



# INSTITUTO POLITECNICO NACIONAL

---

---

ESCUELA SUPERIOR DE INGENIERIA  
MECANICA Y ELECTRICA

SECCION DE ESTUDIOS DE POSGRADO E INVESTIGACION  
DEPARTAMENTO DE INGENIERIA ELECTRICA

**CONTROL DIRECTO DEL PAR Y FLUJO DEL MOTOR DE  
INDUCCION UTILIZANDO UNA RED NEURO-DIFUSA**

**TESIS PROFESIONAL**

QUE PARA OBTENER EL GRADO DE MAESTRO  
EN CIENCIAS CON ESPECIALIDAD EN  
INGENIERIA ELECTRICA  
P R E S E N T A :  
MIGUEL ANGEL GAMA VALDEZ

MEXICO D. F.

2004





CGPI-14

**INSTITUTO POLITECNICO NACIONAL**  
**COORDINACION GENERAL DE POSGRADO E INVESTIGACION**

**ACTA DE REVISION DE TESIS**

En la Ciudad de México, D. F. siendo las 12:00 horas del día 09 del mes de Marzo del 2004 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de la E. S. I. M. E. para examinar la tesis de grado titulada:

**“CONTROL DIRECTO DEL PAR Y FLUJO DEL MOTOR DE INDUCCION UTILIZANDO UNA RED NEURO-DIFUSA”**

Presentada por el alumno:

**GAMA**

Apellido paterno

**VALDEZ**

materno

**MIGUEL ANGEL**

nombre(s)

Con registro: 

0	2	0	0	5	9
---	---	---	---	---	---

Aspirante al grado de:

**MAESTRO EN CIENCIAS**

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

LA COMISION REVISORA

Director de tesis

*David Romero Romero*  
 DR. DAVID ROMERO ROMERO

*Dr. Leszek Kawecki Zlotkowska*  
 DR. LESZEK KAWECKI ZLOTKOWSKA

*Jaime Rodríguez Rivas*  
 DR. JAIME RODRIGUEZ RIVAS

*Dr. Raul Cortes Mateos*  
 DR. RAUL CORTES MATEOS

*Daniel Olguin Salinas*  
 DR. DANIEL OLGUIN SALINAS

*Ignacio Asiain Olivares*  
 DR. IGNACIO ASIAIN OLIVARES

EL PRESIDENTE DEL COLEGIO  
 SECCION DE ESTUDIOS DE  
 POSGRADO E INVESTIGACION

*Florencio Sánchez Silva*  
 DR. FLORENCIO SANCHEZ SILVA



**INSTITUTO POLITECNICO NACIONAL**  
**COORDINACION GENERAL DE POSGRADO E INVESTIGACION**

**CARTA CESION DE DERECHOS**

En la Ciudad de México, Distrito Federal, el día 09 del mes Marzo del año 2004, el (la) que suscribe Miguel Angel Gama Valdéz alumno(a) del Programa de Maestría en Ciencias con especialidad en Ingeniería Eléctrica con número de registro 020059, adscrito a la Sección de Estudios de Posgrado e Investigación de la ESIME Unidad Zacatenco, manifiesta que es autor(a) intelectual del presente Trabajo de Tesis bajo la dirección del Dr. David Romero Romero y cede los derechos del trabajo intitulado: **CONTROL DIRECTO DEL PAR Y FLUJO DE MOTOR DE INDUCCION UTILIZANDO UNA RED NEURO-DIFUSA**, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, graficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección: Matagalpa 1112, Col. Residencial Zacatenco. C. P. 07369. México D. F.

Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

**Miguel Angel Gama Valdéz**

---

Nombre y firma

## Resumen

En los últimos años los accionamientos eléctricos para motores de inducción se han convertido en un campo de investigación ampliamente señalado como uno de los más importantes en el área de la electrónica de potencia, sistemas de control automático y máquinas eléctricas. Esto se debe principalmente a la importancia del motor de inducción en un sinnúmero de procesos industriales. Aunque actualmente se utilizan variadas metodologías de control, existe todavía una gran necesidad de mejorar su desempeño dinámico, principalmente porque dichas metodologías contienen elementos con altas no linealidades así como también debido a que carecen de técnicas que permitan el control adecuado de la frecuencia de conmutación en los inversores, la selección óptima de vectores de voltaje y la disminución en la distorsión armónica contenida en las señales de par y corriente eléctrica, entre otros problemas.

En este trabajo se presenta el análisis y la aplicación de un esquema de control para el motor de inducción utilizando un controlador cuyos fundamentos teóricos se basan en las metodologías y conceptos de la Inteligencia Artificial. Se propone una estructura que combina los Sistemas de Razonamiento Difuso y las Redes Neuronales Artificiales para construir un controlador Neuro-Difuso con propiedades de adaptación.

Este controlador se incorpora dentro del esquema del Control Directo del Par desarrollado por Depenbrock y Takahashi en la década de los 80's, el cual como su nombre lo indica permite que el par y flujo magnético del estator sean controlados de manera desacoplada y directamente a través del voltaje aplicado en las terminales del motor. En este caso, la función del controlador Neuro-Difuso es la de producir un vector de voltaje de referencia, mismo que mediante la técnica de Modulación de Vectores Espaciales proporciona las señales de conmutación necesarias para los diferentes estados del inversor y de esta forma obtener una rápida respuesta en el control del par y flujo magnético.

La inclusión de un controlador inteligente permite la eliminación de los controladores de histéresis así como también de las tablas de conmutación utilizadas en el esquema del Control Directo del Par original. Bajo este esquema se logra fijar la frecuencia de conmutación a un valor constante y apropiado para los dispositivos de potencia del inversor y en general un mejor desempeño en estado estacionario y transitorio.

Finalmente, el desempeño del Control Neuro-Difuso del Par y Flujo se compara con el obtenido del esquema clásico bajo las mismas condiciones de operación. Como se verá, en este trabajo se demuestra que la aplicación de un controlador inteligente no lineal no solo logra un mejor control del par y del flujo, sino también permite la interacción humano-máquina de una manera natural y simplificada.

## **Abstract**

Because of the importance of the induction motors in many industrial applications, electric drives have in the last years become an important field to research in areas like power electronics, automatic control systems and electric machinery. Although there are some control methodologies that are being used successfully in several applications, it is well known that an improvement in this kind of system is still needed. Complicated no-linear mathematical models, operation difficulties due to high harmonic distortions in current, stator flux and torque signals, variable switching frequency, among others, are some special features that yield serious problems when the system is operated.

In this work an examination as well as the application of a control scheme for induction motors using a controller-based Artificial Intelligence concepts and methodologies is presented. This controller is indeed a combination of two well-known work frames such as Fuzzy Inference Systems and Artificial Neural Networks, which are used to produce an adaptable Neuro-Fuzzy controller.

This controller is incorporated in the Direct Torque Control scheme, proposed by Depenbrock and Takahashi in the 80's. This control strategy allows the direct control of stator flux and electromagnetic torque in an uncoupled way throughout the stator voltage applied by the Voltage Source Inverter to the machine. The main function of this Neuro-Fuzzy controller is to generate a reference voltage vector that controls the inverter modes in order to achieve as fast as possible hi-performance in torque and stator flux changes of a squirrel cage induction motor.

Besides this intelligent controller, Space Vector Modulation technique is applied to avoid hysteresis controllers and switching tables which are common in conventional DTC schemes. Under this frame it is possible to set a constant value for the switching frequency and, in general, improve the dynamic behavior of the system.

Finally, the behavior of the Direct Torque and Flux Neuro-Fuzzy Control is compared with those gotten by classical DTC scheme presented in past researchers. This work has shown that the proposed Neuro-Fuzzy control scheme is characterized not only by simple mathematical models, but also by a very fast torque and flux responses. This control provides a natural and well-defined human-machine interaction.

# Índice de Contenido

<b>Resumen</b> .....	<b>4</b>
<b>Abstract</b> .....	<b>5</b>
<b>Índice de Figuras</b> .....	<b>9</b>
<b>Índice de Tablas</b> .....	<b>12</b>
<b>Glosario de Términos</b> .....	<b>13</b>
<b>Agradecimientos</b> .....	<b>15</b>
<b>Contacto</b> .....	<b>15</b>
<b>1. Introducción</b> .....	<b>16</b>
<b>1.1. Generalidades</b> .....	<b>16</b>
1.1.1. Sistemas Difusos .....	17
1.1.2. Redes Neuronales.....	17
1.1.3. Sistemas Neuro-Difusos Integrados.....	18
<b>1.2. Objetivos de la Tesis</b> .....	<b>19</b>
<b>1.3. Descripción del Problema</b> .....	<b>19</b>
<b>1.4. Justificación</b> .....	<b>20</b>
<b>1.5. Estado del Arte</b> .....	<b>21</b>
<b>1.6. Estructura de la Tesis</b> .....	<b>22</b>
<b>1.7. Aportaciones</b> .....	<b>23</b>
<b>2. Modelo Neuro-Difuso: Sistemas de Razonamiento Difuso + Redes Neuronales</b> .....	<b>25</b>
<b>2.1. Introducción</b> .....	<b>25</b>
<b>2.2. Sistemas de Razonamiento Difuso</b> .....	<b>26</b>
2.2.1. Estructura de un Sistema de Razonamiento Difuso.....	27
2.2.2. Sistema tipo Takagi-Sugeno.....	28
<b>2.3. Redes Neuronales Artificiales</b> .....	<b>31</b>
2.3.1. Aprendizaje Supervisado de la Red Neuronal.....	31
2.3.2. Redes Adaptables.....	33
<b>2.4. Redes Neuronales basadas en Sistemas Difusos</b> .....	<b>35</b>
2.4.1. Sistema de Razonamiento Neuro-Difuso Adaptable.....	36
2.4.2. Estructura de la Red Neuro-Difusa.....	37
<b>2.5. Identificación y Estimación</b> .....	<b>40</b>
2.5.1. Algoritmo de Aprendizaje.....	43
2.5.1.1. Método de Estimación por Cuadrados Mínimos.....	44
2.5.1.2. Filtro de Kalman.....	46
2.5.2. Identificación de un Sistema No Lineal.....	49

<b>3. Control Neuro-Difuso .....</b>	<b>52</b>
3.1. Introducción.....	52
3.2. Sistemas de Control Retroalimentados. ....	53
3.3. Sistemas de Control Basados en Lógica Difusa. ....	54
3.4. Controlador Neuro-Difuso. ....	57
3.4.1. Aprendizaje Inverso.....	59
3.4.2. Procedimiento para el Control Neuro-Difuso.....	62
<b>4. Control Neuro-Difuso Directo de Par y Flujo Magnético .....</b>	<b>66</b>
4.1. Introducción.....	66
4.2. Esquema Clásico del Control Directo del Par. ....	67
4.3. Control Directo del Par con Modulación de Vectores Espaciales. ....	71
4.4. Control Neuro-Difuso Directo del Par.....	74
4.4.1. Arquitectura del Controlador Neuro-Difuso.....	75
4.4.2. Esquema de Entrenamiento de la Red. ....	78
4.5. Pruebas bajo distintas Condiciones de Operación.....	79
4.5.1. Desempeño en Estado Estacionario.....	80
4.5.2. Control del Flujo Magnético del Estator. ....	84
4.5.3. Control del Par Electromagnético.....	89
4.6. Comparación con Esquemas Convencionales.....	94
<b>5. Conclusiones .....</b>	<b>103</b>
Recomendaciones y Direcciones Futuras. ....	106
<b>Referencias .....</b>	<b>108</b>

## APENDICES

<b>A. Modelo Matemático del Motor de Inducción .....</b>	<b>113</b>
A.1. Modelo del Motor en Variables Reales. ....	115
A.2. Modelo de la Máquina en el Marco de Referencia Arbitrario.....	118
A.3. Ecuación del Par Electromagnético.....	123
<b>B. Control Directo del Par.....</b>	<b>128</b>
B.1. Introducción General. ....	128
B.2. Esquema del Control Directo del Par. ....	128
B.3. Principios Teóricos del Control Directo del Par.....	129

---

<b>C. Inversor de Potencia y Modulación de Vectores Espaciales</b>	<b>133</b>
C.1. Inversor Fuente de Voltaje (VSI).....	133
C.2. Modulación de Vectores Espaciales (SVM).....	135
<b>D. Parámetros del Motor de Inducción.....</b>	<b>139</b>
<b>E. Programas de Simulación Digital.....</b>	<b>141</b>
E.1. Diagrama de Flujo para la Simulación del DTNFC en su Fase de Entrenamiento.	142
E.2. Diagrama de Flujo para la Simulación del DTNFC en su Fase de Aplicación. ....	145
E.3. Motor de Inducción.....	147
E.4. Modulación de Vectores Espaciales.....	150
E.5. Red Neuro-Difusa.....	154
E.6. DTNFC - Entrenamiento.....	160
E.7. DTNFC - Aplicación.....	170



## Indice de Figuras

<b>Figura 2. 1.</b> Esquema básico de un FIS con múltiples entradas y una salida. ....	27
<b>Figura 2. 2.</b> Modelo Difuso Takagi-Sugeno con dos entradas, una salida. ....	30
<b>Figura 2. 3.</b> Esquema del aprendizaje supervisado de una red neuronal. ....	32
<b>Figura 2. 4.</b> Diagrama de ADALINE (adaptive linear element). ....	32
<b>Figura 2. 5.</b> Estructura de una red adaptable. ....	34
<b>Figura 2. 6.</b> Red neuronal equivalente a tres funciones de membresía de un sistema difuso. ....	36
<b>Figura 2. 7.</b> (a) Sistema de razonamiento difuso tipo TSK, dos entradas, una salida. (b) Red neuronal adaptable equivalente. ....	38
<b>Figura 2. 8.</b> Significado de los parámetros de la función de membresía campana. ....	39
<b>Figura 2. 9.</b> Procedimientos básicos para la identificación de un sistema. ....	41
<b>Figura 2. 10.</b> Fase de experimentación en el proceso de identificación del sistema. ....	42
<b>Figura 2. 11.</b> Modelo neuro-difuso adaptable con 2 entradas, 1 salida, 2 funciones de membresía por entrada, 4 reglas difusas, 12 parámetros antecedentes y 12 consecuentes. ....	50
<b>Figura 2. 12.</b> Salida del sistema no lineal (línea discontinua) y de la red adaptable (línea continua). Identificación con 250 datos de entrenamiento. Eje x: 40 muestras por división. ....	50
<b>Figura 2. 13.</b> Salida del sistema no lineal (línea discontinua) y de la red adaptable (línea continua). Identificación con 1500 datos de entrenamiento. Eje x: 250 muestras por división. ....	51
<b>Figura 3. 1.</b> Diagrama de bloques de un sistema de control retroalimentado. ....	53
<b>Figura 3. 2.</b> Sistema de control difuso. ....	54
<b>Figura 3. 3.</b> FLC para un sistema dinámico. ....	56
<b>Figura 3. 4.</b> Controladores difusos por analogía con los convencionales. (a) PD; (b) PI; (c) PID. ....	57
<b>Figura 3. 5.</b> Diagramas de bloque para el aprendizaje inverso. (a) planta; (b) fase de entrenamiento; (c) fase de aplicación. ....	60
<b>Figura 3. 6.</b> Esquema del control neuro-difuso con aprendizaje inverso en línea. ....	61
<b>Figura 3. 7.</b> Fase experimental del control neuro-difuso. ....	63
<b>Figura 3. 8.</b> Fase de validación o aplicación del modelo inverso identificado. Se muestra el desempeño del NFC en el control del sistema. ....	64
<b>Figura 4. 1.</b> Esquema clásico del Control Directo de Par (DTC). ....	68
<b>Figura 4. 2.</b> Vectores espaciales de voltaje del inversor. ....	70
<b>Figura 4. 3.</b> Control directo de par con modulación de vectores espaciales y controlador predictivo (DTC-SVM). ....	72
<b>Figura 4. 4.</b> Principio de la modulación de vectores espaciales. ....	73
<b>Figura 4. 5.</b> Esquema del control neuro-difuso directo del par (DTNFC). ....	74

<b>Figura 4. 6.</b> Estructura del controlador Neuro-Difuso propuesto.....	75
<b>Figura 4. 7.</b> Cálculo del ángulo del vector espacial de voltaje.....	76
<b>Figura 4. 8.</b> Funciones de membresía utilizadas en la capa 1 del NFC.....	77
<b>Figura 4. 9.</b> Esquema de entrenamiento para el DTNFC. ....	78
<b>Figura 4. 10.</b> Desempeño en estado estacionario. $T_e$ , Par electromagnético; $P_s$ , Magnitud del vector de flujo magnético del estator; $P_{as}$ , Flujo magnético de la fase a del estator; $P_{ar}$ , Flujo magnético de la fase a del rotor.....	81
<b>Figura 4. 11.</b> Corriente eléctrica en estado estacionario. $i_{as}$ , Corriente fase a; $i_{bs}$ , Corriente fase b; $i_{cs}$ , Corriente fase c.....	82
<b>Figura 4. 12.</b> Error del par y flujo en estado estacionario. $e_P$ , Error del flujo magnético del estator; $e_T$ , error del par electromagnético.....	83
<b>Figura 4. 13.</b> Trayectoria del vector de voltaje.....	83
<b>Figura 4. 14.</b> Trayectoria del vector del flujo del estator. ....	84
<b>Figura 4. 15.</b> Desempeño del DTNFC ante cambios en la señal de referencia del flujo magnético. $P_{ref}$ , Señal de referencia del flujo del estator; $P_s$ , Magnitud del vector de flujo magnético del estator; $P_{as}$ , Flujo magnético en la fase a del estator; $P_{ar}$ , Flujo magnético en la fase a del rotor.....	85
<b>Figura 4. 16.</b> Corriente eléctrica ante cambios en la señal del flujo del estator. $i_{as}$ , Corriente fase a; $i_{bs}$ , Corriente fase b; $i_{cs}$ , Corriente fase c.....	86
<b>Figura 4. 17.</b> Trayectoria del vector de voltaje.....	87
<b>Figura 4. 18.</b> Trayectoria del vector del flujo del estator. ....	88
<b>Figura 4. 19.</b> Trayectoria del vector del flujo del rotor. ....	88
<b>Figura 4. 20.</b> Desempeño del DTNFC ante cambios en el par de referencia. $T_{ref}$ , Señal de referencia del par; $T_s$ , Par electromagnético desarrollado; $P_s$ , Flujo magnético del estator. ....	89
<b>Figura 4. 21.</b> Corriente eléctrica ante cambios en la señal del par electromagnético. $i_{as}$ , Corriente fase a; $i_{bs}$ , Corriente fase b; $i_{cs}$ , Corriente fase c.....	90
<b>Figura 4. 22.</b> Trayectoria del vector de voltaje.....	91
<b>Figura 4. 23.</b> Trayectoria del vector del flujo del estator. ....	91
<b>Figura 4. 24.</b> Trayectoria del vector del flujo del rotor. ....	92
<b>Figura 4. 25.</b> Desempeño del DTNFC ante cambios en la señal de referencia d el par. $T_{ref}$ , Par de referencia; $T_e$ , Par electromagnético producido. ....	93
<b>Figura 4. 26.</b> Corriente eléctrica ante cambios en la señal del par electromagnético. $i_{as}$ , Corriente fase a; $i_{bs}$ , Corriente fase b; $i_{cs}$ , Corriente fase c.....	93
<b>Figura 4. 27.</b> Par electromagnético. (a) DTC $\Delta T=0.2$ , $\Delta\lambda=0.1$ ;(b) DTC $\Delta T=0.5$ , $\Delta\lambda=0.01$ ; (c) DTNFC.....	95
<b>Figura 4. 28.</b> Amplitud del vector del flujo del estator. (a) DTC $\Delta T=0.2$ , $\Delta\lambda=0.1$ ;(b) DTC $\Delta T=0.5$ , $\Delta\lambda=0.01$ ; (c) DTNFC. ....	97
<b>Figura 4. 29.</b> Trayectoria del vector de flujo del estator. (a) DTC $\Delta T=0.2$ , $\Delta\lambda=0.1$ ;(b) DTC $\Delta T=0.5$ , $\Delta\lambda=0.01$ ; (c) DTNFC. ....	98
<b>Figura 4. 30.</b> Corriente eléctrica del estator. (a) DTC $\Delta T=0.2$ , $\Delta\lambda=0.1$ ;(b) DTC $\Delta T=0.5$ , $\Delta\lambda=0.01$ ; (c) DTNFC. ....	99
<b>Figura 4. 31.</b> Respuesta del par electromagnético ante un escalón. (a): DTC-SVM; (b): DTNFC. Eje x: 2.65ms por división. ....	100

<b>Figura 4. 32.</b> Corriente del estator ante un escalón en el par. (a): DTC-SVM; (b): DTNFC. Eje x: 2.65ms por división. ....	101
<b>Figura 4. 33.</b> Trayectoria del vector de flujo del estator. (a) DTC-SVM; (b) DTNFC. ....	101
<b>Figura A. 1.</b> Motor de inducción trifásico conectado en estrella, 2 polos 3 fases. ....	114
<b>Figura A. 2.</b> Transformación de las variables de circuitos estacionarios a los vectores espaciales dq. ....	118
<b>Figura A. 3.</b> Libre aceleración del motor de inducción en el marco de referencia estacionario. ....	125
<b>Figura A. 4.</b> Libre aceleración del motor de inducción en el marco de referencia rotatorio. .....	126
<b>Figura A. 5.</b> Libre aceleración del motor de inducción en el marco de referencia síncrono. .....	126
<b>Figura B. 1.</b> Estructura básica del DTC (Figura tomada de www.abb.com).....	129
<b>Figura B. 2.</b> Diagrama equivalente del motor de inducción en el marco de referencia estacionario. (a) Eje $q$ ; (b) Eje $d$ . ....	130
<b>Figura B. 3.</b> Impacto del vector de voltaje sobre el flujo del estator. ....	131
<b>Figura B. 4.</b> Principio del DTC mediante la selección de los estados del VSI. ....	132
<b>Figura C. 1.</b> Inversor Trifásico Fuente de Voltaje (VSI). ....	133
<b>Figura C. 2.</b> Esquema de los seis estados activos del PWM VSI y su representación con vectores espaciales (a) Estados del VSI; (b) Representación con vectores espaciales. .....	134
<b>Figura C. 3.</b> Principio de la técnica del SVM. ....	136
<b>Figura C. 4.</b> Señales de voltaje proporcionadas por un VSI con PWM-SVM. $v_a$ , Voltaje de línea; $v_{an}$ , Voltaje de fase. ....	137
<b>Figura C. 5.</b> Modulación del vector de referencia ante un cambio en su magnitud. $v_a$ , Voltaje de línea; $v_{an}$ , Voltaje de fase. ....	137
<b>Figura C. 6.</b> Señales de voltaje a una frecuencia de conmutación de 20kHz. $v_a$ , Voltaje de línea; $v_{an}$ , Voltaje de fase. ....	138

# Indice de Tablas

<b>Tabla 1. 1</b> Comparación de los Sistemas Difusos (FS), Redes Neuronales (NN) y Teorías Convencionales (TC).....	18
<b>Tabla 2. 1.</b> Procedimientos de aprendizaje y adaptación del ANFIS .....	49
<b>Tabla 4. 1.</b> Incremento del ángulo del vector parcial del vector de voltaje de referencia. ...	77
<b>Tabla 4. 2.</b> Reglas difusas del controlador Neuro-Difuso. ....	80
<b>Tabla A. 1.</b> Interpretación de $\omega$ en los distintos marcos de referencia .....	123

## Glosario de Términos

$A$	Conjunto difuso
$\alpha$	Angulo del vector espacial de voltaje dentro del sector
$\alpha^*$	Angulo del vector de referencia de voltaje
ANFIS	Red Adaptable basada en un Sistema de Inferencia Difuso
$d$	Salida deseada de la red
$\Delta\alpha$	Incremento del ángulo de referencia de voltaje
$\Delta e(\cdot)$	Cambio del error
$\Delta\lambda$	Ancho de banda del controlador de histéresis del flujo
$\delta e(\cdot)$	Suma de los errores
$\Delta T$	Ancho de banda del controlador de histéresis del par
DTC	Control Directo del Par
DTNFC	Control Neuro-Difuso Directo del Par
$\Delta u(\cdot)$	Cambio de la acción de control
$e$	error
$\varepsilon_\lambda$	Error del flujo magnético
$\varepsilon_T$	Error del Par electromagnético
$f(\cdot)$	Función de activación de un nodo
$\Phi(\cdot)$	Función del controlador
FIS	Sistema de Razonamiento Difuso
FLC	Controlador Difuso
FMM	Fuerza magnetomotriz
FOC	Control por Campo Orientado
$f_s$	frecuencia de conmutación
$\gamma$	Angulo del par
$G(\cdot)$	Función inversa
$H$	Inercia del rotor
$i_{dr}$	Corriente del rotor en el eje $d$
$i_{ds}$	Corriente del estator en el eje $d$
$i_{qr}$	Corriente del rotor en el eje $q$
$i_{qs}$	Corriente del estator en el eje $q$
K	Transformación de Park
$K_d$	Ganancia diferencial
$K_i$	Ganancia integral
$K_p$	Ganancia proporcional
$\lambda_{dr}$	Flujo magnético del rotor en el eje $d$
$\lambda_{ds}$	Flujo magnético del estator en el eje $d$
$\lambda_{qr}$	Flujo magnético del rotor en el eje $q$
$\lambda_{qs}$	Flujo magnético del estator en el eje $q$
$\lambda_r$	Vector de flujo del rotor
$\lambda_s$	Vector de flujo del estator
$\mu_A(x)$	Grado de membresía del elemento de conjunto $A$
NFC	Controlador Neuro-Difuso
NN	Red Neuronal Artificial
$O_i$	Salida del nodo $i$

$P$	Matriz de covarianza
$p$	d/dt
$p, q, r$	Parámetros consecuentes
PD	Proporcional derivativo
$P_{ds}$	Flujo magnético del estator en el eje $d$
PI	Proporcional integral
PID	Proporcional derivativo integral
$P_{qs}$	Flujo magnético del estator en el eje $q$
$P_s$	Vector del flujo magnético del estator
PWM	Modulación del Ancho de Pulso
$\theta$	Matriz de parámetros
$\theta_{\lambda s}$	Angulo del vector de flujo del estator
$\theta_{sr}$	Angulo del par
$\theta_v$	Angulo del vector espacial de voltaje
$R$	Relación difusa
$R_i$	Regla difusa
$r_r$	Resistencia del rotor
$r_s$	Resistencia del estator
SVM	Modulación de Vectores Espaciales
$t$	tiempo
$t_a, t_b, t_0$	Tiempo de aplicación del vector de voltaje
$T_e$	Par electromagnético
$T_m$	Par mecánico
$T_s$	Periodo de conmutación
TSK	Sistema difuso Takagi Sugeno
$U$	Universo del discurso
$u$	Acción de control
$V^*$	Vector de referencia de voltaje
$v_{as}$	Voltaje de línea fase A
$v_{bs}$	Voltaje de línea fase B
$V_{cd}$	Voltaje de c.d.
$v_{cs}$	Voltaje de línea fase C
$v_{dr}$	Voltaje del rotor en el eje $d$
$v_{ds}$	Voltaje del estator en el eje $d$
$v_k$	Vector espacial de voltaje
$v_{qr}$	Voltaje del rotor en el eje $q$
$v_{qs}$	Voltaje del estator en el eje $q$
VSI	Inversor Fuente de Voltaje
$\omega$	Velocidad angular
$\omega_b$	Velocidad angular base
$w_i$	Peso de la conexión $i$ entre dos nodos
$\omega_r$	Velocidad del rotor
$x$	Entrada de la red
$X_{lr}$	Reactancia del rotor
$X_{ls}$	Reactancia del estator
$X_M$	Reactancia mutua
$y$	Salida de la red

## Agradecimientos

En primera instancia quiero agradecer el apoyo del Profesor *Dr. David Romero Romero* quien con sus valiosas sugerencias y consejos fue posible la realización de este trabajo. Asimismo quiero expresar mi gratitud al *Dr. Jaime Robles García* y al *Dr. Daniel Olguín Salinas* quienes forman parte importante de mi formación académica y de quienes siento profunda admiración y respeto. Pero por encima de todo agradezco el cariño y apoyo incondicional de mi familia *Rosa María, Julián, Connie, Lizet, Dany, Nohemí*, y mis sobrinos *Naomi* y *Juliancito*, pero especialmente a mi padre *Tarcisio* quien ha sido el fundamento de mi éxito y quien con su paciencia me ha enseñado que el mayor triunfo es levantarse de la peor derrota.

Este trabajo fue realizado en parte gracias al apoyo económico del Consejo Nacional de Ciencia y Tecnología (CONACyT).

## Contacto

El intercambio de conocimiento, opiniones y crítica constructiva siempre es de gran valor y se agradece. Para correcciones, opiniones y aclaraciones, sírvase contactarme al siguiente correo electrónico:

**miguel@gamaware.com**  
**mgamav@netscape.net**

# Capítulo 1

## Introducción

### 1.1. Generalidades.

La *inteligencia* se puede definir como la facultad de aprender, de comprender, de conocer; es la aptitud para relacionar las percepciones sensoriales o para abstraer y asociar conceptos [12]. El ser humano es por excelencia la criatura más inteligente sobre la tierra ya que posee características inherentes tales como el aprendizaje y la adaptación constante, que le han permitido desarrollar cualesquier actividad deseada hasta en los medios y circunstancias menos favorables.

Sin embargo, el concepto de la inteligencia junto con todo lo que lo envuelve ha tomado distintos matices a través de los años. Numerosos avances en la ciencia han dado como resultado nuevos campos de investigación que siguen como modelo el comportamiento natural del ser humano. Uno de esos campos es la llamada *Inteligencia Artificial*, la cual utiliza distintas técnicas que imitan los procesos de aprendizaje, razonamiento y toma de decisiones producidas en el cerebro pero aplicadas y dirigidas a objetos o sistemas y de esta forma proveerles inteligencia. Aunque este es un campo relativamente nuevo, desde sus inicios con las aportaciones de científicos como Lofti Zadeh en 1965 y J. J. Hopfield en 1982 entre otros, las técnicas de inteligencia artificial han sido objeto de gran interés y en la actualidad los sistemas o dispositivos inteligentes están en muchos casos sustituyendo a los convencionales [38]. Ejemplos de esto van desde aparatos electrodomésticos como lavadoras, refrigeradores, hasta equipo fotográfico, cámaras de vídeo, etc. Lo mismo ocurre en la industria, donde los sistemas de control han sido sustituidos por sistemas inteligentes, tales como controladores de procesos químicos de destilación, sistemas de control para aire acondicionado, reguladores de control de combustible, sistemas de transmisión en automóviles y trenes, sistemas de control en aeronaves entre otras aplicaciones [48].

Existen tres principales metodologías en la inteligencia artificial: la *lógica difusa*, las *redes neuronales* y los *algoritmos genéticos*. Su utilidad ha sido completamente apreciada no sólo porque constituyen una nueva dirección en la investigación, sino también



porque proveen poderosas herramientas para crear eficientes y sencillos sistemas inteligentes capaces de resolver complejos y antiguos problemas que las técnicas convencionales no podían. La capacidad de cada metodología como una herramienta de diseño sólo está limitada por la imaginación de quien las utiliza tanto una por separado o en su conjunto ya que como se mencionó son biológicamente inspiradas.

En lo que respecta al control inteligente de motores de inducción, en años recientes se han realizado investigaciones para diseñar controladores basados en los principios y metodologías de la inteligencia artificial. Debido a la importancia del motor de inducción, el control inteligente ha sido propuesto y en ocasiones implementado para controlar accionamientos eléctricos con inversores fuente de voltaje o de corriente los cuales, por ejemplo, utilizan redes neuronales que regulan la velocidad del motor [37]. Por otra parte, se han reportado muchas aplicaciones de las redes neuronales en áreas como electrónica de potencia, incluyendo el diagnóstico de fallas en máquinas eléctricas [2]. Al mismo tiempo se han propuesto sistemas inteligentes que controlan el par electromagnético producido en el motor mediante sistemas neuro-difusos [18][19][32]. Estos sistemas han demostrado tener un mejor desempeño que los controladores convencionales, así como también un menor tiempo de cómputo en la simulación e implementación por sus reducidos modelos matemáticos [49].

### **1.1.1. Sistemas Difusos**

En la década pasada, los sistemas difusos han sustituido a las tecnologías convencionales en muchas aplicaciones, especialmente en sistemas de control. Se ha visto un rápido crecimiento en el uso de la lógica difusa en una amplia variedad de productos. La tecnología difusa se puede definir como la que forma una aproximación de la manera en que el razonamiento humano funciona y su resultante toma de decisiones [34]. Los sistemas difusos permiten la interacción del ser humano en un proceso de control ya que tiene la capacidad de contener la experiencia y los conocimientos del operador. Con estas propiedades, las técnicas de lógica difusa encuentran sus aplicaciones más importantes en áreas tales como: control (en la mayoría de los sistemas); análisis cuantitativos (operaciones, manejo de datos); inferencia (sistemas expertos para el diagnóstico, planeación y predicción, procesamiento del lenguaje natural, robótica, ingeniería de software); y búsqueda y recuperación de información (bases de datos), entre otras aplicaciones [38].

### **1.1.2. Redes Neuronales.**

Las redes neuronales son una nueva generación de estructuras computacionales para el procesamiento de información y están deliberadamente diseñadas para hacer uso de los principios de organización del cerebro humano. El principal objetivo en el que se centra la investigación de las redes neuronales es el de modelar al cerebro como un dispositivo computacional que desarrolle tareas que las computadoras tradicionales han desempeñado deficientemente [29]. El comportamiento de una red neuronal, tal

como el cerebro humano, demuestra la habilidad de aprendizaje una vez que ha sido entrenada de antemano. Como lo sistemas difusos, las aplicaciones donde las redes neuronales han tomado un lugar también son promisorias, por ejemplo dispositivos de reconocimiento de voz, moderno software que convierte un dictado en texto, procesamiento de imágenes y percepción visual, aplicaciones médicas, entre otras.

### 1.1.3. Sistemas Neuro-Difusos Integrados.

La lógica difusa y las redes neuronales son tecnologías que se complementan entre sí en el diseño de sistemas inteligentes. Obviamente cada una tiene sus ventajas y desventajas, sin embargo una comparación con las técnicas convencionales sobre la capacidad de respuesta a distintos aspectos referentes al control de procesos, da clara evidencia que al integrar ambas metodologías inteligentes una cubre las desventajas de la otra (vea Tabla 1.1). Las redes neuronales son esencialmente estructuras y algoritmos computacionales de bajo nivel que ofrecen buen desempeño en el procesamiento de datos, mientras que las técnicas de lógica difusa usualmente tienen que ver con procedimientos de razonamiento de mayor nivel que las redes neuronales [38]. Sin embargo como los sistemas difusos no tienen capacidad de aprendizaje sería complicado para que un operador humano cambiara las reglas difusas y las funciones de membresía de un conjunto de datos de entrenamiento.

**Tabla 1. 1** Comparación de los Sistemas Difusos (FS), Redes Neuronales (NN) y Teorías Convencionales (TC)

	FS	NN	TC
Modelos matemáticos	B	M	E
Habilidad de aprendizaje	M	E	M
Habilidad de razonamiento	E	M	E
Representación de información	E	M	E
No linealidades	E	E	R
Habilidad de optimización	M	E	M
Tolerancia a fallas	E	E	M
Tolerancia a la incertidumbre	E	E	M
Operación en tiempo real	E	B	M

\*(E) Excelente, (B) Bueno, (R) Regular, (M) Malo

Al integrar estas dos tecnologías, se sugiere la idea de transformar los mecanismos de diseño en el razonamiento y decisión del sistema difuso en simplemente una red neuronal dirigida mediante información de entrenamiento y aprendizaje. En otras palabras, la red neuronal provee de una sencilla estructura con habilidades de aprendizaje al sistema difuso que a su vez, le proporciona la parte del razonamiento de alto nivel a la red neuronal. Sin duda características deseables en cualquier sistema de control y automatización.

## 1.2. Objetivos de la Tesis.

- ✓ Proponer la aplicación de un esquema de control que permita un buen desempeño en el desarrollo del par electromagnético de los motores de inducción tipo jaula de ardilla a través de la utilización de un controlador basado en los principios y metodologías de la inteligencia artificial relativos a la aplicación de estructuras con redes neuronales y sistemas de inferencia difusa.
- ✓ Investigar y proponer la aplicación de un sistema neuro-difuso cuya arquitectura sea razonablemente efectiva evitando al mismo tiempo que sea excesivamente compleja.
- ✓ Mediante el empleo de técnicas de simulación digital, se pretende analizar el comportamiento del control neuro-difuso en el desempeño del motor de inducción bajo distintas condiciones de operación y de esta manera determinar la factibilidad de su implementación.

## 1.3. Descripción del Problema.

Más de la mitad de la energía generada actualmente es convertida a energía mecánica mediante los motores eléctricos. Entre los distintos tipos de motores que hoy inundan la industria, los motores de inducción han gozado de una gran importancia sobre otros debido a la amplia diversidad de aplicaciones para las que son utilizados [9]. Con la implementación de rectificadores e inversores, el control de la velocidad y par de un motor de inducción ha hecho posible el crecimiento aún mayor de la popularidad de estas máquinas.

Con los adelantos en el desarrollo de nuevos dispositivos electrónicos, el diseño y la implementación de accionamientos eléctricos para motores de inducción se está convirtiendo en un campo muy importante para la investigación en áreas tales como electrónica de potencia, sistemas de control, máquinas eléctricas, entre otras. Sin embargo, la verdad es que aunque se han encontrado resultados significativos, los esquemas de control en ocasiones resultan tener modelos muy complicados y suelen ser sensibles ante la variación de los parámetros de motor. Otro problema es que usualmente dichos esquemas presentan problemas por operar a frecuencias de conmutación elevadas o bien cuando el motor trabaja en bajas velocidades [54][60][62].

El Control Directo del Par (DTC por sus siglas en inglés *Direct Torque Control*), es una estrategia de control para el motor de inducción cuyo principio fundamental es la

de controlar el par electromagnético y el flujo magnético del estator mediante una selección adecuada de los vectores de voltaje del inversor que a su vez alimenta eléctricamente al motor de inducción. Esta selección se realiza mediante la utilización de dos controladores de histéresis cuyas bandas limitan la señal de error del par y el flujo del estator permitiendo así una rápida respuesta en el par electromagnético ante cambios en la señal de referencia. Sin embargo, una de las mayores desventajas del DTC es la variación impredecible de la frecuencia de conmutación en el inversor la cual varía acuerdo a los parámetros de los controladores y a la velocidad con la que el motor esté operando, aún cuando los anchos de banda de los controladores de histéresis utilizados tengan valores constantes. Esto da como resultado que el ancho de banda de histéresis tenga un valor lo suficientemente grande que permita limitar la frecuencia de conmutación por debajo de cierto nivel que es fijado por las restricciones térmicas de los dispositivos de potencia. Sin embargo esto afecta directamente en la trayectoria del vector de flujo del estator como consecuencia existe mayor distorsión armónica en las señales de corriente eléctrica [54].

Otro problema en el DTC es que existen caídas intermitentes en el flujo del estator debidas no se tiene un vector de voltaje que compense el error del flujo y par en el instante en el que el vector de flujo del estator cambia de un sector a otro. Este problema se empeora todavía más cuando la máquina opera en bajas velocidades [50][52].

Aunque se han hecho algunas modificaciones al esquema original del DTC, tales como utilizar un inversor de tres niveles o bien sustituir los controladores de histéresis por un controlador predictivo [4][5][20], estas modificaciones incrementan la complejidad del sistema lo cual requiere de mayor memoria y tiempo computacional para poder llevar a cabo simulaciones o bien su implementación.

Estos problemas nos conducen a la investigación de nuevas técnicas de control para los accionamientos eléctricos. En este trabajo se pretende dar solución al problema de la frecuencia de conmutación variable del inversor, a las caídas en el flujo de estator al mismo tiempo que se simplificará el esquema original del DTC, mediante la utilización de las técnicas y metodologías de la inteligencia artificial.

## 1.4. Justificación.

Los problemas inherentes del DTC expuestos en la sección anterior, sin duda justifican la realización de una investigación que mejore el desempeño del sistema bajo distintas condiciones de operación y que al mismo tiempo permita el control eficiente del par y flujo magnético del motor de inducción. En este trabajo se propone un esquema de control donde se aprovechen las ventajas que presentan las técnicas inteligentes y analizar si es conveniente y factible su implementación.

Aunque por más de cinco décadas se han desarrollado estrategias de control basadas en herramientas convencionales (principalmente matemáticas), las técnicas inteligentes que tratan de imitar el razonamiento humano han demostrado ser útiles para controlar sistemas cuyos modelos son complejos y con estrictos requerimientos tales como altas no linealidades, cambios abruptos en su desarrollo dinámico, entre otros [39]. Para tales sistemas, es sabido que los convencionales en ocasiones no han podido cumplir con tales requerimientos, por tal razón existe la necesidad de desarrollar sistemas de control altamente inteligentes [48].

Las investigaciones recientes de sistemas de control con redes neuronales y/o control difuso ofrecen una nueva y prometedora dirección para un mejor entendimiento e incluso una mejor solución de los problemas y retos que presenta el control automático. En los últimos años se ha demostrado que los sistemas inteligentes tienen el potencial para manejar modelos complejos y no lineales principalmente debido a sus atractivas características: habilidad de aprendizaje, adaptabilidad, procesamiento masivo de información con estructuras relativamente sencillas, etc.

Por lo anterior, a medida que la ciencia avanza a este respecto, es posible creer que el futuro del control automático se sentará sobre los sistemas inteligentes los cuales den mayor seguridad y confiabilidad en los procesos industriales minimizando al mismo tiempo su complejidad.

## 1.5. Estado del Arte.

La aplicación de la electrónica de potencia en controladores de velocidad y par en motores de inducción es bien conocida [3][30][42]. Dichos dispositivos emplean sistemas digitales basados en diversas estrategias de control que en algunos casos presentan algunas dificultades, entre ellas las debidas a la distorsión armónica producida por la rectificación e inversión de las señales eléctricas, las altas no linealidades que caracterizan a estos sistemas y los modelos matemáticos complejos que requieren su solución en línea.

Sin embargo, en años recientes, distintas arquitecturas de redes neuro-difusas han sido presentadas con la finalidad de reforzar el desarrollo de la ingeniería de la inteligencia artificial en sistemas complejos y con altas no linealidades [38][62]. Arquitecturas simples como el ANFIS (por sus siglas en inglés *Adaptive-Network-Based Fuzzy Inference System*) [24] puede modelar los aspectos cualitativos de los procesos del conocimiento y razonamiento humano sin emplear análisis cuantitativos complejos. Esta inteligencia artificial ha encontrado numerosas y prácticas aplicaciones en el control de procesos. Últimamente se ha investigado en la aplicación de redes neuronales en accionamientos eléctricos para motores de inducción, como por ejemplo, el uso de las redes neuronales para estimar el flujo, par y ángulo del campo para un sistema de control por campo orientado [2]. Igualmente se han diseñado sistemas

basados en redes neuronales para calcular las variables de estado y compensar las variaciones de los parámetros del motor de inducción. También se emplean las habilidades de identificación de las redes neuronales para la estimación de la velocidad del motor o bien para el control de las conmutaciones de inversores tipo PWM (por sus siglas en inglés *Pulse Width Modulation*) [1][49].

La investigación en esta área ha demostrado en las últimas décadas que el diseño de controladores con redes neuronales que trabajan en conjunto con las teorías de la lógica difusa, tienen un potencial considerable en las áreas de control automático, reconocimiento y clasificación de modelos, etc. [32][44].

Las aplicaciones de esta tecnología se desarrollan y se incrementan rápidamente. En los últimos 10 años estas aplicaciones incluyen robótica, sistemas eléctricos de potencia, control de plantas químicas, etc. Una de las áreas de interés en donde se empieza a dar la implementación de controles inteligentes es en la electrónica de potencia aplicada en dispositivos que funcionan como fuente de alimentación de máquinas de c.a. En vista que esta clase de sistemas presentan altas no linealidades, la aplicación de un controlador neuro-difuso, objeto de este trabajo, puede mejorar el desarrollo de dichos sistemas. Por ejemplo en [18] se presenta un controlador neuro-difuso para el par del motor de inducción, sin embargo la estructura no proporcionaba acciones de control que resultaran en la eliminación del error en estado estacionario. Más tarde en [19] el mismo autor propone el cálculo de factores numéricos que compensen el error, sin embargo este cálculo sólo es posible a través de la implementación del sistema.

Por otro lado, en la Sección de Estudios de Posgrado e Investigación de la Escuela Superior de Ingeniería Mecánica y Eléctrica en distintas investigaciones se han abordado distintas metodologías para el control de accionamientos de motores de inducción, las cuales incluyen el Control Directo del Par [54], el Auto-control Directo del Par [52], entre otras investigaciones [14][50][51]. En dichos trabajos se presentan algunas modificaciones realizadas al esquema clásico del DTC para mejorar su desempeño principalmente en bajas velocidades, minimizar la variación de la frecuencia de conmutación del inversor, etc.

## 1.6. Estructura de la Tesis.

El capítulo 2 se centra en los fundamentos teóricos en los que se basa las metodologías de la inteligencia artificial utilizadas en este trabajo para el diseño del controlador neuro-difuso. Se presentan los conceptos de la lógica difusa y los sistemas de razonamiento e inferencia difusa. Se introduce las estructuras y el funcionamiento de las redes neuronales artificiales consideradas como sistemas adaptables con aprendizaje supervisado. En este capítulo se establecen las técnicas de regresión y optimización utilizados para el entrenamiento y adaptación de la red. Finalmente se

considera la estructura básica de la red neuronal basada en un sistema de inferencia difusa y su comportamiento como identificador de sistemas.

El capítulo 3 presenta el análisis del control neuro-difuso proponiendo distintas estructuras e incorporando avanzadas estrategias de adaptación y aprendizaje. Se presentan simulaciones del controlador bajo estudio en aplicaciones sencillas usando el aprendizaje y control inverso.

El capítulo 4 presenta la incorporación del controlador diseñado dentro de un esquema modificado del DTC mismo que es utilizado para llevar a cabo el control de par del motor de inducción. Se muestran los resultados de las simulaciones bajo distintas condiciones de operación mediante un programa de simulación. Por último, estos resultados se comparan con aquellos obtenidos mediante esquemas del DTC clásico y predictivo.

Finalmente, en el capítulo 5 se muestran las conclusiones del trabajo, sus aportaciones y recomendaciones para trabajos futuros. Por otra parte, al final de la tesis se encuentran apéndices que proporcionan mayor información sobre el modelo matemático del motor de inducción utilizado, la estrategia del control a través del esquema de DTC, el modelado y simulación del inversor y la técnica PWM utilizada, así como los programas de simulación en lenguaje *C++* utilizados para la simulación del control inteligente propuesto.

## 1.7. Aportaciones.

Las aportaciones más importantes de este trabajo son:

1. Investigación y aplicación de una red neuro-difusa dentro del esquema del control directo del par y flujo magnético del estator la cual se caracteriza por su simplicidad tanto en su estructura como en su aplicación.
2. Se presenta la utilización del filtro de Kalman en el aprendizaje de una red neuro-difusa adaptable como técnica de estimación y regresión lineal misma que minimiza el error en la salida del controlador.
3. Se muestra la utilización del aprendizaje inverso el cual permite, entre otras cosas, la producción de reglas difusas adecuadas para un control apropiado del sistema.
4. Elaboración de los siguientes programas de simulación digital en un lenguaje de programación de alto nivel (*C++*), los cuales pueden ser utilizado para futuras investigaciones o en actividades docentes.

- ✓ Motor de Inducción.
  - ✓ Inversor Fuente de Voltaje con Modulación de Vectores Espaciales (SVM).
  - ✓ Red Neuro-Difusa Adaptable (ANFIS) con el filtro de Kalman como algoritmo de aprendizaje.
  - ✓ Control Neuro-Difuso Directo del Par y Flujo Magnético (DTNFC).
5. Análisis del marco teórico y computacional para posibles implementaciones futuras del sistema.

Los artículos publicados son:

- ✓ Gama, M. & Romero, D. *Representación Simplificada de un Sistema Rectificador-Inversor con Modulador de Vectores Espaciales para Motores de Inducción*. IEEE RVP-AI/2003, Julio 2003.
- ✓ Gama, M. & Romero, D. *Análisis de un Algoritmo Híbrido de Aprendizaje para una Red Neuro-Difusa Adaptable*. Séptimo Congreso Nacional de Ingeniería Electromecánica y de Sistemas. IPN-SEPI ESIME, Noviembre 2003.



## Capítulo 2

# Modelo Neuro-Difuso: Sistemas de Razonamiento Difuso + Redes Neuronales

### 2.1. Introducción.

En este capítulo se presenta en detalle el análisis y diseño de la estructura neuro-difusa que será empleada como controlador, mismo que representa la parte medular de este trabajo. Dicha estructura se emplea para controlar la señal de referencia de voltaje al inversor de potencia que proporciona las señales de voltaje en las terminales del motor de inducción.

Este controlador está diseñado con técnicas de inteligencia artificial e incluso es una combinación de dos de ellas, el razonamiento a través de la lógica difusa sumada a la capacidad de aprendizaje y adaptación de las redes neuronales, de ahí el nombre de modelo neuro-difuso. Bajo este marco, en el presente capítulo se introducen los conceptos teóricos que fundamentan este trabajo, así como también se muestran las técnicas de identificación y estimación empleadas para que el controlador cuente con funciones adaptables a distintas condiciones de operación de manera sencilla.

Cuando los sistemas de inferencia difusa se representan mediante una estructura o red adaptable, dichos sistemas se conocen como sistemas de inferencia difusa basados en redes adaptables (ANFIS por sus siglas en inglés *Adaptive Networks-Based Fuzzy Inference Systems*). El ANFIS es una estructura equivalente a un sistema de inferencia difuso tipo Takagi-Sugeno con la única diferencia que su estructura, similar a una red neuronal, tiene la capacidad de aprendizaje y adaptación. Esto hace que todas las metodologías de diseño para redes neuronales sean aplicables también a este controlador difuso convirtiéndolo en una herramienta superior [23].

A continuación se presentan brevemente los conceptos de la lógica difusa aplicada a los sistemas de control y sus respectivas características tales como el razonamiento difuso, las reglas que gobiernan la inferencia difusa y los distintos tipos de sistemas difusos. Posteriormente, se presenta el estudio de las redes adaptables y sus

correspondientes reglas de aprendizaje y adaptación. Finalmente, se expone en detalle la arquitectura del controlador neuro-difuso utilizado en este trabajo.

## 2.2. Sistemas de Razonamiento Difuso.

Un sistema de razonamiento difuso (FIS por sus siglas en inglés *Fuzzy Inference System*) es un marco de trabajo computacional basado en los conceptos de los conjuntos difusos, reglas difusas y razonamiento difuso expuesto en la sección anterior. Estos sistemas han encontrado una gran variedad de aplicaciones tales como en ingeniería, ciencia, negocios, medicina, psicología, y otros campos [13][34][48].

Por ejemplo, en ingeniería algunas aplicaciones potenciales incluyen las siguientes áreas:

- ✓ **Naves aeroespaciales:** Control de vuelo, control mecánico, sistema de navegación, diagnóstico de fallas, control satelital.
- ✓ **Control automatizado en autopistas:** semáforos, sistemas de dirección y orientación automáticos.
- ✓ **Automóviles:** sistemas de frenado, transmisión, suspensión, control mecánico.
- ✓ **Vehículos autónomos:** dirección, orientación, control de velocidad.
- ✓ **Sistemas industriales:** control de procesos.
- ✓ **Industria eléctrica y de potencia:** Control de máquinas, control de sistemas eléctricos de potencia, estimación de carga.
- ✓ **Control de procesos:** Temperatura, presión, control de niveles, diagnóstico de fallas, control de columnas de destilación.
- ✓ **Robótica:** Control de posición.

Cabe mencionar que esta lista cubre aquellas aplicaciones que han sido sujetas a estudio en la literatura. Actualmente, la lógica difusa sigue extendiéndose rápidamente debido a su sencillez y su relación con el mundo real.

### 2.2.1. Estructura de un Sistema de Razonamiento Difuso.

La estructura básica de un FIS consiste en tres componentes principales: una *base de reglas*, la cual contiene una selección de las reglas difusas; una *base de datos* que define las funciones de membresía utilizadas en las reglas difusas donde al mismo tiempo se incluye el proceso de difusificación; y un *mecanismo de razonamiento*, mismo que desarrolla un procedimiento de razonamiento basado en la lógica difusa, el cual proporciona las acciones concluyentes a partir de las premisas de entrada [24].

Es importante notar que las entradas del FIS pueden ser tanto difusas como nítidas, pero que las salidas producidas son casi siempre conjuntos difusos. A veces es necesario tener salidas nítidas, especialmente cuando el FIS se utiliza como controlador. Por consiguiente, es necesario un método de *dedifusificación* para obtener un valor nítido que mejor represente a un conjunto difuso. La figura 2.1 muestra un FIS cuya salida es nítida. La línea continua indica la representación básica del FIS donde su salida es difusa y pasa por el bloque de dedifusificación para obtener el valor nítido.

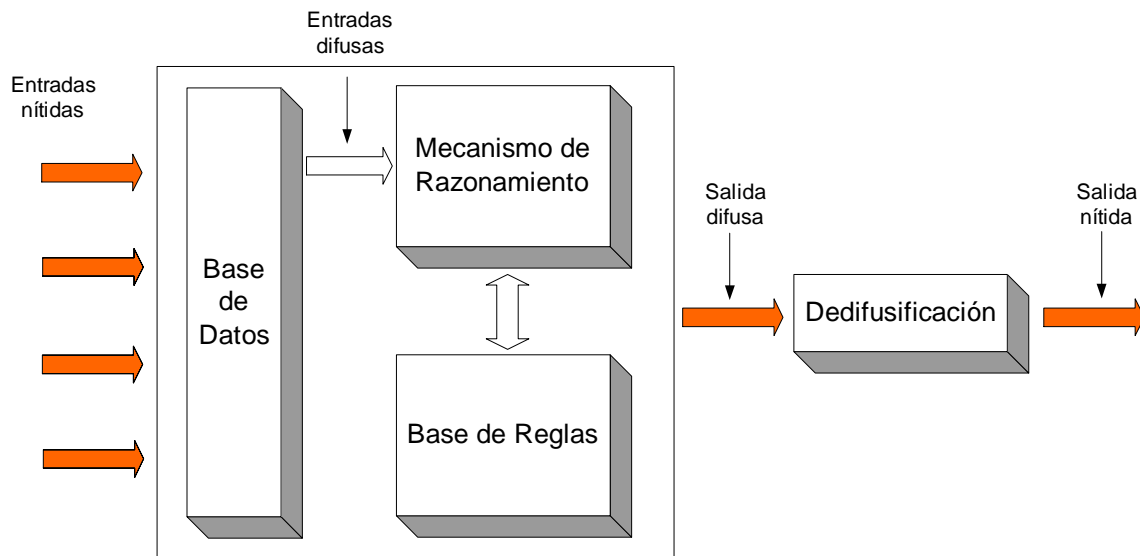


Figura 2. 1. Esquema básico de un FIS con múltiples entradas y una salida.

Con entradas y salidas nítidas, un FIS implementa una transformación no lineal de su espacio de entrada hacia su espacio de salida. Esta transformación se realiza a través de las reglas difusas, las cuales describen el comportamiento local de dicha transformación.

En resumen, en la figura anterior se observa que se realizan cuatro procesos para llevar a cabo la transformación no lineal [15]:

1. **Difusificación**, el cual se refiere a la manera en que un valor nítido es transformado a un valor difuso a través de una función de membresía de un determinado valor lingüístico. El valor difuso representa el grado de membresía del conjunto difuso. El bloque de base de datos contiene las funciones de membresía utilizadas en las reglas difusas, las cuales convierten las entradas del FIS en información que el mecanismo de razonamiento puede usar fácilmente para activar y aplicar las reglas.
2. **Base de reglas**, el cual procesa la transformación sobre el conjunto de las reglas difusas que representan la descripción de las relaciones de las variables lingüísticas.
3. **Mecanismo de razonamiento**, el cual proporciona la toma de decisiones al interpretar y aplicar el conocimiento y los procedimientos del control.
4. **Defusificación**, el cual convierte las conclusiones del razonamiento en señales reales de control para la planta o proceso.

Existen tres tipos de FIS que han sido empleado ampliamente en varias aplicaciones. Estos son:

- ✓ Modelo de Mamdani.
- ✓ Modelo Takagi-Sugeno.
- ✓ Modelo Tsukamoto.

La diferencia entre estos tres recae únicamente en la parte consecuente de sus reglas difusas y por consiguiente, los procedimientos para producir su correspondiente salida nítida. Cabe mencionar que en este trabajo, el diseño del controlador neuro-difuso esta basado en el modelo de Takagi-Sugeno por lo que los otros modelos no son abordados.

### 2.2.2. Sistema tipo Takagi-Sugeno.

El modelo de Takagi-Sugeno (TSK) fue propuesto en un esfuerzo por desarrollar un método sistemático para generar reglas difusas desde un conjunto de datos conocidos correspondientes a los valores de las señales de entrada-salida. Este tipo de sistema se conoce también como un *sistema difuso funcional* [48]. Una regla difusa en este modelo para un sistema múltiples entradas una salida tiene la forma siguiente:

$$\text{si } x_1 \text{ es } A_1^j \text{ y } x_2 \text{ es } A_2^k \text{ y, } \dots, \text{ y } x_n \text{ es } A_n^l \text{ entonces } y_i = g_i(\cdot) \quad (2.1)$$

donde “ $\cdot$ ” simplemente representa el argumento de la función  $g_i$ , y  $y_i$  es la salida del FIS como valor nítido. La parte antecedente de esta regla se define de la misma forma que en los otros modelos difusos, sin embargo la parte consecuente de la regla es diferente. En lugar de un término lingüístico con una función de membresía asociada, se utiliza una función  $y_i = g_i(\cdot)$ , de ahí el nombre de sistema difuso funcional. Usualmente, el argumento de  $g_i$  contiene los términos  $x_i$ ,  $i = 1, 2, \dots, n$ , aunque otras variables pueden ser empleadas. La selección de la función depende de la aplicación. En este caso se emplea una función lineal por lo que el sistema se convierte en un **modelo difuso tipo TSK de primer orden**. De tal forma que:

$$y_i = g_i(\cdot) = a_{i,0} + a_{i,1}x_1 + a_{i,2}x_2 + \dots + a_{i,n}x_n \quad (2.2)$$

donde  $a_{ij}$  son números reales. Y para la salida global del sistema:

$$y = \frac{\sum_{i=1}^R w_i y_i}{\sum_{i=1}^R w_i} \quad (2.3)$$

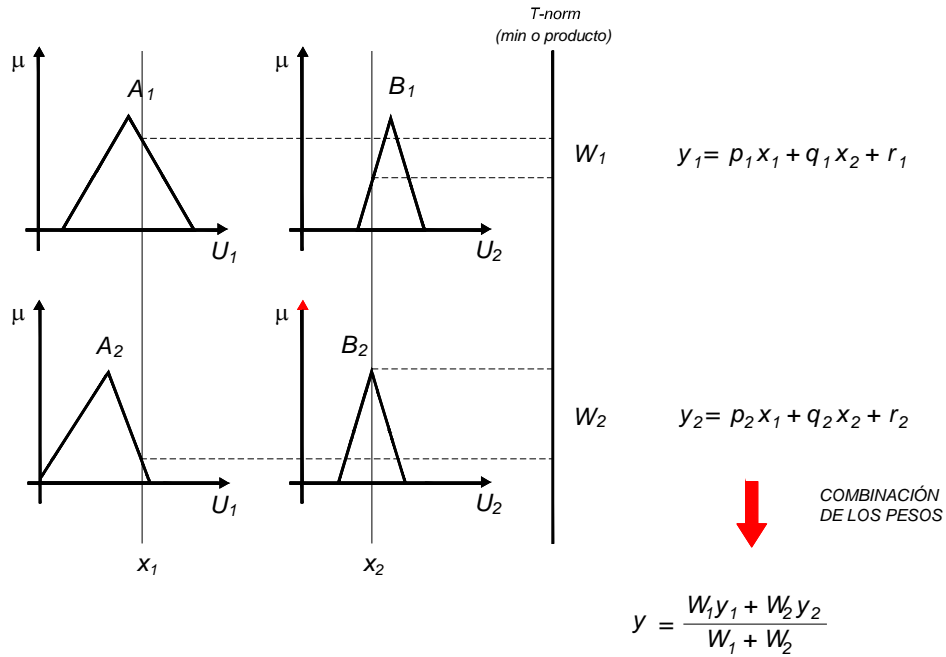
donde  $w_i$  es el peso de cada regla asociada a cada entrada y se define de la siguiente forma:

$$w_i(x_1, x_2, \dots, x_n) = \mu_{A_1^j}(x_1) * \mu_{A_2^k}(x_2) * \dots * \mu_{A_n^l}(x_n) \quad (2.4)$$

Como se puede ver, el FIS tipo TSK realiza una interpolación no lineal entre transformaciones lineales. La figura 2.2 muestra un modelo TSK donde  $n = 2$  y  $R = 2$ , teniendo las siguientes reglas:

$$R_1: \text{ si } x_1 \text{ es } A_1 \text{ y } x_2 \text{ es } B_1 \text{ entonces } y_1 = p_1 x_1 + q_1 x_2 + r_1$$

$$R_2: \text{ si } x_2 \text{ es } A_2 \text{ y } x_2 \text{ es } B_2 \text{ entonces } y_2 = p_2 x_1 + q_2 x_2 + r_2$$



**Figura 2. 2.** Modelo Difuso Takagi-Sugeno con dos entradas, una salida.

Dado  $U_i$  como el universo del discurso de  $x_i$ , entonces, combinando las salidas locales se tiene:

$$y = \frac{w_1 y_1 + w_2 y_2}{w_1 + w_2} \quad (2.5)$$

donde

$$w_1 = \mu_{A_1}(x_1) * \mu_{B_1}(x_2) \quad (2.6)$$

$$w_2 = \mu_{A_2}(x_1) * \mu_{B_2}(x_2) \quad (2.7)$$

Debido a que cada regla tiene una salida nítida, la salida del FIS se obtiene al combinar los pesos de todas las reglas activadas, evitando de esta forma el proceso de defusificación requerido en otros los modelos difusos.

Una de las razones por las cuales el modelo TSK es más eficiente, es debida a que, por su representación matemática en la parte consecuente de sus reglas difusas, puede llevarse a la simulación digital con mayor facilidad y permite el uso de técnicas de optimización las cuales pueden aplicarse para convertir al FIS en un sistema

adaptable. Esto es una característica muy conveniente cuando se utiliza como controlador de un sistema con desempeños dinámicos variantes [24].

## 2.3. Redes Neuronales Artificiales.

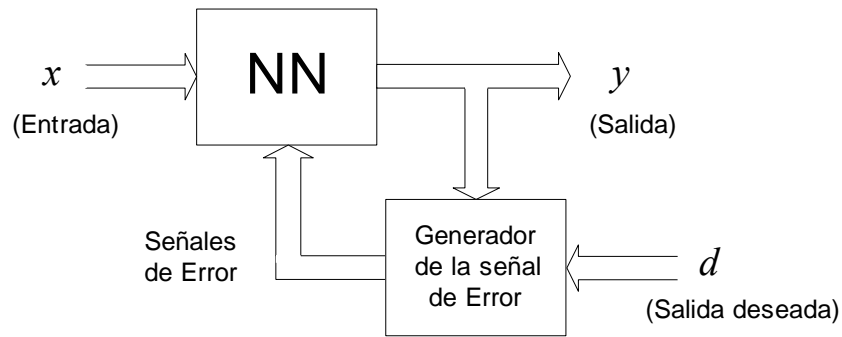
Las redes neuronales artificiales (NN, por sus siglas en inglés *Neural Networks*) son sistemas que están contruidos para hacer uso de los principios de organización de información que caracteriza al cerebro humano. Representan una nueva generación de sistemas que procesan y ordenan información lo que las hace adecuadas para tareas como clasificación y asociación de datos, aproximación de funciones, optimización, reconocimiento, etc [29]. Las NN tienen un gran número de elementos interconectados a través de nodos que usualmente operan en paralelo configurados en varias arquitecturas. El desempeño colectivo de una NN, tal como el cerebro humano, demuestra la habilidad de aprender y reconocer a partir de datos de entrenamiento.

En años recientes, numerosos modelos de NN han sido propuestos en la literatura los cuales pueden ser clasificados tomando en consideración distintos criterios, tales como sus métodos de aprendizaje (supervisado y no supervisado), sus arquitecturas (progresivas o recurrentes), tipos de salida (binarias o continuas), tipos de nodos (uniformes o híbridos, implementaciones (software o hardware), y así por el estilo.

El objetivo del entrenamiento de la NN es la de obtener señales de salida deseados a partir de cierta clase de estímulos; esto hace que la regla de aprendizaje utilizada juegue un papel muy importante en la utilización de una NN [38].

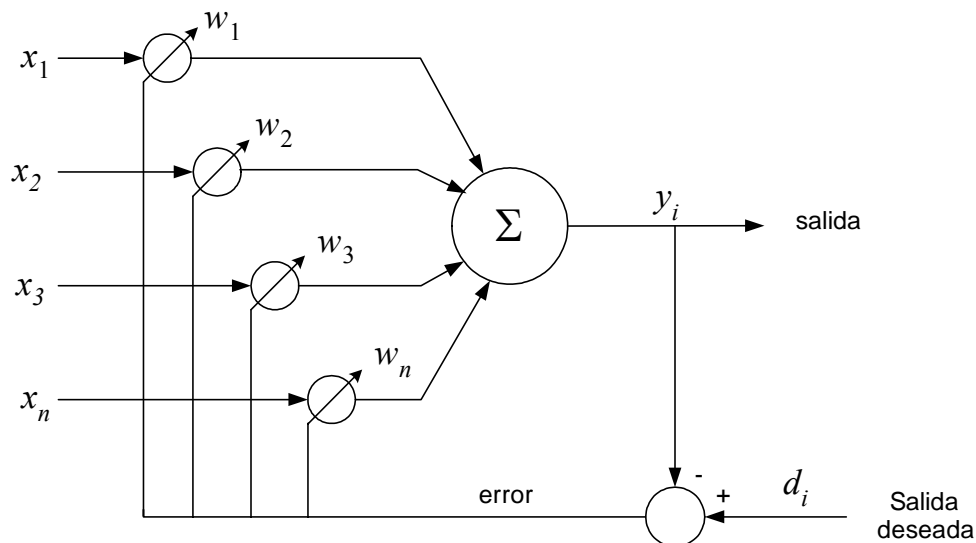
### 2.3.1. Aprendizaje Supervisado de la Red Neuronal.

Existen dos tipos principales de estrategias: aprendizaje para el ajuste de los parámetros de la red, el cual concierne principalmente al ajuste de los pesos de las conexiones, y el aprendizaje para el ajuste de la estructura de la red, que se relaciona con el cambio en el número de capas, conexiones o neuronas. Cabe mencionar que en la mayoría de los casos, incluyendo en este trabajo, se utiliza el aprendizaje para el ajuste de los parámetros de la red. La figura 2.3 presenta el esquema general del aprendizaje supervisado. Como se observa, cuando se aplica la señal de entrada  $x$ , se proporciona en cada instante de tiempo la señal de respuesta deseada  $d$ . En este caso, para entrenar la red es necesario proporcionarle una secuencia de ejemplos,  $(x_1, d_1)$ ,  $(x_2, d_2)$ , ...,  $(x_n, d_n)$  correspondientes a los pares de datos entrada-salida. La diferencia entre la salida  $y$  y la salida deseada  $d$  se calcula en el generador de las señales de error el cual produce la señal de error para que la NN ajuste sus pesos de tal forma que minimice el error y la salida actual se mueva lo más cercano posible a la salida deseada.



**Figura 2. 3.** Esquema del aprendizaje supervisado de una red neuronal.

Un tipo común de NN es la conocida como **ADALINE** (*adaptive linear element*) la cual representa el ejemplo clásico del sistema inteligente de auto-aprendizaje más simple que puede adaptarse automáticamente para lograr una tarea específica. La figura 2.4 muestra el diagrama de esta red.



**Figura 2. 4.** Diagrama de ADALINE (*adaptive linear element*).

Como se observa esta red tiene un bloque de salida puramente lineal lo que hace que la salida de la red  $y$ , sea una combinación lineal de las entradas más un término constante, tal que:



$$y = \sum_{i=1}^n w_i x_i + w_o \quad (2.8)$$

Es obvio que la ecuación (2.6) es un modelo lineal con  $n + 1$  parámetros lineales, de tal forma que se pueden aplicar técnicas simples de estimación como el de cuadrados mínimos considerado como el índice de optimización para ajustar los pesos de la red y minimizar así el error [24][43].

De acuerdo a los conceptos básicos expuestos anteriormente, para que una NN logre un buen desempeño en una determinada aplicación, en el diseño se deben tomar en cuenta los siguientes aspectos:

- ✓ La arquitectura de la red no debe ser excesivamente compleja, lo cual no significa que entre más compleja se obtiene un mejor desempeño.
- ✓ Simplicidad en el aprendizaje.
- ✓ Adaptación en procesos variantes.

La capacidad de adaptación de una NN la convierte en una técnica de inteligencia artificial muy útil. Cuando se logra obtener un modelo de NN que pueda satisfacer los puntos anteriores se puede garantizar, en la mayoría de los casos, excelentes resultados. Esto se comprueba en las innumerables aplicaciones en las que las NN son utilizadas en la actualidad [44].

### 2.3.2. Redes Adaptables.

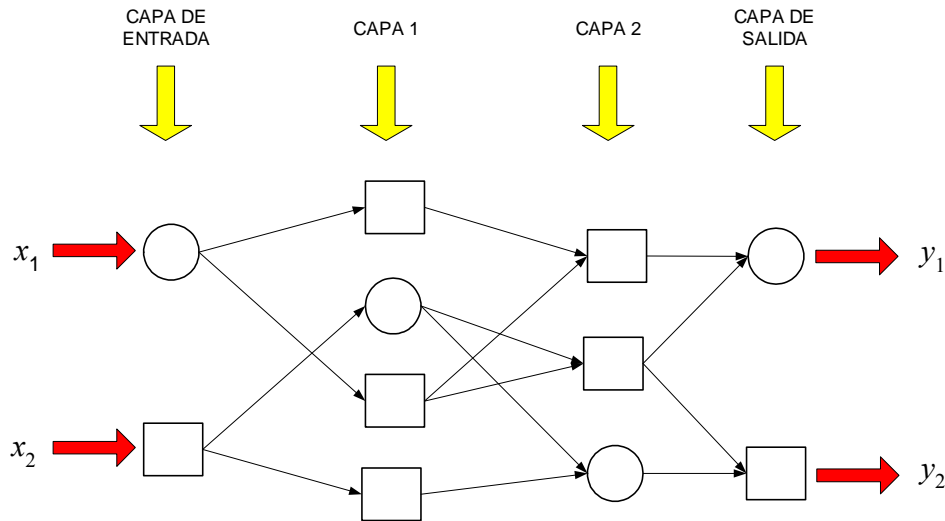
En esta sección se introduce la arquitectura y los procesos de aprendizaje de una red adaptable la cual es de hecho una clase de NN con capacidad de aprendizaje supervisado.

Como ya se mencionó, la adaptación se considera como el proceso de modificar los parámetros del sistema. Se utiliza información de referencia para obtener un estado definitivo y óptimo cuando las condiciones de operación son inciertas y variantes [7]. Este tipo de NN busca continuamente un punto óptimo dentro de sus posibilidades mediante un proceso automático de adaptación que permita desempeñar un desarrollo superior a una no adaptable.

Una red adaptable es una estructura que consiste en una serie de nodos que se relacionan entre sí a través de conexiones que muestran la dirección en la que la información fluye en la red [25]. Puede que todos o sólo algunos de los nodos sean adaptables, lo cual quiere decir que sus salidas dependen de los parámetros que los

caracterizan. La regla de aprendizaje especifica como estos parámetros deben ser ajustados para minimizar un valor de error prescrito, mismo que es la expresión matemática que mide la discrepancia entre la salida actual de la red y la salida deseada, por lo que en el diseño de la red el objetivo primordial es la de encontrar la arquitectura más apropiada y el conjunto de parámetros que mejor modelen el sistema desconocido descrito únicamente por una serie de datos de entrada-salida [24].

Como su nombre lo indica, la red adaptable es una estructura cuyo comportamiento global está determinado por un conjunto de parámetros variables [23]. Usualmente la función de un nodo es una función cuyos parámetros pueden ser modificados; al cambiar estos parámetros, se cambia la función del nodo así como también el comportamiento global de la red. De lo anterior se desprende la salida de cada nodo depende únicamente de su entrada. Para facilitar el desarrollo de algoritmos de aprendizaje, en este trabajo se considera que todas las funciones de los nodos son diferenciables. En el caso más general, una red adaptable es heterogénea de tal forma que cada nodo puede tener distintas funciones. Cabe mencionar que cada conexión es utilizada sólo para especificar el sentido de propagación de la salida del nodo; generalmente no hay pesos asociados con las conexiones. Por ejemplo, la figura 2.5 muestra una red adaptable con dos entradas y dos salidas.



**Figura 2. 5.** Estructura de una red adaptable.

Los parámetros de una red adaptable están distribuidos en todos los nodos, así que todos tienen un conjunto local de parámetros. La unión de todos los parámetros locales conforma el conjunto global de parámetros. Como se ve en la figura 2.5, existen dos clases de nodos, los nodos cuadrados representan nodos con un conjunto de parámetros

locales por lo que es un **nodo adaptable**. Los nodos circulares representan aquellos que no son adaptables, por lo que no contienen parámetros ya que son **nodos estáticos**.

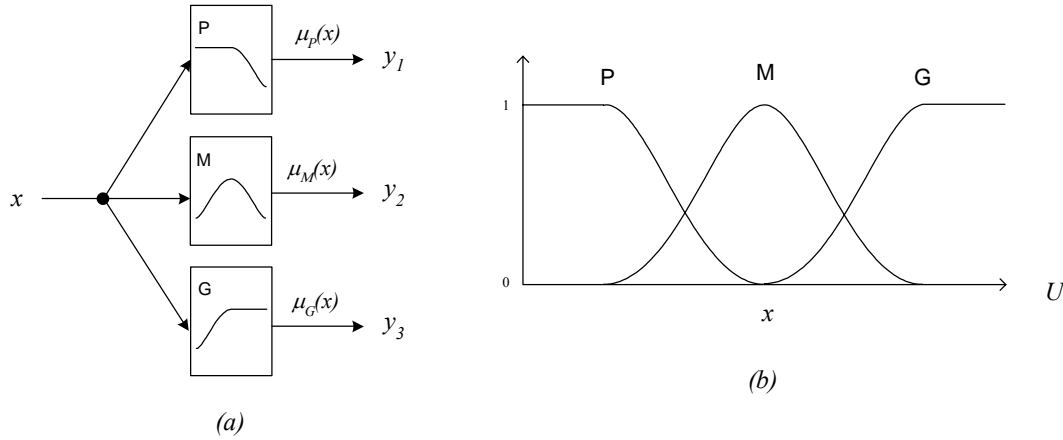
Las redes adaptables son generalmente clasificadas en dos categorías: **progresivas** y **recurrentes** [38]. La figura 2.5 muestra una red adaptable tipo progresiva debido a que la salida de cada nodo propaga la señal desde el lado de entrada (derecho) hacia el lado de la salida (izquierdo). Si hubiese una conexión que forme una trayectoria circular dentro de la red, entonces sería una red recurrente.

Conceptualmente, una red adaptable tipo progresiva es realmente una transformación entre sus espacios de entrada y salida; esta transformación puede ser una relación lineal o bien una con altas no linealidades, dependiendo de la estructura de la red y la función de cada nodo. En este caso, el objetivo es construir una red que aproxime una transformación no lineal regulada por un conjunto de datos que consisten en pares ordenados que representan los datos de entrada y la correspondiente salida deseada de un sistema planteado. Este es el **conjunto de datos de entrenamiento**, y los procedimientos que se siguen para ajustar dichos parámetros con el fin de mejorar el desempeño de la red son usualmente conocidos como **reglas de aprendizaje** o **algoritmos de adaptación** [26]. Regularmente el desempeño de una red se mide por las discrepancias entre la salida deseada y la salida de la red obtenidas bajo las mismas condiciones; esta discrepancia se conoce como **medida de error** y puede asumir diferentes formas para diferentes aplicaciones. Generalmente, la regla de aprendizaje se selecciona mediante aplicar una técnica de optimización específica para un valor del error dado.

## 2.4. Redes Neuronales basadas en Sistemas Difusos.

La habilidad de aprendizaje de las NN así como la manera de organización de su arquitectura y sus cualidades de adaptación son utilizadas para formar sistemas de razonamiento difuso con la finalidad de aumentar su potencial como sistemas inteligentes. Al unir estas dos metodologías es posible construir un sistema donde todas los fundamentos de las NN y la lógica difusa puedan ser aplicadas, superando el desempeño de dichas metodologías tratadas por separado [38][53].

En muchas aplicaciones de la lógica difusa, las funciones de membresía tienen formas regulares tales como triangulares o campanas. Estas funciones de membresía pueden ser implementadas en una neurona al seleccionar dicha función de membresía como su función de activación. Tal como se aprecia en la figura 2.6(a) donde las salidas  $y_1$ ,  $y_2$  y  $y_3$  representan el grado de membresía que la variable lingüística  $x$  tiene en los conjuntos “Pequeño (P)”, “Medio (M)” y “Grande (G)” de la figura 2.6(b).



**Figura 2. 6.** Red neuronal equivalente a tres funciones de membresía de un sistema difuso.

donde:

$$y_1 = \mu_P(x) = \frac{1}{1 + e^{a_P x - c_P}} \quad (2.9)$$

$$y_2 = \mu_M(x) = e^{\frac{-(x - c_M)^2}{2a_M^2}} \quad (2.10)$$

$$y_3 = \mu_G(x) = \frac{1}{1 + e^{-a_G(x - c_G)}} \quad (2.11)$$

Como se ve en la figura 2.6, es relativamente sencillo la construcción de una NN equivalente a un FIS, no solo incluyendo sus funciones de membresía sino también sus operadores lógicos, las reglas difusas y la dedifusificación (si el modelo difuso la necesita). La naturaleza no adaptable de un FIS puede eliminarse con el uso de las NN, por lo que la combinación de ambas técnicas produce el desarrollo de nuevas estructuras y algoritmos que provean desempeños adaptables al mismo tiempo de que se mantiene intacta las características de representación del razonamiento humano que proporciona la lógica difusa [21].

#### 2.4.1. Sistema de Razonamiento Neuro-Difuso Adaptable.

Como se mencionó anteriormente, prácticamente no hay límites en la selección de funciones de activación en los nodos de una red neuronal adaptable a reserva que deben ser diferenciables. En la parte arquitectónica de la red la única limitación que existe para obtener el equivalente de un FIS, es que la red debe ser del tipo progresiva (feedforward), ya que de lo contrario se tendrían que utilizar modelos y algoritmos más

complicados. En esta sección se introduce una clase de NN adaptable que es funcionalmente equivalente a un FIS. La red es conocida como *ANFIS* (*Adaptive Neuro-Fuzzy Inference System*) propuesta por J.-S. R. Jang en la década de los 90's [23]. Esta estructura es equivalente a un FIS tipo TKS de primer orden considerado en la sección 2.2.2. Esta estructura neuro-difusa será utilizada para operar como controlador de las señales de apertura y cierre de los IGBT's del inversor fuente de voltaje que a su vez alimenta al motor de inducción. Su aplicación se presenta en el capítulo 4.

#### 2.4.2. Estructura de la Red Neuro-Difusa.

Por simplicidad se asume que el controlador cuenta con dos entradas  $x_1$  y  $x_2$ , y una salida  $y$ . Para el sistema de razonamiento difuso de primer orden tipo TSK, el conjunto de reglas considerando dos conjuntos difusos es:

- Regla 1: si  $x_1$  es  $A_1$  y  $x_2$  es  $B_1$ , entonces  $f_1 = p_1x_1 + q_1x_2 + r_1$   
 Regla 2: si  $x_1$  es  $A_2$  y  $x_2$  es  $B_2$ , entonces  $f_2 = p_2x_1 + q_2x_2 + r_2$

La figura 2.7(a) ilustra el sistema de razonamiento TSK, mientras que la figura 2.7(b) presenta su equivalencia en una NN adaptable, donde los nodos pertenecientes a la misma capa operan de forma similar. Asimismo se observan los dos tipos de nodos: los cuadrados con parámetros adaptables, y los nodos circulares cuyos parámetros son fijos por lo que la red tiene propiedades de adaptación.

Para describir la estructura de la red neuro-difusa, la salida del  $i$ -ésimo nodo perteneciente a la capa  $l$  se denota como  $O_{l,i}$ .

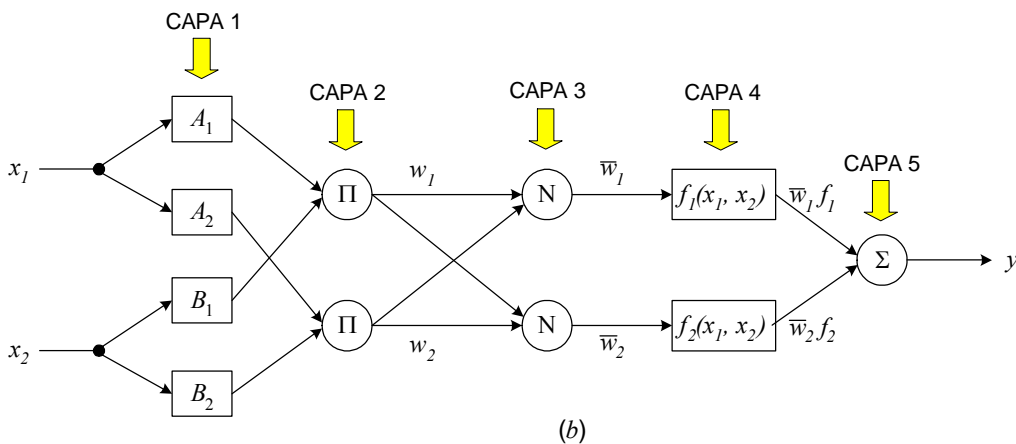
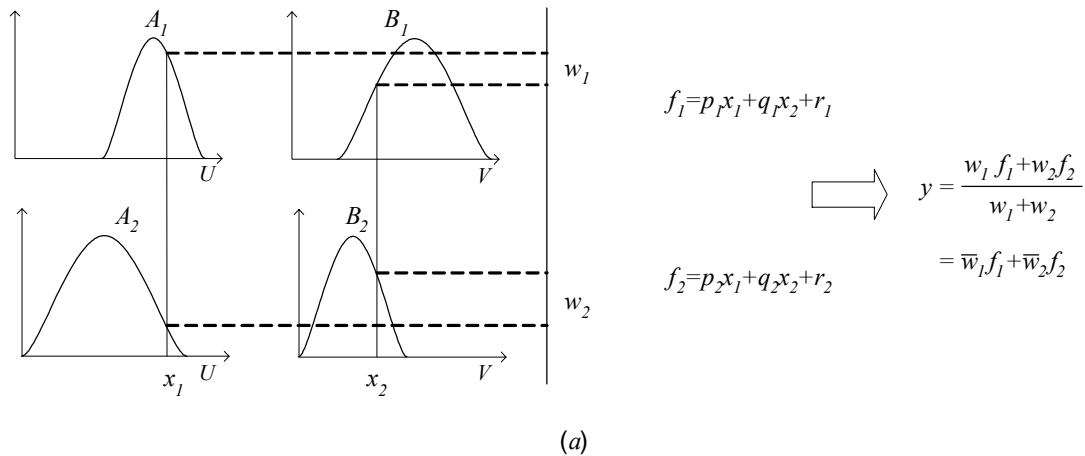
CAPA 1: Cada nodo  $i$  en esta capa es un nodo adaptable y contiene la función de membresía que representa al conjunto difuso, tal que:

$$\begin{aligned} O_{1,i} &= \mu_{A_i}(x_1), & \text{para } i = 1,2, \text{ o} \\ O_{1,i} &= \mu_{B_{i-2}}(x_2), & \text{para } i = 3,4 \end{aligned} \quad (2.12)$$

donde  $x_1$  (o  $x_2$ ) es la entrada al nodo  $i$  y  $A_i$  (o  $B_{i-2}$ ) es el valor lingüístico (como “pequeño” o “grande”) asociada a este nodo. En otras palabras la salidas de los nodos de la capa 1 es el grado de membresía del conjunto difuso  $A$  (sea  $A_1$ ,  $A_2$ ,  $B_1$  o  $B_2$ ). Aquí se utiliza la función campana cuya ecuación es:

$$\mu_A(x) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \quad (2.13)$$

donde  $\{a, b, c\}$  es el conjunto de los parámetros pertenecientes a este nodo. Como los valores de estos parámetros cambian, la función campana varía por lo que puede tomar distintas formas. Los parámetros de este nodo se conocen como **parámetros antecedentes**.

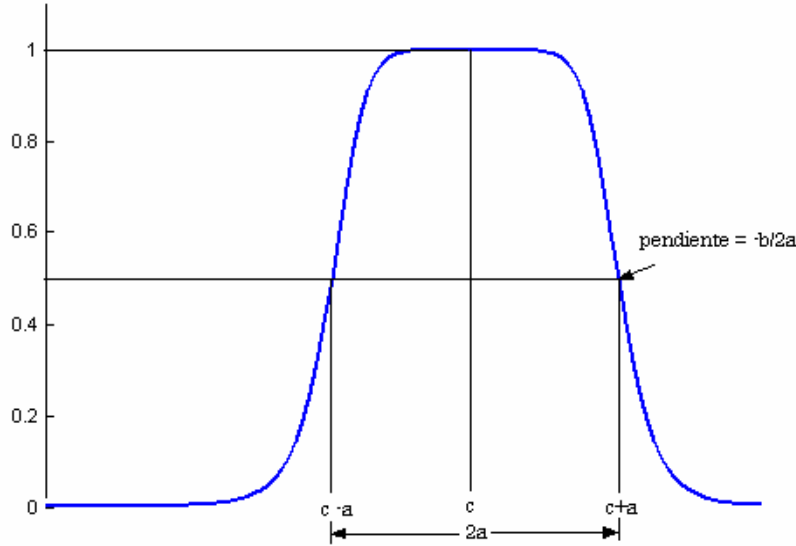


**Figura 2. 7.** (a) Sistema de razonamiento difuso tipo TSK, dos entradas, una salida. (b) Red neuronal adaptable equivalente.

La figura 2.8 muestra el significado físico de los parámetros antecedentes para la función de membresía campana utilizada en este trabajo.

**CAPA 2:** Cada nodo de esta capa es estático (no adaptable) etiquetado con  $\Pi$  y su operación esta asociada con el operador intersección *T-norm* el cual representa la relación difusa intersección o bien la operación lógica AND de la reglas difusas. Por lo tanto la salida de cada nodo es el producto (o valor mínimo) de las señales de entrada, tal que:

$$O_{2,i} = w_i = \mu_{A_i}(x_1)\mu_{B_i}(x_2) \quad i = 1,2 \quad (2.14)$$



**Figura 2. 8.** Significado de los parámetros de la función de membresía campana.

CAPA 3: Cada nodo  $i$  en esta capa es un nodo estático (no adaptable) etiquetado con  $N$ . El  $i$ -ésimo nodo calcula el radio del peso de las regla difusa activada entre la suma de todas los pesos de las reglas activadas de la red. Los nodos de esta capa representan el siguiente cálculo:

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1,2 \quad (2.15)$$

CAPA 4: cada nodo de esta capa es un nodo adaptable con la siguiente función:

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_i x_1 + q_i x_2 + r_i) \quad (2.16)$$

donde  $\bar{w}_i$  es el peso normalizado proveniente de la capa 3 y  $\{p_i, q_i, r_i\}$  es el conjunto de los parámetros de este nodo. Estos parámetros se conocen como **parámetros consecuentes**.

CAPA 5: El único nodo de esta capa, etiquetado como  $\Sigma$ , es estático y calcula la salida global de la red definida como la suma de todas las señales que entran a este nodo, tal que:

$$O_{5,1} = \sum \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} = y \quad (2.17)$$

Así la estructura del ANFIS queda completada aunque cabe mencionar que la estructura presentada no es la única, ya que se pueden obtener otras estructuras equivalentes a los otros modelos de inferencia difusa [38][53].

Una parte importantísima del sistema de razonamiento neuro-difuso son los algoritmos de optimización que proporcionan a la red la capacidad de adaptación. La siguiente sección abarca este aspecto.

## 2.5. Identificación y Estimación.

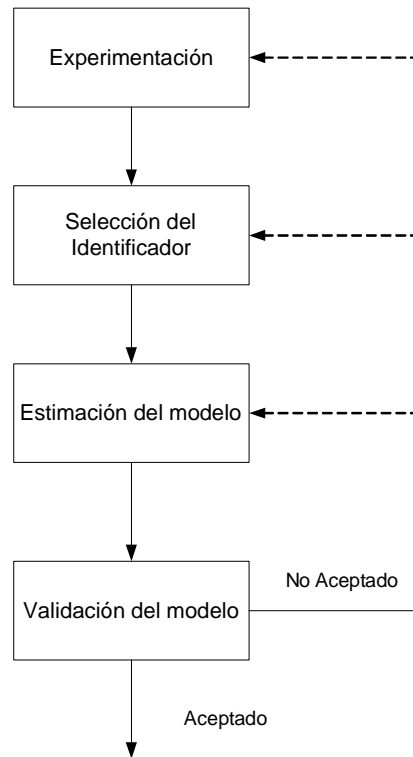
La identificación de sistemas es una tarea que implica la obtención de una descripción matemática, un modelo, de un sistema dinámico a partir de una serie de experimentos a base de mediciones del sistema. Dependiendo del nivel de conocimiento que se tenga del sistema, el problema de la identificación puede abordarse de diferentes maneras. Si la identificación se basa exclusivamente en datos proporcionados por mediciones, asumiendo que no se conoce la parte física del sistema, entonces el proceso de identificación se conoce como modelado de *caja negra*. Este tipo de identificación es útil puesto que en muchos casos, es difícil conocer a detalle la dinámica de un sistema, y el tratar de obtener un modelo que lo represente puede significar complicados e infructíferos procedimientos. Las no linealidades son frecuentemente tan severas que el emplear un modelo no lineal en el diseño de un controlador puede mejorar grandemente el desempeño del sistema [44]. Esta sección presenta como la red neuro-difusa presentada en la sección anterior puede implementarse como una extensión natural de técnicas de identificación convencionales.

Es posible considerar la estimación de los valores de los parámetros variables de la red como un problema de identificación ya que cuando la red es aplicada en cierto proceso no se cuenta con un modelo matemático del sistema sino únicamente se conoce su comportamiento a través de sus datos de entrada y salida [10]. En este caso, al aplicar técnicas de optimización a la red se pretende estimar sus parámetros de manera que el modelo resultante pueda describir el sistema apropiadamente. Otro problema que debe resolver la identificación es el momento en que los métodos de estimación deben aplicarse, es decir, cuando la red está aprendiendo *fuera de línea* (no



conectado al sistema de control por ejemplo) o bien aprendiendo *en línea* (al mismo tiempo que se está aplicando y operando junto con el sistema).

El procedimiento del proceso de la identificación de un sistema dinámico consiste en los pasos que se muestran en la figura 2.9 [43][44]:

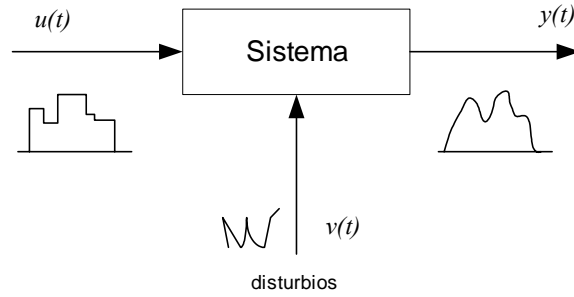


**Figura 2. 9.** Procedimientos básicos para la identificación de un sistema.

**Experimentación:** el propósito de la experimentación es la de coleccionar un conjunto de datos que describen como el sistema se comporta en un rango completo de operación. La idea es variar la entrada o entradas del sistema  $u(t)$ , y observar el impacto sobre la salida,  $y(t)$ . (ver figura 2.10)

El conjunto de datos obtenido de la experimentación puede expresarse como:

$$Z^N = \{[u(t), y(t)], T = 1, \dots, N\} \quad (2.18)$$



**Figura 2. 10.** Fase de experimentación en el proceso de identificación del sistema..

**Selección de identificador:** Para este trabajo se utiliza una estructura neuro-difusa misma que por sus características permite ser aplicada tanto como un medio de identificación como también para validar posteriormente el modelo obtenido.

**Estimación del Modelo:** Una vez que se cuenta con el conjunto de datos y el identificador se procede a formular un modelo matemático a través de la estimación matemática usando estrategias de optimización y regresión. Este proceso también se le conoce como *entrenamiento o aprendizaje*.

**Validación:** Finalmente cuando el modelo ha sido estimado, este debe ser evaluado para investigar si reúne los requerimientos necesarios de exactitud limitados por el sistema mismo. La validación está estrechamente relacionada con el uso del modelo mediante alguna simulación.

Este proceso de identificación cuadra perfectamente con el uso de la red neuro-difusa propuesta en este trabajo. Como se mencionó con anterioridad, las redes adaptables son un tipo avanzado de NN con aprendizaje supervisado, por lo que se asume que se conoce el valor de la salida deseada para cada instante de tiempo y que corresponde a las entradas de la red. La salida global de la red adaptable puede representarse como una función de los valores de entrada y el conjunto de parámetros adaptables, luego entonces [23]:

$$y = F(x, S) \quad (2. 19)$$

donde  $x$  es el vector de las variables de entrada,  $S$  el conjunto de parámetros y  $F$  la función global implementada por la red adaptable. Ahora bien,  $S$  puede dividirse en dos partes:

$$S = S_1 \oplus S_2 \quad (2. 20)$$

donde

$S$  = conjunto de todos los parámetros de la red,  
 $S_1$  = conjunto de los parámetros antecedentes (Capa 1, no lineal),  
 $S_2$  = conjunto de los parámetros consecuentes (Capa 4, lineal)

Entonces, dados los valores de los parámetros de  $S_1$  se tiene que  $F$  es lineal en los elementos de  $S_2$ . Por consiguiente, dado que se conocen los valores de la salida deseada mediante un conjunto de entrenamiento, es posible aplicar alguna técnica de estimación lineal tal como el método de cuadrados mínimos [22][43], para una rápida y sencilla estimación de los parámetros consecuentes.

Por otra parte, de la arquitectura mostrada en la figura 2.7, se observa que cuando los valores de los parámetros antecedentes son fijos o constantes, la salida de la red puede expresarse como una **combinación lineal** de los parámetros consecuentes [22][24][43]. Esto es válido si se toma en cuenta que los parámetros antecedentes, pertenecientes a las funciones de membresía de los valores lingüísticos, se utilizan como el medio por el cual un experto incorpora su conocimiento y experiencia en la red inteligente. Luego entonces la salida  $y$  de la figura 2.7 puede expresarse de la siguiente forma:

$$y = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 \quad (2.21)$$

$$y = \bar{w}_1(p_1x_1 + q_1x_2 + r_1) + \bar{w}_2(p_2x_1 + q_2x_2 + r_2) \quad (2.22)$$

$$y = (\bar{w}_1x_1)p_1 + (\bar{w}_1x_2)q_1 + (\bar{w}_1)r_1 + (\bar{w}_2x_1)p_2 + (\bar{w}_2x_2)q_2 + (\bar{w}_2)r_2 \quad (2.23)$$

Como se observa  $y$  es lineal en los parámetros consecuentes  $p_1, q_1, r_1, p_2, q_2,$  y  $r_2$  pertenecientes a  $S_2$  en la ecuación (2.26). De esta manera, el problema se reduce a estimar los parámetros consecuentes mediante alguna técnica de regresión lineal que minimice el error de la red. La siguiente sección presenta como se resuelve el problema de estimación al utilizar el método de cuadrados mínimos y el filtro de Kalman como algoritmo de aprendizaje de la red neuro-difusa.

### 2.5.1. Algoritmo de Aprendizaje.

De la ecuación (2.23) se observa que la salida global de la red es una combinación lineal siempre y cuando los parámetros antecedentes sean constantes. En primer lugar, para entrenar la red se necesita un **conjunto de datos de entrenamiento** el cual usualmente se obtiene a través de experimentación. Este conjunto de entrenamiento se compone de pares de datos que representan los valores de entrada del sistema,  $x_i,$  y la salida deseada,  $d$ . Para una red con dos entradas y una salida, y utilizando  $m$  datos de entrenamiento en la forma  $(x_1^{(i)}, x_2^{(i)}; d^{(i)}), i=1, 2, \dots, m,$  se tiene:

$$\begin{bmatrix} d^{(1)} \\ d^{(2)} \\ \vdots \\ d^{(m)} \end{bmatrix} = \begin{bmatrix} (\bar{\omega}_1 x_1)^{(1)} p_1 + (\bar{\omega}_1 x_2)^{(1)} q_1 + (\bar{\omega}_1)^{(1)} r_1 + (\bar{\omega}_2 x_1)^{(1)} p_2 + (\bar{\omega}_2 x_2)^{(1)} q_2 + (\bar{\omega}_2)^{(1)} r_2 \\ (\bar{\omega}_1 x_1)^{(2)} p_1 + (\bar{\omega}_1 x_2)^{(2)} q_1 + (\bar{\omega}_1)^{(2)} r_1 + (\bar{\omega}_2 x_1)^{(2)} p_2 + (\bar{\omega}_2 x_2)^{(2)} q_2 + (\bar{\omega}_2)^{(2)} r_2 \\ \vdots \\ (\bar{\omega}_1 x_1)^{(m)} p_1 + (\bar{\omega}_1 x_2)^{(m)} q_1 + (\bar{\omega}_1)^{(m)} r_1 + (\bar{\omega}_2 x_1)^{(m)} p_2 + (\bar{\omega}_2 x_2)^{(m)} q_2 + (\bar{\omega}_2)^{(m)} r_2 \end{bmatrix} \quad (2.24)$$

$$\begin{bmatrix} d^{(1)} \\ d^{(2)} \\ \vdots \\ d^{(m)} \end{bmatrix} = \begin{bmatrix} (\bar{\omega}_1 x_1)^{(1)} & (\bar{\omega}_1 x_2)^{(1)} & (\bar{\omega}_1)^{(1)} & (\bar{\omega}_2 x_1)^{(1)} & (\bar{\omega}_2 x_2)^{(1)} & (\bar{\omega}_2)^{(1)} \\ (\bar{\omega}_1 x_1)^{(2)} & (\bar{\omega}_1 x_2)^{(2)} & (\bar{\omega}_1)^{(2)} & (\bar{\omega}_2 x_1)^{(2)} & (\bar{\omega}_2 x_2)^{(2)} & (\bar{\omega}_2)^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ (\bar{\omega}_1 x_1)^{(m)} & (\bar{\omega}_1 x_2)^{(m)} & (\bar{\omega}_1)^{(m)} & (\bar{\omega}_2 x_1)^{(m)} & (\bar{\omega}_2 x_2)^{(m)} & (\bar{\omega}_2)^{(m)} \end{bmatrix} \begin{bmatrix} p_1 \\ q_1 \\ r_1 \\ p_2 \\ q_2 \\ r_2 \end{bmatrix} \quad (2.25)$$

La ecuación (2.25) presenta un problema clásico de regresión lineal que para este caso se estima el vector de los parámetros consecuentes mediante el algoritmo del filtro de Kalman para un aprendizaje ya sea en línea o bien el método clásico de cuadrados mínimos para un aprendizaje fuera de línea [25]. La siguiente sección presenta con más detalle dichos métodos que como ya se mencionó son utilizados como el algoritmo de aprendizaje de la red.

### 2.5.1.1. Método de Estimación por Cuadrados Mínimos

En general, un problema de estimación por cuadrados mínimos es que la salida del modelo lineal  $d$ , esta dada por la expresión [8]:

$$d = \theta_1 f_1(\mathbf{x}) + \theta_2 f_2(\mathbf{x}) + \cdots + \theta_n f_n(\mathbf{x}) \quad (2.26)$$

donde  $\mathbf{x} = [x_1, x_2, \dots, x_p]^T$  es el vector de entrada del modelo,  $f_1, f_2, \dots, f_n$ , son funciones conocidas de  $\mathbf{x}$ , y  $\theta_1, \theta_2, \dots, \theta_n$  son los parámetros desconocidos que se desean estimar. Sustituyendo cada par de datos  $\{(x_i, d_i), i = 1, 2, \dots, m\}$  en la ecuación (2.26) se obtiene un conjunto de  $m$  ecuaciones lineales idénticas a (2.25):

$$\begin{aligned} f_1(x_1)\theta_1 + f_2(x_1)\theta_2 + \cdots + f_n(x_1)\theta_n &= d_1 \\ f_1(x_2)\theta_1 + f_2(x_2)\theta_2 + \cdots + f_n(x_2)\theta_n &= d_2 \\ &\vdots \\ f_1(x_m)\theta_1 + f_2(x_m)\theta_2 + \cdots + f_n(x_m)\theta_n &= d_m \end{aligned} \quad (2.27)$$

Rescribiendo las ecuaciones anteriores expresándolas en notación matricial, se tiene

$$\mathbf{A}\boldsymbol{\theta} = \mathbf{d} \quad (2.28)$$

donde  $\mathbf{A}$  es una matriz de  $m \times n$

$$\mathbf{A} = \begin{bmatrix} f_1(\mathbf{x}_1) & \cdots & f_n(\mathbf{x}_1) \\ \vdots & \vdots & \vdots \\ f_1(\mathbf{x}_m) & \cdots & f_n(\mathbf{x}_m) \end{bmatrix} \quad (2.29)$$

$\boldsymbol{\theta}$  es una matriz de  $n \times 1$  que contiene los parámetros desconocidos:

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \quad (2.30)$$

y  $\mathbf{d}$  es el vector  $m \times 1$  de salida deseado:

$$\mathbf{d} = \begin{bmatrix} d_1 \\ \vdots \\ d_m \end{bmatrix} \quad (2.31)$$

El  $i$ -ésimo renglón de las matrices  $[\mathbf{A} : \mathbf{d}]$ , denotado por  $[\mathbf{a}_i^T : d_i]$  se relaciona con el mismo renglón de los datos de entrada-salida del modelo  $(x_i, d_i)$  a través de la siguiente ecuación:

$$\mathbf{a}_i^T = [f_1(x_i), \dots, f_n(x_i)] \quad (2.32)$$

Para obtener una solución del vector  $\boldsymbol{\theta}$ , es necesario que  $m \geq n$ . Si  $\mathbf{A}$  es cuadrada ( $m=n$ ) y no singular entonces:

$$\boldsymbol{\theta} = \mathbf{A}^{-1}\mathbf{d} \quad (2.33)$$

Sin embargo, regularmente  $m$  es mayor que  $n$ , indicando que se tienen más pares de datos que parámetros. En este caso, no es posible obtener una solución exacta que

satisfaga todas las ecuaciones debido a que los datos pueden contener ruido, de tal forma que la ecuación (2.28) debe modificarse al incorporar un vector de error, tal que:

$$\mathbf{A}\boldsymbol{\theta} + \mathbf{e} = \mathbf{d} \quad (2.34)$$

Ahora, en lugar de encontrar una solución exacta a la ecuación (2.28), se pretende buscar una solución que minimice el error definido como [48]:

$$E(\boldsymbol{\theta}) = \sum_{i=1}^m (y_i - \mathbf{a}_i^T \boldsymbol{\theta})^2 = \mathbf{e}^T \mathbf{e} = (\mathbf{y} - \mathbf{A}\boldsymbol{\theta})^T (\mathbf{y} - \mathbf{A}\boldsymbol{\theta}) \quad (2.35)$$

donde  $\mathbf{e} = \mathbf{y} - \mathbf{A}\boldsymbol{\theta}$  es el vector del error producido para una selección específica de  $\boldsymbol{\theta}$ . Se puede observar que  $E(\boldsymbol{\theta})$  está en la forma cuadrática y tiene un mínimo en  $\boldsymbol{\theta} = \boldsymbol{\theta}^*$ .

El error cuadrático de la ecuación (2.35) se minimiza cuando  $\boldsymbol{\theta} = \boldsymbol{\theta}^*$ , llamado estimador de cuadrado mínimo el cual satisface la siguiente ecuación [8]:

$$\mathbf{A}^T \mathbf{A} \boldsymbol{\theta}^* = \mathbf{A}^T \mathbf{d} \quad (2.36)$$

Si  $\mathbf{A}^T \mathbf{A}$  es no singular, entonces  $\boldsymbol{\theta}^*$  es única y está dada por:

$$\boldsymbol{\theta}^* = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{d} \quad (2.37)$$

### 2.5.1.2. Filtro de Kalman.

El Filtro de Kalman es considerado como el método de estimación por cuadrados mínimos recursivo [38]. Como se mencionó en la sección anterior, la solución para el vector de parámetros se da por la ecuación (2.37) que aquí se repite por conveniencia (por simplicidad se ha eliminado \*).

$$\boldsymbol{\theta}_k = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{d} \quad (2.38)$$

donde se asume que la dimensión del renglón de  $\mathbf{A}$  y  $\mathbf{d}$  es  $k$ : así, el subíndice  $k$  es insertado para denotar el número de pares de datos usados para la estimación de  $\boldsymbol{\theta}$ . La  $k$  también puede significar la medida del tiempo si los pares de datos están disponibles en orden secuencial. Suponga que un nuevo par de datos ( $\mathbf{a}^T; d$ ) es disponible como la  $(m+1)$ -ésima entrada al conjunto de datos. Entonces en lugar de usar todos los pares de datos del conjunto para recalculer  $\boldsymbol{\theta}_{k+1}$ , se busca tomar ventaja de  $\boldsymbol{\theta}_k$  ya disponible para

obtener  $\theta_{k+1}$  con un mínimo de esfuerzo utilizando los nuevos pares de datos  $(\mathbf{a}^T; d)$ . Es obvio que  $\theta_{k+1}$  puede expresarse como:

$$\theta_{k+1} = \left( \begin{bmatrix} \mathbf{A} \\ \mathbf{a}^T \end{bmatrix}^T \begin{bmatrix} \mathbf{A} \\ \mathbf{a}^T \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{A} \\ \mathbf{a}^T \end{bmatrix}^T \begin{bmatrix} \mathbf{d} \\ d \end{bmatrix} \quad (2.39)$$

Para simplificar la notación, se introducen dos matrices de  $n \times n$ ,  $\mathbf{P}_k$  y  $\mathbf{P}_{k+1}$  definidas por:

$$\mathbf{P}_k = (\mathbf{A}^T \mathbf{A})^{-1} \quad (2.40)$$

$$\begin{aligned} \mathbf{P}_{k+1} &= \left( \begin{bmatrix} \mathbf{A} \\ \mathbf{a}^T \end{bmatrix}^T \begin{bmatrix} \mathbf{A} \\ \mathbf{a}^T \end{bmatrix} \right)^{-1} \\ &= \left( \begin{bmatrix} \mathbf{A}^T & \mathbf{a} \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{a}^T \end{bmatrix} \right)^{-1} \\ &= (\mathbf{A}^T \mathbf{A} + \mathbf{a} \mathbf{a}^T)^{-1} \end{aligned} \quad (2.41)$$

Estas dos matrices se relacionan mediante:

$$\mathbf{P}_k^{-1} = \mathbf{P}_{k+1}^{-1} - \mathbf{a} \mathbf{a}^T \quad (2.42)$$

Usando (2.40) y (2.41), se tiene:

$$\begin{cases} \theta_k = \mathbf{P}_k \mathbf{A}^T \mathbf{d} \\ \theta_{k+1} = \mathbf{P}_{k+1} (\mathbf{A}^T \mathbf{d} + \mathbf{a} d) \end{cases} \quad (2.43)$$

Para expresar  $\theta_{k+1}$  en términos de  $\theta_k$ , es necesario eliminar  $\mathbf{A}^T \mathbf{d}$  en (2.43). De (2.38) se tiene que:

$$\mathbf{A}^T \mathbf{d} = \mathbf{P}_k^{-1} \theta_k \quad (2.44)$$

Sustituyendo (2.44) en (2.43) y aplicando (2.42):

$$\begin{aligned}
\theta_{k+1} &= \mathbf{P}_{k+1}(\mathbf{P}_k^{-1}\theta_k + \mathbf{a}d) \\
&= \mathbf{P}_{k+1}\left[(\mathbf{P}_k^{-1} - \mathbf{a}\mathbf{a}^T)\theta_k + \mathbf{a}d\right] \\
&= \theta_k + \mathbf{P}_{k+1}\mathbf{a}(d - \mathbf{a}^T\theta_k)
\end{aligned} \tag{2.45}$$

Calculando  $\mathbf{P}_{k+1}$  por la ecuación (2.41) implica la inversión de una matriz cuadrada. Esto requiere un costo computacional alto. De la ecuación (2.42), se tiene:

$$\mathbf{P}_{k+1} = (\mathbf{P}_k^{-1} - \mathbf{a}\mathbf{a}^T)^{-1} \tag{2.46}$$

$$\begin{aligned}
\mathbf{P}_{k+1} &= \mathbf{P}_k - \mathbf{P}_k\mathbf{a}(\mathbf{I} + \mathbf{a}^T\mathbf{P}_k\mathbf{a})^{-1}\mathbf{a}^T\mathbf{P}_k \\
&= \mathbf{P}_k - \frac{\mathbf{P}_k\mathbf{a}\mathbf{a}^T\mathbf{P}_k}{1 + \mathbf{a}^T\mathbf{P}_k\mathbf{a}}
\end{aligned} \tag{2.47}$$

En resumen, la estimación por el filtro de Kalman para el problema de  $\mathbf{A}\theta = \mathbf{d}$ , donde el  $k$ -ésimo ( $1 \leq k \leq m$ ) renglón de  $[\mathbf{A}:\mathbf{d}]$ , denotado por  $[\mathbf{a}_k^T: d_k]$ , se obtiene secuencialmente mediante lo siguiente [8][24][38]:

$$\begin{cases} \mathbf{P}_{k+1} &= \mathbf{P}_k - \frac{\mathbf{P}_k\mathbf{a}_{k+1}\mathbf{a}_{k+1}^T\mathbf{P}_k}{1 + \mathbf{a}_{k+1}^T\mathbf{P}_k\mathbf{a}_{k+1}} \\ \theta_{k+1} &= \theta_k + \mathbf{P}_{k+1}\mathbf{a}_{k+1}(d_{k+1} - \mathbf{a}_{k+1}^T\theta_k) \end{cases} \tag{2.48}$$

$$\theta^* = \theta_m \tag{2.49}$$

con condiciones iniciales de:

$$\theta_0 = 0 \quad \text{y} \quad \mathbf{P}_0 = \gamma\mathbf{I} \tag{2.50}$$

donde  $\gamma$  es un número positivo grande e  $\mathbf{I}$  es la matriz identidad de dimensión  $m \times m$ .

El algoritmo de adaptación mostrado en (2.48) trabaja bien en sistemas invariantes en el tiempo, sin embargo para sistemas variantes puede ocasionar problemas de convergencia. Para resolver esto, es apropiado resetear la matriz  $\mathbf{P}_k$  a sus condiciones iniciales  $\mathbf{P}_0$  ocasionalmente, quizás cuando se sospeche que ya ha ocurrido un cambio significativo en los parámetros [38].



Cabe mencionar que es posible aplicar algún algoritmo basado en la obtención del gradiente del error y al propagar las señales del error en sentido contrario al sentido de la red, es decir regresivo, se encuentran los valores de los parámetros antecedentes vía el método de retropropagación [26][29]. La tabla 2.1 resume los procedimientos del aprendizaje que pueden ser aplicados en la red neuro-difusa.

**Tabla 2. 1.** *Procedimientos de aprendizaje y adaptación del ANFIS*

	APRENDIZAJE	
	Progresivo (forward)	Regresivo (backward)
Parámetros antecedentes	Constantes	Retro-propagación
Parámetros consecuentes	Filtro de Kalman	Constantes
Señales utilizadas	Salidas de los nodos	Señales de error

Como se mencionó anteriormente, los parámetros consecuentes así identificados son óptimos desde el punto de vista de la minimización del error y bajo la condición de que los parámetros antecedentes estén fijos. De acuerdo con esto la red se vuelve menos compleja y el aprendizaje se lleva a cabo con más rapidez. Por otra parte, los valores de los parámetros antecedentes no son totalmente arbitrarios. Volviendo al principio de la inteligencia artificial que gobierna esta red, la experiencia y el conocimiento de un experto en el proceso define los valores de dichos parámetros permitiendo una interacción entre el humano con el razonamiento e inteligencia del sistema.

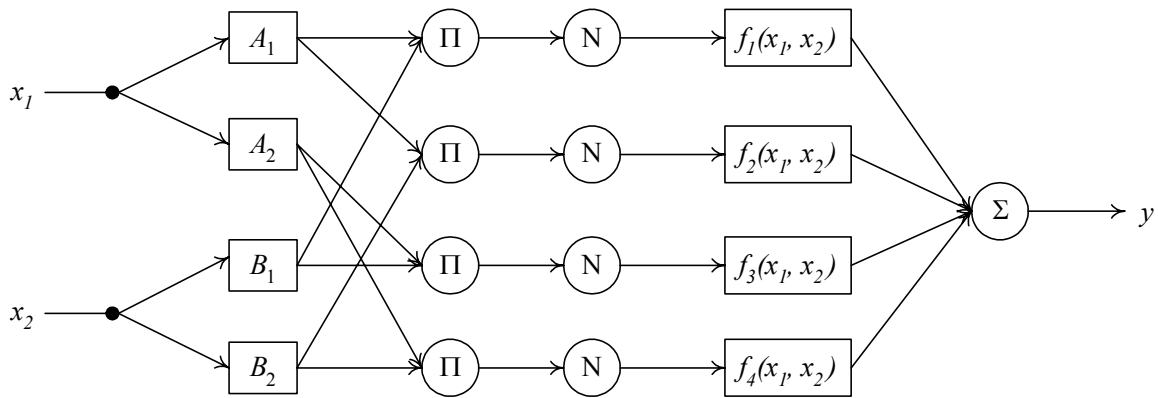
### 2.5.2. Identificación de un Sistema No Lineal.

Para comprobar el potencial de la estructura inteligente desarrollada hasta esta parte, se procede a realizar una simulación digital donde se utiliza el ANFIS para identificar un sistema con altas no linealidades.

La estructura neuro-difusa se muestra en la figura 2.11 la cual consta de 2 entradas, 1 salida, 2 funciones de membresía tipo campana asociadas a cada entrada, 4 reglas difusas, 12 parámetros antecedentes y 12 consecuentes. Esta estructura tiene diversas aplicaciones en sistemas de identificación y estimación así como en sistemas de control en ingeniería.

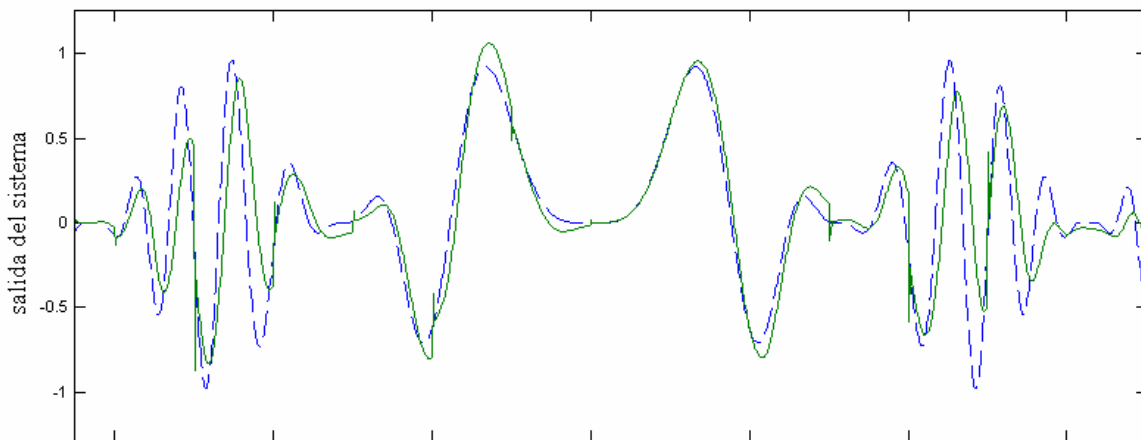
En el siguiente experimento, se aplica la red adaptable para modelar el sistema no lineal de dos entradas y una salida siguiente [16]:

$$y = f(x_1, x_2) = \frac{\sin(x_1)\sin(x_2)}{x_1x_2 \sin(x_1x_2)} \quad (2. 51)$$



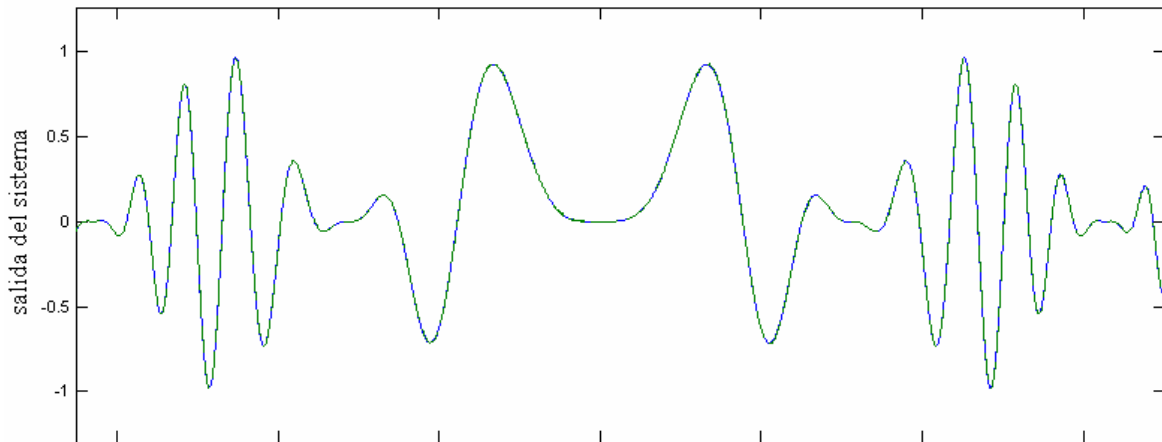
**Figura 2. 11.** Modelo neuro-difuso adaptable con 2 entradas, 1 salida, 2 funciones de membresía por entrada, 4 reglas difusas, 12 parámetros antecedentes y 12 consecuentes.

Los parámetros se adaptan iterativamente mediante el filtro de Kalman considerado anteriormente. Del modelo del sistema de la ecuación (2.51) se obtuvieron 250 datos de entrenamiento. La figura 2.12 muestra la identificación de la red con 250 datos de entrenamiento. A pesar de las altas no linealidades, la red tiene la capacidad de aproximar el modelo real.



**Figura 2. 12.** Salida del sistema no lineal (línea discontinua) y de la red adaptable (línea continua). Identificación con 250 datos de entrenamiento. Eje x: 40 muestras por división.

Para obtener mayor exactitud en el modelo, se utilizan ahora 1500 datos de entrenamiento. En la figura 2.13 es difícil diferenciar las dos funciones.



**Figura 2. 13.** Salida del sistema no lineal (línea discontinua) y de la red adaptable (línea continua). Identificación con 1500 datos de entrenamiento. Eje x: 250 muestras por división.

Los resultados que arrojan las simulaciones realizadas al sistema muestran el potencial que tienen las redes Neuro-Difusas adaptables al ser utilizadas como medio de identificación y modelado de señales a pesar de las altas no linealidades. Esta característica pone a los controladores inteligentes por encima de las técnicas convencionales que presentan dificultades en el manejo de sistemas no lineales. Otra ventaja de los sistemas inteligentes como es que no es necesario contar con un modelo matemático complicado ya que la técnica de identificación y estimación considerada se fundamenta en métodos básicos de cálculo que al mismo tiempo son sencillos de implementar en simulaciones digitales.

En el siguiente capítulo se presenta el análisis y la aplicación del modelo neuro-difuso operando como controlador, que forma parte del proceso de validación del modelo obtenido, así como también es una de las funciones en las que este tipo de estructuras trabaja con mayor frecuencia.

## Capítulo 3

# Control Neuro-Difuso

### 3.1. Introducción.

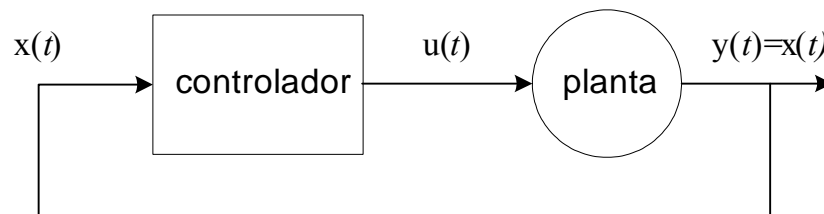
Durante las últimas décadas, el control difuso ha surgido como una de las áreas más activas y fructíferas en la aplicación de la lógica difusa en los sistemas de control, especialmente para automatizar procesos industriales cuyos modelos matemáticos son altamente no lineales. Los sistemas de control difusos se basan en las relaciones e inferencia de la lógica difusa la cual se relaciona con los procesos del razonamiento humano y su correspondiente lenguaje natural. Por esta razón, un controlador basado en lógica difusa provee los medios para convertir una estrategia de control lingüístico operado por la experiencia y conocimiento humano en una estrategia de control automático. Además, las técnicas difusas proveen una metodología para representar, manipular e implementar el conocimiento heurístico del ser humano acerca de cómo controlar un sistema [21].

Por otra parte, la necesidad de encontrar mejores métodos de controlar sistemas dinámicos bajo condiciones de alta incertidumbre ha hecho que el uso de las redes neuronales se convierta en una alternativa clara gracias a sus habilidades de identificación y aproximación de funciones con altas no linealidades. Como ya se mencionó, la combinación de las redes neuronales con los sistemas de razonamiento difuso como un ejemplo de un sistema adaptable permite la transmisión de los conocimientos y experiencia de un experto para diseñar y desarrollar un sistema de control inteligente con cierta facilidad.

En este capítulo se presenta la estrategia del control neuro-difuso empleada para la aplicación de un controlador inteligente dentro del esquema del control directo del par para el motor de inducción. En primer lugar se consideran los esquemas de control difuso que imitan el comportamiento de controles convencionales. Posteriormente el uso de la estructura neuro-difusa como controlador utilizando el aprendizaje inverso.

### 3.2. Sistemas de Control Retroalimentados.

El empleo de la retroalimentación en un sistema de control puede ser una parte importante para que una estrategia de control tenga éxito. La presencia de retroalimentación ofrece la posibilidad de modificar sustancialmente el comportamiento del sistema y a menudo una posible variación en sus parámetros se muestra con resultados dramáticos [36][46]. La figura 3.1 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 3. 1.** Diagrama de bloques de un sistema de control retroalimentado.

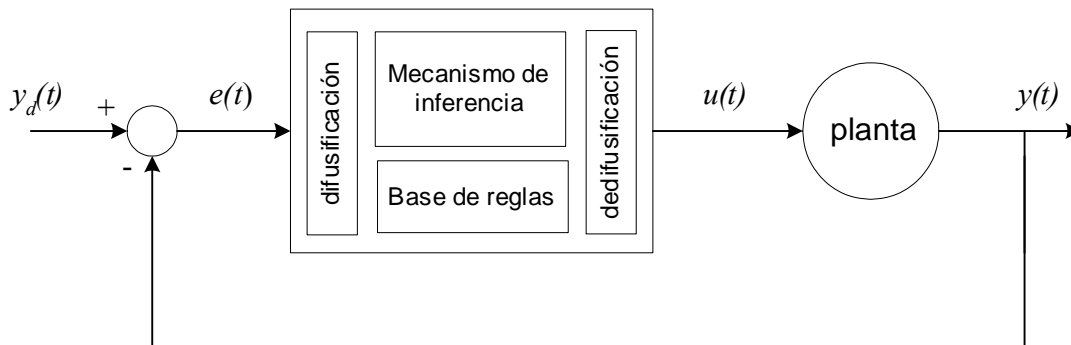
La ecuación en espacio de estado para una planta no lineal se expresa de la siguiente forma matricial:

$$y(t) = f(x(t), u(t)) \quad (3. 1)$$

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 [27][65], es deseable hacer que  $\Phi(\cdot)$  sea adaptable para que el controlador responda de una manera favorable ante las características cambiantes de la planta [35].

### 3.3. Sistemas de Control Basados en Lógica Difusa.

Si se reemplaza el bloque del controlador en la figura 3.1 por un FIS como el considerado en la figura 2.1, el sistema se convierte en un Sistema de Control Difuso (FLC, por sus siglas en inglés *Fuzzy Logic Controller*) como el de la figura 3.2. El objetivo primordial de la ingeniería de control difuso es la de extraer y aplicar el conocimiento acerca de cómo funciona un proceso de tal forma que el sistema de control diseñado garantice un desempeño confiable y eficaz minimizando el error global entre la salida actual y la salida de referencia [48]. En la figura 3.2 se observa que la entrada al FIS es el error en el instante  $k$ .



**Figura 3. 2.** Sistema de control difuso.

La idea básica detrás del FLC es la de incorporar la experiencia de un operador humano en el diseño del controlador para un proceso determinado, cuya relación de señales de entrada-salida se describen mediante una colección de **reglas de control**, involucrando variables lingüísticas en lugar de un modelo matemático [29][34].

Un FLC puede definirse como una estructura computacional basada en los conceptos de la teoría de los conjuntos difusos, reglas difusas y razonamiento difuso. Debido a su naturaleza multidisciplinaria, también se les conoce como *sistemas basado en reglas difusas*, *sistemas expertos difusos*, *modelos difusos*, *memoria asociativa difusa*, etc [24].

Ahora bien, en la figura 3.2 se puede observar que para el diseño del FLC es necesario establecer los parámetros adecuados para cada componente del controlador.

Esto incluye:

1. Definición de las variables de entrada y salida.
2. Definición de la partición difusa de la entrada y salida así como escoger las funciones de membresía para las variables lingüísticas del sistema.
3. Definición de las reglas difusas y la manera de su interrelación.
4. Definición del mecanismo de razonamiento, incluyendo el método de implicación y la aplicación de las reglas difusas.
5. Definición del método de dedifusificación.

Los dos primeros principios de diseño indican que se debe identificar el proceso de las variables de estado del sistema y las variables de control, así como también determinar un conjunto de términos que indiquen el nivel de ambigüedad para las variables lingüísticas. Por ejemplo, tres conjuntos tales como BAJO, MEDIO, ALTO pueden representar los niveles de las variables del sistema, sin embargo en algunos casos esta partición puede no satisfacer ciertos dominios por lo que será necesario ampliarla y utilizar cinco conjuntos difusos tales como MUY BAJO, BAJO, MEDIO, LARGO y MUY LARGO. El número de particiones de cada conjunto difuso debe ser del tamaño suficiente para proveer una aproximación adecuada del valor de membresía del elemento y al mismo tiempo no requerir demasiado espacio en memoria y tiempo de cómputo. Cabe mencionar que si bien es cierto que no existe un método determinado para conocer el número óptimo de particiones, este aspecto tiene un efecto esencial en la fineza con la que el controlador opera en el sistema [13].

Para este caso, las variables de estado del lazo de retroalimentación representan el contenido de la parte antecedente de las reglas difusas. Las variables pueden ser:

- ✓ Error, denotado por  $e(t)$ ,
- ✓ Cambio del error, denotado por  $\Delta e(t)$ ,
- ✓ Suma del error, denotado por  $\delta e(t)$ .

Por otra parte, la variable de salida del controlador, representa la parte consecuente de las reglas difusas. La variable puede ser:

- ✓ Salida del controlador, denotado por  $u(t)$ ,
- ✓ Cambio de la salida del controlador, denotado por  $\Delta u(t)$ .

Además, por analogía de un controlador convencional, se tiene que

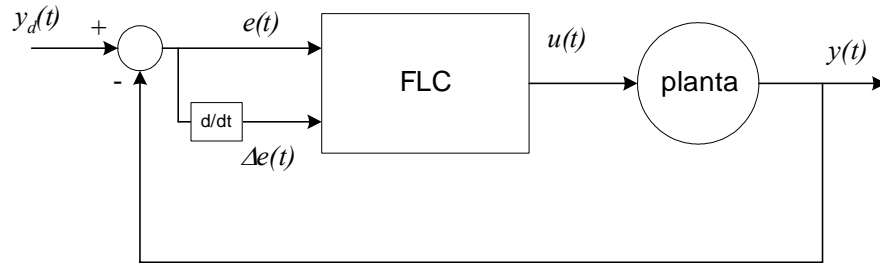
$$e(t) = y_d(t) - y(t) \quad (3.2)$$

$$\Delta e(t) = e(t) - e(t-1) \quad (3.3)$$

$$\Delta u(t) = u(t) - u(t-1) \quad (3.4)$$

donde  $y_d(t)$  es la señal de referencia de entrada al controlador. Debido a que en este caso el sistema es dinámico, las señales están en función del tiempo.

Lo anterior da la pauta para la figura 3.3, donde se muestra la relación de las variables de entrada y salida.

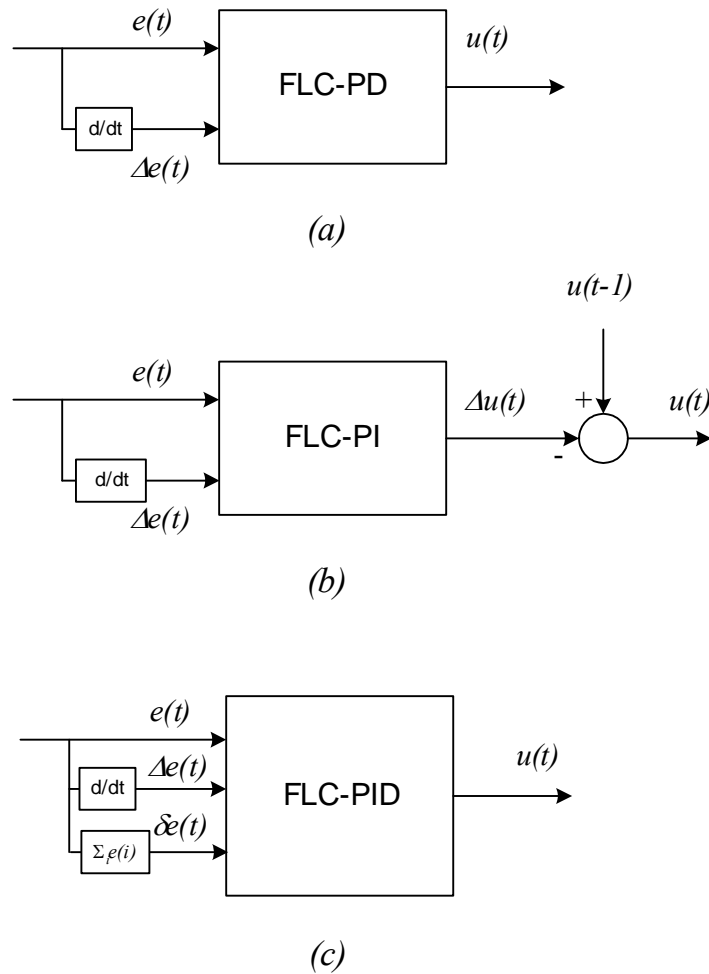


**Figura 3. 3.** *FLC para un sistema dinámico.*

Para seleccionar el tipo de función de membresía a utilizar y las particiones de las variables lingüísticas, existen dos métodos. En el primero, se utiliza la experiencia y el conocimiento adquirido del proceso para seleccionar la función más apropiada que vaya a representar las variables de entrada y salida y luego, mediante procedimientos heurísticos encontrar la partición más adecuada y aproximada [13]. Aunque este método consume tiempo, ha sido utilizado ampliamente y empleado en muchas aplicaciones industriales. En lo que respecta al segundo método, es posible utilizar técnicas de aprendizaje y auto-adaptación. La idea concreta en este caso es que el controlador seleccione automáticamente o dinámicamente las particiones más convenientes y útiles.

Por otro lado, no se puede pasar por alto mencionar que actualmente se han desarrollado controladores difusos que representan analogías a los controladores convencionales [13][48]. Es posible realizar el diseño de un FLC por analogía tipo PD, PI o PID, lo cual indica que la selección adecuada de las variables es muy importante. La figura 3.4 presenta los esquemas de dichos controladores. No obstante aunque la aplicación de la inteligencia artificial a este tipo de analogías con los controladores convencionales proporciona un mejor desempeño por su dominio de control no lineal, es importante señalar que conservan los mismos problemas inherentes de los convencionales. Por ejemplo, en la figura 3.4(a) se muestra el esquema del controlador difuso tipo PD. Como es bien sabido, la acción derivativa causa la amplificación del ruido en la señal de control, así como también no proporciona una acción de control que compense el error en estado estacionario. Pues bien, estos problemas también ocurren con el FLC-PD.





**Figura 3. 4.** Controladores difusos por analogía con los convencionales. (a) PD; (b) PI; (c) PID..

### 3.4. Controlador Neuro-Difuso.

Esta sección reúne las herramientas de razonamiento y aprendizaje desarrolladas en las secciones pasadas correspondientes a la utilización de la lógica difusa en el control de sistemas y a los algoritmos de aprendizaje y adaptación respectivamente en una sola estructura que servirá para el diseño del controlador. El controlador neuro-difuso (NFC, por sus siglas en inglés *Neuro-Fuzzy Controller*) se basa en la fusión del control difuso con las redes neuronales adaptables. Como consecuencia esta estrategia posee las características de ambas metodologías: el aprendizaje, optimización y estructuras flexibles de las redes neuronales con el razonamiento y pensamiento del control difuso.

Como se mencionó en el capítulo anterior, los sistemas de razonamiento difuso y las redes neuronales son ejemplos especiales de un marco computacional general y que en combinación forman redes adaptables. Sin embargo, es importante mencionar que los FIS son sistemas superiores a las estructuras básicas de NN en la manera en que representa el conocimiento humano mientras que estas últimas operan básicamente como una caja negra [24]. Como resultado de esto, es posible identificar las propiedades únicas de los sistemas que combinan ambas tecnologías:

1. Habilidad de aprendizaje
2. Operación en paralelo
3. Representación estructurada del conocimiento
4. Mejor integración con otros métodos de control
5. Flexibilidad en su estructura.

Es sabido que las NN también tienen las propiedades 1 y 2, pero no la 3 ni la 4. No obstante, es interesante observar que cuando una estructura neuro-difusa está siendo operada como controlador, esta se comporta más bien como una NN aunque con propiedades de razonamiento [29]. Es por eso que la mayoría de las estrategias de diseño con NFC se derivan directamente de los métodos de diseño con NN. Por otra parte, por ser controladores no lineales, usualmente contienen un gran número de parámetros lo que los hace más versátiles en controlar sistemas no lineales que los controladores lineales convencionales.

Para construir un NFC, el primer paso es la adquisición de conocimiento, el cual proviene de la experiencia de un operador humano acerca de cómo controlar el sistema y generar así el conjunto de reglas difusas, mismo que , como ya se mencionó, es el corazón de la parte difusa del controlador [35]. Es posible obtener dos tipos de información del operador humano: *información lingüística* e *información numérica*.

La primera tiene que ver con la manera en que el operador toma las decisiones para conducir al sistema a un estado deseado, es decir el proceso de razonamiento que realiza para llevar cabo las acciones de control. El resumen de esta información es la base para establecer las reglas difusas. Por otra parte, la información numérica tiene que ver más con la operación física del sistema a ser controlado. Esta información es obtenida al registrar las variables de entrada del sistema y su correspondiente salida bajo distintas condiciones de operación [13]. Esta información es útil para formar el conjunto de datos de entrenamiento para el NFC.

Una vez organizada esta información en la red, se utiliza la información numérica junto con el algoritmo de aprendizaje para definir las funciones de membresía y los parámetros de la red. En otras palabras, la información lingüística se emplea para el diseño de la parte difusa del controlador, mientras que la información numérica se aplica para la definición de los parámetros de la parte neuronal de la red. A continuación se considera, la estrategia utilizada en este trabajo sobre como utilizar la información lingüística y numérica para operar la red neuro-difusa como controlador.

### 3.4.1. Aprendizaje Inverso.

El desarrollo de este tipo de estrategia de control para el NFC envuelve dos fases. La primera es la *fase de entrenamiento* donde se utiliza la red neuro-difusa para identificar el modelo inverso de la planta. Con el modelo obtenido, la red se utiliza ahora para generar las acciones de control operando en la *fase de aplicación*. Si se decide que el aprendizaje sea en línea entonces estas dos fases pueden llevarse a cabo simultáneamente [44].

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

$$x(k+1) = f(x(k), u(k)) \quad (3.5)$$

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 instante  $k$ . De manera similar, la salida en el instante  $k+2$  puede escribirse:

$$x(k+2) = f(x(k+1), u(k+1)) = f(f(x(k), u(k)), u(k+1)) \quad (3.6)$$

en general se tiene:

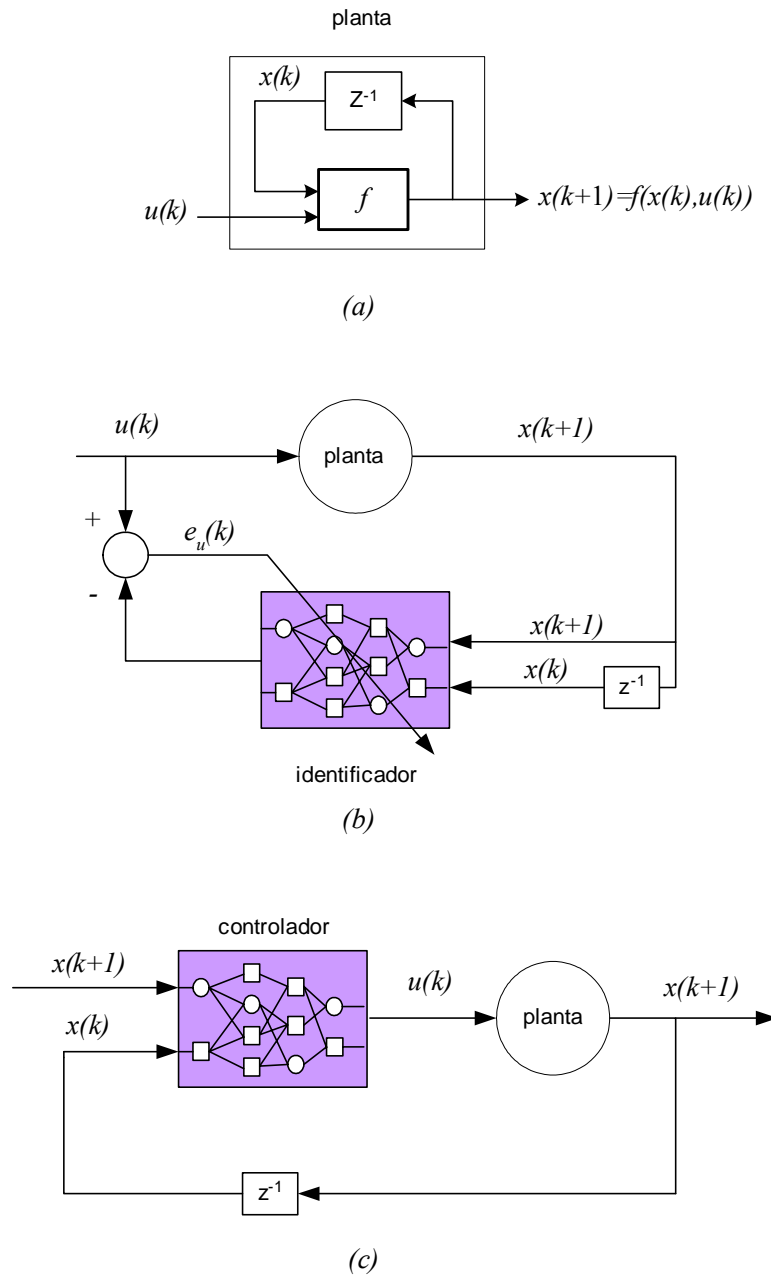
$$x(k+n) = F(x(k), U) \quad (3.7)$$

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  $x(k+n)$ :

$$U = G(x(k), x(k+n)) \quad (3.8)$$

Esta ecuación esencialmente dice que existe una única secuencia de entrada  $U$ , especificada al transformar  $G$ , que puede conducir la planta desde la salida  $x(k)$  hasta  $x(k+n)$  en  $n$  pasos de tiempo. El problema entonces es encontrar la función  $G$ , misma que será aproximada al utilizar la red neuro-difusa para identificarla de manera inversa [24]. La figura 3.5 ilustra la lo anterior cuando  $n$  vale 1.

La figura 3.5(a) muestra la planta, cuya salida  $x(k+1)$  esta en función del previo estado  $x(k)$  (gracias al bloque de retardo  $z^{-1}$ ) y la entrada  $u(k)$ ; la figura 3.5(b) es el diagrama de bloques durante la fase de entrenamiento; la figura 3.5(c) es el diagrama de bloque durante la fase de aplicación.

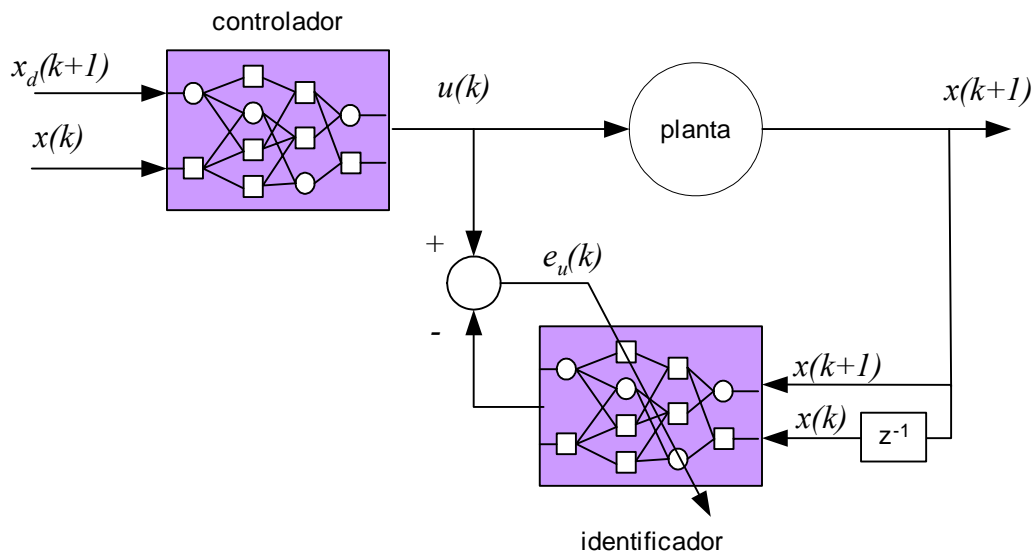


**Figura 3. 5.** Diagramas de bloque para el aprendizaje inverso. (a) planta; (b) fase de entrenamiento; (c) fase de aplicación.

La figura anterior muestra como la red neuro-difusa se utiliza 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, que como ya se

mencionó, en este trabajo se utiliza el filtro de Kalman. Una vez que la red ha identificado el modelo inverso de la planta, ahora se emplea como controlador [38]. La señal de referencia indica la salida deseada de la red y se utiliza como entrada del NFC así como el estado actual de la planta. La salida del NFC 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.

En caso de utilizar el NFC cuyo aprendizaje sea fuera de línea, es necesario recolectar el conjunto de pares de entrenamiento mediante utilizar valores aleatorios como entrada y observar la salida correspondiente de la red. El dominio de los valores de entrenamiento corresponderá posteriormente al dominio del controlador. Por otra parte, si el aprendizaje de la red es efectuado en línea con la planta, las acciones de control se generan en cada instante de tiempo. Esto es una técnica ventajosa cuando se tratan sistemas variantes en el tiempo. Sin embargo, cabe mencionar que el aprendizaje en línea resulta en mayor tiempo de cómputo así como también elevar la complejidad del sistema de control [24]. La figura 3.6 presenta el esquema para el NFC con aprendizaje inverso en línea. Los bloques de las estructuras neuro-difusas conservan idénticas características.



**Figura 3. 6.** Esquema del control neuro-difuso con aprendizaje inverso en línea.

### 3.4.2. Procedimiento para el Control Neuro-Difuso.

Utilizando las herramientas y técnicas de estimación presentadas hasta esta parte de la tesis, sería útil enlistar los pasos a seguir para llevar a cabo el control de un sistema a través de la red Neuro-Difusa propuesta en este trabajo. Cabe mencionar que este procedimiento originalmente fue propuesto para el control con NN [38][44], sin embargo como ya se abordó en el capítulo anterior, las redes Neuro-Difusas son una extensión de NN, sin embargo son superiores a estas desde el punto de vista que cuentan con razonamiento. El cuadro siguiente muestra el procedimiento para el control Neuro-Difuso.

1. 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 2.10)
2. Establecimiento de la estructura de la red Neuro-Difusa en lo referente al número de entradas y salidas, número y tipo de funciones de membresía, tipo de modelo de razonamiento difuso y reglas difusas, etc. (Figura 2.7, sección. 2.4.2)
3. Interacción experto-controlador donde se transmiten los conocimientos de cómo controlar el sistema. Esto se refleja específicamente en la parte antecedente de las reglas difusas. El experto tendrá que seleccionar los valores de los parámetros antecedentes (parámetros de las funciones de membresía), mismos que permanecerán constantes durante todo el proceso de aprendizaje y control. Los parámetros consecuentes inicialmente son cero.
4. 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. (Figura 3.5(a) y figura 3.6, sección 2.5.1)
5. Aplicación de la red en el esquema de control. Usando el modelo inverso de la planta, la red Neuro-Difusa como controlador será capaz de generar las acciones de control que lleven a la planta al estado deseado. (Figura 3.5, sección 3.4.1)

Como se puede observar, los pasos mencionados anteriormente concuerdan bien con el diagrama de bloques que representa un proceso de identificación mostrado en la figura 2.9. Esto nos conduce a pensar que en realidad el control neuro-difuso se reduce a una identificación de procesos junto con una validación que en este caso se presenta

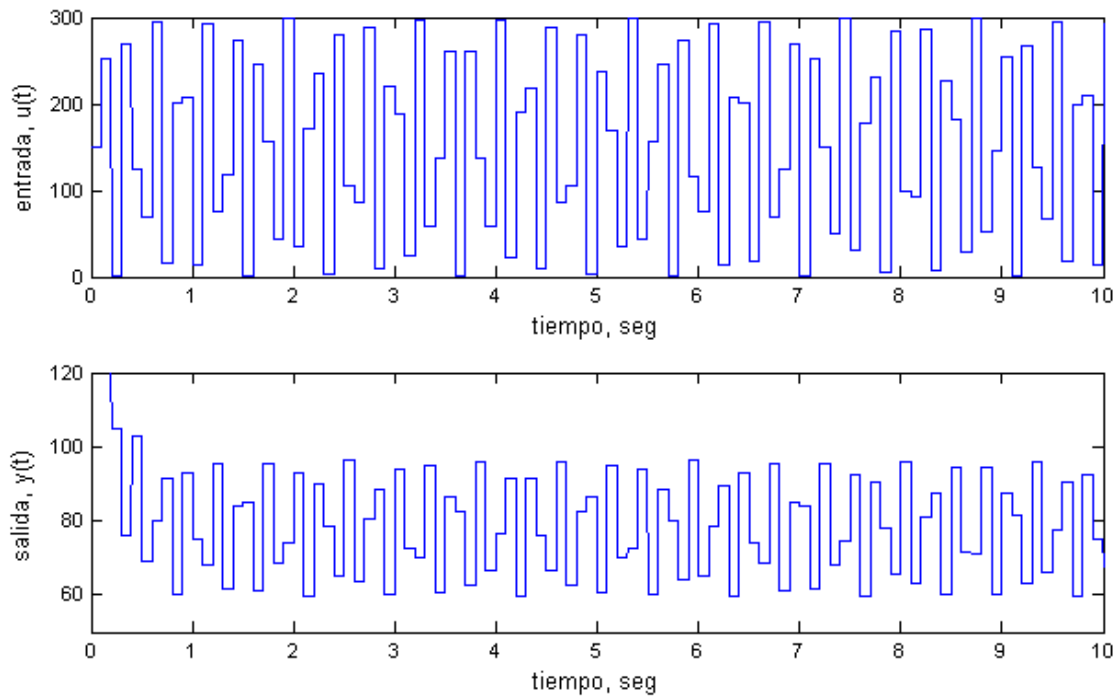
en la fase de control. Esto permite que las redes Neuro-Difusas puedan emplearse en otras aplicaciones de ingeniería.

Por otra parte, 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 aplicación el controlador opere en las mismas condiciones con las que fue entrenado. Esto quiere 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 y así por el estilo.

Para comprobar el procedimiento presentado, la figura 3.7 muestra la fase de experimentación de un sistema dado por el siguiente conjunto de ecuaciones diferenciales:

$$\begin{aligned} \dot{x}_1(t) &= -2x_1(t) + 2 \\ \dot{x}_2(t) &= -150x_1(t)x_3(t) - 50x_2(t) + 83.33u(t) \\ \dot{x}_3(t) &= 1.8x_1(t)x_2(t) - 29.2 \\ y(t) &= x_3(t) \end{aligned} \quad (3.9)$$

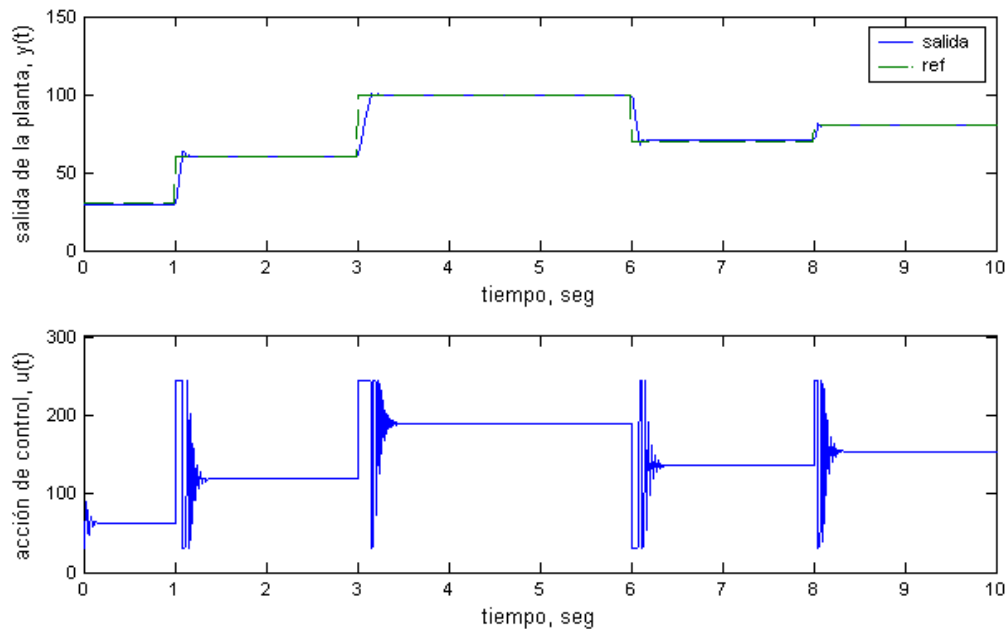
donde  $x_1(t)$ ,  $x_2(t)$  y  $x_3(t)$  son las variables de estado,  $u(t)$  es la entrada de la planta y  $y(t)$  su salida.



**Figura 3. 7.** Fase experimental del control neuro-difuso.

Como se observa en la figura anterior, en la fase experimental se varía de forma aleatoria la entrada de la planta que corresponde en realidad a la señal que generará el controlador en su momento. El impacto de dicha variación se mide en la salida de la planta para obtener el conjunto de datos de entrenamiento.

La estructura utilizada en este experimento es similar a la de la figura 2.11. Una vez aplicado el algoritmo de entrenamiento fuera de línea, en este caso el filtro de Kalman, la red obtenida es entonces aplicada como controlador del sistema. La figura 3.8 muestra el desempeño del sistema a diferentes cambios en la señal de referencia, así como el comportamiento del NFC.



**Figura 3. 8.** Fase de validación o aplicación del modelo inverso identificado. Se muestra el desempeño del NFC en el control del sistema.

El esquema de control utilizado es el de la figura 3.5 donde en la fase de aplicación las entradas del NFC es el estado actual del sistema y el estado de referencia, mientras que la salida del controlador es la acción de control que entra a la planta. En la figura 3.8 se observa como el NFC, después de aproximar el modelo inverso de la planta, genera la acción de control de tal forma que la salida de la planta sigue la referencia. Por otra parte se observa que el error en estado estacionario tiende a cero, lo que indica que el controlador no necesita un integrador gracias a la estrategia de control utilizada.



Hasta esta parte del trabajo ya se han considerado las bases teóricas y los experimentos a base de simulaciones digitales suficientes para esperar resultados satisfactorios en la aplicación de una estructura neuro-difusa para la identificación y control de un motor de inducción tipo jaula de ardilla bajo el esquema de control directo del par. El capítulo siguiente considera este apartado.

## Capítulo 4

# Control Neuro-Difuso Directo de Par y Flujo Magnético

### 4.1. Introducción.

Los avances tecnológicos en los dispositivos de electrónica de potencia y microprocesadores han hecho posible el mejoramiento de distintas técnicas de control en aplicaciones con motores de inducción y accionamientos eléctricos [11][42][60]. En años recientes la fabricación de tales dispositivos utilizando métodos de Control de Campo Orientado (FOC, por sus siglas en inglés *Field Oriented Control*) ha dominado el mercado ya que ofrecen desempeños dinámicos similares a los obtenidos en motores de corriente directa en lo relacionado con el control de par electromagnético. Sin embargo la complejidad de esos métodos junto a otras desventajas han dado lugar a otras técnicas de control las cuales no sólo reducen los esquemas del FOC sino también que garantizan una operación confiable y eficaz [20][40][62]. Entre estas nuevas técnicas se encuentra el Control Directo de Par (DTC, por sus siglas en inglés *Direct Torque Control*).

Es bien sabido que el concepto básico del DTC es la de controlar tanto el flujo magnético del estator como el par electromagnético producido por el motor de inducción de manera simultánea. Como ya se ha demostrado en la literatura, el par y el flujo de un motor de inducción controlado por un accionamiento basado en el DTC son controlados sin el uso de lazos de corriente [56][62]. Aunado a esto, el DTC no requiere la transformación de las variables de la máquina entre el marco de referencia estacionario y el síncrono (ver apéndice A) tal como es el caso en los métodos del FOC [47].

En aplicaciones con accionamientos de motores de inducción, en muchas ocasiones es deseable o hasta primordial el control del par electromagnético donde quizá la velocidad no sea una variable preocupante. Un ejemplo de esto son los sistemas de tracción donde existe una señal de par deseada proporcionada directamente por un

operador. Para este tipo de aplicaciones el DTC es una estrategia de control ventajosa ya que para el control del par y flujo no se necesita información sobre la velocidad [60].

En este capítulo se utiliza el esquema del DTC para establecer el concepto del Control Neuro-Difuso Directo de Par (DTNFC) y su esquema de control de par y flujo magnético del motor de inducción tipo jaula de ardilla. El esquema propuesto presenta un inversor fuente de voltaje controlado por una estructura neuro-difusa adaptable la cual proporciona la señal de control para la apertura y cierre de los IGBT's del inversor y de esta forma proveer al estator del motor de inducción el voltaje apropiado para un rápido control del par y el flujo magnético del estator. Se presentan simulaciones bajo distintas condiciones de operación a través de programas digitales elaborados en lenguaje C++ (vea apéndice D).

La justificación por la cual en este trabajo se utiliza este lenguaje de programación tiene que ver principalmente por los intereses académicos del autor, que si bien es cierto que existen otras plataformas que proveen mayores herramientas para la simulación (como por ejemplo MATLAB-Simulink), se cree que en esta etapa de formación profesional, el desarrollar todo lo necesario para la elaboración de este trabajo resulta en mayores satisfacciones.

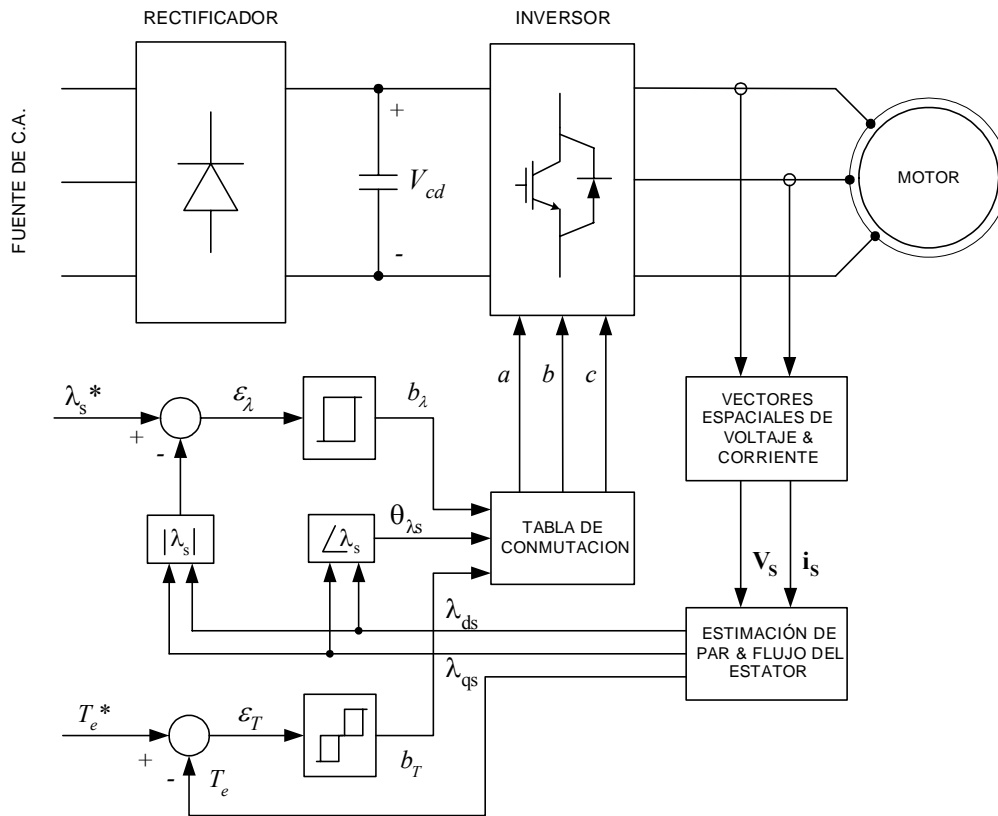
## 4.2. Esquema Clásico del Control Directo del Par.

Los primeros accionamientos basados en el Control Directo del Par para motores de inducción fueron desarrollados hace más de 10 años. Sin embargo, en la actualidad la compañía ABB ha sido la primera en comercializar esta tecnología. La figura 4.1 muestra el esquema general del DTC, creado en la década de los 80's por Depenbrock [11] y Takahashi [56].

Las premisas y principios básicos de esta estrategia pueden resumirse como sigue (ver anexo B) [60]:

- ✓ El flujo del estator es la integral con respecto al tiempo de la fuerza electromotriz del estator, de tal forma que su magnitud depende fuertemente del voltaje aplicado al estator.
- ✓ El par electromagnético desarrollado por el motor es proporcional al seno del ángulo entre los vectores espaciales del flujo del estator y el flujo del rotor.
- ✓ El efecto del flujo magnético del rotor ante cambios en el voltaje aplicado al estator es más lento que el producido en el flujo del estator.

De lo anterior se desprende que tanto la magnitud del flujo del estator y el par electromagnético producido puede controlarse mediante la adecuada selección de estado del inversor, representados por los vectores espaciales de voltaje.



**Figura 4. 1.** Esquema clásico del Control Directo de Par (DTC).

En la figura anterior se observa que no es necesaria la utilización de un sensor de velocidad para el control del par. En este caso es necesario medir las corrientes  $i_a$  e  $i_b$  así como los voltajes  $v_a$  y  $v_b$  para transformar las variables de la máquina y referirlas al marco de referencia estacionario y de esta forma determinar los vectores de voltaje del estator  $V_s$  y el vector de corriente  $i_s$  necesarios para la estimación del vector de flujo del estator  $\lambda_s$  y el par electromagnético  $T_e$  producido en ese preciso instante. De acuerdo con el modelo del motor de inducción la estimación del par y el flujo puede realizarse a través de las siguientes ecuaciones (vea apéndice A) [5]:

$$\lambda_{qs} = \int (v_{qs} - r_s i_{qs}) dt \quad (4. 1)$$

$$\lambda_{ds} = \int (v_{ds} - r_s i_{ds}) dt \quad (4.2)$$

donde

$$v_{qs} = \frac{1}{3} (2v_{as} - v_{bs} + v_{cs}) \quad (4.3)$$

$$v_{ds} = \frac{1}{\sqrt{3}} (v_{cs} - v_{bs}) \quad (4.4)$$

Pero como el sistema es balanceado  $v_{cs} = -(v_{as} + v_{bs})$ , por lo tanto  $v_{qs}$  puede ser calculado solo con los valores de dos fases del motor. Algo similar ocurre con el cálculo de  $i_{qs}$  e  $i_{ds}$ . [17][30] La magnitud y el ángulo del vector del flujo magnético del estator está dado por:

$$|\lambda_s| = \sqrt{\lambda_{qs}^2 + \lambda_{ds}^2} \quad (4.5)$$

$$\theta_{\lambda_s} = \tan^{-1} \frac{\lambda_{qs}}{\lambda_{ds}} \quad (4.6)$$

Finalmente el par electromagnético puede estimarse mediante [31][47]:

$$T_e = \lambda_{ds} i_{qs} - \lambda_{qs} i_{ds} \quad (4.7)$$

La magnitud del vector de flujo se compara dentro del lazo de control de flujo con la referencia  $\lambda_s^*$ . De igual forma el par electromagnético  $T_e$  es comparado con la señal de referencia  $T_e^*$  dentro de su lazo de control.

Los errores de flujo y par,  $\varepsilon_\lambda$  y  $\varepsilon_T$ , son aplicados a sus respectivos controladores de histéresis. La señal de salida del controlador del flujo,  $b_\lambda$ , puede tomar los valores de 0 y 1, mientras que la salida del controlador del par,  $b_T$  toma los valores de -1, 0 y 1. Estos valores son utilizados junto con el ángulo del vector del flujo del estator para seleccionar el vector de voltaje apropiado que será aplicado al motor de inducción [60][62]. La figura 4.2 muestra la distribución en el plano  $dq$  de los vectores de voltaje disponibles. Cabe recordar que cada vector de voltaje representa un estado de conmutación del inversor fuente de voltaje (ver apéndice C). Usualmente, en el esquema original del DTC esta selección se realiza a través de una **tabla de conmutación**. [6][40]

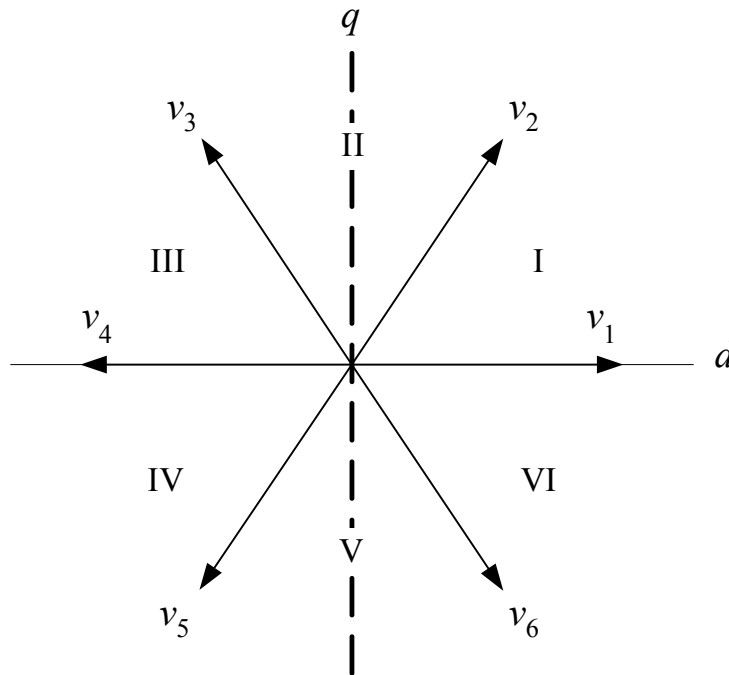
El  $k$ -ésimo ( $k = \text{I, II, III, IV, V, VI}$ ) vector de voltaje está dado por:

$$v_k = V_{cd} e^{j\theta_{v,k}} \quad (4.8)$$

donde  $V_{cd}$  es el voltaje de corriente directa aplicado al inversor, cuyo valor se considera constante, y  $\theta_v$  es el ángulo del vector de voltaje dado por:

$$\theta_v = (k-1) \frac{\pi}{3} \quad (4.9)$$

Una vez seleccionado el vector de voltaje adecuado, se generan los pulsos  $a$ ,  $b$  y  $c$  para la apertura o cierre de los dispositivos semiconductores del inversor.



**Figura 4.2.** Vectores espaciales de voltaje del inversor.

La aplicación correcta de este principio permite un control desacoplado del flujo magnético de estator y del par electromagnético sin la necesidad de transformaciones entre marcos de referencia tal como sucede en otras técnicas de control [33]. Sin embargo se ha demostrado que la presencia de los controladores de histéresis provoca una frecuencia de conmutación del inversor variable entre otras desventajas tales como [4][18][19][62]:

- ✓ Distorsión en las señales de corriente y par causadas por el cambio de sector en la selección del vector espacial de voltaje.

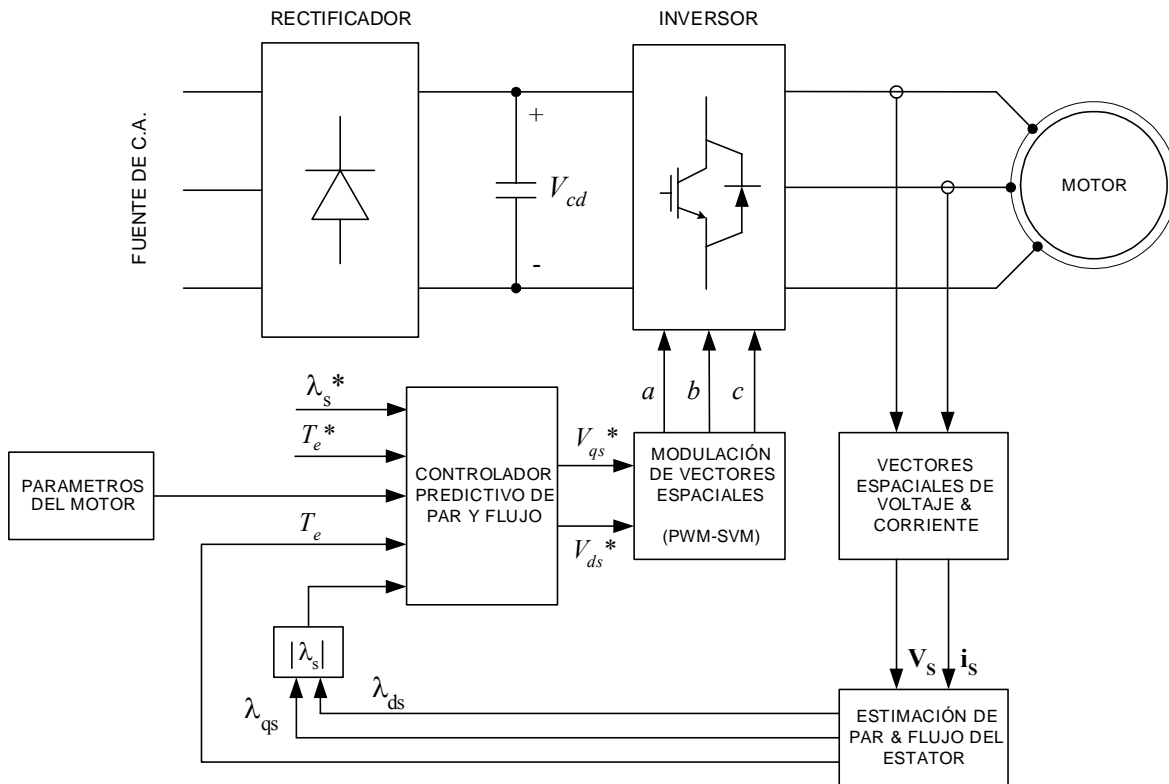
- ✓ Problemas en el arranque y en bajas velocidades del motor.
- ✓ En caso de implementarse, es necesario altas frecuencias de muestreo debidas a los bloques de histéresis.
- ✓ Número limitado de vectores de voltaje.

Por otra parte existe otra desventaja cuando se utilizan tablas de conmutación la cual se refleja justo después que la posición del vector de flujo del estator cambia de un sector a otro. Debido a que no hay un vector de voltaje disponible que asegure un incremento en la magnitud del vector de flujo, este tiende a caer en cada cambio de sector provocando que su trayectoria no sea circular sino más bien hexagonal y que se incremente la distorsión armónica en la corriente del estator [40][50][54].

### **4.3. Control Directo del Par con Modulación de Vectores Espaciales.**

Como una alternativa para eliminar las dificultades que presenta el uso de tablas de conmutación, se incorpora al esquema la técnica de Modulación de Vectores Espaciales [20][33][51] (SVM, ver apéndice C). La figura 4.3 muestra el esquema de control usando esta metodología.

Bajo este esquema de control se logra una reducción en la distorsión del par y la corriente eléctrica ya que en cada instante se calcula un vector de voltaje que compensa exactamente a los errores del par y el flujo magnético del estator [33][40]. Este cálculo se realiza en el controlador, que de manera predictiva genera un vector de referencia que será aproximado por el modulador de vectores espaciales. Aunque bajo este esquema se obtiene un control rápido del par en sistemas de alta potencia, con media y baja potencia no es muy aceptable ya que aumenta en gran medida la complejidad del sistema. Además el controlador requiere calcular variadas y complicadas ecuaciones en línea requiriendo un alto tiempo de cómputo cuando es simulado [20]. Por otra parte, como se muestra en la figura 4.3, esta metodología es sensible a la variación de los parámetros de la máquina [62].

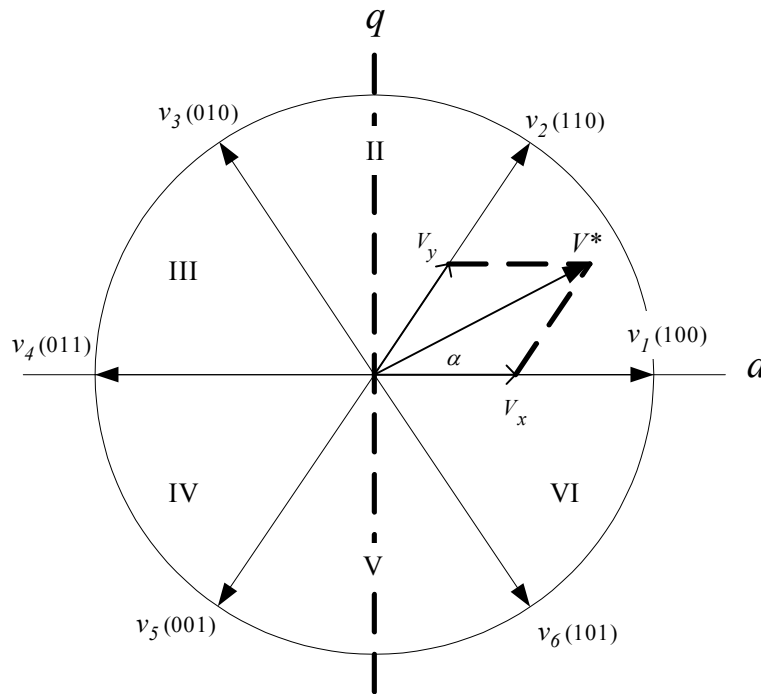


**Figura 4.3.** Control directo de par con modulación de vectores espaciales y controlador predictivo (DTC-SVM).

La técnica SVM se ha convertido recientemente en la técnica PWM más común para inversores trifásicos debido a que tiene un rango de operación extendido y a que genera señales con un contenido armónico menor comparado con las demás técnicas, además que permite mantener constante la frecuencia de conmutación del inversor [66]. El principio de operación es muy sencillo y es fácil su implementación. Para este caso se requiere un vector de voltaje de referencia mismo que será modulado o aproximado por el SVM. Esta aproximación se logra al aplicar los vectores de voltaje adyacentes al vector de referencia por intervalos bien definidos de tiempo. La figura 4.4 presenta la forma en la que tiene lugar la modulación del vector de referencia [1][61][64].

En la figura se observa la manera como el vector de voltaje de referencia  $V^*$  es aproximado en el sector I. En este sector los vectores  $v_1$  y  $v_2$  son aplicados en intervalos de tiempo. Se puede observar también que cada vector representa uno de los seis estados activos del inversor fuente de voltaje. Mediante esta técnica, el inversor es capaz de generar un número ilimitado de vectores a ser aplicados al motor a diferencia de la tabla de conmutación. Más detalles sobre el SVM se encuentran en el apéndice C.





**Figura 4. 4.** Principio de la modulación de vectores espaciales.

En resumen se puede observar que existen ventajas y desventajas en cada una de las metodologías mencionadas sin embargo en esta última puede decirse que aunque la utilización de la técnica de SVM mejora el desempeño del sistema, el controlador requiere modelos matemáticos complicados. Por el otro lado, en el esquema clásico del DTC, mientras los controladores de histéresis son relativamente sencillos el uso de las tablas de conmutación provocan serios problemas.

Por lo anterior vale la pena investigar el comportamiento del sistema incorporando una estructura basada en las metodologías de inteligencia artificial que como ya se ha mencionado a lo largo de los capítulos anteriores conserva cualidades que pueden servir para solucionar los problemas que los esquemas convencionales tienen. Como ya se mencionó las estructuras neuro-difusas no necesitan modelos matemáticos, su arquitectura está basada en operaciones lógicas básicas por lo que su implementación es más sencilla. Además, se adapta ante la variación de los parámetros del sistema y por ser una estructura no lineal tiene ventajas al controlar sistemas no lineales como es el caso del motor de inducción. En la siguiente sección se presenta el esquema de control propuesto incorporando un controlador neuro-difuso adaptable [18][19].

#### 4.4. Control Neuro-Difuso Directo del Par.

La figura 4.5 presenta el esquema del Control Neuro-Difuso Directo del Par y Flujo magnético (DTNFC).

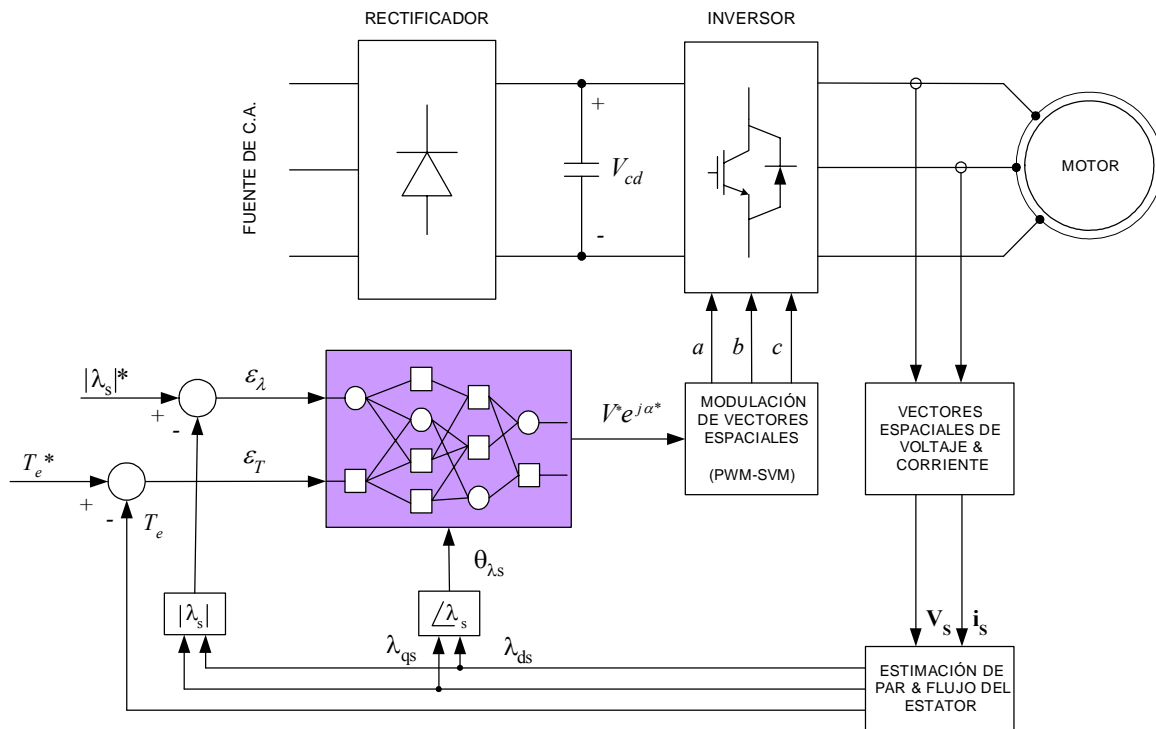


Figura 4.5. Esquema del control neuro-difuso directo del par (DTNFC).

Como se puede ver en el esquema anterior, las señales de error del flujo del estator y el par electromagnético,  $\epsilon_\lambda$  y  $\epsilon_T$ , son utilizadas como entradas del NFC, mismo que también utiliza la información sobre la posición del vector de flujo del estator,  $\theta_{\lambda_s}$ , para generar el vector de voltaje de referencia  $V^* e^{j\alpha^*}$  que será aproximado por el SVM el cual finalmente produce los pulsos  $a$ ,  $b$  y  $c$  que controlan las conmutaciones del inversor. Bajo este esquema se eliminan los controladores de histéresis y las tablas de conmutación del esquema clásico del DTC de la figura 4.1 [18][19]. Sin embargo, se conservan los bloques de estimación del par y flujo, el bloque de producción de los vectores de voltaje y corriente en coordenadas  $dq$ , así como el cálculo de la magnitud del vector de flujo y su correspondiente ángulo.

#### 4.4.1. Arquitectura del Controlador Neuro-Difuso.

De acuerdo a lo establecido en la sección 2.4, la estructura neuro-difusa posee las características inherentes de las metodologías de la inteligencia artificial. Al combinar el razonamiento de los sistemas difusos y las habilidades de aprendizaje y adaptación de las redes neuronales se produce un ANFIS o bien una red adaptable neuro-difusa. La figura 4.6 muestra la arquitectura de la red neuro-difusa propuesta para el control de par del motor de inducción bajo el esquema de la figura 4.5.

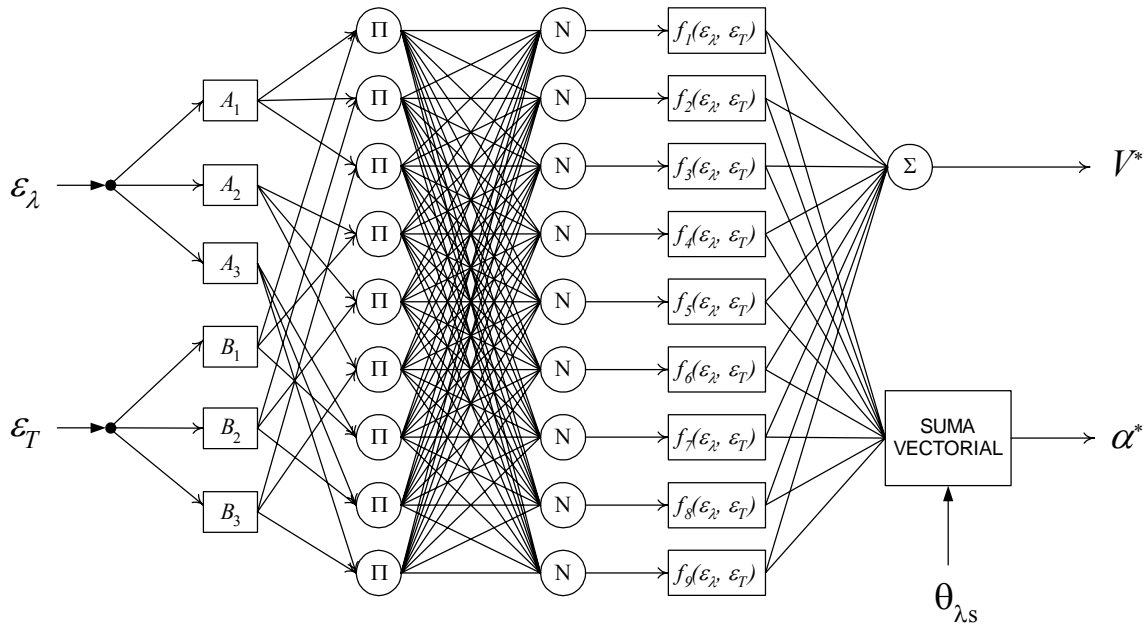


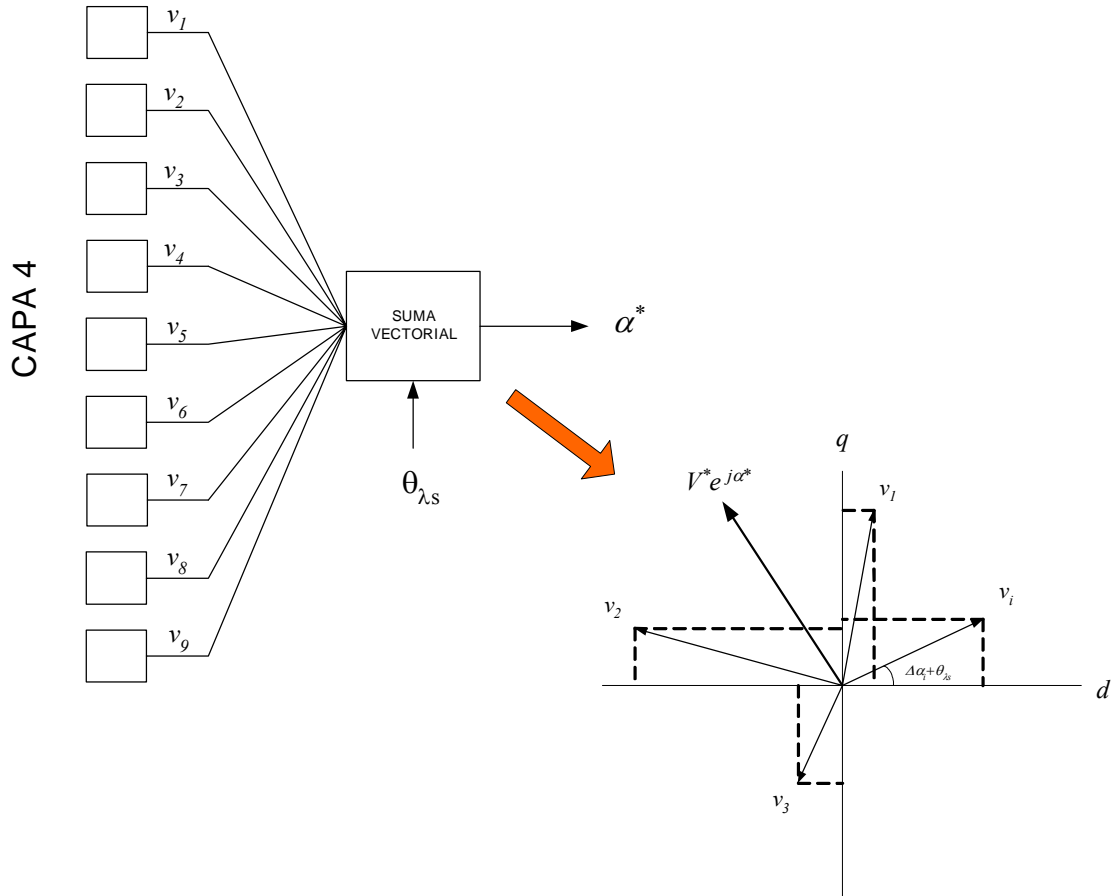
Figura 4.6. Estructura del controlador Neuro-Difuso propuesto.

En la figura anterior los nodos  $A_1$  y  $B_1$  corresponden a las variables lingüísticas “Negativo”, los nodos  $A_2$  y  $B_2$  es “Cero” y  $A_3$  y  $B_3$  “Positivo”.

Aunque la figura anterior muestra dos salidas, en realidad la red sólo consta de una salida, a saber la que proporciona el vector de voltaje en sus coordenadas polares,  $Ve^{j\alpha}$ , sin embargo para mayor claridad la figura 4.6 muestra como la salida de la red se divide en 2 partes: la magnitud del voltaje de referencia y su ángulo. De la figura queda claro que la magnitud del vector,  $V^*$ , resulta de sumar los pesos de la capa 4.

Por otro lado, para el cálculo del ángulo,  $\alpha^*$ , se utiliza la información proveniente de la capa 4 tal como se explica a continuación. El bloque etiquetado como suma vectorial utiliza la información que ya ha sido procesada después que las capas anteriores han asignado un valor lingüístico al error de par y flujo, así como también ha pasado a través de las reglas difusas para tomar la acción de control apropiada. En otras palabras, la  $i$ -ésima salida ( $i=1,2,\dots, 9$ ) de la capa 4 representa una componente del

vector de voltaje con un ángulo definido con anterioridad. Esta operación se muestra en la figura 4.7 donde se puede observar que el bloque cuya salida es el ángulo del vector de referencia,  $\alpha^*$ , es simplemente una suma vectorial de los vectores parciales correspondiente a las salidas de la capa 4.



**Figura 4.7.** Cálculo del ángulo del vector espacial de voltaje.

De la figura anterior se tiene que:

$$v_i = w_{4i} e^{j(\Delta\alpha + \theta_{\lambda s})} \quad (4.10)$$

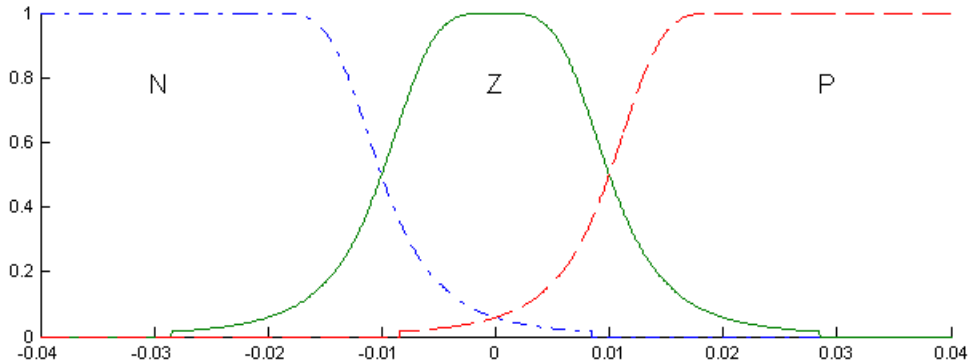
donde  $w_{4i}$  es el peso de la  $i$ -ésima salida de la capa 4,  $\theta_{\lambda s}$  es la posición que guarda el vector de flujo del estator con respecto al eje  $d$ , y  $\Delta\alpha$  es el incremento de dicho ángulo para el vector parcial  $v_i$  [60]. Es importante destacar, que el incremento  $\Delta\alpha$ , también resulta de utilizar un razonamiento difuso TKS pero de orden cero. Esto se resume en

la tabla 4.1, donde cada vector contiene la información antecedente del razonamiento (por ejemplo  $v_1$  resulta de un error de par negativo y un error de flujo negativo).

**Tabla 4.1.** Incremento del ángulo del vector parcial del vector de voltaje de referencia.

Vector parcial	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$	$v_6$	$v_7$	$v_8$	$v_9$
$\Delta\alpha$	$2\pi/3$	$\pi$	$2\pi/3$	$-\pi/2$	$\pi/2$	$\pi/2$	$-\pi/3$	0	$\pi/3$

Las variables lingüísticas que etiquetan la capa 1 se muestran en la figura 4.8, donde se aprecia que las funciones de membresía son del tipo campana. Cabe recordar que para este caso los parámetros de las funciones son fijos ya que de esta manera se permite que un operador experto transmita sus conocimientos y experiencia al sistema ya que sólo él es el único que conoce bien el significado de “error negativo”, “error cero” y “error positivo” (N, Z y P respectivamente) para una aplicación en particular.



**Figura 4.8.** Funciones de membresía utilizadas en la capa 1 del NFC.

Se puede decir entonces que en general, la operación de la red mostrada en la figura 4.6 es idéntica a la descrita en la sección 2.4 con la diferencia que esta red cuenta con 9 reglas difusas, 3 funciones de membresía asociadas a cada entrada, 18 parámetros antecedentes y 27 consecuentes. La salida entonces del controlador es el vector de voltaje que será aplicado al SVM para ser aproximado y así controlar las conmutaciones de inversor que alimenta al motor de inducción. Sin embargo debe recordarse que antes de que la red neuro-difusa pueda ser aplicada como controlador, es necesario realizar el proceso de identificación del modelo inverso para garantizar que el vector de referencia sea el óptimo, desde el punto de vista de la minimización del error, para un control rápido del par y flujo [23][24][38]. Este proceso es quizás una desventaja de esta estrategia de control ya que se tiene que desarrollar un esquema de

entrenamiento. La sección siguiente presenta el esquema de entrenamiento que se aplica a la red que para este caso funciona como identificador del sistema.

#### 4.4.2. Esquema de Entrenamiento de la Red.

Como se mencionó en la sección 3.4.1, la aplicación de una red neuro-difusa con propiedades de adaptación consta de dos fases: la fase de entrenamiento y la fase de aplicación. Sin embargo, para la fase de entrenamiento es necesario plantear un nuevo esquema. Tal como se planteo en la sección 3.4.2, en esta fase la red identifica el sistema a controlar a través de aproximar su modelo inverso. La figura 4.9 muestra el esquema requerido para entrenar la red.

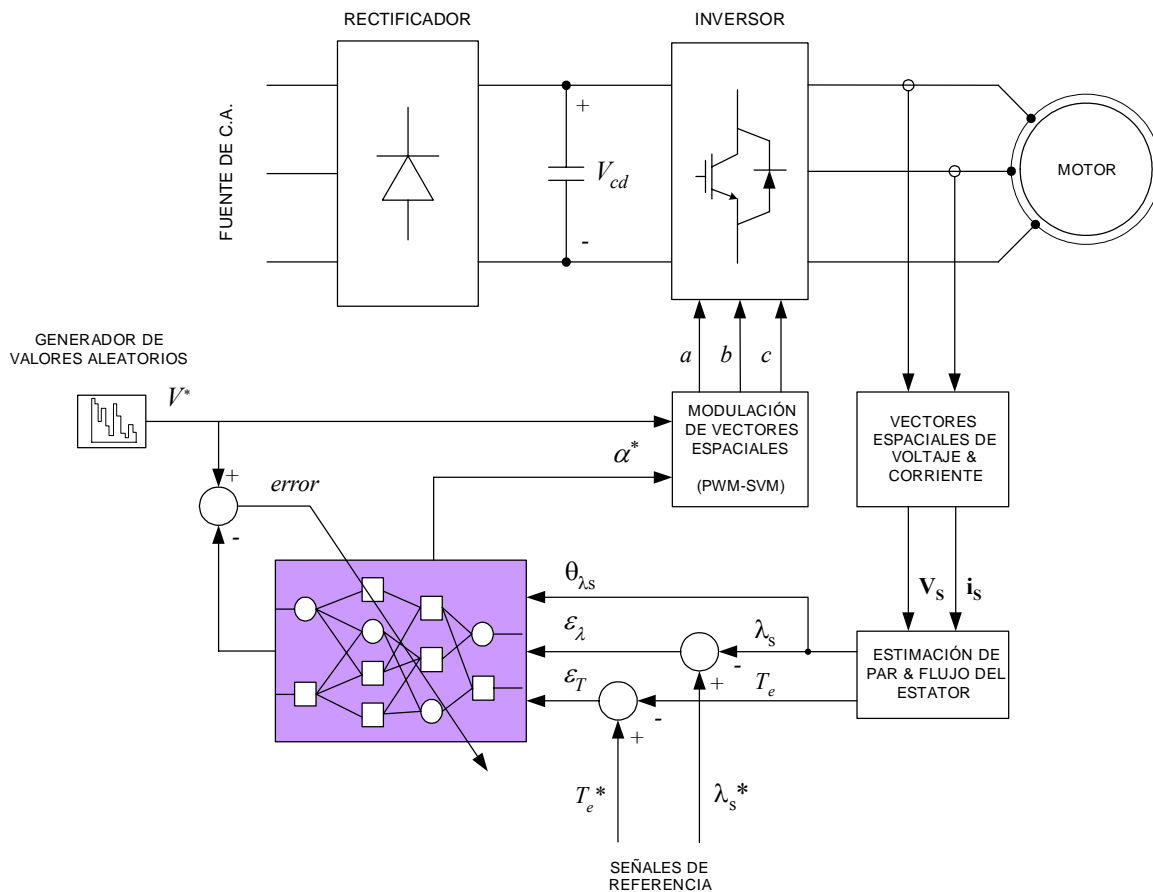


Figura 4. 9. Esquema de entrenamiento para el DTNFC.

En la figura anterior se puede apreciar el entrenamiento de la red a través del aprendizaje inverso introducido en la sección 3.4.1. Como se puede observar la salida de la red se compara con la señal de entrada a la planta, en este caso la magnitud del vector de referencia,  $V^*$ , para el PWM-SVM, que para efectos del proceso de entrenamiento puede ser aleatoria o bien constante, si se conoce cual será el efecto esta causará en el par o flujo magnético. Luego entonces para cada vector de voltaje aplicado en las terminales del motor se producirán las correspondientes señales estimadas de par y flujo magnético,  $T_e$  y  $\lambda_s$  respectivamente. Estas señales son comparadas con las de referencia,  $T_e$  y  $\lambda_s$ , mismas que pueden ser señales predefinidas de antemano o bien aleatorias. Cabe aclarar que es recomendable que las señales de referencia de par y flujo sean similares a las que se utilizaran en la fase de aplicación. Esto es sencillo desde el punto de vista que es común que un experto conozca perfectamente las condiciones de operación a las que el NFC será sometido.

Los errores del par y flujo,  $\varepsilon_T$  y  $\varepsilon_\lambda$ , sirven de señales de entrada a la red. Como se mencionó en la sección anterior, es necesario conocer el valor del ángulo del vector de flujo del estator,  $\theta_{\lambda_s}$ , para que de la misma manera que en la fase de aplicación, la red genere el ángulo del vector de voltaje de referencia,  $\alpha^*$ .

El proceso anterior permite que el algoritmo del filtro de Kalman encuentre los valores óptimos de los parámetros consecuentes minimizando el error y generando de esta forma las reglas difusas para distintos puntos de operación [10][38].

Por simplicidad, en la figura 4.9 se han omitido los bloques del cálculo de la magnitud del vector del flujo del estator y su correspondiente ángulo.

## 4.5. Pruebas bajo distintas Condiciones de Operación.

Para llevar a cabo la simulación digital del DTNFC fue necesario la elaboración de un programa de cómputo en lenguaje C++ tanto para la fase de entrenamiento mostrada en la figura 4.9 como para la fase de aplicación de la figura 4.5. Estos programas incluyen el modelo matemático del motor de inducción compuesto por 5 ecuaciones diferenciales de primer orden integradas numéricamente por el método de Runge Kutta, la estructura de la red neuro-difusa de la figura 4.6, el filtro de Kalman como algoritmo de aprendizaje de la red, la técnica de modulación de vectores espaciales y el inversor fuente de voltaje. Estos programas pueden revisarse en el apéndice E en la parte final de este trabajo.

Como ya se mencionó, para asegurar un buen comportamiento del controlador, la fase de entrenamiento es indispensable ya que de otra forma, los parámetros consecuentes de la red tendrían que obtenerse por métodos heurísticos que, aunque han sido aplicados con éxito en numerosos experimentos, consume tiempo y no se sabe con certeza si dichos valores son los óptimos desde el punto de vista de la minimización

del error [48]. Ya se mencionó en la sección 2.5.1 la manera como el filtro de Kalman es empleado para encontrar los valores óptimos de los parámetros consecuentes y de esta forma construir las reglas difusas las cuales son el corazón del controlador.

Una vez que se ha llevado a cabo el proceso de entrenamiento mediante el aprendizaje inverso, la parte consecuente de las reglas difusas se genera automáticamente. La tabla 4.2 muestra las nueve reglas difusas que describen el comportamiento no lineal del controlador Neuro-Difuso.

Esta claro que para cada experimento diferente al que se va a aplicar el NFC es necesario un nuevo entrenamiento por lo que los parámetros se ajustan. El programa de cómputo correspondiente al entrenamiento tiene la ventaja que genera un archivo de texto con los valores de los parámetros. Este archivo es leído posteriormente por el programa de aplicación.

**Tabla 4. 2.** Reglas difusas del controlador Neuro-Difuso.

Antecedente		Consecuente	
<i>negativo</i>	<i>negativo</i>	<i>negativo</i>	$f_1 = -0.033103\epsilon_\lambda + 0.073129\epsilon_T + 2.50514$
<i>negativo</i>	<i>cero</i>	<i>cero</i>	$f_2 = -0.026197\epsilon_\lambda - 0.052393\epsilon_T + 2.498416$
<i>negativo</i>	<i>positivo</i>	<i>positivo</i>	$f_3 = -0.029641\epsilon_\lambda + 0.001635\epsilon_T + 2.499936$
<i>cero</i>	<i>negativo</i>	<i>negativo</i>	$f_4 = 0.10587\epsilon_\lambda + 0.0298553\epsilon_T + 2.500301$
Si $\epsilon_\lambda$ es <i>cero</i>	y $\epsilon_T$ es <i>cero</i>	<i>cero</i>	entonces $f_5 = 0.005804\epsilon_\lambda + 0.016568\epsilon_T + 2.500246$
<i>cero</i>	<i>positivo</i>	<i>positivo</i>	$f_6 = 0.000288\epsilon_\lambda + 0.000662\epsilon_T + 2.499992$
<i>positivo</i>	<i>negativo</i>	<i>negativo</i>	$f_7 = 0.019027\epsilon_\lambda - 0.050093\epsilon_T + 2.498475$
<i>positivo</i>	<i>cero</i>	<i>cero</i>	$f_8 = 0.021317\epsilon_\lambda + 0.040937\epsilon_T + 2.500862$
<i>positivo</i>	<i>positivo</i>	<i>positivo</i>	$f_9 = 0.024664\epsilon_\lambda + 0.000176\epsilon_T + 2.499781$

$\epsilon_T$ : error del par electromagnético.

$\epsilon_\lambda$ : error del flujo magnético del estator.

#### 4.5.1. Desempeño en Estado Estacionario.

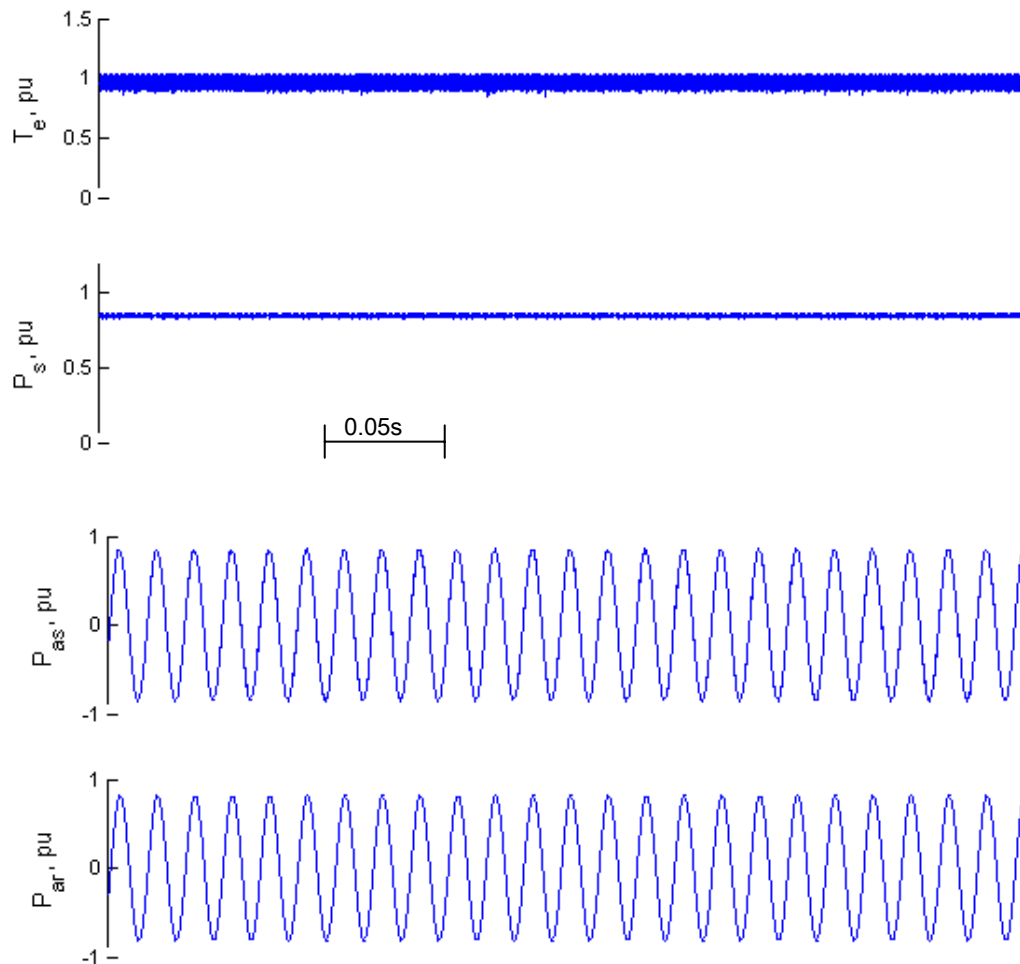
El desempeño del sistema en estado estacionario se muestra en las figuras 4.10-4.14. El tiempo de muestreo es de  $10\mu s$ , la frecuencia fundamental es de 60Hz mientras que la frecuencia de conmutación del inversor es de 15kHz. Los parámetros del motor de inducción se muestran en el apéndice D. El tiempo de simulación mostrado es de 0.4 segundos. La simulación se realiza considerando que el motor esta a plana carga ( $T_m=1.0$  p.u).

En la figura 4.10 se aprecia el par electromagnético desarrollado por el motor de inducción, así como también la magnitud del vector de flujo del estator. El aspecto más



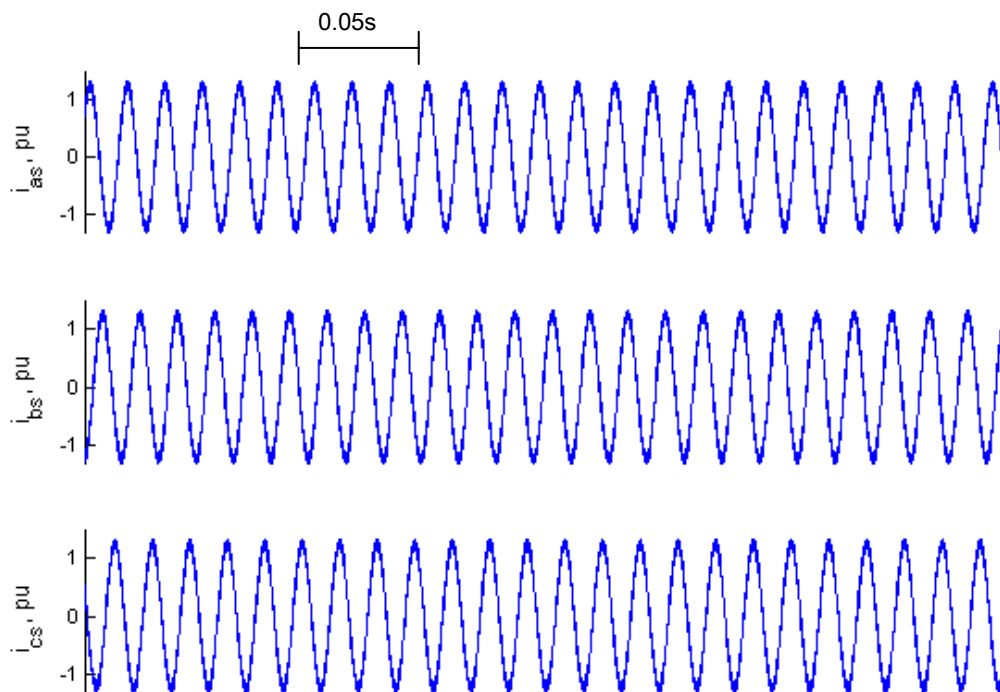
importante que se puede observar se relaciona con la uniformidad de las señales, es decir, a diferencia de DTC clásico, aquí no se presenta ninguna variación o caída en el par o el flujo debida al cambio de sector por el vector del flujo del estator. Bajo este esquema, el NFC provee una infinidad de vectores para cada sector del plano  $dq$  por lo que en cada instante existe un vector adecuado que compense el error del flujo y de esta manera garantizar la magnitud constante del flujo magnético [40][54].

Por otra parte, en la figura se muestran las señales del flujo magnético del estator y del rotor. De una comparación entre estas dos señales se puede observar el rizado en la señal del flujo del estator debida a las inevitables conmutaciones del inversor.

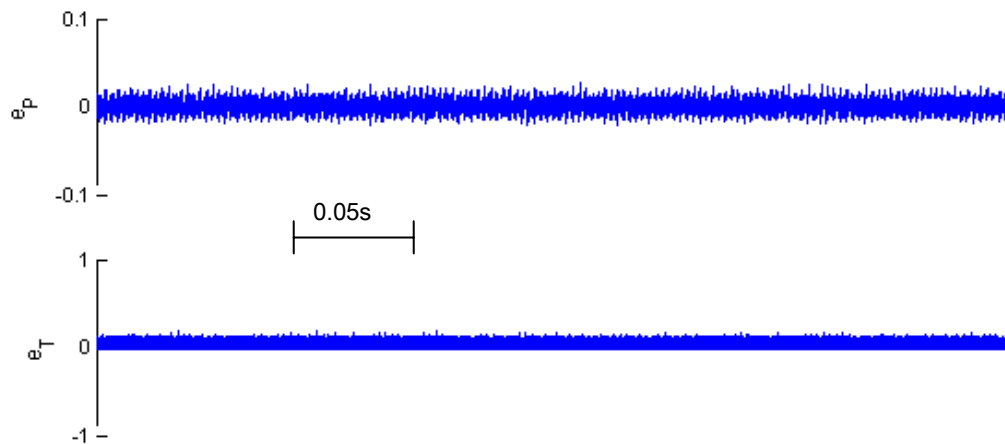


**Figura 4. 10.** Desempeño en estado estacionario.  $T_e$ , Par electromagnético;  $P_s$ , Magnitud del vector de flujo magnético del estator;  $P_{as}$ , Flujo magnético de la fase a del estator;  $P_{ar}$ , Flujo magnético de la fase a del rotor.

Ahora bien, la figura 4.11 muestra las señales de la corriente eléctrica que fluye en los devanados del estator del motor. Como se aprecia, estas funciones cuentan con una forma de onda casi senoidal aunque con el ruido causado también por los efectos de las conmutaciones del inversor. Por otro lado, el error en estado estacionario del flujo y par se presenta en la figura 4.12 en donde para ambos casos el error tiende a ser cercano a cero aunque con las variaciones debidas a las conmutaciones inevitables del inversor. Esto demuestra que en el proceso de entrenamiento de la red neuro-difusa mediante el aprendizaje inverso, la red identifica y modela la función inversa del sistema permitiendo que se generen las acciones de control lo más aproximadas posible.

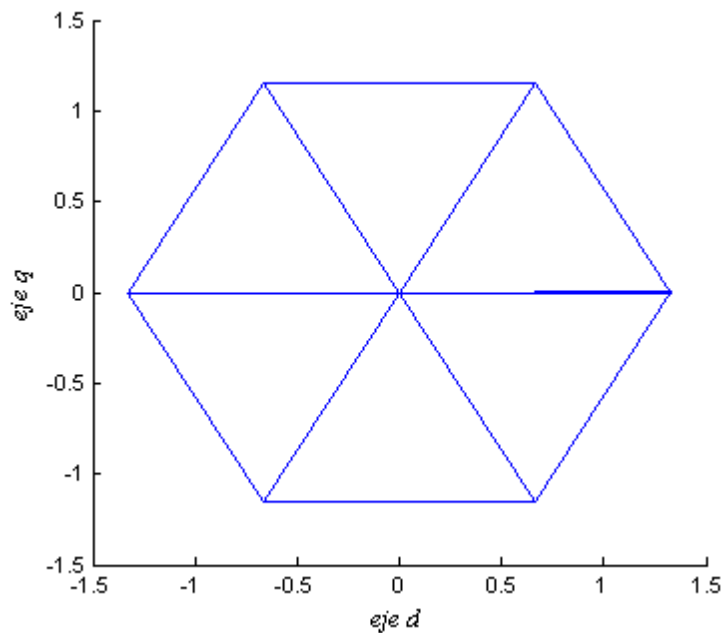


**Figura 4. 11.** Corriente eléctrica en estado estacionario.  $i_{as}$ , Corriente fase a;  $i_{bs}$ , Corriente fase b;  $i_{cs}$ , Corriente fase c.



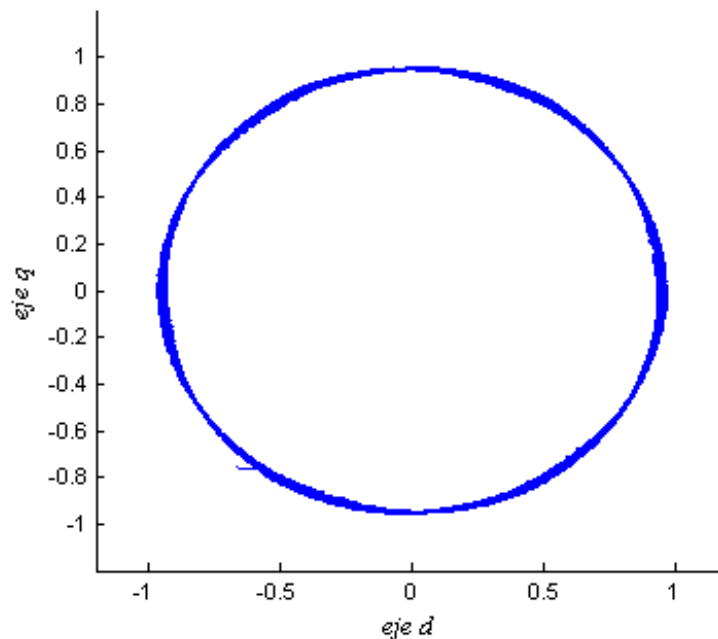
**Figura 4. 12.** Error del par y flujo en estado estacionario.  $e_p$ , Error del flujo magnético del estator;  $e_T$ , error del par electromagnético.

Las figuras 4.13 y 4.14 presentan la posición en el plano complejo de los vectores de voltaje y de flujo magnético del estator en el espacio. De la primera se deduce que el control de los estados de conmutación del inversor es apropiado gracias a la utilización del SVM y que el voltaje de referencia proporcionado por el NFC es el óptimo para cada instante de operación.



**Figura 4. 13.** Trayectoria del vector de voltaje.

El buen desempeño del DTNFC se observa por la trayectoria del vector de flujo mostrado en la figura 4.14, la cual es prácticamente una circunferencia perfecta. Esto se debe a que el NFC provee una infinidad de vectores para cada sector del plano  $dq$  mismos que son aproximados por el SVM y transmitidos al inversor [40]. Esto es una gran mejoría con respecto al DTC clásico donde para obtener una trayectoria circular es necesario un valor pequeño en el ancho de banda del controlador de histéresis. Sin embargo, entre más pequeño sea el ancho de banda se tendrá como resultado una alta frecuencia de conmutación.



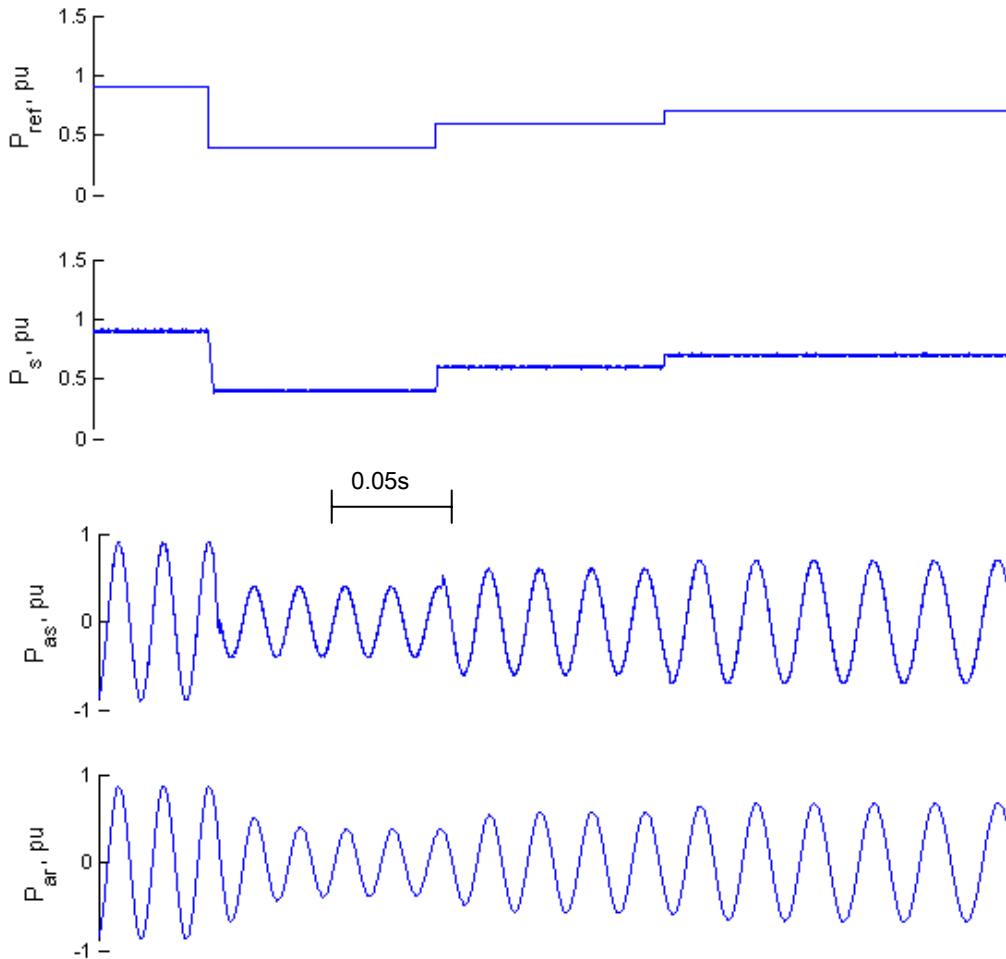
**Figura 4. 14.** Trayectoria del vector del flujo del estator.

#### 4.5.2. Control del Flujo Magnético del Estator.

Una de las ventajas del esquema del DTC es que es posible el control desacoplado del flujo y del par electromagnético de manera que se pueden manejar las dos variables de manera separada. Esto se puede observar a partir del siguiente experimento en donde el sistema se ve sometido a distintos cambios en la señal de referencia del flujo magnético del estator pero conservando constante el par electromagnético. Para esta prueba se considera que el tiempo de muestreo es de  $10\mu s$ , la frecuencia fundamental es de 60Hz mientras que la frecuencia de conmutación del inversor es de 15kHz. El par mecánico está al 30% del nominal. El tiempo de simulación mostrado es de 0.4 segundos.

La figura 4.15 muestra el desempeño del DTNFC al someter al motor a diferentes cambios en la magnitud del vector del flujo magnético. Después de estar operando en estado estacionario, la señal de referencia del flujo cambia repentinamente a 40, luego 60 y finalmente al 70% del valor nominal dejando pequeños lapsos de tiempo entre dichos cambios. Como se aprecia, la respuesta del sistema es muy rápida, casi instantánea. La magnitud del vector de flujo sigue perfectamente a la referencia sin que esto signifique en un aumento en el rizado de las ondas de corriente o las del flujo magnético.

Por otra parte, la magnitud del par electromagnético producido es constante aunque su distorsión se ve afectada por la amplitud de onda de los enlaces del flujo. Sin embargo esa pequeña variación es trivial.

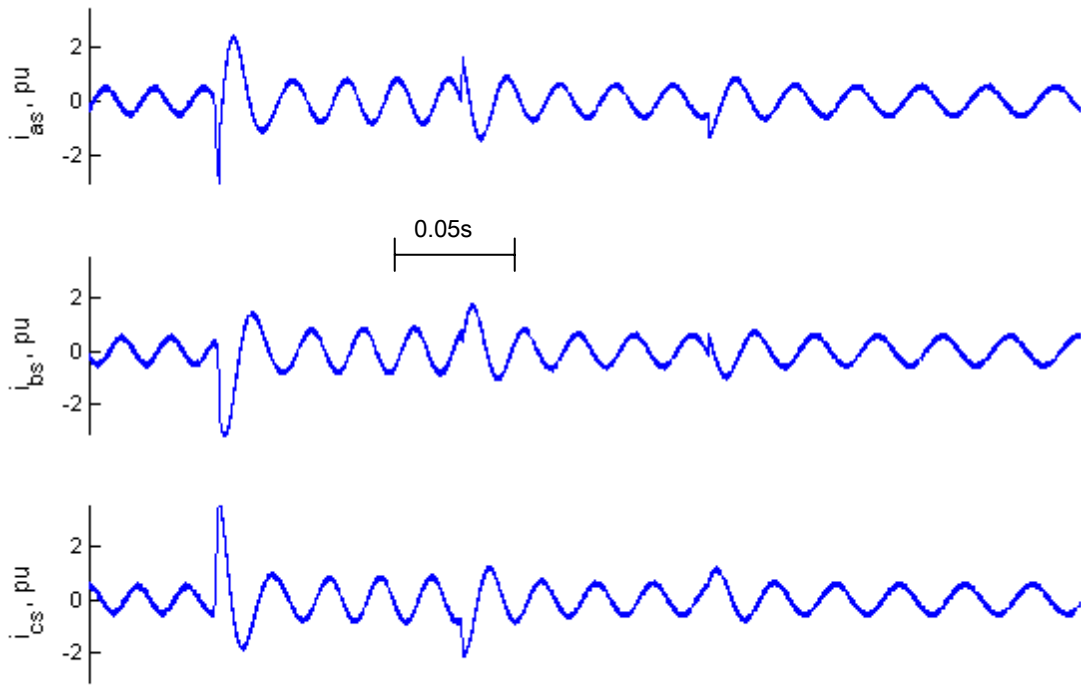


**Figura 4. 15.** Desempeño del DTNFC ante cambios en la señal de referencia del flujo magnético.

$P_{ref}$ , Señal de referencia del flujo del estator;  $P_s$ , Magnitud del vector de flujo magnético del estator;  $P_{as}$ , Flujo magnético en la fase a del estator;  $P_{ar}$ , Flujo magnético en la fase a del rotor.

La respuesta del flujo a su señal de referencia prueba claramente que el NFC proporciona la acción de control apropiada para suministrar un cambio en el estado de conmutación del inversor.

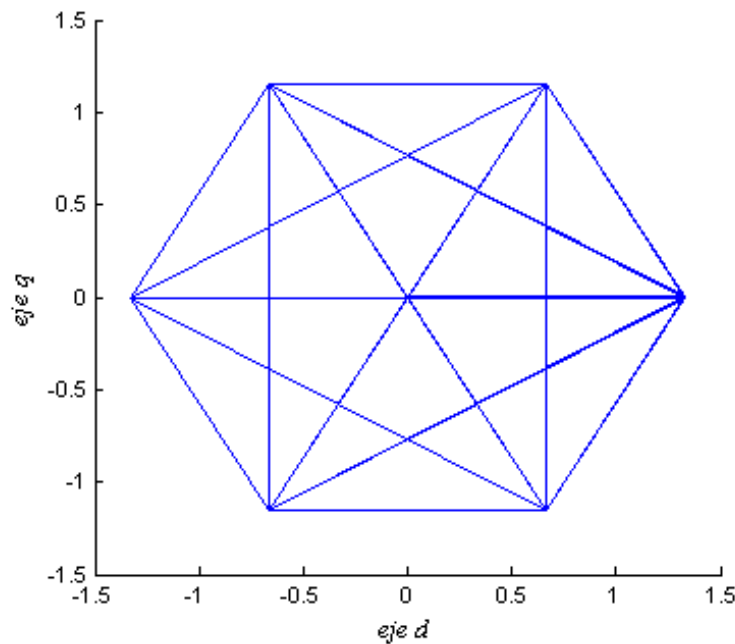
La figura 4.16 presenta el efecto que causa un cambio en el flujo magnético en la corriente eléctrica que circula en los devanados del estator.



**Figura 4. 16.** Corriente eléctrica ante cambios en la señal del flujo del estator.  $i_{as}$ , Corriente fase a;  $i_{bs}$ , Corriente fase b;  $i_{cs}$ , Corriente fase c.

Las figuras 4.17. 4.18 y 4.19 presentan las trayectorias de los vectores de voltaje, flujo del estator y flujo del rotor respectivamente en el plano complejo. Como ya se mencionó, para un buen control del flujo y par es necesario que el NFC proporcione la referencia óptima de voltaje para realizar rápidamente el cambio en el estado del inversor. A diferencia de la figura 4.13, donde los cambios en la aplicación de los vectores de voltaje se realizan de manera consecutiva, ahora ante un cambio en la referencia, el inversor cambia de estado según la necesidad en ese instante. Por lo tanto se puede decir que [60]:

- ✓ Vectores de voltaje activos cuyo desalineamiento con el vector del flujo no excede  $\pm 90^\circ$  provoca que el flujo se incremente.
- ✓ Vectores de voltaje activos cuyo desalineamiento con el vector del flujo excede  $\pm 90^\circ$  provoca que el flujo disminuya.
- ✓ Los vectores cero (no activos) de corta duración prácticamente no afectan el flujo pero si detiene su movimiento.

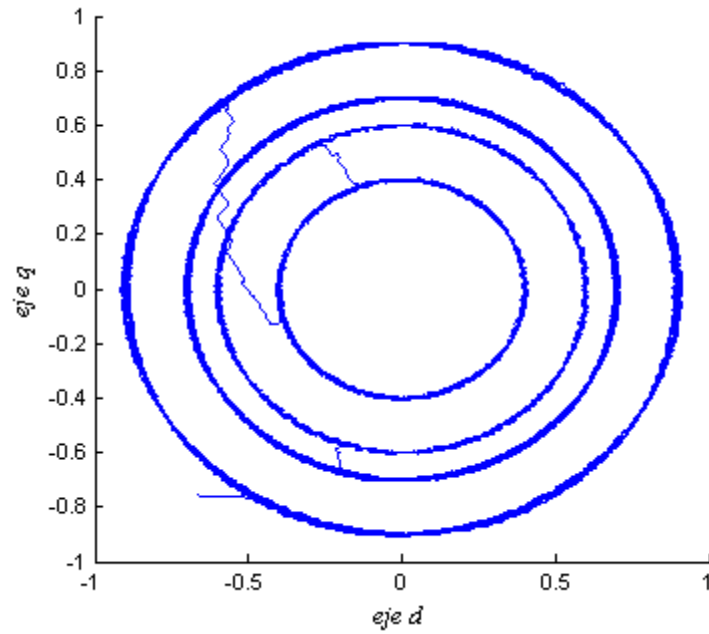


**Figura 4. 17.** Trayectoria del vector de voltaje.

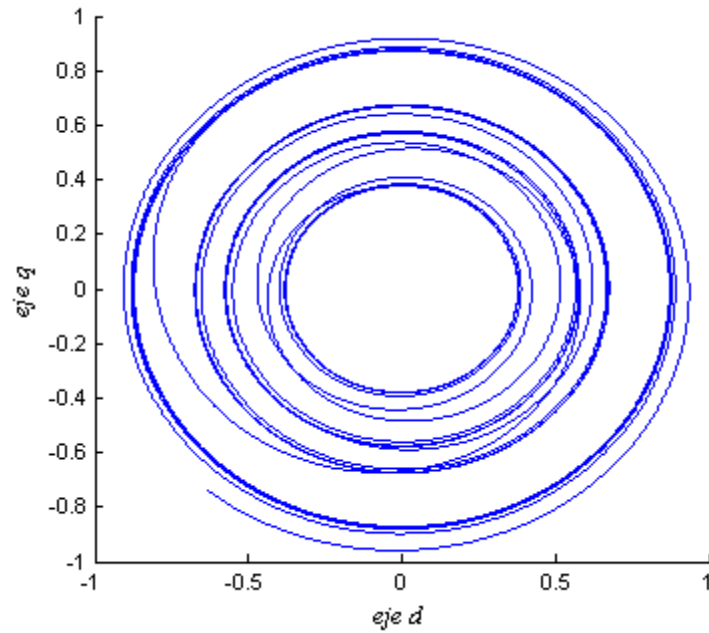
Este cambio en la magnitud de flujo se observa más explícitamente en la figura 4.18 donde se aprecian las cuatro etapas del experimento. Cabe destacar que la trayectoria del vector es prácticamente circular tomando en cuenta que para esta prueba se controla sobre un dominio de entre el 40 y 95% de su valor nominal.

Finalmente, la figura 4.19 muestra la trayectoria del vector del flujo del rotor. De esta figura se desprenden dos aspectos importantes. El primero tiene que ver con la forma de la trayectoria; esta es más circular compara con la trayectoria del estator ya que el entrehierro de la máquina sirve como filtro de armónicos por lo que la señal de flujo rotor es más “limpia” que la del estator. Por otra parte, el segundo aspecto interesante es la respuesta del flujo del rotor ante los cambios en la referencia. Como

la constante de tiempo del rotor es más grande que la del estator, la respuesta es más lenta tal como puede apreciarse.



**Figura 4. 18.** Trayectoria del vector del flujo del estator.



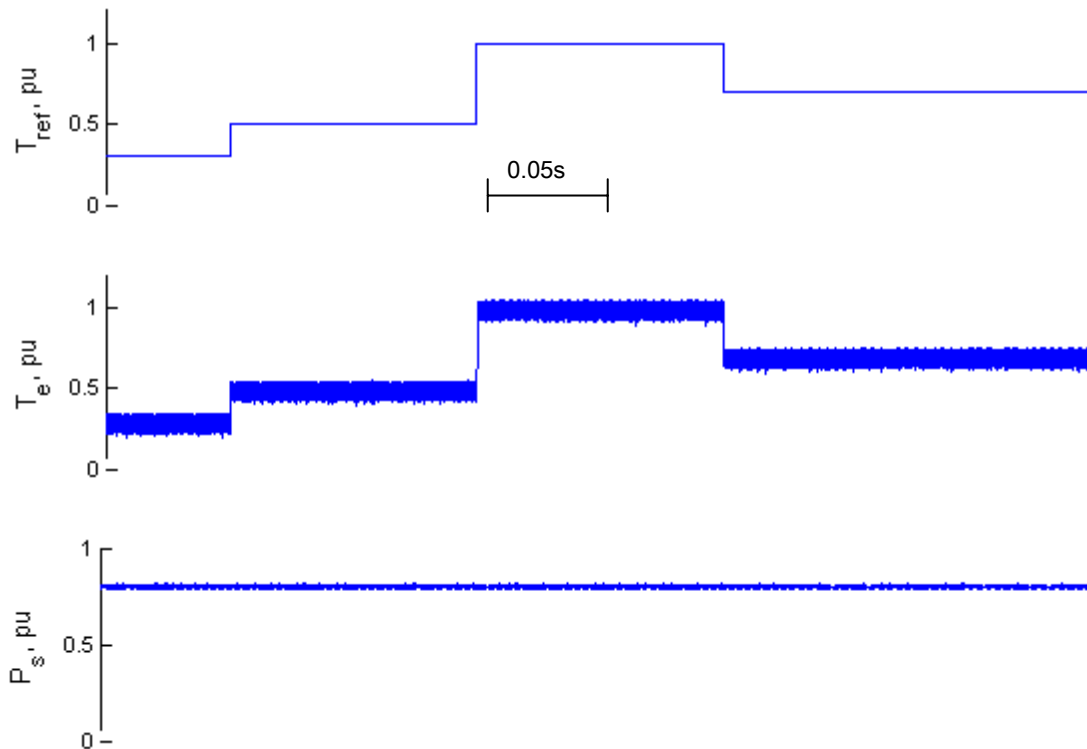
**Figura 4. 19.** Trayectoria del vector del flujo del rotor.



### 4.5.3. Control del Par Electromagnético.

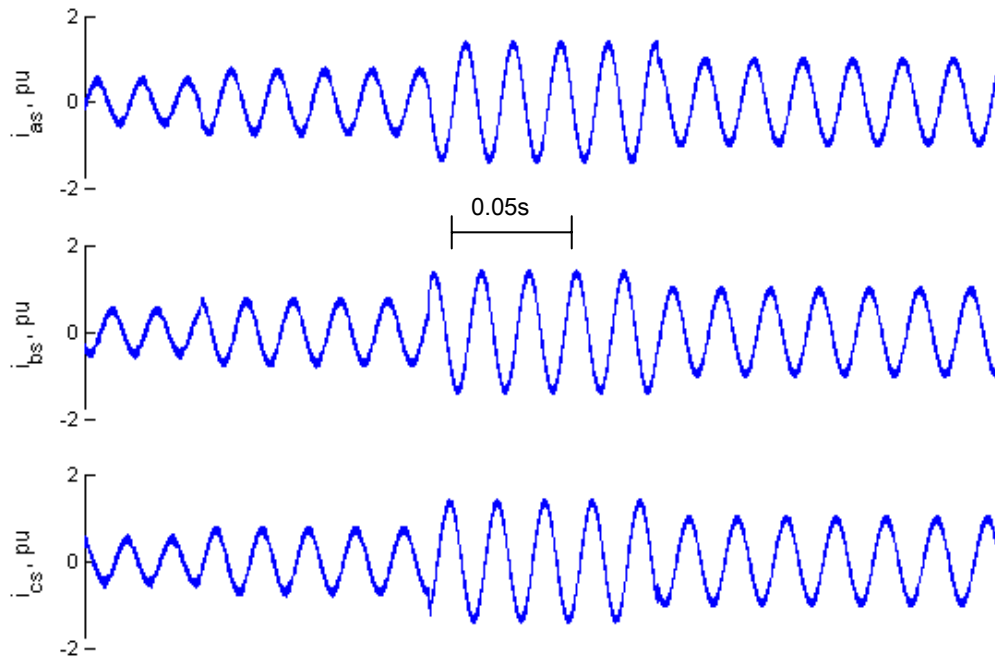
El siguiente experimento consiste en simular el comportamiento del sistema ante cambios repentinos en la señal de referencia del par electromagnético. Para este experimento se considera que el tiempo de muestreo es de  $10\mu\text{s}$ , la frecuencia fundamental es de 60Hz mientras que la frecuencia de conmutación del inversor es de 15kHz. El par mecánico está al 30% del nominal. El tiempo de simulación mostrado es de 0.4 segundos. La referencia del flujo ahora se considera constante para este caso.

Después de estar operando en estado estacionario, la señal de referencia del par electromagnético cambia repentinamente al 50, luego 100 y finalmente al 70% del valor nominal dejando pequeños lapsos de tiempo entre dichos cambios. Como se aprecia en la figura 4.20, al igual que como sucedió en el flujo magnético, la respuesta es rápida. El par electromagnético producido en el motor sigue perfectamente a la referencia sin que el flujo magnético varíe como resultado de estos cambios.



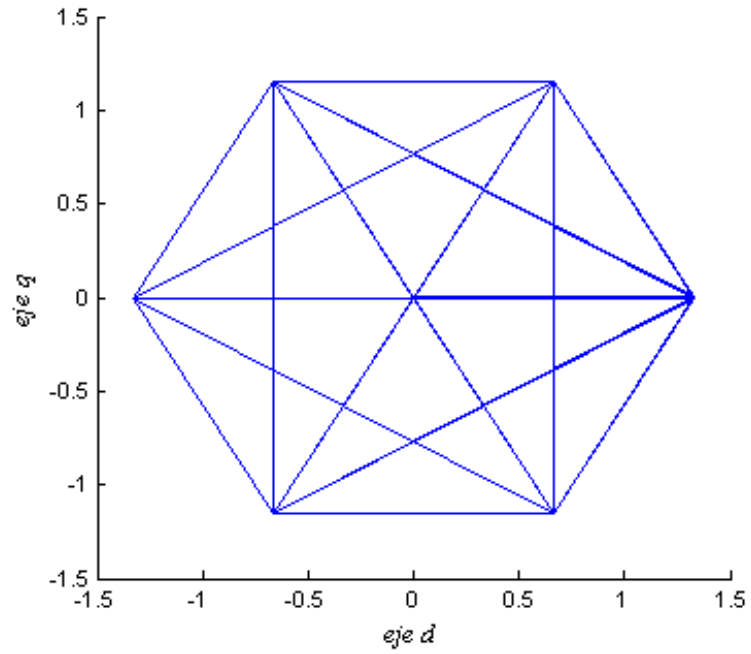
**Figura 4. 20.** Desempeño del DTNFC ante cambios en el par de referencia.  $T_{ref}$ , Señal de referencia del par;  $T_s$ , Par electromagnético desarrollado;  $P_s$ , Flujo magnético del estator.

La figura 4.21 presenta las señales de corriente eléctrica en las fases del estator. Nuevamente se observa que la corriente eléctrica conserva pequeños rizados debidas a la conmutación del inversor.

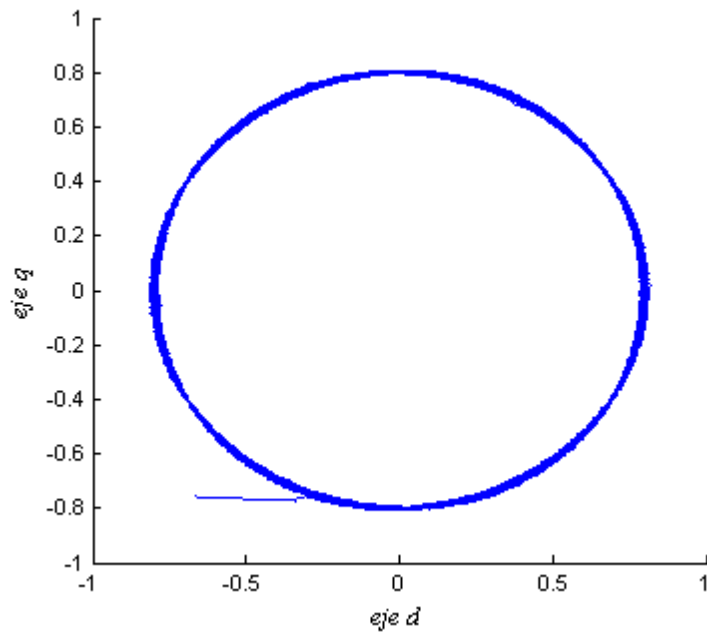


**Figura 4. 21.** Corriente eléctrica ante cambios en la señal del par electromagnético.  $i_{as}$ , Corriente fase a;  $i_{bs}$ , Corriente fase b;  $i_{cs}$ , Corriente fase c.

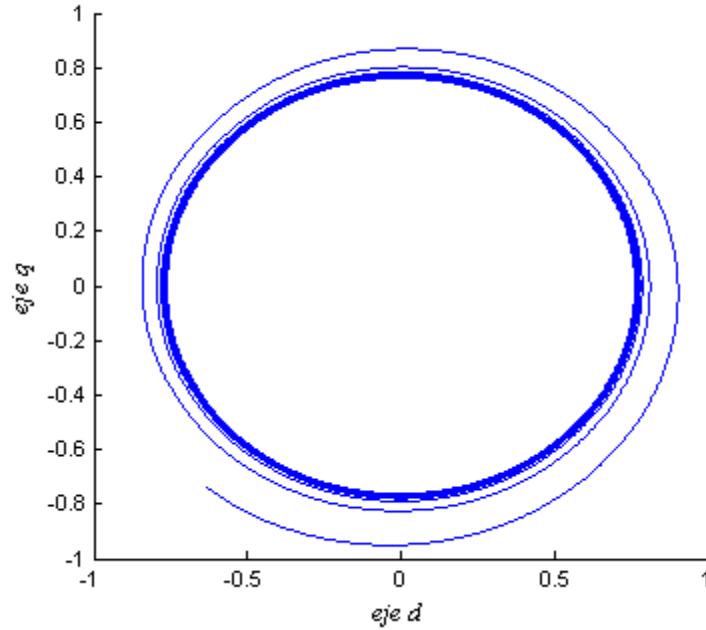
Finalmente, las figuras 4.22-4.24 presentan las trayectorias de los vectores de voltaje, flujo del estator y flujo del rotor en el plano complejo. Tal como se mencionó anteriormente, el vector del flujo magnético del estator no presenta variación alguna debida a los cambios en el par electromagnético conservando así una trayectoria circular.



**Figura 4. 22.** Trayectoria del vector de voltaje.



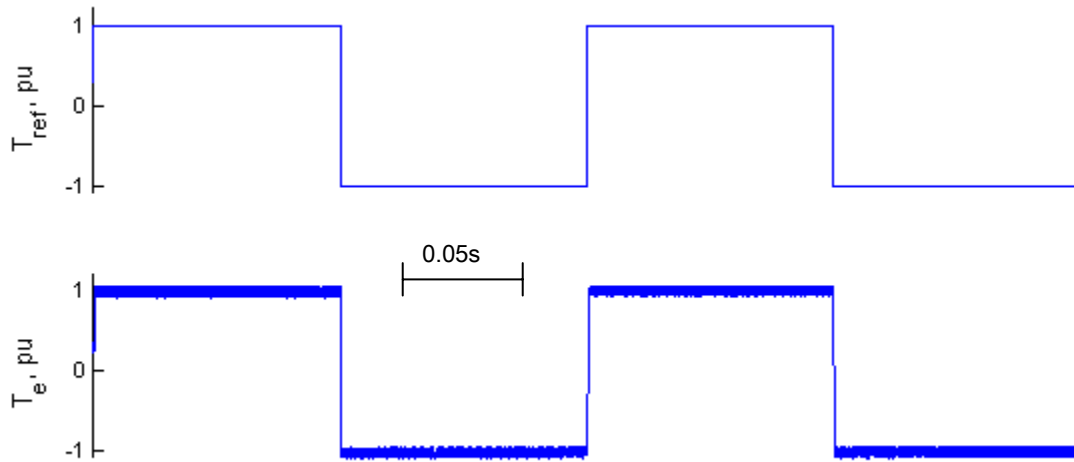
**Figura 4. 23.** Trayectoria del vector del flujo del estator.



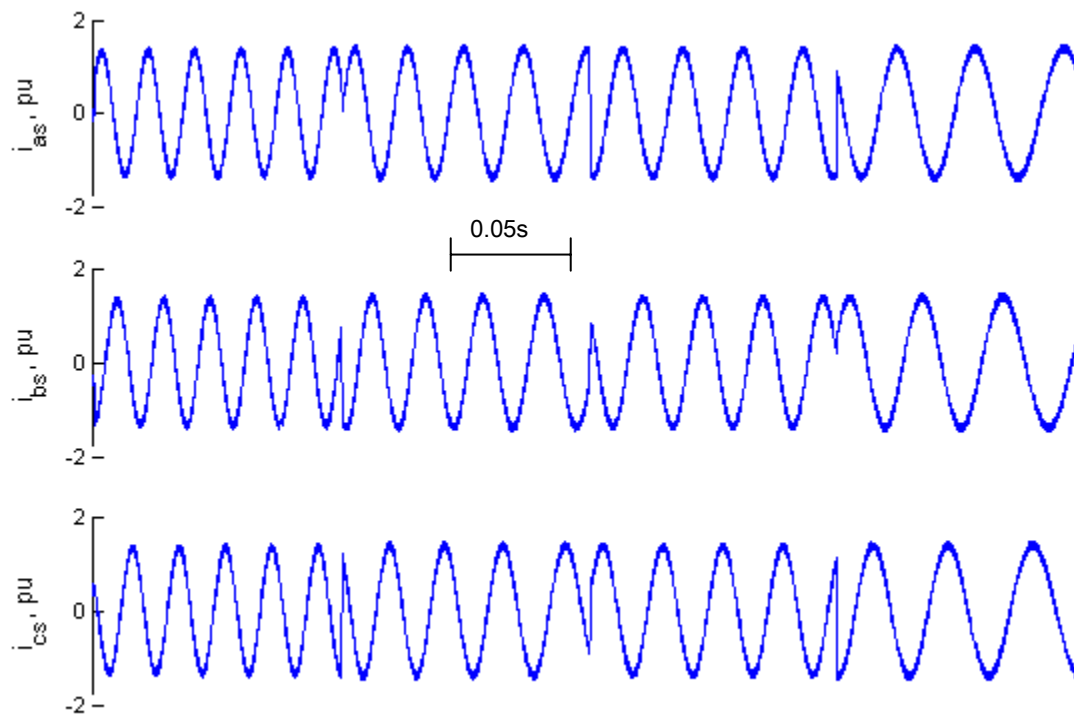
**Figura 4. 24.** Trayectoria del vector del flujo del rotor.

El DTNFC también puede extenderse a la zona de operación donde el motor de inducción requiere de una acción de reversa o cambio de giro [18] de tal forma que la señal de referencia del par tome valores negativos de manera intermitente. El siguiente experimento consiste en simular cambios de giro en el motor de inducción. Para este experimento se considera que el tiempo de muestreo es de  $10\mu\text{s}$ , la frecuencia fundamental es de 60Hz mientras que la frecuencia de conmutación del inverso es de 15kHz. El par mecánico está al 50% del nominal. El tiempo de simulación mostrado es de 0.4 segundos. La referencia del flujo se considera constante.

La figura 4.25 muestra el desempeño del motor cuando la señal de referencia del par toma valores negativos de manera sistemática de tal forma que el motor cambie su sentido de giro. Este experimento es muy común en sistemas de tracción donde se requiere que el giro de rotor pueda operarse en sentido reversible, o bien en caso de que la flecha del rotor de atasque por alguna razón, se tenga la capacidad de recuperarse a través de la acción reversa. La figura 4.26 presenta el efecto que causa en las corrientes el cambio de giro del motor Cabe mencionar que el NFC debió ser entrenado de antemano para poder realizar esta prueba.



**Figura 4. 25.** Desempeño del DTNFC ante cambios en la señal de referencia d el par.  $T_{ref}$ , Par de referencia;  $T_e$ , Par electromagnético producido.



**Figura 4. 26.** Corriente eléctrica ante cambios en la señal del par electromagnético.  $i_{as}$ , Corriente fase a;  $i_{bs}$ , Corriente fase b;  $i_{cs}$ , Corriente fase c.

El NFC propuesto es un controlador no lineal por lo que el dominio de control es más amplio que si se utilizara un controlador lineal tal como PD o PI. Sin embargo su mayor desventaja es que es necesario entrenar a la red de antemano. Por otro lado si se requiere un aprendizaje en línea, el sistema resultaría más complejo y el uso de algoritmos de optimización basados en el gradiente descendiente provocaría un aumento en el tiempo de cómputo para la simulación. Para este caso, el NFC aplicado al DTC en la manera presentada proporciona un desempeño razonablemente aceptable eliminando en gran medida los efectos negativos producidos por los controladores de histéresis utilizados en el esquema original del DTC.

## 4.6. Comparación con Esquemas Convencionales.

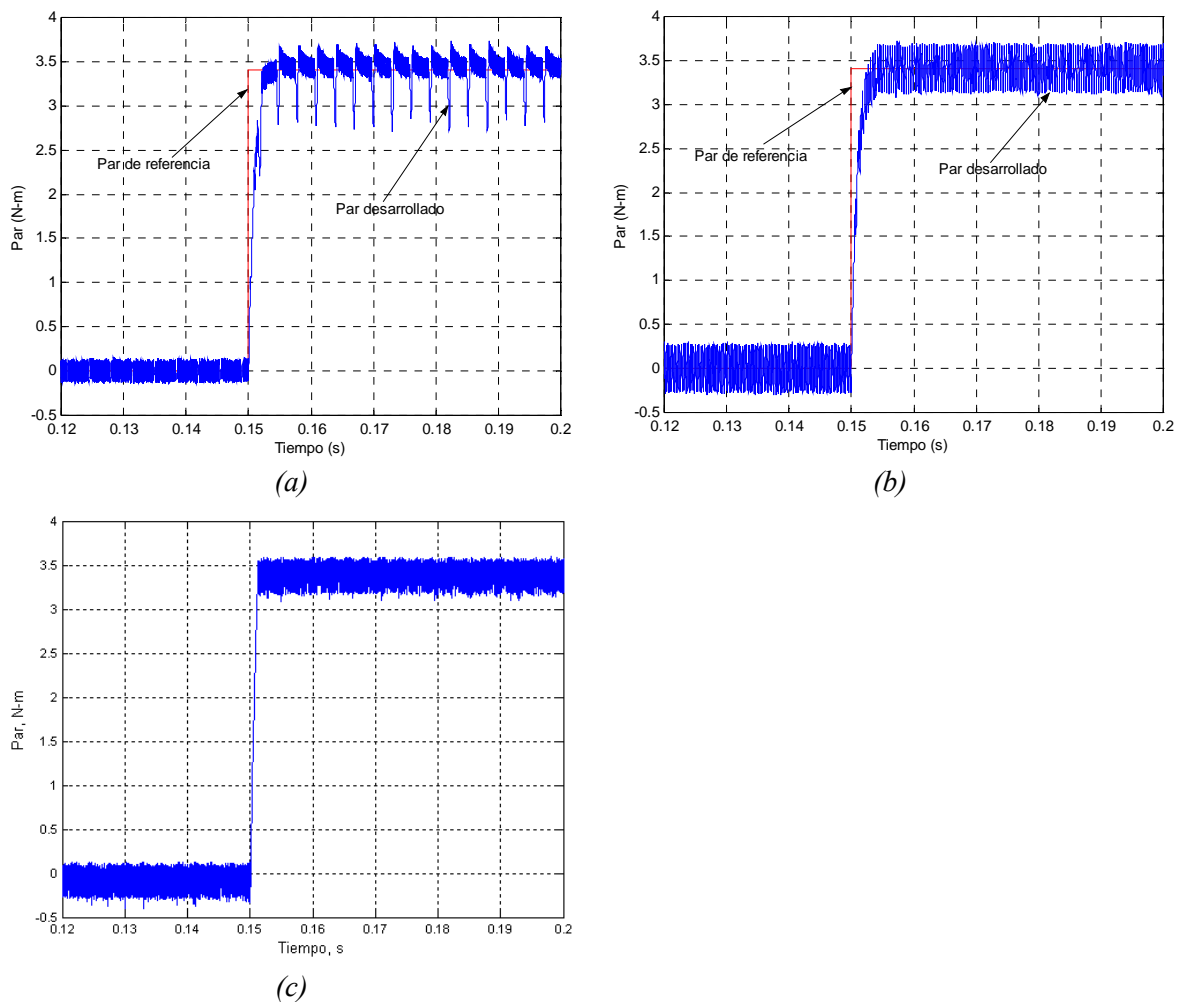
En las secciones anteriores se han presentado tres esquemas en los que los principios teóricos del DTC pueden ser implementados. Estos son el esquema clásico del DTC presentado en la figura 4.1, el DTC con modulación de vectores espaciales (DTC-SVM) mostrado en la figura 4.3 y finalmente el control del par a través del DTNFC de la figura 4.5. Cabe mencionar que estos no son los únicos esquemas que utilizan esta misma estrategia para el control directo del par y el flujo ya que como se ha mencionado, los principios del DTC permiten un desarrollo dinámico del motor de inducción eficiente y relativamente sencillo en comparación con otras técnicas [6][11][33][57].

En esta sección se realiza una comparación entre las dos técnicas presentadas que han servido de base para otros esquemas del DTC (de ahí el adjetivo de convencionales) con los resultados obtenidos del DTNFC propuesto en este trabajo. Esta comparación se realiza haciendo un análisis del desempeño del sistema al ser sometido a un cambio de tipo escalón en la señal de referencia del par. Se ha hecho un énfasis especial en la distorsión de la señal de corriente del estator y en el par electromagnético. Los parámetros del motor utilizado aparecen el apéndice D.

En primer lugar se comparan los resultados del DTC clásico presentados en [54] con los del DTNFC desarrollado en este trabajo. Dentro del esquema clásico, se sabe que la selección del vector de voltaje se realiza con el objetivo de mantener el par electromagnético y el flujo dentro de sus respectivas bandas de histéresis. Por esa razón, el ancho de banda de los controladores de histéresis tanto del par como del flujo, tienen una influencia muy importante en la operación del inversor, en la magnitud de los rizados del flujo y del par, los armónicos de corriente y sobre todo en la variación de la frecuencia de conmutación, misma que se considera como la mayor desventaja de este esquema [54][62] De acuerdo con esto, se utilizan dos simulaciones del DTC clásico con diferentes anchos de banda en los controladores de histéresis para demostrar que una pequeña variación en los controles repercute en gran medida en el comportamiento del

sistema, mientras que en el esquema del control inteligente se carece totalmente de estos problemas.

La figura 4.27 presenta la comparación en la señal del par electromagnético del motor ante un escalón en su referencia. Después de estar operando en vacío, el motor es requerido para proporcionar su par nominal de 3.41 N·m repentinamente. Aunque en las tres gráficas se observa que la respuesta del motor es muy rápida existen aspectos notables que establecen una diferencia entre el DTC clásico y el DTNFC. En la figura 4.27(a) se aprecia el DTC clásico cuando los anchos de banda de los controladores de histéresis del par y flujo,  $\Delta T$  y  $\Delta \lambda$ , es igual a 0.2 y 0.1 respectivamente. Es notable que cuando el motor trabaja a plena carga el par carece de uniformidad en su señal debido a sus intermitentes caídas. La frecuencia de conmutación para este caso oscila en 9 kHz.



**Figura 4. 27.** Par electromagnético. (a) DTC  $\Delta T=0.2$ ,  $\Delta \lambda=0.1$ ; (b) DTC  $\Delta T=0.5$ ,  $\Delta \lambda=0.01$ ; (c) DTNFC.

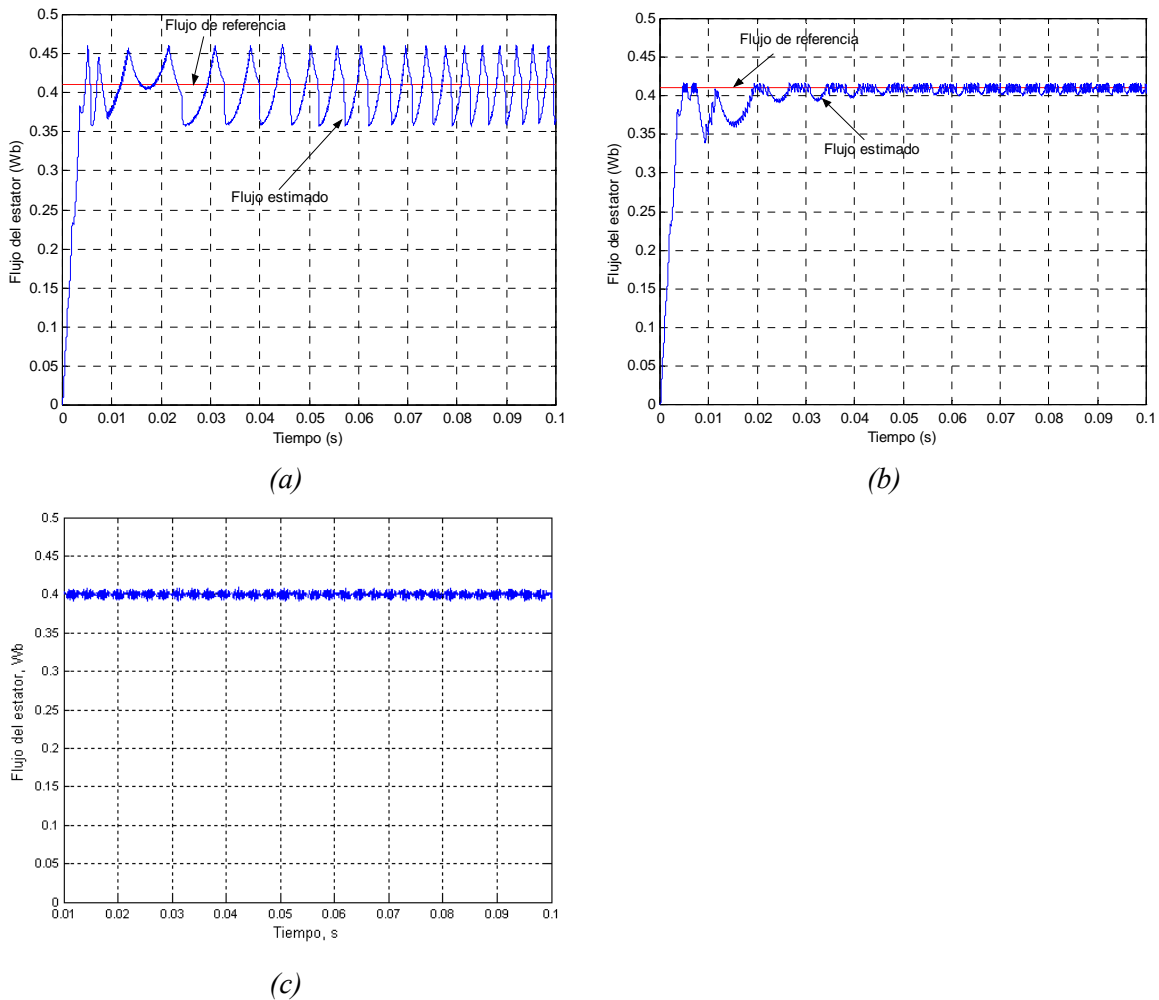
Ahora bien, en la figura 4.27(b) no existen los problemas de las caídas en el par como resultado de reducir el ancho de banda del controlador de histéresis del flujo,  $\Delta\lambda$ , a 0.01. Sin embargo esta acción usualmente repercute en un incremento en la frecuencia de conmutación del inversor la cual en este caso es compensada por un aumento en el ancho de banda del par. En este caso la frecuencia de conmutación varía entre 6 y 4 kHz. Por otro lado, la respuesta del motor a través del DTNFC se muestra en la figura 4.27(c). La respuesta es más rápida, la señal es más uniforme y la frecuencia de conmutación es constante, en este caso de 10 kHz lo cual ofrece algunas ventajas para los dispositivos de potencia [3][42].

Otro aspecto interesante es observar la amplitud del rizado en las señales del par. En la figura 4.27 también puede verse que en el caso del DTC la amplitud del rizado está directamente relacionada con el ancho de banda del controlador de histéresis del par. Un aumento en este ancho de banda provoca al mismo tiempo un incremento en el rizado de la señal de corriente eléctrica del estator. Por otro lado en el DTNFC, la amplitud del rizado se da principalmente por la magnitud del vector de flujo que a su vez depende de la magnitud del vector del voltaje aplicado al inversor a través del PWM-SVM. Considerando que desde el punto de vista de la minimización del error, el NFC provee la magnitud óptima del vector de voltaje para cada instante, la amplitud del rizado en la señal es la mínima para estas condiciones de operación.

La siguiente comparación tiene que ver con el flujo magnético del estator. La figura 4.28 presenta la amplitud del flujo bajo las mismas condiciones del DTC con respecto a los anchos de banda y las mismas circunstancias para el DTNFC. Como ya se ha mencionado, el DTNFC no tiene tablas de conmutación las cuales son inherentes en el método clásico, cuyos efectos se ven claramente en la figura 4.28(a). Debido a que justo después que el vector de flujo del estator cambia su posición de un sector a otro no hay un vector activo disponible que pueda asegurar un incremento en el par y flujo, existe una caída intermitente que incluso sale del ancho de banda establecido en su controlador. Este es un problema que persiste aún variando este ancho tal como se ve en la figura 4.28(b). Sin embargo en el DTNFC este fenómeno no existe. En la figura 4.28(c) no hay caída en el flujo ya que el NFC provee un número infinito de vectores que mediante el PWM-SVM son aproximados y de esta forma se genera en cualquier instante un vector de voltaje que minimice el error.

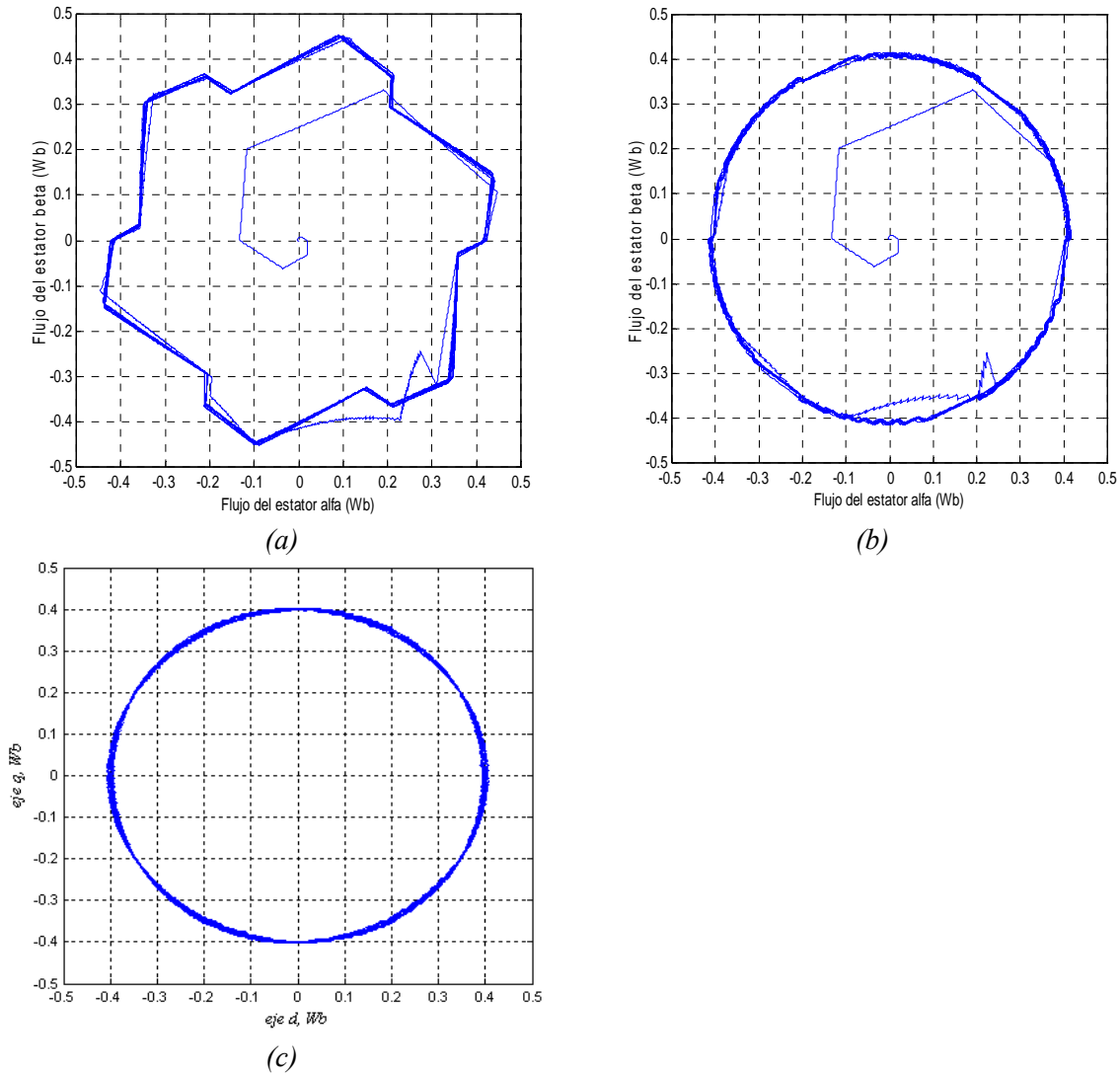
El problema de las tablas de conmutación todavía es más crítico cuando el motor opera en bajas velocidades especialmente con cargas pesadas ya que las caídas son tan intensas que la trayectoria del vector del flujo sea más hexagonal que circular así como el incremento considerable en la distorsión y deformación de la señal de corriente del estator [40][50].





**Figura 4. 28.** Amplitud del vector del flujo del estator. (a) DTC  $\Delta T=0.2$ ,  $\Delta\lambda=0.1$ ; (b) DTC  $\Delta T=0.5$ ,  $\Delta\lambda=0.01$ ; (c) DTNFC.

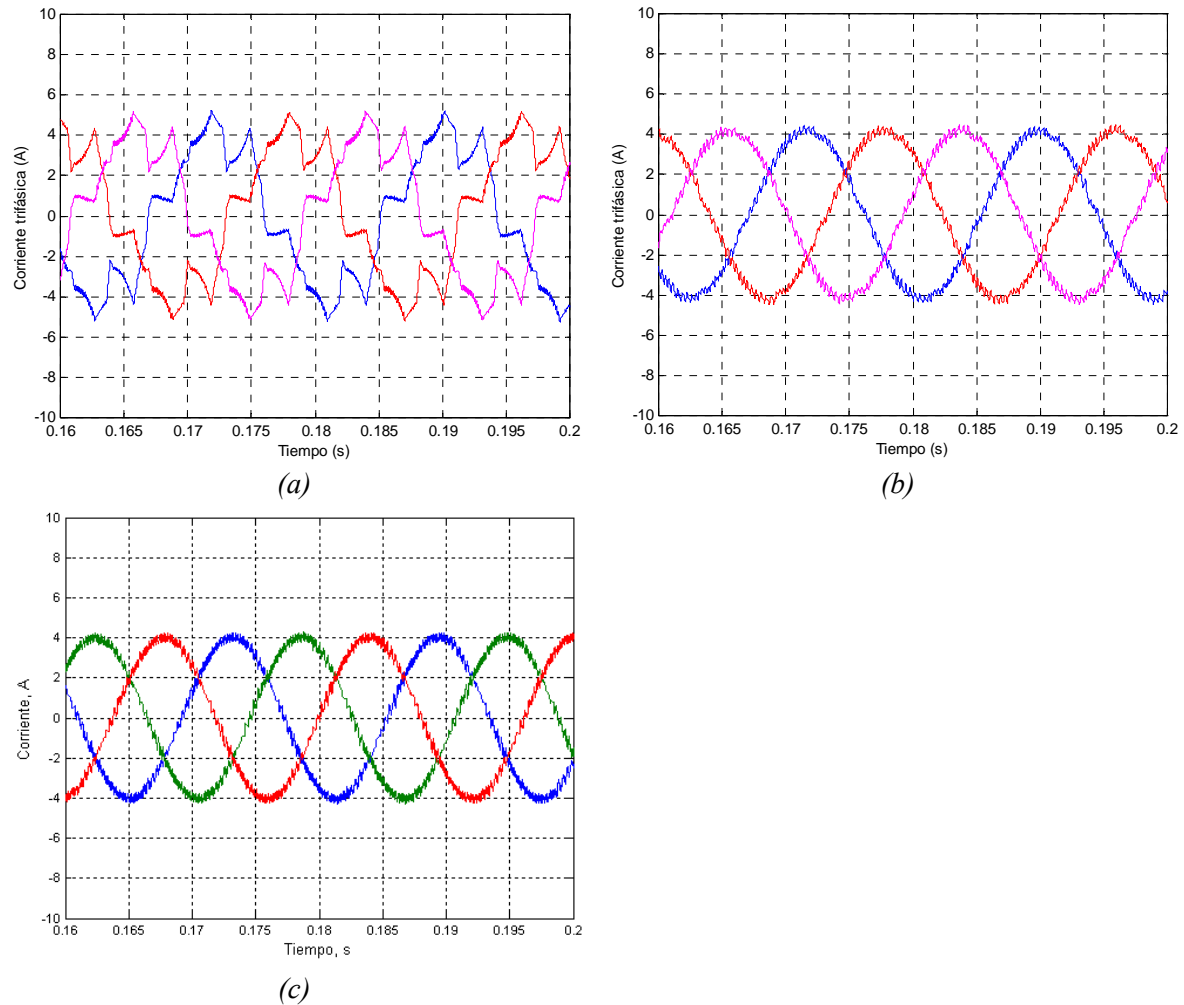
Una vista de la trayectoria que traza el vector del flujo del estator en el plano complejo, demuestra otro aspecto importante de las mejoras obtenidas por el DTNFC. La figura 4.29, muestra como la trayectoria es deformada cuando el ancho de banda del controlador de histéresis del flujo se incrementa. La figura 4.29(a) muestra el inconveniente de usar un ancho de banda mayor con respecto a la figura 4.29(b). Nuevamente, de la figura 4.29 (c) se observa que el uso del control inteligente permite que la trayectoria del vector de flujo es circular asegurando que las formas de onda de flujo y corriente sean senoidales [54].



**Figura 4.29.** Trayectoria del vector de flujo del estator. (a) DTC  $\Delta T=0.2$ ,  $\Delta\lambda=0.1$ ; (b) DTC  $\Delta T=0.5$ ,  $\Delta\lambda=0.01$ ; (c) DTNFC.

Finalmente, una comparación en las señales de corriente eléctrica puede ser un buen parámetro que indique el buen funcionamiento del sistema. La deformación y el rizado en la corriente eléctrica también dependen de los anchos de banda de los controladores de histéresis. Un incremento del ancho de banda de histéresis del flujo provoca una deformación en la onda de corriente, pero con un incremento en el ancho de banda de histéresis del par provoca un incremento en el rizado de la corriente aunque se conserva su forma senoidal [50][54]. La figura 4.30(a) presenta una onda de corriente deformada cuya forma senoidal mejora en la figura 4.30(b) gracias a la disminución del ancho de banda de histéresis de flujo. La figura 4.30(c) muestra la

corriente eléctrica del DTNFC, donde puede esperarse una forma de onda senoidal para todas las condiciones de operación gracias a la naturaleza del controlador.



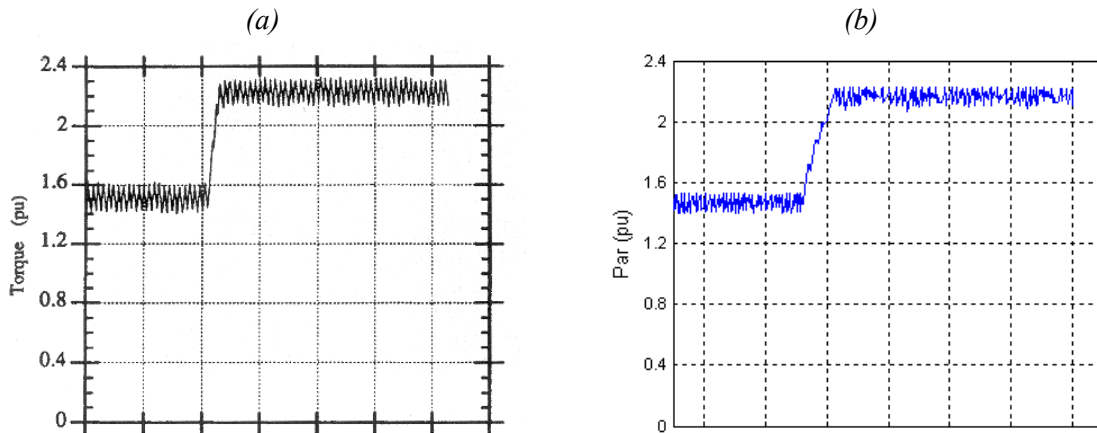
**Figura 4. 30.** Corriente eléctrica del estator. (a) DTC  $\Delta T=0.2$ ,  $\Delta\lambda=0.1$ ; (b) DTC  $\Delta T=0.5$ ,  $\Delta\lambda=0.01$ ; (c) DTNFC.

De las comparaciones anteriores queda claro que los problemas del esquema del DTC clásico debidos a los anchos de banda y las tablas de conmutación se solucionan cuando en lugar de utilizar dos controladores de histéresis se utiliza un controlador Neuro-Difuso. Por su sencillez y capacidad el NFC es muy superior a la técnica convencional.

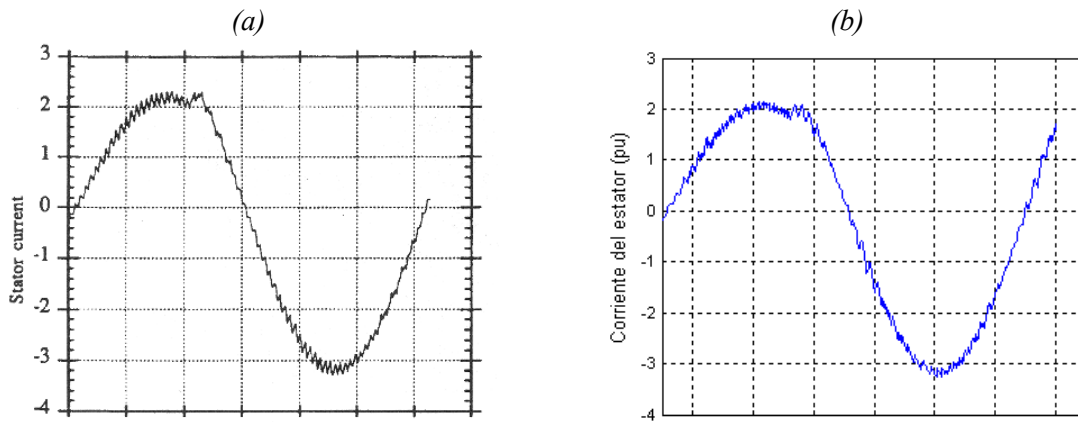
Tal como ya se ha mencionado en este trabajo, la utilización del SVM es un cambio definitivo que provee una mejora significativa en las aplicaciones con accionamientos eléctricos basados en el control directo. Es por esta razón que puede esperarse que el DTNFC se comporte de una manera muy similar al DTC-SVM (figura 4.3), sin embargo la utilización de un NFC en lugar de un controlador predictivo hace que el DTNFC sea una opción más factible no solo en la simulación sino también en la implementación. Cabe recordar que el NFC es un controlador que se fundamenta en técnicas matemáticas y operaciones lógicas muy simples y que trata de imitar a través de una manera sencilla la manera con la que el razonamiento humano toma sus decisiones y acciones de control. Esto tiene una doble ventaja, en primer lugar no necesita cálculos complicados y en segundo lugar permite la influencia externa de un experto para un mejor y más claro entendimiento del proceso.

La figuras 4.31 y 4.32 presentan comparaciones entre el DTC-SVM y el DTNFC. De la misma manera que en la comparación anterior el sistema opera bajo las mismas condiciones de operación. Los parámetros del motor utilizado en esta simulación se presentan en el apéndice D.

Las señales correspondientes al DTC-SVM fueron obtenidos de la referencia [20] mismo que fue uno de los primeros trabajos en incorporar la técnica del SVM dentro del esquema del DTC eliminando las tablas de conmutación.

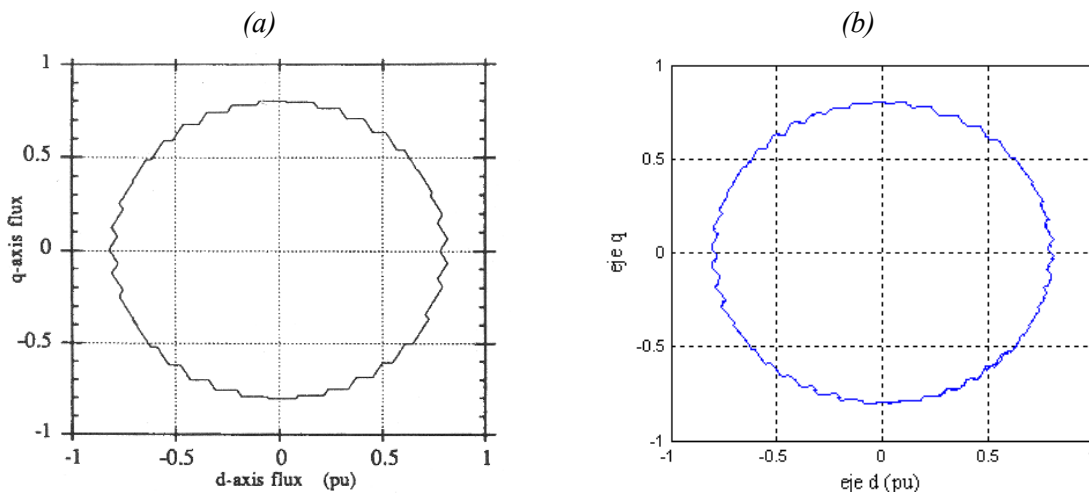


**Figura 4. 31.** Respuesta del par electromagnético ante un escalón. (a): DTC-SVM; (b): DTNFC. Eje x: 2.65ms por división.



**Figura 4. 32.** Corriente del estator ante un escalón en el par. (a): DTC-SVM; (b): DTNFC. Eje x: 2.65ms por división.

Como era de esperarse, las respuestas son muy similares. Esto da validez a los resultados del DTNFC propuesto y lo convierte en otra metodología útil para el control eficiente del par electromagnético y del flujo del motor de inducción. Finalmente, la figura 4.33 muestra la trayectoria del vector del flujo del estator en el plano complejo. Puede notarse que la trayectoria del vector es prácticamente circular lo que favorece al buen desempeño del motor.



**Figura 4. 33.** Trayectoria del vector de flujo del estator. (a) DTC-SVM; (b) DTNFC.

De acuerdo con los resultados obtenidos resulta interesante pensar que la implementación del DTNFC puede ser factible. Es importante aclarar que en el proceso de simulación, otras estructuras neuro-difusas fueron empleadas las cuales contenían en su arquitectura 4 entradas y 48 parámetros consecuentes, un número distinto de funciones de membresía, entre otras combinaciones. Sin embargo estas no demostraron una mejoría significativa en el desempeño del sistema pero si aumentaban el tiempo de cómputo requerido para ajustar los parámetros en el proceso de entrenamiento. Esto quiere decir que otra característica de las estructuras inteligentes es que su buen funcionamiento no depende de su complejidad. Como se ha analizado, el esquema del DTNFC incluyendo el NFC propuesto es relativamente sencillo de programar y por consiguiente podría ser implementado de la misma forma. Esta y otras afirmaciones se presentan en el capítulo siguiente el cual finaliza con las conclusiones y aportaciones finales de este trabajo de tesis.

## Capítulo 5

# Conclusiones

En este trabajo se ha descrito el análisis y la aplicación de una red neuronal basada en un sistema de razonamiento difuso tipo Takagi-Sugeno de primer orden, la cual forma una estructura adaptable con características propias de las metodologías de la inteligencia artificial.

El Control Neuro-Difuso Directo del Par (DTNFC) demostró tener mayor potencial que el DTC clásico gracias a su flexibilidad y a los mecanismos lógicos que lo fundamentan los cuales son caracterizados por su simplicidad.

Lo anterior se comprobó principalmente en que el DTNFC elimina por completo los problemas debidos a la frecuencia de conmutación variable, los problemas con la selección óptima del ancho de banda, las caídas intermitentes de flujo y provee al motor de un mejor desempeño en el desarrollo del par electromagnético bajo distintas condiciones de operación, una mejor trayectoria al vector del flujo del estator lo que resulta en una mejor señal de corriente eléctrica.

La aplicación del DTNFC arroja las siguientes ventajas:

1. A diferencia del esquema convencional donde se utilizan dos controladores de histéresis, el DTNFC sólo emplea uno, el cual utiliza las mismas señales de entrada por lo que el esquema no requiere de muchas modificaciones (**Figura 4.5**).
2. El filtro de Kalman utilizado como algoritmo de aprendizaje es bastante simple de incorporarlo en el programa de entrenamiento y no necesita mucho tiempo computacional para su convergencia (**Ecuación 2.48**).

3. La frecuencia del inversor es constante gracias a la utilización de la técnica de modulación de vectores espaciales lo que permite una frecuencia de conmutación constante (**Figura 4.27 explicación**).
4. La respuesta del par electromagnético y el flujo magnético del estator es más rápida a los cambios en las señales de referencia que el DTC clásico (**Figura 4.27 y 4.28**).
5. El error en estado estacionario tiende a cero gracias al aprendizaje inverso previo de la planta (**Figura 4.12**).
6. La estructura es adaptable y puede ser entrenada en línea o fuera de línea (**Figura 3.5 y 3.6**).
7. No hay distorsión en las señales de flujo y par por el cambio de sector, problema inherente del esquema convencional (**Figura 4.28**).
8. La trayectoria del vector de flujo del estator es circular en todo el dominio de control (**Figura 4.18**).
9. Las señales de corriente son senoidales en todas las condiciones de operación probadas en este trabajo (**Figura 4.11, 4.16, 4.21, 4.26 y 4.30**).
10. El desempeño del DTNFC es muy similar al realizado por el DTC-SVM aunque con una gran ventaja: es más simple ya que no requiere modelos matemáticos complicados (**Figura 4.9**).

No obstante se presentan las siguientes desventajas:

1. La fase de aprendizaje previo es crucial para la obtención de resultados aceptables lo cual lleva a la realización de un esquema de entrenamiento (**Figura 4.31 y 4.32**).
2. El conjunto de los datos de entrenamiento muchas veces no refleja el comportamiento real del sistema ya que, en el aprendizaje inverso, estos se generan de manera aleatoria.
3. La minimización del error de la red a través del filtro de Kalman, no garantiza la minimización del error global del sistema.

Al emplear un mecanismo de aprendizaje, esta estructura es capaz de producir reglas difusas que describen el comportamiento de un sistema no lineal a través de una transformación entre datos de entrada y sus correspondientes salidas, imitando así los procesos del razonamiento humano.



La arquitectura propuesta permite la transmisión de los conocimientos y experiencias de un operador humano que de manera intuitiva seleccione los parámetros de las funciones de membresía para cada valor lingüístico de los conjuntos difusos. Como se observó en las primeras simulaciones, la red neuro-difusa trabaja como una caja negra con la capacidad de identificar y modelar cualquier tipo función no lineal.

Debido a la alta flexibilidad de la red, su estructura puede tener un gran número de variantes, por ejemplo, el tipo de funciones de membresía pueden cambiarse, al mismo tiempo que los operadores lógicos que representan las operaciones lógicas de las relaciones de los conjuntos difusos. También puede aplicarse otro tipo de algoritmo de aprendizaje, como aquellos basados en la obtención del gradiente descendiente, o bien la aplicación de otras metodologías de inteligencia artificial como los algoritmos genéticos.

En el proceso de investigación de la red neuro-difusa propuesta, se realizaron otras estructuras con diversas variaciones en su arquitectura. Por ejemplo se empleó una estructura con 4 entradas y una salida, dos funciones de membresía triangulares asociadas a cada entrada, 24 parámetros antecedentes y 48 consecuentes. No obstante, aunque la complejidad de la red fue mayor, los resultados obtenidos no distaban significativamente de aquellos proporcionados por la estructura aplicada en este trabajo por lo que se demuestra que la complejidad de estos sistemas inteligentes no va acompañada de mejores resultados. Esta afirmación conduce a concluir que otra cualidad de los sistemas inteligentes es que no se requieren arquitecturas muy elaboradas para esperar buenos resultados lo que refuerza la idea que este tipo de metodologías sí tienen cierta clase de inteligencia.

Otro aspecto importante que se ha demostrado en este trabajo es el potencial que tiene esta clase de estructuras para simular los procesos de razonamiento humano en aplicaciones con sistemas de control. Para demostrar esto, la red neuro-difusa propuesta se incorporó dentro del esquema clásico del Control Directo del Par sustituyendo los bloques correspondientes a los controladores de histéresis utilizados en el esquema convencional. Previa identificación del sistema, la red es empleada para controlar las conmutaciones del inversor fuente de voltaje que a su vez alimenta eléctricamente al motor de inducción tipo jaula de ardilla.

### Recomendaciones y Direcciones Futuras.

La recomendación más importante que puede tomar lugar para un mejor desempeño de la red neuro-difusa propuesta, es la utilización de otros algoritmos de aprendizaje que puedan conducir a la red a un estado óptimo de manera más rápida. En este trabajo se resolvió el problema de identificación de parámetros mediante la aplicación de una de las técnicas de estimación por cuadrados mínimos ya que los métodos clásicos de optimización por gradiente descendiente aumentan el tiempo del proceso de entrenamiento lo cual puede ser prohibitivo en aplicaciones con tareas más complicadas.

Aunque la necesidad de encontrar mejores algoritmos de aprendizaje para las redes neuronales y sistemas difusos es hoy día un campo de investigación, aquí se recomiendan algunos que ya han sido probados [24][38]:

- ✓ Retro-propagación de segundo orden.
- ✓ Métodos clásicos de Newton.
- ✓ Métodos de Gauss-Newton.
- ✓ Procedimientos de Levenberg-Marquardt.
- ✓ Mecanismos estocásticos.
- ✓ Algoritmos genéticos.

Otra recomendación es la simplificar un poco más la arquitectura de la red propuesta mediante combinar las capas 3 y 4 y así obtener una red equivalente con solo 4 capas de tal forma que la normalización de los pesos se efectuó en la última capa.

Se recomienda también el estudio del sistema propuesto utilizando una estructura con aprendizaje no supervisado lo cual elimine el esquema de entrenamiento. O bien, otra opción sería que en la fase de entrenamiento, la red pueda aprender tal como se analizó en este trabajo, sin embargo en la fase de aplicación se utilice una estructura con aprendizaje no supervisado para que sea capaz de adaptarse en línea.

En vista de que en los últimos años los intereses en la investigación de las metodologías que integran la llamada inteligencia artificial se ha incrementado notablemente, las implementaciones de los sistemas neuro-difusos cada día son más comunes tanto en el ámbito experimental como en el comercial. Por esta razón, una vez que de manera computacional se ha demostrado un buen desempeño del sistema propuesto, la implementación a base de dispositivos electrónicos y hardware especializado sería el siguiente paso.

En lo referente al uso de la inteligencia artificial en accionamientos electrónicos de máquinas eléctricas, otras direcciones pueden tomarse. Por ejemplo, la aplicación de sistemas difuso-neuronal (difusificación de los modelos convencionales de las redes

neuronales dentro del mismo esquema del DTC, o bien sistemas híbridos difusos-neuronales los cuales permiten realizar funciones de forma separada para un mismo sistema. Más específicamente, puede utilizarse un controlador difuso para el control del ángulo del vector de voltaje de referencia, mientras que una red neuronal puede servir como estimador del par y/o flujo del estator para el cálculo del error. De esta manera el entrenamiento únicamente sería para la red neuronal simplificando este proceso. O bien, además del sistema propuesto en este trabajo, pueden emplearse dos bloques con sistemas de control difusos para el control de velocidad y posición a través del DTC y así aumentar la funcionalidad del esquema de control.

Se recomienda investigar el efecto que puede causar la saturación y las pérdidas de hierro en el motor de inducción en el controlador y en el sistema en general.

## Referencias

- [1] Bakhshai, A., Espinoza, J. Joós, G., Jin, H. *A Combined Artificial Neural Network and DSP Approach to the Implementation of Space Vector Modulation Techniques*. IEEE 0-7803-3544-9/1996.
- [2] Ba-Razzou, A., Chériti, A., Olivier, G., Sicard, P. *Field Oriented Control of Induction Motors Using Neural-Network Decouplers*. IEEE Transactions on Power Electronics, Vol. 12, No. 4, July 1997.
- [3] Bose, B. K. *Power Electronics and AC Drives*. Englewood Cliffs, NJ, Prentice Hall, 1986.
- [4] Buja, G. *A New Control Strategy of the Induction Motor Drives*. IEEE Ind. Electron. Soc. Newslett, Vol 45, pp. 14-16. Dec. 1998.
- [5] Buja, G., Casadei, D., Serra, G. *DTC-Based Strategies for Induction Motor Drives*. Proceedings of the ISIE Conference, 1997, pp. TU2 .
- [6] Casadei, D., Serra, G. *Implementation of a Direct Torque Control Algorithm for Induction Motor Drives*. IEEE Transactions on Power Electronics, Vol. 15, pp. 769-777, July. 2000.
- [7] Chalam, V., *Adaptive Control Systems, Techniques and Applications*. Dekker. New York 1987.
- [8] Chang, E. S., Chen, S., Mulgrew, B. *Efficient Computational Schemes for the Orthogonal Least Squares Algorithm*. IEEE Transactions on Signal Processing, Vol. 43, No. 1, January 1995.
- [9] Chapman, S. *Electric Machinery Fundamentals*. McGraw Hill. 1999.
- [10] Chen, S. Cowan, C. F., Grant, P. M. *Orthogonal Least Squares Learning for Radial Basis Function Network*. IEEE Transactions on Neural Networks, Vol. 2, No. 2, March 1991.
- [11] Depenbrock, M. *Direct Self Control of Inverter-Fed Induction Machines*. IEEE Trans. Power Electron. Vol. 3, pp. 420-429, Oct. 1988.
- [12] *Diccionario Enciclopédico Ilustrado Océano Uno*. Ediciones Océano. Barcelona España, 1992.
- [13] Dryankov, D. *An Introduction to Fuzzy Control*. Springer Verlag. 1993.
- [14] Elfilali, B. *Utilización del Filtro de Kalman como Estimador de la Velocidad en el Control Vectorial de Motores de Inducción*. Tesis de Maestría. IPN SEPI-ESIME. México D. F. 2001.
- [15] *Fuzzy Logic Toolbox User's Guide in Control Systems*: The MathWorks, Inc. 1995 – 2000.
- [16] Gama, M. & Romero, D. *Análisis de un Algoritmo Híbrido de Aprendizaje para una Red Neuro-Difusa Adaptable*. Séptimo Congreso Nacional de Ingeniería Electromecánica y de Sistemas. IPN-SEPI ESIME, Noviembre 2003.
- [17] Gama, M. & Romero, D. *Representación Simplificada de un Sistema Rectificador-Inversor con Modulador de Vectores Espaciales para Motores de Inducción*. IEEE RVP-AI/2003, Julio 2003.

- [18] Grabowski, P. Z. *Direct-Torque Neuro-Fuzzy Control of Induction Motor Drive*. Proc. IEEE IECON, 1997.
- [19] Grabowski, P. Z., Kazmierkowski, M., Bose, B., Blaabjerg. *A Simple Direct-Torque Neuro-Fuzzy Control of PWM-Inverter-Fed Induction Motor Drive*. IEEE Transactions on Industrial Electronics, Vol. 47, No. 4, August 2000.
- [20] Hableter, T., Profumo, F., Pastorelli, M., Tolbert, L. *Direct Torque Control of Induction Machines Using Space Vector Modulation*. IEEE Transactions Ind. Applications, vol. 28, pp. 1045-1053, Sep./Oct. 1992.
- [21] Harris, C. J. *Advances in Intelligent Control*. Taylor & Francis Ltd. 1994.
- [22] Hoel, P., Port, S., Stone, C. *Introduction to Statistical Theory*. Houghton Mifflin Company. Boston. 1971.
- [23] Jang, J.-S. R., ANFIS: *Adaptive-Networks-Based Fuzzy Inference System*. IEEE Transactions on Systems, Man and Cybernetics, Vol. 23, No. 3, May/June 1993.
- [24] Jang, J.-S. R., *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice Hall. 1997.
- [25] Jang, J.-S. R., *Neuro-Fuzzy Modeling and Control*. Proceedings of the IEEE, Vol. 83, No. 3, pp 378-405, March 1985.
- [26] Jang, J.-S. R., *Self-Learning Fuzzy Controllers Based on Temporal Back Propagation*. IEEE Transactions on Neural Networks, Vol. 3, No. 5, September 1992.
- [27] Kailath, T. *Linear Systems*. Prentice Hall. New York. 1980.
- [28] Kim, M.-H., Kim, N.-H., Kim, M., Kim, D.-H. *An Induction Motor Position Control System with Direct Torque Control*. IEEE 0-7803-7090-2/2001.
- [29] Kosko, B. *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Prentice Hall, Upper Saddle River, NJ, 1991.
- [30] Krause, P. C. *Analysis of Electric Machinery and Drive Systems*. Wiley-Interscience, 2002.
- [31] Krause, P. C. *Method of Multiple Reference Frames Applied to the Analysis of Symmetrical Induction Machinery*. IEEE Transactions of Power Apparatus and Systems. Vol. PAS-87, pp 218-227, January, 1968.
- [32] Kumbala, K. & Jamshidi, M. *Real Time Control of a Direct Drive Motor by a Learning Neuro-Fuzzy Controller*. IEEE 0-7803-4053-1/1997.
- [33] Lai, Y.-S. & Chen, J.-H. *A New Approach to Direct Torque Control of Induction Motor Drives for Constant Inverter Switching Frequency and Torque Ripple Reduction*. IEEE Transactions on Energy Conversion, Vol. 16, No. 3, September 2001.
- [34] Lee, C.-C., *Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part I*. IEEE Transactions on Systems, Man and Cybernetics, Vol. 20, No. 2, March/April 1990.
- [35] Lee, C.-C., *Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part II*. IEEE Transactions on Systems, Man and Cybernetics, Vol. 20, No. 2, March/April 1990.
- [36] Lewis, P., Yang, C. *Basic Control Systems Engineering*. Prentice Hall, 1997.
- [37] Lin, B.-R. & Hofy, R.G. *Power Electronics Converter Control Based on Neural Network and Fuzzy Logic Methods*. IEEE 0-7803-1243-0, 1993.
- [38] Lin, C. T., Lee, C. S. G. *Neural Fuzzy System: A Neuro-Fuzzy Synergism to Intelligent Systems*. Prentice Hall. 1996.
- [39] Lin, J.-N. & Song, S.-M. *A Novel Fuzzy Neural Network for the Control of Complex Systems*. IEEE 0-7803-1901-X/1994.

- 
- [40] Marino, P., D'Incecco, M., Visciano, N. *A Comparison of Direct Torque Control Methodologies for Induction Motor*. 2001 IEEE Porto Power Tech Conference, September, Porto, Portugal
- [41] Mohamadiant, M. Nowicki, E. P. Salmon, J. C. *A Neural Network Controller for Indirect Field Orientation Control*. IEEE 0-7803-3008-0/1995.
- [42] Mohan, N., Undeland, T., Robbins, W.P. *Power Electronics. Converters, Applications and Design*. John Wiley & Sons, USA 2003.
- [43] Montgomery, D. *Introduction to Linear Regression Analysis*. John Wiley & Sons, Inc. USA. 1982.
- [44] Norgaard, M., *Neural Networks for Modeling and Control of Dynamic Systems*: Springer-Verlag. London 2000.
- [45] O'Kelly, D. & Simmons, S. *Generalized Electrical Machine Theory*, McGraw Hill, 1968.
- [46] Ogata, K. *Modern Control Engineering*. Fourth edition. Prentice Hall. NJ 2003.
- [47] Ong, C-M. *Dynamic Simulation of Electric Machinery*, Prentice Hall, 1998.
- [48] Passino, K. M. *Fuzzy Control*. Addison Wesley Longman, Inc., 1998.
- [49] Pinto, J. O. P., Bose, B. K., Borges da Silva, L., Kazmierkowsky, M. P. *A Neural-Network-Based Controller for Voltage-Fed Inverter Induction Motor Drive*. IEEE Transactions on Ind. Applications, Vol. 36. No. 6. Nov./Dec. 2000.
- [50] Ponce, P. *Control Directo del Par de un Motor de Inducción sin Sensor de Velocidad*. Tesis de Doctorado. IPN SEPI-ESIME. México D. F. 2002.
- [51] Ramirez, J. *Control Directo del Par de un Motor de Inducción Aplicando una Técnica de Modulación de Ancho de Pulso con Vectores Espaciales*. Tesis de Maestría. IPN SEPI-ESIME. México D. F. 2002.
- [52] Rivera, A. *Control de un Motor de Inducción utilizando el Método de Auto-control Directo de Flujo y Par*. Tesis de Maestría. IPN SEPI-ESIME. México D. F. 2001.
- [53] Ruan, D. *Intelligent Hybrid Systems: Fuzzy logic, Neural Networks and Genetic Algorithms*. Kluwer Academic Publishers. 1997.
- [54] Sampe, F. *Análisis del Control Directo del Par de un Motor de Inducción*. Tesis de Maestría. IPN SEPI-ESIME. México D. F. 2002.
- [55] Soloway, D. & Haley, P. *Neural/Generalized Predictive Control, a Newton-Raphson Implementation*. In Proceedings of the 11th IEEE International Symposium on Intelligent Control, pp. 277-282. 1996.
- [56] Takahashi, I., & Noguchi, T. *A New Quick-Response and High-Efficiency Strategy of an Induction Motor*. IEEE Transactions of Ind. Applications, vol. 22, no.7, pp. 820-827, 1986.
- [57] Takahashi, I., & Ohmori, Y. *High Performance Direct Torque Control of an Induction Motor*. IEEE Transactions on Industry Applications, Vol. 25, No. 2. March/April 1989.
- [58] Tang, L. & Rahman, M. F. *A New Direct Torque Control Strategy for Flux and Torque Ripple Reduction for Induction Motors Drive -- A Matlab/Simulink Model*. IEEE 0-7803-7091-01/2001.
- [59] Telford, D., Dunningan, M. W., Williams, B. W. *A Comparison of Vector Control and Direct Torque Control of an Induction Machine*. IEEE 0-7803-5692-6/2000.
- [60] Trzynadlowski, A. *Control of Induction Motors*. Academic Press. 2001.

- 
- [61] Van Der Broeck, H., Skudelny, H. Ch., Stanke, G. *Analysis and Realization of a Pulsewidth Modulator Based on Voltage Space Vectors*. IEEE Transactions on Ind. Applications, vol. 24. No. 1. Jan/Feb. 1988.
  - [62] Vas, P. *Sensorless Vector and Direct Torque Control*. Oxford University Press. 1998.
  - [63] Wang, L.-X. & Mendel, J. M. *Fuzzy Basis Functions, Universal Approximation, and Orthogonal Least-Squares Learning*. IEEE Transactions on Neural Networks, Vol. 3, No. 5, September 1992.
  - [64] Youm, J.-H. & Kwon, B.-H. *An Effective Software Implementation of Space-Vector Modulation*. IEEE Transactions on Industrial Electronics, Vol. 46, No. 4, August 1999.
  - [65] Zadeh, L. *Linear System Theory*. McGraw Hill. New York. 1963.
  - [66] Zhenyu, Y. *Space-Vector PWM with TMS320C24x/F24x using Hardware and Software Determined Switching Patterns*. Texas Instruments. DSP Solutions. Application Report SPRA524. March 1999.

# **APENDICES**



## Apéndice A

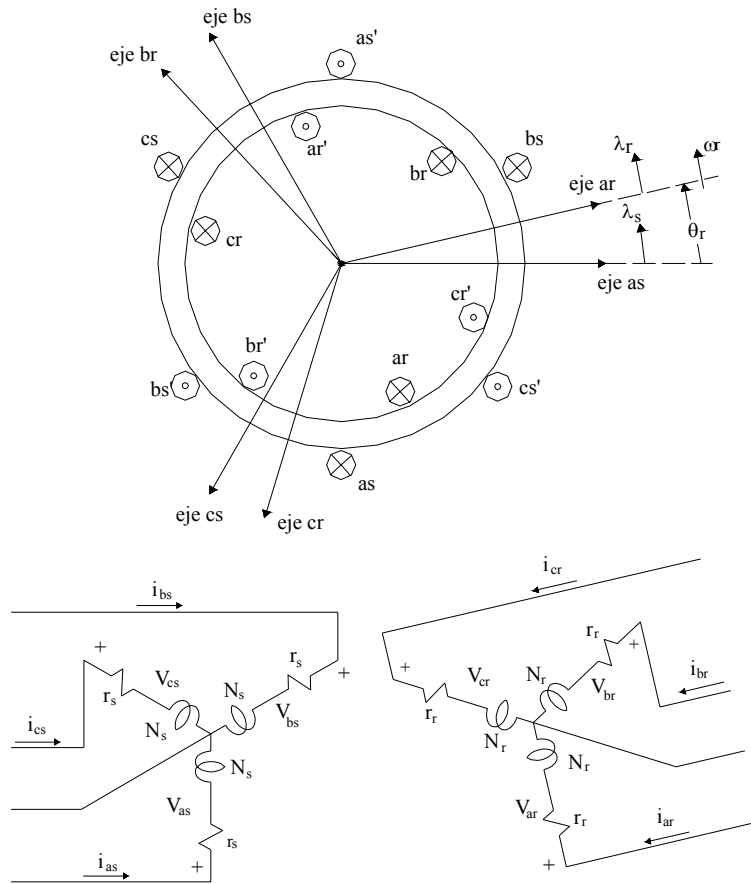
# Modelo Matemático del Motor de Inducción

La máquina de inducción es usada en una amplia variedad de aplicaciones. Como motor, esta máquina ha inundado la industria con sus múltiples usos en virtud de su capacidad para mover pequeñas y grandes cargas. Como consecuencia de esto, a través de los años, el análisis matemático de la máquina asíncrona, como también se le conoce, ha dado como resultado que este dispositivo se implante en sustitución de otros como el motor de c.d. [60].

Dentro del modelo matemático del motor encontramos que sus variables están acopladas entre sí lo cual hace que su análisis sea complicado. Un cambio de variables es en ocasiones una técnica apropiada para reducir dicha complejidad en la solución de las ecuaciones diferenciales. Este cambio consiste en transformar las variables reales refiriéndolas a un marco de referencia el cual rota a una velocidad angular arbitraria.

Aunque el comportamiento de una máquina de inducción simétrica puede describirse en cualquier marco de referencia, existen tres que son los más comúnmente utilizados: el marco de referencia estacionario, el marco de referencia del rotor y el marco de referencia síncrono [31]. Las ecuaciones de voltaje para cada uno de estos marcos de referencia se obtienen a partir de las ecuaciones en el marco de referencia arbitrario y mediante asignar la velocidad angular ( $\omega$ ) apropiada. Esto es  $\omega = 0$  para el estacionario,  $\omega = \omega_r$  para el rotor, y  $\omega = \omega_b$  para el síncrono. Generalmente, las condiciones de operación determinan el marco de referencia más conveniente para el análisis y/o simulación.

El arreglo de una máquina de inducción simétrica de 2 polos, tres fases, conexión estrella se muestra en la figura A.1. Los devanados del estator son idénticos, concentrados y distribuidos, desplazados  $120^\circ$  uno respecto del otro.



**Figura A. 1.** Motor de inducción trifásico conectado en estrella, 2 polos 3 fases.

En casi todas sus aplicaciones, incluyendo en esta, se considera al motor de inducción teniendo los devanados del estator conectados a una fuente externa de voltaje trifásico balanceado y los devanados del rotor en cortocircuito. Asimismo se sabe que los devanados del estator son idénticos tanto en resistencia  $r_s$  como en el número de vueltas  $N_s$ . Se asume que el entrehierro de la máquina es uniforme. En este trabajo los efectos de la saturación magnética o las pérdidas producidas por efectos de histéresis y corrientes eddy son despreciados.

El principio de operación del motor es sencillo. Con corrientes trifásicas balanceadas fluyendo a través de los devanados del estator, se crea una fuerza magnetomotriz (FMM) rotatoria en el entrehierro el cual gira a una velocidad determinada por la frecuencia de las corrientes del estator y el número de polos de la máquina. El movimiento relativo del rotor con respecto al campo magnético del estator

produce un voltaje inducido en el rotor. La velocidad del rotor con respecto a este campo magnético produce corrientes en los devanados cortocircuitados del rotor. A su vez, el flujo de corriente en el rotor produce un campo magnético rotatorio. Debido a que estos campos magnéticos tratan de alinearse en el entrehierro, se produce un par [9]. Sin embargo, existe un límite finito para la velocidad del motor. Si el rotor del motor de inducción estuviera rotando a la velocidad síncrona, el movimiento relativo del rotor con respecto al campo magnético del estator sería cero o estacionario y no habría voltaje inducido, por lo tanto no habría corriente en el rotor ni campo magnético y como consecuencia no habría par y el motor se frenaría como resultado de las pérdidas por rozamiento. En operación normal, los campos magnéticos del rotor y estator rotan conjuntamente a velocidad síncrona mientras que el rotor en sí gira a una velocidad menor.

### A.1. Modelo del Motor en Variables Reales.

Las inductancias de los devanados de la máquina de inducción pueden ser expresadas similarmente a las de la máquina síncrona de polos salientes. Sin embargo en este caso, el entrehierro de la máquina es uniforme, por lo tanto las variaciones en las inductancias propias del rotor y estator en función de la posición del rotor no existen. De tal forma que la matriz de inductancias propias del estator y rotor pueden expresarse

Inductancias propias del estator:

$$L_{asas} = L_{bsbs} = L_{cscs} = L_{ls} + L_{ms} \quad (\text{A. 1})$$

Inductancias mutuas entre devanados del estator:

$$L_{asbs} = -\frac{1}{2}L_{ms} \quad (\text{A. 2})$$

$$L_s = \begin{bmatrix} L_{ls} + L_{ms} & -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & L_{ls} + L_{ms} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} & L_{ls} + L_{ms} \end{bmatrix} \quad (\text{A. 3})$$

Inductancias propias del rotor:

$$L_{arar} = L_{brbr} = L_{crcr} = L_{lr} + L_{mr} \quad (\text{A. 4})$$

Inductancias mutuas entre devanados del rotor:

$$L_{arbr} = -\frac{1}{2}L_{mr} \quad (\text{A. 5})$$

$$L_r = \begin{bmatrix} L_{lr} + L_{mr} & -\frac{1}{2}L_{mr} & -\frac{1}{2}L_{mr} \\ -\frac{1}{2}L_{mr} & L_{lr} + L_{mr} & -\frac{1}{2}L_{mr} \\ -\frac{1}{2}L_{mr} & -\frac{1}{2}L_{mr} & L_{lr} + L_{mr} \end{bmatrix} \quad (\text{A. 6})$$

Las inductancias mutuas entre devanados del rotor y estator están en función de la posición angular [30]. Se pueden definir como:

$$L_{asar} = L_{bsbr} = L_{cscr} = L_{sr} \text{Cos} \theta_r \quad (\text{A. 7})$$

$$L_{asbr} = L_{sr} \text{Cos}(\theta_r + \frac{2\pi}{3}) \quad (\text{A. 8})$$

$$L_{ascr} = L_{sr} \text{Cos}(\theta_r - \frac{2\pi}{3}) \quad (\text{A. 9})$$

$$L_{sr} = L_{sr} \begin{bmatrix} \text{Cos} \theta_r & \text{Cos}(\theta_r + \frac{2\pi}{3}) & \text{Cos}(\theta_r - \frac{2\pi}{3}) \\ \text{Cos}(\theta_r - \frac{2\pi}{3}) & \text{Cos} \theta_r & \text{Cos}(\theta_r + \frac{2\pi}{3}) \\ \text{Cos}(\theta_r + \frac{2\pi}{3}) & \text{Cos}(\theta_r - \frac{2\pi}{3}) & \text{Cos} \theta_r \end{bmatrix} \quad (\text{A. 10})$$

En las ecuaciones anteriores,  $L_{ls}$  y  $L_{ms}$  son, respectivamente, las inductancias de dispersión y magnetizante de los devanados del estator;  $L_{lr}$  y  $L_{mr}$  son, respectivamente, las inductancias de dispersión y magnetizante de los devanados del rotor. La inductancia  $L_{sr}$  es la amplitud de las inductancias mutuas ente los devanados del estator y rotor.

Para estas y posteriores ecuaciones se considera que los valores de los parámetros del rotor están referidos al estator. Las ecuaciones de voltaje para cada devanado son [47]:

$$v_{as} = r_s i_{as} + \frac{d\lambda_{as}}{dt} \quad (\text{A. 11})$$

$$v_{bs} = r_s i_{bs} + \frac{d\lambda_{bs}}{dt} \quad (\text{A. 12})$$

$$v_{cs} = r_s i_{cs} + \frac{d\lambda_{cs}}{dt} \quad (\text{A. 13})$$

$$v_{ar} = r_r i_{ar} + \frac{d\lambda_{ar}}{dt} \quad (\text{A. 14})$$

$$v_{br} = r_r i_{br} + \frac{d\lambda_{br}}{dt} \quad (\text{A. 15})$$

$$v_{cr} = r_r i_{cr} + \frac{d\lambda_{cr}}{dt} \quad (\text{A. 16})$$

donde  $r_s$  y  $r_r$  es la resistencia de cada devanado de fase del estator y rotor respectivamente. Los enlaces de flujo pueden escribirse de la misma forma que para  $\lambda_{as}$

$$\lambda_{as} = L_{asas} i_{as} + L_{asbs} i_{bs} + L_{ascs} i_{cs} + L_{asar} i_{ar} + L_{asbr} i_{br} + L_{ascr} i_{cr} \quad (\text{A. 17})$$

Matricialmente,

$$\begin{bmatrix} \lambda_{abcs} \\ \lambda_{abcr} \end{bmatrix} = \begin{bmatrix} L_s & L_{sr} \\ (L_{sr})^T & L_r \end{bmatrix} \begin{bmatrix} i_{abcs} \\ i_{abcr} \end{bmatrix} \quad (\text{A. 18})$$

$$\begin{bmatrix} v_{abcs} \\ v_{abcr} \end{bmatrix} = \begin{bmatrix} r_s + pL_s & pL_{sr} \\ p(L_{sr})^T & r_r + pL_r \end{bmatrix} \begin{bmatrix} i_{abcs} \\ i_{abcr} \end{bmatrix} \quad (\text{A. 19})$$

La ecuación electromecánica está dada por:

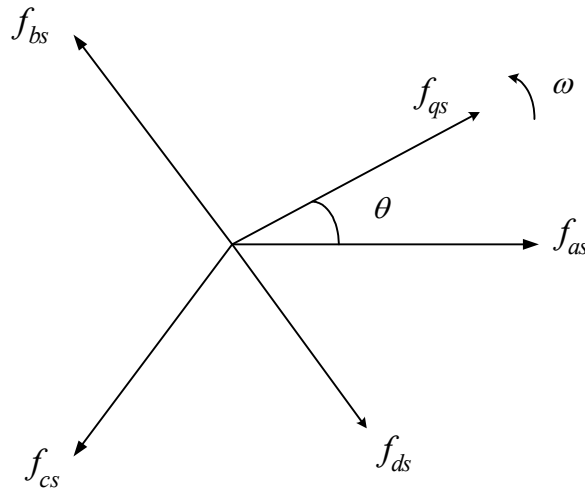
$$T_e = J \left( \frac{2}{P} \right) p \omega_r + T_m \quad (\text{A. 20})$$

Las expresiones anteriores forman el modelo del motor de inducción dados en variables reales de la máquina. Al igual que la máquina síncrona, nuevamente se observa la complejidad del modelo debido a las inductancias variantes en la matriz de inductancias mutuas.

## A.2. Modelo de la Máquina en el Marco de Referencia Arbitrario.

Para evitar manejar en el modelo matemático los términos que incluyen las inductancias variantes en el tiempo, se utiliza las técnicas de Park para hacer un cambio de variables y referirlas a un marco de referencia [30][45]. El cambio de variables se realiza mediante aplicar la transformación de las variables reales  $abc$  a un marco de referencia arbitrario en los componentes ficticios  $qd0$ .

La idea de considerar a las variables de la máquina como vectores o vectores espaciales para su análisis es algo práctico y conveniente, ya que las estrategias más actuales de control y simulación utilizadas se basan precisamente en esta [62]. El cambio de variables o transformación se formula mediante referir las variables de los circuitos estacionarios y rotatorios a dos vectores espaciales ficticios  $d$  y  $q$  que se mueven a una velocidad angular arbitraria. La figura A.2 muestra esta transformación obtenida mediante simples relaciones trigonométricas.



**Figura A. 2.** Transformación de las variables de circuitos estacionarios a los vectores espaciales  $dq$ .

De la figura anterior se desprende que la transformación puede expresarse

$$f_{qd0s} = K_s f_{abc} \quad (\text{A. 21})$$

donde

$$(\mathbf{f}_{qd0s})^T = [f_{qs} \quad f_{ds} \quad f_{0s}] \quad (\text{A. 22})$$

$$(\mathbf{f}_{abcs})^T = [f_{as} \quad f_{bs} \quad f_{cs}] \quad (\text{A. 23})$$

$$\mathbf{K}_s = \frac{2}{3} \begin{bmatrix} \cos \theta & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ \sin \theta & \sin(\theta - \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (\text{A. 24})$$

donde  $\theta$  se obtiene al integrar la velocidad angular de los vectores espaciales  $dq$ . El subíndice  $s$  refiere a las variables del estator.

Prácticamente puede o no haber ciertas ventajas de utilizar el modelo de la máquina de inducción usando un marco de referencia u otro cuando se simula bajo condiciones balanceadas [30]. Sin embargo hay situaciones donde puede ser deseable representar las variables de las fases del estator con sus valores reales ( $abc$ ). En este caso, el marco de referencia estacionario ofrece ventajas ya que los valores reales se relacionan aritméticamente con los valores referidos a los ejes  $dq$  [31]. En este trabajo se utiliza dicha ventaja para presentar el modelo del motor en el marco de referencia estacionario donde los vectores  $dq$  permanecen estáticos ( $\omega=0$ ). En tal caso y teniendo en cuenta que consideramos al motor como un sistema trifásico balanceado las variables de las componentes  $f_\theta$  son cero, la ecuación (A.24) se puede expresar

$$\mathbf{K}_s = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & \frac{1}{2} \\ 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \quad (\text{A. 25})$$

y su inversa

$$(\mathbf{K}_s)^{-1} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \quad (\text{A. 26})$$

Cualquier vector-variable (voltajes, corrientes, flujos, FMM, etc) puede ser obtenido a partir de las ecuaciones anteriores.

Aplicando la transformación anterior a las ecuaciones de voltaje expresadas en (A.11)-(A.16) se eliminan los términos variantes y resulta en el modelo matemático de la máquina de inducción en el marco de referencia arbitrario [47]. De tal forma que:

$$v_{qd0s} = r_s i_{qd0s} + \omega \lambda_{dqs} + p \lambda_{qd0s} \quad (\text{A. 27})$$

$$v_{qd0r} = r_r i_{qd0r} + (\omega - \omega_r) \lambda_{dqr} + p \lambda_{qd0r} \quad (\text{A. 28})$$

Por otra parte como los parámetros de la máquina casi siempre son dados en ohms, por ciento o en por unidad es conveniente expresar las ecuaciones de voltaje y enlaces de flujo en términos de reactancias en vez de inductancias. De acuerdo a lo anterior y expandiendo (A.27) y (A.28), se tiene [30]:

$$v_{qs} = r_s i_{qs} + \frac{\omega}{\omega_b} \lambda_{ds} + \frac{p}{\omega_b} \lambda_{qs} \quad (\text{A. 29})$$

$$v_{ds} = r_s i_{ds} - \frac{\omega}{\omega_b} \lambda_{qs} + \frac{p}{\omega_b} \lambda_{ds} \quad (\text{A. 30})$$

$$v_{0s} = r_s i_{0s} + \frac{\omega}{\omega_b} \lambda_{0s} \quad (\text{A. 31})$$

$$v_{qr} = r_r i_{qr} + \left( \frac{\omega - \omega_r}{\omega_b} \right) \lambda_{dr} + \frac{p}{\omega_b} \lambda_{qr} \quad (\text{A. 32})$$

$$v_{dr} = r_r i_{dr} + \left( \frac{\omega - \omega_r}{\omega_b} \right) \lambda_{qr} + \frac{p}{\omega_b} \lambda_{dr} \quad (\text{A. 33})$$

$$v_{0r} = r_r i_{0r} + \frac{p}{\omega_b} \lambda_{0r} \quad (\text{A. 34})$$

donde  $\omega_b$  es la velocidad angular base usada para calcular las reactancias inductivas. Los enlaces de flujo están dados por:

$$\lambda_{qs} = X_{ls} i_{qs} + X_M (i_{qs} + i_{qr}) \quad (\text{A. 35})$$

$$\lambda_{ds} = X_{ls} i_{ds} + X_M (i_{ds} + i_{dr}) \quad (\text{A. 36})$$

$$\lambda_{0s} = X_{ls} i_{0s} \quad (\text{A. 37})$$



$$\lambda_{qr} = X_{lr}i_{qr} + X_M(i_{qs} + i_{qr}) \quad (\text{A. 38})$$

$$\lambda_{dr} = X_{lr}i_{dr} + X_M(i_{ds} + i_{dr}) \quad (\text{A. 39})$$

$$\lambda_{0r} = X_{lr}i_{0r} \quad (\text{A. 40})$$

Sustituyendo (A.35)-(A.40) en la ecuaciones de voltaje (A.29)-(A.34), y teniendo a los enlaces de flujo como variables de estado llegamos al modelo de la máquina de inducción en el marco de referencia arbitrario.

Para simular el comportamiento de la máquina de inducción simétrica en el marco de referencia arbitrario es más conveniente reacomodar las ecuaciones de tal forma que puedan considerarse los enlaces de flujo como variables de estado. En las ecuaciones de voltaje es necesario poner las corrientes en términos de los flujos magnéticos correspondientes, de tal manera que de (A.35)-(A.40) se tiene:

$$i_{qs} = \frac{1}{X_{ls}}(\lambda_{qs} - \lambda_{mq}) \quad (\text{A. 41})$$

$$i_{ds} = \frac{1}{X_{ls}}(\lambda_{ds} - \lambda_{md}) \quad (\text{A. 42})$$

$$i_{0s} = \frac{1}{X_{ls}}\lambda_{0s} \quad (\text{A. 43})$$

$$i_{qr} = \frac{1}{X_{lr}}(\lambda_{qr} - \lambda_{mq}) \quad (\text{A. 44})$$

$$i_{dr} = \frac{1}{X_{lr}}(\lambda_{dr} - \lambda_{md}) \quad (\text{A. 45})$$

$$i_{0r} = \frac{1}{X_{lr}}\lambda_{0r} \quad (\text{A. 46})$$

donde

$$\lambda_{mq} = X_M(i_{qs} + i_{qr}) \quad (\text{A. 47})$$

$$\lambda_{md} = X_M(i_{ds} + i_{dr}) \quad (\text{A. 48})$$

Empleando (A-41)-(A-46) en (A-47) y (A-48) se tiene que

$$\lambda_{mq} = X_{aq} \left( \frac{\lambda_{qs}}{X_{ls}} + \frac{\lambda_{qr}}{X_{lr}} \right) \quad (\text{A. 49})$$

$$\lambda_{md} = X_{ad} \left( \frac{\lambda_{ds}}{X_{ls}} + \frac{\lambda_{dr}}{X_{lr}} \right) \quad (\text{A. 50})$$

donde

$$X_{aq} = X_{ad} = \left( \frac{1}{X_M} + \frac{1}{X_{ls}} + \frac{1}{X_{lr}} \right)^{-1} \quad (\text{A. 51})$$

Sustituyendo las corrientes en las ecuaciones de voltaje y resolviendo para los enlaces de flujo como variables de estado el modelo que produce es:

$$\lambda_{qs} = \frac{\omega_b}{p} \left[ v_{qs} - \frac{\omega}{\omega_b} \lambda_{ds} + \frac{r_s}{X_{ls}} (\lambda_{mq} - \lambda_{qs}) \right] \quad (\text{A. 52})$$

$$\lambda_{ds} = \frac{\omega_b}{p} \left[ v_{ds} + \frac{\omega}{\omega_b} \lambda_{qs} + \frac{r_s}{X_{ls}} (\lambda_{mq} - \lambda_{ds}) \right] \quad (\text{A. 53})$$

$$\lambda_{0s} = \frac{\omega_b}{p} \left[ v_{0s} - \frac{r_s}{X_{ls}} \lambda_{0s} \right] \quad (\text{A. 54})$$

$$\lambda_{qr} = \frac{\omega_b}{p} \left[ v_{qr} - \left( \frac{\omega - \omega_r}{\omega_b} \right) \lambda_{dr} + \frac{r_r}{X_{lr}} (\lambda_{mq} - \lambda_{qr}) \right] \quad (\text{A. 55})$$

$$\lambda_{dr} = \frac{\omega_b}{p} \left[ v_{dr} + \left( \frac{\omega - \omega_r}{\omega_b} \right) \lambda_{qr} + \frac{r_r}{X_{lr}} (\lambda_{md} - \lambda_{dr}) \right] \quad (\text{A. 56})$$

$$\lambda_{0r} = \frac{\omega_b}{p} \left[ v_{0r} - \frac{r_r}{X_{lr}} \lambda_{0r} \right] \quad (\text{A. 57})$$

Para obtener el modelo en cualquiera de los tres marcos de referencia mencionados, basta con considerar los siguientes aspectos. Podemos definir al marco de referencia síncrono como el marco de referencia que rota a la misma velocidad angular correspondiente a la frecuencia fundamental de las variables asociadas con circuitos estacionarios, aquí denotada como  $\omega_b$ . En el caso de las máquinas de c.a.,  $\omega_b$  es la velocidad angular del campo magnético rotatorio establecido por las corrientes del estator. Como se menciono las ecuaciones de voltaje de la máquina en el marco de

referencia arbitrario pueden ser usadas para cualquier marco de referencia [17]. La transformación para un marco de referencia específico se obtiene mediante sustituir la velocidad  $\omega$ , del marco que se trate. En la tabla siguiente se muestra la apropiada velocidad angular para cada marco de referencia.

**Tabla A. 1.** Interpretación de  $\omega$  en los distintos marcos de referencia

Velocidad del marco de referencia	Interpretación
$\omega$ (no especificado)	VARIABLES de los circuitos estacionarios referidos al marco de referencia arbitrario
0	VARIABLES de los circuitos estacionarios referidos al marco de referencia estacionario
$\omega_r$	VARIABLES de los circuitos estacionarios referidos al marco de referencia del rotor
$\omega_b$	VARIABLES de los circuitos estacionarios referidos al marco de referencia síncrono

### A.3. Ecuación del Par Electromagnético.

En general, la producción del par electromagnético en el motor de inducción puede expresarse mediante la siguiente ecuación [62]

$$T_e = c \bar{\lambda}_s \times \bar{i}_r \quad (\text{A. 58})$$

donde bajo condiciones magnéticas lineales  $c$  es una constante,  $\lambda_s$  e  $i_r$  son los vectores del flujo del estator y corriente del rotor respectivamente expresados en el marco de referencia estacionario. En la ecuación anterior se ve que el par electromagnético desarrollado es el producto cruz de los vectores flujo y corriente, sin embargo (A.58) puede describirse usando sus magnitudes

$$T_e = c |\bar{\lambda}_s| |\bar{i}_r| \sin \gamma \quad (\text{A. 59})$$

donde  $\gamma$  es el ángulo del par. Es lógico que cuando  $\gamma = 90^\circ$  el motor produce el par máximo.

Una vez más usando la transformación (A.24) para desacoplar las variables de la máquina, se obtiene la siguiente ecuación para el par electromagnético en términos de las corrientes y enlaces de flujos en los ejes  $dq$  en el marco de referencia arbitrario [30]

$$T_e = \left(\frac{3}{2}\right) \left(\frac{1}{\omega_b}\right) (\lambda_{qr} i_{dr} - \lambda_{dr} i_{qr}) \quad (\text{A. 60})$$

$$T_e = \left(\frac{3}{2}\right) \left(\frac{1}{\omega_b}\right) (\lambda_{ds} i_{qs} - \lambda_{qs} i_{ds}) \quad (\text{A. 61})$$

De igual manera que las ecuaciones de voltaje, las expresiones anteriores también pueden escribirse en el sistema por unidad mediante ser divididas por el par electromagnético base

$$T_{eB} = \frac{P_B}{\omega_b} \quad (\text{A. 62})$$

donde,  $P_B = (3/2) V_B(dq) I_B(dq)$  entonces las ecuaciones del par electromagnético producido en por unidad son:

$$T_e = \lambda_{qr} i_{dr} - \lambda_{dr} i_{qr} \quad (\text{A. 63})$$

ó

$$T_e = \lambda_{ds} i_{qs} - \lambda_{qs} i_{ds} \quad (\text{A. 64})$$

La ecuación que relaciona el par electromagnético producido con la velocidad del rotor es

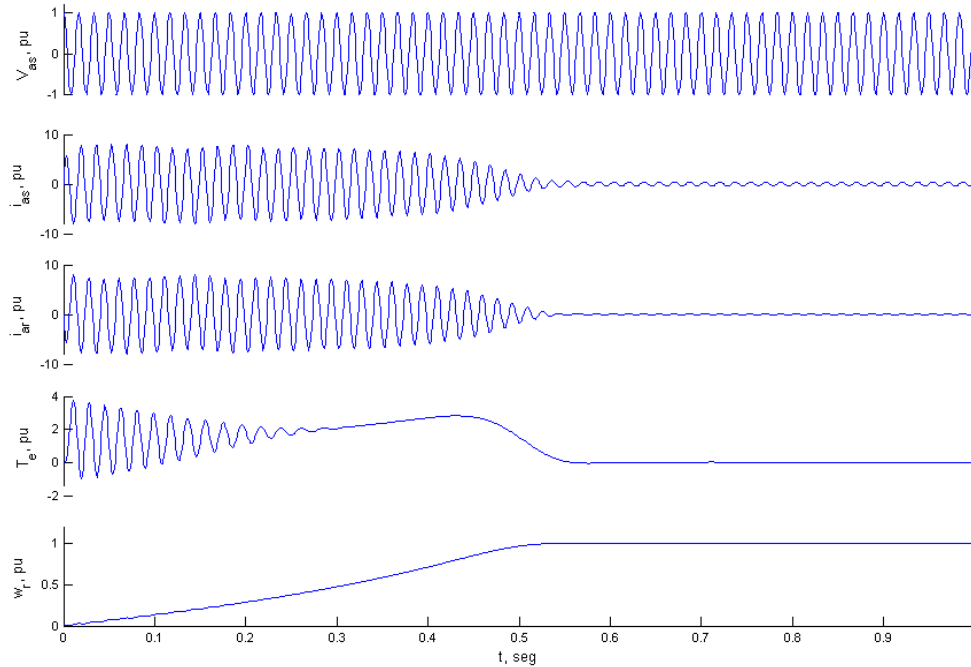
$$T_e = 2Hp \frac{\omega_r}{\omega_b} + T_m \quad (\text{A. 65})$$

donde  $H$  es la constante de inercia expresada en segundos,  $\omega_r$  y  $\omega_b$  son la velocidad angular del rotor y la velocidad angular base en radianes por segundo respectivamente,  $T_m$  es el par de la carga conectada a la flecha del motor y  $p$  es el operador  $d/dt$ .

Para comprobar el modelo matemático, las figuras siguientes presentan el comportamiento dinámico del motor operando en vacío, así como la trayectoria del vector del flujo del estator. Se simula el motor desde el arranque hasta que alcanza su velocidad nominal, siendo alimentado por una fuente de voltaje senoidal a 60Hz trifásica ideal. Usando el modelo matemático de la máquina de inducción en el marco

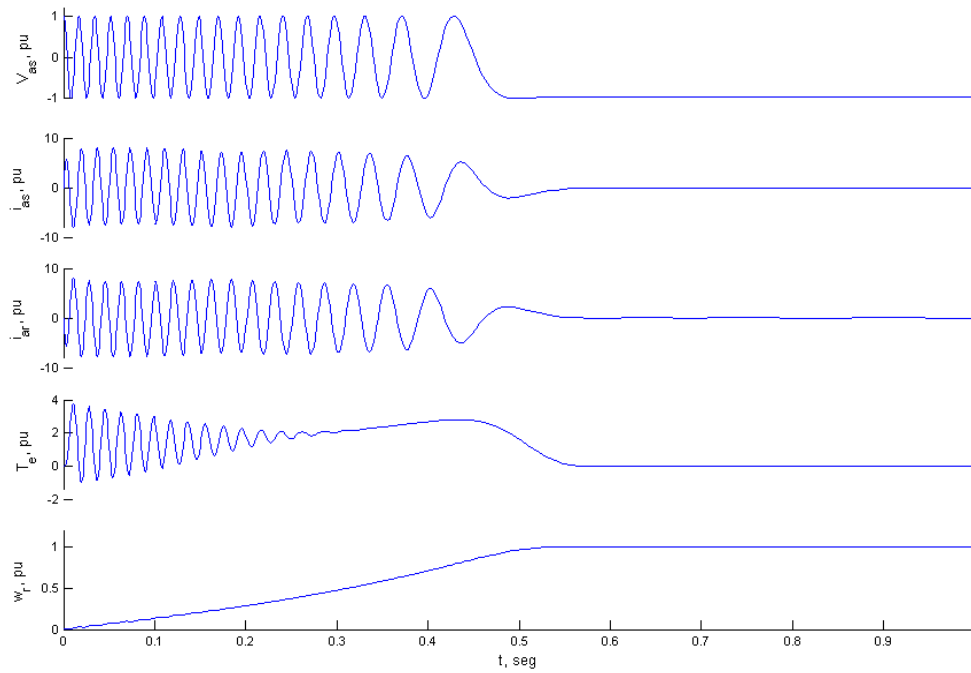
de referencia arbitrario, se realiza la simulación de un motor cuyos parámetros se muestran en el apéndice D.

Para estas simulaciones se toma el valor de 1.0pu para las variables sinusoidales. El rotor acelera desde una velocidad cero hasta la velocidad síncrona (377 rad/seg) ya que no hay par mecánico y como las pérdidas por fricción se han despreciado, el rotor se acelera libremente. El comportamiento del motor desarrollado en el marco de referencia estacionario durante libre aceleración se muestra en la figura A.3. Bajo este marco referido al estator, las variables en  $q_s$  y  $d_s$  se relacionan aritméticamente con las variables reales en  $abc$ . En particular la posición cero del marco de referencia estacionario es en eje  $a_s$ , por lo tanto  $f_{as} = f_{qs}$ . Así,  $V_{qs}$  e  $i_{qs}$  son idénticos a  $V_{as}$  e  $i_{as}$  respectivamente. Las variables del rotor están referidas al estator por esa razón varían a una frecuencia de 60 Hz.

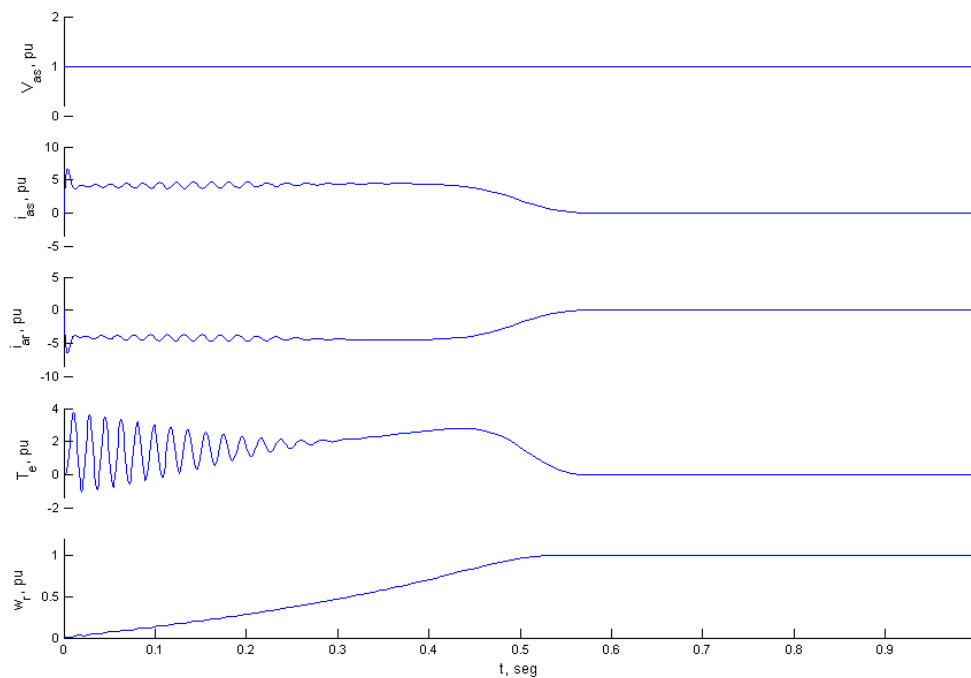


**Figura A. 3.** Libre aceleración del motor de inducción en el marco de referencia estacionario.

Las características de libre aceleración del motor con el marco de referencia en el rotor se muestran en la figura A.4. Aquí, la posición cero del rotor y del marco de referencia son ambos cero, por consiguiente  $f_{ar}=f_{qr}$ . Así,  $i_{qr}$  es idéntica a  $i_{ar}$  y como las variables del estator están referidas al rotor, estas varían a la frecuencia de deslizamiento.



**Figura A. 4.** Libre aceleración del motor de inducción en el marco de referencia rotatorio.



**Figura A. 5.** Libre aceleración del motor de inducción en el marco de referencia síncrono.

La libre aceleración del motor en el marco de referencia síncrono con la velocidad angular de los voltajes aplicados al motor se muestra en la figura A.5. En este modo, la posición cero del marco de referencia es seleccionado de tal manera que  $v_{qs}$  tenga el valor de la amplitud del voltaje de fase aplicado al estator y  $v_{ds}$  sea igual a cero.

Comparando los resultados de los tres marcos de referencia es interesante notar que las características del par electromagnético son las mismas independientemente de la velocidad o el cambio en la velocidad del marco de referencia.

# Apéndice B

## Control Directo del Par

### B.1. Introducción General.

Los accionamientos eléctricos de control del par y flujo magnético tanto en motores de c.d. como en motores de inducción se han utilizado por más de 20 años. Basados en los trabajos de Blaschke, Hasse y Leonhard, el control vectorial ha llegado a ser muy popular en este tipo de aplicaciones al grado de utilizarse como estándares tecnológicos en el ramo [62]. Los primeros accionamientos basados en el Control Directo del Par para motores de inducción fueron desarrollados hace más de 10 años. Sin embargo, en la actualidad la compañía ABB ha sido la primera en comercializar esta tecnología. La figura B.1 muestra el esquema general del Control Directo de Par (DTC por sus siglas en inglés *Direct Torque Control*), creada en la década de los 80's por Depenbrock [11] y Takahashi [56] y que actualmente está siendo desarrollada industrialmente por la compañía ABB.

En un accionamiento de motores de inducción basado en el DTC y alimentado por un inversor fuente de voltaje (VSI), es posible controlar directamente los enlaces de flujo del estator y el par electromagnético producido mediante una selección óptima de los modos o estados de conmutación del inversor. La selección se realiza para restringir los errores del flujo y par dentro de sus respectivas bandas de histéresis y así obtener respuestas instantáneas del par, baja frecuencia de conmutación y bajas pérdidas por armónicas.

### B.2. Esquema del Control Directo del Par.

El DTC es una estrategia de control para accionamientos de motores de inducción donde a través de las conmutaciones del inversor se puede controlar directamente las variables del motor tales como el par electromagnético y el flujo del estator. La figura B.1 presenta el esquema de DTC. Con los niveles de corriente y voltaje del motor se estiman valores actuales del par y flujo magnéticos producidos a través de un modelo matemático de la máquina (*Adaptive motor model*). Estos valores son comparados en los



controles de histéresis del par y del flujo (*Torque comparator and Flux comparator*) con los valores de referencia producidos por los controles de referencia (*Torque and Flux reference controllers*). Dependiendo de las salidas de los comparadores, un selector de pulsos (*optimum pulse selector*) determina la posición óptima de los IGBT's del inversor que a su vez determina el voltaje aplicado en las terminales del motor. El voltaje apropiado controla la producción del par y flujo cerrándose así el lazo de control.

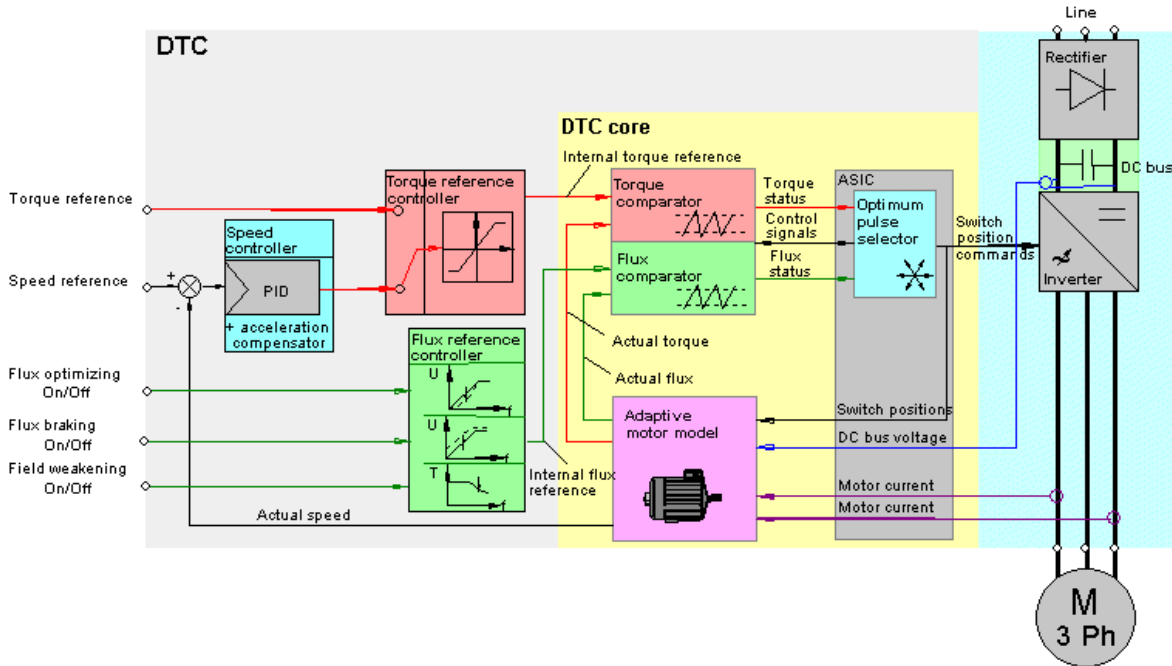


Figura B. 1. Estructura básica del DTC (Figura tomada de [www.abb.com](http://www.abb.com))

### B.3. Principios Teóricos del Control Directo del Par.

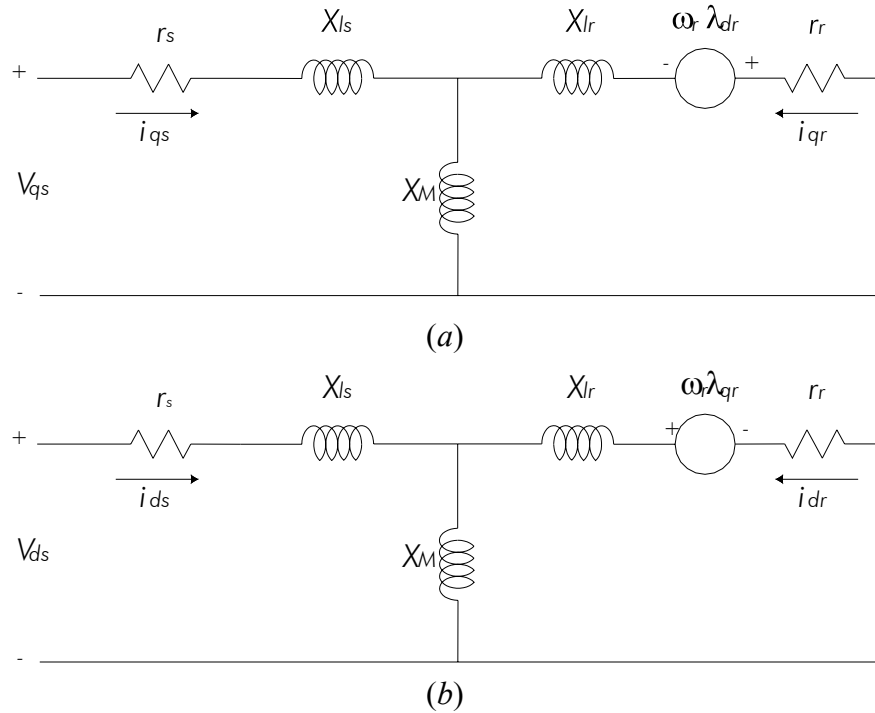
El par electromagnético producido por el motor se puede expresar en términos de los vectores espaciales del flujo del estator y la corriente del rotor mediante la ecuación [60]:

$$T_e = c |\bar{\lambda}_s| |\bar{i}_r| \sin \gamma \quad (\text{B. 1})$$

donde  $\gamma$  es el ángulo del par. Mediante la ecuación (B.1) se puede observar que el par electromagnético puede controlarse mediante ajustar dicho ángulo. Lo mismo ocurre si se emplea la siguiente expresión para el par electromagnético [62]:

$$T_e = c |\bar{\lambda}_s| |\bar{\lambda}_r| \sin \theta_{sr} \quad (\text{B. 2})$$

donde  $\theta_{sr}$  también representa el ángulo del par. Por otro lado la magnitud del vector  $\lambda_s$ , flujo del estator, la cual es una medida de la intensidad del campo magnético en el motor, es directamente dependiente del voltaje aplicado al estator. Esto se ve claramente utilizando el diagrama equivalente del motor de inducción que se muestra en la figura siguiente [30]:

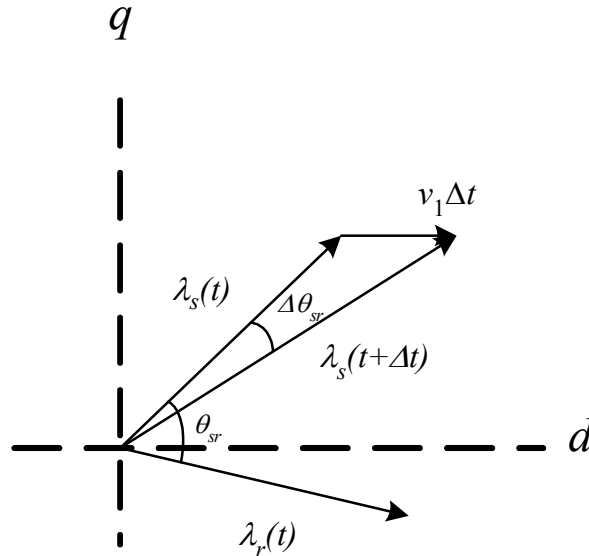


**Figura B. 2.** Diagrama equivalente del motor de inducción en el marco de referencia estacionario. (a) Eje q; (b) Eje d.

Del diagrama equivalente, se observa que el vector del flujo del estator reacciona instantáneamente al cambio del vector del voltaje aplicado al estator siendo separados solo por su resistencia. Sin embargo, se espera que la reacción del vector del flujo del rotor sea mucho menor, incluso casi despreciable para intervalos de tiempo pequeños, gracias a la acción de filtrado que proporcionan las reactancias de dispersión,  $X_{ls}$  y  $X_{lr}$ , del estator y del rotor respectivamente. Por lo anterior, si se considera que el vector del flujo del rotor permanece constante, es posible ajustar el ángulo de par,  $\theta_{sr}$ , mediante variar las componentes del vector del flujo del estator aplicando un vector de voltaje adecuado en las terminales del motor.

El impacto del voltaje del estator sobre el flujo del estator se ilustra en la figura B.3. En un instante  $t$ , el inversor que alimenta al motor cambia al estado 4 (ver apéndice C) representado por el vector espacial de voltaje  $v_1$ . Los vectores iniciales del flujo de estator y del rotor son denotados por  $\lambda_s(t)$  y  $\lambda_r(t)$  respectivamente. Después de un pequeño intervalo de tiempo  $\Delta t$ , el nuevo vector de flujo del estator,  $\lambda_s(t+\Delta t)$ , difiere

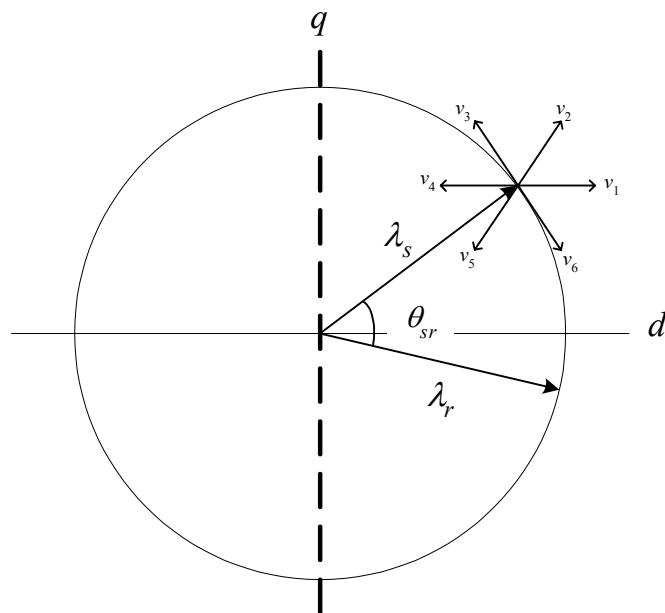
de  $\lambda_s(t)$  tanto en magnitud como en posición. Si este intervalo de tiempo es suficientemente pequeño los cambios en el vector de flujo del rotor pueden ser despreciados. El flujo del estator se ha incrementado y el ángulo del par,  $\theta_{sr}$  se ha reducido por  $\Delta\theta_{sr}$ . Claro está, si otro vector de voltaje fuera aplicado, los cambios en el vector de flujo del estator sería diferente.



**Figura B. 3.** Impacto del vector de voltaje sobre el flujo del estator.

Las distintas direcciones en las que el vector de flujo puede cambiar asociadas a cada vector de voltaje activo,  $v_1$  a  $v_6$  del VSI se muestra en la figura B.4 misma que ilustra la trayectoria circular a la que el vector de flujo debe ser ajustado. De esta forma, la selección apropiada de los estados del inverso permite ajustes tanto de la magnitud del flujo magnético como del par desarrollado. En otras palabras, esta selección proporciona el vector de voltaje aplicado al motor de manera que el flujo del estator se mantenga constante (o a un valor deseado), pero al mismo tiempo cambie su posición rápidamente de acuerdo con el valor de par deseado. Recuérdese que el cambio en el ángulo  $\theta_{sr}$  es casi proporcional al cambio del par electromagnético producido.

En resumen el par electromagnético puede cambiarse rápidamente mediante controlar el vector de flujo del estator mismo que se ajusta mediante usar un vector espacial de voltaje apropiado generado por el inversor que alimenta al motor. Este principio es el que le da el nombre a esta estrategia ya que como se ve, el flujo y par son controlados directamente por el voltaje aplicado al motor.



**Figura B. 4.** Principio del DTC mediante la selección de los estados del VSI.

Por otra parte, actualmente existen distintas técnicas de implementar el DTC y la selección de los estados del VSI, entre ellas predominan la Tabla de Conmutación, el Autocontrol Directo y la Modulación de Vectores Espaciales [6][11][20][33].

## Apéndice C

# Inversor de Potencia y Modulación de Vectores Espaciales

### C.1. Inversor Fuente de Voltaje (VSI)

La figura C.1 muestra un inversor tipo fuente de voltaje VSI el cual es utilizado en este trabajo.

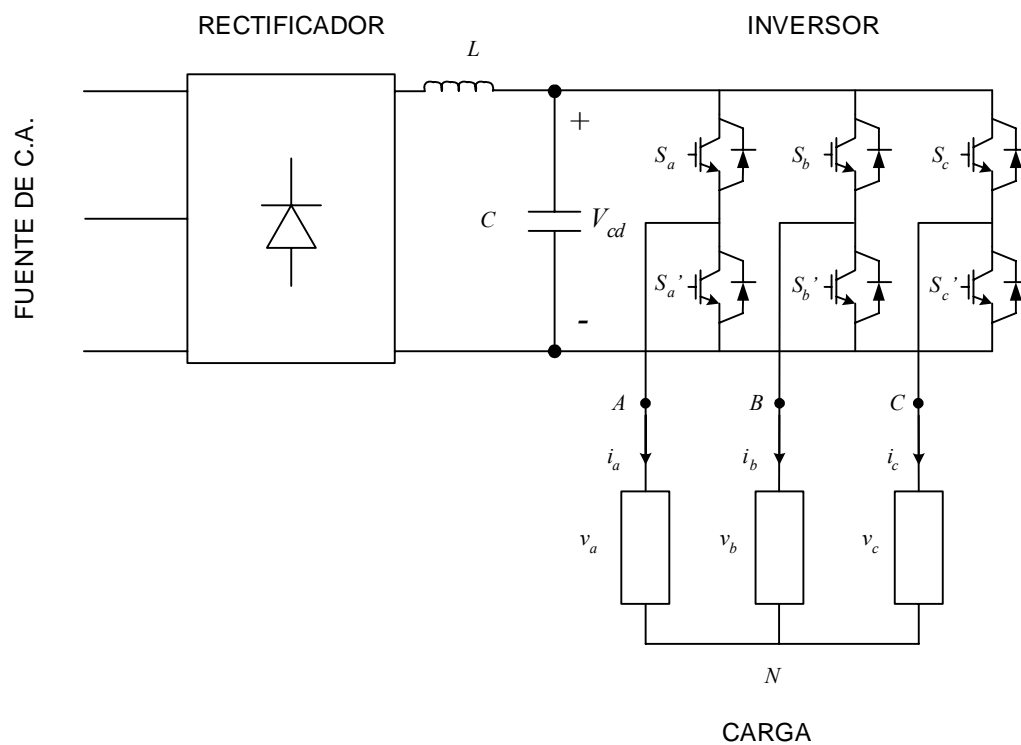
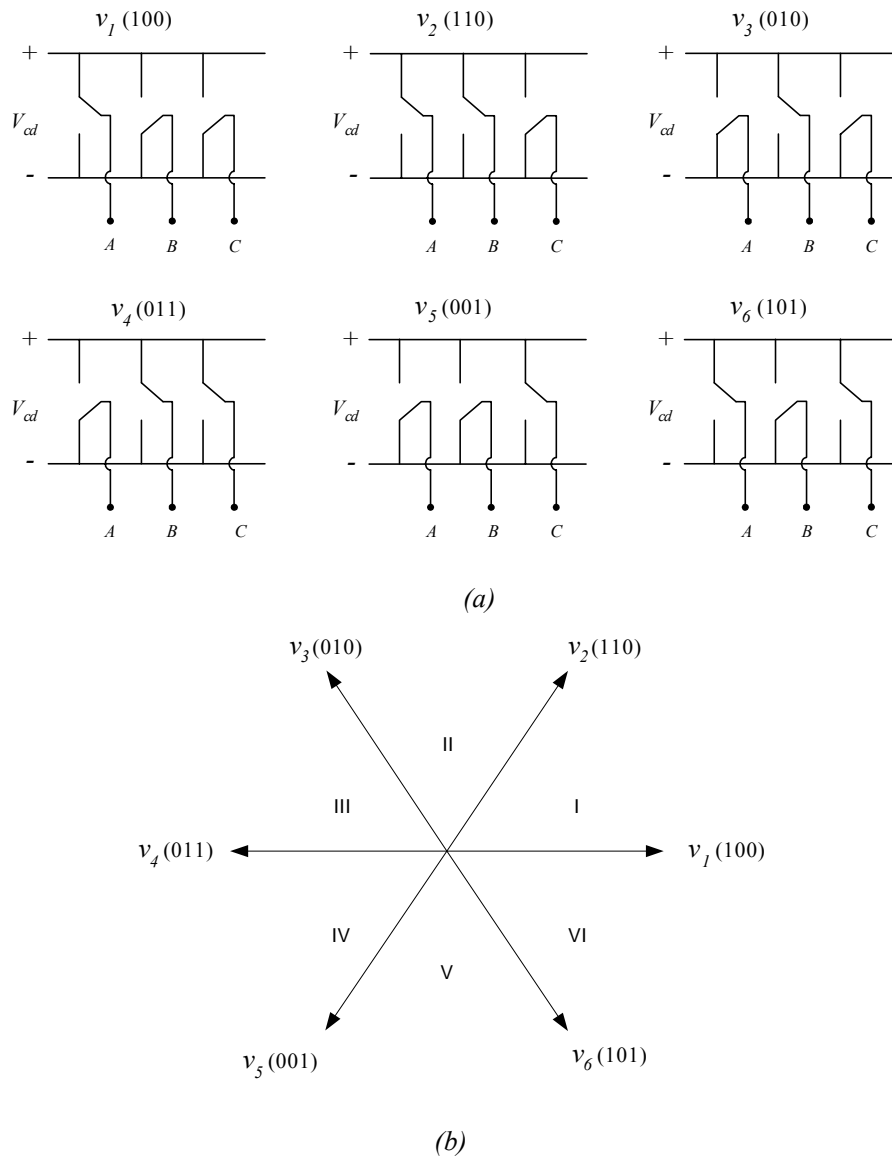


Figura C. 1. Inversor Trifásico Fuente de Voltaje (VSI).

La estructura del VSI mostrada en la figura C.1 presenta el uso de seis IGBT's que controlan la señal de salida, en este caso los voltajes  $v_a$ ,  $v_b$  y  $v_c$ , mediante las conmutaciones de  $S_a$ ,  $S_a'$ ,  $S_b$ ,  $S_b'$ ,  $S_c$  y  $S_c'$ . Es obvio que en cada fase los IGBT's no pueden estar en un mismo estado al mismo tiempo. Utilizando una notación binaria ( $1 = on$ ;  $0 = off$ ) se considera que la fase  $A$  está en posición *on* cuando  $S_a = 1$  y  $S_a' = 0$ . La figura C.2 muestra los ocho diferentes estados del inversor (6 activos y 2 cero) y su correspondiente representación vectorial [62].



**Figura C. 2.** Esquema de los seis estados activos del PWM VSI y su representación con vectores espaciales (a) Estados del VSI; (b) Representación con vectores espaciales.

Los estados  $v_7$  (111) y  $v_8$  (000) corresponden a cuando el inversor está en corto circuito y circuito abierto respectivamente. La selección del vector espacial de la figura C.2 que será aplicado al motor se puede expresar por el siguiente algoritmo [20][62]:

$$v_{s(k)} = \begin{cases} \frac{2}{3} V_{cd} \exp[j(k-1)\pi/3] & k = 1, 2, \dots, 6 \\ 0 & k = 7, 8 \end{cases} \quad (\text{C. 1})$$

donde  $V_{cd}$  es el voltaje de corriente directa que alimenta al inversor y  $k$  es uno de los seis sectores en que se divide el plano  $dq$  de vectores espaciales de voltaje.

## C.2. Modulación de Vectores Espaciales (SVM).

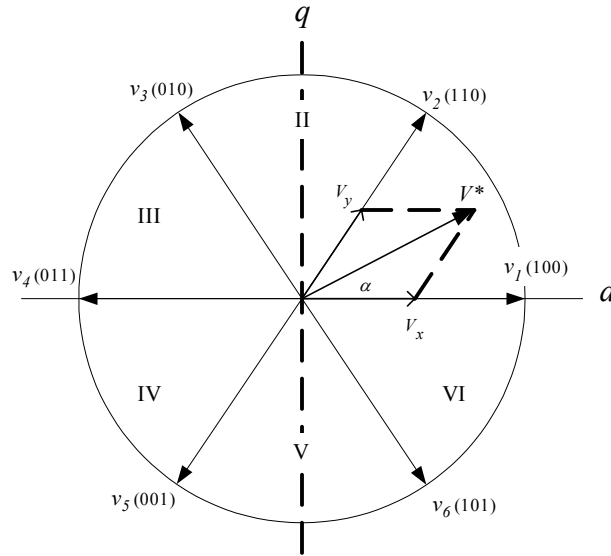
La técnica SVM (por sus siglas en inglés *Space Vector Modulation*) se refiere a una especial manera de determinar la secuencia de conmutación de los IGBT's del VSI y seleccionar así el vector de voltaje apropiado [49]. Esta técnica se ha hecho muy popular recientemente como un método PWM porque logra menor distorsión en las señales de voltaje y corriente provocada por armónicas. Asimismo su incorporación mediante microprocesadores esta bien documentada. Un ejemplo de esto son [1][20][49][49][66].

El objetivo de esta técnica es de aproximar una referencia de voltaje deseado  $V^*$  instantáneamente mediante la combinación de los estados de conmutación que corresponda a los vectores espaciales. La idea de SVM se puede entender con la figura C.3 donde se asume que para cierto instante el vector de referencia se encuentra en el sector I.

El vector de referencia  $V^*$  puede obtenerse mediante la combinación de los vectores  $V_x$  y  $V_y$  que son fracciones de los vectores  $v_1$  y  $v_2$  respectivamente. Esto quiere decir que si durante un tiempo  $t_a$  se aplica el vector  $v_1$  e inmediatamente después, durante un tiempo  $t_b$  se aplica  $v_2$ , lo que se obtiene es una aproximación o promedio del vector  $V^*$  deseado. Finalmente para cerrar el periodo de conmutación  $T_s$  del inversor se aplica un vector cero ( $v_7$  o  $v_8$ ) durante un tiempo  $t_0$ . Matemáticamente el SVM se puede expresar de la siguiente forma [20], [49]:

$$V_s^* T_s = V_{s(k)} t_a + V_{s(k+1)} t_b + V_{7,8} t_0 \quad (\text{C. 2})$$

donde  $V_{s(k)}$  y  $V_{s(k+1)}$  son los vectores adyacentes a  $V^*$  y se obtienen mediante la ecuación (C.1). Finalmente, de acuerdo con [49], los tiempos efectivos de los estados de conmutación del inversor están dados por:



**Figura C. 3.** Principio de la técnica del SVM.

$$t_a = 2 \cdot K \cdot V^* \sin(\pi/3 - \alpha) \quad (\text{C. 3})$$

$$t_b = 2 \cdot K \cdot V^* \sin \alpha \quad (\text{C. 4})$$

$$t_0 = \frac{T_s}{2} - (t_a + t_b) \quad (\text{C. 5})$$

donde:

$t_a$  = tiempo de aplicación del vector  $V_x$

$t_b$  = tiempo de aplicación del vector  $V_y$

$t_0$  = tiempo de aplicación del vector cero

$T_s = 1/f_s$ , periodo de conmutación PWM ( $f_s$  = frecuencia de conmutación)

$\alpha$  = ángulo de  $V^*$  dentro del sector  $k$

$$K = (\sqrt{3} \cdot T_s) / (4 \cdot V_{cd})$$

el ángulo  $\alpha$  se obtiene por

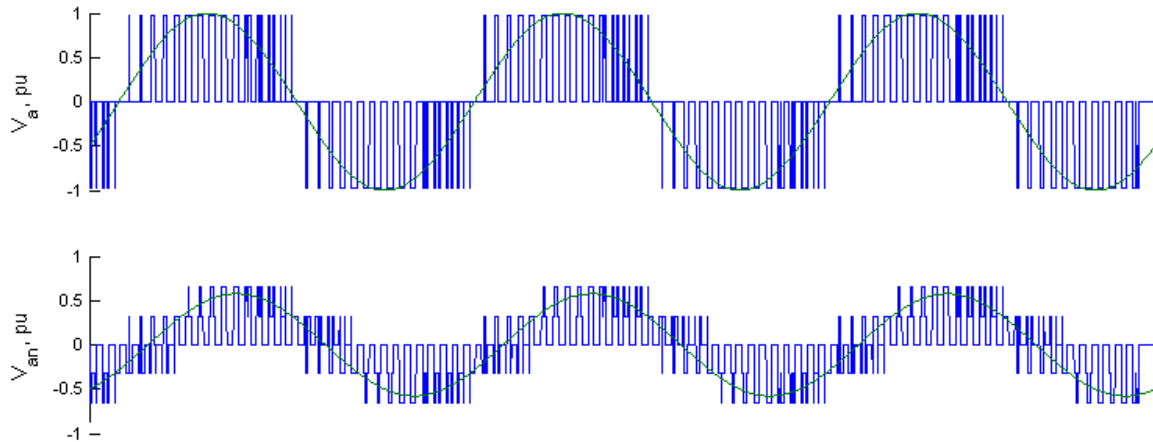
$$\alpha = \theta - (k-1) \frac{\pi}{3} \quad (\text{C. 6})$$

donde  $\theta$  es la posición del vector  $V^*$  en el plano  $dq$ . Los tiempos son distribuidos para generar pulsos simétricos PWM.

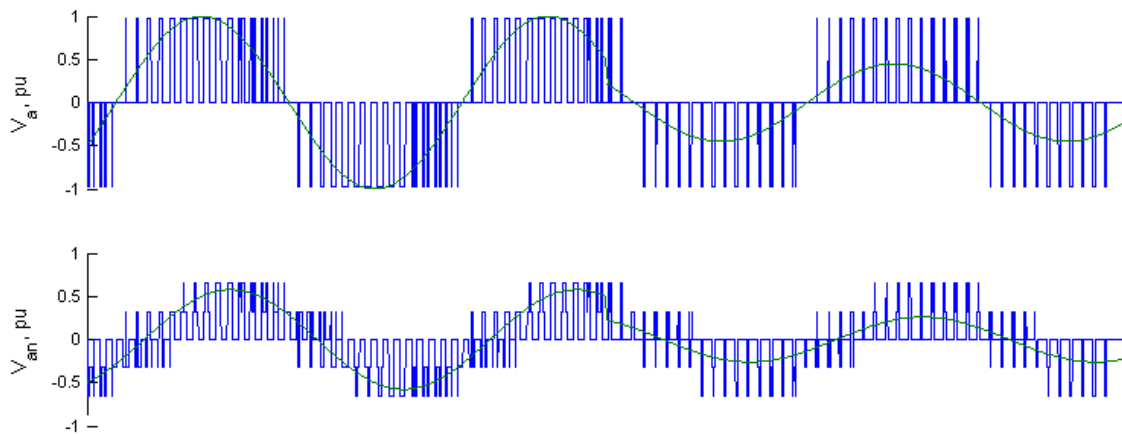
La figura C.4 muestra la aplicación de esta técnica en el VSI considerado. Se considera una frecuencia de conmutación de 1.8 kHz. En la figura se puede apreciar



los pulsos generados por la apertura y cierre de los IGBT's del inversor. Dichos pulsos son controlados a través de técnica SVM. La función senoidal representa la señal de referencia proporcionado por el vector de voltaje de referencia.



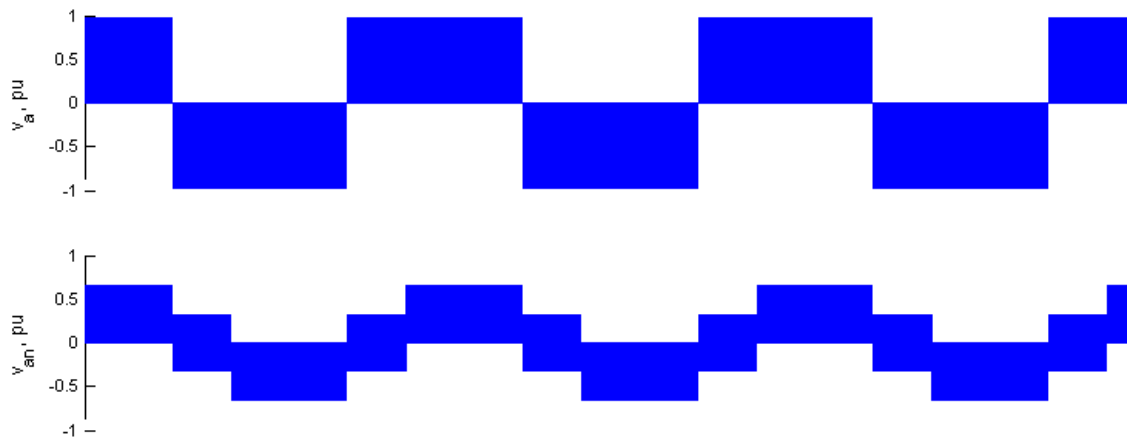
**Figura C. 4.** Señales de voltaje proporcionadas por un VSI con PWM-SVM.  $v_a$ , Voltaje de línea;  $v_{an}$ , Voltaje de fase.



**Figura C. 5.** Modulación del vector de referencia ante un cambio en su magnitud.  $v_a$ , Voltaje de línea;  $v_{an}$ , Voltaje de fase.

La figura C.5 muestra como el vector de voltaje de referencia  $V^*$ , es modulado por el SVM ante un cambio repentino en su magnitud. Como se observa la variación de magnitud en el voltaje de salida así como también la frecuencia puede variarse de manera sencilla mediante esta técnica PWM. Una de sus mayores ventajas es que no usa tablas de conmutación y que la frecuencia de conmutación es constante.

Finalmente la figura C.6 presenta las señales de voltaje proporcionadas por el inversor a una frecuencia de conmutación de 20kHz.



**Figura C. 6.** Señales de voltaje a una frecuencia de conmutación de 20kHz.  $v_a$ , Voltaje de línea;  $v_{an}$ , Voltaje de fase.

## Apéndice D

### Parámetros del Motor de Inducción

Parámetros del motor usado en la sección 4.5.

Potencia nominal	10	HP
Tensión nominal	220	Volts
Corriente nominal	28	Amperes
Frecuencia	60	Hz
No. de polos	6	polos
Resistencia del estator ( $r_s$ )	0.0453	p.u.
Resistencia del rotor ( $r_r$ )*	0.0222	p.u.
Reactancia del estator ( $X_{ls}$ )	0.0775	p.u.
Reactancia del rotor ( $X_{lr}$ )*	0.0322	p.u.
Reactancia mutua ( $X_m$ )	2.042	p.u.
Inercia del rotor ( $H$ )	0.5	seg.
Velocidad base ( $\omega_b$ )	377	rad/seg

\* Los parámetros del rotor están referidos al estator.

Parámetros del motor usados en la comparación del DTC clásico [54] y el DTNFC.

Potencia nominal	500	W
Tensión nominal	127	Volts (fase)
Corriente nominal	2.9	Amperes
Frecuencia	50	Hz
No. de polos	4	polos
Resistencia del estator	4.491	$\Omega$
Resistencia del rotor	5.365	$\Omega$ .
Inductancia de magnetización	149	mH.
Inductancia de dispersión del estator	16	mH.
Inductancia del estator	165	mH.
Inductancia de dispersión del rotor	13	mH.
Inductancia del rotor	162	mH.
Inercia del rotor	$0.95e^{-3}$	$\text{Kg}\cdot\text{m}^2$ .
Velocidad nominal	1400	rpm

Parámetros del motor usados en la comparación del DTC-SVM [20] y el DTNFC.

Potencia nominal	15	HP
Tensión nominal	380	Volts
Corriente nominal	21.6	Amperes
Frecuencia	50	Hz
No. de polos	4	polos
Resistencia del estator	0.371	$\Omega$
Resistencia del rotor	0.415	$\Omega$ .
Inductancia mutua	84.33	mH.
Inductancia de dispersión del estator	2.72	mH.
Inductancia de dispersión del rotor	3.3	mH.
Inercia del rotor	0.445	$\text{Kg}\cdot\text{m}^2$ .
Velocidad nominal	1430	rpm

## Apéndice E

### Programas de Simulación Digital

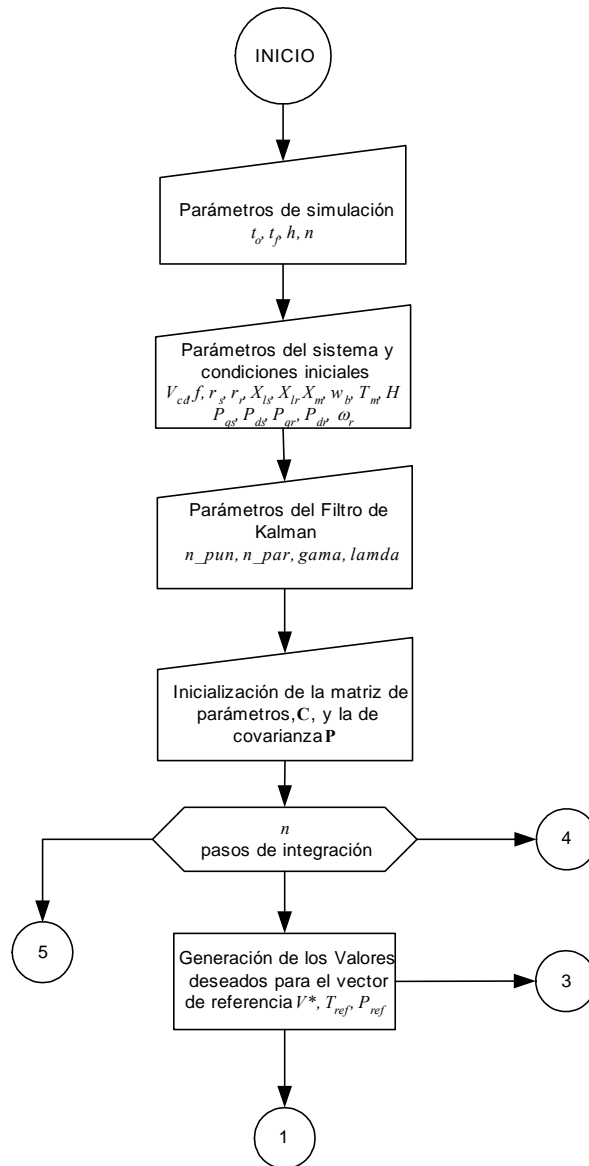
Este apéndice presenta el código en lenguaje C++ de todos los programas de simulación utilizados en este trabajo así como también los diagramas de flujo correspondientes de los programas principales tales como el DTNFC en su fase de entrenamiento y el DTNFC en su fase de aplicación.

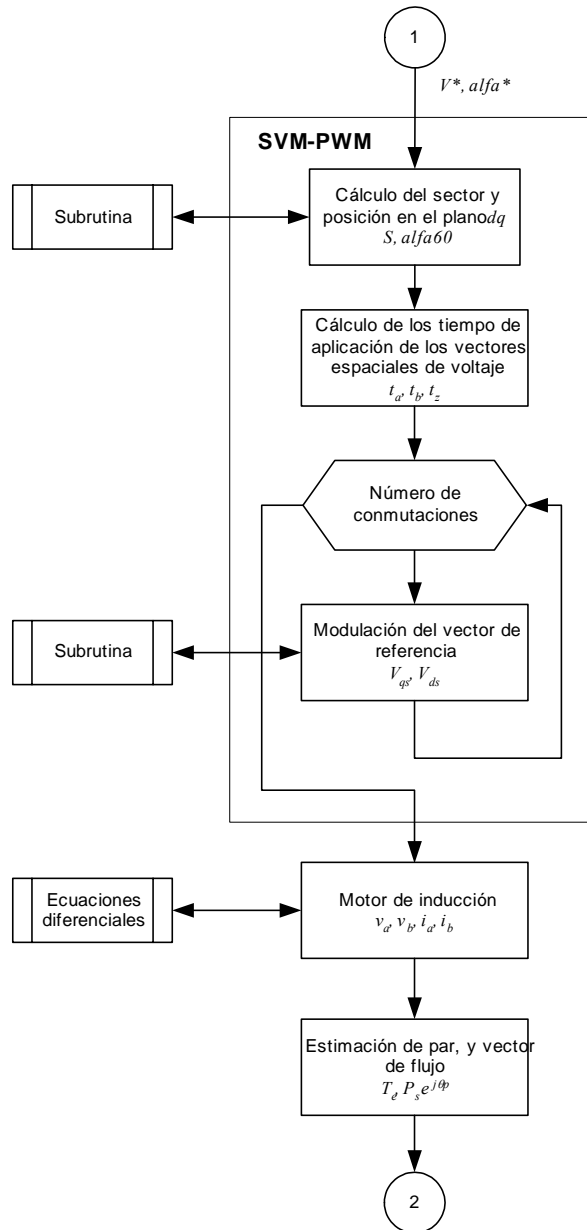
Es importante señalar que los tres primeros programas son independientes al programa principal del sistema y se han incluido como parte de las aportaciones de este trabajo para ser utilizados de forma independiente para futuras investigaciones o en actividades docentes.

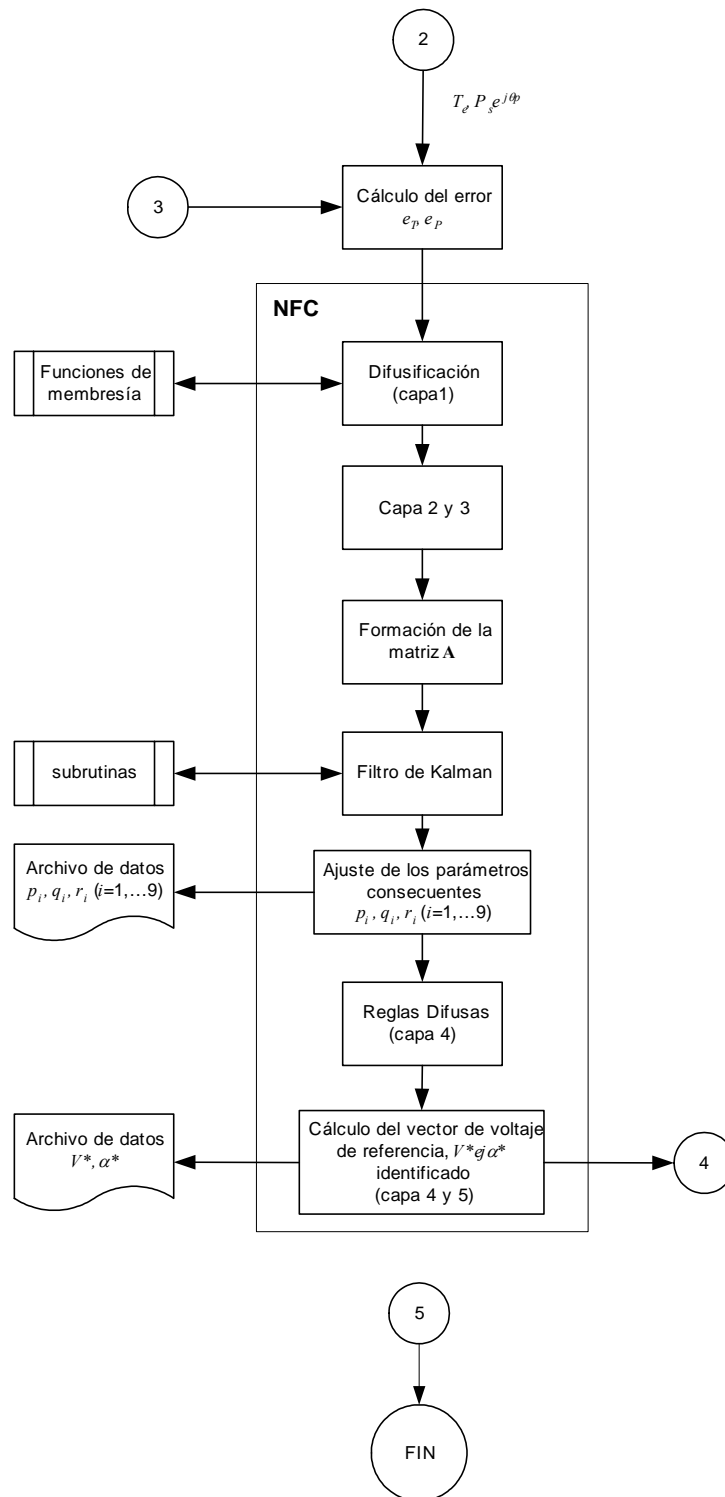
Los programas incluidos son:

- ✓ Motor de Inducción.
- ✓ Modulación de ancho de pulso de Vectores Espaciales para Inversores Fuente de Voltaje.
- ✓ Red Neuro-Difusa adaptable con Filtro de Kalman como algoritmo de aprendizaje.
- ✓ Control Neuro-Difuso Directo del Par y Flujo en su fase de entrenamiento.
- ✓ Control Neuro-Difuso Directo del Par y Flujo en su fase de aplicación.

Todos los programas de simulación producen un archivo como salida para almacenar los valores de las variables involucradas.

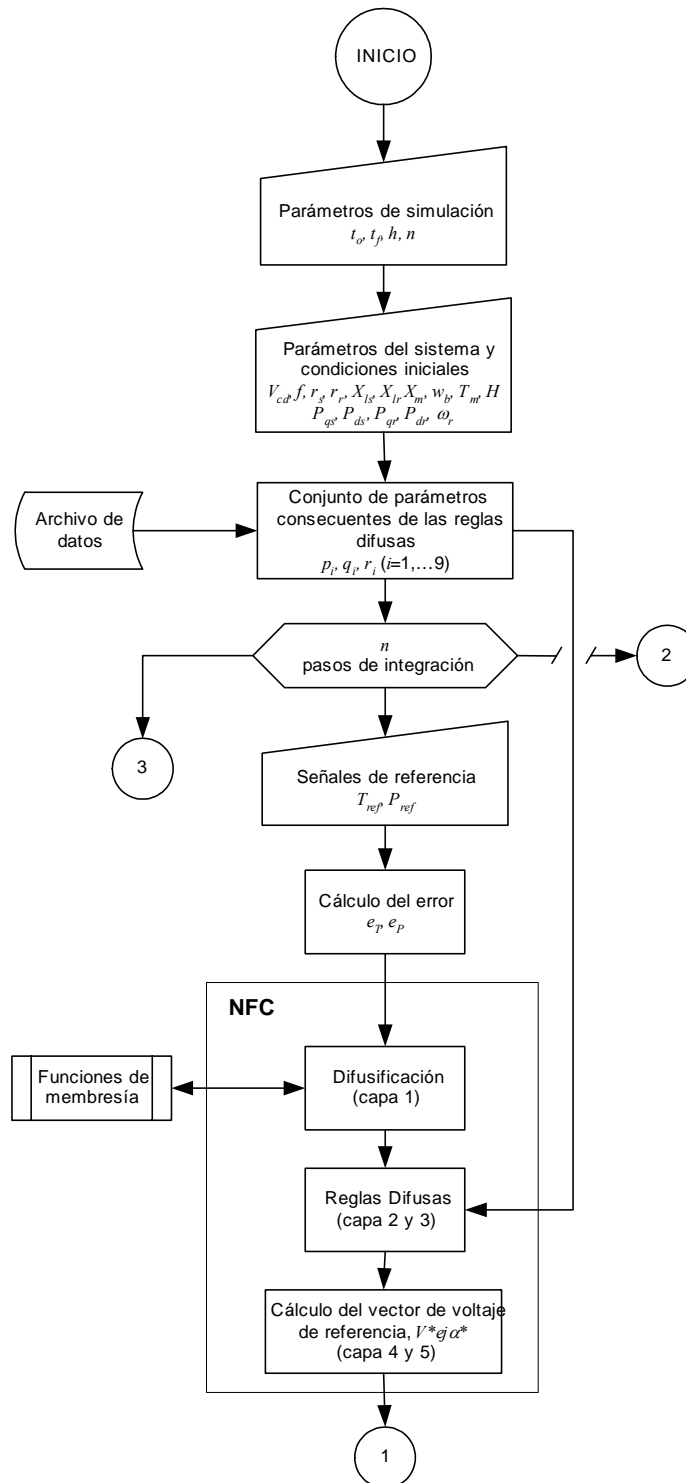
**E.1. Diagrama de Flujo para la Simulación del DTNFC en su Fase de Entrenamiento.**

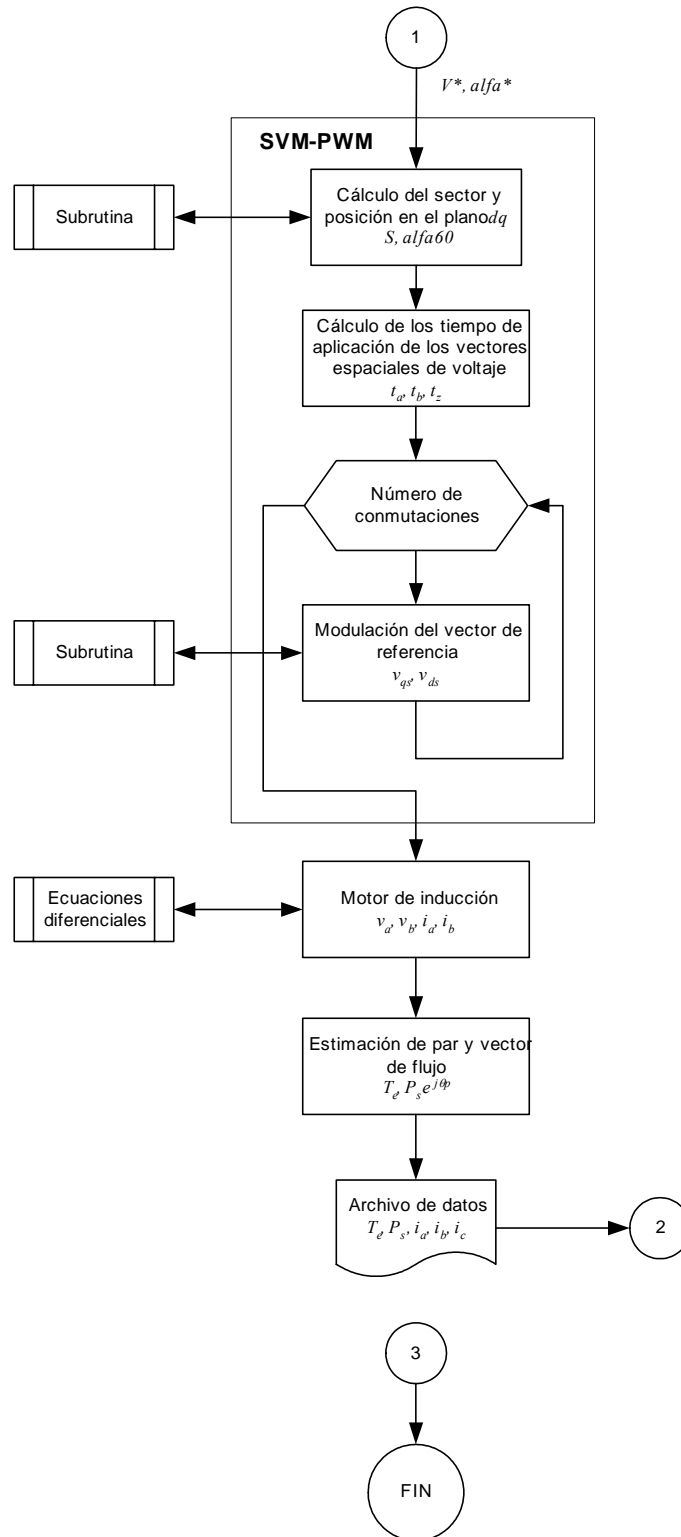






## E.2. Diagrama de Flujo para la Simulación del DTNFC en su Fase de Aplicación.





### E.3. Motor de Inducción.

**//MOTOR DE INDUCCION TIPO JAULA DE ARDILLA TRIFASICO  
//SIMULADO EN EL MARCO DE REFERENCIA ESTACIONARIO  
//PARAMETROS EN P.U.**

```
#include <string.h>
#include <iostream.h>
#include <fstream.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>

double funPqs(double,double,double,double,double,double);
double funPds(double,double,double,double,double,double);
double funPqr(double,double,double,double,double,double,double);
double funPdr(double,double,double,double,double,double,double);
double funwr(double,double,double);

void main()
{
    double to,tf,i,n;
    double Pqs,Pds,Pqr,Pdr,Pmq,Pmd,Vas,Vbs,Vcs,Vqs,Vds,iqs,ids,iqr,idr,Te,Tm;
    double h,hh,h2,t;
    double rs,rr,Xlr,Xls,Xm,Xaq,Xad,Vm,f,wf,pi,wb,wr,H,ias,ibs,ics;
    double k_Pqs,k_Pds,k_Pqr,k_Pdr,k_wr,m_Pqs,m_Pds,m_Pqr,m_Pdr,m_wr;
    FILE *fi;

    fi=fopen("min.dat","w"); //DATAFILE

    to=0.0; //TIEMPO DE INICIO
    tf=2.0; //TIEMPO FINAL
    h=0.0001; //PASO DE INTEGRACION
    hh=tf-to; //TIEMPO DE SIMULACIÓN
    n=hh/h; //NUMERO DE PASOS DE INTEGRACION
    h2=h/2;

    rs=0.0453; //RESISTENCIA DEL ESTATOR
    rr=0.0222; //RESISTENCIA DEL ROTOR
    Xls=0.0775; //REACTANCIA DEL ESTATOR
    Xlr=0.0322; //REACTANCIA DEL ROTOR
    Xm=2.042; //REACTANCIA MUTUA
    Vm=1.0; //VOLTAJE NOMINAL
    f=60; //FRECUENCIA NOMINAL
    pi=3.1416;
    wf=2*pi*f; //VELOCIDAD ANGULAR
    wb=377; //VELOCIDAD ANGULAR BASE
    Tm=0.0; //PAR MECANICO
    H=0.5; //INERCIA DEL ROTOR
    Xaq=1/((1/Xm)+(1/Xls)+(1/Xlr));
    Xad=Xaq;
```

```

/*****CONDICIONES INICIALES DEL SISTEMA *****/
Pqs=0.0; //FLUJO DEL ESTATOR EJE q
Pds=0.0; //FLUJO DEL ESTATOR EJE d
Pqr=0.0; //FLUJO DEL ROTOR EJE q
Pdr=0.0; //FLUJO DEL ROTOR EJE d
wr=0.0; //VELOCIDAD DEL ROTOR
t=to;

/*****CALCULO DE VOLTAJES abc*****/
printf("RUNNING SIMULATION\n");
for (i=0;i<=n;i++)
{
t=t+h;
Vas=Vm*cos(wf*t);
Vbs=Vm*cos(wf*t-(2*pi/3));
Vcs=Vm*cos(wf*t+(2*pi/3));

/*****TRANSFORMACION DE VOLTAJES abc-dq,CALCULO DE CORRIENTES dq Y PAR*****/
Vqs=(2*Vas/3)-(Vbs/3)-(Vcs/3);
Vds=(Vcs-Vbs)/sqrt(3);
Pmq=Xaq*((Pqs/Xls)+(Pqr/Xlr));
Pmd=Xad*((Pds/Xls)+(Pdr/Xlr));
iqs=(1/Xls)*(Pqs-Pmq);
ids=(1/Xls)*(Pds-Pmd);
iqr=(1/Xlr)*(Pqr-Pmq);
idr=(1/Xlr)*(Pdr-Pmd);
Te=(Pds*iqs)-(Pqs*ids); //PAR ELECTROMAGNETICO

/*****METODO DE INTEGRACION (RK)*****/
k_Pqs=h*funPqs(Pqs,Pmq,Vqs,wb,rs,Xls);
k_Pds=h*funPds(Pds,Pmd,Vds,wb,rs,Xls);
k_Pqr=h*funPqr(Pqr,Pdr,wr,Pmq,wb,rr,Xlr);
k_Pdr=h*funPdr(Pqr,Pdr,wr,Pmd,wb,rr,Xlr);
k_wr=h*funwr(Te,Tm,H);
m_Pqs=h*funPqs(Pqs+h2*k_Pqs,Pmq,Vqs,wb,rs,Xls);
m_Pds=h*funPds(Pds+h2*k_Pds,Pmd,Vds,wb,rs,Xls);
m_Pqr=h*funPqr(Pqr+h2*k_Pqr,Pdr+h2*k_Pdr,wr,Pmq,wb,rr,Xlr);
m_Pdr=h*funPdr(Pqr+h2*k_Pqr,Pdr+h2*k_Pdr,wr,Pmd,wb,rr,Xlr);
m_wr=h*funwr(Te,Tm,H);

/*****CALCULO DE VARIABLES DE ESTADO (FLUJOS dq Y dqr)*****/
Pqs=Pqs+m_Pqs;
Pds=Pds+m_Pds;
Pqr=Pqr+m_Pqr;
Pdr=Pdr+m_Pdr;
wr=wr+m_wr;

/*****TRANSFORMACION DE CORRIENTES dq-abc*****/
ias=iqs;
ibs=-(iqs+sqrt(3)*ids)/2;
ics=-(iqs-sqrt(3)*ids)/2;

//IMPRESION DE RESULTADOS
fprintf(fi,"%f %f %f %f %f %f %f %f %f %f\n",t,Vas,ias,Pqs,Pds,Pqr,Pdr,wr,Te,Tm);
}

```

```
fclose(fi);
printf("READY\n");
printf("DATAFILE: 'min.dat'\n\n");
}

/**DEFINICION DE ECUACIONES DIFERENCIALES EN MODELO DE ESPACIO DE ESTADO***/
double funPqs(double Pqs,double Pmq,double Vqs,double wb,double rs,double Xls)
{
    double funPqs;
    funPqs=wb*(Vqs+(rs/Xls)*(Pmq-Pqs));
    return(funPqs);
}
double funPds(double Pds,double Pmd,double Vds,double wb,double rs,double Xls)
{
    double funPds;
    funPds=wb*(Vds+(rs/Xls)*(Pmd-Pds));
    return(funPds);
}
double funPqr(double Pqr,double Pdr,double wr,double Pmq,double wb,double rr,double Xlr)
{
    double funPqr;
    funPqr=wb*((rr/Xlr)*(Pmq-Pqr)+wr*Pdr);
    return(funPqr);
}
double funPdr(double Pqr,double Pdr,double wr,double Pmd,double wb,double rr,double Xlr)
{
    double funPdr;
    funPdr=wb*((rr/Xlr)*(Pmd-Pdr)-wr*Pqr);
    return(funPdr);
}
double funwr(double Te,double Tm,double H)
{
    double funwr;
    funwr=(1/(2*H))*(Te-Tm); // wr/wb
    return(funwr);
}
```

## E.4. Modulación de Vectores Espaciales.

### //MODULACION DE VECTORES ESPACIALES (SVM)

```

#include <string.h>
#include <iostream.h>
#include <fstream.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>

double sector(double);
double stq(double,double);
double std(double,double);

void main()
{
    double i,j,S;
    double to,tf,n,h,hh,t;
    double vq,vd,vas,vbs,vcs,va,vb,vc,V;
    double Vref,Vcd,Vq,Vd,Vas,Vbs,Vcs,Va,Vb,Vc;
    double alfa,beta,wf,f,fs,N,pi;
    double m,K,T,Ts,To,ta,tb,tz,ts;

    FILE *fi;
    fi=fopen("svm.dat","w");
    to=0.0;
    tf=0.05;
    h=0.00001;
    hh=tf-to;
    n=hh/h;
    Vcd=1.7; //VOLTAJE DE C.D.
    Vref=1.0; //VOLTAJE NOMIMAL
    f=60; //FRECUENCIA NOMINAL
    T=1/f; //PERIODO
    fs=1800; //FRECUENCIA DE SWITCHEO
    Ts=1/fs; //PERIODO DE SWITCHEO
    N=fs/f;
    pi=3.1416;
    wf=2*pi*f; //VELOCIDAD ANGULAR
    t=to;

    printf("RUNNING SIMULATION\n");
    for (i=0;i<=n;i++)
    {
        t=t+h;
        if(t>0.025)
            Vref=0.45; //CAMBIO EN LA MAGNITUD DE Vref

    /******GENERACION DEL VECTOR DE REFERENCIA******/
        vq=Vref*cos(wf*t);

```

```

vd=-Vref*sin(wf*t);
V=sqrt(pow(vd,2)+pow(vq,2));
alfa=atan2(vq,vd);
if (alfa<0.0)
  alfa=alfa+(2*pi);

/*****SPACE VECTOR MODULATOR (SVM)*****/
m=V/(2*Vcd/pi);           //INDICE DE MODULACION
S=sector(alfa);           //SECTOR DEL PLANO DQ
beta=alfa-(S-1)*pi/3;     //ANGULO DE Vref DENTRO DEL SECTOR
To=Ts/2;
K=(sqrt(3)*To)/(4*Vcd);
ta=2*K*V*sin((pi/3)-beta); //TIEMPO DE APLICACION DEL VECTOR Vx
tb=2*K*V*sin(beta);       //TIEMPO DE APLICACION DEL VECTOR Vy
tz=To-(ta+tb);           //TIEMPO DE APLICACION DEL VECTOR CERO
tz=tz/2;
ts=0.0;

for(j=1;j<=N*(tf/T);j++)
{
    if(t>=ts && t<ts+tz)
    {
        Vq=0.0;
        Vd=0.0;
    }
    else if(t>=ts+tz && t<=ts+tz+ta)
    {
        Vq=stq(S,Vcd);
        Vd=std(S,Vcd);
    }
    else if(t>=ts+tz+ta && t<=ts+tz+ta+tb)
    {
        Vq=stq(S+1,Vcd);
        Vd=std(S+1,Vcd);
    }
    else if(t>=ts+tz+ta+tb && t<=ts+(3/2*tz)+ta+tb)
    {
        Vq=0.0;
        Vd=0.0;
    }
    else if(t>=ts+(3/2*tz)+ta+tb && t<=ts+(3/2*tz)+ta+2*tb)
    {
        Vq=stq(S+1,Vcd);
        Vd=std(S+1,Vcd);
    }
    else if(t>=ts+(3/2*tz)+ta+2*tb && t<=ts+(3/2*tz)+2*ta+2*tb)
    {
        Vq=stq(S,Vcd);
        Vd=std(S,Vcd);
    }
    else if(t>=ts+(3/2*tz)+2*ta+2*tb && t<=ts+2*tz+2*ta+2*tb)
    {
        Vq=0.0;
    }
}

```

```

        Vd=0.0;
    }
    ts=ts+Ts;
}

/*****TRANSFORMACION DQ-ABC*****/
Va=Vq; //VOLTAJE DE LINEA FASE A
Vb=-((Vq+sqrt(3)*Vd)/2); //VOLTAJE DE LINEA FASE B
Vc=-((Vq-sqrt(3)*Vd)/2); //VOLTAJE DE LINEA FASE C
Vas=(Va-Vc)/3; //VOLTAJE DE FASE FASE A
Vbs=(-Va+Vb)/3; //VOLTAJE DE FASE FASE B
Vcs=(-Vb+Vc)/3; //VOLTAJE DE FASE FASE C

//SEÑALES DE REFERENCIA
va=vq;
vb=-((vq+sqrt(3)*vd)/2);
vc=-((vq-sqrt(3)*vd)/2);
vas=(va-vc)/3;
vbs=(-va+vb)/3;
vcs=(-vb+vc)/3;

//IMPRESION DE RESULTADOS
fprintf(fi,"%f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f\n",
        t,Vas,vas,Vbs,vbs,Vcs,vcs,Va,va,Vb,vb,Vc,vc,Vq,vq,Vd,vd);
}

fclose(fi);
printf("READY\n");
printf("DATAFILE: 'svm.dat'\n\n");
}

/*****FUNCIONES*****/
double sector(double alfa)
{
    double S,pi;
    pi=3.1416;
    if (alfa>=0 && alfa<=pi/3)
        S=1.0;
    else if (alfa>=pi/3 && alfa<=2*pi/3)
        S=2.0;
    else if (alfa>=2*pi/3 && alfa<=pi)
        S=3.0;
    else if (alfa>=pi && alfa<=4*pi/3)
        S=4.0;
    else if (alfa>=4*pi/3 && alfa<=5*pi/3)
        S=5;
    else if(alfa>=5*pi/3 && alfa<=2*pi)
        S=6.0;
    return(S);
}

double stq(double S,double Vcd)
{

```



```
double Vq,pi;
pi=3.1416;
Vq=2*Vcd*sin((S-1)*(pi/3))/3;
return(Vq);
}
double std(double S, double Vcd)
{
double Vd,pi;
pi=3.1416;
Vd=2*Vcd*cos((S-1)*(pi/3))/3;
return(Vd);
}
```

## E.5. Red Neuro-Difusa.

```

//ANFIS
//2 ENTRADAS, 1 SALIDA
//3 FUNCIONES DE MEMBRESIA TIPO GAUSS POR ENTRADA
//9 REGLAS DIFUSAS
//18 PARAMETROS ANTECEDENTES Y 27 CONSECUENTES

#include <string.h>
#include <iostream.h>
#include <fstream.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>

double bell(double,double,double,double);
double sright(double,double,double,double);
double sleft(double,double,double,double);

void escmat(double [31][31],int,int,int);
void trans (double [31][31],double [31][31],int nfa_,int nca_);
void mumat(double [31][31],double [31][31],double [31][31],int,int,int);
void divmat(double [31][31],double esc,double [31][31],int nfa_,int nca_);
void muesc(double [31][31],double esc,double [31][31],int nfa_,int nca_);
void resmat(double [31][31],double [31][31],double [31][31],int nfa_,int nca_);
void summat(double [31][31],double [31][31],double [31][31],int nfa_,int nca_);

void main()
{
    int i,j,k;
    int n_par,n_pun,gama,lamda;
    double to,tf,t,h,hh,n;
    double a,b,cn,cz,cp,x1,x2,y;
    double  w11,w12,w13,w14,w15,w16,w21,w22,w23,w24,w25,w26,w27,w28,w29,w31,w32,
            w33,w34,w35,w36,w37,w38,w39,w41,w42,w43,w44,w45,w46,w47,w48,w49,W,
            OUT;
    double  p1,p2,p3,p4,p5,p6,p7,p8,p9,q1,q2,q3,q4,q5,q6,q7,q8,q9,r1,r2,r3,r4,r5,r6,r7,r8,r9,
            f1,f2,f3,f4,f5,f6,f7,f8,f9;
    double  A[31][31],At[31][31],P[31][31],C[31][31],res[31][31],res1[31][31],res2[31][31],
            res3[31][31],res4[31][31],res5[31][31],res6[31][31],res7[31][31],res8[31][31],
            res9[31][31],esc,esc1,CON;

    FILE *fi;
    fi=fopen("anf.dat","w");

    to=0;
    tf=3;
    h=0.0001;
    hh=tf-to;
    n=hh/h;
    t=to;

```

```

/*****PARAMETROS DEL FILTRO DE KALMAN*****/
n_pun=1;
n_par=27;
gama=5000;
lamda=100;
CON=0.0;
for(i=1;i<=n_par;i++)
  for(j=1;j<=n_pun;j++)
    C[i][j]=0.0;
for(i=1;i<=n_par;i++)
  for(j=1;j<=n_par;j++)
  {
    if(i==j)
      P[i][j]=gama;
    else
      P[i][j]=0.0;
  }

/*****INICIO DE ITERACIONES*****/
printf("RUNNING SIMULATION\n");
for(i=1;i<=n;i++)
{
  t=t+h;

/*****ENTRADAS Y SALIDA DESEADA DE LA RED*****/
  x1=10*t; //SEÑAL DE PRUEBA
  x2=cos(6*t); //SEÑAL DE PRUEBA
  y=sin(x1)*sin(x2)/x1*x2*sin(x1*x2); //SEÑAL DE PRUEBA

/*****PARAMETROS DE LAS FUNCIONES DE MEMBRESIA*****/
  a=23;b=1;cn=30;cz=75,cp=120;

//LAYER 1 (GRADO DE MEMBRESIA DE LAS ENTRADAS)
  w11=sleft(x1,a,b,cn); w12=bell(x1,a,b,cz); w13=sright(x1,a,b,cp);
  w14=sleft(x2,a,b,cn); w15=bell(x2,a,b,cz); w16=sright(x2,a,b,cp);

//LAYER 2 (OPERADOR T-NORM: valor minimo o producto)
  w21=w11*w14; w22=w11*w15; w23=w11*w16;
  w24=w12*w14; w25=w12*w15; w26=w12*w16;
  w27=w13*w14; w28=w13*w15; w29=w13*w16;

//LAYER 3 (NORMALIZACION DE LOS PESOS)
  W=w21+w22+w23+w24+w25+w26+w27+w28+w29;
  w31=w21/W; w32=w22/W; w33=w23/W;
  w34=w24/W; w35=w25/W; w36=w26/W;
  w37=w27/W; w38=w28/W; w39=w29/W;

//FILTRO DE KALMAN
//MATRIZ a
A[1][1]=w31*x1; A[1][2]=w31*x2; A[1][3]=w31;
A[1][4]=w32*x1; A[1][5]=w32*x2; A[1][6]=w32;
A[1][7]=w33*x1; A[1][8]=w33*x2; A[1][9]=w33;
A[1][10]=w34*x1; A[1][11]=w34*x2; A[1][12]=w34;

```

```

A[1][13]=w35*x1;          A[1][14]=w35*x2;          A[1][15]=w35;
A[1][16]=w36*x1;          A[1][17]=w36*x2;          A[1][18]=w36;
A[1][19]=w37*x1;          A[1][20]=w37*x2;          A[1][21]=w37;
A[1][22]=w38*x1;          A[1][23]=w38*x2;          A[1][24]=w38;
A[1][25]=w39*x1;          A[1][26]=w39*x2;          A[1][27]=w39;

//MATRIZ TRANSPUESTA DE a
trans (A,At,n_pun,n_par);

//OBTENCION DE LA MATRIZ P(k+1)
//P(k+1)=P-(P*at*a*P/1+a*P*at)
mumat (P,At,res,n_par,n_pun,n_pun);
mumat (res,A,res1,n_par,n_pun,n_par);
mumat (res1,P,res2,n_par,n_par,n_par);
mumat (A,res,res3,n_pun,n_par,n_pun);
esc=res3[1][1]+1.0;
divmat(res2,esc,res4,n_par,n_par);
resmat(P,res4,res5,n_par,n_par);
for(j=1;j<=n_par;j++)
  for(k=1;k<=n_par;k++)
    P[j][k]=res5[j][k];

//OBTENCION DE LA MATRIZ C(k+1)
//C(k+1)=C+P(k+1)*at(y-a*C)
mumat (P,At,res6,n_par,n_pun,n_pun);
mumat(A,C,res7,n_pun,n_par,n_pun);
esc1=y-res7[1][1];
muesc(res6,esc1,res8,n_par,n_pun);
summat(C,res8,res9,n_par,n_pun);
for(j=1;j<=n_par;j++)
  for(k=1;k<=n_pun;k++)
    C[j][k]=res9[j][k];

//RESETEO DE LA MATRIZ DE COVARIANZA P
CON++;
if(CON==lamda)
  {
  CON=0.0;
  for(j=1;j<=n_par;j++)
    for(k=1;k<=n_par;k++)
      {
      if(j==k)
        P[j][k]=gama;
      else
        P[j][k]=0.0;
      }
  }

//LAYER 4 (NODO ADAPTABLE)
p1=C[1][1];          q1=C[2][1];          r1=C[3][1];
p2=C[4][1];          q2=C[5][1];          r2=C[6][1];
p3=C[7][1];          q3=C[8][1];          r3=C[9][1];
p4=C[10][1];         q4=C[11][1];         r4=C[12][1];
p5=C[13][1];         q5=C[14][1];         r5=C[15][1];

```

```

p6=C[16][1];      q6=C[17][1];      r6=C[18][1];
p7=C[19][1];      q7=C[20][1];      r7=C[21][1];
p8=C[22][1];      q8=C[23][1];      r8=C[24][1];
p9=C[25][1];      q9=C[26][1];      r9=C[27][1];

f1=p1*x1+q1*x2+r1;  f2=p2*x1+q2*x2+r2;  f3=p3*x1+q3*x2+r3;
f4=p4*x1+q4*x2+r4;  f5=p5*x1+q5*x2+r5;  f6=p6*x1+q6*x2+r6;
f7=p7*x1+q7*x2+r7;  f8=p8*x1+q8*x2+r8;  f9=p9*x1+q9*x2+r9;

w41=w31*f1;      w42=w32*f2;      w43=w33*f3;
w44=w34*f4;      w45=w35*f5;      w46=w36*f6;
w47=w37*f7;      w48=w38*f8;      w49=w39*f9;

//LAYER 5 (SUMA GLOBAL)
OUT=w41+w42+w43+w44+w45+w46+w47+w48+w49;

//IMPRESION DE RESULTADOS
fprintf(fi,"%f %f %f %f %f\n",t,x1,x2,y,OUT);

}
printf("READY\n");
printf("DATAFILE: 'anf.dat'\n\n");
fclose(fi);

}

/*****FUNCIONES*****/
//FUNCION DE MEMBRESIA CAMPANA
double bell(double x,double a,double b,double c)
{
double y;
y=1/(1+pow((x-c)/a,2*b));
if(y<0.003)
y=0.0;
return(y);
}

//FUNCION DE MEMBRESIA S-DERECHA
double sright(double x,double a,double b,double c)
{
double y;
if (x>=c)
y=1.0;
else
y=1/(1+pow((x-c)/a,2*b));
if(y<0.003)
y=0.0;
return(y);
}

//FUNCION DE MEMBRESIA S-IZQUIERDA
double sleft(double x,double a,double b,double c)
{
double y;

```

```
if (x<=c)
    y=1.0;
else
    y=1/(1+pow((x-c)/a,2*b));
if(y<0.003)
    y=0.0;
return(y);
}

//FUNCION TRANSPUESTA DE UNA MATRIZ
void trans (double a_[31][31],double b_[31][31],int nfa_,int nca_)
{
    int i,j;
    for (i=1;i<=nfa_;i++)
        for (j=1;j<=nca_;j++)
            b_[j][i]=a_[i][j];
}

//FUNCION MULTIPLICACION DE MATRICES
void mumat (double a_[31][31],double b_[31][31],double c_[31][31],int nfa_,int nca_,int ncb_)
{
    int i,j,k;
    for (i=1;i<=nfa_;i++)
        for (j=1;j<=ncb_;j++)
        {
            c_[i][j]=0.0;
            for (k=1;k<=nca_;k++)
                c_[i][j]=c_[i][j]+a_[i][k]*b_[k][j];
        }
}

//FUNCION ESCRIBE EN PANTALLA MATRIZ
void escmat(double a_[31][31],int filas_,int colu_,int fii_)
{
    int i,j,fil,col;
    fil=fii_;
    for (i=1;i<=filas_;i++)
    {
        col=2;
        for (j=1;j<=colu_;j++)
        {
            printf ("%0.4f",a_[i][j]);
            col=col+10;
        }
        fil=fil+1;
    }
}

//FUNCION DIVISION DE MATRICES
void divmat (double a_[31][31],double esc,double b_[31][31],int nfa_,int nca_)
{
    int i,j;
    for (i=1;i<=nfa_;i++)
        for (j=1;j<=nca_;j++)
```

```
b_[i][j]=a_[i][j]/(1+esc);  
}
```

```
//FUNCION MULTIPLICACION DE ESCALAR POR MATRIZ  
void muesc (double a_[31][31],double esc,double b_[31][31],int nfa_,int nca_)  
{  
int i,j;  
for (i=1;i<=nfa_;i++)  
for (j=1;j<=nca_;j++)  
b_[i][j]=a_[i][j]*esc;  
}
```

```
//FUNCION RESTA DE MATRICES  
void resmat(double a_[31][31],double b_[31][31],double c_[31][31],int nfa_,int nca_)  
{  
int i,j;  
for (i=1;i<=nfa_;i++)  
for (j=1;j<=nca_;j++)  
c_[i][j]=a_[i][j]-b_[i][j];  
}
```

```
//FUNCION SUMA DE MATRICES  
void summat(double a_[31][31],double b_[31][31],double c_[31][31],int nfa_,int nca_)  
{  
int i,j;  
for (i=1;i<=nfa_;i++)  
for (j=1;j<=nca_;j++)  
c_[i][j]=a_[i][j]+b_[i][j];  
}
```

## E.6. DTNFC - Entrenamiento.

```

//DTNFC
//FASE DE ENTRENAMIENTO DE LA RED NEURO-DIFUSA

#include <string.h>
#include <iostream.h>
#include <fstream.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>

double bell(double,double,double,double);
double sright(double,double,double,double);
double sleft(double,double,double,double);
void escmat(double [31][31],int,int,int);
void trans (double [31][31],double [31][31],int nfa_,int nca_);
void mumat(double [31][31],double [31][31],double [31][31],int,int,int);
void divmat(double [31][31],double esc,double [31][31],int nfa_,int nca_);
void muesc(double [31][31],double esc,double [31][31],int nfa_,int nca_);
void resmat(double [31][31],double [31][31],double [31][31],int nfa_,int nca_);
void summat(double [31][31],double [31][31],double [31][31],int nfa_,int nca_);
double funPqs(double,double,double,double,double,double);
double funPds(double,double,double,double,double,double);
double funPqr(double,double,double,double,double,double,double);
double funPdr(double,double,double,double,double,double,double);
double funwr(double,double,double);
double sector(double);
double stq(double,double);
double std(double,double);

void main()
{
    int i,j,k,l;
    int n_par,n_pun,gama;
    double a,b,cn,cz,cp,x1,x2,y,lamda,CON;
    double w11,w12,w13,w14,w15,w16,w21,w22,w23,w24,w25,w26,w27,w28,w29,w31,w32,
        w33,w34,w35,w36,w37,w38,w39,w41,w42,w43,w44,w45,w46,w47,w48,w49,W,OUT;
    double p1,p2,p3,p4,p5,p6,p7,p8,p9,q1,q2,q3,q4,q5,q6,q7,q8,q9,r1,r2,r3,r4,r5,r6,r7,r8,r9,
        f1,f2,f3,f4,f5,f6,f7,f8,f9;
    double A[31][31],At[31][31],P[31][31],C[31][31],res[31][31],res1[31][31],res2[31][31],
        res3[31][31],res4[31][31],res5[31][31],res6[31][31],res7[31][31],res8[31][31],
        res9[31][31],esc,esc1;
    double h,hh,h2,t,to,tf,n,S;
    double Pqs,Pds,Ps,THps,Pqr,Pdr,Pmq,Pmd,Vqs,Vds,iqs,ids,iqr,idr,Te,Tm,eT,eP,Tref,Pref;
    double Vas,Vbs,Vcs,ias,ibs,ics;
    double rs,rr,Xlr,Xls,Xm,Xaq,Xad,f,wf,pi,wb,wr,H,Vs_ref,th[10],v[10],vx,vy;
    double k_Pqs,k_Pds,k_Pqr,k_Pdr,k_wr,m_Pqs,m_Pds,m_Pqr,m_Pdr,m_wr;
    double Vcd,Vq,Vd,alfa,alfa60,N,fs;
    double K,T,Ts,To,ta,tb,tz,ts;

```



```

FILE *fi, *fj;
fi=fopen("dte.dat", "w");
fj=fopen("par.dat", "w");
/*****PARAMETROS DE SIMULACION*****/
to=0;
tf=0.5;
h=0.00001;
hh=tf-to;
n=hh/h;
h2=h/2;

/*****PARAMETROS DEL INVERSOR*****/
Vcd=2.0;
f=60;
T=1/f;
fs=15000;
Ts=1/fs;
N=fs/f;
pi=3.1416;
wf=2*pi*f;

/*****PARAMETROS DEL MOTOR*****/
rs=0.0453;
rr=0.0222;
Xls=0.0775;
Xlr=0.0322;
Xm=2.042;
wb=377;
Tm=1.0;
H=0.5;
Xaq=1/((1/Xm)+(1/Xls)+(1/Xlr));
Xad=Xaq;

/*****PARAMETROS DEL FILTRO DE KALMAN*****/
n_pun=1;
n_par=27;
gama=100;
lamda=10;
CON=0.0;
for(i=1;i<=n_par;i++)
    for(j=1;j<=n_pun;j++)
        C[i][j]=0.0;

for(i=1;i<=n_par;i++)
    for(j=1;j<=n_par;j++)
    {
        if(i==j)
            P[i][j]=gama;
        else
            P[i][j]=0.0;
    }

/*****CONDICIONES INICIALES DEL SISTEMA *****/
Pqs=-0.747359;
Pds=-0.664695;

```

```

Pqr=-0.733256;
Pdr=-0.634919;
wr=0.998917;
Ps=sqrt(pow(Pds,2)+pow(Pqs,2));           //MAGNITUD DEL VECTOR DE FLUJO
THps=atan2(Pqs,Pds);                     //ANGULO DEL VECTOR DE FLUJO
if(THps<0)
    THps=THps+2*pi;
else if(THps>2*pi)
    THps=THps-2*pi;

alfa=THps;
t=to;
Pref=1.0;                                //SEÑAL DE REFERENCIA DEL FLUJO
Tref=1.0;                                //SEÑAL DE REFERENCIA DEL PAR

/*****INICIO DE ITERACIONES*****/
printf("RUNNING SIMULATION\n");

/*****CALCULO DE VOLTAJES abc*****/
for (i=1;i<=n+1;i++)
{
    t=t+h;
    if(t>0.0)                            //CAMBIOS EN LAS SEÑALES DE
        Pref=1.0;                        //REFERENCIA
    if(t>0.05)
        Tref=1.0;
    if(t>0.3)                             //CAMBIO EN EL PAR MECANICO
        Tm=1.0;
/*****SPACE VECTOR MODULATOR (SVM)*****/
    Vs_ref=2.5;
    S=sector(alfa);
    alfa60=alfa-(S-1)*pi/3;
    To=Ts/2;
    K=(sqrt(3)*To)/(4*Vcd);
    ta=2*K*Vs_ref*sin((pi/3)-alfa60);
    tb=2*K*Vs_ref*sin(alfa60);
    tz=To-(ta+tb);
    tz=tz/2;
    ts=0.0;
    for(j=1;j<=N*(tf/T);j++)
    {
        if(t>=ts && t<ts+tz)
        {
            Vq=0.0;
            Vd=0.0;
        }
        else if(t>=ts+tz && t<=ts+tz+ta)
        {
            Vq=stq(S,Vcd);
            Vd=std(S,Vcd);
        }
        else if(t>=ts+tz+ta && t<=ts+tz+ta+tb)
        {
            Vq=stq(S+1,Vcd);

```

```

        Vd=std(S+1,Vcd);
    }
    else if(t>=ts+tz+ta+tb && t<=ts+(3/2*tz)+ta+tb)
    {
        Vq=0.0;
        Vd=0.0;
    }
    else if(t>=ts+(3/2*tz)+ta+tb && t<=ts+(3/2*tz)+ta+2*tb)
    {
        Vq=stq(S+1,Vcd);
        Vd=std(S+1,Vcd);
    }
    else if(t>=ts+(3/2*tz)+ta+2*tb && t<=ts+(3/2*tz)+2*ta+2*tb)
    {
        Vq=stq(S,Vcd);
        Vd=std(S,Vcd);
    }
    else if(t>=ts+(3/2*tz)+2*ta+2*tb && t<=ts+Ts)
    {
        Vq=0.0;
        Vd=0.0;
    }
    ts=ts+Ts;
}
Vqs=Vq;
Vds=Vd;

/*****MOTOR DE INDUCCION*****/
Pmq=Xaq*((Pqs/Xls)+(Pqr/Xlr));
Pmd=Xad*((Pds/Xls)+(Pdr/Xlr));
iqs=(1/Xls)*(Pqs-Pmq);
ids=(1/Xls)*(Pds-Pmd);
iqr=(1/Xlr)*(Pqr-Pmq);
idr=(1/Xlr)*(Pdr-Pmd);
Te=(Pds*iqs)-(Pqs*ids);
k_Pqs=h*funPqs(Pqs,Pmq,Vqs,wb,rs,Xls);
k_Pds=h*funPds(Pds,Pmd,Vds,wb,rs,Xls);
k_Pqr=h*funPqr(Pqr,Pdr,wr,Pmq,wb,rr,Xlr);
k_Pdr=h*funPdr(Pqr,Pdr,wr,Pmd,wb,rr,Xlr);
k_wr=h*funwr(Te,Tm,H);
m_Pqs=h*funPqs(Pqs+h2*k_Pqs,Pmq,Vqs,wb,rs,Xls);
m_Pds=h*funPds(Pds+h2*k_Pds,Pmd,Vds,wb,rs,Xls);
m_Pqr=h*funPqr(Pqr+h2*k_Pqr,Pdr+h2*k_Pdr,wr,Pmq,wb,rr,Xlr);
m_Pdr=h*funPdr(Pqr+h2*k_Pqr,Pdr+h2*k_Pdr,wr,Pmd,wb,rr,Xlr);
m_wr=h*funwr(Te,Tm,H);
Pqs=Pqs+m_Pqs;
Pds=Pds+m_Pds;
Pqr=Pqr+m_Pqr;
Pdr=Pdr+m_Pdr;
wr=wr+m_wr;

/*****ESTIMACION DEL PAR Y FLUJO*****/
Te=(Pds*iqs)-(Pqs*ids);
Ps=sqrt(pow(Pds,2)+pow(Pqs,2));

```

```

    THps=atan2(Pqs,Pds);
    if(THps<0)
        THps=THps+2*pi;
    else if(THps>2*pi)
        THps=THps-2*pi;

    eT=Tref-Te;           //SEÑAL DE ERROR DEL PAR
    eP=Pref-Ps;         //SEÑAL DE ERROR DEL FLUJO

/*****NEURO-FUZZY CONTROLLER*****/
//ENTRADAS Y SALIDA DESEADA
x1=eP;
x2=eT;
y=Vs_ref;

//PARAMETROS DE LAS FUNCIONES DE MEMBRESIA
a=0.01;b=2;cn=-0.02;cz=0.0,cp=0.02;

//LAYER 1
w11=sleft(x1,a,b,cn);   w12=bell(x1,a,b,cz);   w13=sright(x1,a,b,cp);
w14=sleft(x2,a,b,cn);   w15=bell(x2,a,b,cz);   w16=sright(x2,a,b,cp);

//LAYER 2
w21=w11*w14;           w22=w11*w15;           w23=w11*w16;
w24=w12*w14;           w25=w12*w15;           w26=w12*w16;
w27=w13*w14;           w28=w13*w15;           w29=w13*w16;

//LAYER 3
W=w21+w22+w23+w24+w25+w26+w27+w28+w29;
w31=w21/W;           w32=w22/W;           w33=w23/W;
w34=w24/W;           w35=w25/W;           w36=w26/W;
w37=w27/W;           w38=w28/W;           w39=w29/W;

//FILTRO DE KALMAN
A[1][1]=w31*x1;       A[1][2]=w31*x2;       A[1][3]=w31;
A[1][4]=w32*x1;       A[1][5]=w32*x2;       A[1][6]=w32;
A[1][7]=w33*x1;       A[1][8]=w33*x2;       A[1][9]=w33;
A[1][10]=w34*x1;      A[1][11]=w34*x2;      A[1][12]=w34;
A[1][13]=w35*x1;      A[1][14]=w35*x2;      A[1][15]=w35;
A[1][16]=w36*x1;      A[1][17]=w36*x2;      A[1][18]=w36;
A[1][19]=w37*x1;      A[1][20]=w37*x2;      A[1][21]=w37;
A[1][22]=w38*x1;      A[1][23]=w38*x2;      A[1][24]=w38;
A[1][25]=w39*x1;      A[1][26]=w39*x2;      A[1][27]=w39;

trans (A,At,n_pun,n_par);
mumat (P,At,res,n_par,n_pun,n_pun);
mumat (res,A,res1,n_par,n_pun,n_par);
mumat (res1,P,res2,n_par,n_pun,n_par);
mumat (A,res,res3,n_pun,n_par,n_pun);
esc=res3[1][1]+1.0;
divmat(res2,esc,res4,n_par,n_par);
resmat(P,res4,res5,n_par,n_par);
for(j=1;j<=n_par;j++)
    for(k=1;k<=n_par;k++)

```

```

        P[j][k]=res5[j][k];
mumat (P,At,res6,n_par,n_par,n_pun);
mumat(A,C,res7,n_pun,n_par,n_pun);
esc1=y-res7[1][1];
muesc(res6,esc1,res8,n_par,n_pun);
summat(C,res8,res9,n_par,n_pun);
for(j=1;j<=n_par;j++)
    for(k=1;k<=n_pun;k++)
        C[j][k]=res9[j][k];

CON++;
if(CON==lamda)
    {
    CON=0.0;
    for(j=1;j<=n_par;j++)
        for(k=1;k<=n_pun;k++)
            {
            if(j==k)
                P[j][k]=gama;
            else
                P[j][k]=0.0;
            }
    }

//LAYER 4
p1=C[1][1];      q1=C[2][1];      r1=C[3][1];
p2=C[4][1];      q2=C[5][1];      r2=C[6][1];
p3=C[7][1];      q3=C[8][1];      r3=C[9][1];
p4=C[10][1];     q4=C[11][1];     r4=C[12][1];
p5=C[13][1];     q5=C[14][1];     r5=C[15][1];
p6=C[16][1];     q6=C[17][1];     r6=C[18][1];
p7=C[19][1];     q7=C[20][1];     r7=C[21][1];
p8=C[22][1];     q8=C[23][1];     r8=C[24][1];
p9=C[25][1];     q9=C[26][1];     r9=C[27][1];

f1=p1*x1+q1*x2+r1;      f2=p2*x1+q2*x2+r2;      f3=p3*x1+q3*x2+r3;
f4=p4*x1+q4*x2+r4;      f5=p5*x1+q5*x2+r5;      f6=p6*x1+q6*x2+r6;
f7=p7*x1+q7*x2+r7;      f8=p8*x1+q8*x2+r8;      f9=p9*x1+q9*x2+r9;

w41=w31*f1;      w42=w32*f2;      w43=w33*f3;
w44=w34*f4;      w45=w35*f5;      w46=w36*f6;
w47=w37*f7;      w48=w38*f8;      w49=w39*f9;

//LAYER 5
OUT=w41+w42+w43+w44+w45+w46+w47+w48+w49;

//LAYER 5' (ESTIMACION DEL ANGULO DEL VECTOR DE VOLTAJE)
v[1]=w41;      v[2]=w42;      v[3]=w43;
v[4]=w44;      v[5]=w45;      v[6]=w46;
v[7]=w47;      v[8]=w48;      v[9]=w49;
th[1]=THps+(2*pi/3);  th[2]=THps+(pi);      th[3]=THps+(2*pi/3);
th[4]=THps+(-pi/2);  th[5]=THps+(pi/2);    th[6]=THps+(pi/2);
th[7]=THps+(-pi/3);  th[8]=THps;           th[9]=THps+(pi/3);
vx=0.0;vy=0.0;

```

```

for(l=1;l<=9;l++)
    {
        vx=vx+v[l]*cos(th[l]);
        vy=vy+v[l]*sin(th[l]);
    }
alfa=atan2(vy,vx);
if(alfa<0)
    alfa=alfa+2*pi;
else if(alfa>2*pi)
    alfa=alfa-2*pi;

/*****TRANFORMACION DE CORRIENTES dq-abc*****/
Vas=Vqs;
Vbs=-(Vqs+sqrt(3)*Vds)/2;
Vcs=-(Vqs-sqrt(3)*Vds)/2;
ias=iqs;
ibs=-(iqs+sqrt(3)*ids)/2;
ics=-(iqs-sqrt(3)*ids)/2;

//IMPRESION DE RESULTADOS EN ARCHIVO DE DATOS
fprintf(fi,"%f %f %f %f %f %f %f %f %f %f %f %f %f %f %f\n",
        t,Te,THps*180/pi,alfa*180/pi,alfa60*180/pi,S,Vs_ref,OUT,Pqs,Pds,Ps,
        Vqs,Vds,wr,iqs);

}

//IMPRESION DE LOS PARAMETROS CONSECUENTES
for(i=1;i<=n_par;i++)
    fprintf(fj,"%f",C[i][1]);
fclose(fi);
fclose(fj);
printf("READY\n\n");
}

/*****FUNCIONES*****/
double funPqs(double Pqs,double Pmq,double Vqs,double wb,double rs,double Xls)
    {
        double funPqs;
        funPqs=wb*(Vqs+(rs/Xls)*(Pmq-Pqs));
        return(funPqs);
    }
double funPds(double Pds,double Pmd,double Vds,double wb,double rs,double Xls)
    {
        double funPds;
        funPds=wb*(Vds+(rs/Xls)*(Pmd-Pds));
        return(funPds);
    }
double funPqr(double Pqr,double Pdr,double wr,double Pmq,double wb,double rr,double Xlr)
    {
        double funPqr;
        funPqr=wb*((rr/Xlr)*(Pmq-Pqr)+wr*Pdr);
        return(funPqr);
    }
double funPdr(double Pqr,double Pdr,double wr,double Pmd,double wb,double rr,double Xlr)

```

```

        {
            double funPdr;
            funPdr=wb*((rr/Xlr)*(Pmd-Pdr)-wr*Pqr);
            return(funPdr);
        }
double funwr(double Te,double Tm,double H)
    {
        double funwr;
        funwr=(1/(2*H))*(Te-Tm); /*wr/wb*/
        return(funwr);
    }
double sector(double alfa)
{
    double S,pi;
    pi=3.1416;
    if (alfa>=0 && alfa<pi/3)
        S=1.0;
    else if (alfa>=pi/3 && alfa<2*pi/3)
        S=2.0;
    else if (alfa>=2*pi/3 && alfa<pi)
        S=3.0;
    else if (alfa>=pi && alfa<4*pi/3)
        S=4.0;
    else if (alfa>=4*pi/3 && alfa<5*pi/3)
        S=5;
    else if(alfa>=5*pi/3 && alfa<2*pi)
        S=6.0;
    return(S);
}

double stq(double S,double Vcd)
    {
        double Vq,pi;
        pi=3.1416;
        Vq=2*Vcd*sin((S-1)*(pi/3))/3;
        return(Vq);
    }
double std(double S, double Vcd)
    {
        double Vd,pi;
        pi=3.1416;
        Vd=2*Vcd*cos((S-1)*(pi/3))/3;
        return(Vd);
    }
double bell(double x,double a,double b,double c)
    {
        double y;
        y=1/(1+pow((x-c)/a,2*b));
        if(y<0.003)
            y=0.0;
        return(y);
    }
double sright(double x,double a,double b,double c)
    {

```

```

        double y;
        if (x>=c)
            y=1.0;
        else
            y=1/(1+pow((x-c)/a,2*b));
        if(y<0.003)
            y=0.0;
        return(y);
    }
double sleft(double x,double a,double b,double c)
{
    double y;
    if (x<=c)
        y=1.0;
    else
        y=1/(1+pow((x-c)/a,2*b));
    if(y<0.003)
        y=0.0;
    return(y);
}

double max(double a,double b)
{
    double y;
    if (a>b)
        y=a;
    else
        y=b;
    return(y);
}

void trans (double a_[31][31],double b_[31][31],int nfa_,int nca_)
{
    int i,j;
    for (i=1;i<=nfa_;i++)
        for (j=1;j<=nca_;j++)
            b_[j][i]=a_[i][j];
}

void mumat (double a_[31][31],double b_[31][31],double c_[31][31],int nfa_,int nca_,int ncb_)
{
    int i,j,k;
    for (i=1;i<=nfa_;i++)
        for (j=1;j<=ncb_;j++)
            {
                c_[i][j]=0.0;
                for (k=1;k<=nca_;k++)
                    c_[i][j]=c_[i][j]+a_[i][k]*b_[k][j];
            }
}

void escmat(double a_[31][31],int filas_,int colu_,int fii_)
{
    int i,j,fil,col;
    fil=fii_;

```



```
        for (i=1;i<=filas_;i++)
        {
            col=2;
            for (j=1;j<=colu_;j++)
            {
                printf ("%0.4f",a_[i][j]);
                col=col+10;
            }
            fil=fil+1;
        }
}
void divmat (double a_[31][31],double esc,double b_[31][31],int nfa_,int nca_)
{
    int i,j;
    for (i=1;i<=nfa_;i++)
    for (j=1;j<=nca_;j++)
        b_[i][j]=a_[i][j]/(1+esc);
}
void muesc (double a_[31][31],double esc,double b_[31][31],int nfa_,int nca_)
{
    int i,j;
    for (i=1;i<=nfa_;i++)
    for (j=1;j<=nca_;j++)
        b_[i][j]=a_[i][j]*esc;
}
void resmat(double a_[31][31],double b_[31][31],double c_[31][31],int nfa_,int nca_)
{
    int i,j;
    for (i=1;i<=nfa_;i++)
    for (j=1;j<=nca_;j++)
        c_[i][j]=a_[i][j]-b_[i][j];
}
void summat(double a_[31][31],double b_[31][31],double c_[31][31],int nfa_,int nca_)
{
    int i,j;
    for (i=1;i<=nfa_;i++)
    for (j=1;j<=nca_;j++)
        c_[i][j]=a_[i][j]+b_[i][j];
}
```

## E.7. DTNFC - Aplicación.

```

//DTNFC
//FASE DE APLICACION

#include <string.h>
#include <iostream.h>
#include <fstream.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>

double bell(double,double,double,double);
double sright(double,double,double,double);
double sleft(double,double,double,double);
double funPqs(double,double,double,double,double,double);
double funPds(double,double,double,double,double,double);
double funPqr(double,double,double,double,double,double,double);
double funPdr(double,double,double,double,double,double,double);
double funwr(double,double,double);
double sector(double);
double stq(double,double);
double std(double,double);

void main()
{

double a,b,cn,cz,cp,x1,x2;
double w11,w12,w13,w14,w15,w16,w21,w22,w23,w24,w25,w26,w27,w28,w29,w31,w32,
w33,w34,w35,w36,w37,w38,w39,w41,w42,w43,w44,w45,w46,w47,w48,w49,W,Vs_ref;
float p1,p2,p3,p4,p5,p6,p7,p8,p9,q1,q2,q3,q4,q5,q6,q7,q8,q9,r1,r2,r3,r4,r5,r6,r7,r8,r9,
f1,f2,f3,f4,f5,f6,f7,f8,f9;
int i,j,l;
double h,hh,h2,t,to,tf,n,n_rules,S;
double Pqs,Pds,Ps,Pqr,Pdr,Pmq,Pmd,Vas,Vbs,Vcs,Vqs,Vds,iqs,ids,iqr,idr,eP,eT,Te,
Tref,Pref,Tm;
double rs,rr,Xlr,Xls,Xm,Xaq,Xad,f,wf,pi,wb,wr,H,ias,ibs,ics;
double k_Pqs,k_Pds,k_Pqr,k_Pdr,k_wr,m_Pqs,m_Pds,m_Pqr,m_Pdr,m_wr;
double Vcd,Vq,Vd,alfa,alfa60,THps,N,fs,v[10],th[10],vx,vy;
double K,T,Ts,To,ta,tb,tz,ts;

FILE *fi,*fj;
fi=fopen("dta.dat","w");
fj=fopen("par.dat","r");

/*****PARAMETROS DE SIMULACION*****/
to=0;
tf=1.0;
h=0.00001;
hh=tf-to;
n=hh/h;

```

```

h2=h/2;
n_rules=9;

/*****PARAMETROS DEL INVERSOR*****/
Vcd=2.0;
f=60;
T=1/f;
fs=15000;
Ts=1/fs;
N=fs/f;
pi=3.1416;
wf=2*pi*f;

/*****PARAMETROS DEL MOTOR*****/
rs=0.0453;
rr=0.0222;
Xls=0.0775;
Xlr=0.0322;
Xm=2.042;
wb=377;
Tm=1.0;
Tref=0.1666;
Pref=0.57;
H=0.5;
Xaq=1/((1/Xm)+(1/Xls)+(1/Xlr));
Xad=Xaq;

/*****CONDICIONES INICIALES DEL SISTEMA *****/
Pqs=-0.747359;
Pds=-0.664695;
Pqr=-0.733256;
Pdr=-0.634919;
wr=0.998917;
Te=0.1666;
Ps=sqrt(pow(Pds,2)+pow(Pqs,2));
THps=atan2(Pqs,Pds);
if(THps<0)
    THps=THps+2*pi;
else if(THps>2*pi)
    THps=THps-2*pi;
eT=0.0;
eP=0.0;
t=to;

//LECTURA DE LOS PARAMETROS CONSECUENTES
fscanf(fj,"%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,%f",
        &p1,&q1,&r1,&p2,&q2,&r2,&p3,&q3,&r3,&p4,&q4,&r4,&p5,&q5,&r5,&p6,&q6,&r6,&p7,&q7,&r7,&p8,&q8,&r8,&p9,&q9,&r9);

/*****INICIO DE ITERACIONES*****/
printf("RUNNING SIMULATION\n\n");
for (i=1;i<=n;i++)
{
    t=t+h;

```

```

if(t>0.2)
    Tref=0.833;
if(t>0.2)
    Ps=0.6;

eT=Tref-Te;
eP=Pref-Ps;

/*****NEURO-FUZZY CONTROLLER*****/
//ENTRADAS
x1=eP;
x2=eT;

//PARAMETROS DE LAS FUNCIONES DE MEMBRESIA
a=0.01;b=2;cn=-0.02;cz=0.0,cp=0.02;

//LAYER 1
w11=sleft(x1,a,b,cn);  w12=bell(x1,a,b,cz);  w13=sright(x1,a,b,cp);
w14=sleft(x2,a,b,cn);  w15=bell(x2,a,b,cz);  w16=sright(x2,a,b,cp);

//LAYER 2
w21=w11*w14;  w22=w11*w15;  w23=w11*w16;
w24=w12*w14;  w25=w12*w15;  w26=w12*w16;
w27=w13*w14;  w28=w13*w15;  w29=w13*w16;

//LAYER 3
W=w21+w22+w23+w24+w25+w26+w27+w28+w29;
w31=w21/W;  w32=w22/W;  w33=w23/W;
w34=w24/W;  w35=w25/W;  w36=w26/W;
w37=w27/W;  w38=w28/W;  w39=w29/W;

//LAYER 4
f1=p1*x1+q1*x2+r1;  f2=p2*x1+q2*x2+r2;  f3=p3*x1+q3*x2+r3;
f4=p4*x1+q4*x2+r4;  f5=p5*x1+q5*x2+r5;  f6=p6*x1+q6*x2+r6;
f7=p7*x1+q7*x2+r7;  f8=p8*x1+q8*x2+r8;  f9=p9*x1+q9*x2+r9;

w41=w31*f1;  w42=w32*f2;  w43=w33*f3;
w44=w34*f4;  w45=w35*f5;  w46=w36*f6;
w47=w37*f7;  w48=w38*f8;  w49=w39*f9;

//LAYER 5 (MAGNITUD DEL VECTOR DE VOLTAJE)
Vs_ref=w41+w42+w43+w44+w45+w46+w47+w48+w49;

//CALCULO DEL ANGULO DEL VECTOR DE VOLTAJE
v[1]=w41;  v[2]=w42;  v[3]=w43;
v[4]=w44;  v[5]=w45;  v[6]=w46;
v[7]=w47;  v[8]=w48;  v[9]=w49;
th[1]=THps+(2*pi/3);  th[2]=THps+(pi);  th[3]=THps+(2*pi/3);
th[4]=THps+(-pi/2);  th[5]=THps+(pi/2);  th[6]=THps+(pi/2);
th[7]=THps+(-pi/3);  th[8]=THps;  th[9]=THps+(pi/3);

vx=0.0;vy=0.0;
for(l=1;l<=n_rules;l++)
{

```

```

        vx=vx+v[l]*cos(th[l]);
        vy=vy+v[l]*sin(th[l]);
    }

    alfa=atan2(vy,vx);
    if(alfa<0)
        alfa=alfa+2*pi;
    else if(alfa>2*pi)
        alfa=alfa-2*pi;

    /*****SPACE VECTOR MODULATOR (SVM)*****/
    S=sector(alfa);
    alfa60=alfa-(S-1)*pi/3;
    To=Ts/2;
    K=(sqrt(3)*To)/(4*Vcd);
    ta=2*K*Vs_ref*sin((pi/3)-alfa60);
    tb=2*K*Vs_ref*sin(alfa60);
    tz=To-(ta+tb);
    tz=tz/2;
    ts=0.0;

    for(j=1;j<=N*(tf/T);j++)
    {

        if(t>=ts && t<ts+tz)
        {
            Vq=0.0;
            Vd=0.0;
        }
        else if(t>=ts+tz && t<=ts+tz+ta)
        {
            Vq=stq(S,Vcd);
            Vd=std(S,Vcd);
        }
        else if(t>=ts+tz+ta && t<=ts+tz+ta+tb)
        {
            Vq=stq(S+1,Vcd);
            Vd=std(S+1,Vcd);
        }
        else if(t>=ts+tz+ta+tb && t<=ts+(3/2*tz)+ta+tb)
        {
            Vq=0.0;
            Vd=0.0;
        }
        else if(t>=ts+(3/2*tz)+ta+tb && t<=ts+(3/2*tz)+ta+2*tb)
        {
            Vq=stq(S+1,Vcd);
            Vd=std(S+1,Vcd);
        }
        else if(t>=ts+(3/2*tz)+ta+2*tb && t<=ts+(3/2*tz)+2*ta+2*tb)
        {
            Vq=stq(S,Vcd);
            Vd=std(S,Vcd);
        }
    }

```

```

        else if(t>=ts+(3/2*tz)+2*ta+2*tb && t<=ts+Ts)
        {
            Vq=0.0;
            Vd=0.0;
        }
        ts=ts+Ts;
    }
    Vqs=Vq;
    Vds=Vd;

/*****MOTOR DE INDUCCION*****/
Pmq=Xaq*((Pqs/Xls)+(Pqr/Xlr));
Pmd=Xad*((Pds/Xls)+(Pdr/Xlr));
iqs=(1/Xls)*(Pqs-Pmq);
ids=(1/Xls)*(Pds-Pmd);
iqr=(1/Xlr)*(Pqr-Pmq);
idr=(1/Xlr)*(Pdr-Pmd);
Te=(Pds*iqs)-(Pqs*ids);
k_Pqs=h*funPqs(Pqs,Pmq,Vqs,wb,rs,Xls);
k_Pds=h*funPds(Pds,Pmd,Vds,wb,rs,Xls);
k_Pqr=h*funPqr(Pqr,Pdr,wr,Pmq,wb,rr,Xlr);
k_Pdr=h*funPdr(Pqr,Pdr,wr,Pmd,wb,rr,Xlr);
k_wr=h*funwr(Te,Tm,H);
m_Pqs=h*funPqs(Pqs+h2*k_Pqs,Pmq,Vqs,wb,rs,Xls);
m_Pds=h*funPds(Pds+h2*k_Pds,Pmd,Vds,wb,rs,Xls);
m_Pqr=h*funPqr(Pqr+h2*k_Pqr,Pdr+h2*k_Pdr,wr,Pmq,wb,rr,Xlr);
m_Pdr=h*funPdr(Pqr+h2*k_Pqr,Pdr+h2*k_Pdr,wr,Pmd,wb,rr,Xlr);
m_wr=h*funwr(Te,Tm,H);
Pqs=Pqs+m_Pqs;
Pds=Pds+m_Pds;
Pqr=Pqr+m_Pqr;
Pdr=Pdr+m_Pdr;
wr=wr+m_wr;

/*****ESTIMACION DEL PAR Y FLUJO*****/
Te=(Pds*iqs)-(Pqs*ids);
Ps=sqrt(pow(Pds,2)+pow(Pqs,2));
THps=atan2(Pqs,Pds);
if(THps<0)
    THps=THps+2*pi;
else if(THps>2*pi)
    THps=THps-2*pi;

Vas=Vqs;
Vbs=-(Vqs+sqrt(3)*Vds)/2;
Vcs=-(Vqs-sqrt(3)*Vds)/2;
ias=iqs;
ibs=-(iqs+sqrt(3)*ids)/2;
ics=-(iqs-sqrt(3)*ids)/2;

//IMPRESION DE RESULTADOS
fprintf(fi,"%f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f\n",
        t,Pqs,Pds,Pqr,Pqs,wr,Te,Vqs,Vds,iqs,ids,Vs_ref,Ps,eT,eP,Pref,Tref,
        THps*180/pi,alfa*180/pi);

```

```

    }
    fclose(fi);
    fclose(fj);
    printf("READY\n\n");
}

/*****FUNCIONES*****/
double funPqs(double Pqs,double Pmq,double Vqs,double wb,double rs,double Xls)
{
    double funPqs;
    funPqs=wb*(Vqs+(rs/Xls)*(Pmq-Pqs));
    return(funPqs);
}
double funPds(double Pds,double Pmd,double Vds,double wb,double rs,double Xls)
{
    double funPds;
    funPds=wb*(Vds+(rs/Xls)*(Pmd-Pds));
    return(funPds);
}
double funPqr(double Pqr,double Pdr,double wr,double Pmq,double wb,double rr,double Xlr)
{
    double funPqr;
    funPqr=wb*((rr/Xlr)*(Pmq-Pqr)+wr*Pdr);
    return(funPqr);
}
double funPdr(double Pqr,double Pdr,double wr,double Pmd,double wb,double rr,double Xlr)
{
    double funPdr;
    funPdr=wb*((rr/Xlr)*(Pmd-Pdr)-wr*Pqr);
    return(funPdr);
}
double funwr(double Te,double Tm,double H)
{
    double funwr;
    funwr=(1/(2*H))*(Te-Tm); /*wr/wb*/
    return(funwr);
}
double sector(double THps)
{
    double S,pi;
    pi=3.1416;
    if (THps>=0 && THps<=pi/3)
        S=1.0;
    else if (THps>=pi/3 && THps<=2*pi/3)
        S=2.0;
    else if (THps>=2*pi/3 && THps<=pi)
        S=3.0;
    else if (THps>=pi && THps<=4*pi/3)
        S=4.0;
    else if (THps>=4*pi/3 && THps<=5*pi/3)
        S=5;
    else if(THps>=5*pi/3 && THps<=2*pi)
        S=6.0;
    return(S);
}

```

```
    }  
  
double stq(double S,double Vcd)  
{  
    double Vq,pi;  
    pi=3.1416;  
    Vq=2*Vcd*sin((S-1)*(pi/3))/3;  
    return(Vq);  
}  
  
double std(double S, double Vcd)  
{  
    double Vd,pi;  
    pi=3.1416;  
    Vd=2*Vcd*cos((S-1)*(pi/3))/3;  
    return(Vd);  
}  
  
double bell(double x,double a,double b,double c)  
{  
    double y;  
    y=1/(1+pow((x-c)/a,2*b));  
    if(y<0.003)  
        y=0.0;  
    return(y);  
}  
  
double sright(double x,double a,double b,double c)  
{  
    double y;  
    if (x>=c)  
        y=1.0;  
    else  
        y=1/(1+pow((x-c)/a,2*b));  
    if(y<0.003)  
        y=0.0;  
    return(y);  
}  
  
double sleft(double x,double a,double b,double c)  
{  
    double y;  
    if (x<=c)  
        y=1.0;  
    else  
        y=1/(1+pow((x-c)/a,2*b));  
    if(y<0.003)  
        y=0.0;  
    return(y);  
}
```