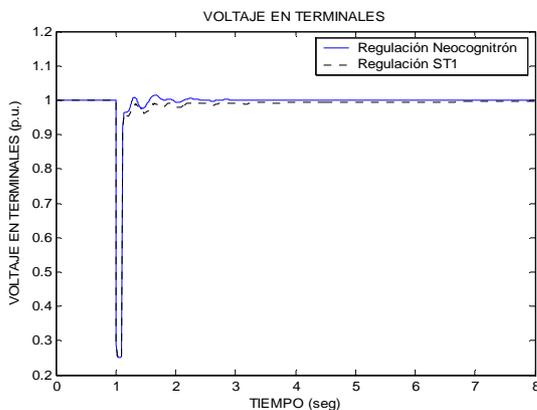


Figura 4.4. Comportamiento del voltaje en terminales ante una falla de 6 ciclos de duración.

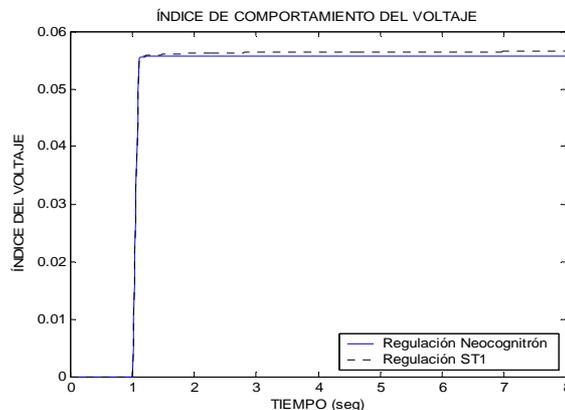


Figura 4.5. Índice de comportamiento del voltaje en terminales.

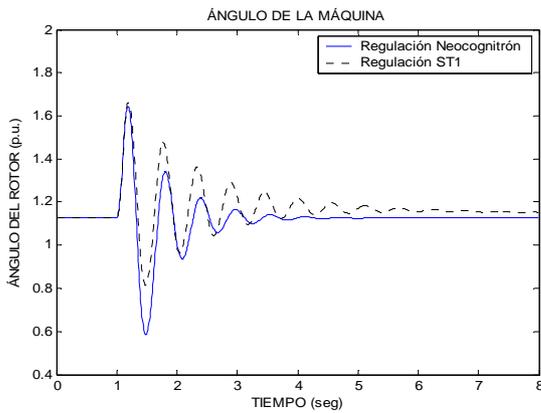


Figura 4.6. Comportamiento del ángulo de la máquina ante una falla de 6 ciclos de duración.

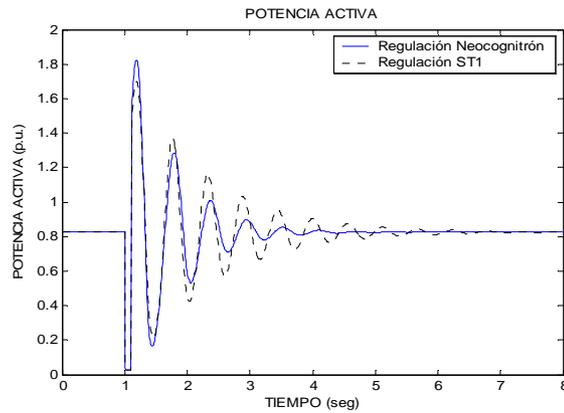


Figura 4.7. Comportamiento de la potencia activa ante una falla de 6 ciclos de duración.

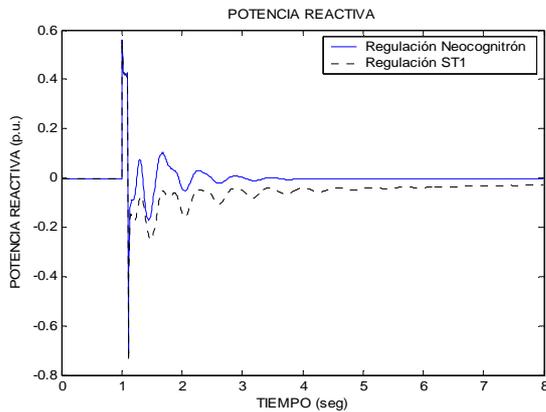


Figura 4.8. Comportamiento de la potencia reactiva ante una falla de 6 ciclos de duración.

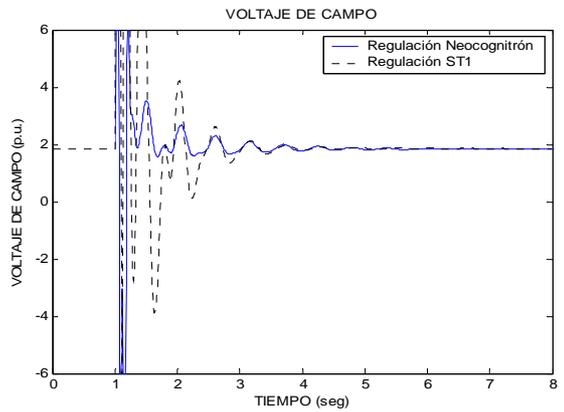


Figura 4.9. Comportamiento del voltaje de campo ante una falla de 6 ciclos de duración.

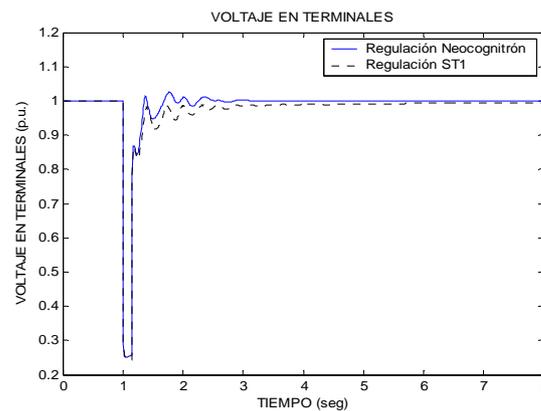


Figura 4.10. Comportamiento del voltaje en terminales ante una falla de 9 ciclos de duración.

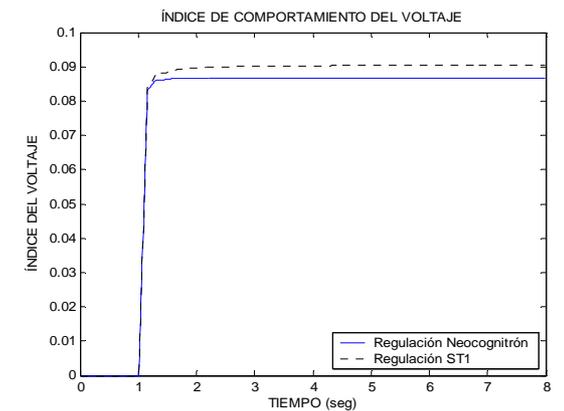


Figura 4.11. Índice de comportamiento del voltaje en terminales.

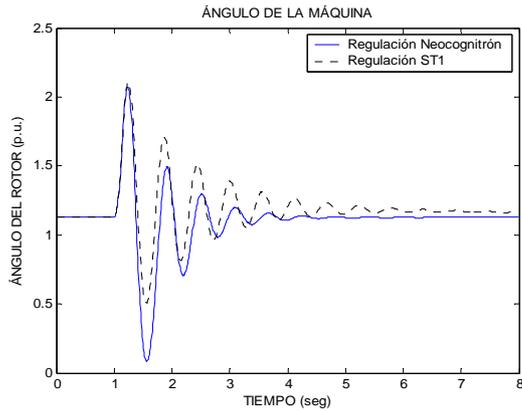


Figura 4.12. Comportamiento del ángulo de la máquina ante una falla de 9 ciclos de duración.

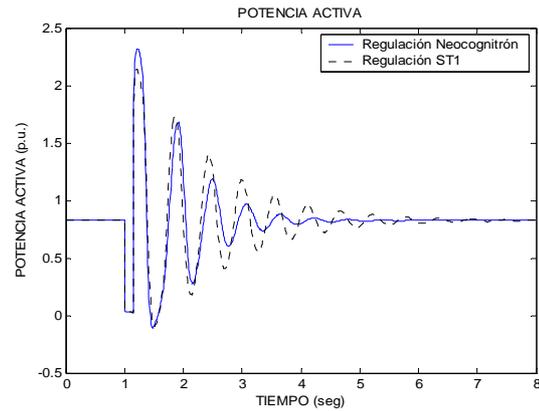


Figura 4.13. Comportamiento de la potencia activa ante una falla de 9 ciclos de duración.

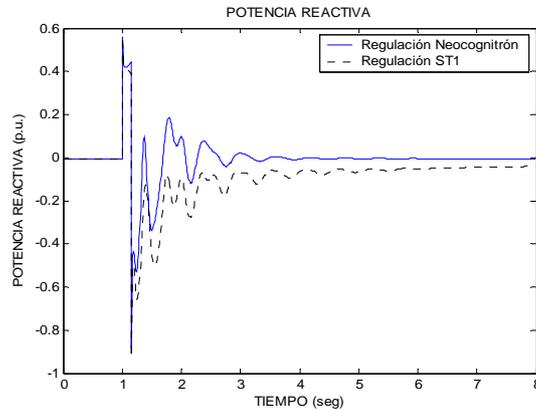


Figura 4.14. Comportamiento de la potencia reactiva ante una falla de 9 ciclos de duración.

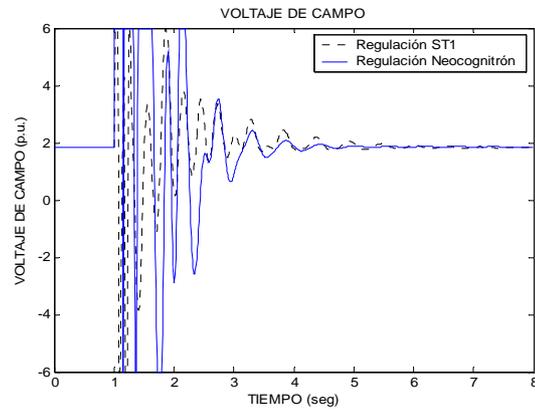


Figura 4.15. Comportamiento del voltaje de campo ante una falla de 9 ciclos de duración.

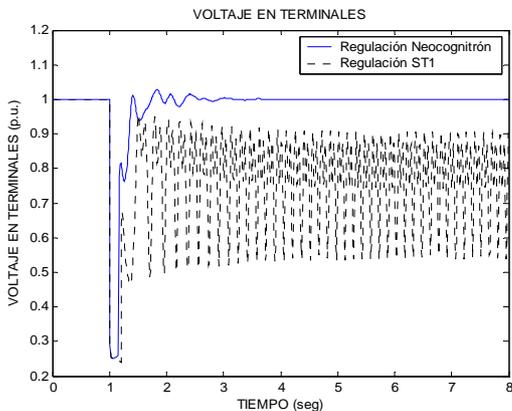


Figura 4.16. Comportamiento del voltaje en terminales ante una falla de 12 ciclos de duración.

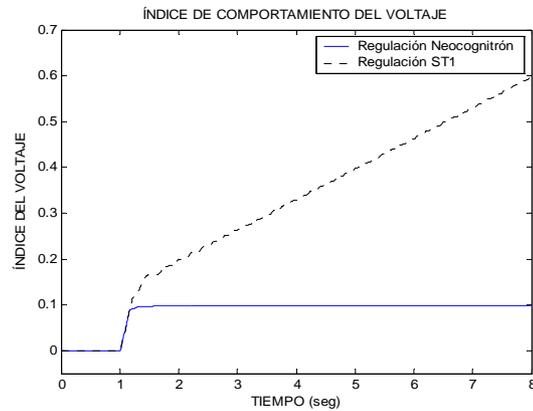


Figura 4.17. Índice de comportamiento del voltaje en terminales.

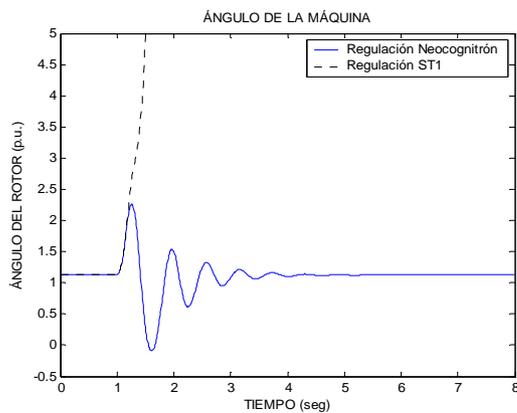


Figura 4.18. Comportamiento del ángulo de la máquina ante una falla de 12 ciclos de duración.

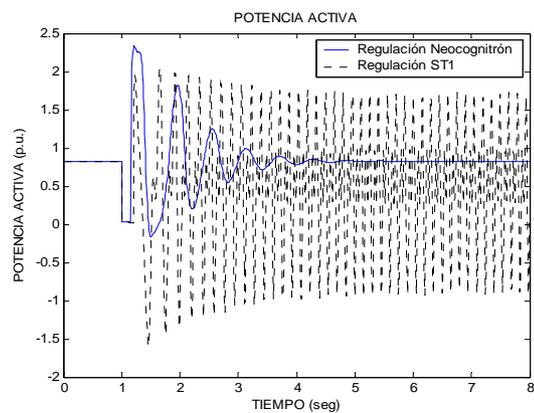


Figura 4.19. Comportamiento de la potencia activa ante una falla de 12 ciclos de duración.

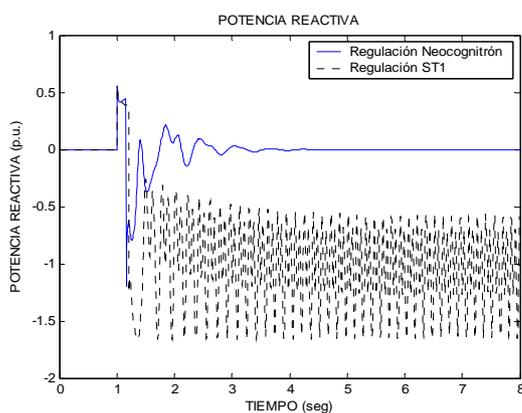


Figura 4.20. Comportamiento de la potencia reactiva ante una falla de 12 ciclos de duración.

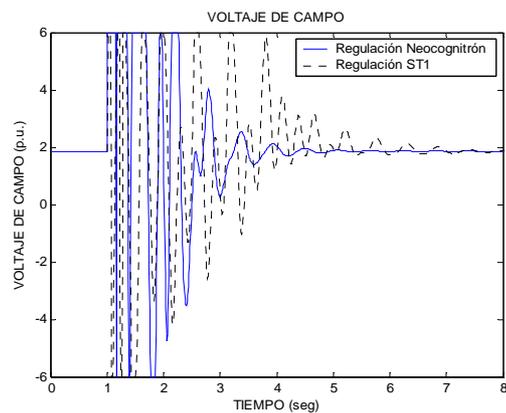


Figura 4.21. Comportamiento del voltaje de campo ante una falla de 12 ciclos de duración.

4.4.2. Análisis de resultados del caso 1.

La simulación del caso 1, para una falla de 6 ciclos de duración tiene el siguiente análisis de resultados.

En la figura 4.4 se puede observar como el voltaje en terminales con el regulador Neocognitrón tiene un mejor comportamiento en comparación con el regulador ST1. Al inicio de la simulación el sistema se encuentra en estado estacionario, después cuando se aplica la falla de corto circuito en 1.0 seg. el voltaje en terminales para el regulador Neocognitrón presenta un número menor de oscilaciones que el regulador ST1 además de que alcanza primero el estado estacionario.

El índice de desempeño de un sistema permite evaluar el funcionamiento de un sistema teniendo un mejor funcionamiento, el sistema que minimice este índice. Se puede observar en la figura 4.5 que el regulador Neocognitrón presenta un índice de funcionamiento menor que el que presenta el regulador ST1, así se permite decir que el regulador neocognitrón es mejor porque minimiza este índice.



El comportamiento del ángulo del rotor en la figura 4.6 para el regulador ST1 presenta un mayor número de oscilaciones y tarda más en alcanzar el estado estacionario con respecto al regulador Neocognitrón.

La figura 4.7 muestra como la potencia activa para el regulador neocognitrón alcanza su estado estacionario en un menor tiempo en comparación con el regulador ST1, de igual forma tiene un número menor de oscilaciones con menor amplitud que las que presenta el regulador ST1.

La figura 4.8 de la potencia reactiva presenta una mayor amplitud de oscilaciones cuando se utiliza el regulador ST1 con respecto de las alcanzadas con el regulador Neocognitrón.

El comportamiento del voltaje de campo se observa en la figura 4.9, aquí es importante mencionar que el voltaje de campo esta limitado en un rango de operación de ± 6 volts. En la figura 4.9 se observa que el regulador Neocognitrón tiene menos oscilaciones y llega al estado estacionario en un tiempo menor en comparación con el ST1.

Ahora bien, para el mismo caso 1, cuando se aplica una falla con 9 ciclos de duración las gráficas presentan un comportamiento casi similar al que presentan las gráficas cuando se aplica una falla con 6 ciclos de duración.

La figura 4.10 del voltaje en terminales con el regulador Neocognitrón tiene una mejor regulación en comparación con el regulador ST1, el regulador Neocognitrón presenta un número menor de oscilaciones que el regulador ST1 además de que alcanza primero el estado estacionario en un tiempo menor en comparación con el regulador ST1.

La figura 4.11 muestra como el regulador ST1 presenta un índice de funcionamiento mayor y conforme pasa el tiempo este se va incrementando. Al contrario del regulador ST1, el regulador neocognitrón minimiza el índice de funcionamiento.

De igual forma que en la figura 4.6 el comportamiento del ángulo del rotor en la figura 4.12 para el regulador ST1 presenta un mayor número de oscilaciones y tarda más en alcanzar el estado estacionario con respecto al regulador Neocognitrón.

La figura 4.13 y figura 4.14 potencia activa y potencia reactiva, respectivamente, muestran como el regulador Neocognitrón presenta un número menor de oscilaciones y un tiempo menor en llegar al estado estacionario en comparación con el regulador ST1 en ambas potencias.

El comportamiento del voltaje de campo en la figura 4.15 muestra como este oscila en mayor amplitud con el regulador ST1 alcanzando los valores máximo y mínimo dentro de su rango de operación, presenta un mayor número de oscilaciones y un mayor tiempo para llegar al estado estacionario.

Ahora, cuando se aplica una falla con 12 ciclos de duración para el mismo caso 1 ha diferencia cuando se aplica una falla de 6 y 9 ciclos de duración, los resultados difieren de manera considerable. Aquí se tiene el siguiente análisis.



El comportamiento del voltaje en terminales de la figura 4.16 para el regulador ST1 se torna inestable en comparación con el regulador Neocognitrón, el regulador ST1 presenta oscilaciones permanentes y es incapaz de regresar al estado estacionario, mientras que el regulador Neocognitrón es capaz de regular para este tipo de falla con 12 ciclos de duración.

La figura 4.17 muestra como el índice de funcionamiento del regulador ST1 conforme pasa el tiempo se va incrementando, por el contrario del regulador Neocognitrón que es menor su índice de funcionamiento.

El comportamiento del ángulo del rotor en la figura 4.18 para el regulador Neocognitrón presenta un número pequeño de oscilaciones y alcanza el estado estacionario en un tiempo corto, en comparación al regulador ST1 cuyo comportamiento muestra como se incrementa el ángulo del rotor conforme pasa el tiempo de simulación.

La potencia activa y potencia reactiva, figura 4.19 y figura 4.20, respectivamente, muestran como el regulador Neocognitrón presenta llega al estado estacionario en comparación con el regulador ST1 que permanece oscilando sin alcanzar el estado estacionario para ambas potencias.

El comportamiento del voltaje de campo en la figura 4.21 muestra como este oscila en mayor amplitud con el regulador ST1, presentando un mayor número de oscilaciones y un mayor tiempo para llegar al estado estacionario.

El comportamiento de las variables gráficas ante una falla de 12 ciclos de duración, muestra como el regulador Neocognitrón logra mantener la estabilidad de la máquina síncrona en comparación con el regulador ST1, cuyo comportamiento es con oscilaciones permanentes sin llegar al estado estacionario.

4.4.3. Caso 2. $P= 0.9833$. Con una falla de 6, 9 y 12 ciclos de duración.

Las siguientes gráficas muestran el comportamiento del sistema máquina bus infinito con una condición inicial de la potencia activa igual a $P=0.9833$ p.u., aplicando una falla de 6, 9 y 12 ciclos de duración.

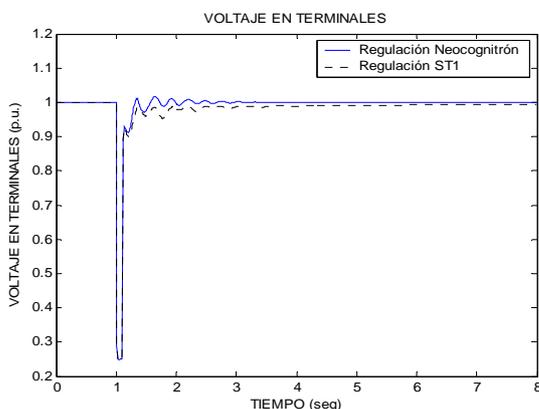


Figura 4.22. Comportamiento del voltaje en terminales ante una falla de 6 ciclos de duración.

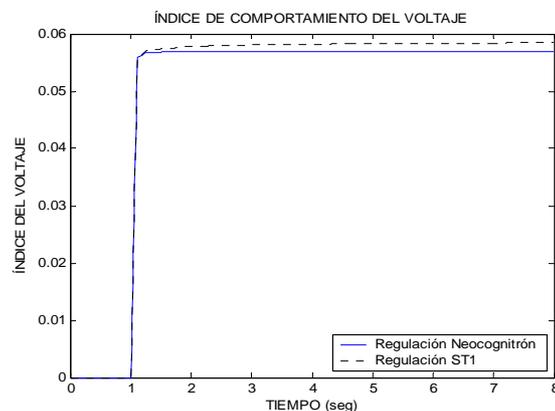


Figura 4.23. Índice de comportamiento del voltaje en terminales.

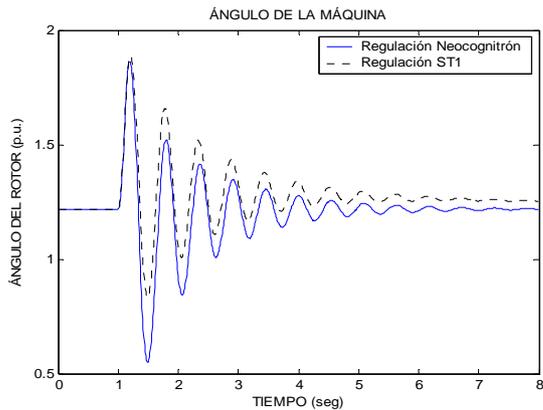


Figura 4.24. Comportamiento del ángulo de la máquina ante una falla de 6 ciclos de duración.

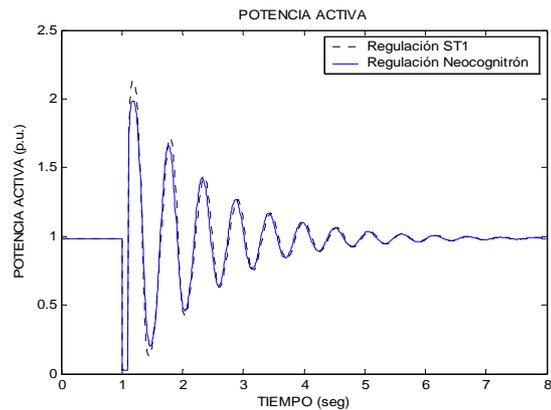


Figura 4.25. Comportamiento de la potencia activa ante una falla de 6 ciclos de duración.

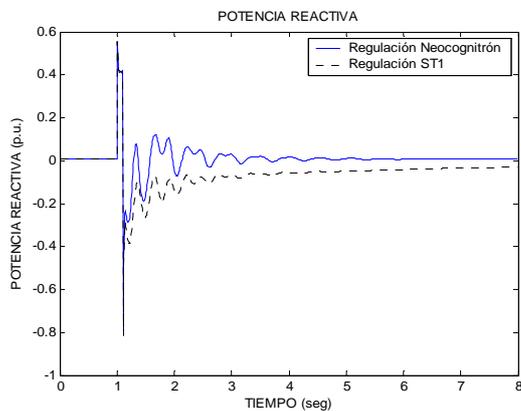


Figura 4.26. Comportamiento de la potencia reactiva ante una falla de 6 ciclos de duración.

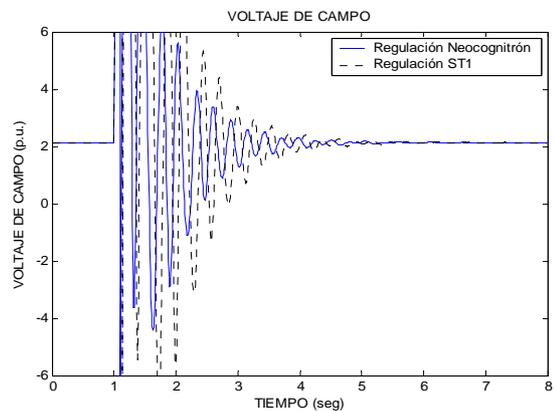


Figura 4.27. Comportamiento del voltaje de campo ante una falla de 6 ciclos de duración.

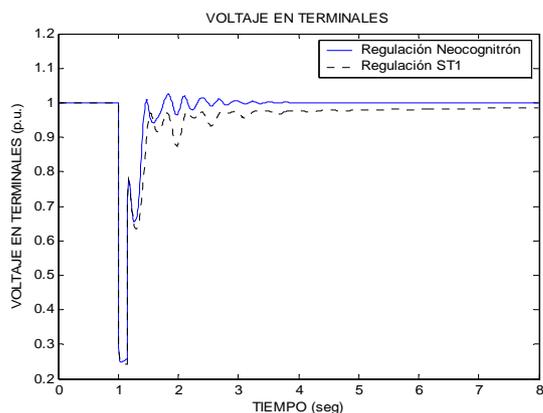


Figura 4.28. Comportamiento del voltaje en terminales ante una falla de 9 ciclos de duración.

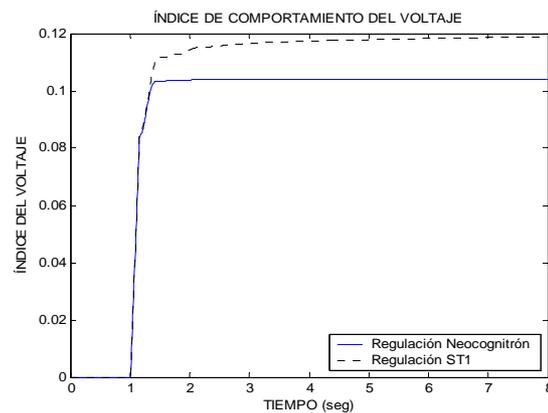


Figura 4.29. Índice de comportamiento del voltaje en terminales.

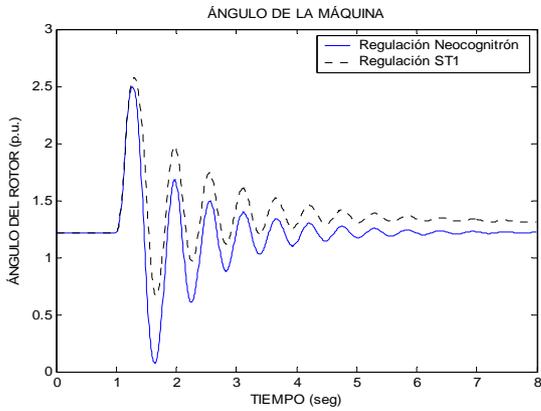


Figura 4.30. Comportamiento del ángulo de la máquina ante una falla de 9 ciclos de duración.

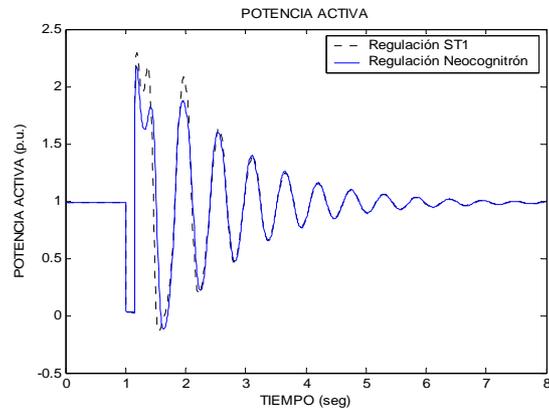


Figura 4.31. Comportamiento de la potencia activa ante una falla de 9 ciclos de duración.

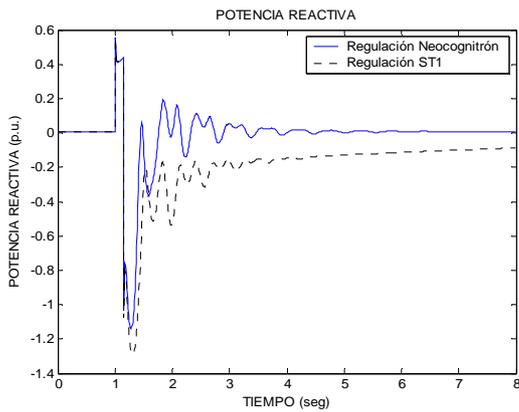


Figura 4.32. Comportamiento de la potencia reactiva ante una falla de 9 ciclos de duración.

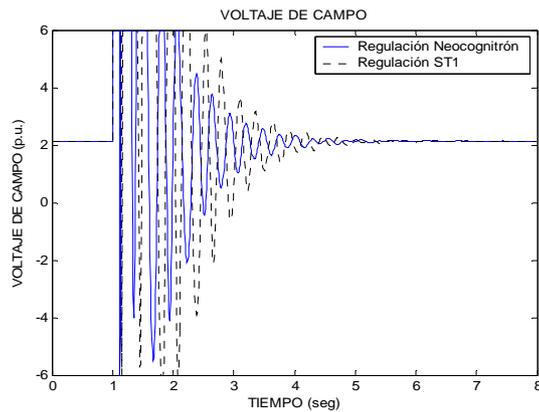


Figura 4.33. Comportamiento del voltaje de campo ante una falla de 9 ciclos de duración.

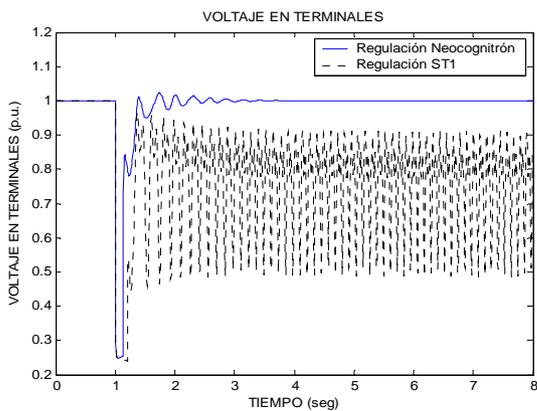


Figura 4.34. Comportamiento del voltaje en terminales ante una falla de 12 ciclos de duración.

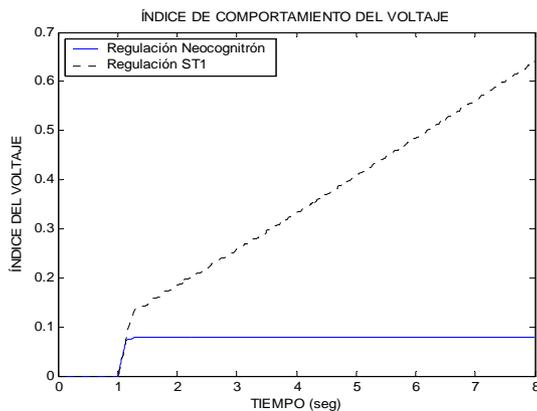


Figura 4.35. Índice de comportamiento del voltaje en terminales.

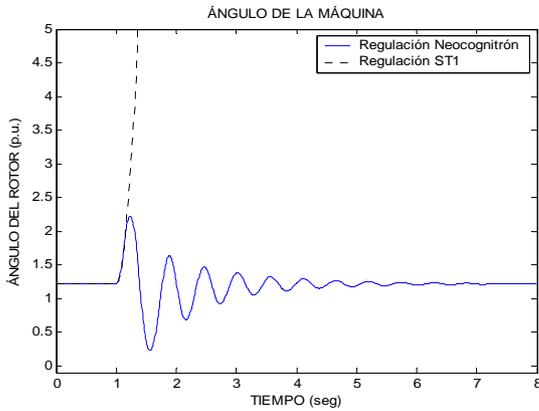


Figura 4.36. Comportamiento del ángulo de la máquina ante una falla de 12 ciclos de duración.

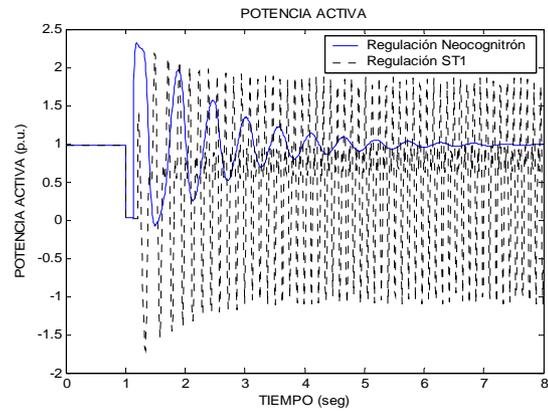


Figura 4.37. Comportamiento de la potencia activa ante una falla de 12 ciclos de duración.

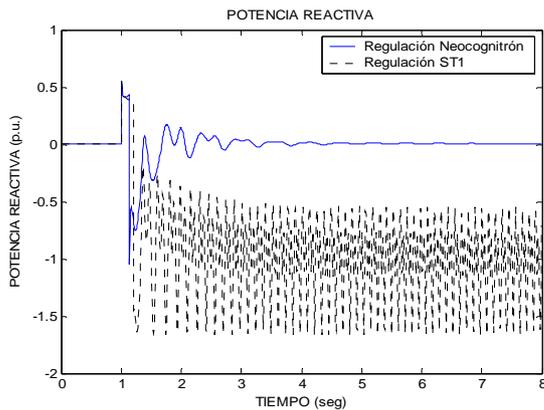


Figura 4.38. Comportamiento de la potencia reactiva ante una falla de 12 ciclos de duración.

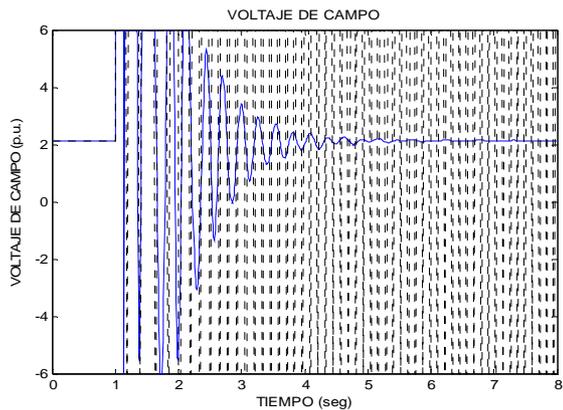


Figura 4.39. Comportamiento del voltaje de campo ante una falla de 12 ciclos de duración.

4.4.4. Análisis de resultados del caso 2.

El comportamiento del voltaje en terminales como se observa en las figuras 4.22, 4.28 y 4.34, para una falla de 6, 9 y 12 ciclos de duración, respectivamente, es de forma similar en las tres gráficas. Se observa como el voltaje en terminales con el regulador Neocognitrón tiene una mejor regulación en comparación con el regulador ST1. El voltaje en terminales para el regulador Neocognitrón presenta un número menor de oscilaciones que el regulador ST1 en las tres gráficas, además de que alcanza primero el estado estacionario. Cuando se presenta una falla con 12 ciclos de duración el sistema se vuelve inestable para el regulador ST1, es decir, no regresa al estado estacionario, mientras que el regulador Neocognitrón si es capaz de regresar al estado estacionario.

El índice de desempeño en las figuras 4.23, 4.29 y 4.35 muestra como el sistema se comporta de una mejor manera cuando se utiliza el regulador Neocognitrón al ver como este índice es más pequeño que cuando se utiliza el regulador ST1, para la falla de 6 y 9 ciclos de duración. Así, para la falla con 12 ciclos de duración el índice tiende a incrementarse cuando se utiliza el regulador ST1, caso contrario cuando se utiliza el regulador Neocognitrón.



El comportamiento del ángulo del rotor en la figura 4.24 y 4.30 para el regulador ST1 presenta un mayor número de oscilaciones y tarda más en alcanzar el estado estacionario con respecto al regulador Neocognitrón. El comportamiento del ángulo del rotor en la figura 4.36 para el regulador Neocognitrón presenta un número pequeño de oscilaciones y alcanza el estado estacionario en un tiempo corto, en comparación al regulador ST1 cuyo comportamiento muestra como el ángulo del rotor tiene una tendencia a incrementarse conforme pasa el tiempo de simulación, es decir, no llega al estado estacionario.

La figura 4.25 y figura 4.31 muestra como el comportamiento de la potencia activa para el regulador neocognitrón y para el regulador ST1 es casi similar, pero se puede observar que cuando se utiliza el regulador Neocognitrón las oscilaciones son de menor amplitud que las que presenta el regulador ST1. En cambio en la figura 4.37 se observa como el regulador ST1 se vuelve inestable, es decir la potencia activa no regresa al estado de prefalla.

El comportamiento de la potencia reactiva en la figura 4.26 y 4.32 para el regulador ST1 presenta un mayor número de oscilaciones y tarda más en alcanzar el estado estacionario con respecto al regulador Neocognitrón. La figura 4.38 muestra como el comportamiento de la potencia reactiva utilizando el regulador Neocognitrón es más robusto, el regulador ST1 se vuelve inestable.

El comportamiento del voltaje de campo, figura 4.27 y 4.33 muestra como el regulador Neocognitrón tiene un número menor de oscilaciones y con menor amplitud, además que llegar al estado estacionario en un tiempo menor en comparación con el regulador ST1. En la figura 4.33 muestra como el sistema de regulación ST1 dej de regular a partir de fallas con un tiempo de duración de 12 ciclos, comportandose de igual forma como se observo en el caso 1.

4.4.5. Caso 3. $P=0.5284$. Con una falla de 6, 9 y 12 ciclos de duración.

Las siguientes simulaciones son realizadas con una potencia activa igual a $P=0.5284$ p.u., con una falla de 6, 9 y 12 ciclos de duración, obteniéndose los resultados siguientes:

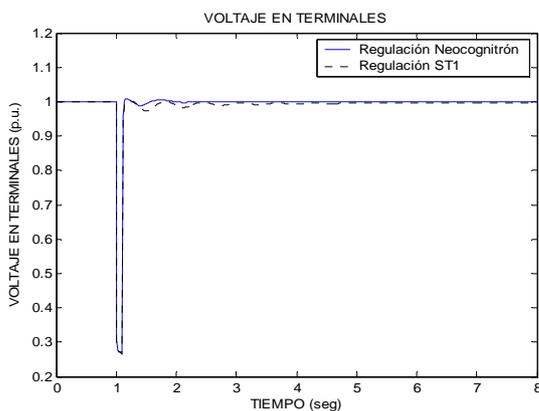


Figura 4.40. Comportamiento del voltaje de campo ante una falla de 6 ciclos de duración.

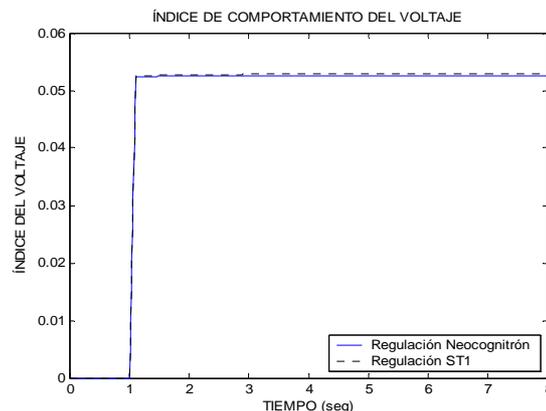


Figura 4.41. Índice de comportamiento del voltaje en terminales.

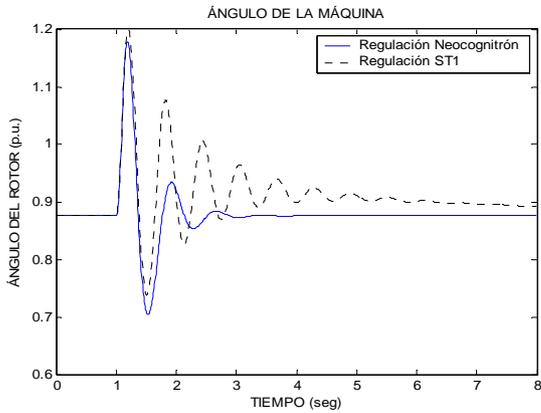


Figura 4.42. Comportamiento del ángulo de la máquina ante una falla de 6 ciclos de duración.

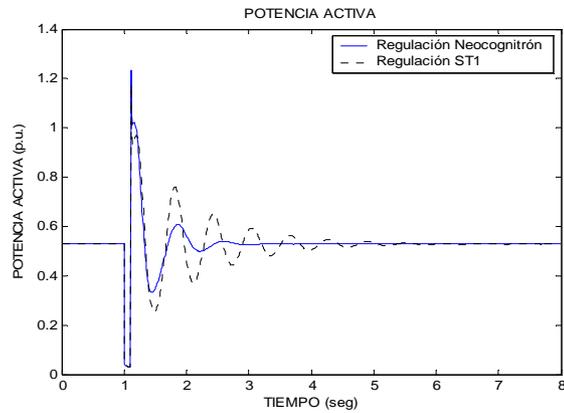


Figura 4.43. Comportamiento de la potencia activa ante una falla de 6 ciclos de duración.

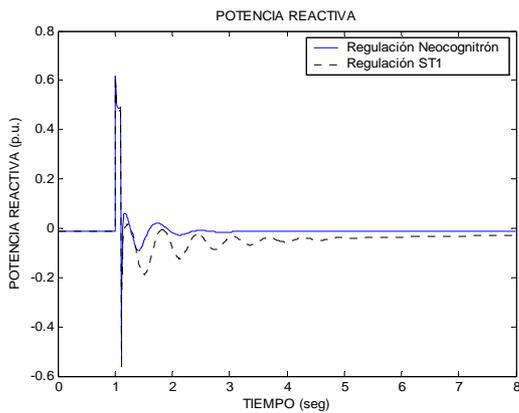


Figura 4.44. Comportamiento de la potencia reactiva ante una falla de 6 ciclos de duración.

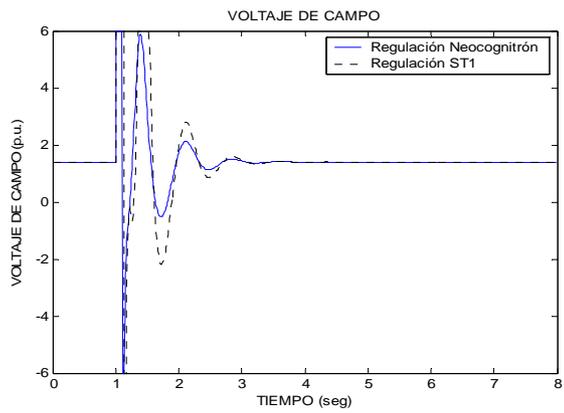


Figura 4.45. Comportamiento del voltaje de campo ante una falla de 6 ciclos de duración.

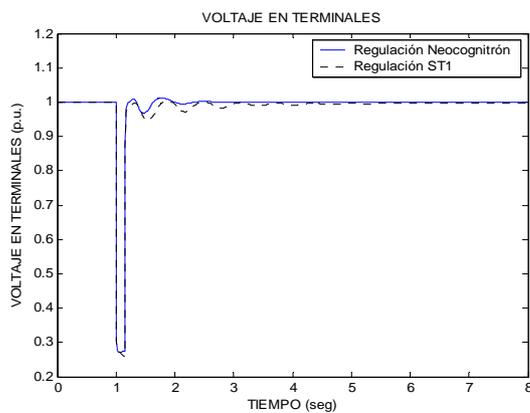


Figura 4.46. Comportamiento del voltaje en terminales ante una falla de 9 ciclos de duración.

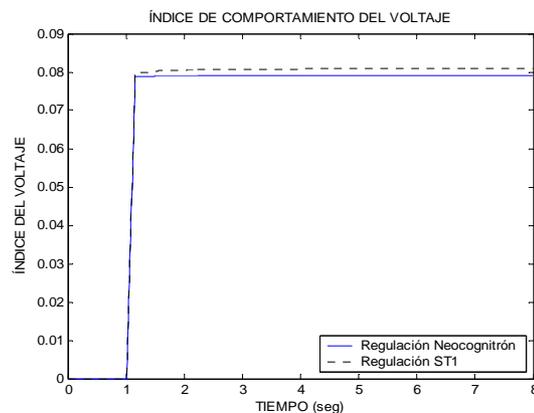


Figura 4.47. Índice de comportamiento del voltaje en terminales.

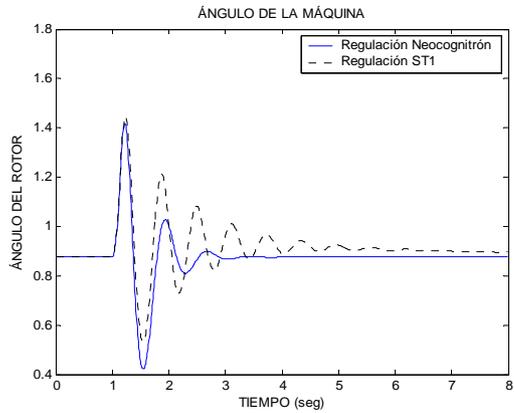


Figura 4.48. Comportamiento del ángulo de la máquina ante una falla de 9 ciclos de duración.

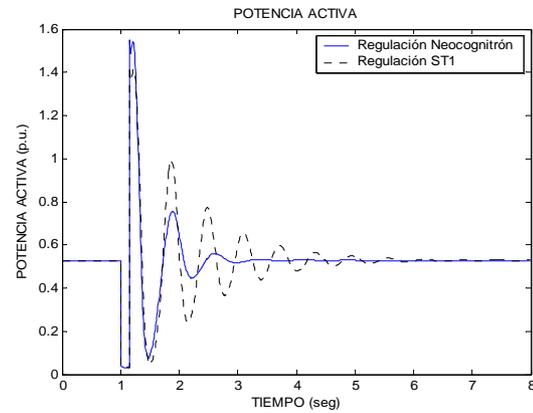


Figura 4.49. Comportamiento de la potencia activa ante una falla de 9 ciclos de duración.

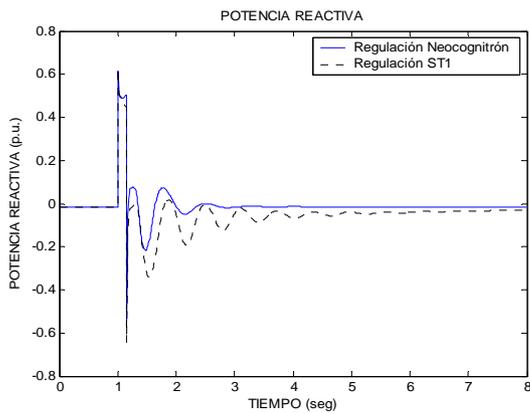


Figura 4.50. Comportamiento de la potencia reactiva ante una falla de 9 ciclos de duración.

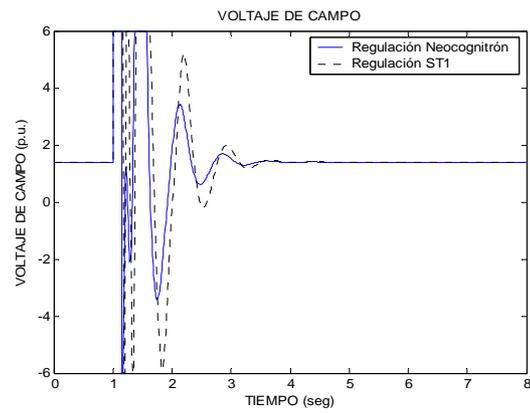


Figura 4.51. Comportamiento del voltaje de campo ante una falla de 9 ciclos de duración.

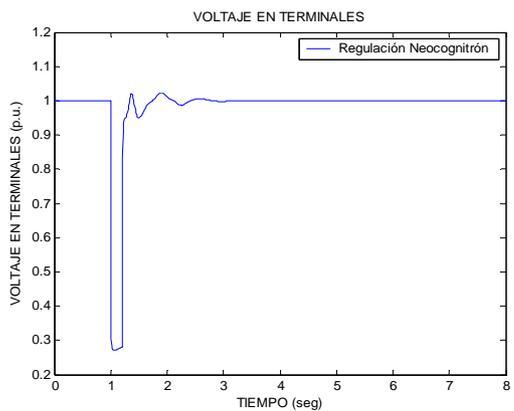


Figura 4.52. Comportamiento del voltaje en terminales ante una falla de 12 ciclos de duración.

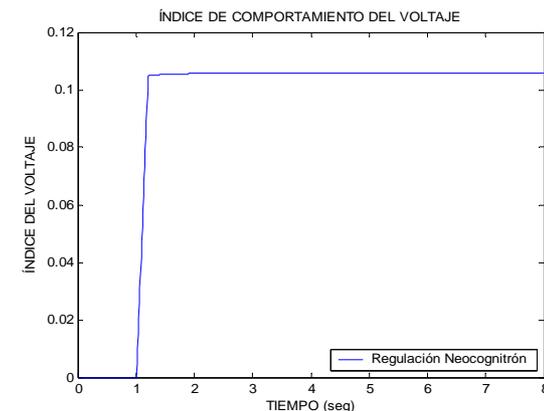


Figura 4.53. Índice de comportamiento del voltaje en terminales.

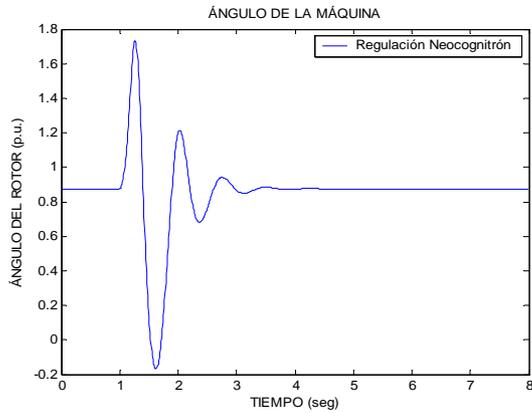


Figura 4.54. Comportamiento del ángulo de la máquina ante una falla de 12 ciclos de duración.

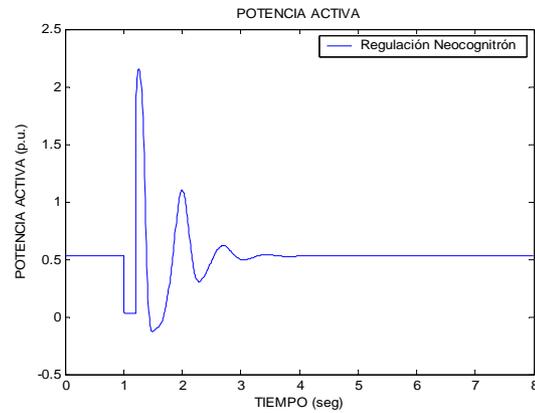


Figura 4.55. Comportamiento de la potencia activa ante una falla de 12 ciclos de duración.

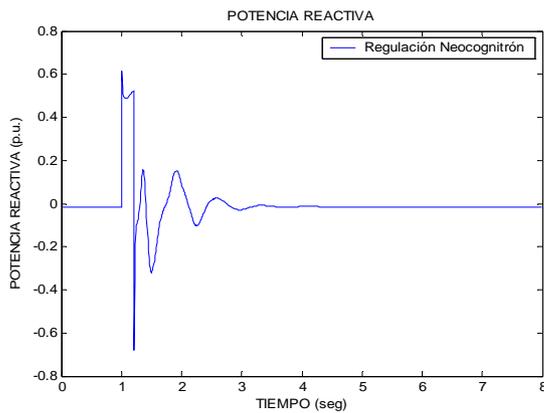


Figura 4.56. Comportamiento de la potencia reactiva ante una falla de 12 ciclos de duración.

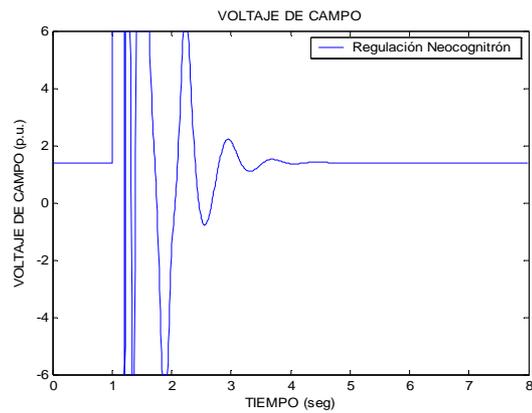


Figura 4.57. Comportamiento del voltaje de campo ante una falla de 12 ciclos de duración.

4.4.6. Análisis de resultados del caso 3.

En la figura 4.40 se puede observar como el voltaje en terminales con el regulador Neocognitrón tiene una mejor regulación en comparación con el regulador ST1. El voltaje en terminales para el regulador Neocognitrón presenta un número menor de oscilaciones que el regulador ST1 además de que alcanza primero el estado estacionario en un tiempo menor.

Se puede observar en la figura 4.41 que el regulador Neocognitrón presenta un índice de funcionamiento ligeramente menor al que presenta el regulador ST1, así se permite decir que el regulador neocognitrón es mejor porque minimiza este índice.

El comportamiento del ángulo del rotor en la figura 4.42 para el regulador ST1 presenta un mayor número de oscilaciones y tarda mas en alcanzar el estado estacionario con respecto al regulador Neocognitrón.

La figura 4.43 muestra como la potencia activa para el regulador neocognitrón alcanza su estado estacionario en un menor tiempo que el regulador ST1, de igual forma tiene un número menor de oscilaciones con menor amplitud que las que presenta el regulador ST1.



La figura 4.44 de la potencia reactiva presenta una mayor amplitud en sus oscilaciones cuando se utiliza el regulador ST1 con respecto a las alcanzadas por el regulador Neocognitrón.

El comportamiento del voltaje de campo se observa en la figura 4.45. En la figura se observa que el regulador Neocognitrón tiene menos oscilaciones y llega al estado estacionario en un tiempo menor en comparación con el ST1.

Para el caso 3, cuando se aplica una falla con 9 ciclos de duración, las gráficas presentan un comportamiento casi similar al que presentan las gráficas cuando se aplica una falla con 6 ciclos de duración. Ver figuras 4.46, 4.47, 4.48, 4.49, 4.50 y 4.51. En estas figuras se observa como el sistema con regulación Neocognitrón presenta un número menor de oscilaciones y regresa al estado estacionario en un intervalo de tiempo menor que el regulador ST1.

Como se puede observar en las figuras 4.52, 4.53, 4.54, 4.55, 4.56 y 4.57 se excluye la graficación de las variables obtenidas con el regulador ST1, debido a que para este caso a partir de fallas con 12 ciclos de duración es sistema deja de regular, al contrario del regulador Neocognitrón que como se observa el sistema presenta un número pequeño en oscilaciones y regresa al estado estacionario.

4.4.7. Caso 4. $P=1.1000$. Con una falla de 6, 9 y 12 ciclos de duración.

Las siguientes simulaciones son realizadas con una potencia activa igual a $P=1.1000$ p.u., con una falla de 6, 9 y 12 ciclos de duración, obteniéndose los resultados siguientes:

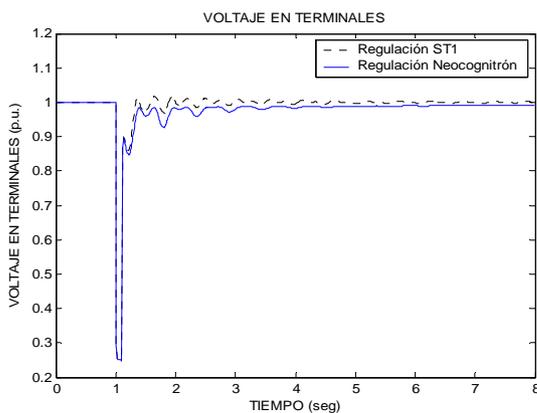


Figura 4.58. Comportamiento del voltaje en terminales ante una falla de 6 ciclos de duración.

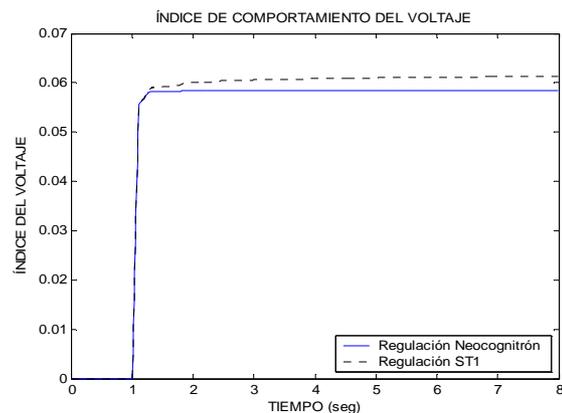


Figura 4.59. Índice de comportamiento del voltaje en terminales.

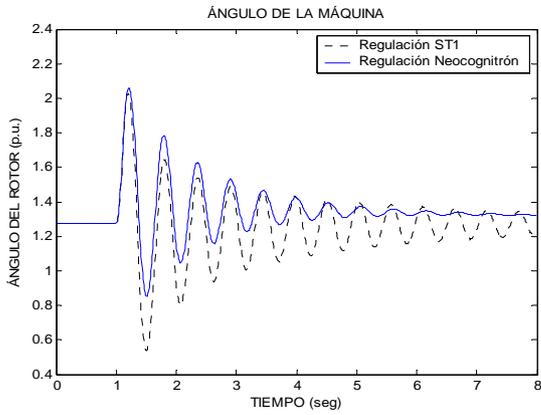


Figura 4.60. Comportamiento del ángulo de la máquina ante una falla de 6 ciclos de duración.

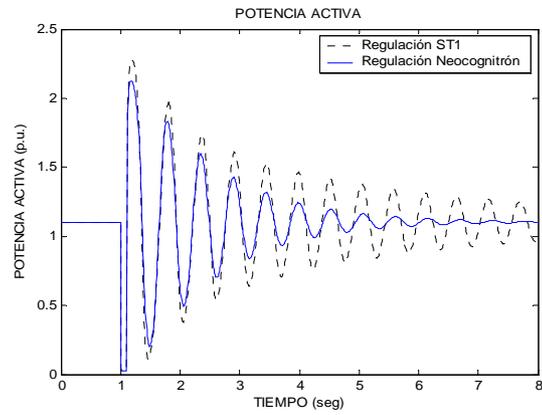


Figura 4.61. Comportamiento de la potencia activa ante una falla de 6 ciclos de duración.

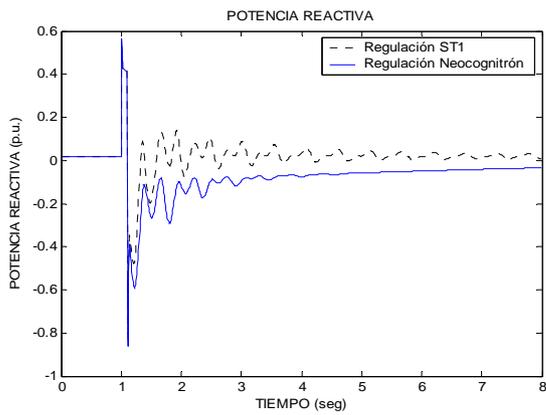


Figura 4.62. Comportamiento de la potencia reactiva ante una falla de 6 ciclos de duración.

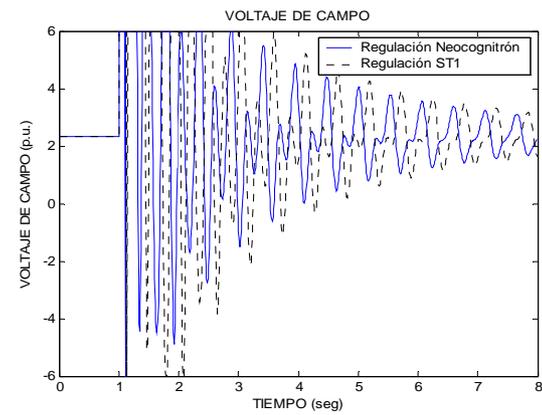


Figura 4.63. Comportamiento del voltaje de campo ante una falla de 6 ciclos de duración.

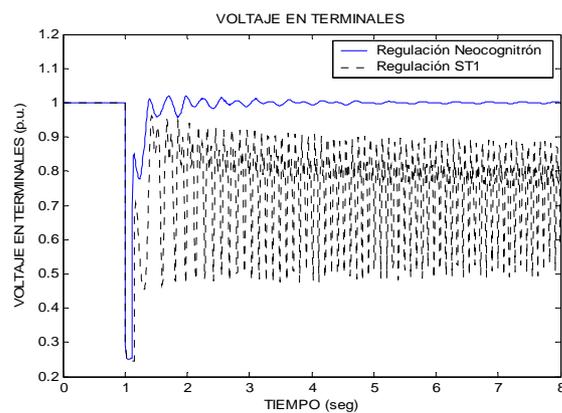


Figura 4.64. Comportamiento del voltaje en terminales ante una falla de 9 ciclos de duración.

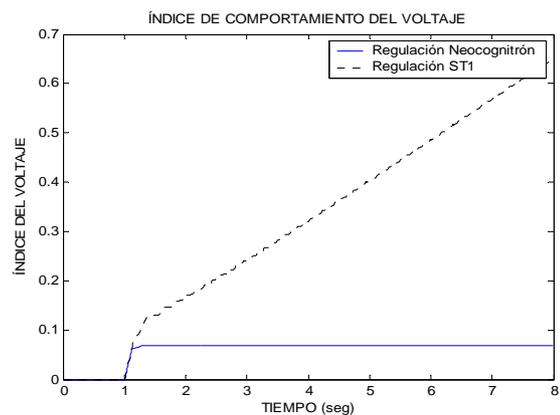


Figura 4.65. Índice de comportamiento del voltaje en terminales.

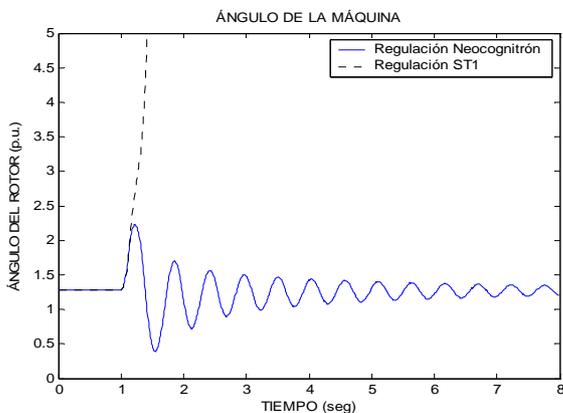


Figura 4.66. Comportamiento del ángulo de la máquina ante una falla de 9 ciclos de duración.

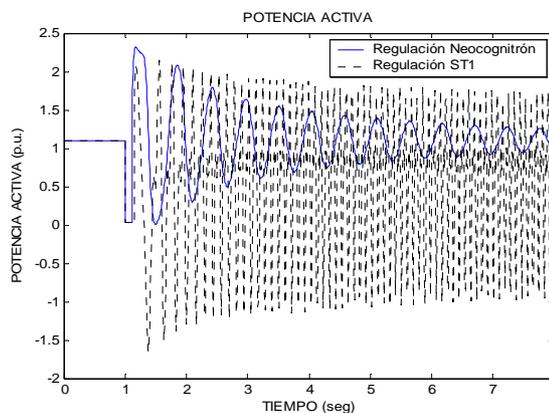


Figura 4.67. Comportamiento de la potencia activa ante una falla de 9 ciclos de duración.

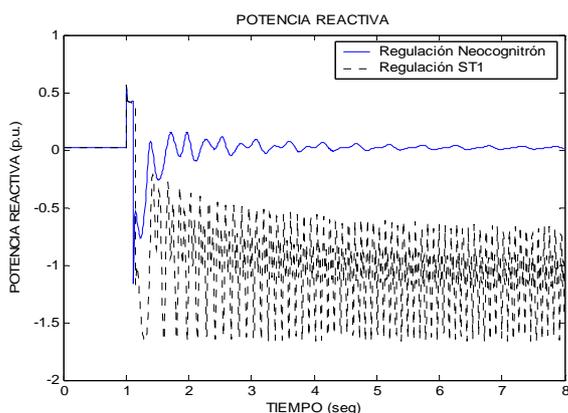


Figura 4.68. Comportamiento de la potencia reactiva ante una falla de 9 ciclos de duración.

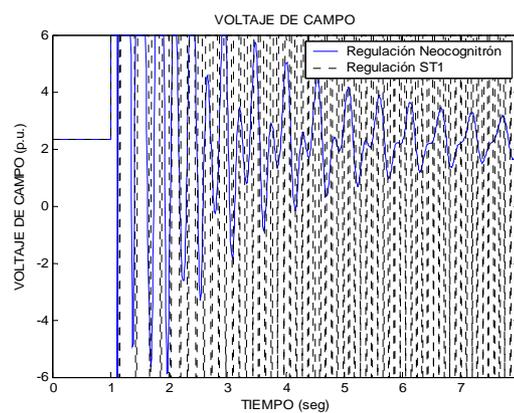


Figura 4.69. Comportamiento del voltaje de campo ante una falla de 9 ciclos de duración.

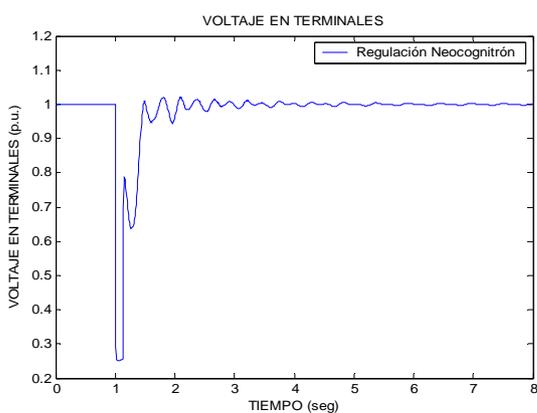


Figura 4.70. Comportamiento del voltaje en terminales ante una falla de 12 ciclos de duración.

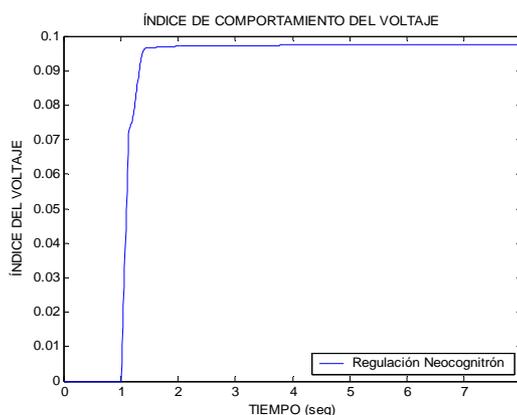


Figura 4.71. Índice de comportamiento del voltaje en terminales.

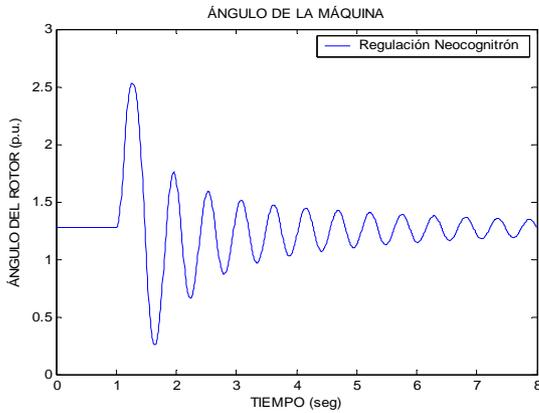


Figura 4.72. Comportamiento del ángulo de la máquina ante una falla de 12 ciclos de duración.

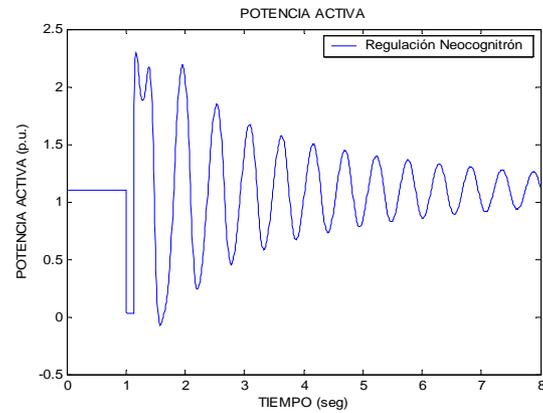


Figura 4.73. Comportamiento de la potencia activa ante una falla de 12 ciclos de duración.

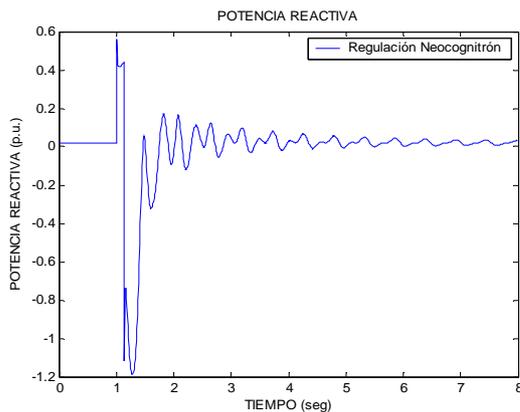


Figura 4.74. Comportamiento de la potencia reactiva ante una falla de 12 ciclos de duración.

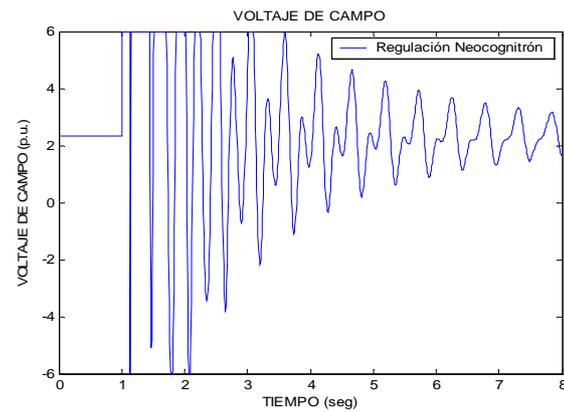


Figura 4.75. Comportamiento del voltaje de campo ante una falla de 12 ciclos de duración.

4.4.8. Análisis de resultados del caso 4.

En la figura 4.58 se puede observar como el voltaje en terminales con el regulador Neocognitrón tiene una mejor comportamiento en comparación con el regulador ST1. El voltaje en terminales para el regulador Neocognitrón alcanza primero el estado estacionario en un intervalo de tiempo menor que el regulador ST1.

Se puede observar en la figura 4.59 que el regulador Neocognitrón presenta un índice de funcionamiento menor al que presenta el regulador ST1, esto permite decir que el regulador neocognitrón es mejor porque minimiza este índice.

El comportamiento del ángulo del rotor en la figura 4.60 para el regulador ST1 presenta un mayor número de oscilaciones y como se observa en la figura el regulador Neocognitrón si alcanza el estado estacionario a diferencia del regulador ST1.

La figura 4.61 muestra como la potencia activa para el regulador Neocognitrón alcanza su estado estacionario al final del tiempo de la simulación, además de que tiene un número menor de oscilaciones con menor amplitud que las que presenta el regulador ST1.



La figura 4.62 de la potencia reactiva presenta una mayor amplitud en sus oscilaciones cuando se utiliza el regulador ST1 con respecto a las alcanzadas por el regulador Neocognitrón.

El comportamiento del voltaje de campo se observa en la figura 4.63. En la figura se observa que el regulador Neocognitrón tiene menos oscilaciones con menor amplitud en comparación a las que presenta el regulador ST1.

Para cuando se aplica una falla con 9 ciclos de duración, las gráficas presentan un comportamiento casi similar al que presentan las gráficas cuando se aplica una falla con 6 ciclos de duración. Ver figuras 4.64, 4.65, 4.66, 4.67, 4.68 y 4.69. En estas figuras se observa como el sistema con regulación Neocognitrón presenta un número menor de oscilaciones y regresa al estado estacionario en comparación a las que presenta el regulador ST1 que se comporta de manera inestable.

Al igual como se menciona anteriormente para una falla de 9 ciclos de duración, para una falla de 12 ciclos de duración el sistema con regulación ST1 se vuelve inestable. Como se puede observar en las figuras 4.70, 4.71, 4.72, 4.73, 4.74 y 4.75 se excluye la graficación de las variables obtenidas con el regulador ST1, debido a que para este caso a partir de fallas con doce ciclos de duración es sistema deja de regular, al contrario del regulador Neocognitrón.

4.4.9. Caso 5. $P=0.9592$. Con una falla de 6, 9 y 12 ciclos de duración.

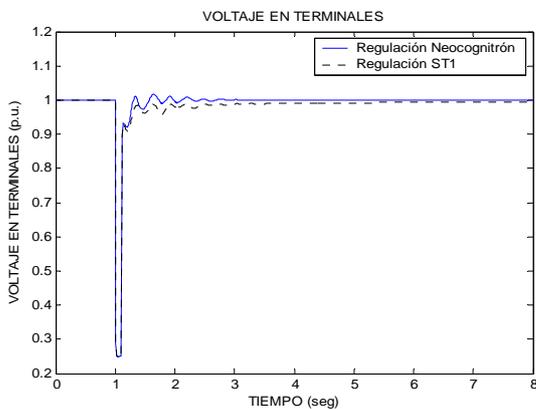


Figura 4.76. Comportamiento del voltaje en terminales ante una falla de 6 ciclos de duración.

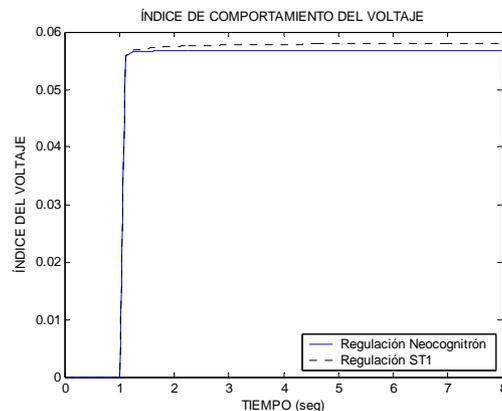


Figura 4.77. Índice de comportamiento del voltaje en terminales.

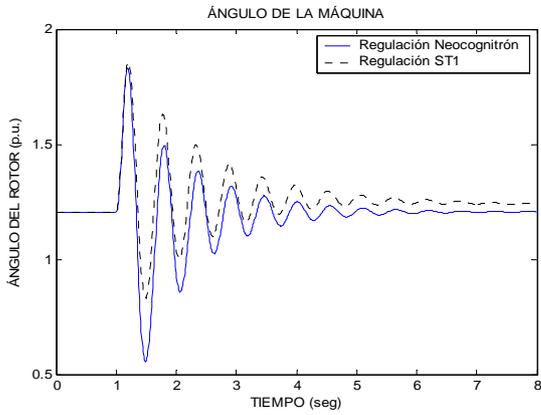


Figura 4.78. Comportamiento del ángulo de la máquina ante una falla de 6 ciclos de duración.

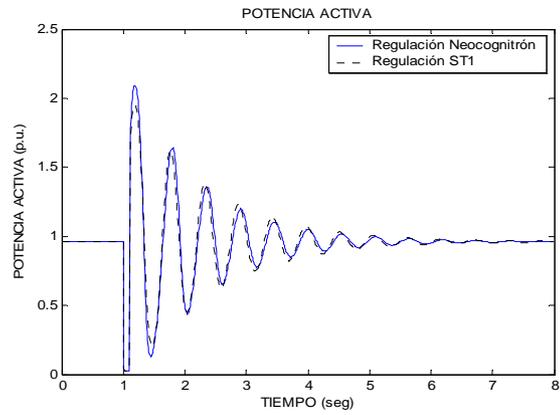


Figura 4.79. Comportamiento de la potencia activa ante una falla de 6 ciclos de duración.

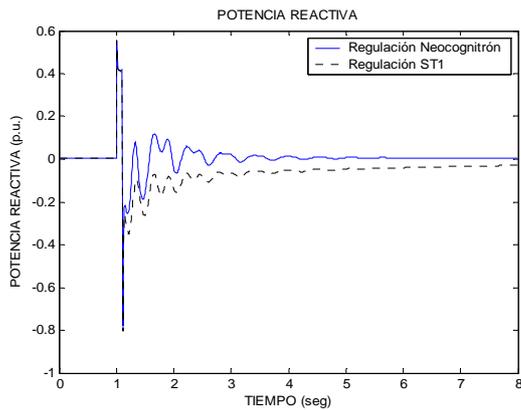


Figura 4.80. Comportamiento de la potencia reactiva ante una falla de 6 ciclos de duración.

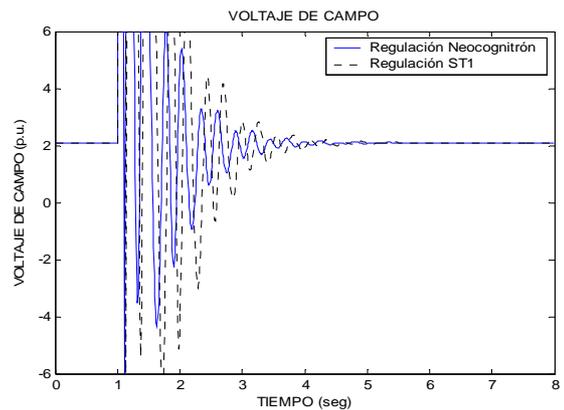


Figura 4.81. Comportamiento del voltaje de campo ante una falla de 6 ciclos de duración.

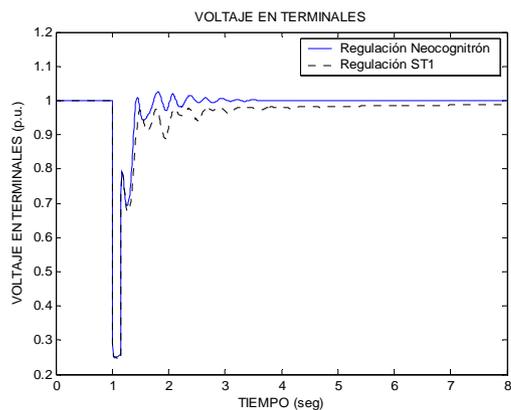


Figura 4.82. Comportamiento del voltaje en terminales ante una falla de 9 ciclos de duración.

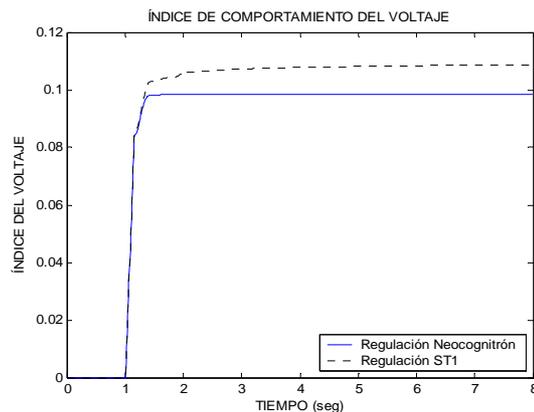


Figura 4.83. Índice de comportamiento del voltaje en terminales.

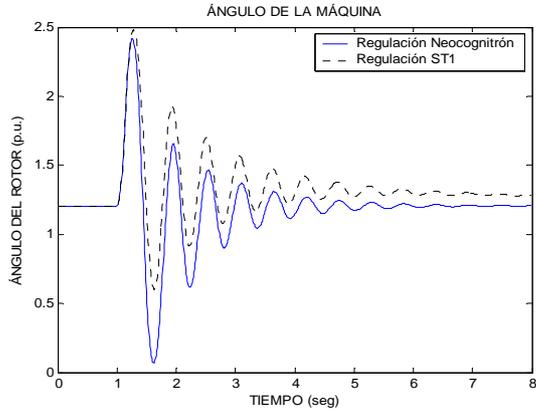


Figura 4.84. Comportamiento del ángulo de la máquina ante una falla de 9 ciclos de duración.

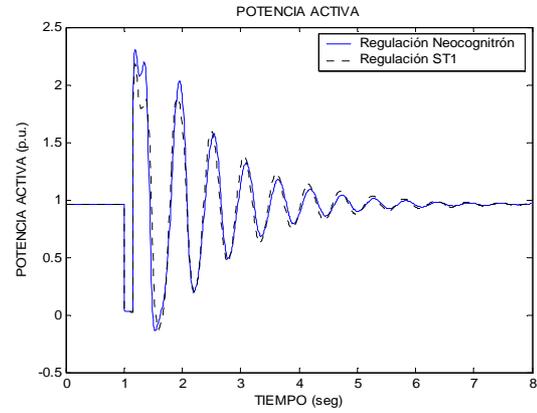


Figura 4.85. Comportamiento de la potencia activa ante una falla de 9 ciclos de duración.

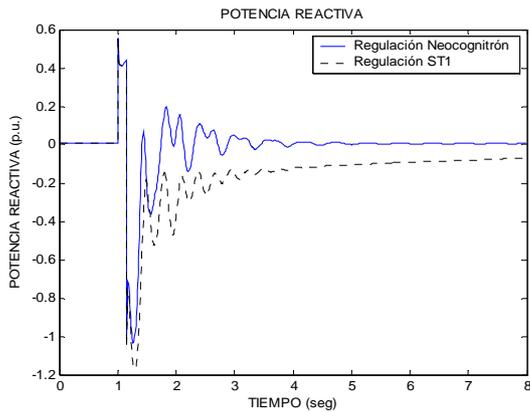


Figura 4.86. Comportamiento de la potencia reactiva ante una falla de 9 ciclos de duración.

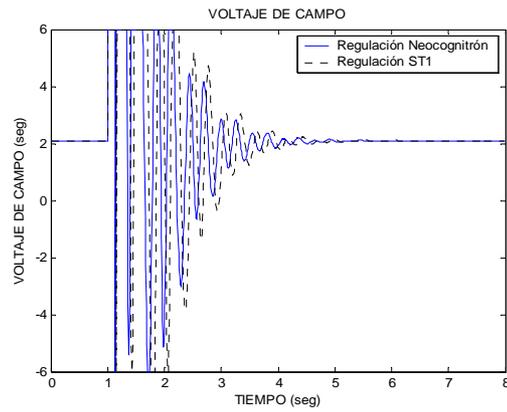


Figura 4.87. Comportamiento del voltaje de campo ante una falla de 9 ciclos de duración.

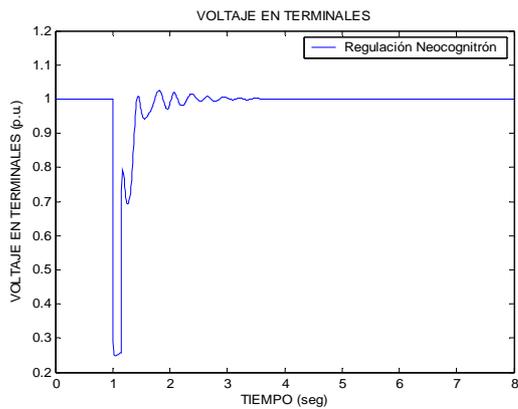


Figura 4.88. Comportamiento del voltaje en terminales ante una falla de 12 ciclos de duración.

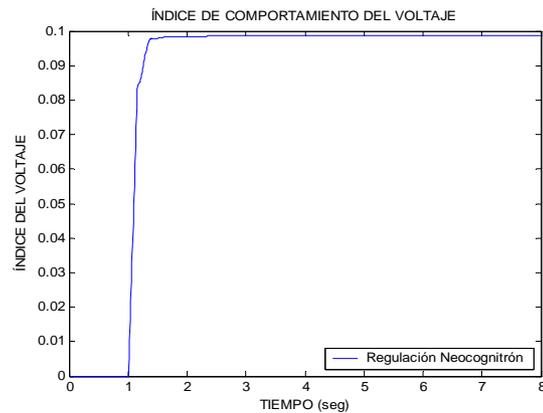


Figura 4.89. Índice de comportamiento del voltaje en terminales.

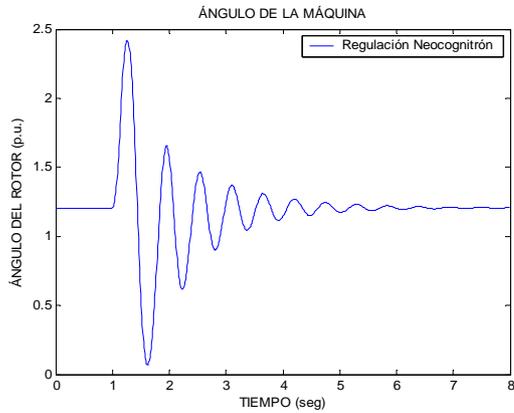


Figura 4.90. Comportamiento del ángulo de la máquina ante una falla de 12 ciclos de duración.

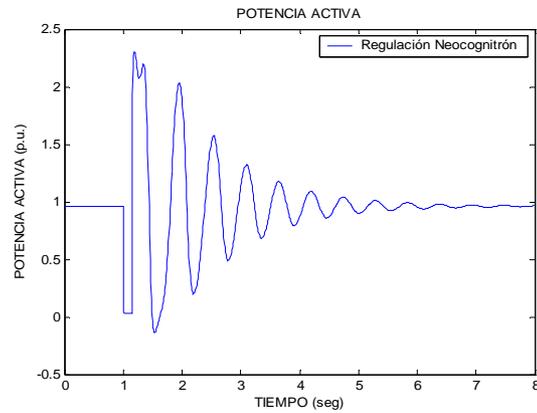


Figura 4.91. Comportamiento de la potencia activa ante una falla de 12 ciclos de duración.

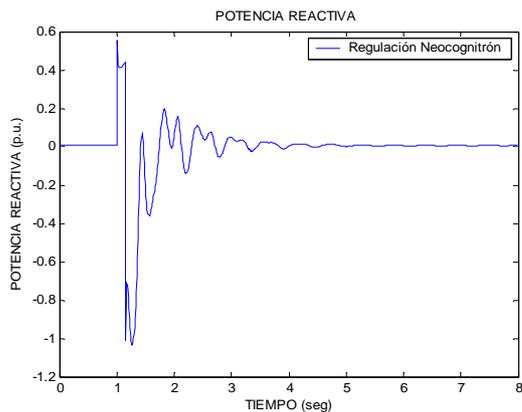


Figura 4.92. Comportamiento de la potencia reactiva ante una falla de 12 ciclos de duración.

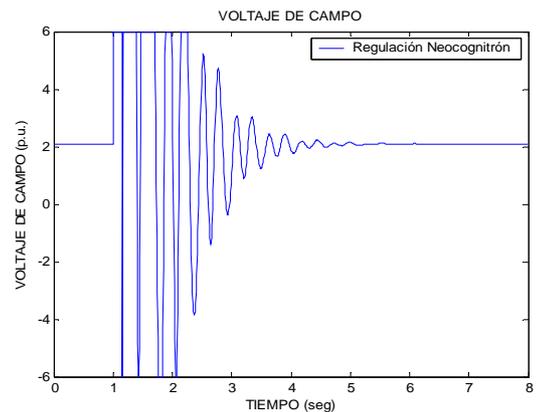


Figura 4.93. Comportamiento del voltaje de campo ante una falla de 12 ciclos de duración.

4.4.10. Análisis de resultados del caso 5.

El comportamiento del voltaje en terminales como se observa en las figuras 4.76 y 4.82 para una falla de 6, y 9 ciclos de duración, respectivamente, es de forma similar. Se observa como el voltaje en terminales con el regulador Neocognitrón tiene una mejor regulación en comparación con el regulador ST1. El voltaje en terminales para el regulador Neocognitrón presenta un número menor de oscilaciones en comparación con el regulador ST1, además de que alcanza primero el estado estacionario.

El índice de desempeño en las figuras 4.77 y 4.83 muestra como el sistema se comporta de una mejor manera cuando se utiliza el regulador Neocognitrón al ver como este índice es más pequeño que cuando se utiliza el regulador ST1, para la falla de 6 y 9 ciclos de duración.

El comportamiento del ángulo del rotor en la figura 4.78 y 4.84 para el regulador ST1 presenta un mayor número de oscilaciones y tarda mas en alcanzar el estado estacionario con respecto al regulador Neocognitrón.



La figura 4.79 y figura 4.85 muestran como el comportamiento de la potencia activa para el regulador neocognitrón y para el regulador ST1 es casi similar, pero se puede observar que cuando se utiliza el regulador Neocognitrón las oscilaciones son ligeramente de menor amplitud que las que presenta el regulador ST1.

El comportamiento de la potencia reactiva en la figura 4.80 y 4.86 para el regulador ST1 presenta un mayor número de oscilaciones y tarda mas en alcanzar el estado estacionario con respecto al regulador Neocognitrón.

El comportamiento del voltaje de campo, figura 4.81 y 4.87, muestra como el regulador Neocognitrón tiene un número menor de oscilaciones y con menor amplitud, además que llegar al estado estacionario en un tiempo ligeramente menor en comparación con el regulador ST1.

Debido a que se observo que el regulador ST1, deja de regular a partir de fallas con un tiempo de duración de 12 ciclos, como lo muestran las gráficas de los casos 1 y 2. Para los casos 3, 4 y 5 no se grafican los resultados obtenidos con el regulador ST1 y solo se grafican los resultados del regulador Neocognitrón ya que este presenta un buen desempeño para las condiciones de falla presentadas. Como se muestra en las figuras 4.88, 4.89, 4.90, 4.91, 4.92 y 4.93.

Así mismo, el índice de comportamiento del voltaje en terminales en todos los casos presentados, muestra un mejor desempeño para el regulador propuesto, lo que permite decir que este tipo de regulador es mejor que el regulador ST1, esto se justifica con lo que se describe en el apéndice C.



Capítulo 5

Conclusiones

5.1. Conclusiones.

La motivación de este trabajo es aprovechar la tecnología moderna del control inteligente numérico para obtener un mejor potencial en la estabilidad del sistema de potencia bajo estudios de cortos circuitos.

En este trabajo se ha descrito el análisis y la aplicación de una red neuronal neocognitrón la cual forma una estructura adaptable con características propias de las metodologías de la inteligencia artificial, esas características permiten a una red neuronal poder controlar la dinámica de una planta no lineal como lo es la máquina síncrona.

El regulador propuesto disfruta las ventajas generales de las redes neuronales como capacidad de generalización, tolerancia a fallas y la propiedad de adaptabilidad debido a su aprendizaje.

En los resultados de la simulación del capítulo anterior, se nota como las oscilaciones producidas por la falla aplicada en las variables graficadas, son menores en número y amplitud cuando se utiliza el sistema de regulación basado en el neocognitrón.

El regulador propuesto arroja las siguientes ventajas:

1. El método de diseño es robusto. Los resultados de la simulación muestran que el controlador obtenido trabaja muy bien con el modelo de la máquina. La robustez del controlador fue probado cambiando los parámetros del sistema y las condiciones de operación. Resultados satisfactorios fueron obtenidos.
2. Arquitectura paralela. La arquitectura paralela habilita más rápidamente la implementación de los algoritmos de control.
3. Capacidad de entrenamiento para varias condiciones de operación, por consiguiente este se puede adaptar para distintas situaciones de eventos.
4. La estructura utilizada en el regulador es adaptable y puede ser adaptado en línea o fuera de línea gracias al aprendizaje inverso utilizado.



No obstante el regulador propuesto presenta las siguientes desventajas:

1. Para nuestro regulador la fase de entrenamiento es decisivo para la obtención de resultados aceptables.
2. Un punto de precaución sobre el entrenamiento es que el criterio tiene un número de mínimo local. Los métodos de entrenamiento generalmente encuentran uno de ellos, que dependen de la inicialización de los pesos. Uno típicamente tratara con este problema entrenando la red un par de veces, cada vez se empieza con una inicialización diferente de los pesos.
3. Además de que los datos de entrenamiento deben escogerse por ejemplo de una gama amplia de entradas de los sistemas, y las entradas operacionales actuales pueden ser difíciles de definir.

5.2. Recomendaciones.

Una recomendación importante es variar el diseño de la estructura de la red neocognitrón propuesta, es decir variar el número de capas y número de células de manera aleatoria y probar si la estructura arroja resultados convenientes, ya que esto puede influir el desempeño y el comportamiento del control propuesto.

Se recomienda empezar el entrenamiento de la red neocognitrón con entradas parecidas a las de un control convencional hasta determinar cuales tienen mayor influencia en el proceso que se desea controlar.

Otra recomendación es estudiar la posibilidad de utilizar una estructura con aprendizaje no supervisado que sea capaz de adaptarse en línea y de esta manera permita la aplicación de este control en tiempo real.

5.3. Trabajos Futuros.

La implementación a base de dispositivos electrónicos y hardware especializado sería el siguiente paso en vista de que de manera computacional se ha demostrado un buen desempeño del sistema propuesto y además de que en los últimos años los intereses en la investigación de las metodologías que integran la llamada inteligencia artificial se han incrementado notablemente.

Probar el esquema de control propuesto en sistemas multimáquinas y no solo con el modelo de la máquina bus infinito.

Otro trabajo útil sería la aplicación del esquema de control propuesto en otro tipo de plantas a controlar para observar que tan factible puede ser bajo otro tipo de condiciones.



Referencias

- [1] Romero, D., "Notas del Curso de Control de Sistemas Eléctricos de Potencia", IPN. ESIME-Zacatenco, México 1996.
- [2] Guo, G., Wang, K., & Gao, L., "Robust Nonlinear Controller for Power System Transient Stability Enhancement with Voltage Regulation", IEEE Procedures. Generation, Transmission and Distribution. Vol. 143, No. 5, September 1996.
- [3] Nielsen, R., "Neurocomputing", HNC, Inc. and University of California, San Diego, Addison-Wesley Publishing Company, 1989.
- [4] Cortes, R., "Control de Excitación Difuso de un Generador Síncrono", SEPI-ESIME. Tesis para obtener el Grado de Doctor en Ciencias, México 1997.
- [5] Johnson, R., & Chappell B., Cognizers "Neural Networks and Machines that think", Wiley Science Editions, John Wiley and Sons, Inc, 1988.
- [6] Norgaard, M., Ravn, O., Poulsen, N., & Hansen, L., "Neural Networks for Modelling and Control of Dynamic Systems". Springer, 2000.
- [7] King, R., "Computational Intelligence in Control Engineering". University of Patras, Greece, Marcel Dekker, Inc. New York, 1999.
- [8] Haykin, S., "Neural Networks, A Comprehensive Foundation". McMaster University Hamilton, Ontario, Canada. Prentice Hall, 1999.
- [9] IEEE Power Society, "A Tutorial Course on Artificial Neural Networks with Applications to Power Systems", Edited by Mohamed El-Sharkawi and Dagmar Niebur, 1990.
- [10] Díaz, J., "Desarrollo de un Regulador de Voltaje para un Generador Síncrono Utilizando la Teoría de Resonancia Adaptable". Tesis de Maestría. IPN-SEPI ESIME, 2001.
- [11] Cochofel, H., Wooten, D., & Principe, J., "A Neural Network Development Environment for Adaptive for Adaptive Inverse Control". IEEE, University of Florida. 1998.
- [12] King, R., "Computational Intelligence in Control Engineering". University of Patras, Greece, Marcel Dekker, Inc. New York, 1999.
- [13] Gupta, M., & Rao, D., "Neuro-Control Systems, Theory and Applications", IEEE Press, USA, New York, 1994.
- [14] Kanjilal, P., "Adaptive Prediction and Predictive Control". IEE Control Engineering series 52, 1995.
- [15] Ogata, K., "Modern Control Engineering". Fourth edition. Prentice Hall. NJ 2003.



-
- [16] Lewis, P., & Yang, C., "Basic Control Systems Engineering". Prentice Hall, 1997.
- [17] Gama, M., "Control Directo del Par y Flujo del Motor de Inducción Utilizando una Red Neuro-Difusa". Tesis de Maestría. IPN-SEPI ESIME, 2004.
- [18] Kailath, T., "Linear Systems". Prentice Hall. New York. 1980.
- [19] Zadeh, L., "Linear System Theory". McGraw Hill. New York. 1963.
- [20] Tanomaru, T., "Process Control by On-Line Trained Neural Controllers". Student Member, IEEE, and Sigeru Omatu, Member, IEEE Transactions on industrial electronics, Vol. 39, 1992.
- [21] Ren, Y., & Gang, G., "Neural Networks for the Inverse Control of Discrete-Time Nonlinear Plant". Department of Computer Science and Technology. Tsinghua University. 1994.
- [22] Norgaard, M., "Neural Networks for Modeling and Control of Dynamic Systems": Springer-Verlag. London 2000.
- [23] Jang, R., "Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence". Prentice Hall. 1997.
- [24] Lin, T., & Lee, G., "Neural Fuzzy System: A Neuro-Fuzzy Synergism to Intelligent Systems". Prentice Hall. 1996.
- [25] Jang, R., "Neuro-Fuzzy Modeling and Control". Proceedings of the IEEE, Vol. 83, No. 3, pp 378-405, March 1985.
- [26] Harnold, C., & Lee, K., "A Free Model Based Model Reference Adaptive Inverse Controller" Senior Member, IEEE. Department of Electrical Engineering. The Pennsylvania State University, USA. 2000.
- [27] Freeman, A., & Skapura, D., "Neural Networks Algorithms, Applications and Programming Techniques". Addison-Wesley Publishing Company, 1992.
- [28] Fukushima, K., & Miyake, S., "Neocognitron: A New Algorithm Pattern Recognition Tolerant of Deformations and Shifts in Position", Pattern Recognition, 15[6], pp. 445-469 (1982).
- [29] Fukushima, K., "Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition", Neural Networks, 1[2], pp. 119-130 (1988).
- [30] Pandya, A., & Macy, R., "Pattern Recognition with Neural Networks in C++". CRC PRESS, IEEE PRESS, 1996.
- [31] Fukushima, K., Miyake, S., & Ito, T., "Neocognitron: A Neural Network Model for a Mechanism of Visual Pattern Recognition," IEEE Trans. Systems, Man Cyber., SMC-13 (5), 826-834, 1983.



-
- [32] Fukushima, K., "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position," *Biol. Cyber.*, 36 (4), 193-202, 1980.
- [33] Fukushima, K., And N. Wake: "Handwritten Alphanumeric Character Recognition by the Neocognitron." *IEEE Trans. Neural Networks* 2(3): 355-365, 1991.
- [34] Fukushima, K., "Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position", *Trans. IECE Japan*, Vol. 62-A, no 10, pp. 658-665 (1979).
- [35] Harris, C., "Advances in Intelligent Control". Taylor and Francis, Great Britain, 1994.
- [36] Werbos, P., "Neurocontrol and Related Techniques". In *Handbook of Neural Comp. Appl.* A. Maren. Ed. Orlando. Fl. Academic. 1990.
- [37] Shamsollahi, P., & Fellow, O., "On-line Identification of Synchronous Generator Using Neural Networks", IEEE. Department of Electrical and Computer Engineering. The University of Calgary. Canada. 1996.
- [38] Hernandez, J., "Control Neuronal del Sistema de Excitación de un Generador Síncrono". Tesis de Maestría. IPN-SEPI ESIME, Agosto de 2001.
- [39] Mohame E., & Hiwary, E., "Electric Power Applications of Fuzzy Systems". IEEE Press Series on Power Engineering. P. M. Anderson, Series Editor. 1998.
- [40] Daozhi, G., "Self-Tuning Controller for Generator Excitation Control". *IEEE Transaction on Power Apparatus and Systems*. Vol. PAS-102. No.6 June 1983 pp. 1877-1884.
- [41] IEEE Committee Report. "Excitation Systems Models for Power System Stability", *IEEE Trans. Power Apparatus and Systems*, Vol. PAS-100. No. 2. February 1981. pp 494-509.
- [42] Werbos, P., "An Overview of Neural Networks for Control". *IEEE Control Systems* 1991.
- [43] Dorf, R., "Sistemas Modernos de Control". Addison-Wesley Iberoamericana. 1989.



Otras referencias consultadas son:

[44] Fukushima, K., "Cognitrón: A Self-Organized Multi-Layered Neural Network Model," NHK Technical Monograph, 30, NHK Technical Research Laboratories, Tokyo, 1981.

[45] Fukushima, K., "Cognitrón: A Self-Organizing Multi-Layered Neural Network," Biol. Cyber., 20 (3/4), 121-136, 1975.

[46] IEEE Committee Report. Proposed "Excitation System Definition for Synchronous Machines". IEEE Transactions on Power Apparatus and Systems. Vol. PAS-88. No. 8. August 1969, pp 1248-1258.

[47] Koessler, R., "Techniques for Tuning Excitation System Parameters". IEEE Transaction on Power Apparatus and Systems. Vol. PAS.248. No. 1. January 1988.

[48] Demello, F., & Concordia, C., "Concepts of Synchronous Machine Stability as Affected by Excitation Control". IEEE Trans. On Power Apparatus and Systems, Vol. PAS-88, No. 4, April 1969, pp 316-329.

[49] Soberanes, M., "Control Adaptable de Predicción Generalizada para un Sistema de Excitación". Tesis. Sección de estudios de posgrado e investigación. ESIME. Marzo 1997.

[50] Gomáriz, S., Biel, D., Matas, J., & Reyes, M., "Teoría de Control: Diseño Electrónico", Alfaomega, México 1999.

[51] Garcia, C., & Morari, M., "Internal Model Control -1. A Unifying Review and Some New Results", Ind. Eng. Chem. Process De.Dev., vol. 21, pp 308-323, 1989.

[52] Wen, J., & Malik, O., "A Synchronous Generator Fuzzy Excitation Controller Optimally Designed with a Genetic Algorithm". IEEE Transactions on Power Systems, Vol. 13, No. 3 August 1998.

[53] Lee, J., "Inversion Control of Nonlinear Systems with Neural Network Modelling". IEEE Proc.- Control Theory Appl. Vol. 144. No. 5. September 1997.

[54] Lesueur, S., Massicotte D., & Sicard P., "Indirect Inverse Control Based on Neural Networks using Dynamic Back Propagation for Nonlinear Dynamic Systems". IEEE, 2001.

[55] Norgaard, M., "Tools for System Identification and Control with Neural Networks". Computing & Control Engineering Journal. February 2001.

[56] Widrow, B., & Klett, G., "Nonlinear Adaptive Inverse Control". IEEE, 1997.

[57] Plett, G., "Adaptive Inverse Control of Linear and Nonlinear Systems Using Dynamic Neural Networks". IEEE Transactions on Neural Networks, Vol. 14. No. 2, March 2003.



APENDICES



Apéndice A

Modelo Matemático de la Máquina Síncrona

A.1. Modelo Matemático.

A.1.1. Modelo de la Máquina Síncrona [40].

Las ecuaciones diferenciales del modelo de 5º orden [40], usadas en esta tesis son:

$$p\delta_r = \omega_0 s \quad \text{A.1}$$

$$Mps = -K_d s + Tm - Te \quad \text{A.2}$$

$$T'_{d0} p e'_q = V_f - (X_d - X'_d) i_d - e'_q \quad \text{A.3}$$

$$T''_{d0} p e''_q = e'_q - (X'_d - X''_d) i_d - e''_q \quad \text{A.4}$$

$$T''_{q0} p e''_d = (X_q - X''_q) i_q - e''_d \quad \text{A.5}$$

$$e''_d = V_d + r_a i_d - X''_q i_q \quad \text{A.6}$$

$$e''_q = V_q + r_a i_q + X''_d i_d \quad \text{A.7}$$

$$Te = e''_d i_d + e''_q i_q - (X''_d - X''_q) i_d i_q \quad \text{A.8}$$

A.1.2. Modelo de la Red [40].

Sin pérdida de generalidad, la línea de transmisión se describe por una impedancia equivalente de Thevenin. Por lo tanto, el voltaje terminal y sus componentes en eje directo y eje en cuadratura, son las siguientes:

$$V_d = V_\infty \text{Sen} \delta_r + r_e i_d - X_e i_q \quad \text{A.9}$$

$$V_q = V_\infty \text{Cos} \delta_r + r_e i_q + X_e i_d \quad \text{A.10}$$

$$V_T^2 = V_d^2 + V_q^2 \quad \text{A.11}$$



A.1.3. Modelo de la Turbina de Vapor y del Gobernador de Velocidad [40].

El gobernador de velocidad que se utiliza es del tipo mecánico – hidráulico, el modelo esta idealizado, debido a que no se esta considerando el limitador de posición de la válvula, así como tampoco la función de transferencia que describe al relevador de velocidad, quedando únicamente representado por el bloque (b) de la figura A.1.

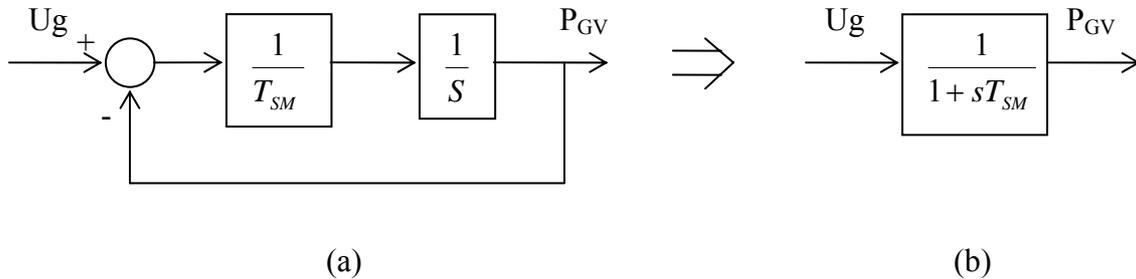


Figura A.1. Diagrama de bloques del gobernador de velocidad mecánico - hidráulico

En el caso de la turbina, se utilizó una turbina térmica de simple recalentamiento modificada. Se usa únicamente el modelo gobernador-turbina con los elementos básicos necesarios, para realizar las funciones de incremento y decremento de potencia activa, mediante la manipulación de la entrada al gobernador U_g sin hacer control sobre el lazo primario. Por lo tanto no se necesitan modelos más detallados en esta parte del sistema. Así, el conjunto de ecuaciones diferenciales que describen este sistema son las siguientes:

$$T_{SM} pP_{GV} = U_g - P_{GV} \quad A.12$$

$$T_{CH} pP_{HP} = P_{GV} - P_{HP} \quad A.13$$

$$T_{RH} pTm = F_{HP} pP_{HP} + P_{HP} - Tm \quad A.14$$

A.1.4. Modelo del Sistema de Excitación [40] y Regulador de Voltaje [41].

El sistema de excitación empleado es del tipo rotatorio, este sistema queda representado por la ecuación diferencial siguiente:

$$T_x pV_f = u - V_f \quad A.15$$

El sistema de regulación de voltaje que se usa en este trabajo, es el regulador tipo ST1 (excitatriz con fuente de potencial a través de rectificadores controlados), propuesto por el comité sobre sistemas de excitación modernos como el considerado aquí.

Tomando en cuenta que las constantes de tiempo de los transformadores de potencial para acondicionar la señal de voltaje terminal a niveles apropiados, son del orden de 0.0001 segundos, en este trabajo se desprecia. En este tipo de sistema el voltaje límite es proporcional al voltaje generado, por esta razón no se considera el efecto de la saturación de la excitatriz. Para obtener las ecuaciones de estado del



sistema excitación – regulación, se parte del diagrama de bloques de la figura A.2, que representa este regulador.

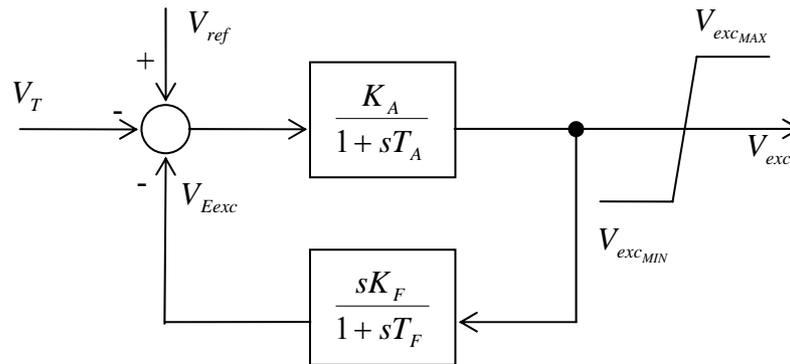


Figura A.2. Diagrama de bloques del regulador tipo ST1

Por lo tanto las ecuaciones son:

$$T_A pV_f = K_A (V_{ref} - V_T - V_{Eexc}) - V_f \quad \text{A.16}$$

$$T_f pV_{Eexc} = K_f pV_f - V_{Eexc} \quad \text{A.17}$$

Para todos los casos:

$$p = \frac{d}{dt}$$

A.1.5. Datos del Sistema Bajo Prueba.

Los datos del sistema máquina bus infinito se muestran en la tabla A.1. Los datos de la línea de transmisión, el gobernador de velocidad, la turbina térmica y de la máquina síncrona, la cual corresponde a un generador son tomados de la referencia [40]. Para el regulador de voltaje ST1 de la referencia [41].

Parámetros del Sistema Máquina Bus Infinito			
Variable	Valor	Variable	Valor
ω_0	377.0	X_e	0.15
M	5.5294	Re	0.01
Kd	3.0	TSM	0.1
$T'do$	5.66	TCH	0.15
$T''do$	0.041	TRH	5.0
$T''qo$	0.065	FHP	0.33
X_d	1.904	KA	400.0
$X'd$	0.312	TA	0.02
$X''d$	0.266	KF	0.008
X_q	1.881	TF	1.0
$X''q$	0.260	TX	0.025
ra	0.0		

Tabla A.1 Datos del sistema bajo prueba



A.2. Determinación de las Condiciones Iniciales y de las Ecuaciones Dinámicas para el Sistema Máquina Bus Infinito.

Para determinar las condiciones iniciales, se parte del sistema máquina bus infinito que se muestra en la (Fig. A.3) [40], del diagrama vectorial de la máquina síncrona en estado subtransitorio (Fig. A.4), y de las ecuaciones para el modelo subtransitorio (A.1 – A.11).

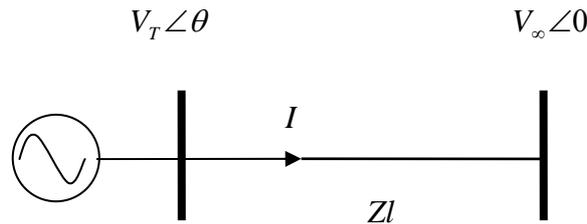


Figura A.3 Diagrama del sistema máquina bus infinito usado en esta tesis

A.2.1. Red.

Se considera que la máquina síncrona está conectada a un bus infinito mediante una impedancia externa $Zl = r_e + jX_e$ representada por una línea de transmisión, como se muestra en la figura A.3.

El Voltaje Terminal (V_T) y su ángulo (θ), son datos conocidos previamente. Para determinar las componentes real ($e_T r$) e imaginaria ($e_T i$), se trabaja el voltaje terminal en forma compleja, quedando en la siguiente forma:

$$V_T = e_T r + j e_T i \quad \text{A.18}$$

Donde:

$$e_T r = V_T \cos \theta$$

$$e_T i = V_T \text{sen} \theta$$

El Voltaje de Bus, también es un dato conocido con anterioridad. Este se trabaja en forma compleja y queda de la siguiente forma $V_b = V_b \angle 0$.

De la Potencia Eléctrica se sabe que, $S = V * I = P - jQ$, por lo tanto tenemos que:

$$S = V * \frac{V_T - V_b}{Zl} \quad \text{A.19}$$

$$P = \text{real}(S) \quad \text{A.20}$$

$$Q = -\text{imag}(S) \quad \text{A.21}$$

Corriente de Armadura. Partiendo del triángulo de potencias se tiene que:



Del diagrama vectorial se determina que:

$$e_q dx = V_T + r_a i_a \cos \phi + X_q i_a \sin \phi \quad \text{A.25}$$

$$e_q dy = -r_a i_a \sin \phi + X_q i_a \cos \phi \quad \text{A.26}$$

Donde:

$$\delta_i = \arctan \frac{e_q dy}{e_q dx} \quad \text{A.27}$$

$$\delta_r = \theta + \delta_i \quad \text{A.28}$$

Haciendo cero la derivada de la ecuación A.1, obtenemos para el Deslizamiento s que:

$$0 = \omega_0 s \quad \Leftrightarrow \quad s = 0 \quad \text{A.29}$$

Haciendo cero la derivada en la ecuación A.4, obtenemos para el Voltaje transitorio en eje en cuadratura e'_q que:

$$0 = e'_q - (X'_d - X''_d) i_d - e''_q \quad \Leftrightarrow \quad e'_q = (X'_d - X''_d) i_d + e''_q \quad \text{A.30}$$

Ahora, de las ecuaciones del sistema máquina bus infinito, para un modelo subtransitorio se tiene que el voltaje subtransitorio en eje en cuadratura e''_q es:

$$e''_q = V_T \cos \delta_i + r_a i_q + X''_d i_d \quad \text{A.31}$$

El Voltaje subtransitorio en eje directo e''_d :

$$e''_d = V_T \sin \delta_i + r_a i_d - X''_q i_q \quad \text{A.32}$$

El par mecánico T_m :

$$T_m = e''_d i_d + e''_q i_q \quad \text{A.33}$$

Par eléctrico T_e :

$$T_e = T_m \quad \text{A.34}$$

A.2.3. Sistema de excitación – Regulador de voltaje.

Para obtener las condiciones iniciales, de la excitatriz y del regulador de voltaje tipo ST1, se hacen cero las derivadas de las ecuaciones A.15, A.16 y A.17.



Cuando se utiliza el regulador adaptable, el voltaje de campo es:

$$0 = u - V_f \quad \Leftrightarrow \quad V_f = u \quad \text{A.35}$$

Cuando se utiliza el regulador ST1 las condiciones iniciales son:

$$0 = K_A (V_{ref} - V_T - V_{Eexc}) - V_f$$

$$V_f = K_A (V_{ref} - V_T - V_{Eexc}) \quad \text{A.36}$$

$$0 = -V_{Eexc}$$

$$V_{Eexc} = 0 \quad \text{A.37}$$

$$V_{ref} = \frac{V_f}{K_A} + V_T \quad \text{A.38}$$

Voltaje máximo y mínimo de campo en el limitador de voltaje:

$$V_{exc_{min}} = -6.0 \text{ p.u.}$$

$$V_{exc_{max}} = 6.0 \text{ p.u.} \quad \text{A.39}$$

A.2.4. Turbina de Vapor y Gobernador de Velocidad.

Haciendo cero las derivadas en las ecuaciones A.12, A.13 y A.14, se tiene que:

$$0 = U_g - P_{GV} \quad \Leftrightarrow \quad P_{GV} = U_g \quad \text{A.40}$$

$$0 = P_{GV} - P_{HP} \quad \Leftrightarrow \quad P_{HP} = P_{GV} \quad \text{A.41}$$

$$0 = P_{HP} - Tm \quad \Leftrightarrow \quad Tm = P_{HP} \quad \text{A.42}$$

A.2.5. Ecuaciones Dinámicas.

Cuando se simula el sistema, la primera iteración se realiza usando las condiciones iniciales, sin embargo, para las iteraciones posteriores el programa debe calcular el punto de operación para ese instante de tiempo. Para realizar esto, se deben proporcionar las ecuaciones que realicen esta función, en base a las ecuaciones A.6, A.7, A.9 y A.10, obteniendo las ecuaciones siguientes:

Sustituyendo A.9 en A.6:

$$e''_d = Vbsen\delta_r + r_e i_d - X_e i_q + r_a i_d - X''_q i_q$$

$$e''_d = Vbsen\delta_r + (r_e + r_a) i_d - (X_e + X''_q) i_q \quad \text{A.43}$$



Sustituyendo A.10 en A.7:

$$e_q'' = Vb \cos \delta_r + r_e i_q + X_e i_d + r_a i_q + X_d'' i_d$$

$$e_q'' = Vb \cos \delta_r + (r_e + r_a) i_q + (X_e + X_d'') i_d \quad \text{A.44}$$

Resolviendo el sistema de dos ecuaciones simultáneas A.43 y A.44 para obtener i_d e i_q , se obtiene:

$$i_d = \frac{(r_e + r_a)(e_d'' - Vb \sin \delta_r) + (X_e + X_q'')(e_q'' - Vb \cos \delta_r)}{(r_e + r_a)^2 + (X_e + X_d'')(X_e + X_q'')} \quad \text{A.45}$$

$$i_q = \frac{(r_e + r_a)(e_q'' - Vb \cos \delta_r) - (X_e + X_d'')(e_d'' - Vb \sin \delta_r)}{(r_e + r_a)^2 + (X_e + X_d'')(X_e + X_q'')} \quad \text{A.46}$$

$$I = i_q + j i_d \quad \text{A.47}$$

Los voltajes V_q y V_d se obtienen de A.6 y A.7:

$$V_d = e_d'' - r_a i_d - X_q'' i_q \quad \text{A.48}$$

$$V_q = e_q'' - r_a i_q - X_d'' i_d \quad \text{A.49}$$

$$V = V_q + j V_d \quad \text{A.50}$$

Las potencias son:

$$S = VI^* \quad \text{A.51}$$

$$P = \text{real}(S) \quad \text{A.52}$$

$$Q = -\text{imag}(S) \quad \text{A.53}$$

El par eléctrico es:

$$T_e = e_d'' i_d + e_q'' i_q \quad \text{A.54}$$

Así las ecuaciones dinámicas del sistema son de la ecuación A.45 a la ecuación A.54.



Apéndice B

Teoría de las Redes Neuronales

B.1. Introducción.

Las redes neuronales son una tecnología informática poco conocida, aunque antigua, perteneciente a la inteligencia artificial. Basadas en el funcionamiento de la neurona cerebral, sus orígenes se remontan a los primeros años de la informática. Tal vez su aplicación más conocida es la relacionada con el reconocimiento de patrones visuales (imágenes).

Las características de las redes neuronales son:

B.1.1. Capacidad de Aprendizaje [42].

Las redes neuronales tienen la capacidad de aprender a realizar tareas basadas en un entrenamiento o una experiencia inicial. No es necesario elaborar modelos con anterioridad o expresar funciones de procesamiento. La adaptación de las redes neuronales se debe a la capacidad de auto ajuste de los elementos de proceso y sus conexiones para adaptarse a las nuevas condiciones. Una red neuronal no necesita un algoritmo para resolver un problema, ya que ella puede generar su propia distribución de información interna para solucionarlo.

B.1.2. Autoorganización [42].

La capacidad de aprendizaje de una red neuronal, mencionada arriba, se basa en la capacidad de la red de organizar internamente la información recibida de forma que los pesos asociados a las conexiones entre elementos de procesamiento (neuronas) varían su valor. Esta variación de pesos se basa en la modificación automática de sus valores en base a un algoritmo de aprendizaje concreto para cada tipo de red neuronal que establece los mecanismos, condiciones y variaciones que se dan durante el aprendizaje.

B.1.3. Tolerancia a Fallos [42].

La destrucción parcial de una red neuronal implica una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden retener, pese a esta degradación. Esta capacidad de tolerancia a fallos es inherente al modelo de red neuronal, ya que la información que retiene la red se encuentra distribuida por todo su conexionado con un cierto grado de redundancia, al contrario de lo que ocurre con los sistemas tradicionales que almacenan la información sólo una vez, dando así pie a que un fallo de memoria pueda anular todo el sistema. Con respecto a los datos, las redes neuronales poseen también cierto nivel de tolerancia a errores, ya que son capaces de aprender a reconocer patrones con ruido, distorsionados e incompletos.



B.1.4. Operación en Tiempo Real [42].

Las redes neuronales, al ser sistemas masivos paralelos, pueden ser implementadas bajo hardware para proceso paralelo. Esto, unido a que la variación de los pesos de las conexiones en tiempo de explotación de la red es nula (en la mayoría de las redes), hacen a estos modelos especialmente idóneos para la realización de tareas en tiempo real.

B.1.5. Fácil Inserción en la Tecnología Existente [42].

Debido a que es posible implementar físicamente una red neuronal a un costo relativamente bajo debido a su naturaleza modular, es posible ir mejorando un sistema existente adaptando redes neuronales para tareas concretas y definidas e integrándolas poco a poco en el sistema.

No obstante, no todo podía ser bueno. Hay problemas que no pueden ser implementados bajo una red neuronal con facilidad o fiabilidad, debido a la naturaleza del problema o bien a que no existe un modelo de red bien definido para solucionarlo. La aplicación de las redes neuronales no termina de salir completamente de los laboratorios. El público en general y un amplio espectro de los profesionales de la informática las desconoce o conocen muy poco de ellas.

B.2. Teoría de las Redes Neuronales [42].

Las redes neuronales artificiales fueron inicialmente desarrolladas de acuerdo al principio elemental del funcionamiento del sistema nervioso humano. Desde entonces, una variedad muy grande de redes han sido construidas. Todas están compuestas de unidades (neuronas), y conexiones entre ellas, que juntas determinan la conducta de la red. La opción del tipo de red depende del problema a ser resuelto; la red de gradiente con retropropagación es la más frecuentemente usada. Esta red consiste de tres o más capas de neuronas: una capa de la entrada, una capa de salida y por lo menos una capa oculta (figura B.1). En la mayoría de los casos, una red con sólo una capa oculta, se usa para restringir el tiempo del cálculo, sobre todo cuando los resultados obtenidos son satisfactorios.

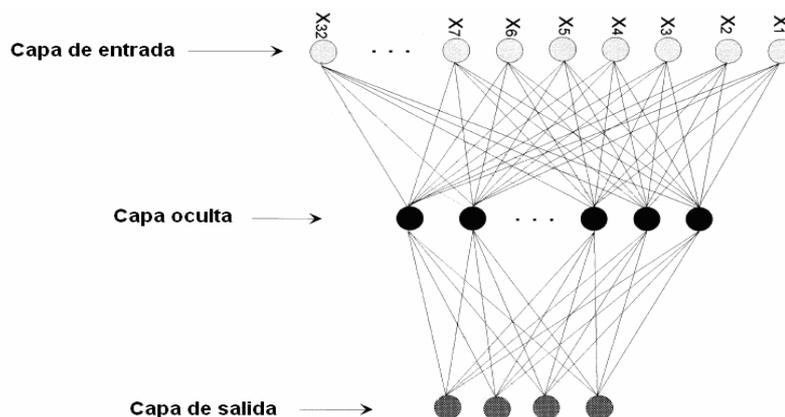


Figura B.1. Estructura de una red neuronal.



La capa de entrada comprende n neuronas que codifican los n piezas de la señal de entrada ($X_1 \dots X_n$) de la red (variables independientes). El número de neuronas de la capa oculta son empíricamente escogidas por el usuario. Finalmente, la capa de salida comprende k neuronas para las salidas k (variables dependientes). Cada conexión entre dos neuronas es asociada con un factor de peso (valor al azar entre -0.3 y $+0.3$ en principio); este peso se modifica por iteraciones sucesivas durante el entrenamiento de la red de acuerdo al dato de entrada y salida. En la capa de entrada, el estado de cada neurona es determinado por la variable de entrada; las otras neuronas (capa oculta y capa de salida) evalúan el estado de la señal de la capa anterior (Figura B.2) como:

$$a_j = \sum_{l=1}^l X_l W_{jl} \quad \text{B.1}$$

Donde a_j es la entrada net de la neurona j ; X_l es el valor de salida de la neurona l de la capa anterior; W_{jl} es el factor de peso de la conexión entre la neurona l y neurona j .

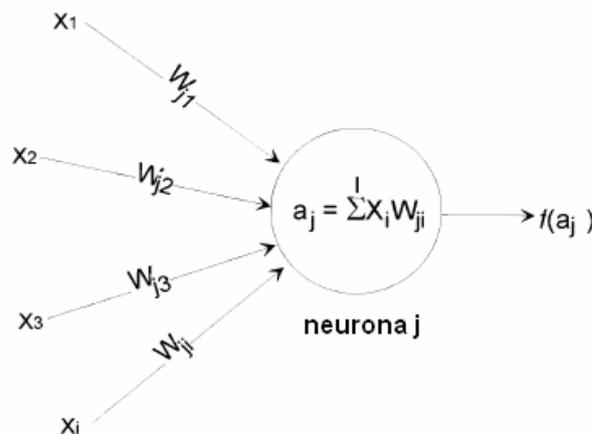


Figura B.2. Detalle de una neurona.

La actividad de las neuronas es normalmente determinada vía una función sigmoide:

$$f(a_j) = \frac{1}{1 + \exp^{-a_j}} \quad \text{B.2}$$

Así, los factores de peso representan la respuesta de la red neuronal artificial al problema a enfrentarse.

B.2.1. Redes Neuronales [42].

La figura B.3 describe el procedimiento para operar con redes neuronales. Originalmente la red neuronal no dispone de ningún tipo de conocimiento útil



almacenado. Para que la red neuronal ejecute una tarea es preciso entrenarla. Primero se selecciona un conjunto de datos, o patrones de aprendizaje. Después se desarrolla la arquitectura neuronal, número de neuronas, tipo de red. Por decirlo con otras palabras, se selecciona el modelo y el número de variables dependientes e independientes. Se procede a la fase de aprendizaje o estimación del modelo y a continuación se validan los resultados.

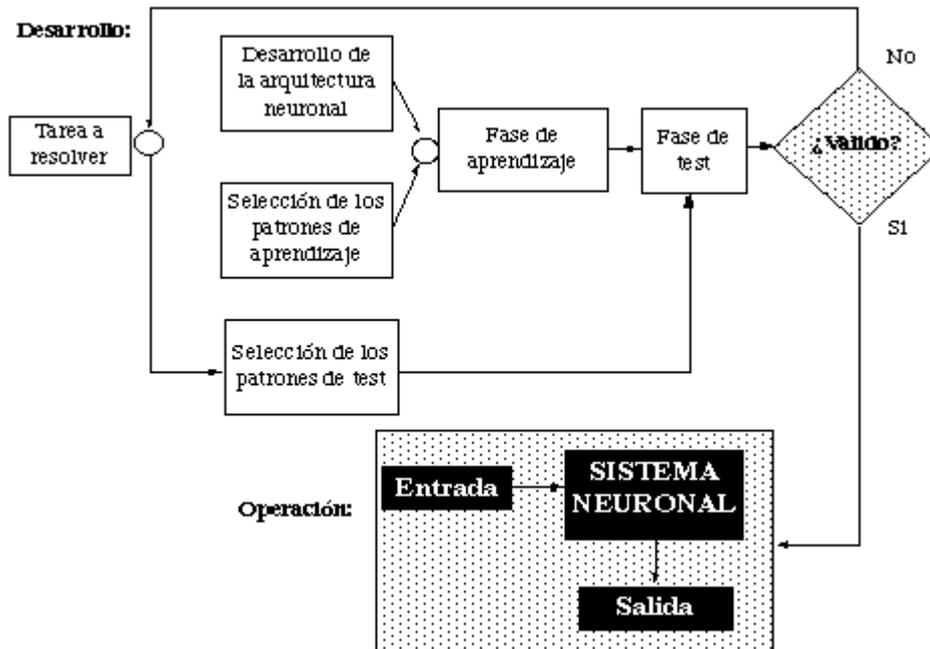


Figura B.3. Modo de trabajo con redes neuronales

El entrenamiento se realiza mediante patrones-ejemplo. Existen dos tipos de aprendizaje: supervisado y no supervisado. Si la red utiliza un tipo de aprendizaje supervisado debemos proporcionarle parejas de patrones entrada-salida y la red neuronal aprende a asociarlos. Si el entrenamiento es no supervisado, únicamente debemos suministrar a la red los datos de entrada para que extraiga los rasgos característicos esenciales. Durante la fase de aprendizaje en la mayor parte de los modelos se produce una variación de los pesos sinápticos, es decir, de la intensidad de interacción entre las neuronas.

B.2.2. Entrenamiento de la Red [42].

La técnica de retropropagación es semejante al aprendizaje supervisado, debido a que la red es entrenada con la respuesta esperada. Cada iteración modifica los pesos de conexión para minimizar el error de la respuesta (valor esperado valor estimado). El ajuste de los pesos, capa por capa, se calcula hacia atrás de la capa de salida a la capa de entrada.

Esta corrección se hace por:

$$\Delta W_{ji} = \eta \delta_j f'(a_j)$$

B.3



Donde ΔW_{jl} es el ajuste de peso entre la neurona j y la neurona i de la capa anterior; $f(a_j)$ es la salida de la neurona i , η es la velocidad de aprendizaje, y δ_j depende de la capa. Para la capa de salida, el δ_j es:

$$\delta_j = (Y_j - \hat{Y}_j) f'_j(a_j) \quad \text{B.4}$$

Donde Y_j es el valor esperado (valor estimado) y \hat{Y}_j es el valor de salida actual (valor estimado) de la neurona j . Para la capa oculta, el δ_j es:

$$\delta_j = f'_j(a_j) \sum_{k=1}^K \delta_k W_{kj} \quad \text{B.5}$$

Donde K es el número de neuronas en la próxima capa. La velocidad de aprendizaje juega un papel importante en el entrenamiento. Cuando esta velocidad es baja, la convergencia del peso a un óptimo es muy lenta, cuando la velocidad es demasiado alta, la red puede oscilar, o más seriamente puede conseguirse en un mínimo local. Para reducir estos problemas, el término α se usa y ΔW_{jl} se actualiza:

$$\Delta W_{jl} = \eta \delta_j f(a_j) + \alpha \Delta W_{jl}^{\text{Prev}} \quad \text{B.6}$$

Donde $\Delta W_{jl}^{\text{Prev}}$ denota la corrección en la anterior iteración. En algunos estudios, inicialmente $\alpha = 0.7$ y $\eta = 0.01$, entonces ellos se modifican según la importancia del error por el algoritmo siguiente:

Si $\text{error_actual} > \text{error_anterior} * 1.04$

entonces

$$\eta = \eta * 0.75,$$

$$\alpha = 0$$

B.7

de otra forma

$$\eta = \eta * 1.05,$$

$$\alpha = 0.95$$

fin.

El entrenamiento, desarrollado en un grupo de datos representativos, corre hasta que la suma cuadrada del error (SSE) se minimiza:

$$SSE = \frac{1}{2} \sum_{p=1}^P \sum_{j=1}^N (Y_{pj} - \hat{Y}_{pj})^2 \quad \text{B.8}$$



Donde: Y_{pj} es el valor de salida esperado, \hat{Y}_{pj} es el valor estimado por la red, $j = 1 \dots N$ es el número de registro y $p = 1 \dots P$ es el número de neuronas en la capa de salida.

La estructura de la red, el número de registros en el grupo de datos y el número de iteraciones determinan la duración del entrenamiento.

B.2.3. Prueba de la red [42].

Después del entrenamiento, el funcionamiento de la red tiene que ser probado. Como un análisis, una primera indicación es dada por el porcentaje de clasificaciones correctas del grupo de registros del entrenamiento. No obstante, el funcionamiento de la red con un grupo de prueba (grupo de datos similares sin usar durante el entrenamiento) es más pertinente.

En el paso de prueba, los datos de entrada son alimentados en la red y los valores deseados se comparan a los valores de salida de la red. El acuerdo o la discordancia de los resultados dan una indicación del funcionamiento de la red entrenada.



Apéndice C

Índice de error de desempeño

En la bibliografía reciente sobre el control se observa un interés creciente en la medición y formulación matemática del desempeño o funcionamiento de un sistema de control. Un índice de desempeño o funcionamiento es una medida cuantitativa del funcionamiento de un sistema y se elige de forma que resalte las especificaciones importantes del sistema.

Se puede calcular un índice de desempeño y utilizarlo para evaluar el funcionamiento de un sistema. Se necesita una medida cuantitativa del funcionamiento de un sistema para la operación de sistemas de control modernos adaptables, para la optimización de parámetros en un sistema de control y para el diseño de sistemas óptimos.

Si el propósito es mejorar el diseño de un sistema o diseñar un sistema de control adaptable, se debe elegir y medir un índice de funcionamiento. Para que un índice de funcionamiento resulte útil, siempre debe ser un número positivo o cero. Entonces el mejor sistema se define como aquel que minimiza este índice [43].

Un índice de funcionamiento adecuado es la integral del cuadrado del error, que se expresa:

$$I = \int_0^T e^2(t) dt \quad \text{C.1}$$

Donde el límite superior T es un tiempo finito elegido un tanto arbitrariamente de forma que la integral tienda a un valor en estado estacionario. El índice de funcionamiento de la ecuación C.1 se adapta fácilmente para mediciones prácticas, y el cuadrado del error es matemáticamente conveniente para propósitos analíticos y de cálculo.

La integral de $e^2(t)$ desde 0 a T es el área total bajo la curva de $e^2(t)$. Una característica de este índice de funcionamiento es que da una mayor importancia a los errores grandes y una menor a los errores pequeños.

Como un ejemplo de la aplicación de este índice presentamos la siguiente gráfica que se obtuvo de la simulación de los dos tipos de reguladores, se muestra el índice de comportamiento del voltaje en terminales del generador para una condición donde $P=0.9833$ p.u. aplicando una falla de 12 ciclos de duración:



En la figura C.1 se muestran los índices de error de funcionamiento para el regulador propuesto y para el regulador ST1, se puede observar que el regulador propuesto presenta un índice de funcionamiento mínimo en tanto el regulador ST1 presenta un índice de funcionamiento que se incrementa, así por lo explicado anteriormente en este apéndice se puede decir que el sistema propuesto es mejor porque minimiza este índice.

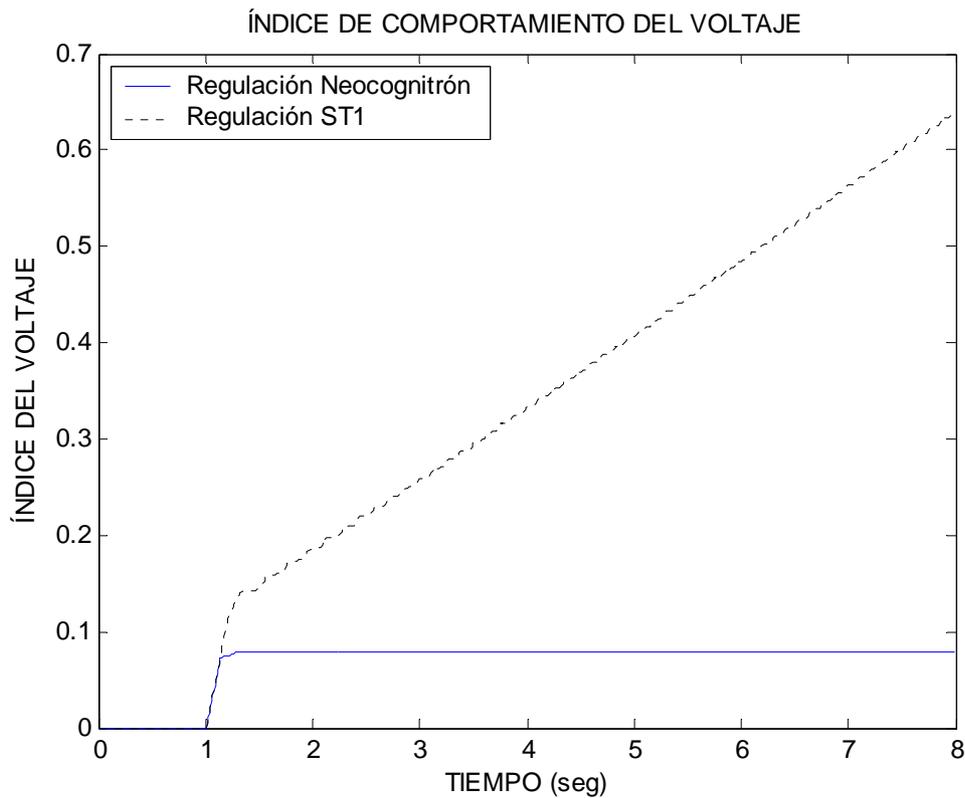


Figura C.1. Índice de comportamiento del voltaje en terminales.



Apéndice D

Programas de Simulación

En este apéndice se presenta el código en lenguaje C++ de los programas de simulación utilizados en este trabajo.

Los programas incluidos son:

- √ Máquina Síncrona.
- √ Máquina Síncrona Bus Infinito con Regulador ST1.
- √ Programa de Aplicación de la Máquina Síncrona con Regulador Neocognitrón.

D.1. Máquina Síncrona.

```

/*Programa máquina síncrona*/
#include <string.h>
#include <iostream.h>
#include <fstream.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>

double funSr(double,double,double);
double funS(double,double,double,double,double);
double fune1q(double,double,double,double,double,double);
double fune11q(double,double,double,double,double,double);
double fune11d(double,double,double,double,double);

void main()
{
    double to,tf,i,n;
    double Sr,S,e1q,e11q,e11d,Vf,Tm;
    double h,hh,h2,t;
    double ra,Xd,X1d,X11d,Xq,X1q,X11q,Kd,T1d0,T11d0,T11q0,M;
    double Vref;
    double Vt,f,Vd,Vq;
    double Xe,Re,Te,id,iq;
    double Tx,pi,wo,P,Q;
    double k_Sr,k_S,k_e1q,k_e11q,k_e11d;
    double m_Sr,m_S,m_e1q,m_e11q,m_e11d;
    double delta;
    FILE *fi;
    fi=fopen("MAQ1","w");
    to=0.0;
    tf=40.0;
    h=0.0005;
    hh=tf-to;

```



```
n=hh/h;
h2=h/2;
/*****DATOS DEL GENERADOR*****/
ra=0.0;
Xd=1.904;
X1d=0.312;
X11d=0.266;
Xq=1.881;
X11q=0.260;
Kd=3.0;
T1d0=5.66;
T11d0=0.041;
T11q0=0.065;
M=5.5294;
f=60.0;
P=0.1086;
Q=-0.0186;
Sr=18.94;
iq=0.1100;
id=0.0042;
S=0.0;
e11d=0.1782;
e11q=0.9794;
e1q=0.9796;
Tm=Te=0.1084;
Vf=0.9862;
Vref=1.0024;
t=to;

printf("Corriendo simulación\n\n");
for (i=0;i<=n;i++)
{
    t=t+h;
/***** (Método de integración RK) *****/
    k_Sr=h*funSr(Sr,wo,S);
    k_S=h*funS(S,Tm,Te,Kd,M);
    k_e1q=h*fune1q(e1q,Vf,Xd,X1d,id,T1d0);
    k_e11q=h*fune11q(e11q,e1q,X1d,X11d,id,T11d0);
    k_e11d=h*fune11d(e11d,Xq,X11q,iq,T11q0);
    m_Sr=h*funSr(Sr+h2*k_Sr,wo,S);
    m_S=h*funS(S+h2*k_S,Tm,Te,Kd,M);
    m_e1q=h*fune1q(e1q+h2*k_e1q,Vf,Xd,X1d,id,T1d0);
    m_e11q=h*fune11q(e11q+h2*k_e11q,e1q,X1d,X11d,id,T11d0);
    m_e11d=h*fune11d(e11d+h2*k_e11d,Xq,X11q,iq,T11q0);

/***** CALCULO DE VARIABLES DE ESTADO *****/
    Sr=Sr+m_Sr;
    S=S+m_S;
    e1q=e1q+m_e1q;
    e11q=e11q+m_e11q;
    e11d=e11d+m_e11d;
    delta=pow((Re+ra),2)+(Xe+X11d)*(Xe+X11q);
    id=((Re+ra)*(e11d-Vb*sin(Sr))+(Xe+X11q)*(e11q-Vb*cos(Sr)))/delta;
    iq=((Re+ra)*(e11q-Vb*cos(Sr))+(Xe+X11d)*(e11d-Vb*sin(Sr)))/delta;
    Vd=Vb*sin(Sr)+(Re*id)-(Xe*iq);
    Vq=Vb*cos(Sr)+(Re*iq)+(Xe*id);
```



```

e11d=Vd+(ra*id)-(X11q*iq);
e11q=Vq+(ra*iq)+(X11d*id);
Te=(e11d*id)+(e11q*iq)-((X11d-X11q)*(id*iq));
Vt=sqrt((Vd*Vd)+(Vq*Vq));
P=(Vd*id)+(Vq*iq);
Q=(Vq*id)-(Vd*iq);
/*****IMPRESION DE RESULTADOS EN ARCHIVO DE DATOS*****/
fprintf(fi," %f %f %f %f %f \n",t,Sr,Vt,P,Q);
if( i+15 >= n )
printf("%f %f %f %f %f \n",t,Sr,Vt,P,Q);
}
/*****CIERRA ARCHIVO DE DATOS Y FIN DEL PROGRAMA *****/
fclose(fi);
printf("READY\n\n");
printf("DATAFILE: 'MAQ1'");
getche();
}
/*****SUBROUTINAS*****/
double funSr(double Sr,double wo,double S)
{
double funSr;
funSr=wo*S;
return(funSr);
}
double funS(double S,double Tm,double Te,double Kd,double M)
{
double funS;
funS=-S*(Kd/M)+(Tm/M)-(Te/M);
return(funS);
}
double fune1q(double e1q,double Vf,double Xd,double X1d,double id,double T1d0)
{
double fune1q;
fune1q=(Vf/T1d0)-id*(Xd/T1d0)+id*(X1d/T1d0)-(e1q/T1d0);
return(fune1q);
}
double fune11q(double e11q,double e1q,double X1d,double X11d,double id,double T11d0)
{
double fune11q;
fune11q=(e1q/T11d0)-id*(X1d/T11d0)+id*(X11d/T11d0)-(e11q/T11d0);
return(fune11q);
}
double fune11d(double e11d,double Xq,double X11q,double iq,double T11q0)
{
double fune11d;
fune11d=iq*(Xq/T11q0)-iq*(X11q/T11q0)-(e11d/T11q0);
return(fune11d);
}

```



D.2. Máquina Síncrona Bus Infinito con regulador ST1.

```

/*Programa máquina síncrona bus infinito con regulador ST1*/
#include <string.h>
#include <iostream.h>
#include <fstream.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#define DIM 30

void Datos ( );
void Cinicial ( );
void VUg(double cont, double Tm, char mando) ;
void MatrizA(double A[DIM][DIM]);
void VectorB(double B[DIM]);
void VectorVX(double VX[DIM]);
void VTR(double VX[DIM]);
void PotC();
void Potencia(double VX[DIM]);
void idiqTe(double VX[DIM]);
void Rungekutta(int numva, double h, double A[DIM][DIM], double B[DIM], double VX[DIM],int
u2);

void main()
{
    Int numva, i;
    char nombre[16], ctrol, mando, oper, ccto;
    double A[DIM][DIM],B[DIM],VX[DIM],XP[DIM],DX[DIM][DIM],KX[DIM];
    double re, xe, ra, xd, xpd, xppd, xq, xppq, Kd, M, Tx, tpdo, tppdo, tppqo;
    double Tsm, Trh, Tch, Fhp, u2, cont, tf, h, gb, fo, Ug, U, s, Pgv, Php, Tm, theta, eqx, eqy, phi,
    delta, deltar, iq, id, Te,Pcal,Vref, Verror, eppd, eppq, epq, Vf, Vt, Vb, ear, eai, Pr, Pi, I, Ka, Ta,
    double Kf, Tf, AUX1,w[10][10],t1,delta1;

    /*****Parámetros del sistema máquina bus infinito*****/
    fo=60.0;      /*Frecuencia base*/
    ra=0.0;      //Resistencia de armadura
    re=0.01;     //Resistencia equivalente
    xe=0.15;     //Reactancia equivalente
    xd=1.904;    //Reactancia síncrona
    xpd=0.312;   //Reactancia transitoria en eje d
    xppd=0.266;  //Reactancia subtransitoria en eje d
    xq=1.881;    //Reactancia transitoria en eje q
    xppq=0.260;  //Reactancia subtransitoria en eje q
    Kd = 3.0;    //Coeficiente de amortiguamiento
    M = 5.5294;  //Momento de inercia
    Tx = 0.025;  //Constante de tiempo del excitador
    tpdo = 5.66; //Constante de tiempo transitoria en eje d
    tppdo = 0.041;//Constante de tiempo subtransitoria en eje d
    tppqo = 0.065;//Constante de tiempo subtransitoria en eje q
    Tsm = 0.1;   //Constante de tiempo del servomotor de la válvula
    Trh = 5.0;   //Constante de tiempo de recalentamiento
    Tch = 0.15;  //Constante de tiempo de vapor
    Fhp = 0.33;  //Fracción de la potencia total generada

```



```
Ta = 0.02; //Constante de tiempo del regulador de voltaje
Ka = 400.0; //Ganancia del regulador de voltaje
Tf = 1.0; //Constante de tiempo del sistema de excitación
Kf = 0.008; //Ganancia del estabilizador del sistema de excitación
numva=9; //Número de variables
u2=1.0; //Carga
cont=0.0; //Contador para la simulación
tf=8.0; //Tiempo final de simulación
h=0.001; //Paso de integración
t1=delta1=0.01;
```

```
Datos ( );
Cinicial ( );
VUg(cont, Tm, mando);
MatrizA(A);
VectorB(B);
VectorVX(VX);
Rungekutta(numva,h,A,B,VX,u2);
return 0;
}
```

/***** Funciones *****/

void Datos () //Datos del sistema máquina bus infinito

```
{
printf("Voltaje en terminales ( p.u. ) : \n");
scanf("%f",&VT);
printf("Ángulo del voltaje en terminales ( grados ) : \n");
scanf("%f",&theta);
printf("Voltaje del bus infinito ( p.u. ) : \n");
scanf("%f",&Vb);
xdd=sqrt(re^2+xe^2);
theta=asin((Pote*xdd)/(Vt*Vb));
theta=theta*180/pi;
clrscr();
printf("Máquina con control :\n");
ctrol=toupper(getch());
printf("%c", ctrol);
if (ctrol == 'S')
{
printf("Regulador ST1 (1)");
printf("Escoja una opción :\n");
scanf("%f",&cint);
}
printf("Máquina con variación de carga : \n");
mando=toupper(getch());
printf("%c",mando);
if ( mando == 'S' )
{
printf("\nTiempo final t= ",tf);
scanf("%f",&tf);
}
printf("Máquina en corto circuito :\n ");
ccto=toupper(getch());
printf("%c",ccto);
if ( ccto == 'S' )
{
```



```

printf("\nTiempo del corto en ciclos ",tc);
scanf("%f",&tc);
}
printf("\nTiempo de inicio del corto ",tin);
scanf("%f",&tin);
}
printf("\nTiempo final de la simulación",tf);
scanf("%f",&tf);}
else
{tin=tf}
}

```

```
void Cinicial ( ) //Cálculo de condiciones iniciales
```

```

{
theta=((theta*M_PI)/180);
PotC();
I=sqrt(Pr*Pr+Pi*Pi)/Vt;
if (I<=1.0e-06)
phi=0.0;
else
phi = atan2(Pi,Pr);
eqx=Vt+ra*I*cos(phi)+xq*I*sin(phi);
eqy=-ra*I*sin(phi)+xq*I*cos(phi);
delta = atan2(eqy, eqx);
id=I*sin(delta+phi);
iq=I*cos(delta+phi);
eppd=Vt*sin(delta)+ra*id-xppq*iq;
eppq=Vt*cos(delta)+ra*iq+xppd*id;
epq=eppq+(xpd-xppd)*id;
Vf=(xd-xpd)*id+epq;
Pcal=id*eppd+iq*eppq;
Te=Pcal;
deltar=(theta+delta);
Vref=Vt;
if ( ctrol == 'S' )
{
Vref=(Vf/Ka)+Vt, Verror=s=0.0,
U=Vf, Ug=Pgv=Php=Tm=Te, numva=11;
}
else
{
U=Vf, s=0.0, Ug=Pgv=Php=Tm=Te;
}
}

```

```
void MatrizA(double A[DIM][DIM])
```

```

{
int i, j, k;
for(i=0;i<=numva;i++)
for(j=0;j<=numva;j++)
A[i][j]=0.0;
A[0][1]= 2*M_PI*fo;
A[1][1]= -Kd/M;
A[1][8]= 1/M;
A[2][2]= -1/tpdo;
A[2][5]= 1/tpdo;

```



```

A[3][2]= 1/tppdo;
A[3][3]= -1/tppdo;
A[4][4]= -1/tppqo;
A[5][5]= -1/Tx;
A[6][6]= -1/Tsm;
A[7][6]= 1/Tch;
A[7][7]= -1/Tch;
A[8][6]= Fhp/(Trh*Tch);
A[8][7]= (1-Fhp/Tch)/Trh;
A[8][8]= -1/Trh;
if ( ctrl == 'S' )
{
printf("\n Máquina con controles 1:");printf("%c", ctrl);
A[9][5] = -Kf/(Tf*Tx);
A[9][9]= -1/Tf;
A[9][10] = Kf/(Tf*Tx);
A[10][9] = -Ka/Ta;
A[10][10] = -1/Ta;
}
}

void VectorB(double B[DIM])
{
int i;
for(i=0;i<=numva;i++)
B[i]=0.0;
B[1]= -Te/M;
B[2]= -(xd-xpd)*id/tpdo;
B[3]= -(xpd-xppd)*id/tppdo;
B[4]= (xq-xppq)*iq/tppqo;
B[5]= U/Tx;
B[6]= Ug/Tsm;
if ( ctrl == 'S' )
{ B[10] = (Ka/Ta)*(Vref-Vt);}
}

void VectorVX(double VX[DIM])
{
VX[0] = deltar;
VX[1] = s;
VX[2] = epq;
VX[3] = eppq;
VX[4] = eppd;
VX[5] = Vf;
VX[6] = Pgv;
VX[7] = Php;
VX[8] = Tm;
if ( ctrl == 'S' )
{
VX[9] = Verror;
VX[10] = U;
}
}

void idiqTe(double VX[DIM]) //Cálculo de corrientes y par eléctrico
{

```



```

double r_t, e_q, e_d, x_d, x_q, det;
r_t=e_q=e_d=x_d=x_q=det=0.0;
r_t=(re+ra);
x_d=(xe+xppd);
x_q=(xe+xppq);
det=r_t*r_t+x_d*x_q;
e_d=VX[4]-Vb*sin(VX[0]);
e_q=VX[3]-Vb*cos(VX[0]);
id=(r_t*e_d+x_q*e_q)/det;
iq=(r_t*e_q-x_d*e_d)/det;
Te=VX[4]*id+VX[3]*iq;
}

void VTR(double VX[DIM]) //Cálculo de voltajes
{
double vd,vq;
vd= Vb*sin(VX[0])+re*id-xe*iq;
vq= Vb*cos(VX[0])+re*iq+xe*id;
Vt=sqrt(vd*vd+vq*vq);
}

void PotC( ) //Cálculo de potencias
{
double cear, ceai,rr,ri,rr2,ri2,rr3,ri3;
ear=Vt*cos(theta);
eai=Vt*sin(theta);
cear=ear;
ceai=-eai; // conjugado
rr=ear-Vb;
ri=eai-0; // resta
rr2= (rr*re+ri*xe)/(re*re+xe*xe); // división
ri2= (ri*re-rr*xe)/(re*re+xe*xe);
rr3=cear*rr2-ceai*ri2; // multiplicación
ri3=cear*ri2+ceai*rr2;
Pr=rr3;
Pi=-ri3; // potencias
}

//Método de integración
void Rungekutta(int numva,double h,double A[DIM][DIM], double B[DIM], double VX[DIM],int u2)
{
FILE *fp;
if(control=='s')
{ printf(" Máquina con controles :\n");}
else { printf(" Máquina sin controles :\n");}
printf(" Nombre del archivo de salida :\n");}
scanf("%s",nombre);
fp=fopen(nombre,"w");
double cont,suma,z;
int i,j,k;
//Escritura de los resultados en el archivo de salida.
fprintf(fp,"%f %f %f %f %f %f %f %f %f",cont,((VX[0]*180)/M_PI),VX[5],Vt,Pr,Pi,Ug,VX[1]);
fprintf(fp,"\n");
for(cont=cont+h;cont<=tf;cont+=h)
{
XP[0]= XP[1]=XP[2]=XP[3]=u2;

```



```

    for(k=0;k<=numva;k++)
    {
    DX[k][0]=0;
    for(i=0;i<numva;i++)
    DX[k][0]=DX[k][0]+A[k][i]*VX[i];
    }
    for(i=0;i<numva;i++)
    DX[i][0]=DX[i][0]+B[i]*XP[0];
    z=0.5*h;
    for(i=1;i<4;i++)
    {
    if( i == 3 ) z=h;
    for(j=0;j<numva;j++) KX[j]=VX[j]+z*DX[j][i-1];
    for(k=0;k<=numva;k++)
    {
    DX[k][i]=0;
    for(j=0;j<numva;j++)
    DX[k][i]=DX[k][i]+A[k][j]*KX[j];
    }
    for(j=0;j<numva;j++)
    DX[j][i]=DX[j][i]+B[j]*XP[i];
    }
    for(k=0;k<=numva;k++)
    VX[k]=VX[k]+h*(DX[k][0]+2*DX[k][1]+2*DX[k][2]+DX[k][3])/6;
    idiqTe(VX);
    if ( ccto == 'S' )
    {
    if ( cont >= 1.0 && cont <= 1.150)
    { Vb = 0.0;}
    else { Vb = 1.0;}
    if ( cont >= 11.0 && cont <= 11.150)
    { Vb = 0.0;}
    else
    { Vb = 1.0;}
    }
    if (cont>=21.0)
    {
    if ( cont >= 21.0 && cont <= 21.150)
    { Vb = 0.0;}
    else
    { Vb = 1.0; }
    }
    }

VTR(VX); //Mando del gobernador
potencia(VX);
if ( ctrol == 'S' )
{ U=VX[10], Ug=VX[8]; }
else
{ U=VX[5], Ug=VX[8]; }
VUg(cont, Tm, mando);
VectorB(B);
}
fclose(fp);
}

```



```
Void VUg(double t, double inic, char mando) //Incremento de carga
```

```
{
double tt[8]={10, 20, 30, 40, 50, 60, 70, 80};
double smg, m[4],y[4];
if (mando !='s')
{Ug=inic;}
else{
y[0] = inic+20*(1-inic)/100; y[1] = inic+50*(1-inic)/100;
y[2] = inic+75*(1-inic)/100; y[3] = inic+100*(1-inic)/100;
m[0] = (y[0] - inic) /(tt[1]-tt[0]);
m[1] = (y[1] - y[0])/(tt[3]-tt[2]);
m[2] = (y[2] - y[1])/(tt[5]-tt[4]);
m[3] = (y[3] - y[2])/(tt[7]-tt[6]);
if( t <= tt[0] ) smg = inic;
if( t > tt[0] && t <= tt[1] ) smg = inic + m[0] * ( t - tt[0] );
if( t > tt[1] && t <= tt[2] ) smg = y[0];
if( t > tt[2] && t <= tt[3] ) smg = y[0] + m[1] * ( t - tt[2] );
if( t > tt[3] && t <= tt[4] ) smg = y[1];
if( t > tt[4] && t <= tt[5] ) smg = y[1] + m[2] * ( t - tt[4] );
if( t > tt[5] && t <= tt[6] ) smg = y[2];
if( t > tt[6] && t <= tt[7] ) smg = y[2] + m[3] * ( t - tt[6] );
if( t > tt[7] ) smg = y[3];
Ug = smg;
}
}
```

```
void potencia(double VX[DIM]) //Cálculo de potencias nuevas
```

```
{
double rr3,ri3,vd,vq;
vd = Vb*sin( VX[0] ) + re *id - xe *iq;
vq = Vb*cos( VX[0] ) + re *iq + xe *id;
rr3=vd*id-vq*(-iq);
ri3=vd*(-iq)+vq*id;
Pr = rr3; Pi = ri3;
}
```



D.3. Programa de Aplicación de la Máquina Síncrona con Regulador Neocognitrón.

Para la simulación de la tesis se realizó un programa de aplicación que contiene al sistema máquina bus infinito con el regulador Neocognitrón. En esta sección se presenta el diagrama de flujo del programa de aplicación con el regulador Neocognitrón, así como una breve descripción del manejo del programa.

El programa se divide en 4 partes:

1. Datos del sistema máquina bus infinito.
2. Cálculo de condiciones iniciales.
3. Método de integración.
4. Salida de resultados en archivo.

El programa contiene las llamadas de las subrutinas utilizadas para la simulación y los valores de los parámetros del sistema máquina bus infinito necesarios.

En la primera parte del programa se pregunta al usuario datos adicionales para la simulación como el voltaje en terminales, el ángulo del voltaje en terminales y el voltaje del bus. Estos datos se deben proporcionar como se muestra a continuación:

Datos adicionales necesarios para el sistema máquina bus infinito:

¿VOLTAJE EN TERMINALES? = 1.0

¿ÁNGULO DEL VOLTAJE EN TERMINALES? = 7.0

¿VOLTAJE DEL BUS INFINITO? = 1.0

En esta misma parte se le pregunta al usuario si quiere aplicar falla, al contestar que si el usuario debe proporcionar el tiempo inicial de la falla, el tiempo de la falla en ciclos y el tiempo final de la simulación, esto se hace como se muestra a continuación:

¿MÁQUINA EN CORTO CIRCUITO?

Al proporcionar el usuario una respuesta afirmativa, entonces se realizan las siguientes preguntas referente a la falla.

¿TIEMPO DE INICIO DEL CORTO CIRCUITO? = 1.0

¿TIEMPO DEL CORTO CIRCUITO EN CICLOS? = 6.0

¿TIEMPO FINAL DE LA SIMULACIÓN? = 8.0

La falla puede ser iniciada en el tiempo que el usuario proporcione, se recomienda proporcionar un tiempo de inicio entre 0.1 y 1.0 seg. Debido a que el comportamiento de la máquina puede ser inadvertido si se aplica una falla después de estos límites de tiempo. El usuario también debe proporcionar el tiempo de duración de la falla en ciclos, para nuestro estudio se proporcionaron valores de 6, 9 y 12 ciclos obteniendo buenos resultados durante la realización de las pruebas.



La segunda parte del programa realiza el cálculo de las condiciones iniciales del sistema para las cuales empezará la simulación.

En la tercera parte del programa se contempla el método de integración (Runge Kutta), se realiza el cálculo del sistema de ecuaciones.

Por último, en la cuarta parte del programa se crea un archivo de salida para almacenar los resultados de las variables a graficar. Esto se realiza de la siguiente forma:

¿DE EL NOMBRE DEL ARCHIVO DE SALIDA?

Aquí el usuario proporciona el nombre del archivo de resultados que desee, por ejemplo: prueba.dat

Es importante mencionar que el programa genera un archivo de salida donde se almacenan las variables del sistema necesarias para analizar el comportamiento de nuestro sistema.

Así, las gráficas presentadas en el capítulo 4 fueron realizadas en un gráficator comercial que permitiera mostrar el comportamiento de nuestro sistema de manera satisfactoria.

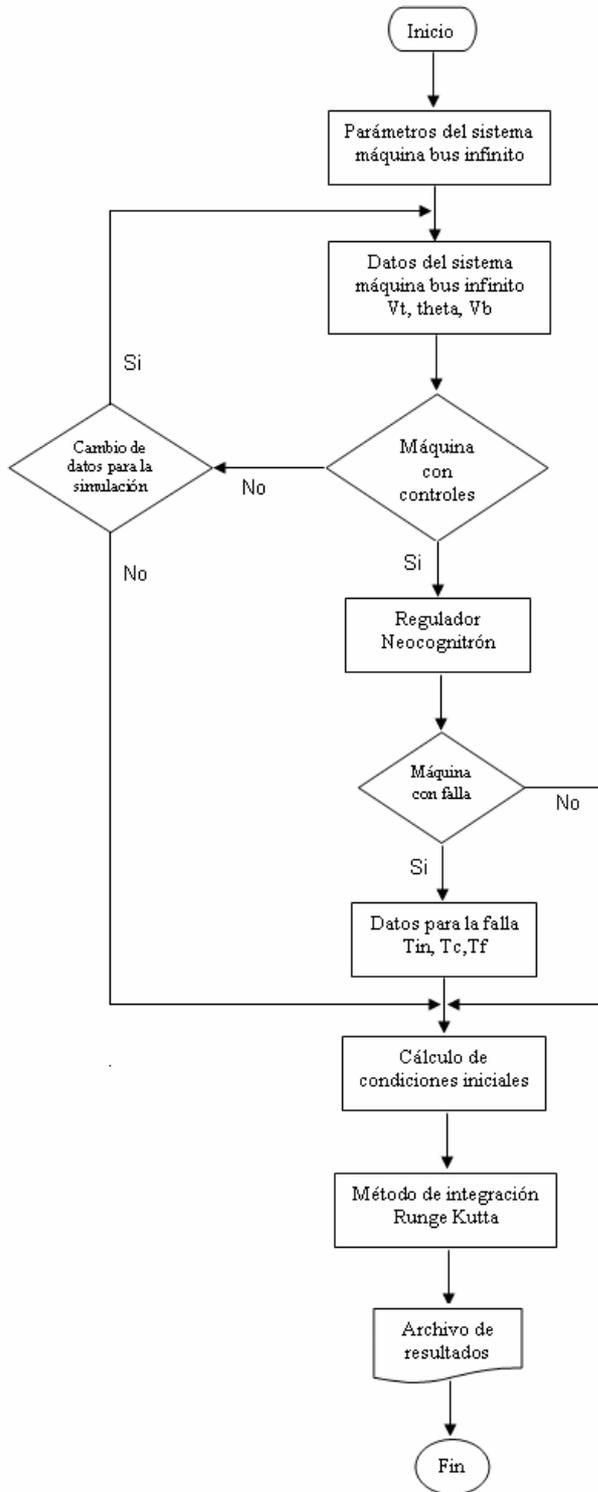


Figura D.1. Diagrama de flujo del programa de aplicación con el regulador Neocognitrón



```

/* Programa de aplicación de la máquina síncrona bus infinito con regulador neocognitrón*/
#include <string.h>
#include <iostream.h>
#include <fstream.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#include <ctype.h>
#include <stdarg.h>
#include <matrix.h>

#define DIM 30

void Datos ( );
void Cincial ( );
void VUg(double cont, double Tm, char mando);
void MatrizA(double A[DIM][DIM]);
void VectorB(double B[DIM]);
void VectorVX(double VX[DIM]);
void VTR(double VX[DIM]);
void PotC();
void Potencia(double VX[DIM]);
void idiqTe(double VX[DIM]);
void Rungekutta(int numva, double h, double A[DIM][DIM], double B[DIM], double VX[DIM],int
u2);
double neo(double *W1, double *W2, double *W3, double *W4,double *Error, double *Verror,
double z);

void main()
{
    Int numva, i;
    char nombre[16], ctrol, mando, oper, ccto;
    double A[DIM][DIM],B[DIM],VX[DIM],XP[DIM],DX[DIM][DIM],KX[DIM];
    double re, xe, ra, xd, xpd, xppd, xq, xppq, Kd, M, Tx, tpdo, tppdo, tppqo;
    double Tsm, Trh, Tch, Fhp, u2, cont, tf, h, gb, fo, Ug, U, s, Pgv, Php, Tm, theta, eqx, eqy, phi,
    delta, deltar, iq, id, Te,Pcal,Vref, Verror, eppd, eppq, epq, Vf, Vt, Vb, ear, eai, Pr, Pi, I, Ka, Ta,
    double Kf, Tf, AUX1,w[10][10],t1,delta1;

    /*****Parámetros del sistema máquina bus infinito*****/
    fo=60.0;
    ra=0.0;
    re=0.01;
    xe=0.15;
    xd=1.904;
    xpd=0.312;
    xppd=0.266;
    xq=1.881;
    xppq=0.260;
    Kd = 3.0;
    M = 5.5294;
    Tx = 0.025;
    tpdo = 5.66;
    tppdo = 0.041;
    tppqo = 0.065;
    Tsm = 0.1;

```



```

Trh = 5.0;
Tch = 0.15;
Fhp = 0.33;
Ta = 0.02;
Ka = 400.0;
Tf = 1.0;
Kf = 0.008;
numva=9;
u2=1.0;
cont=0.0;
tf=8.0;
h=0.001;
t1=delta1=0.01;

Datos ( );
Cinicial ( );
VUg(cont,Tm,mando);
MatrizA(A);
VectorB(B);
VectorVX(VX);
VTR(VX);
Rungekutta(numva,h,A,B,VX,u2);
return 0;
}

```

/****** Funciones *****/

```

void Datos ( )
{
printf("Voltaje en terminales ( p.u. ) :\n ");
scanf("%f",&VT);
printf("Ángulo del voltaje en terminales ( grados ) :\n");
scanf("%f",&theta);
printf("Voltaje del bus infinito ( p.u. ) :\n ");
scanf("%f",&Vb);
xdd=sqrt(re^2+xe^2);
theta=asin((Pote*xdd)/(Vt*Vb));
theta=theta*180/pi;
clrscr();
printf("Máquina con control : \n");
ctrol=toupper(getch());
printf("%c", ctrol);
if (ctrol == 'S')
{
printf("Regulador ST1 (1) \n");
printf("Regulador neocognitrón (2) \n");
printf("Escoja una opción :\n");
scanf("%f",&cint);
}
printf("Máquina con variación de carga : \n ");
mando=toupper(getch());
printf("%c",mando);
if ( mando == 'S' )
{
printf("\nTiempo final t= ",tf);
scanf("%f",&tf);
}
}

```



```

}
printf("Máquina en corto circuito : \n");
ccto=toupper(getch());
printf("%c",ccto);
if ( ccto == 'S' )
{
printf("\nTiempo del corto en ciclos ",tc);
scanf("%f",&tc);
}
printf("\nTiempo de inicio del corto ",tin);
scanf("%f",&tin);
}
printf("\nTiempo final de la simulación",tf);
scanf("%f",&tf);}
else
{tin=tf}
}

void Cincial ( ) //cálculo de condiciones iniciales
{
theta=((theta*M_PI)/180);
clrscr();
PotC();
I=sqrt(Pr*Pr+Pi*Pi)/Vt;
if (I<=1.0e-06)
phi=0.0;
else
phi = atan2(Pi,Pr);
eqx=Vt+ra*I*cos(phi)+xq*I*sin(phi);
eqy=-ra*I*sin(phi)+xq*I*cos(phi);
delta = atan2(eqy, eqx);
id=I*sin(delta+phi);
iq=I*cos(delta+phi);
eppd=Vt*sin(delta)+ra*id-xppq*iq;
eppq=Vt*cos(delta)+ra*iq+xppd*id;
epq=eppq+(xpd-xppd)*id;
Vf=(xd-xpd)*id+epq;
Pcal=id*eppd+iq*eppq;
Te=Pcal;
deltar=(theta+delta);
Vref=Vt;
if ( ctrol == 'S' )
{
Vref=(Vf/Ka)+Vt, Verror=s=0.0,
U=Vf, Ug=Pgv=Php=Tm=Te, numva=11;
}
else
{
U=Vf, s=0.0, Ug=Pgv=Php=Tm=Te;
}

if ( cint == 2)
{

```



//PESOS DEL NEOCOGNITRÓN

net[1]=1; net[2]=3; net[3]=1; net[4]=5; net[5]=1; net[6]=1;
net[7]=2; net[8]=2; net[9]=2; net[10]=1; net[11]=3; net[12]=1;

//CAPA DE CÉLULAS 1

w1[1][1]=3.15; w1[1][2]=1.35; w1[1][3]=4.67; w1[1][4]=8.83; w1[1][5]=11.9; w1[1][6]=9.25;
w1[1][7]=15.5; w1[1][8]=-1.53; w1[1][9]=-3.08; w1[1][10]=-4.77; w1[1][11]=0.0; w1[1][12]=0.0;
w1[2][1]=-0.51; w1[2][2]=1.18; w1[2][3]=2.55; w1[2][4]=2.12; w1[2][5]=0.046; w1[2][6]=-0.39;
w1[2][7]=3.05; w1[2][8]=1.15; w1[2][9]=-0.13; w1[2][10]=-0.38; w1[2][11]=0.0; w1[2][12]=0.0;

//CAPA DE CÉLULAS 2

w2[1][1]=0.77; w2[1][2]=-0.36; w2[1][3]=1.74; w2[1][4]=-2.11;
w2[1][5]=0.71; w2[1][6]=0.5; w2[1][7]=0.43; w2[1][8]=1.15;
w2[1][9]=-0.70; w2[1][10]=-0.01; w2[1][11]=0.0; w2[1][12]=0.0;

w2[2][1]=0.31; w2[2][2]= -0.50; w2[2][3]= -0.81; w2[2][4]= 2.4;
w2[2][5]=-0.87; w2[2][6]= 0.95; w2[2][7]= -1.18; w2[2][8]= -0.16;
w2[2][9]=0.09; w2[2][10]=-0.20; w2[2][11]=0.0; w2[2][12]=0.0;

w2[3][1]=-1.05; w2[3][2]=1.44; w2[3][3]=2.06; w2[3][4]=1.46;
w2[3][5]=6.75; w2[3][6]=7.00; w2[3][7]=2.79; w2[3][8]=-0.61;
w2[3][9]=-1.03; w2[3][10]=-2.13; w2[3][11]=0.00; w2[3][12]=0.00;

w2[4][1]=1.36; w2[4][2]=2.46; w2[4][3]=0.75; w2[4][4]= 1.16;
w2[4][5]=1.69; w2[4][6]=-1.03; w2[4][7]=1.02; w2[4][8]=1.46;
w2[4][9]=-0.004; w2[4][10]=-0.32; w2[4][11]=0.0; w2[4][12]=0.0;

w2[5][1]=-0.10; w2[5][2]=0.68; w2[5][3]=-0.76; w2[5][4]=-1.06;
w2[5][5]=0.40; w2[5][6]=-1.92; w2[5][7]=1.73; w2[5][8]=-1.50;
w2[5][9]=-0.83; w2[5][10]=1.02; w2[5][11]=0.0; w2[5][12]=0.0;

w2[6][1]=-1.71; w2[6][2]=-3.14; w2[6][3]=-5.49; w2[6][4]=-2.82;
w2[6][5]=-2.6; w2[6][6]=-3.61; w2[6][7]=-5.35; w2[6][8]=-0.16;
w2[6][9]=-0.12; w2[6][10]=0.76; w2[6][11]=0.0; w2[6][12]=0.0;

w2[7][1]=-1.11; w2[7][2]=-2.01; w2[7][3]=1.42; w2[7][4]=2.00;
w2[7][5]=0.99; w2[7][6]=0.60; w2[7][7]=4.85; w2[7][8]=0.27;
w2[7][9]=-0.57; w2[7][10]=-1.38; w2[7][11]=0.0; w2[7][12]=0.0;

w2[8][1]=-1.06; w2[8][2]=2.25; w2[8][3]=-1.79; w2[8][4]=-1.4255;
w2[8][5]=1.2; w2[8][6]=-0.51; w2[8][7]=1.36; w2[8][8]=0.4820;
w2[8][9]=-0.08; w2[8][10]=-0.58; w2[8][11]=0.0; w2[8][12]=0.0;

w2[9][1]=-1.11; w2[9][2]=-2.01; w2[9][3]=1.42; w2[9][4]=2.00;
w2[9][5]=0.99; w2[9][6]=0.60; w2[9][7]=4.85; w2[9][8]=0.27;
w2[9][9]=-0.57; w2[9][10]=-1.38; w2[9][11]=0.0; w2[9][12]=0.0;

w2[10][1]=-1.71; w2[10][2]=-3.14; w2[10][3]=-5.49; w2[10][4]=-2.82;
w2[10][5]=-2.6; w2[10][6]=-3.61; w2[10][7]=-5.35; w2[10][8]=-0.16;
w2[10][9]=-0.12; w2[10][10]=0.76; w2[10][11]=0.0; w2[10][12]=0.0;

w2[11][1]=0.18; w2[11][2]=0.84; w2[11][3]= -0.78; w2[11][4]= 2.22;
w2[11][5]=0.06; w2[11][6]=-1.95; w2[11][7]= 1.66; w2[11][8]= 0.06;
w2[11][9]=-0.78; w2[11][10]=-1.50; w2[11][11]=0.0; w2[11][12]=0.0000;

w2[12][1]=3.5; w2[12][2]=0.76; w2[12][3]=1.43; w2[12][4]=-0.30;
w2[12][5]=-1.45; w2[12][6]=3.44; w2[12][7]=5.08; w2[12][8]= -1.32;



w2[12][9]=2.57; w2[12][10]=3.06; w2[12][11]=0.0; w2[12][12]=0.0;

//CAPA DE CELULAS 3, PESOS ACTUALIZADOS

w3[1][1]=0.24; w3[1][2]= -0.2; w3[1][3]= -0.61; w3[1][4]= 1.4;
w3[1][5]=-0.42; w3[1][6]= 0.1; w3[1][7]= -1.23; w3[1][8]= -0.2;
w3[1][9]=0.1; w3[1][10]=-0.2; w3[1][11]=0.0; w3[1][12]=0.0;

w3[2][1]=0.2; w3[2][2]=0.64; w3[2][3]= -0.52; w3[2][4]= 1.22;
w3[2][5]=0.04; w3[2][6]=-1.2; w3[2][7]= 1.22; w3[2][8]= 0.04;
w3[2][9]=-0.8; w3[2][10]=-1.30; w3[2][11]=0.0; w3[2][12]=0.0000;

w3[3][1]=1.22; w3[3][2]=2.32; w3[3][3]=1.75; w3[3][4]=1.3;
w3[3][5]=1.34; w3[3][6]=-1.03; w3[3][7]=0.02; w3[3][8]=2.46;
w3[3][9]=-0.04; w3[3][10]=-0.5; w3[3][11]=0.0; w3[3][12]=0.0;

w3[4][1]=-1.8; w3[4][2]=2.25; w3[4][3]=-0.79; w3[4][4]=-1.42;
w3[4][5]=1.3; w3[4][6]=-0.89; w3[4][7]=2.36; w3[4][8]=0.48;
w3[4][9]=-0.8; w3[4][10]=-0.6; w3[4][11]=0.0; w3[4][12]=0.0;

w3[5][1]=-3.29; w3[5][2]=-1.4; w3[5][3]=-3.23; w3[5][4]=-3.17;
w3[5][5]=-9.2; w3[5][6]=-12.8; w3[5][7]=-11.9; w3[5][8]=0.8;
w3[5][9]=3.33; w3[5][10]=1.6; w3[5][11]=0.0; w3[5][12]=0.0;

w3[6][1]=-2.71; w3[6][2]=-1.14; w3[6][3]=-4.5; w3[6][4]=-2.98;
w3[6][5]=-1.6; w3[6][6]=-2.61; w3[6][7]=-3.4; w3[6][8]=-0.23;
w3[6][9]=-1.12; w3[6][10]=0.76; w3[6][11]=0.0; w3[6][12]=0.0;

w3[7][1]=2.0; w3[7][2]=0.76; w3[7][3]=1.5; w3[7][4]=-0.35;
w3[7][5]=-1.45; w3[7][6]=5.44; w3[7][7]=4.08; w3[7][8]= -1.22;
w3[7][9]=1.57; w3[7][10]=4.06; w3[7][11]=0.0; w3[7][12]=0.0;

w3[8][1]=-2.05; w3[8][2]=1.44; w3[8][3]=2.06; w3[8][4]=1.64;
w3[8][5]=6.75; w3[8][6]=7.00; w3[8][7]=1.79; w3[8][8]=-0.16;
w3[8][9]=-1.03; w3[8][10]=-1.13; w3[8][11]=0.00; w3[8][12]=0.00;

w3[9][1]=-1.98; w3[9][2]=-1.01; w3[9][3]=2.42; w3[9][4]=2.34;
w3[9][5]=1.12; w3[9][6]=0.60; w3[9][7]=4.85; w3[9][8]=0.34;
w3[9][9]=-0.23; w3[9][10]=-1.4; w3[9][11]=0.0; w3[9][12]=0.0;

w3[10][1]=-0.54; w3[10][2]=-2.58; w3[10][3]=1.84; w3[10][4]=-2.81;
w3[10][5]=1.13; w3[10][6]=1.5; w3[10][7]=3.16; w3[10][8]=-0.45;
w3[10][9]=-2.20; w3[10][10]=0.17; w3[10][11]=0.0; w3[10][12]=0.0;

w3[11][1]=1.83; w3[11][2]=0.67; w3[11][3]=2.14; w3[11][4]=0.60;
w3[11][5]=3.70; w3[11][6]=0.87; w3[11][7]=-0.24; w3[11][8]=0.84;
w3[11][9]=-0.64; w3[11][10]=2.46; w3[11][11]=0.0; w3[11][12]=0.0;

w3[12][1]=-0.2; w3[12][2]=0.68; w3[12][3]=-0.67; w3[12][4]=-1.50;
w3[12][5]=0.44; w3[12][6]=-0.92; w3[12][7]=1.36; w3[12][8]=-1.05;
w3[12][9]=-0.53; w3[12][10]=0.99; w3[12][11]=0.0; w3[12][12]=0.0;

//CAPA DE CELULAS 4, PESOS ACTUALIZADOS

w4[1][1]=3.79; w4[1][2]=2.1; w4[1][3]=-6.86; w4[1][4]=-3.6;
w4[1][5]=4.95; w4[1][6]=5.54; w4[1][7]=2.09; w4[1][8]=-6.76;
w4[1][9]=3.97; w4[1][10]=1.09; w4[1][11]=0.0; w4[1][12]=0.00;

}}



```

void MatrizA(double A[DIM][DIM])
{
    int i, j, k;
    for(i=0;i<=numva;i++)
    for(j=0;j<=numva;j++)
    A[i][j]=0.0;
    A[0][1]= 2*M_PI*fo;
    A[1][1]= -Kd/M;
    A[1][8]= 1/M;
    A[2][2]= -1/tpdo;
    A[2][5]= 1/tpdo;
    A[3][2]= 1/tppdo;
    A[3][3]= -1/tppdo;
    A[4][4]= -1/tppqo;
    A[5][5]= -1/Tx;
    A[6][6]= -1/Tsm;
    A[7][6]= 1/Tch;
    A[7][7]= -1/Tch;
    A[8][6]= Fhp/(Trh*Tch);
    A[8][7]= (1-Fhp/Tch)/Trh;
    A[8][8]= -1/Trh;
    if ( ctrol == 'S' )
    {
        printf("\n Máquina con controles 1 :");printf("%c", ctrol);
        A[9][5] = -Kf/(Tf*Tx);
        A[9][9]= -1/Tf;
        A[9][10] = Kf/(Tf*Tx);
        A[10][9] = -Ka/Ta;
        A[10][10] = -1/Ta;
    }
}

```

```

void VectorB(double B[DIM])
{
    int i;
    for(i=0;i<=numva;i++)
    B[i]=0.0;
    B[1]= -Te/M;
    B[2]= -(xd-xpd)*id/tpdo;
    B[3]= -(xpd-xppd)*id/tppdo;
    B[4]= (xq-xppq)*iq/tppqo;
    B[5]= U/Tx;
    B[6]= Ug/Tsm;
    if ( ctrol == 'S' )
    { B[10] = (Ka/Ta)*(Vref-Vt);}
}

```

```

void VectorVX(double VX[DIM])
{
    VX[0] = deltar;
    VX[1] = s;
    VX[2] = epq;
    VX[3] = eppq;
    VX[4] = eppd;
    VX[5] = Vf;
    VX[6] = Pgv;
}

```



```

    VX[7] = Php;
    VX[8] = Tm;
    if ( ctrol == 'S' )
    {
        VX[9] = Verror;
        VX[10] = U;
    }
}

void idiqTe(double VX[DIM])
{
    double r_t, e_q, e_d, x_d, x_q, det;
    r_t=e_q=e_d=x_d=x_q=det=0.0;
    r_t=(re+ra);
    x_d=(xe+xppd);
    x_q=(xe+xppq);
    det=r_t*r_t+x_d*x_q;
    e_d=VX[4]-Vb*sin(VX[0]);
    e_q=VX[3]-Vb*cos(VX[0]);
    id=(r_t*e_d+x_q*e_q)/det;
    iq=(r_t*e_q-x_d*e_d)/det;
    Te=VX[4]*id+VX[3]*iq;
}

void VTR(double VX[DIM])
{
    double vd,vq;
    vd= Vb*sin(VX[0])+re*id-xe*iq;
    vq= Vb*cos(VX[0])+re*iq+xe*id;
    Vt=sqrt(vd*vd+vq*vq);
}

void PotC( )
{
    double ceai,rr,ri,rr2,ri2,rr3,ri3;
    ear=Vt*cos(theta);
    eai=Vt*sin(theta);
    ceai=ear;
    ceai=-eai; // conjugado
    rr=ear-Vb;
    ri=eai-0; // resta
    rr2= (rr*re+ri*xe)/(re*re+xe*xe); // división
    ri2= (ri*re-rr*xe)/(re*re+xe*xe);
    rr3=ceai*rr2-ceai*ri2; // multiplicación
    ri3=ceai*ri2+ceai*rr2;
    Pr=rr3;
    Pi=-ri3; // potencias
}

void Rungekutta(int numva,double h,double A[DIM][DIM], double B[DIM], double VX[DIM],int u2)
{
    FILE *fi;
    if (control == 's')
    { printf("\n Máquina con controles :");}
    else { printf("\n Máquina sin controles :");}
    printf("\n Nombre del archivo de salida :");
}

```



```

scanf("%s", nombre);
fi=fopen(nombre,"w");
double cont,suma,z;
int i,j,k;
fprintf(fi,"%f %f %f %f %f %f %f %f %f ",cont,((VX[0]*180)/M_PI),VX[5],Vt,Pr,Pi,Ug,
VX[1]);
fprintf(fi,"\n");
for(cont=cont+h;cont<=tf;cont+=h)
{
XP[0]= XP[1]=XP[2]=XP[3]=u2;
for(k=0;k<=numva;k++)
{
DX[k][0]=0;
for(i=0;i<numva;i++)
DX[k][0]=DX[k][0]+A[k][i]*VX[i];
}
for(i=0;i<numva;i++)
DX[i][0]=DX[i][0]+B[i]*XP[0];
z=0.5*h;
for(i=1;i<4;i++)
{
if( i == 3 ) z=h;
for(j=0;j<numva;j++) KX[j]=VX[j]+z*DX[j][i-1];
for(k=0;k<=numva;k++)
{
DX[k][i]=0;
for(j=0;j<numva;j++)
DX[k][i]=DX[k][i]+A[k][j]*KX[j];
}
for(j=0;j<numva;j++)
DX[j][i]=DX[j][i]+B[j]*XP[i];
}
for(k=0;k<=numva;k++)
VX[k]=VX[k]+h*(DX[k][0]+2*DX[k][1]+2*DX[k][2]+DX[k][3])/6;
idiqTe(VX);
if ( ccto == 'S' )
{
if ( cont >= 1.0 && cont <= 1.150)
{ Vb = 0.0;}
else { Vb = 1.0;
}
if ( cont >= 11.0 && cont <= 11.150)
{ Vb = 0.0;}
else
{ Vb = 1.0;}
}
if (cont>=21.0)
{
if ( cont >= 21.0 && cont <= 21.150)
{ Vb = 0.0;}
else
{ Vb = 1.0; }
}
}
}

VTR(VX);

```




}

```
void potencia(double VX[DIM])
```

{

```
double rr3,ri3,vd,vq;
vd = Vb*sin( VX[0] ) + re *id - xe *iq;
vq = Vb*cos( VX[0] ) + re *iq + xe *id;
rr3=vd*id-vq*(-iq);
ri3=vd*(-iq)+vq*id;
Pr = rr3; Pi = ri3;
```

}

```
double neo(double *W1, double *W2, double *W3, double *W4, *Error, double *Verror, double z);
```

{

```
k=0;
```

```
for(k=0;1000;k++)
```

{

```
n1=[[w1*p(:,1),w1*p(:,2),w1*p(:,3),w1*p(:,4),w1*p(:,5),w1*p(:,6),w1*p(:,7),w1*p(:,8),
w1*p(:,9),w1*p(:,10),w1*p(:,11),w1*p(:,12)]+[w2*p(:,1),w2*p(:,2), w2*p(:,3),
w2*p(:,4),w2*p(:,5),w2*p(:,6),w2*p(:,7),w2*p(:,8),w2*p(:,9),w2*p(:,10),w2*p(:,11),
w1*p(:,12)]+[w3*p(:,1),w3*p(:,2), w3*p(:,3), w3*p(:,4),w3*p(:,5),w3*p(:,6),w3*p(:,7),
w3*p(:,8),w3*p(:,9),w3*p(:,10),w3*p(:,11),w3*p(:,12)]] ;
```

```
A1=tansig(n1)
```

```
n2=w4*A1;
```

```
z=purelin(w4*A1);
```

}

}