



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SECCIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

Estimación de la Resistencia del Rotor en el
Control Vectorial Indirecto del Motor de
Inducción utilizando una Red Neuronal Artificial

TESIS

QUE PARA OBTENER EL GRADO DE:

**MAESTRO EN CIENCIAS
EN INGENIERÍA ELÉCTRICA**

P R E S E N T A:

Pedro Francisco Huerta González



México, D. F.

Noviembre de 2005



INSTITUTO POLITECNICO NACIONAL
COORDINACION GENERAL DE POSGRADO E INVESTIGACION

CARTA CESIÓN DE DERECHOS

En la Ciudad de México, Distrito Federal, el día 20 del mes Mayo del año 2005, el (la) que suscribe Pedro Francisco Huerta González alumno(a) del Programa de Posgrado de Ingeniería Eléctrica con número de registro B001195, 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. Jaime José Rodríguez Rivas y cede los derechos del trabajo intitulado: "Estimación de la Resistencia del Rotor en el Control Vectorial Indirecto del Motor de Inducción utilizando una Red Neuronal Artificial", 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: phuertaglez@hotmail.com, jjrodriguezr@ipn.mx

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

Pedro Francisco Huerta González
Nombre y firma



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

ACTA DE REVISION DE TESIS

En la Ciudad de México, D. F. siendo las 13:30 horas del día 25 del mes de Febrero del 2005 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:

“ESTIMACION DE LA RESISTENCIA DEL ROTOR EN EL CONTROL VECTORIAL INDIRECTO DEL MOTOR DE INDUCCION UTILIZANDO UNA RED NEURONAL ARTIFICIAL”

Presentada por el alumno:

HUERTA

GONZALEZ

PEDRO FRANCISCO

Apellido paterno

materno

nombre(s)

Con registro:

B	0	0	1	1	9	5
---	---	---	---	---	---	---

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

DR. JAIME JOSE RODRIGUEZ RIVAS

DR. DANIEL OLGUIN SALINAS

DR. RAUL CORTES MATEOS

M. EN C. TOMAS I. ASIAIN OLIVARES

DR. DAVID ROMERO ROMERO

M. EN C. JESUS REYES GARCIA

EL PRESIDENTE DEL COLEGIO

SECCION DE ESTUDIOS DE POSGRADO E INVESTIGACION

DR. FLORENCIO SANCHEZ SILVA

RESUMEN

En el presente trabajo se muestra la simulación del control vectorial de la velocidad de un motor de inducción con el método indirecto, utilizándose un estimador neuronal de la resistencia del rotor para incrementar el desempeño del control y evitar su desintonización debido a la desorientación del fasor espacial del flujo magnético del rotor.

El trabajo inicia con una descripción del modelo del motor de inducción utilizado y del control de la velocidad. En el control se utilizó un regulador de corriente síncrono donde se muestran los resultados que se obtienen pudiéndose valorar el desempeño de todo el sistema. También se analiza cualitativamente la influencia de las desviaciones del valor de la resistencia del rotor en el desempeño del sistema y se muestra la necesidad de la estimación en línea y de la actualización de su valor en el algoritmo de control. Mediante simulación se muestra el desempeño del sistema de control cuando se realiza la actualización en línea de la resistencia del rotor y cuando su valor no es actualizado mediante un estimador.

Para el estimador desarrollado en esta tesis, se utilizó un estimador de la resistencia del rotor basándose en la función de referencia, la función estimada y en un controlador difuso, el cual se tomó de la literatura, simulándose en el sistema de control y observando algunos problemas en su desempeño específicamente en condiciones dinámicas. Para mejorar el desempeño se diseñó un estimador neuronal tomando como patrón al estimador de la literatura. Mediante simulación se demuestra el mejor comportamiento del sistema de control cuando se utiliza el estimador neuronal comparado con el estimador de la literatura. La mayor aportación de este trabajo, es la mejora en el desempeño del sistema de control sobre un sistema similar que aparece en la literatura. Los resultados de este trabajo fueron presentados en The 40th International Universities Power Engineering Conference (UPEC-2005).

ABSTRACT

Presently work is shown the simulation of the speed vector control of an induction motor with the indirect method, being used a neural estimator of the rotor resistance to increase the performance of the control and to avoid it's detuning due to the disorientation of the rotor magnetizing flux space phasor.

The work begins with a description of the used model of the induction motor and speed control. In the control a synchronous current regulator was used, where the results are shown that they are obtained being able to value the performance of the complete system. It is also analyzed qualitatively the influence of the deviations of the value of the rotor resistance in the performance of the system, and the requirement of the on-line estimate is shown and of the upgrade of its value in the control algorithm. By means of simulation the performance of the control system is shown when is carried out the on-line upgrade of the rotor resistance and when its value is not upgraded by means of an estimator.

For the estimator developed in this thesis, an estimator of the rotor resistance was used being based on the reference function, the estimated function and in a fuzzy controller, which took of the literature, being simulated in the control system and observing some problems specifically in its performance under dynamic conditions. To improve the performance a neural estimator it was designed taking as pattern to the estimator of the literature. By means of simulation the best behavior in the control system is demonstrated when the neural estimator is used compared with the estimator of the literature. The biggest contribution in this work is the improvement in the performance of the control system on a similar system that appears in the literature. The results of this work were presented in The 40th International Universities Power Engineering Conference (UPEC-2005).

ÍNDICE

RESUMEN	i
ABSTRACT	ii
GLOSARIO DE SÍMBOLOS	vii
LISTADO DE FIGURAS	xv
LISTADO DE TABLAS	xix
1 DESCRIPCIÓN Y OBJETIVOS DEL TRABAJO DE TESIS	
1.1 Objetivos generales	1
1.2 Justificación	1
1.3 Estado del arte	2
1.4 Aportaciones	4
1.5 Contenido del trabajo	4
2 CONTROL VECTORIAL INDIRECTO	
2.1 Introducción	5
2.2 Modelo y simulación del motor de inducción	6
2.2.1 Modelo del motor de inducción	6
2.2.2 Simulación del motor de inducción	9
2.3 Control vectorial indirecto del motor de inducción	13
2.4 Regulador vectorial síncrono	16
2.5 Controlador PI autosintonizado con lógica difusa	17
2.5.1 Acción de control del controlador PI	18
2.5.2 Autosintonización difusa	19
2.5.3 Reglas difusas	19
2.5.4 Función de membresía de la premisa	20
2.5.5 Parámetro de la consecuencia	21
2.5.6 Razonamiento	22
2.6 Simulación del control vectorial indirecto con regulador vectorial síncrono	23
2.7 Resultados de la simulación del control vectorial indirecto	28
2.7.1 Campo debilitado	29
2.7.2 Par de carga variable a una velocidad de referencia fija	30
2.7.3 Cambio cíclico de la velocidad de referencia	34

3	ESTIMACIÓN DE LA RESISTENCIA DEL ROTOR UTILIZANDO UNA RED NEURONAL ARTIFICIAL	
3.1	Introducción	38
3.2	Efecto de los cambios de parámetros en el control vectorial indirecto	39
3.3	Adaptación de los parámetros en el control vectorial indirecto	42
3.4	Descripción del método usado para la estimación y actualización de la resistencia del rotor	44
3.5	Estimación de la resistencia del rotor basado en el uso de las funciones F_{est} , F_{act} y en un controlador difuso	46
3.5.1	Análisis de la señal de entrada al estimador	46
3.5.2	Estimación de la resistencia del rotor	48
3.6	Estimación de la resistencia del rotor basado en una red neuronal artificial	50
3.6.1	Introducción	50
3.6.2	Descripción general de la red neuronal artificial	51
3.6.3	Estructura y normalización de la red neuronal artificial	53
3.6.4	Generación de datos y entrenamiento de la red	54
4	RESULTADOS DE LA SIMULACIÓN	
4.1	Introducción	58
4.2	Simulación de la estimación de la resistencia del rotor basado en las funciones F_{est} , F_{act} y en un controlador difuso	59
4.3	Simulación del estimador neuronal de la resistencia del rotor en el control vectorial indirecto con regulador de corriente síncrono	62
4.3.1	Capa de entradas	62
4.3.2	Capa de salida	64
4.3.3	Entrenamiento de la red neuronal artificial	64
4.3.3.1	Entrenamiento con 1 capa oculta con 3 neuronas	65
4.3.3.2	Entrenamiento con 2 capas ocultas	65
4.3.3.3	Entrenamiento con 1 capa oculta con 5 neuronas	66
4.3.4	Estimador neuronal de la resistencia del rotor	70
4.4	Resultados de la simulación del control vectorial indirecto sin la estimación de la resistencia del rotor	72
4.5	Resultados de la simulación de la estimación de la resistencia del rotor en estado estable con el motor de inducción de 20 HP	74
4.5.1	Primer caso estable	74
4.5.2	Segundo caso estable	78
4.6	Resultados de la simulación de la estimación de la resistencia del rotor en estado transitorio con el motor de inducción de 20 HP	83
4.6.1	Primer caso transitorio	83
4.6.2	Segundo caso transitorio	88
4.6.3	Tercer caso transitorio	93

4.6.4	Cuarto caso transitorio	98
4.7	Resultados de la simulación del estimador neuronal con otros motores de inducción	104
4.7.1	Simulación con un motor de inducción de 100 HP	104
4.7.2	Simulación con un motor de inducción de 1 HP	107
4.7.3	Simulación con un motor de inducción de 5 HP	108
5	CONCLUSIONES Y RECOMENDACIONES	
5.1	Conclusiones	111
5.2	Recomendaciones a trabajos futuros	113
Apéndice A PROGRAMAS EN MATLAB		
A.1	Programa <i>m1.m</i> : Parámetros de la maquina de inducción trifásica de 20 HP	114
A.2	Programa <i>m2.m</i> : Parámetros de la maquina de inducción trifásica de 20 HP para el control vectorial indirecto	115
A.3	Programa <i>m3.m</i> : Entrenamiento de la red neuronal para el estimador de la resistencia del rotor	118
A.4	Programa <i>m4.m</i> : Simulación en estado estable de los estimadores de la resistencia del rotor	119
A.5	Programa <i>m5.m</i> : Simulación en estado transitorio de los estimadores de la resistencia del rotor	120
A.6	Programa <i>m6.m</i> : Parámetros de la maquina de inducción trifásica de 100 HP	122
Apéndice B OPERACIÓN CON CAMPO DEBILITADO		
B.1	Introducción	124
B.2	Estrategia de control de la máquina de inducción para la operación de potencia constante	125
B.3	Obtención del flujo del rotor ψ_{dr}^{te} de referencia y la velocidad mecánica ω_{rm} del rotor	126
Apéndice C PRINCIPIOS DE LAS REDES NEURONALES ARTIFICIALES		
C.1	Introducción	127
C.2	Red neuronal artificial multicapa	129
C.3	Estructura de la red neuronal artificial multicapa	131
C.4	Regla de aprendizaje de la red neuronal artificial multicapa	132
C.5	Algoritmo de Levenberg - Marquardt	140
C.6	Descripción de las funciones utilizadas en MATLAB	144
C.6.1	Red tipo Perceptrón	144
C.6.2	Red tipo Adaline	144

C.6.3	Red tipo Retropropagación	145
C.6.4	Algoritmos de entrenamiento para la red de Retropropagación	145
C.6.4.1	Traingd	145
C.6.4.2	Traingdm	146
C.6.4.3	Traingda	146
C.6.4.4	Trainrp	147
C.6.4.5	Trainbfg	147
C.6.4.6	Trainlm	148
	Apéndice D	149
	BIBLIOGRAFÍA	156

GLOSARIO DE SÍMBOLOS

SÍMBOLO	DESCRIPCIÓN
r_s	Resistencia del estator
r_r'	Resistencia del rotor.
L_r'	Inductancia del rotor
L_m	Inductancia de magnetización del estator
L_s	Inductancia del estator
L_M	Inductancia mutua.
T_r	Constante de tiempo del rotor real
T_r^*	Constante de tiempo del rotor predicha
T_{r0}	Constante de tiempo del rotor a la carga nominal
T_r^{con}	Constante de tiempo del rotor en el circuito de control
X_{lr}'	Reactancia de enlace del rotor
X_r	Reactancia del rotor
X_{ls}	Reactancia de enlace del estator
X_s	Reactancia del estator
X_M	Reactancia mutua
X_m	Reactancia de magnetización del estator
X'	Reactancia transitoria del estator
σ	Coefficiente de dispersión
Z_b	Impedancia base
C_{sg}	Capacitor entre el neutro del devanado del estator (s) y de la fuente (g)
J	Inercia del rotor
H	Constante de inercia
P	Numero de Polos

SÍMBOLO	DESCRIPCIÓN
T_{mec}	Par mecánico
T_b	Par base
T_{nom}	Par nominal
T_{em}	Par electromagnético
T_{amort}	Par de amortiguamiento
v_{ag}, v_{bg}, v_{cg}	Voltajes de la fuente de alimentación del estator
v_{as}, v_{bs}, v_{cs}	Voltajes de fase del estator
v_{an}, v_{bn}, v_{cn}	Voltajes de la fuente de alimentación del rotor
v_{ar}, v_{br}, v_{cr}	Voltajes de fase del rotor
v_{sg}	Voltaje entre el neutro del devanado del estator (s) y de la fuente (g)
v_{rn}	Voltaje entre el neutro del devanado del rotor (r) y de la fuente (n)
v_{0s}	Voltaje del estator de secuencia cero
v'_{0r}	Voltaje del rotor de secuencia cero
$v'_{dr}{}^s$	Voltaje del rotor del eje d en el marco de referencia estacionario
$v'_{qr}{}^s$	Voltaje del rotor del eje q en el marco de referencia estacionario
$v'_{dr}{}^r$	Voltajes de la frecuencia de deslizamiento del eje d
$v'_{qr}{}^r$	Voltajes de la frecuencia de deslizamiento del eje q
$v_{ds}{}^s$	Voltaje del estator del eje d en el marco de referencia estacionario
$v_{qs}{}^s$	Voltaje del estator del eje q en el marco de referencia estacionario
V_a^*, V_b^*, V_c^*	Voltajes de referencia
V_m	Voltaje máximo
$v_{ds}{}^e$	Voltaje del estator del eje d en el marco de referencia síncrono
$v_{qs}{}^e$	Voltaje del estator del eje q en el marco de referencia síncrono
E'	Voltaje presente en la reactancia transitoria del estator

SÍMBOLO	DESCRIPCIÓN
i'_{dr}	Corriente del rotor del eje d en el marco de referencia del rotor
i'_{qr}	Corriente del rotor del eje q en el marco de referencia del rotor
i_{0s}	Corriente del estator de secuencia cero
i'^s_{qr}	Corriente del rotor del eje q en el marco de referencia estacionario
i'^s_{dr}	Corriente del rotor del eje d en el marco de referencia estacionario
i'_{0r}	Corriente del rotor de secuencia cero
i^s_{ds}	Corriente del estator del eje d en el marco de referencia estacionario
i^s_{qs}	Corriente del estator del eje q en el marco de referencia estacionario
i_{as}, i_{bs}, i_{cs}	Corrientes de fase del estator
i_{ar}, i_{br}, i_{cr}	Corrientes de fase del rotor
i^e_{ds}	Corriente del estator del eje d en el marco de referencia síncrono
i^e_{qs}	Corriente del estator del eje q en el marco de referencia síncrono
i^e_{dr}	Corriente del rotor del eje d en el marco de referencia síncrono
i^e_{dsN}	Corriente nominal del estator del eje d en el marco de referencia síncrono
i^e_{qsN}	Corriente nominal del estator del eje q en el marco de referencia síncrono
i_{Mr}	Corriente de magnetización
i_{ds}	Corriente del estator del eje d , componente de flujo
i_{qs}	Corriente del estator del eje q , componente de par
i_{sN}	Corriente del estator nominal
i_s	Corriente del estator
S_N	Potencia nominal
$\varepsilon_a, \varepsilon_b, \varepsilon_c$	Error entre las corrientes reales y de referencia
ω_{rm}	Velocidad mecánica del rotor

SÍMBOLO	DESCRIPCIÓN
ω_{ref}	Velocidad mecánica del rotor de referencia
ω_e	Velocidad angular del estator
θ_r	Angulo del rotor
ω_b	Velocidad angular base
ω_{bm}	Velocidad mecánica base
ω_r	Velocidad del rotor
ω_2	Velocidad de deslizamiento
θ_2	Angulo de deslizamiento
ρ	Angulo del vector espacial flujo del rotor
f_N	Frecuencia nominal
ψ_{ds}^s	Enlace de flujo del estator del eje d en el marco de referencia estacionario
ψ_{qs}^s	Enlace de flujo del estator del eje q en el marco de referencia estacionario
ψ_{dr}^{ts}	Enlace de flujo del rotor del eje d en el marco de referencia estacionario
ψ_{qr}^{ts}	Enlace de flujo del rotor del eje q en el marco de referencia estacionario
ψ_{md}^s	Enlace de flujo mutuo del eje d en el marco de referencia estacionario
ψ_{mq}^s	Enlace de flujo mutuo del eje q en el marco de referencia estacionario
ψ_{dr}^e	Enlace de flujo del rotor del eje d en el marco de referencia síncrono
ψ_{qr}^e	Enlace de flujo del rotor del eje q en el marco de referencia síncrono
ψ_r^{ts}	Flujo del rotor en el marco de referencia estacionario
ψ_r^{t*}	Flujo del rotor, valor de referencia
ψ_{dr}^{te*}	Enlace de flujo del rotor del eje d en el marco de referencia síncrono, valor de referencia
F_{act}	Función real
F_{est}	Función estimada

SÍMBOLO	DESCRIPCIÓN
E_F	Error entre las funciones F_{est} y F_{act}
ΔE_F	Cambio de error entre las funciones F_{est} y F_{act}
$\varepsilon_F(k)$	Error de la etapa de <i>fuzzificación</i> entre las funciones F_{est} y F_{act}
$\Delta \varepsilon_F(k)$	Cambio de error de la etapa de <i>fuzzificación</i> entre las funciones F_{est} y F_{act}
k_E	Ganancia para normalizar el error
$k_{\Delta E}$	Ganancia para normalizar el cambio del error
,	Valores referidos al lado del estator
*	Valores de referencia o comando
S_a, S_b, S_c	Estados de conmutación
n	Neutro de la fuente del rotor
g	Neutro de la fuente del estator
r	Punto de referencia del rotor
s	Punto de referencia del estator
$qd0$	Marco de referencia estacionario
abc	Sistema trifásico
a_s, b_s, c_s	Fases del estator
a_r, b_r, c_r	Fases del rotor
p	Operador de Laplace
K_p	Ganancia proporcional
ΔK_p	Cambio de la ganancia proporcional
τ_I	Tiempo integral
$\Delta \tau_I$	Cambio del tiempo integral
u	Acción de control del controlador PI
ε	Error entre la señal de referencia y la señal real
$\Delta \varepsilon$	Cambio de error entre la señal de referencia y la señal real
y_r	Señal de referencia

SÍMBOLO	DESCRIPCIÓN
y	Señal de salida de la planta
A, B	Valores lingüísticos de las variables lingüísticas del <i>Error</i> , $\varepsilon(k)$, y del <i>Cambio de Error</i> , $\Delta\varepsilon(k)$
C, D	Valores lingüísticos de las variables lingüísticas de la <i>Ganancia Proporcional</i> , $K_P(k)$ y del <i>Tiempo Integral</i> $\tau_I(k)$
N, Z, P	Variables lingüísticas de $\varepsilon(t)$ y $\Delta\varepsilon(t)$: negativo, cero y positivo, respectivamente
N, Z, P	Variables lingüísticas de K_P y τ_I : negativo, cero y positivo, respectivamente
I, M	Número de reglas
$NG, NM, NP, Z, PP, PM, PG$	Variables lingüísticas de $\Delta r_r(k)$: Negativo Grande, Negativo Mediano, Negativo Pequeño, Cero, Positivo Pequeño, Positivo Mediano y Positivo Grande, respectivamente
$\Delta r_r(k)$	Incremento de la resistencia del rotor en por unidad obtenido en la etapa de defuzzificación
$\Delta R_r(k)$	Incremento de la resistencia del rotor real
$E_{red}(k)$	Error entre la resistencia del rotor de referencia y la calculada con la red
$k_{\Delta r}$	Ganancia para obtener $\Delta R_r(k)$
$r'_{r(est)}$	Resistencia del rotor estimada
$r'_{r(ref)}$	Resistencia del rotor estimada de referencia para el entrenamiento de la red
$r'_{r(real)}$	Resistencia del rotor real
$\mu_A(\bullet)$	Función de membresía de A
$\mu_A(x)$	Grado de la función de membresía de x en A
α_i, α_m	Operación difusa de intersección
COG	Centro de gravedad
x_i	Entradas de la red neuronal
y_j	Salidas de la red neuronal

SÍMBOLO	DESCRIPCIÓN
W_i	Pesos
W_1	Matriz de pesos para la primera capa
W_{ji}^o	Peso que une la componente i de la entrada con la neurona j de la capa oculta.
W_{kj}^s	Peso que une la neurona j de la capa oculta con la neurona k de la capa de salida, la cual cuenta con s neuronas.
k	Intervalo de muestreo presente
$(k-i)$	Retardo en los intervalos de i -muestreo
ep^2	Error medio cuadrático para cada patrón de entrada p .
$e(x)$	Método de Newton para optimizar el rendimiento
e^2	Error total en el proceso de aprendizaje en una iteración luego de haber presentado a la red los r patrones de entrenamiento.
f_i	Función de activación
f^s	Función de transferencia de las neuronas de la capa de salida.
f^o	Función de transferencia de las neuronas de la capa oculta.
S	Número de neuronas
a_k^s	Salida de la red de cada neurona
S^M	Sensitividad para la capa
a_j^o	Salida de la neurona j de la capa oculta..
n_k^s	Entrada neta a la neurona k de la capa de salida
p_i	Componente i del vector p que contiene el patrón de entrenamiento de q componentes
p_Q	Entrada a la red
b_j^o	Ganancia de la neurona j de la capa oculta
b_k^s	Ganancia de la neurona k de la capa de salida.
δ_k	Error en la neurona k de la capa de salida con l neuronas
∇ep^2	Gradiente negativo de ep^2

SÍMBOLO	DESCRIPCIÓN
$\nabla e(x)$	Gradiente del método de Newton
$\frac{\partial ep^2}{\partial W_{kj}^s}$	Componente del gradiente $-\nabla ep^2$ respecto al peso de la conexión de la neurona de la capa de salida y la neurona j de la capa oculta W_{kj}^s .
$\frac{\partial a_k^s}{\partial W_{kj}^s}$	Derivada de la salida de la neurona k de la capa de salida respecto, al peso W_{kj}^s .
$\frac{\partial a_k^s}{\partial n_k^s}$	Derivada de la salida de la neurona k de la capa de salida respecto a su entrada neta
$\frac{\partial n_k^s}{\partial W_{kj}^s}$	Derivada de la entrada neta a la neurona k de la capa de salida respecto a los pesos de la conexión entre las neuronas de la capa oculta y la capa de salida.
θ	Función umbral
μ	Constante que determina la tendencia del algoritmo
\mathbf{J}^T	Matriz Jacobiana
m	Número de neuronas de la capa oculta.
σ	Coefficiente de dispersión
α	Índice de aprendizaje que varía entre 0 y 1 dependiendo de las características del problema a solucionar
a_1	Valores entregados por la red
t_1	Valores esperados por la red
η	Coefficiente de aprendizaje

LISTADO DE FIGURAS

- Figura 2.1 Flujo de variables en la simulación de los circuitos en los ejes q y d .
- Figura 2.2 Simulación del motor de inducción en el marco de referencia estacionario.
- Figura 2.3 Curvas de estado estable de la máquina de inducción.
- Figura 2.4 Arranque del motor de inducción sin carga y con cambios de carga.
- Figura 2.5 Orientación adecuada en el marco de referencia síncrono.
- Figura 2.6 Control vectorial indirecto del motor de inducción alimentado con un inversor de voltaje con corrientes reguladas.
- Figura 2.7 Esquema funcional básico del regulador de corriente para el inversor PWM.
- Figura 2.8 Regulador de corriente lineal síncrono que trabaja en coordenadas estacionarias con componentes de CA.
- Figura 2.9 Estructura general del autosintonizador difuso.
- Figura 2.10 Funciones de membresía.
- Figura 2.11 Curva del error de la salida.
- Figura 2.12 Simulación en SIMULINK del control vectorial indirecto del motor de inducción con inversor de voltaje PWM con el regulador de corriente síncrono.
- Figura 2.13 Campo debilitado.
- Figura 2.14 Arranque con variación de carga a velocidad fija con regulador de corriente síncrono en el control vectorial indirecto.
- Figura 2.15 Gráficas de las trayectorias de las corrientes estacionarias dqs en el regulador de corriente síncrono. Con variación de carga a velocidad fija.
- Figura 2.16 Arranque con variación de la velocidad de referencia con carga fija con regulador de corriente síncrono en el control vectorial indirecto.
- Figura 2.17 Gráficas de las trayectorias de las corrientes estacionarias dqs en el regulador de corriente síncrono. Con variación de la velocidad de referencia con carga fija.
- Figura 3.1 Extensión del esquema de control orientado del flujo del rotor indirecto para operación en campo debilitado.
- Figura 3.2 Desintonización en estado estable del efecto de los parámetros para las corrientes comando nominales de flujo y de par: (a) motores de alta potencia, (b) motores de baja potencia.

- Figura 3.3 Principio de adaptación de la T_r basado en el esquema adaptable al modelo de referencia.
- Figura 3.4 Relación entre el valor de F/F^{mot} y x .
- Figura 3.5 Configuración del bloque de estimación de la resistencia del rotor.
- Figura 3.6 Configuración de la estimación de la resistencia del rotor basado en las funciones F_{est} , F_{act} y en un controlador difuso.
- Figura 3.7 Control vectorial indirecto del motor de inducción y configuración del estimador de la resistencia del rotor basado en las funciones F_{est} , F_{act} y en un controlador difuso.
- Figura 3.8 Arquitectura de una red neuronal de realimentación.
- Figura 3.9 Procesamiento en una neurona.
- Figura 3.10 Diagrama general del patrón de entradas y salidas del estimador neuronal de la resistencia del rotor.
- Figura 3.11 Fase de entrenamiento del estimador neuronal de la resistencia del rotor.
- Figura 3.12 Control vectorial indirecto del motor de inducción con el estimador neuronal de la resistencia del rotor.
-
- Figura. 4.1 Diagrama de bloques del programa en SIMULINK utilizado para la simulación del control vectorial indirecto del motor de inducción con PWM y regulador de corriente síncrono, en el cual se realiza la estimación y actualización de la resistencia del rotor usando un estimador basado en las funciones F_{est} , F_{act} y en un controlador difuso.
- Figura 4.2 Error entre la resistencia del rotor de referencia y la resistencia del rotor que proporciona la red neuronal artificial.
- Figura 4.3 Configuración de la red neuronal artificial para la estimación de la resistencia del rotor.
- Figura 4.4 Diagrama de bloques del programa en SIMULINK utilizado para la simulación del control vectorial indirecto del motor de inducción con PWM y regulador de corriente síncrono, en el cual se realiza la estimación y adaptación de la resistencia del rotor por medio de una red neuronal artificial.
- Figura 4.5 Efecto de la variación de la resistencia del rotor sin realizar la estimación y la actualización de su valor en el algoritmo de control.
- Figura 4.6 Estimación de la resistencia del rotor en estado estable con el estimador basado en un controlador difuso y el comportamiento de las variables del control vectorial indirecto del motor de inducción. *Primer caso estable.*
- Figura 4.7 Estimación de la resistencia del rotor en estado estable con el estimador neuronal y el comportamiento de las variables del control vectorial indirecto del motor de inducción. *Primer caso estable.*

- Figura 4.8 Estimación de la resistencia del rotor en estado estable con el estimador basado en un controlador difuso y comportamiento de las variables del control vectorial indirecto del motor de inducción. Se realizan cambios de paso en la resistencia del rotor. *Segundo caso estable.*
- Figura 4.9 Estimación de la resistencia del rotor en estado estable con el estimador neuronal y comportamiento de las variables del control vectorial indirecto del motor de inducción. Se realizan cambios de paso en la resistencia del rotor. *Segundo caso estable.*
- Figura 4.10 Estimación de la resistencia del rotor en estado transitorio con el estimador basado en un controlador difuso y comportamiento de las variables del control vectorial indirecto del motor de inducción. *Primer caso transitorio.*
- Figura 4.11 Estimación de la resistencia del rotor en estado transitorio con el estimador neuronal y comportamiento de las variables del control vectorial indirecto del motor de inducción. *Primer caso transitorio.*
- Figura 4.12 Estimación de la resistencia del rotor en estado transitorio con el estimador basado en un controlador difuso y comportamiento de las variables del control vectorial indirecto del motor de inducción. *Segundo caso transitorio.*
- Figura 4.13 Estimación de la resistencia del rotor en estado transitorio con el estimador neuronal y comportamiento de las variables del control vectorial indirecto del motor de inducción. *Segundo caso transitorio.*
- Figura 4.14 Estimación de la resistencia del rotor en estado transitorio con el estimador basado en un controlador difuso y comportamiento de las variables del control vectorial indirecto del motor de inducción. *Tercer caso transitorio.*
- Figura 4.15 Estimación de la resistencia del rotor en estado transitorio con el estimador neuronal y comportamiento de las variables del control vectorial indirecto del motor de inducción. *Tercer caso transitorio.*
- Figura 4.16 Estimación de la resistencia del rotor en estado transitorio con el estimador basado en un controlador difuso y comportamiento de las variables del control vectorial indirecto del motor de inducción. *Cuarto caso transitorio.*
- Figura 4.17 Estimación de la resistencia del rotor en estado transitorio con el estimador neuronal y comportamiento de las variables del control vectorial indirecto del motor de inducción. *Cuarto caso transitorio.*
- Figura 4.18 Estimación de la resistencia del rotor con el estimador neuronal bajo diferentes condiciones de operación en el motor de inducción de 100 HP.
- Figura 4.19 Estimación de la resistencia del rotor con el estimador neuronal bajo diferentes condiciones de operación en el motor de inducción de 1 HP.
- Figura 4.20 Estimación de la resistencia del rotor con el estimador neuronal bajo diferentes condiciones de operación.

- Figura B.1 Curva del par contra la velocidad para un accionamiento del motor de CD.
- Figura B.2 Diagrama de bloques del campo debilitado usado en el debilitamiento del flujo del rotor.
- Figura C.1. Neurona Artificial.
- Figura C.2. De la neurona biológica a la neurona artificial.
- Figura C.3. Proceso de una red neuronal.
- Figura C.4. Red de tres capas.
- Figura C.5. Notación compacta de una red de tres capas.
- Figura C.6. Disposición de una red sencilla de 3 capas.
- Figura C.7. Superficie típica de error
- Figura C.8. Trayectoria del algoritmo Levenberg-Marquardt

LISTADO DE TABLAS

- Tabla 2.1 Consecuencia de la variable K_p .
- Tabla 2.2 Consecuencia de la variable τ_I .
- Tabla 2.3 Valores para el ψ'_{dr} y sus correspondientes valores de ω_{rm} para el campo debilitado.
- Tabla 3.1 Algoritmos de adaptación de T_r .
- Tabla 3.2 Base de reglas para la estimación de la resistencia del rotor.
- Tabla 4.1 Generación de las entradas de la red neuronal artificial.
- Tabla 4.2 Vector de entradas.
- Tabla 4.3 Vector de salidas.
- Tabla 4.4 Creación de una red con 1 capa oculta.
- Tabla 4.5 Entrenamiento de la red neuronal artificial.
- Tabla 4.6 Parámetros para realizar el entrenamiento.
- Tabla 4.7 Creación de una red con 2 capas ocultas.
- Tabla 4.8 Características de la red con 2 capas ocultas.
- Tabla 4.9 Creación de una red con 2 capas ocultas.
- Tabla 4.10 Características de la red con 1 capa oculta.
- Tabla 4.11 Obtención de la red neuronal en Simulink después del entrenamiento.
- Tabla 4.12 Resultados obtenidos con la red neuronal artificial entrenada.
- Tabla 4.13 Valores de la simulación.

Capítulo 1

DESCRIPCIÓN Y OBJETIVOS DEL TRABAJO DE TESIS

En este capítulo se presentan los objetivos generales, se expone el estado del arte, la justificación, se mencionan las aportaciones del trabajo y su contenido.

1.1 Objetivos generales

- Mejorar el desempeño del control vectorial de la velocidad del motor de inducción con el método indirecto en condiciones dinámicas, mediante la utilización de un estimador de la resistencia del rotor que actualice su valor en el algoritmo de control.

1.2 Justificación

La red neuronal artificial y el controlador de lógica difusa se utilizan en el control vectorial y en la estimación de parámetros de los motores de inducción. Los resultados de simulaciones de trabajos anteriores [1]-[2] muestran que estas técnicas pueden usarse como alternativas inteligentes a los controladores convencionales del control vectorial, pudiéndose lograr respuestas más rápidas y con menos oscilaciones.

En este trabajo se desarrolla el método de control vectorial indirecto, en donde se utilizan controladores PI autosintonizados con lógica difusa, para el control de la velocidad. Este algoritmo de autosintonización difuso para el controlador PI reduce el sobretiro y el tiempo de establecimiento comparado con cualquier otro algoritmo de sintonización para el controlador PI. Asimismo, se realiza la regulación de la corriente por medio de un regulador vectorial síncrono. También se presenta un nuevo esquema para la estimación de la resistencia del rotor basado en una red neuronal artificial. Este esquema requiere de las mediciones del voltaje bifásico síncrono y de la corriente bifásica síncrona cuyas transformaciones dependen de los voltajes y corrientes que se toman de las terminales del motor de inducción; también se utilizan las señales de referencia, tanto del flujo del rotor, y de la corriente síncrona en el eje d , así como de la frecuencia del estator.

1.3 Estado del arte

Las máquinas eléctricas son imprescindibles en cualquier sistema industrial. Su evolución durante el último siglo ha sido lenta en comparación con el rápido avance de los dispositivos semiconductores de potencia y de los circuitos convertidores. El advenimiento de computadoras digitales modernas, el mejoramiento del modelado, la simulación y la disponibilidad de nuevos materiales ha contribuido al incremento de los niveles de potencia, de una eficiencia más alta, mejor fiabilidad, costo reducido, y al mejoramiento del diseño mecánico y térmico de la máquina eléctrica. Aunque se han usado las máquinas de corriente directa (CD) tradicionalmente para aplicaciones de velocidad variable [41], el uso de los inversores de frecuencia variable de estado sólido, de los dispositivos semiconductores de potencia mejorados, de los microprocesadores, de DSP y de las técnicas de estimación y control avanzadas han reforzado la aplicación de los accionamientos de corriente alterna (CA) de frecuencia variable [41].

El control dinámico de alto desempeño de los motores de inducción puede llevarse a cabo usando el método de control vectorial, donde el motor de inducción es controlado de forma desacoplada, lográndose un comportamiento similar al del motor de CD con excitación separada. El desacoplamiento de las componentes de flujo y de par de las corrientes del estator en el control vectorial se realiza a partir de la determinación del vector unitario, el cual es necesario para la transformación de coordenadas.

Los métodos de control vectorial de campo orientado directo e indirecto del motor de inducción permiten el control independiente del par y del flujo por el desacoplamiento de la corriente del estator en dos componentes ortogonales: una componente de par (i_{qs}) y una componente de flujo (i_{ds}) [3]. El primer método necesita información sobre la posición instantánea del flujo que puede obtenerse de los sensores de flujo (bobinas de búsqueda, sensores de efecto hall), o por algún esquema de estimación de flujo (componentes de saturación de terceros armónicos [4], redes neuronales artificiales [5], etc.). Considerando que el método indirecto es basado en la dinámica inversa del modelo del flujo del motor, en donde hay tres posibles aplicaciones básicas, las cuales están basadas en la orientación del flujo del estator, del entrehierro o del rotor [6]. Entre estos métodos, el control indirecto del flujo del rotor es el más usado debido a su simplicidad de implementación [16]. Sin embargo, este esquema de control se basa en el modelo del motor y es más sensible a las variaciones de los parámetros. El desacoplamiento sólo se mantiene si los parámetros reales del motor de inducción permanecen iguales a aquellos programados en el controlador. Si hay alguna desviación, el desacoplamiento se pierde y la dinámica de ejecución se deteriora drásticamente. La resistencia del rotor tiene un rol dominante en la desintonización del regulador. Este parámetro varía debido a la temperatura y a la frecuencia del rotor que a su vez produce un acoplamiento cruzado entre el par y el flujo, y por consiguiente, degrada la actuación del accionamiento. Debido a lo anterior, es necesario estimar la resistencia del rotor real del motor de inducción, y eliminar el problema que surge al considerar un valor erróneo de la resistencia del rotor. Los métodos para la estimación de la constante de tiempo del rotor o de la resistencia del rotor son actualmente investigados y se han

propuesto varias soluciones, algunos de estos métodos dependen de la resistencia del estator, requieren de mucho tiempo de procesamiento para converger, o solamente son convenientes para una aplicación en específico [7]-[9].

Desafortunadamente, la constante de tiempo del rotor también depende del nivel del flujo del motor que directamente afecta el valor de la inductancia de magnetización. La variación de esta inductancia puede llegar a perturbar a la máquina, si esta es operada en la región del campo debilitado. Cuando el motor es operado en la región del par constante el flujo se mantiene constante y la inductancia de magnetización que corresponde al nivel de saturación es considerada constante y no es afectada por las diversas condiciones de operación del motor. La constante de tiempo del rotor también es sensible a la saturación de la inductancia de fuga la cual varía con la carga del motor. En la referencia [7] se obtiene un factor de desorientación que es una función de las variables finales del motor y constituye una medida de la desviación de la constante de tiempo del rotor de su valor nominal. Este esquema usa los parámetros del motor (resistencia del estator r_s , resistencia del rotor r_r' y la inductancia mutua L_M) para calcular este factor y un controlador PI lleva a cero la diferencia entre el valor fijo y el valor estimado de la constante de tiempo del rotor. Considerando que la referencia [8] trata con un ángulo de salida del flujo del rotor del *eje-d*, este ángulo se evalúa usando el voltaje del estator y la corriente y depende de r_s , L_M y r_r' . Como la resistencia del estator se usa en la estimación, el esquema no podría usarse a bajas velocidades a menos que esta resistencia se identifique con precisión en este rango. La referencia [9] presenta diferentes esquemas de estimación que usan funciones de referencia y explica sus respectivas ventajas y desventajas. En el proceso de estimación, se calcula una función de referencia usando las variables comando del motor que aseguran la condición de orientación del flujo (F_{est}). Esta función es nuevamente evaluada usando las variables reales del motor (F_{act}). La desigualdad entre F_{est} y F_{act} constituye una condición de desorientación y debe ponerse en cero. Esta referencia propone al *eje-d* del algoritmo del Modelo de Referencia Adaptable del Sistema (MRAS) como el más interesante para la adaptación en línea de la constante de tiempo del rotor cuando es usado junto con un algoritmo exacto para la estimación de r_s [12].

Otros tipos de identificadores de parámetros usados en el problema de la estimación de la constante de tiempo del rotor, se encuentran: el Filtro de Kalman Extendido (EKF) [10], el Observador de Luenberger Extendido (ELO) [11]. El EKF es un estimador estocástico, mientras el ELO es un estimador determinístico. Estos métodos son computacionalmente demandantes, ya que ambos estimadores usan un modelo de la máquina de inducción de cuarto orden. El MRAS [12] ha mostrado una convergencia precisa, pero es más lenta que el EKF o ELO. El método de los Mínimos Cuadrados Recursivos (RLS) [13], proporciona una estimación pobre de la resistencia del rotor con errores del 20% al 30% en el valor estimado, comparado con las posibles variaciones térmicas en la resistencia del rotor por arriba del 50%.

Las redes neuronales artificiales han tenido gran interés en el campo de la electrónica de potencia. Esto se debe a su paralelismo inherente que permite alta velocidad de procesamiento y permite la implementación de aplicaciones de control en tiempo real. Las redes neuronales artificiales también poseen la habilidad de conservar su desempeño en ambientes con ruido y son tolerantes a las fallas y a la pérdida de datos.

En este trabajo se realiza el control vectorial indirecto por medio de controladores PI autosintonizados con lógica difusa, asimismo se realiza la estimación y adaptación de la resistencia del rotor por medio de una red neuronal artificial. La condición de aprendizaje es realizada por el algoritmo de entrenamiento de Levenberg-Marquart que es proporcionado por MATLAB Neural Networks Toolbox.

1.4 Aportaciones

- Se desarrollo un estimador de la resistencia del rotor basado en una red neuronal artificial, el cual permite estimar este parámetro y cuyo valor se actualiza en las ecuaciones del control vectorial indirecto.
- El desempeño dinámico del estimador neuronal propuesto supera al desempeño de un estimador basado en la función de referencia, la función estimada y en un controlador difuso el cual fue tomado de la literatura.
- Los resultados del trabajo fueron presentados en el congreso internacional: The 40th International Universities Power Engineering Conference (UPEC-2005), en Cork, Irlanda.

1.5 Contenido del trabajo

En este trabajo, se propone un nuevo método para la estimación de la resistencia del rotor basado en una red neuronal artificial. La red se entrena para estimar este parámetro a partir de un patrón de referencia el cual esta basado en un modelo matemático del sistema. Una vez entrenada, la red realiza el procesamiento de la estimación de la resistencia del rotor.

Este trabajo se organiza como sigue, en el Capítulo 1 se da una introducción general del trabajo a desarrollar, en el Capítulo 2 se modela el motor de inducción en lazo abierto el cual se alimenta con una fuente trifásica senoidal, asimismo se desarrolla un algoritmo de autosintonización difuso para el controlador PI y se realiza la regulación de corriente para el control vectorial indirecto. En el Capítulo 3 se desarrolla la metodología para entrenar a una red neuronal artificial con la finalidad de estimar la resistencia del rotor y se presenta el principio de adaptación de la resistencia del rotor al control vectorial indirecto. Los resultados de la simulación del método propuesto se muestran en el Capítulo 4, y por último se exponen las conclusiones y recomendaciones en el Capítulo 5.

Capítulo 2

CONTROL VECTORIAL INDIRECTO

2.1 Introducción

La máquina de inducción comparada con la de corriente directa tiene varias ventajas en lo referente al tamaño, simplicidad, robustez, costo, etc., aunque tiene un comportamiento eminentemente no lineal con alto grado de interacción entre sus variables, siendo estas condiciones las que complican el estudio de su dinámica.

Los avances alcanzados en la teoría del control de las máquinas de inducción, vinculados con el control vectorial y los progresos tecnológicos alcanzados en el campo de los semiconductores de potencia, permiten lograr avances sustanciales en la potencia, integración y costos necesarios para el diseño, difusión y comercialización de estos accionamientos de potencia.

En general se usan dos esquemas básicos en el control de campo orientado [14]. El esquema de orientación directo del campo, donde la posición del flujo del rotor es directamente medida por medio de bobinas de búsqueda o se estima a partir de las mediciones de corrientes y voltajes en las terminales del estator, usándose esta señal en el control para desacoplar las componentes de par y de flujo de las corrientes del estator. El segundo esquema es la orientación indirecta del campo magnético del rotor, donde la velocidad de deslizamiento se emplea para estimar la posición del flujo magnético del rotor.

En ambos métodos existe una etapa muy importante entre el control y el motor, que es el inversor trifásico y por consiguiente la técnica de regulación de corriente para el modulador del ancho del pulso (PWM). Por consiguiente, el desempeño del sistema del inversor depende de la calidad de la estrategia de control de corriente aplicada [15].

En este capítulo se describe el modelo del motor de inducción y se realiza su simulación, posteriormente se establecen los principios del control vectorial indirecto y se desarrolla un algoritmo de autosintonización difuso para el controlador PI para la regulación de la velocidad mecánica del rotor, asimismo, se realiza la regulación de la corriente para el control vectorial indirecto. Finalmente, se realiza la simulación de este esquema de control bajo diferentes condiciones de operación.

2.2 Modelo y simulación del motor de inducción

2.2.1 Modelo del motor de inducción

Las ecuaciones del modelo de la máquina de inducción de jaula de ardilla, trifásica, de P polos y en el marco de referencia estacionario $qd0$ tomadas de la referencia [16], pueden organizarse en la siguiente forma para la simulación:

$$\begin{aligned} v_{ag} &= V_m \cos(\omega t) \\ v_{bg} &= V_m \cos\left(\omega t - \frac{2}{3}\pi\right) \\ v_{cg} &= V_m \cos\left(\omega t + \frac{2}{3}\pi\right) \end{aligned} \quad (2.1)$$

$$\begin{aligned} v_{qs}^s &= \frac{2}{3}v_{as} - \frac{1}{3}v_{bs} - \frac{1}{3}v_{cs} = \frac{2}{3}v_{ag} - \frac{1}{3}v_{bg} - \frac{1}{3}v_{cg} - v_{sg} \\ v_{ds}^s &= \frac{1}{\sqrt{3}}(v_{cs} - v_{bs}) = \frac{1}{\sqrt{3}}(v_{cg} - v_{bg}) \\ v_{0s} &= \frac{1}{3}(v_{as} + v_{bs} + v_{cs}) = \frac{1}{3}(v_{ag} + v_{bg} + v_{cg}) - v_{sg} \end{aligned} \quad (2.2)$$

$$\begin{aligned} v_{qr}' &= \frac{2}{3}v_{ar}' - \frac{1}{3}v_{br}' - \frac{1}{3}v_{cr}' = \frac{2}{3}v_{an}' - \frac{1}{3}v_{bn}' - \frac{1}{3}v_{cn}' - v_{rm}' \\ v_{dr}' &= \frac{1}{\sqrt{3}}(v_{cr}' - v_{br}') = \frac{1}{\sqrt{3}}(v_{cn}' - v_{bn}') \\ v_{0r}' &= \frac{1}{3}(v_{ar}' + v_{br}' + v_{cr}') = \frac{1}{3}(v_{an}' + v_{bn}' + v_{cn}') - v_{rm}' \end{aligned} \quad (2.3)$$

$$\begin{aligned} v_{qr}'^s &= v_{qr}' \cos\theta_r(t) + v_{dr}' \text{sen}\theta_r(t) \\ v_{dr}'^s &= -v_{qr}' \text{sen}\theta_r(t) + v_{dr}' \cos\theta_r(t) \\ \theta_r(t) &= \int_0^t \omega_r(t) dt + \theta_r(0) \end{aligned} \quad (2.4)$$

$$\begin{aligned} i_{as} &= i_{qs}^s + i_{0s} \\ i_{bs} &= -\frac{1}{2}i_{qs}^s - \frac{\sqrt{3}}{2}i_{ds}^s + i_{0s} \\ i_{cs} &= -\frac{1}{2}i_{qs}^s + \frac{\sqrt{3}}{2}i_{ds}^s + i_{0s} \end{aligned} \quad (2.5)$$

$$\begin{aligned} i_{qr}'^r &= i_{qr}'^s \cos\theta_r(t) - i_{dr}'^s \text{sen}\theta_r(t) \\ i_{dr}'^r &= i_{qr}'^s \text{sen}\theta_r(t) + i_{dr}'^s \cos\theta_r(t) \end{aligned} \quad (2.6)$$

$$\begin{aligned}
i'_{ar} &= i'_{qr} + i'_{0r} \\
i'_{br} &= -\frac{1}{2}i'_{qr} - \frac{\sqrt{3}}{2}i'_{dr} + i'_{0r} \\
i'_{cr} &= -\frac{1}{2}i'_{qr} + \frac{\sqrt{3}}{2}i'_{dr} + i'_{0r}
\end{aligned} \tag{2.7}$$

$$\begin{aligned}
\psi_{qs}^s &= \omega_b \int \left\{ v_{qs}^s + \frac{r_s}{X_{ls}} (\psi_{mq}^s - \psi_{qs}^s) \right\} dt \\
\psi_{ds}^s &= \omega_b \int \left\{ v_{ds}^s + \frac{r_s}{X_{ls}} (\psi_{md}^s - \psi_{ds}^s) \right\} dt \\
i_{0s} &= \frac{\omega_b}{X_{ls}} \int (v_{0s} - i_{0s} r_s) dt
\end{aligned} \tag{2.8}$$

$$\begin{aligned}
\psi_{qr}'s &= \omega_b \int \left\{ v_{qr}'s + \frac{\omega_r}{\omega_b} \psi_{dr}'s + \frac{r'_r}{X'_{lr}} (\psi_{mq}^s - \psi_{qr}'s) \right\} dt \\
\psi_{dr}'s &= \omega_b \int \left\{ v_{dr}'s - \frac{\omega_r}{\omega_b} \psi_{qr}'s + \frac{r'_r}{X'_{lr}} (\psi_{md}^s - \psi_{dr}'s) \right\} dt \\
i'_{0r} &= \frac{\omega_b}{X'_{lr}} \int (v'_{0r} - i'_{0r} r'_r) dt
\end{aligned} \tag{2.9}$$

$$\begin{aligned}
\psi_{qs}^s &= X_{ls} i_{qs}^s + \psi_{mq}^s & i_{qs}^s &= \frac{\psi_{qs}^s - \psi_{mq}^s}{X_{ls}} \\
\psi_{ds}^s &= X_{ls} i_{ds}^s + \psi_{md}^s & i_{ds}^s &= \frac{\psi_{ds}^s - \psi_{md}^s}{X_{ls}} \\
\psi_{qr}'s &= X'_{lr} i_{qr}'s + \psi_{mq}^s & i_{qr}'s &= \frac{\psi_{qr}'s - \psi_{mq}^s}{X'_{lr}} \\
\psi_{dr}'s &= X'_{lr} i_{dr}'s + \psi_{md}^s & i_{dr}'s &= \frac{\psi_{dr}'s - \psi_{md}^s}{X'_{lr}}
\end{aligned} \tag{2.10}$$

$$\begin{aligned}
\psi_{mq}^s &= X_M \left(\frac{\psi_{qs}^s}{X_{ls}} + \frac{\psi_{qr}'s}{X'_{lr}} \right) \\
\psi_{md}^s &= X_M \left(\frac{\psi_{ds}^s}{X_{ls}} + \frac{\psi_{dr}'s}{X'_{lr}} \right)
\end{aligned} \tag{2.11}$$

La ecuación del par electromagnético es:

$$T_{em} = \frac{3}{2} \frac{P}{2\omega_b} (\psi_{ds}^s i_{qs}^s - \psi_{qs}^s i_{ds}^s) \tag{2.12}$$

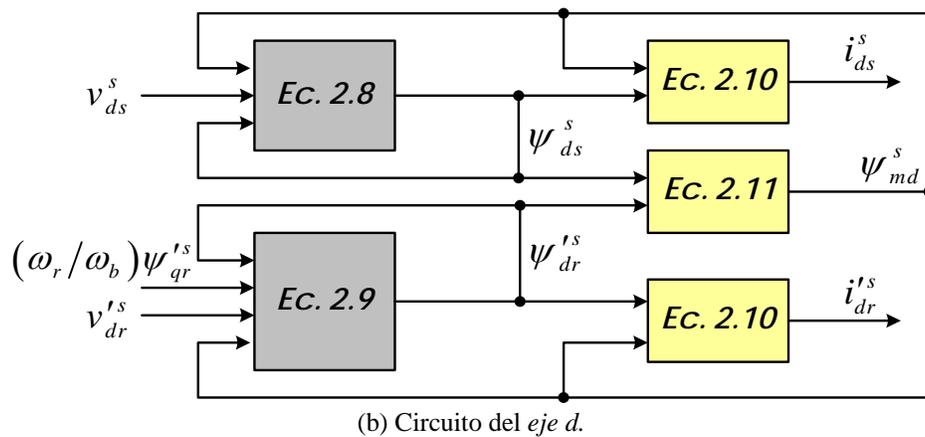
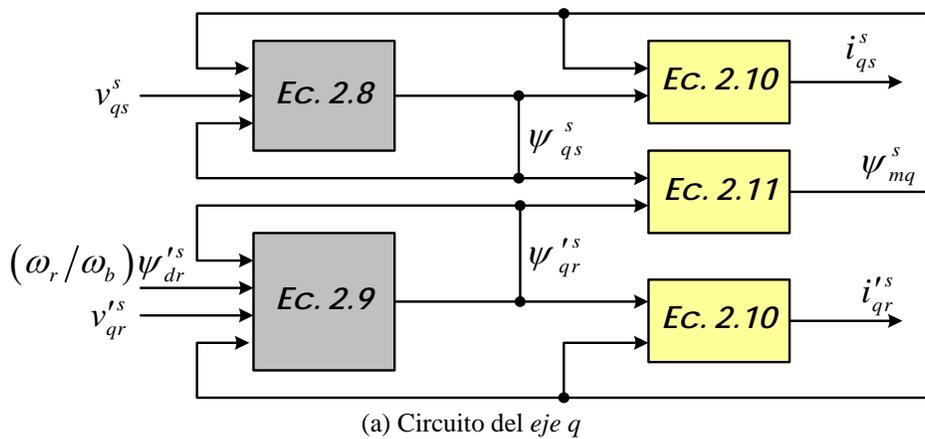
La segunda ley de Newton aplicada al eje del motor es:

$$\frac{2J\omega_b}{P} \frac{d(\omega_r / \omega_b)}{dt} = T_{em} + T_{mec} - T_{amort} \quad (2.13)$$

La ecuación anterior de movimiento se escribe en términos de la constante de inercia, H , definida como la relación entre la energía cinética de la masa rotatoria y la velocidad base a la potencia nominal, que es:

$$H = \frac{J\omega_{bm}^2}{2S_N} \quad (2.14)$$

En la Fig. 2.1 se muestra el diagrama de flujo de las variables en la simulación de los circuitos en los ejes q y d de la máquina de inducción trifásica. Observe las condiciones de la velocidad-voltaje con acoplamiento cruz entre las partes de la simulación del circuito del eje q y del eje d . Las componentes de secuencia cero, sin embargo, no se acoplan a los circuitos qd .



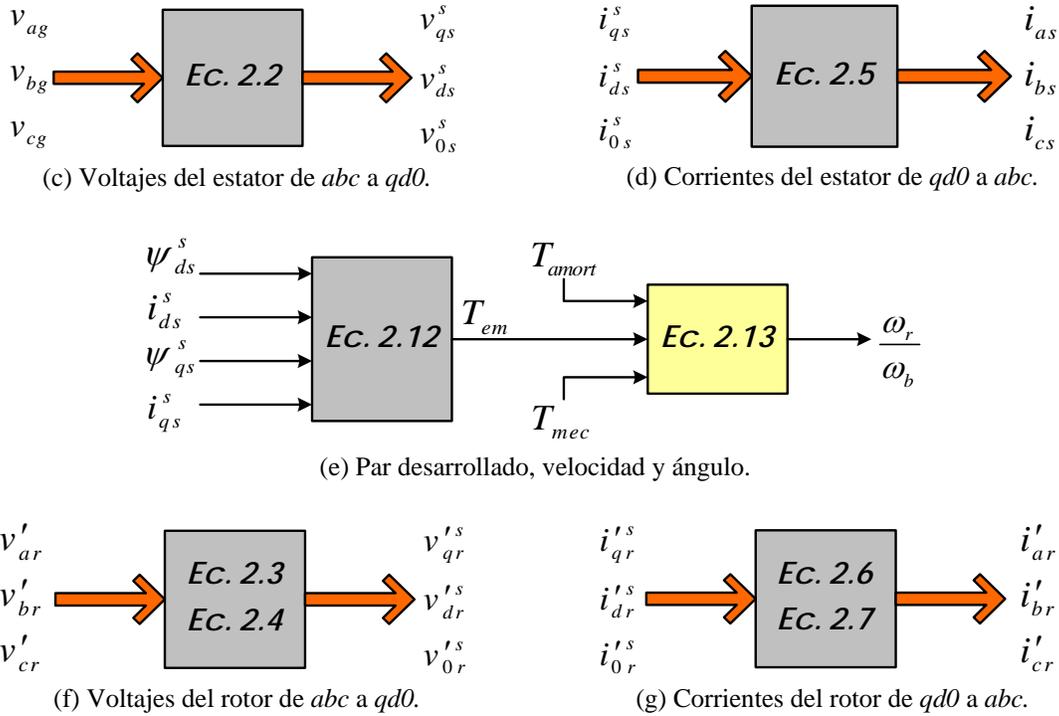
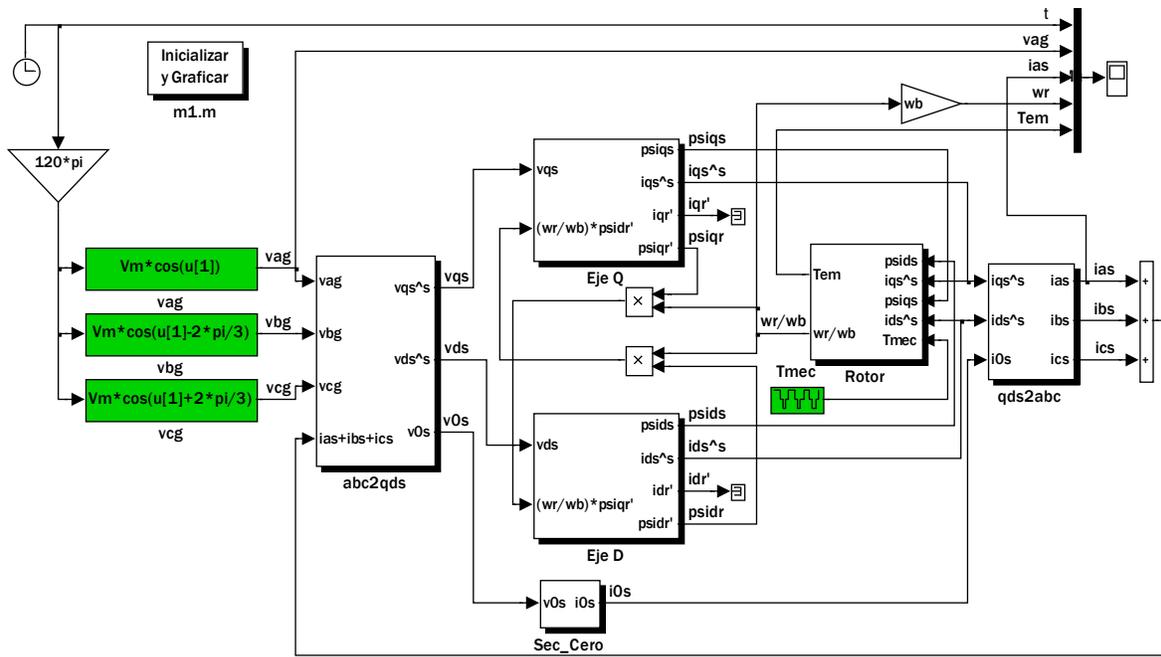


Figura 2.1. Flujo de variables en la simulación de los circuitos en los ejes q y d .

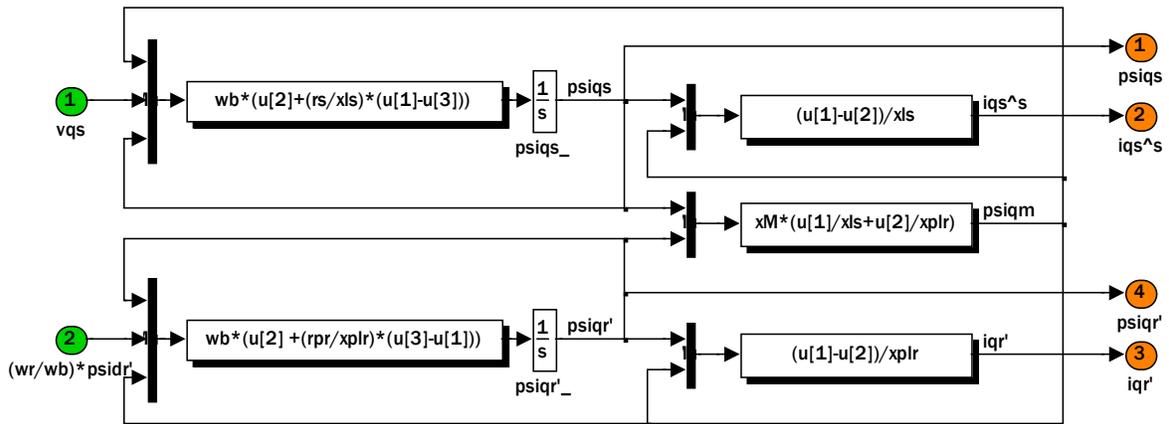
2.2.2 Simulación del motor de inducción

En este punto se realiza la simulación de la máquina de inducción en el marco de referencia estacionario $qd0$. Posteriormente, se usa la simulación para examinar la característica motora y generadora, de la máquina de inducción alimentada con voltajes senoidales.

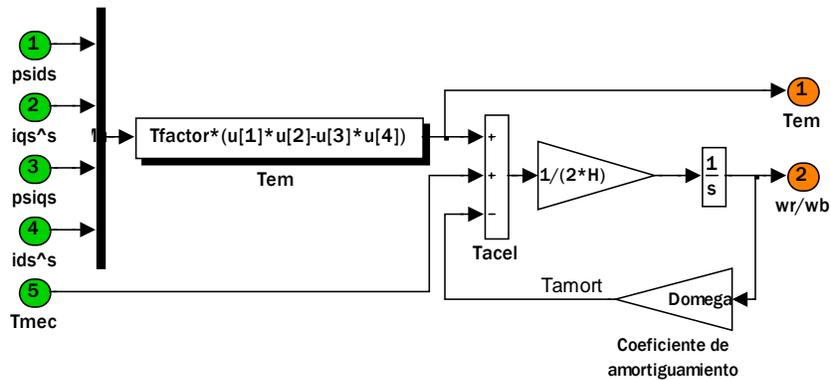
La Figura 2.2(a) muestra un diagrama general del programa de simulación en SIMULINK de un motor de inducción en el marco de referencia estacionario. Se proporcionan los detalles dentro de los bloques principales de las Figs. 2.2(b) a la 2.2(e), con la excepción del bloque del *eje d* que por la simetría, es similar al bloque del *eje q*. Como no existe el hilo neutro en la alimentación del motor, entonces las corrientes de secuencia cero no fluyen y no es necesario considerar el bloque de secuencia cero. Sin embargo, el voltaje entre el punto de referencia de los devanados del estator, s , y el punto de referencia del sistema, g , se obtiene conectando un pequeño capacitor ficticio, C_{sg} , entre estos puntos.



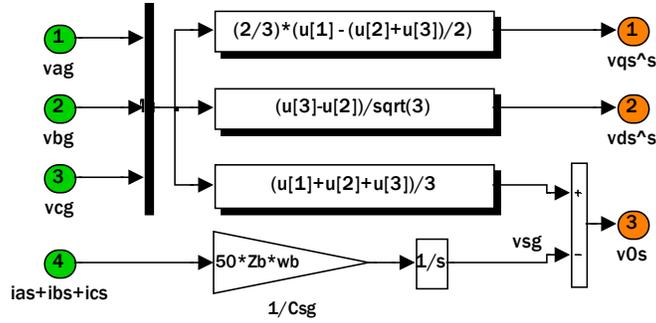
(a) Diagrama global de la simulación del motor de inducción.



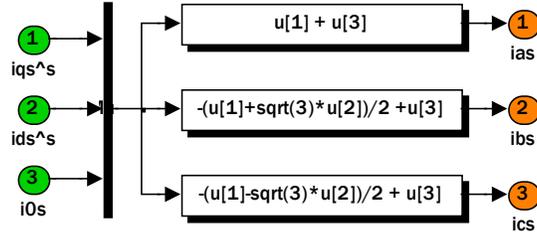
(b) Bloque del circuito del *eje q*.



(c) Bloque del circuito del *rotor*.



(d) Bloque de la transformación de *abc2qds* (trifásica-bifásica estacionaria).



(e) Bloque de la transformación de *qds2abc* (bifásica estacionaria-trifásica).

Figura 2.2 Simulación del motor de inducción en el marco de referencia estacionario.

Para realizar la simulación del motor de inducción de 20 HP, trifásico, 60 Hz, 4 polos, 220 V en SIMULINK, se declaran los parámetros en el archivo *m1.m* de MATLAB (ver Apéndice A), los cuales son:

$$\begin{aligned}
 r_s &= 0.1062 \, \Omega & X_{ls} &= 0.2145 \, \Omega \\
 r_r' &= 0.0764 \, \Omega & X_{lr}' &= 0.2145 \, \Omega \\
 X_m &= 5.834 \, \Omega & J &= 2.8 \, \text{kgm}^2
 \end{aligned}$$

La velocidad nominal de la máquina es de 1748 RPM a un deslizamiento de 0.0287, y la corriente nominal es de 46.68 A.

Cuando el valor de C_{sg} se mantiene en $1/(50Z_b\omega_b)$, la simulación se ejecuta con un paso mínimo de 2×10^{-4} s, un paso máximo de 1×10^{-2} s, y una tolerancia de 1×10^{-7} s usando el método de ode15s. Cuando el neutro del estator se conecta con tierra, C_{sg} es infinito y la ganancia en el bloque de $1/C_{sg}$ debe establecerse en cero.

Usando un tiempo de ejecución de 2 s., se simula el arranque del motor con voltaje nominal y sin carga mecánica. En la Fig. 2.3 se muestran las curvas correspondientes a la operación del motor bajo condiciones de estado estable.

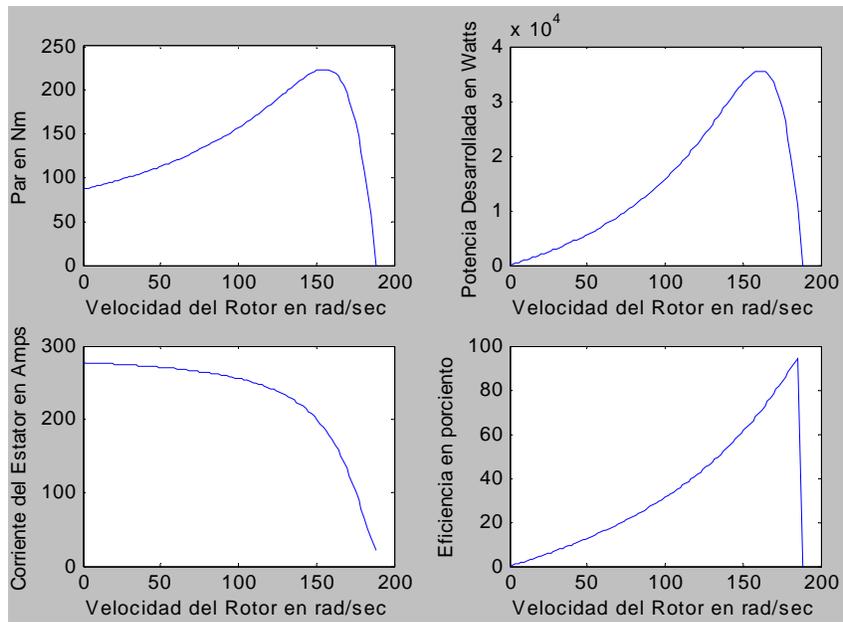


Figura 2.3 Curvas de estado estable de la máquina de inducción.

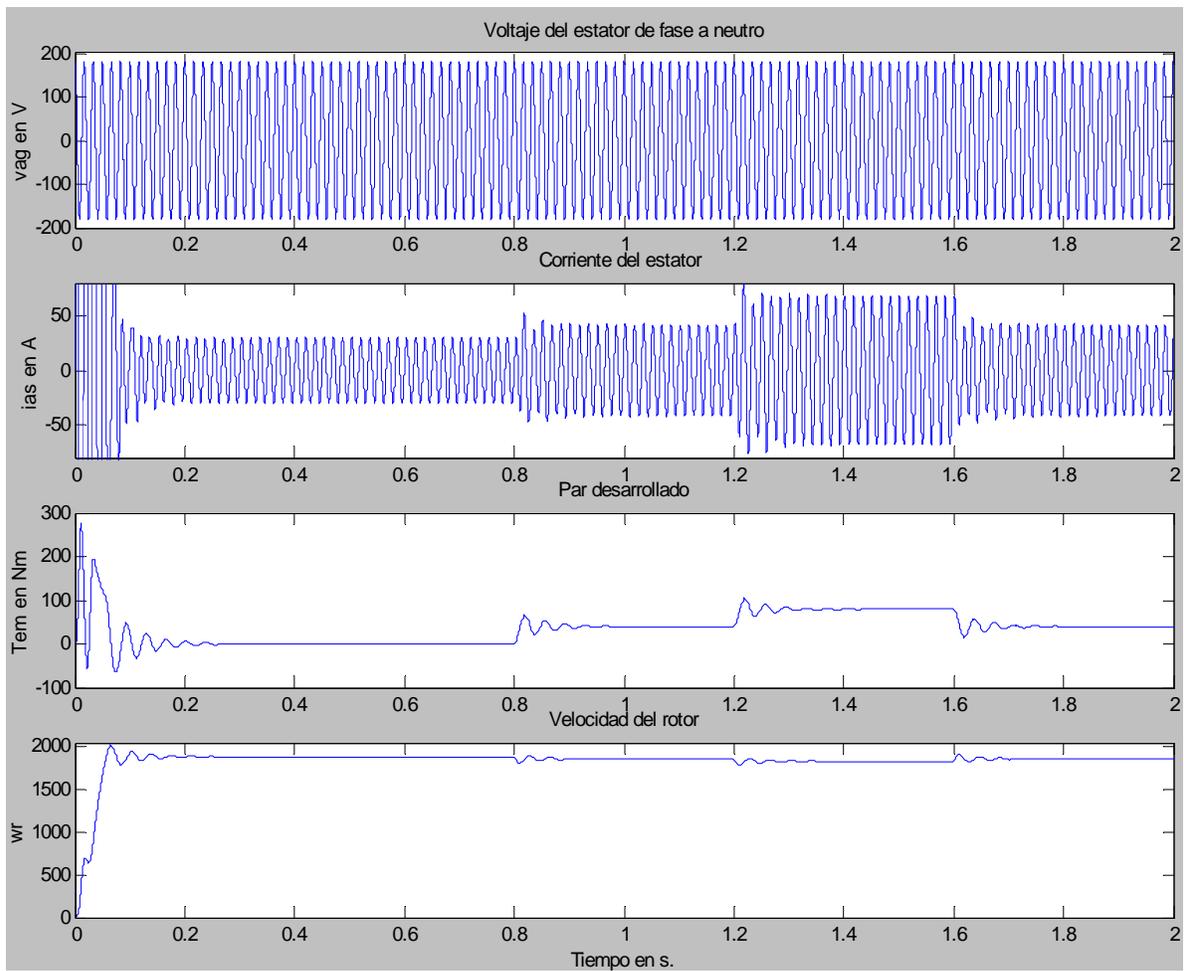


Figura 2.4. Arranque del motor de inducción sin carga y con cambios de carga.

En la simulación del rotor que se muestra en la Fig. 2.2(c), la máquina actúa como motor cuando T_{mec} es negativo, y como generador cuando T_{mec} es positivo. En dos incrementos se cambia el par de carga desde cero al valor base de T_b , es decir, la potencia nominal, S_N entre la velocidad mecánica base, ω_{bm} . Se grafican las respuestas transitorias de i_{as} , T_{mec} , y ω_r de acuerdo a los cambios que se aplicaron en el par de carga. En la Fig. 2.4 se muestra el arranque de la máquina sin carga y con cambios en el par de carga, T_{mec} , de cero al 50%, de 50% al 100%, y de 100% al 50% del par nominal aplicado en $t = 0.8, 1.2,$ y 1.6 s, respectivamente.

2.3 Control vectorial indirecto del motor de inducción

Los esquemas de control de campo orientado para la máquina de inducción son del tipo *directo* cuando el ángulo, ρ , es determinado con la medición directa del flujo del entrehierro como se muestra en la Fig. 2.5, y del tipo *indirecto* cuando el ángulo del rotor se determina de las medidas estimadas, como es la velocidad de deslizamiento.

El esquema del control vectorial indirecto se toma de la referencia [16], en donde el ángulo espacial del flujo del rotor se obtiene de la suma del ángulo del rotor (θ_r) y del ángulo de deslizamiento de referencia (θ_2), donde el ángulo de deslizamiento proporciona la posición del enlace de flujo del rotor relativo al eje directo del marco de referencia estacionario referido al rotor.

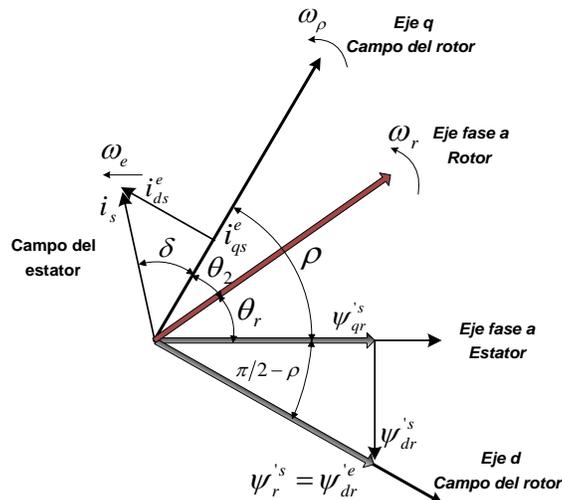


Figura 2.5. Orientación adecuada en el marco de referencia síncrono.

En el control vectorial indirecto se realizan las acciones de control siguientes:

- 1) Regulación de i'_{qs} para el control del par desarrollado (se emplea un controlador PI autosintonizado con lógica difusa, que se describe en 2.4) y de la velocidad de deslizamiento, $\omega_e - \omega_r$.
- 2) Regulación de i'_{ds} para el control del flujo del rotor.

Si se proporcionan algunos niveles deseados del flujo del rotor, ψ_r^{e*} , el valor deseado de i_{ds}^{e*} se obtiene con la ecuación siguiente [16]:

$$i_{ds}^{e*} = \psi_{dr}^{e*} \frac{r_r' + L_r' p}{r_r' L_m} \quad (2.15)$$

A partir del par de referencia, se puede calcular el valor de la corriente de referencia i_{qs}^{e*} de acuerdo con la ecuación siguiente [16]:

$$i_{qs}^{e*} = \frac{2}{3} \frac{2}{P} \frac{L_r'}{L_m} \frac{T_{em}^*}{\psi_{dr}^{e*}} \quad (2.16)$$

Cuando el flujo se orienta apropiadamente, i_{dr}^e es cero y el $\psi_{dr}^{e*} = L_m i_{ds}^e$; de esta manera, la velocidad de deslizamiento también pueden escribirse como [16]:

$$\omega_2^* = \omega_e - \omega_r = \frac{r_r' i_{qs}^{e*}}{L_r' i_{ds}^{e*}} \quad (2.17)$$

Las condiciones anteriores, aseguran el desacoplamiento de las ecuaciones de voltaje del rotor [16]. El desacoplamiento depende de la exactitud de los parámetros del motor, principalmente de la resistencia del rotor y de la inductancia de magnetización, siendo la resistencia del rotor la que se estimará en este trabajo.

La Fig. 2.6 muestra el esquema del control vectorial indirecto para el accionamiento del motor de inducción con un regulador de corriente, el cual utiliza un inversor PWM senoidal fuente de voltaje. El ángulo del vector espacial flujo del rotor, ρ , es la suma del ángulo de posición del rotor, θ_r , y del ángulo, θ_2 , obtenida de la integración de la velocidad de deslizamiento. Si las salidas ortogonales de la forma $\cos \theta_r$ y $\sin \theta_r$ son proporcionadas por el encoder, los valores de $\cos \rho$ y $\sin \rho$ pueden generarse de las siguientes identidades trigonométricas [16]:

$$\begin{aligned} \cos \rho &= \cos(\theta_r + \theta_2) = \cos \theta_r \cos \theta_2 - \sin \theta_r \sin \theta_2 \\ \sin \rho &= \sin(\theta_r + \theta_2) = \sin \theta_r \cos \theta_2 + \cos \theta_r \sin \theta_2 \end{aligned} \quad (2.18)$$

Las salidas del bloque del control vectorial indirecto son las corrientes bifásicas síncronas, i_{ds}^{e*} , i_{qs}^{e*} , cuyos valores son de referencia. Para realizar la transformación de las coordenadas síncronas qd a las coordenadas trifásicas estacionarias abc , se realizan a su vez las siguientes transformaciones [16]:

$$\begin{aligned} i_{qs}^{s*} &= i_{qs}^{e*} \cos \rho + i_{ds}^{e*} \sin \rho \\ i_{ds}^{s*} &= -i_{qs}^{e*} \sin \rho + i_{ds}^{e*} \cos \rho \end{aligned} \quad (2.19)$$

$$\begin{aligned}
 i_{as}^* &= i_{qs}^{s*} \\
 i_{bs}^* &= -\frac{1}{2}i_{qs}^{s*} - \frac{\sqrt{3}}{2}i_{ds}^{s*} \\
 i_{cs}^* &= -\frac{1}{2}i_{qs}^{s*} + \frac{\sqrt{3}}{2}i_{ds}^{s*}
 \end{aligned}
 \tag{2.20}$$

Las corrientes en coordenadas síncronas qd del estator, son determinadas transformando las corrientes trifásicas medidas a las corrientes estacionarias qd , usando la siguiente ecuación [16]:

$$\begin{aligned}
 i_{qs}^s &= \frac{2}{3}i_{as} - \frac{1}{3}i_{bs} - \frac{1}{3}i_{cs} \\
 i_{ds}^s &= \frac{1}{\sqrt{3}}(i_{cs} - i_{bs})
 \end{aligned}
 \tag{2.21}$$

Posteriormente, se usa el valor del ángulo de orientación del campo, ρ , que es el ángulo del vector espacial al flujo del rotor respecto a los ejes de coordenadas, para obtener la siguiente transformación [16]:

$$\begin{aligned}
 i_{qs}^e &= i_{qs}^s \cos \rho - i_{ds}^s \sin \rho \\
 i_{ds}^e &= i_{qs}^s \sin \rho + i_{ds}^s \cos \rho
 \end{aligned}
 \tag{2.22}$$

Las ecuaciones (2.21) y (2.22) se usan en el bloque del regulador de corriente de la Fig. 2.6.

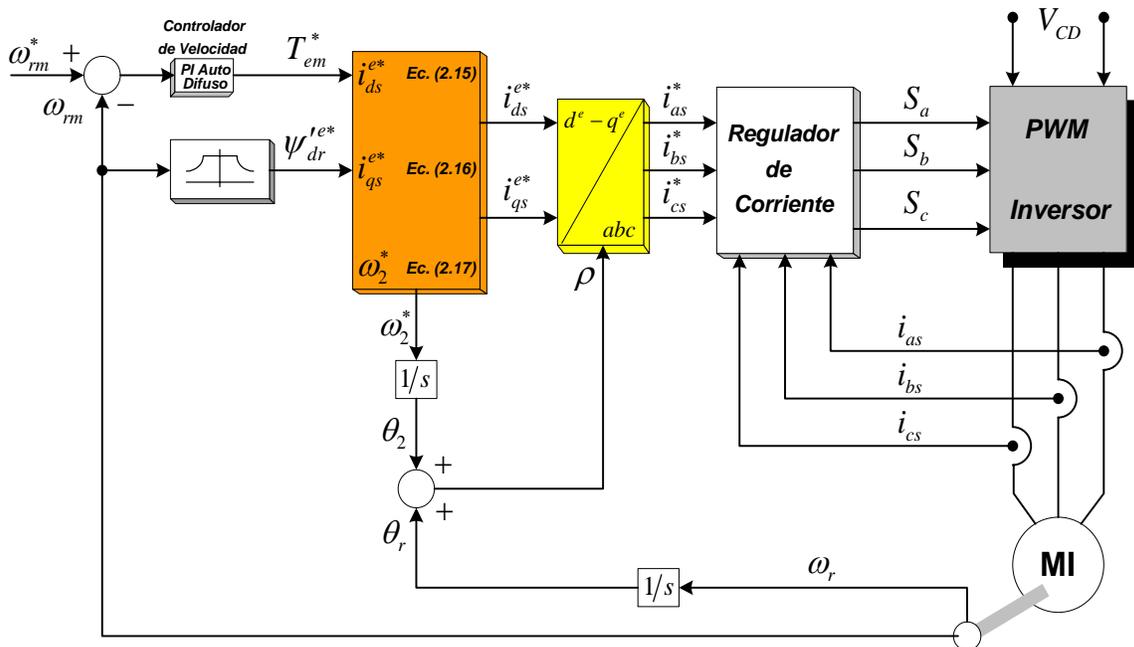


Figura 2.6. Control vectorial indirecto del motor de inducción alimentado con un inversor de voltaje con corrientes reguladas.

Las señales S_a , S_b y S_c de la Fig. 2.6 se obtienen cuando se utiliza al regulador de corriente de histéresis [15], las cuales se mandan al inversor. Para el caso de los reguladores estacionario y síncrono [15] se deben considerar los voltajes de referencia V_a^* , V_b^* y V_c^* en lugar de las señales S_a , S_b y S_c , de esta manera los voltajes obtenidos son las señales de entrada del PWM de donde se obtienen las señales moduladas para el inversor. En la sección 2.5 se describe al regulador de corriente síncrono, que será utilizado en las simulaciones posteriores.

2.4 Regulador vectorial síncrono

La función principal del esquema de control de corriente en un convertidor con inversor PWM (Fig. 2.7) es forzar a las corrientes en una carga trifásica de CA a que sigan a las señales de referencia [15]. Lo cual se realiza comparando los valores instantáneos de las corrientes de fase de referencia i_{as}^* , i_{bs}^* , i_{cs}^* con las corrientes de fase medidas i_{as} , i_{bs} , i_{cs} , el regulador de corriente genera los estados de conmutación S_a , S_b y S_c para los dispositivos de potencia del inversor los cuales tienden a disminuir los errores de las corrientes ε_a , ε_b , ε_c . En general, el regulador de corriente realiza dos tareas [15]: la *compensación* del error (decrecimiento de ε_a , ε_b , ε_c) y la *modulación* (determinación de los estados de conmutación S_a , S_b y S_c).

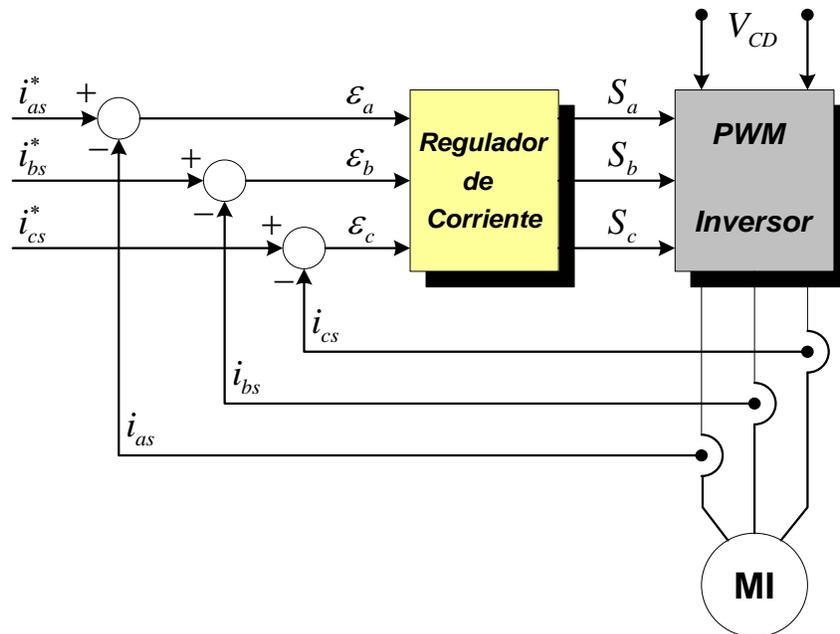


Figura 2.7. Esquema funcional básico del regulador de corriente para el inversor PWM.

Los reguladores lineales operan con inversores de voltaje PWM [26]-[28]. En contraste con los reguladores no lineales, los esquemas con reguladores lineales han separado claramente la compensación del error de corriente y las partes de modulación de

voltaje. Este concepto nos permite aprovechar las ventajas de los moduladores en lazo abierto (PWM senoidal, modulador de vectores espaciales), las cuales son: frecuencia de conmutación constante y señal modulada. También, se tiene un diseño independiente de la estructura general de control, asimismo pueden realizarse pruebas fácilmente en la carga y en el lazo abierto del inversor.

La referencia [31] proporciona un estudio a profundidad del regulador síncrono y realiza una comparación detallada con el regulador estacionario. Asimismo, en [33] se resaltan las ventajas del regulador síncrono con respecto al estacionario. Por lo tanto, en este trabajo se presenta un *regulador síncrono* que trabaja en *coordenadas estacionarias* i_{ds}^s e i_{qs}^s , el cual es idéntico al regulador en el marco síncrono. Como se muestra en la Fig. 2.8, el lazo interno del sistema de control es un generador de frecuencia variable, que consiste de dos integradores y multiplicadores, en donde se obtienen los voltajes de referencia v_{ds}^{*s} , v_{qs}^{*s} para el modulador PWM, incluso cuando en estado estable las señales del error de la corriente son cero.

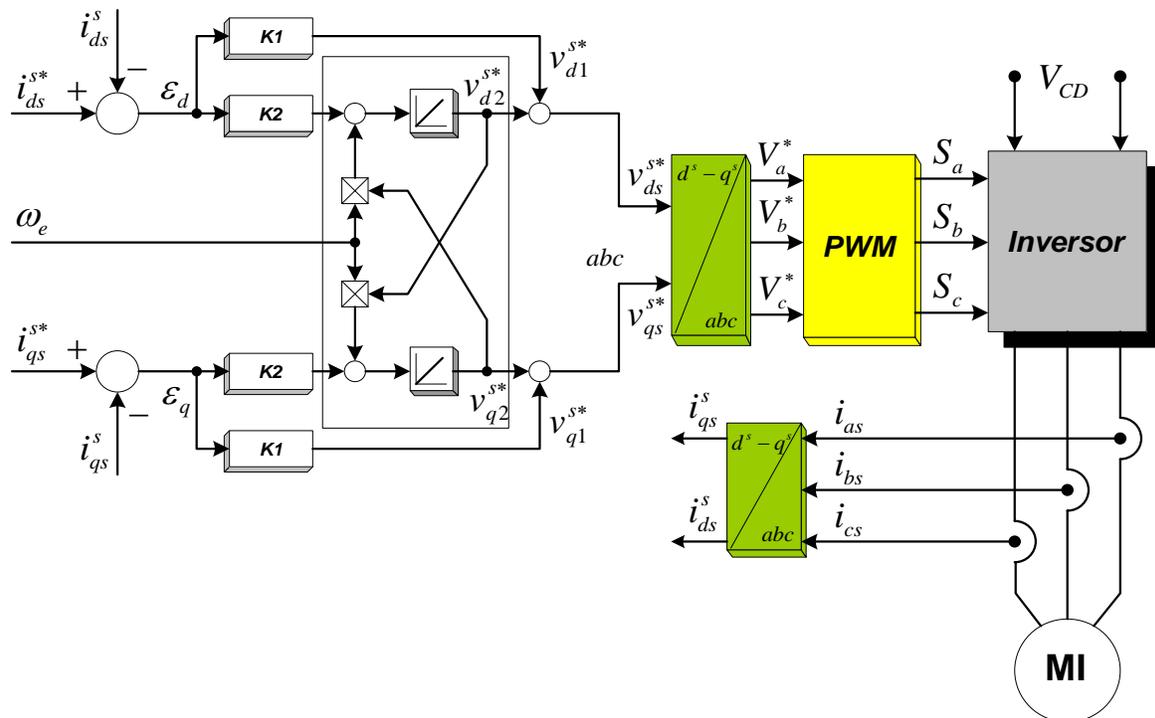


Figura 2.8. Regulador de corriente lineal síncrono que trabaja en coordenadas estacionarias con componentes de CA.

2.5 Controlador PI autosintonizado con lógica difusa

El uso de controladores clásicos (P, PI) calculados matemáticamente a partir de un modelo aproximado de la planta, puede presentar serias deficiencias en su comportamiento cuando el sistema a controlar está sometido a perturbaciones externas, o cuando el proceso de encontrar un modelo adecuado de la dinámica del sistema es excesivamente complejo.

Este tipo de situaciones se ha venido afrontando en los últimos tiempos con técnicas avanzadas de control, entre las cuales se pueden nombrar el control adaptable por modelo de referencia [45], los esquemas basados en el criterio de estabilidad de Lyapunov [17], y los controladores de estructura variable [20]. Por otro lado, las técnicas de inteligencia artificial como la utilización de mecanismos de inferencia basados en lógica difusa [22], también han adquirido gran importancia en las aplicaciones de control.

En el control vectorial de un motor de inducción, el controlador PI se usa en la implementación de los lazos de control de la velocidad, del flujo magnético y de las componentes de corriente del estator [19]. Sin embargo, para lograr un mayor desempeño, los parámetros deben reajustarse según el cambio del momento de inercia y de la carga [18]. Los sistemas autosintonizados se han aplicado con el objetivo de superar los problemas mencionados anteriormente, sin embargo, el desarrollo del sistema de autosintonización para el control del motor de inducción es relativamente reciente [21].

El algoritmo de autosintonización para el controlador PI utiliza los valores iniciales, obtenidos del método de Ziegler-Nichols que utiliza la última ganancia y el último período de un sistema sintonizado, y se calcula la relación del error y el cambio del error de la respuesta de la planta que corresponde al valor inicial de los parámetros, por lo tanto se obtiene la nueva ganancia proporcional (K_p) y el tiempo integral (τ_I) del sintonizador difuso. Este algoritmo de autosintonización difuso para un controlador PI reduce considerablemente el sobretiro y el tiempo de establecimiento comparado con cualquier otro algoritmo de sintonización [18]. Por lo tanto, en el presente trabajo se aplica un algoritmo de autosintonización para el controlador PI por la teoría de la lógica difusa para mejorar su desempeño, el cual se toma de la referencia [18].

2.5.1 Acción de control del controlador PI

La acción de control se expresa de la siguiente forma:

$$u = K_p \left[\varepsilon(t) + \frac{1}{\tau_I} \int_0^t \varepsilon(t) \right] \quad (2.23)$$

donde $\varepsilon(t)$, u , K_p , τ_I son la señal de entrada (error), la señal de salida (acción de control), la ganancia proporcional, y la constante de tiempo integral, respectivamente.

De la ecuación anterior, el error se determina con la señal de referencia, $y_r(t)$, y la señal de salida de la planta, $y(t)$, de la siguiente forma:

$$\varepsilon(t) = y_r(t) - y(t) \quad (2.24)$$

Para el método de autosintonización difuso, primero se requiere calcular los parámetros K_p y τ_I de las fórmulas de sintonización de Ziegler-Nichols [18], usándose estos parámetros como valores iniciales. Por lo tanto, el método de autosintonización difuso utiliza la relación del error y del cambio del error para producir la función de membresía y las reglas difusas.

2.5.2 Autosintonización difusa

La estructura del autosintonizador difuso se muestra en la Fig. 2.9, que es aplicada para determinar dos componentes de la Ec. (2.23), que son la ganancia proporcional, K_p , y la constante de tiempo integral, τ_I , de acuerdo al punto de operación, el cual está definido por el error, y el cambio de error. Esto resulta en el uso de cuatro variables lingüísticas, dos para las entradas y dos para las salidas.

De lo anterior, la relación del cambio del error se calcula a partir del error de salida por la siguiente ecuación:

$$\Delta \equiv \Delta \varepsilon(k) = \varepsilon(k) - \varepsilon(k-1) \quad (2.25)$$

Por medio del autosintonizador difuso y por la expresión de la Ec. (2.26) se determinan $K_p(k)$ y $\tau_I(k)$ a través del razonamiento del cambio de la ganancia proporcional $K_p(k)$ y del cambio del tiempo integral $\tau_I(k)$ de la salida de la planta.

$$\begin{aligned} \nabla_p \equiv K_p(k) &= K_p(k-1) + \Delta K_p(k) \\ \nabla_I \equiv \tau_I(k) &= \tau_I(k-1) + \Delta \tau_I(k) \end{aligned} \quad (2.26)$$

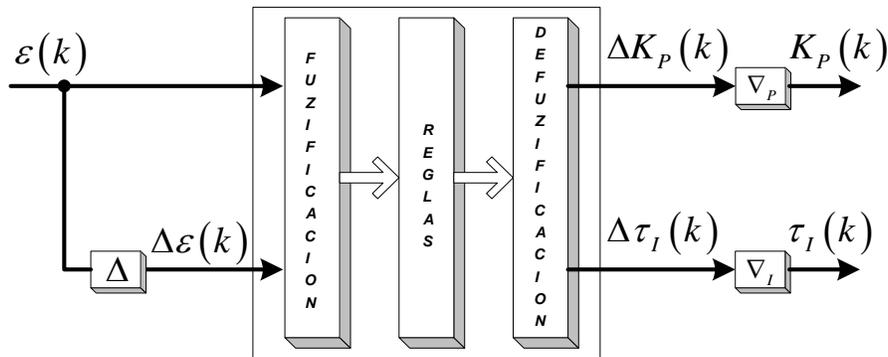


Figura 2.9. Estructura general del autosintonizador difuso.

2.5.3 Reglas difusas

Las reglas difusas que describen el mecanismo de inferencia consisten de dos grupos de reglas de control difuso, que sirven para determinar $\Delta K_p(k)$ y $\Delta \tau_I(k)$, las cuales tienen la siguiente forma:

1. La regla para $\Delta K_p(k)$:

$$R^i : \text{SI } \varepsilon(k) \text{ ES } A_j \text{ Y } \Delta \varepsilon(k) \text{ ES } B_j \text{ ENTONCES } \Delta K_p(k) \text{ ES } C_{j,i} \quad (2.27)$$

2. La regla para $\Delta\tau_l(k)$:

$$R^m : \text{SI } \varepsilon(k) \text{ ES } A_j \text{ Y } \Delta\varepsilon(k) \text{ ES } B_j \text{ ENTONCES } \Delta\tau_l(k) \text{ ES } D_{j,l} \quad (2.28)$$

donde $i, m (1, \dots, 9)$ es el número de reglas, $j = 1, 2, 3$, y $l = 1, 2, 3$. Asimismo, A y B son los valores lingüísticos de las variables lingüísticas del *Error*, $\varepsilon(k)$, y del *Cambio de Error*, $\Delta\varepsilon(k)$; así como, C y D son los valores lingüísticos de las variables lingüísticas de la *Ganancia Proporcional*, $K_p(k)$ y del *Tiempo Integral* $\tau_l(k)$.

2.5.4 Función de membresía de la premisa

Las funciones de membresía son del tipo triangular, tanto para la premisa del error, $\varepsilon(k)$, como para el cambio del error, $\Delta\varepsilon(k)$, las cuales se muestran en la Fig. 2.10(a) y (b). Asimismo, las funciones de membresía para la ganancia proporcional, K_p , y el tiempo integral, τ_l , también son del tipo triangular y se muestran en la Fig. 2.10(c) y (d).

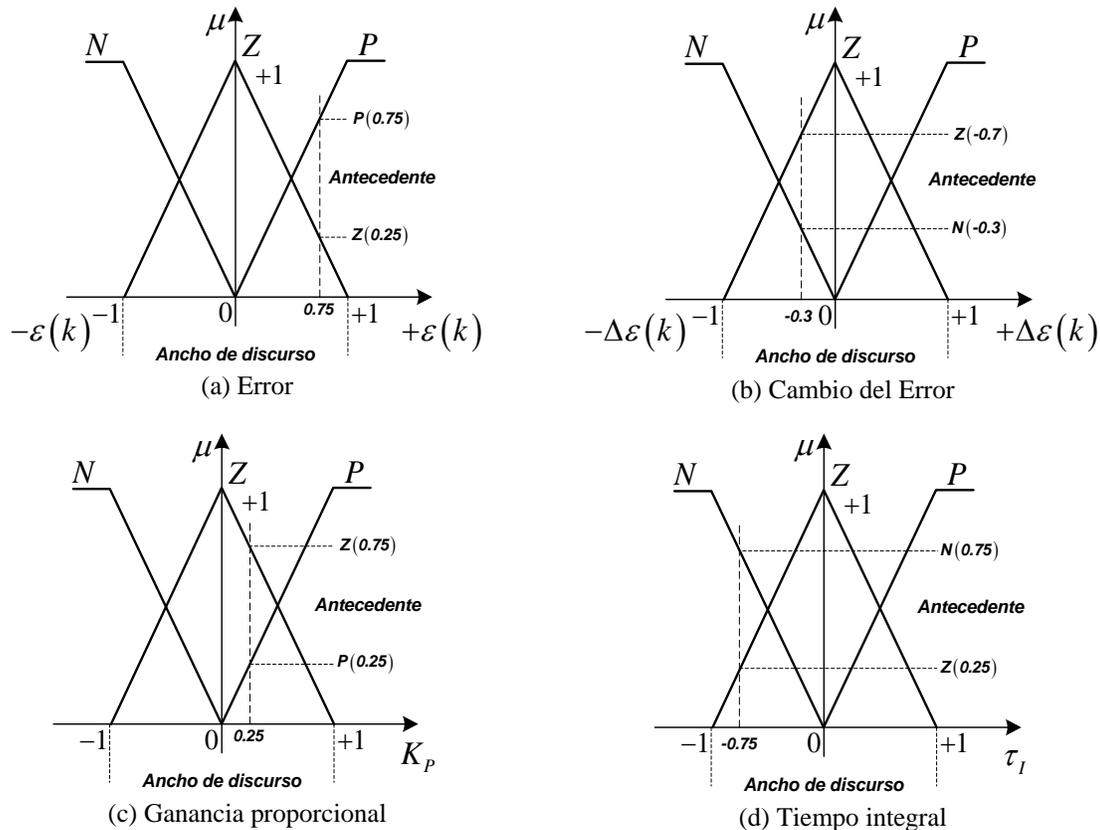


Figura 2.10. Funciones de membresía.

Los conjuntos difusos para las variables lingüísticas de entrada han sido definidos como sigue (Figs. 2.10(a) y 2.10(b)): N , negativo; Z , cero; P , positivo. Similarmente, para cada uno de los dos universos de salida los tres conjuntos difusos primarios de salida están definidos como (Figs. 2.10(c) y 2.10(d)): N , negativo; Z , cero; P , positivo. Todos los conjuntos difusos son triangulares. La forma de las funciones de membresía, los conjuntos difusos, y sus ubicaciones se seleccionan arbitrariamente.

2.5.5 Parámetro de la consecuencia

El parámetro de la consecuencia, definido por la variable lingüística difusa a partir del índice de desempeño, muestra que el incremento de la ganancia proporcional K_p produce un sobretiro grande, mientras que el incremento del tiempo integral τ_I produce una disminución del tiempo de establecimiento.

Las reglas de control de la consecuencia de K_p y τ_I se definen a partir de la curva del error de la respuesta de una planta que se muestra en la Fig. 2.11.

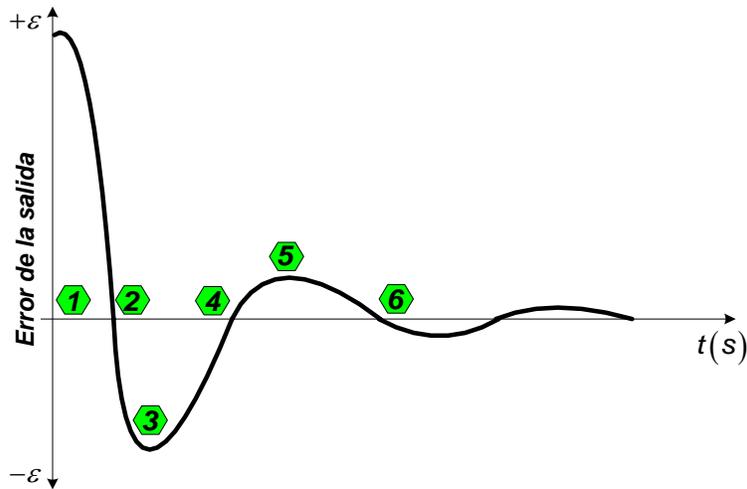


Figura 2.11. Curva del error de la salida.

Las reglas de control de la consecuencia de K_p son:

Zona	Variación de la consecuencia K_p	Acción
①–②	Incrementar la $K_p(k)$	Incrementar la salida
②–③	Disminuir la $K_p(k)$	Disminuir la salida
③–④	Disminuir la $K_p(k)$	Disminuir la salida
④–⑤	Incrementar la $K_p(k)$	Incrementar la salida

Por lo tanto, las reglas de control de K_p , se muestran en la Tabla 2.1.

$\Delta\epsilon$	\mathcal{N}	\mathcal{Z}	\mathcal{P}
ϵ	\mathcal{N}	\mathcal{Z}	\mathcal{P}
\mathcal{N}	N	P	N
\mathcal{Z}	P	Z	N
\mathcal{P}	P	N	P

Tabla 2.1. Consecuencia de la variable K_p

Las reglas de control de la consecuencia de τ_I son:

Zona	Variación de la consecuencia τ_I	Acción
①–②	Disminuir el $\tau_I(k)$	incrementar la salida
②–③	Incrementar el $\tau_I(k)$	disminuir la salida
③–④	Incrementar el $\tau_I(k)$	disminuir la salida
④–⑤	Disminuir el $\tau_I(k)$	incrementar la salida

Asimismo, las reglas de control de τ_I , se muestran en la Tabla 2.2.

$\Delta\varepsilon$	\mathcal{N}	\mathcal{Z}	\mathcal{P}
ε			
\mathcal{N}	P	N	P
\mathcal{Z}	N	Z	P
\mathcal{P}	N	P	N

Tabla 2.2. Consecuencia de la variable τ_I

2.5.6 Razonamiento

El razonamiento del $\Delta K_p(k)$, se describe por la siguiente ecuación:

$$\Delta K_p(k) = \frac{\sum_{i=1}^I \alpha_i (\Delta K_p(k))_i}{\sum_{i=1}^I \alpha_i} \quad (2.29)$$

donde:

$$\alpha_i = \mu_{A_i}(\varepsilon) \wedge \mu_{B_i}(\Delta\varepsilon) \quad \text{Operación difusa de intersección} \quad (2.29(a))$$

$$(\Delta K_p(k))_i = \text{COG}(C_{ji}) \quad \text{COG es el centro de gravedad} \quad (2.29(b))$$

$$I (i = 1 \dots 9) \quad \text{Número de reglas de } \Delta K_p(k) \quad (2.29(c))$$

De igual manera, el razonamiento del $\Delta\tau_I(k)$, se describe por la siguiente ecuación:

$$\Delta \tau_I(k) = \frac{\sum_{m=1}^M \alpha_m (\Delta \tau_I(k))_m}{\sum_{m=1}^M \alpha_m} \quad (2.30)$$

donde:

$$\alpha_m = \mu_{A_m}(\varepsilon) \wedge \mu_{B_m}(\Delta \varepsilon) \quad \text{Operación difusa de intersección} \quad (2.30(a))$$

$$(\Delta \tau_I(k))_m = \text{COG}(D_{jl}) \quad \text{COG es el centro de gravedad} \quad (2.30(b))$$

$$M (m = 1 \dots 9) \quad \text{Número de reglas de } \Delta \tau_I(k) \quad (2.30(c))$$

Los valores reales de la salida se obtienen usando el método del centro de gravedad (COG) [23].

2.6 Simulación del control vectorial indirecto con regulador vectorial síncrono

En este apartado se muestra la simulación en SIMULINK, del control vectorial indirecto del motor de inducción utilizando un inversor de voltaje PWM con corrientes reguladas. Los objetivos de la simulación son examinar la forma en que el control vectorial indirecto mantiene el flujo del rotor constante durante los cambios en el par de carga, así como observar la respuesta dinámica con el método de control vectorial indirecto con el regulador de corriente síncrono. Para realizar una comparación con el funcionamiento del motor en lazo abierto, se usa la misma máquina de inducción de 20 HP, 220 V, 4 polos del apartado 2.2.2.

A partir del esquema de la Fig. 2.6, en la Fig. 2.12(a) se muestra el diagrama general de la simulación en SIMULINK del control vectorial indirecto del motor de inducción con inversor de voltaje PWM con el regulador de corriente síncrono.

En la Fig. 2.12(b) se muestra el regulador síncrono, que se basa en la Fig. 2.8, en donde se observa el lazo interno del sistema de control, el cual es un generador de frecuencia variable, y que genera los voltajes de referencia v_{ds}^{s*} y v_{qs}^{s*} . Este tipo de regulador requiere de la velocidad angular del estator, ω_e . Por lo tanto, para cada componente, se convierte el error de la corriente a un voltaje de referencia, v_{ds}^{s*} y v_{qs}^{s*} . Este voltaje bifásico estacionario se transforma a un voltaje trifásico de referencia (Fig. 2.12(c)), obteniéndose las señales moduladas por medio del generador PWM (ver Fig. 2.12(d)), las cuales son las señales de control que activan a los dispositivos del inversor fuente de voltaje (Fig. 2.12(e)), generándose de esta forma el voltaje trifásico para alimentar al motor de inducción, realizándose posteriormente la transformación del voltaje trifásico al voltaje bifásico estacionario (Fig. 2.12(f)).

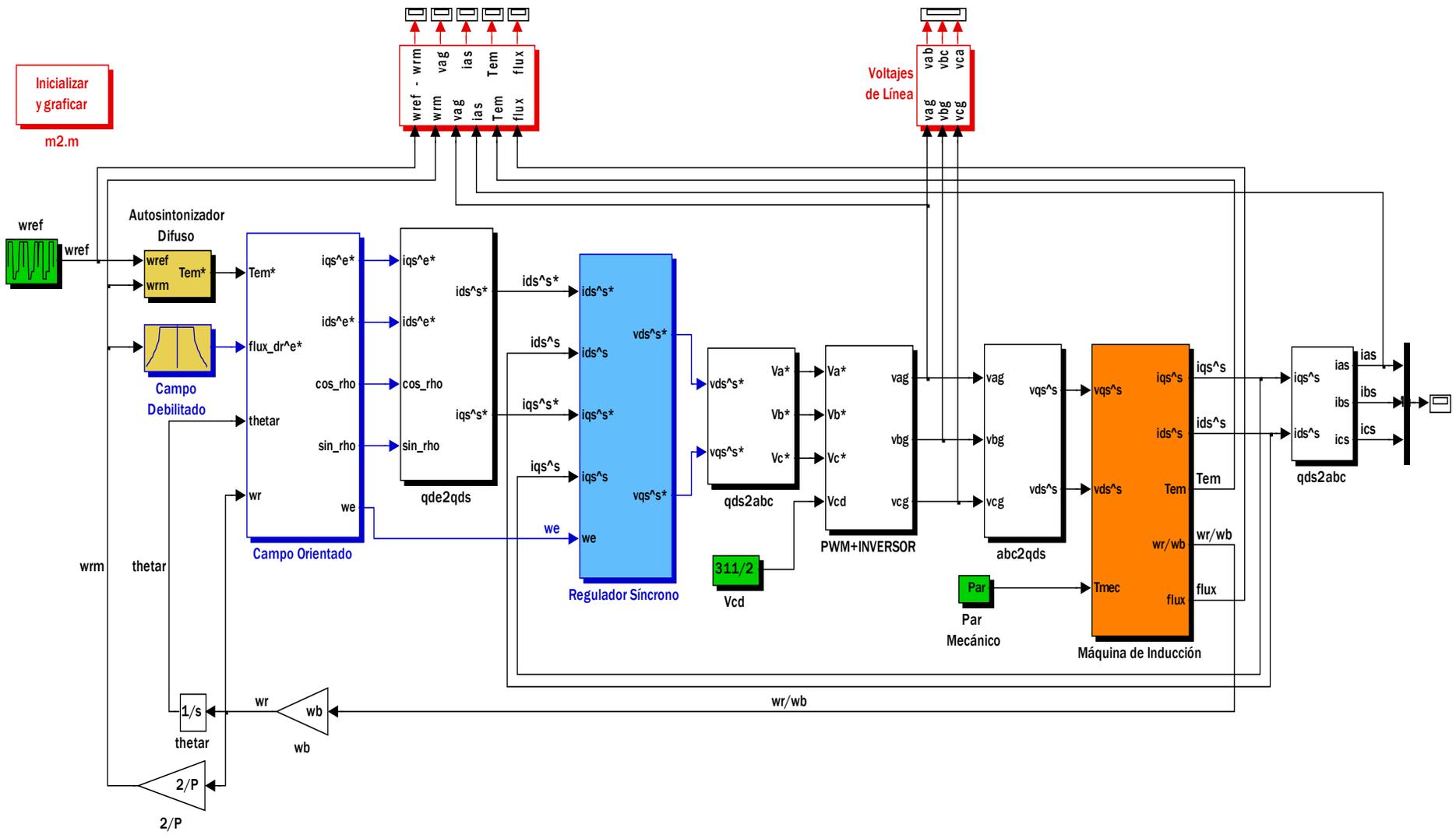
En la Fig. Fig. 2.12(g), se muestra el autosintonizador difuso del controlador PI, cuya función es la de regular la velocidad, es decir, el controlador de velocidad convierte el error de velocidad a un par de referencia, T_{em}^* .

Para la sintonización del controlador PI se utilizó el criterio de la respuesta a la frecuencia de Ziegler-Nichols [36], el cual a su vez se emplea en la referencia [57] para la sintonización del regulador de velocidad del control vectorial indirecto del motor de inducción. Por lo tanto, haciendo uso de lo expuesto en estas referencias se obtienen los valores de los parámetros, los cuales son: $K_p = 200$ y $\tau_I = 0.05$, siendo éstos, los valores iniciales para el autosintonizador difuso.

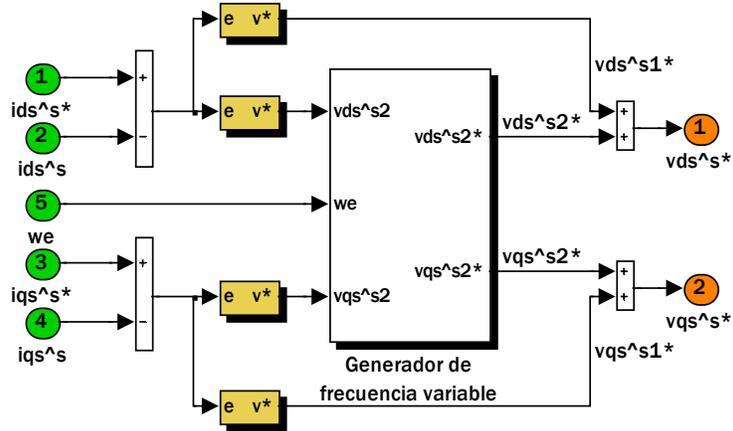
En la Fig. 2.12(f) se muestra el bloque de *Campo Orientado*, en donde las señales de entrada son el par de referencia, T_{em}^* , el flujo del rotor en el eje d , ψ_{dr}^{e*} , y el ángulo del rotor, θ_r . Asimismo, se observa el interior del bloque de *Campo Orientado*, dentro de este bloque, las Ecs. (2.15), (2.16), y (2.17) se usan para calcular los valores de i_{ds}^{e*} , i_{qs}^{e*} y ω_2^* . El ángulo, ρ , es la suma del ángulo de deslizamiento, θ_2 , y del ángulo del rotor, θ_r , además de este bloque se obtiene a la velocidad angular del estator, ω_e , ya que el regulador síncrono la ocupa como señal de entrada.

En este caso se utiliza el bloque *qde2qds* (Fig. 2.12(i)), para realizar la transformación de la corriente bifásica síncrona a la corriente bifásica estacionaria y generar de esta forma la corriente bifásica de referencia en coordenadas *qds*.

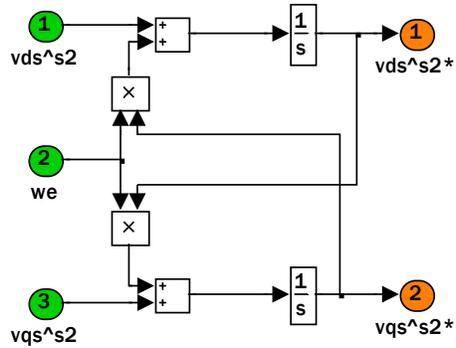
En la simulación de la Fig. 2.12 se utiliza el regulador síncrono que junto con el generador PWM y el inversor fuente de voltaje se usan para generar el voltaje trifásico en los devanados del estator del motor de inducción. La corriente trifásica del motor de inducción se obtiene del bloque *qds2abc* (ver Fig. 2.12 (j)).



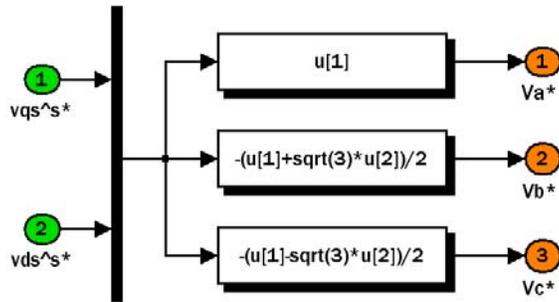
(a) Diagrama general de la simulación.



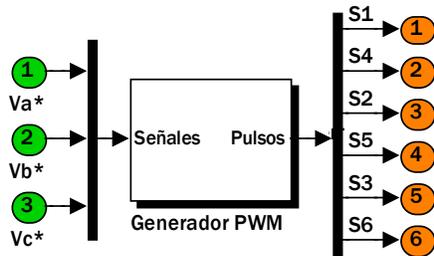
(b.1) Esquema general del regulador de corriente síncrono.



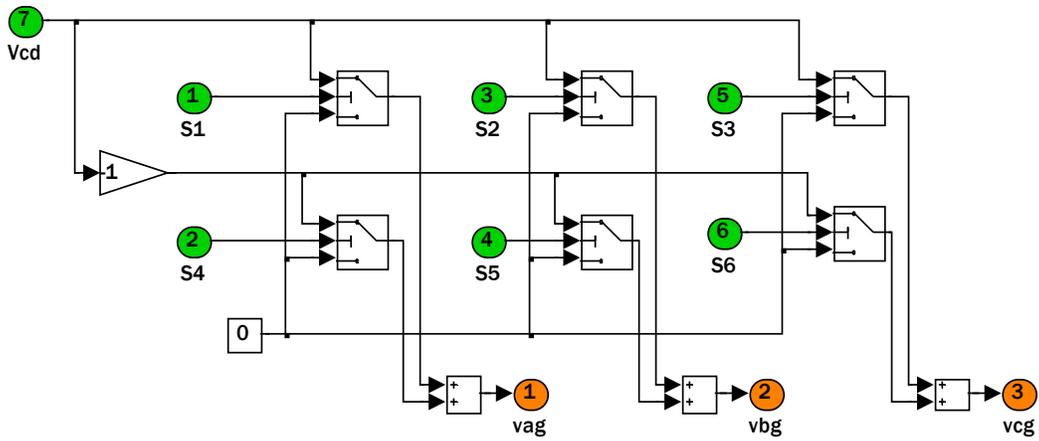
(b.2) Generador de frecuencia variable.



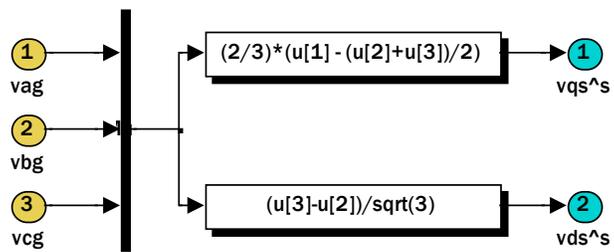
(c) Bloque de la transformación de voltaje $qds2abc$, (bifásica estacionaria-trifásica).



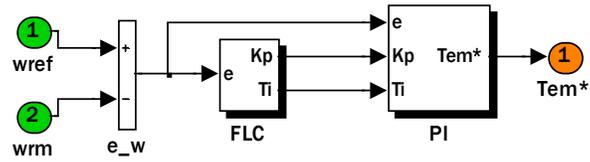
(d) Bloque general del generador PWM senoidal.



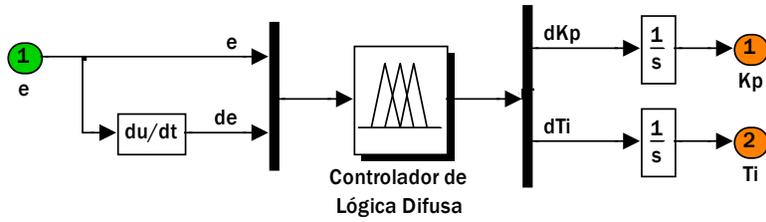
(e) Inversor fuente de voltaje.



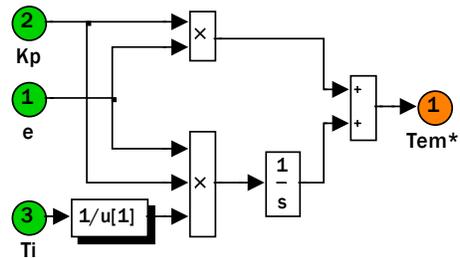
(f) Transformación trifásica a bifásica estacionaria (*abc2qds*).



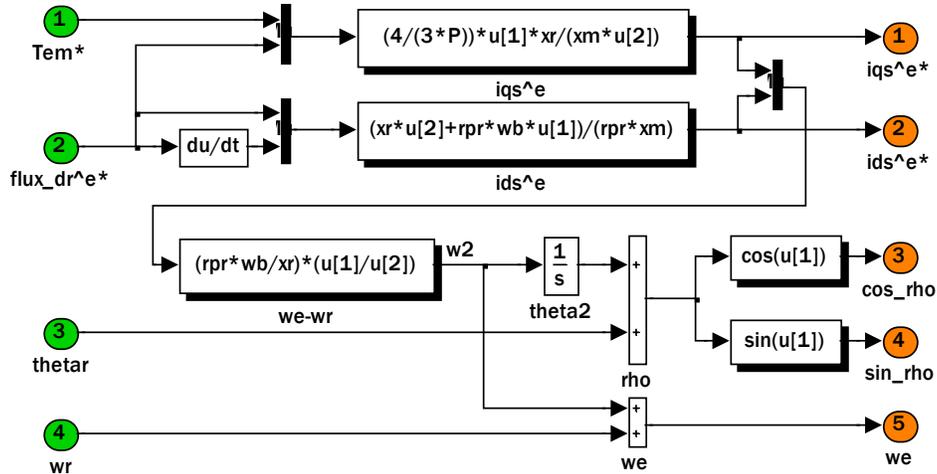
(g1) Esquema general del auto sintonizador difuso del PI.



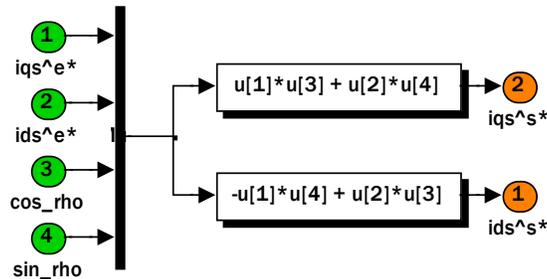
(g2) Sintonizador de lógica difusa.



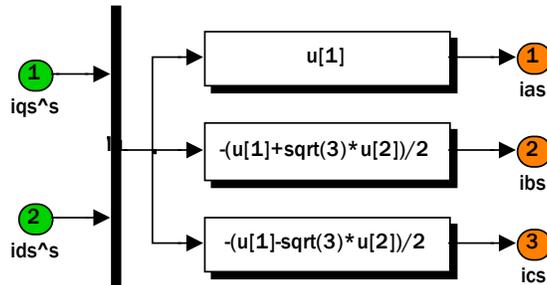
(g3) Controlador Proporcional-Integral.



(h) Control vectorial por campo orientado, con orientación del flujo del rotor.



(i) Bloque de la transformación $qde2qds$, bifásica síncrona–bifásica estacionaria.



(j) Bloque de la transformación $qds2abc$, (bifásica estacionaria–trifásica).

Figura 2.12. Simulación en SIMULINK del control vectorial indirecto del motor de inducción con inversor de voltaje PWM con el regulador de corriente síncrono.

2.7 Resultados de la simulación del control vectorial indirecto

La simulación del control vectorial indirecto con el regulador de corriente síncrono, como se mencionó se realiza en SIMULINK. Se usa el archivo *m2.m* de MATLAB para establecer los parámetros de la máquina y del control, la señal de referencia, y la perturbación de la carga para los siguientes estudios: variación en el par de carga a una velocidad de referencia fija y cambio cíclico de la velocidad de referencia. También se establece las condiciones para el campo debilitado.

2.7.1 Campo debilitado

La relación entre la magnitud del flujo del rotor ψ_{dr}^{te} de referencia y la velocidad mecánica ω_{rm} del rotor se establece en la Tabla 2.3. Para una velocidad menor que la velocidad base o nominal, el ψ_{dr}^{te} se mantiene igual sí es que no se tiene carga, esto se consigue suministrando el voltaje nominal de alimentación [16], en el Apéndice B se describe la operación con campo debilitado. Los valores del ψ_{dr}^{te} y la velocidad mecánica son generados por el archivo *m2.m* de MATLAB (Apéndice A), que también se usa para establecer los parámetros y otras condiciones de ejecución de la simulación en el área de trabajo de MATLAB. Por lo tanto, en la Tabla 2.3, se muestran los valores para el ψ_{dr}^{te} y sus correspondientes valores de ω_{rm} para las condiciones del campo debilitado y en la Fig. 2.13, se muestra su bloque. En el mismo archivo *m2.m* se realiza la interpolación para otros valores de ψ_{dr}^{te} y ω_{rm} que no se muestran en la Tabla 2.3.

	ψ_{dr}^{te} (Wb)	ω_{rm} (rad/s)		ψ_{dr}^{te} (Wb)	ω_{rm} (rad/s)
1	61.2233	-376.9910	22	122.4844	18.8496
2	64.4654	-358.1416	23	122.4844	37.6991
3	68.0468	-339.2920	24	122.4844	56.5487
4	72.0429	-320.4425	25	122.4844	75.3982
5	76.5291	-301.5929	26	122.4844	94.2478
6	81.6562	-282.7433	27	122.4844	113.0973
7	87.4996	-263.8938	28	122.4844	131.9469
8	94.2100	-245.0442	29	122.4844	150.7964
9	102.0514	-226.1947	30	122.4844	169.6460
10	122.4844	-207.3451	31	122.4844	188.4956
11	122.4844	-188.4956	32	122.4844	207.3451
12	122.4844	-169.6460	33	102.0514	226.1947
13	122.4844	-150.7964	34	94.2100	245.0442
14	122.4844	-131.9469	35	87.4996	263.8938
15	122.4844	-113.0973	36	81.6562	282.7433
16	122.4844	-94.2478	37	76.5291	301.5929
17	122.4844	-75.3982	38	72.0429	320.4425
18	122.4844	-56.5487	39	68.0468	339.2920
19	122.4844	-37.6991	40	64.4654	358.1416
20	122.4844	-18.8496	41	61.2233	376.991
21	122.4844	0			

Tabla 2.3. Valores para el ψ_{dr}^{te} y sus correspondientes valores de ω_{rm} para el campo debilitado.

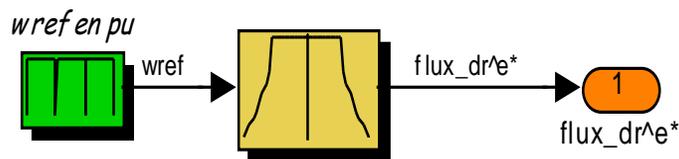


Figura 2.13. Campo debilitado.

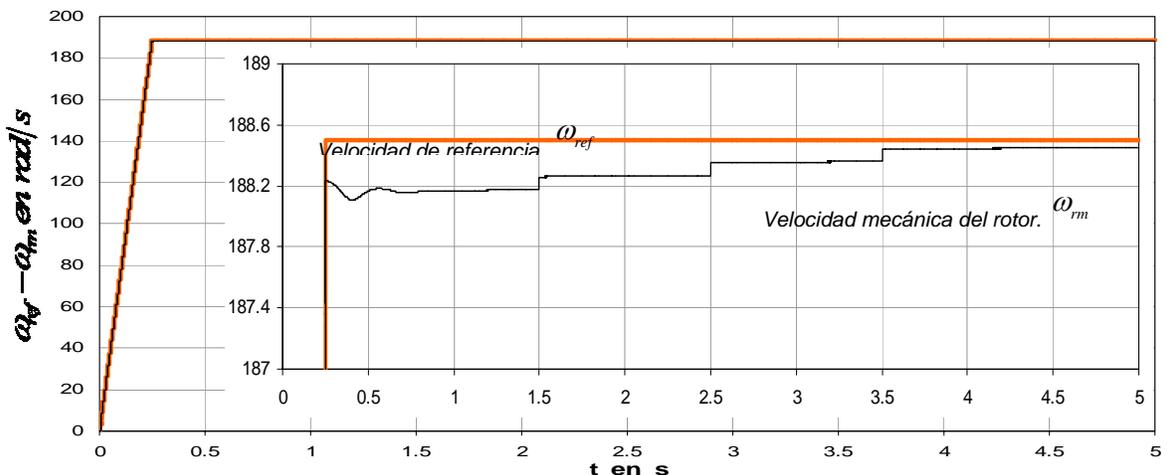
2.7.2 Par de carga variable a una velocidad de referencia fija

En este estudio, la velocidad de la máquina se incrementa hasta su velocidad nominal y posteriormente se somete a una secuencia de disturbios debido a variaciones en el valor del par de carga constante (escalón). El archivo *m2.m* de MATLAB coloca los siguientes valores:

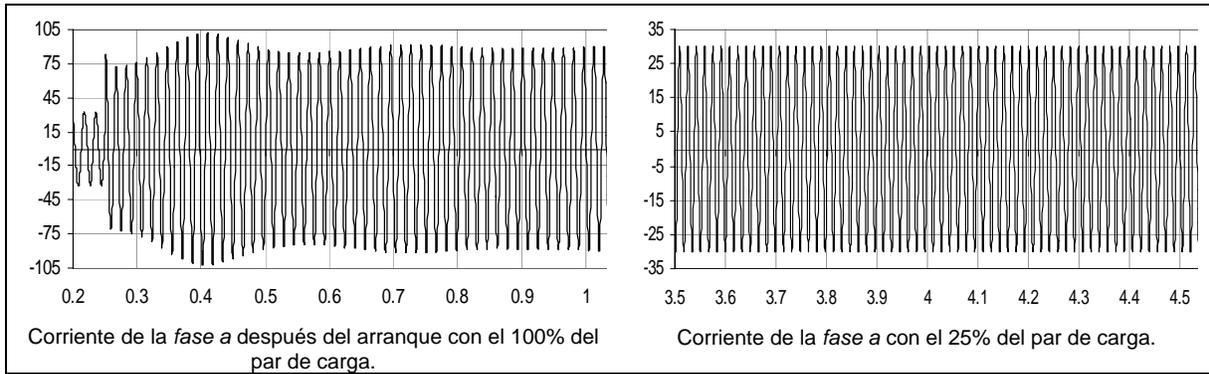
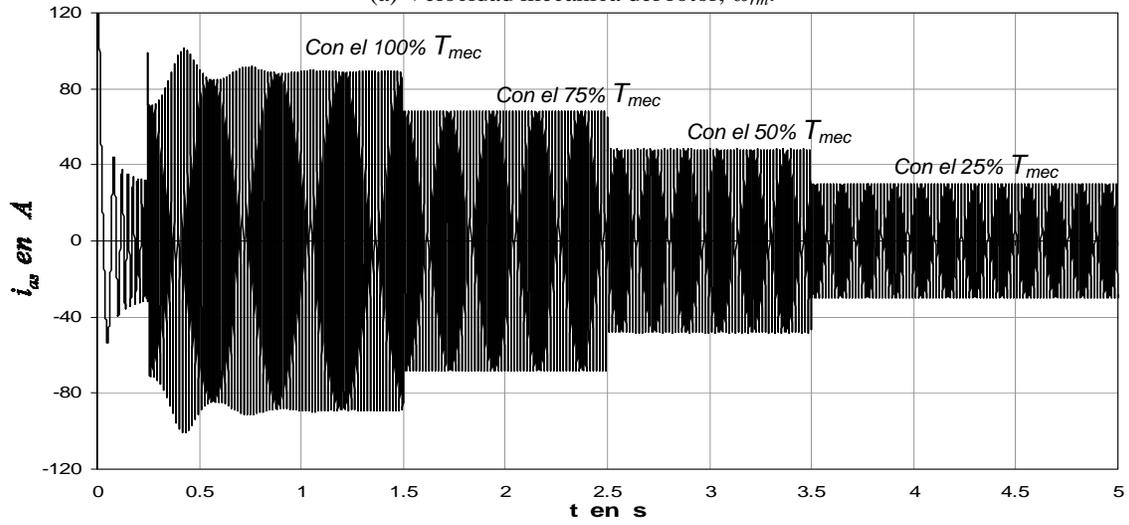
<i>t_paro</i>	El tiempo de estudio a 5 segundos.
<i>tiempo_wref</i>	Lapso de tiempo para la secuencia de la velocidad de referencia: [0 0.25 <i>t_paro</i>].
<i>vel_wref</i>	Secuencia de la velocidad de referencia: [0 ω_{ref} ω_{ref}].
<i>tiempo_tmec</i>	Lapso de tiempo para la secuencia del par de carga constante: [0 0.25 0.25 1.5 1.5 2.5 2.5 3.5 3.5 5].
<i>tmec_tmec</i>	Secuencia del par de carga constante: [0 0 $-T_{nom}$ $-T_{nom}$ $-0.75 \times T_{nom}$ $-0.75 \times T_{nom}$ $-0.5 \times T_{nom}$ $-0.5 \times T_{nom}$ $-0.25 \times T_{nom}$ $-0.25 \times T_{nom}$]

Se ejecuta la simulación usando el método ode4 (fórmula de Runge-Kutta de cuarto orden), con un tiempo de ejecución de 5 s. y un tamaño de paso fijo de 2×10^{-6} s.

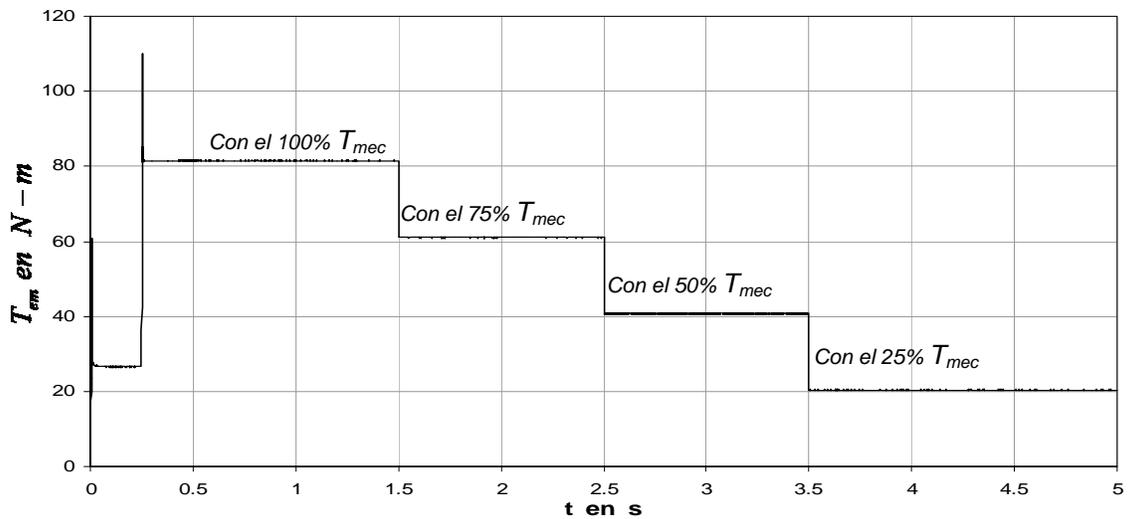
Se trazan los valores de la velocidad de referencia, ω_{ref} , la velocidad mecánica del rotor, ω_{rm} , la corriente de la fase *a* del estator, i_{as} , el par electromagnético, T_{em} , y la magnitud del flujo del rotor, $\sqrt{\psi_{qr}^2 + \psi_{dr}^2}$. En la Fig. 2.14 se muestran los resultados de la simulación para el regulador de corriente síncrono. Observe qué la velocidad del motor responde a la velocidad de referencia programada con el control vectorial indirecto. También observe como el control vectorial indirecto mantiene la magnitud del flujo del rotor durante las diferentes condiciones de operación del motor de inducción.

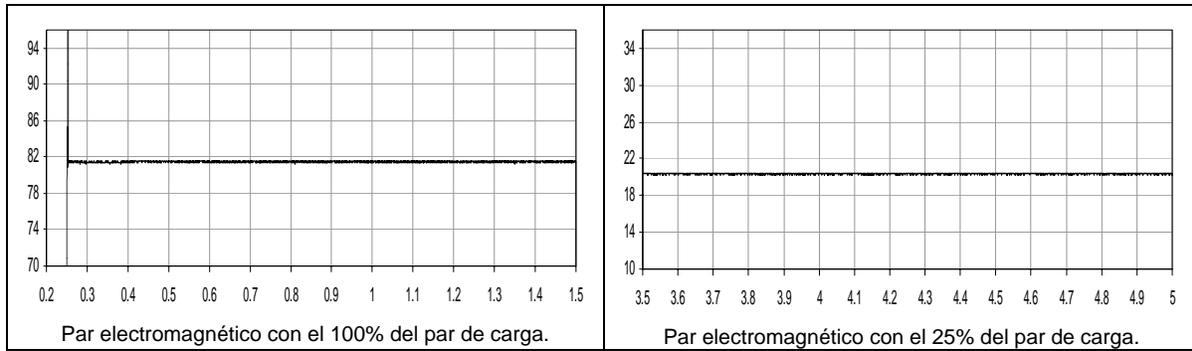


(a) Velocidad mecánica del rotor, ω_m .

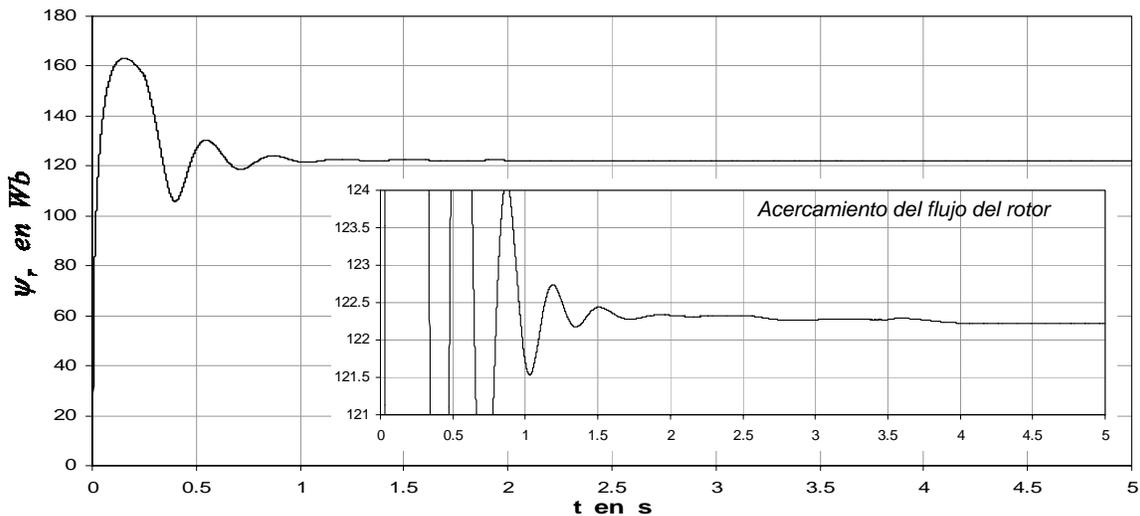


(b) Corriente de la fase a del estator, i_{as} .





(c) Par electromagnético, T_{em} .



(d) Flujo del rotor ψ_r .

Figura 2.14. Arranque con variación de carga a velocidad fija con regulador de corriente síncrono en el control vectorial indirecto.

En las Figs. 2.14(a) y 2.14(d) se realiza un acercamiento de las graficas de la velocidad mecánica del rotor y del flujo del rotor, respectivamente, en donde se observa el desempeño del regulador de corriente síncrono en el control vectorial indirecto. Asimismo se observa que la velocidad mecánica del rotor con los cambios de carga tiende a seguir a la velocidad de referencia y la magnitud del flujo del rotor se mantiene en su valor nominal.

Se muestran en las Fig. 2.14(b) la corriente de la *fase a* del estator y en la Fig. 2.14(c) el par electromagnético. Se realiza un acercamiento en ambas graficas, observando el comportamiento de la corriente i_{as} y del par electromagnético cuando el motor trabaja con el 100% del par de carga y en el momento en el cual solo tiene el 25% del par de carga.

En la Fig. 2.15 se muestran las gráficas de las trayectorias de las corrientes estacionarias, dqs . Las graficas que se ilustran consideran los siguientes casos:

- Tiempo de simulación de 1.0 – 1.5 s con el 100% del par de carga (Fig. 2.15(a)).

- Tiempo de simulación de 2.0 – 2.5 s con el 75% del par de carga (Fig. 2.15(b)).
- Tiempo de simulación de 3.0 – 3.5 s con el 50% del par de carga (Fig. 2.15(c)).
- Tiempo de simulación de 4.0 – 4.5 s con el 25% del par de carga (Fig. 2.15(d)).

En las graficas de la Fig. 2.15, se observa que las trayectorias de las corrientes, exhiben un cambio instantáneo siguiendo las variaciones del par de carga, asimismo se genera una trayectoria circular, ya que la corriente bifásica en coordenadas estacionarias están desfasadas 180° entre sí. De esta manera, para las simulaciones posteriores relacionadas a la estimación de la resistencia del rotor (ver Capítulo 3), se utilizará el regulador de corriente síncrono.

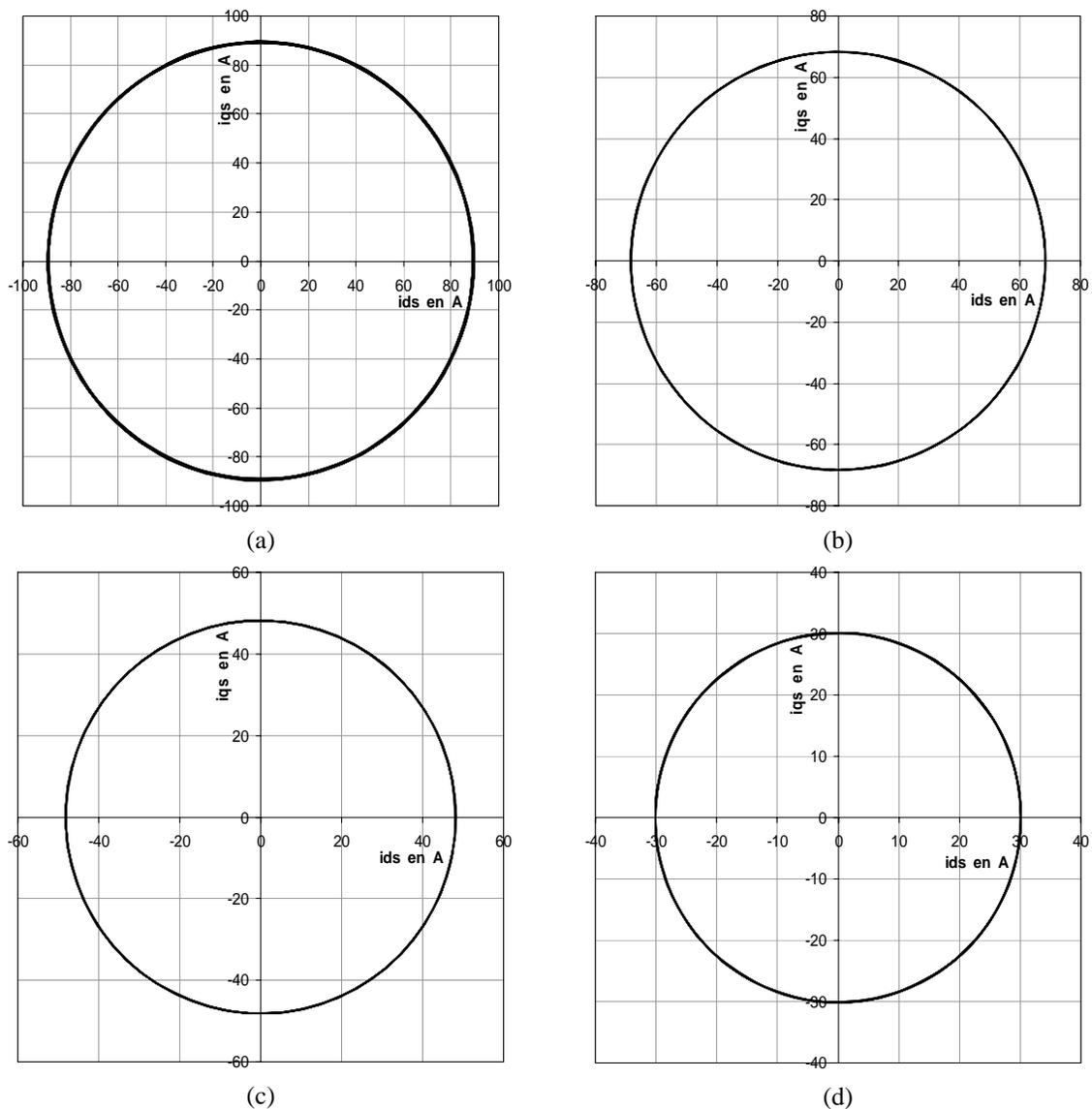


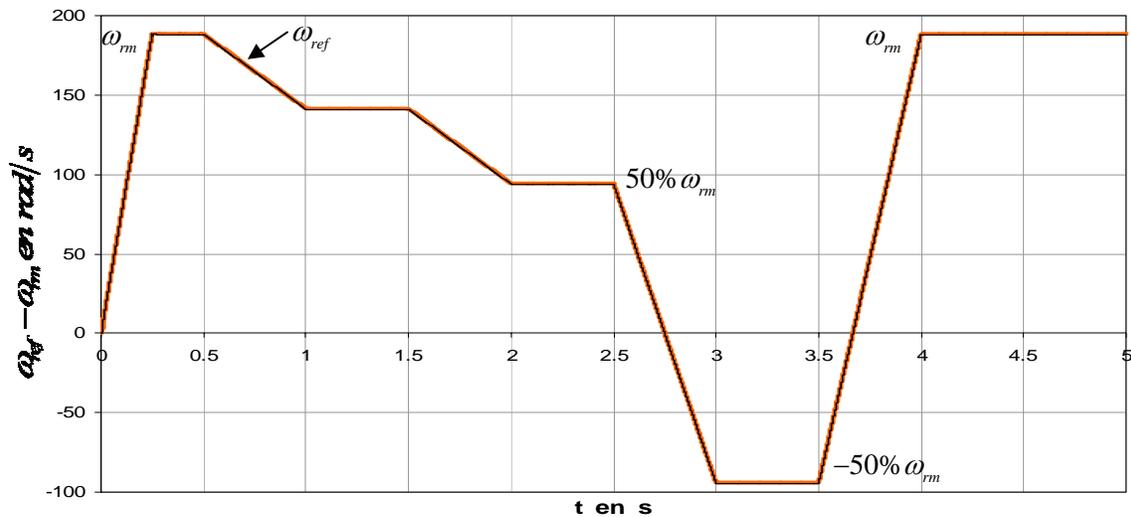
Figura 2.15. Gráficas de las trayectorias de las corrientes estacionarias dqs usando el regulador de corriente síncrono. Con variación de carga a velocidad fija.

2.7.3 Cambio cíclico de la velocidad de referencia

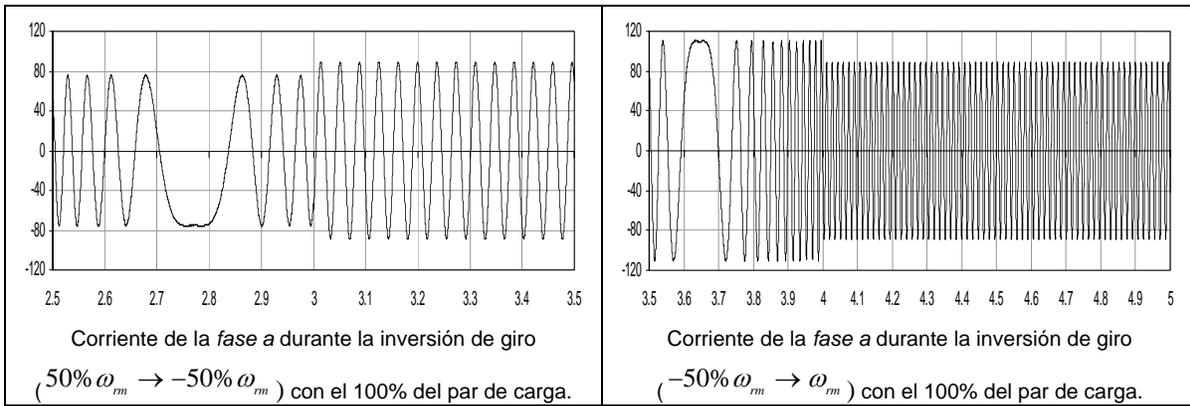
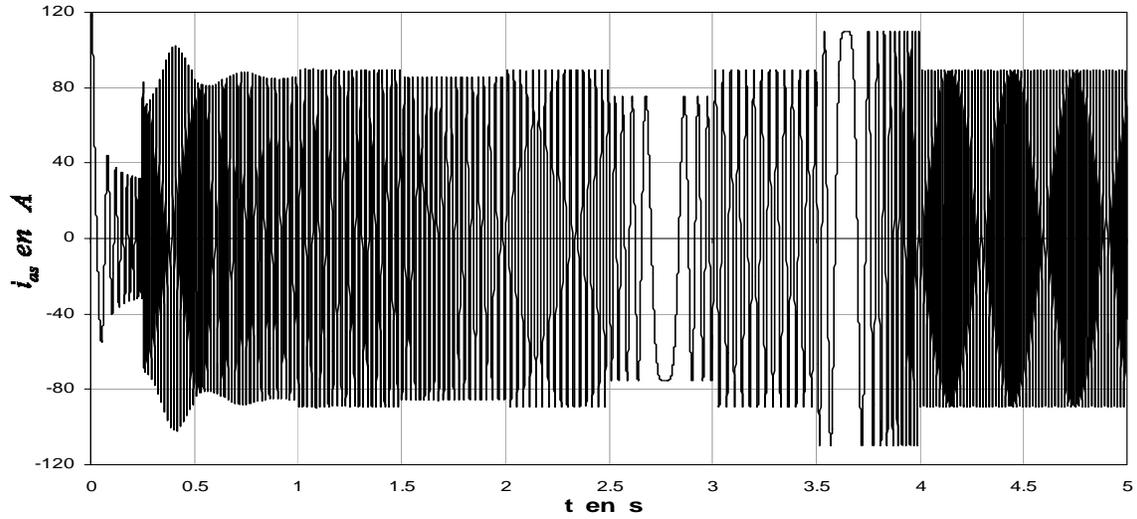
Para este estudio, el archivo *m2.m* de MATLAB coloca los siguientes valores:

<i>t_paro</i>	El tiempo de estudio a 5 segundos.
<i>tiempo_wref</i>	Lapso de tiempo para la secuencia de la velocidad de referencia: [0 0.25 0.5 0.75 1.5 1.5 2.5 2.5 3.5 3.5 5].
<i>vel_wref</i>	Secuencia de la velocidad de referencia: [0 ω_{ref} ω_{ref} $0.75 \times \omega_{ref}$ $0.75 \times \omega_{ref}$ $0.5 \times \omega_{ref}$ $0.5 \times \omega_{ref}$ $-0.5 \times \omega_{ref}$ $-0.5 \times \omega_{ref}$ ω_{ref} ω_{ref}]
<i>tiempo_tmec</i>	Lapso de tiempo para la secuencia del par de carga constante: [0 0.25 <i>t_paro</i>].
<i>tmec_tmec</i>	Secuencia del par de carga constante: [0 0 $-T_{nom}$].

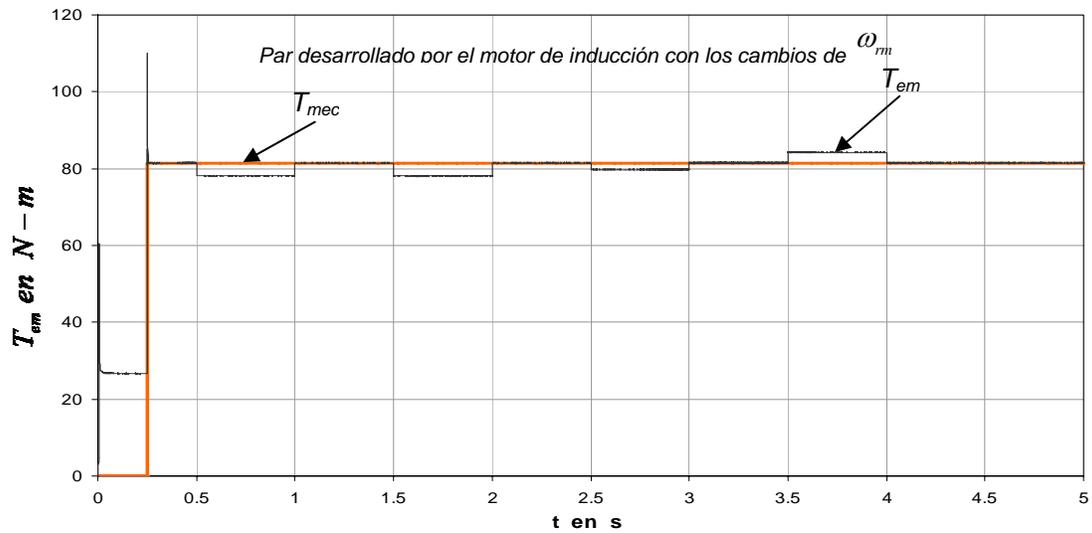
Se realiza la simulación aplicándose al motor un par mecánico igual al par de carga nominal, que es $T_{mec} = -T_{nom}$. En la Fig. 2.16 se muestran los resultados, para el regulador de corriente síncrono. Así como en el apartado 2.7.2, también se observa que la velocidad mecánica del rotor responde a la velocidad de referencia programada con el control vectorial indirecto. Del mismo modo se observa como el control mantiene la magnitud del flujo del rotor durante las diferentes condiciones de operación del motor de inducción.



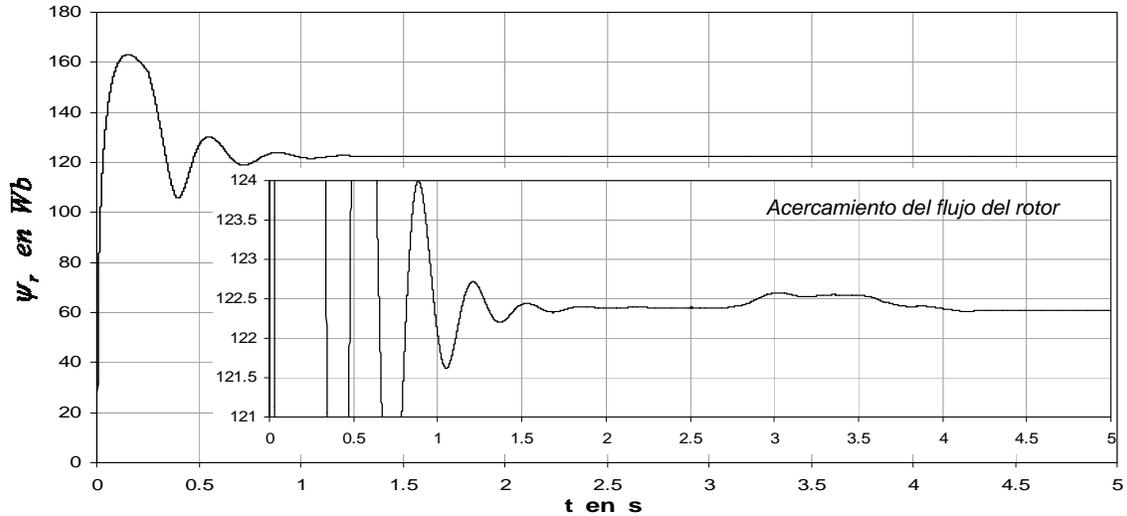
(a) Velocidad mecánica del rotor, ω_m .



(b) Corriente de la fase a del estator, i_{as} .



(c) Par electromagnético, T_{em} .



(d) Flujo del rotor.

Figura 2.16. Arranque con variación de la velocidad de referencia con carga fija con regulador de corriente síncrono en el control vectorial indirecto.

Esta prueba permite evaluar el comportamiento del motor de inducción frente a la presencia de cambios de la velocidad de referencia, manteniendo el par de carga nominal. Al igual que en la prueba anterior, en la Fig. 2.16(d) se realiza un acercamiento de la grafica del flujo del rotor, en donde la magnitud del flujo se mantiene en su valor nominal durante los cambios de la velocidad de referencia.

También se observa en la Fig. 2.16(a) el desempeño del regulador de corriente síncrono en el control vectorial indirecto, en donde la velocidad mecánica del rotor tiende a seguir a la velocidad de referencia, es decir, cuando la velocidad disminuye (al 75% y 50% de ω_{rm}), cuando se realiza la inversión de giro hacia una velocidad negativa (al -50% de ω_{rm}), y finalmente se establece la velocidad mecánica del rotor nominal (ω_{rm}). También se observa que la velocidad de referencia y la velocidad mecánica del rotor están totalmente sobrepuestas.

En la Fig. 2.16(c), el motor de inducción responde produciendo el par electromagnético dependiendo de la variación de la velocidad mecánica del rotor, además se observa de que al permanecer constantes tanto el flujo como la corriente estacionaria dqs , lo mismo sucede con el par electromagnético.

Por otro lado se observa que la variación de la corriente trifásica del estator (en la Fig. 2.16(b) se muestra la corriente de la fase a), esta en relación a los cambios de la velocidad de referencia, ya que el par de carga se mantiene constante, por lo tanto, como se puede observar, en el momento en que la velocidad se hace negativa ($t = 2.75$ s), lo mismo sucede con la corriente trifásica, cambiando su secuencia de fase. Asimismo, en el momento en que la velocidad se vuelve a hacer positiva ($t = 3.67$ s), la corriente trifásica, sigue con la secuencia de fase inicial.

En la Fig. 2.17 se muestran las gráficas de las trayectorias de las corrientes estacionarias dqs . Las graficas que se ilustran consideran el 100% del par de carga con los siguientes casos:

- Tiempo de simulación de 2.0 – 3.0 s con la velocidad de referencia variando de +50% ω_{rm} a -50% ω_{rm} (Fig. 2.17(a)).
- Tiempo de simulación de 3.5 – 4.5 s con la velocidad de referencia variando de -50% ω_{rm} a +100% ω_{rm} (Fig. 2.17(b)).

Como se observa, las trayectorias de las corrientes, claramente exhiben un cambio instantáneo en la trayectoria, siguiendo las variaciones de la velocidad de referencia (Fig. 2.17), además el desempeño del regulador de corriente síncrono genera una trayectoria circular, debido a que la corriente bifásica en coordenadas estacionarias están desfasadas 180° entre si. De esta manera, para las simulaciones posteriores relacionadas a la estimación de la resistencia del rotor, se utilizara el regulador síncrono.

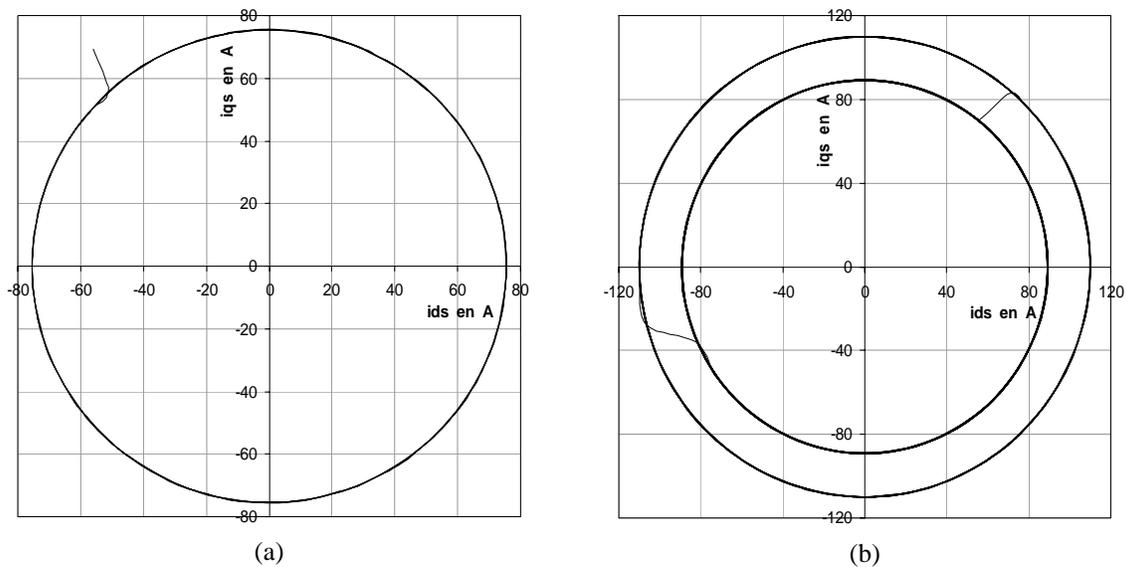


Figura 2.17. Gráficas de las trayectorias de las corrientes estacionarias dqs en el regulador de corriente síncrono. Con variación de la velocidad de referencia con carga fija.

Capítulo 3

ESTIMACIÓN DE LA RESISTENCIA DEL ROTOR UTILIZANDO UNA RED NEURONAL ARTIFICIAL

3.1 Introducción

En el control vectorial indirecto del motor de inducción, una sintonización adecuada de sus parámetros, ocasiona que el motor de inducción este desacoplado para que se comporte como un motor de CD de excitación separada, con un control independiente de la corriente de armadura (par) y la corriente de campo (flujo) [54]-[55]. En general, la medición y sintonización de los parámetros del motor de inducción son normalmente complicados y consumen tiempo de procesamiento.

Se han realizado investigaciones sobre la medición o ajuste de la resistencia del rotor durante su operación en estado estable o antes del arranque del accionamiento [9], [24], [25], [29], [30], [56]. Sin embargo, estos no son métodos efectivos para conseguir un control desacoplado ya que solamente se conoce la resistencia del rotor en estado estable y no durante el arranque y la operación transitoria del motor de inducción. Por lo tanto, en este capítulo, se desarrolla un procedimiento sistemático para la estimación de la resistencia del rotor del motor de inducción, cuyo valor estimado se utiliza para actualizar el valor de la resistencia en el algoritmo de control y evitar la desintonización del control vectorial indirecto. Este estimador se basa en una red neuronal artificial.

Las redes neuronales artificiales al tener la capacidad de aprender, han mostrado prometedoras aplicaciones en el campo de la ingeniería de procesos así como en la electrónica de potencia, particularmente se han aprovechado como estimadores en sistemas con dinámica altamente no lineal. De esta manera se presenta una aplicación donde las redes neuronales artificiales estiman la resistencia del rotor en un motor de inducción, el cual esta siendo controlado por un accionamiento que se basa en el control vectorial indirecto. En forma general, este método se basa en la red neuronal artificial multicapa de realimentación (feedforward) cuyo entrenamiento se realiza con el algoritmo de retropropagación, por lo tanto, el error entre el valor deseado de la resistencia del rotor, el cual se obtiene a partir de un modelo matemático que incluye a un controlador de lógica difusa tomado de la referencia [42]-[43], y el valor de la resistencia del rotor obtenido de un modelo neuronal es retropropagado para ajustar los pesos de la red neuronal artificial propuesta, de manera que el valor real siga al valor deseado. El modelo matemático que se tomo como patrón para entrenar la red, también puede ser utilizado como estimador de la resistencia del rotor [43]. Este estimador está basado como se mostrará posteriormente, en un controlador difuso. Los resultados que se obtienen en este trabajo con el estimador neuronal propuesto, son mejores en condiciones dinámicas de operación que los resultados

que se obtienen con el estimador usado como patrón, como se demostrará posteriormente en este trabajo.

3.2 Efecto de los cambios de parámetros en el control vectorial indirecto

En el control por orientación del flujo del rotor, el bloque principal es el llamado controlador vectorial indirecto (Fig. 3.1), el cual se describió en el apartado 2.3, en donde se utilizaron las siguientes relaciones:

$$i_{ds}^{e*} = \psi_{dr}'^{e*} \frac{r_r' + L_r' p}{r_r' L_m} \quad (3.1)$$

$$i_{qs}^{e*} = \frac{2}{3} \frac{2}{P} \frac{L_r'}{L_m} \frac{T_{em}^*}{\psi_{dr}'^{e*}} \quad (3.2)$$

$$\omega_2^* = \omega_e - \omega_r = \frac{r_r'}{L_r'} \frac{i_{qs}^{e*}}{i_{ds}^{e*}} \quad (3.3)$$

donde el operador p ahora es interpretado como el equivalente del operador s de Laplace, lo cual se hace para evitar confusión con la s del deslizamiento. En las ecuaciones anteriores, el superíndice $*$ denota el valor comando o de referencia de las variables del motor. Estos constituyen la base para el control en las regiones de par constante y de campo debilitado (Fig. 3.1). Sin embargo, en el caso de flujo constante $\psi_{dr}'^{e*} = \text{constante}$, las Ecs. (3.1)–(3.3) se simplifican considerablemente:

$$i_{ds}^{e*} = \frac{\psi_{dr}'^{e*}}{L_m} \quad (3.4)$$

$$i_{qs}^{e*} \sim T_{em}^* \quad (3.5)$$

$$\omega_2^* \sim \frac{r_r'}{L_r'} i_{qs}^{e*} \quad (3.6)$$

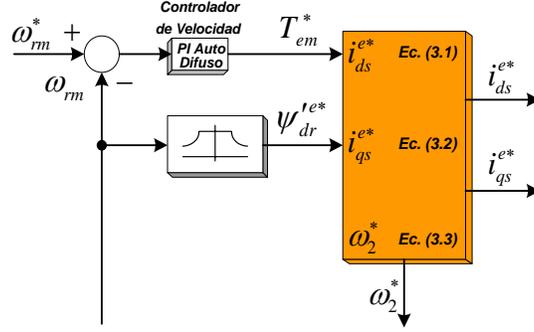


Figura 3.1. Extensión del esquema de control orientado del flujo del rotor indirecto para operación en campo debilitado.

El esquema del control vectorial indirecto sólo es efectivo mientras el conjunto de valores de los parámetros del motor en el controlador vectorial son iguales a los valores de los parámetros reales del motor [32]. Para la región de operación constante del flujo del rotor, un cambio de la constante de tiempo del rotor T_r (igual a L'_r/r'_r) produce la desviación en el valor de la velocidad de deslizamiento, ω_2^* calculada en la Ec. (3.3). Por lo tanto, la posición prevista del flujo del rotor, ρ^* , se desvía de la posición real, ρ , la cual produce una desviación angular del par $\Delta\delta = \theta_2 = \rho^* - \rho$ y, por consiguiente, la condición de desacoplamiento del flujo y del par no puede alcanzarse, tomando en cuenta que el vector de la corriente del estator i_s produce las dos componentes i_{ds}^e e i_{qs}^e (Fig. 2.5). Esto ocasiona los dos casos siguientes:

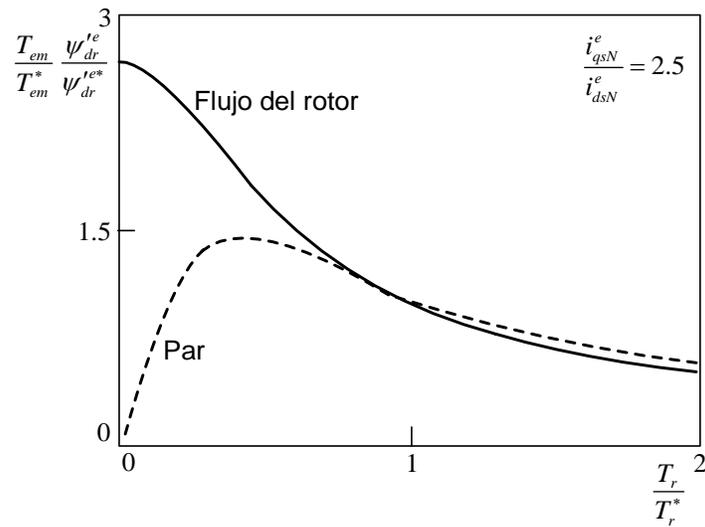
- Valores incorrectos del flujo del rotor ψ_{dr}^{e*} y de la componente de corriente del par i_{qs}^e en los puntos de operación en estado estable (para $T_{em}^* = \text{constante}$).
- El motor de inducción no responde a los cambios del par comando T_{em}^* .

Se han analizado en varias publicaciones [33], [34], [35] los efectos de desintonización de los parámetros del motor. Para un punto predeterminado de operación, definido por los valores de las corrientes comando del par y del flujo i_{qs}^{e*} , i_{ds}^{e*} , es posible determinar el efecto de los cambios de la constante de tiempo del rotor sobre el par real y el flujo del rotor del motor. Las siguientes relaciones (Ecs. (3.7) y (3.8)), que son presentadas en [32] de donde fueron derivadas del circuito equivalente del motor en estado estable, son expresadas adecuadamente de la siguiente forma:

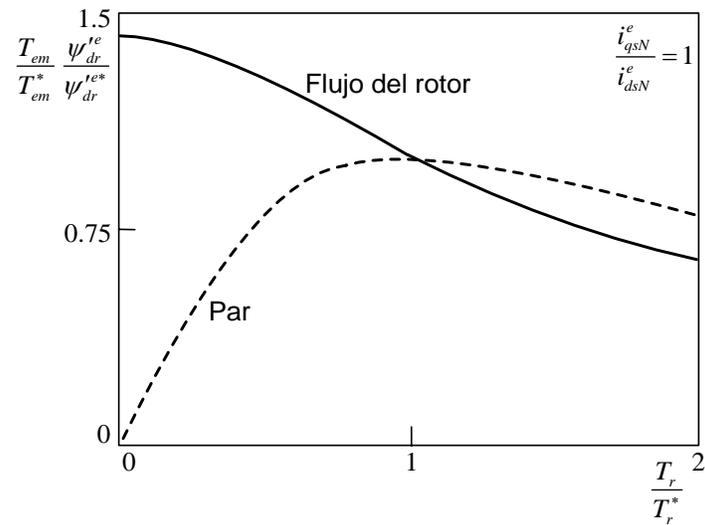
$$\frac{T_{em}}{T_{em}^*} = \frac{T_r}{T_r^*} \frac{1 + \left(\frac{i_{qs}^{e*}}{i_{ds}^{e*}}\right)^2}{1 + \left[\frac{T_r}{T_r^*} \left(\frac{i_{qs}^{e*}}{i_{ds}^{e*}}\right)\right]^2} \quad (3.7)$$

$$\frac{\psi_{dr}^{ie}}{\psi_{dr}^{ie*}} = \frac{\sqrt{1 + \left(\frac{i_{qs}^{e*}}{i_{ds}^{e*}}\right)^2}}{\sqrt{1 + \left[\frac{T_r}{T_r^*} \left(\frac{i_{qs}^{e*}}{i_{ds}^{e*}}\right)\right]^2}} \quad (3.8)$$

Por lo tanto, de las ecuaciones anteriores se deduce que los valores normalizados del par y del flujo del rotor no son funciones lineales de la relación *real/predicha* de la constante de tiempo del rotor (T_r/T_r^*) y del punto de operación del motor (i_{qs}^{e*}/i_{ds}^{e*}).



(a)



(b)

Figura 3.2. Desintonización en estado estable del efecto de los parámetros para las corrientes comando nominales de flujo y de par: (a) motores de alta potencia, (b) motores de baja potencia.

Para los valores nominales (N) de las corrientes comando del campo orientado $i_{qs}^{e*} = i_{qsN}^e$ e $i_{ds}^{e*} = i_{dsN}^e$, se obtienen las curvas de la Fig. 3.2 (donde el efecto de la saturación se omite), a partir de las Ecs. (3.7) y (3.8). Se observa que los motores de alta potencia, tienen una pequeña corriente de magnetización i_{Mr} (en estado estable $i_{ds}^e = i_{Mr}$) relativa a la corriente nominal i_{sN} , estos tipos de motores se caracterizan por tener valores grandes de la relación $i_{qsN}^e / i_{dsN}^e = 2 \rightarrow 2.8$. Por otro lado, para motores de baja potencia, se tienen valores pequeños de la relación $i_{qsN}^e / i_{dsN}^e = 1 \rightarrow 2$ [32].

Las curvas de la Fig. 3.2 muestran que el flujo se incrementa, si el valor de la constante de tiempo del rotor es más bajo que el predicho ($T_r / T_r^* < 1$) (ver la Fig. 2.5). Los cambios del par y del flujo dependen del valor de i_{qsN}^e / i_{dsN}^e . Nótese que los motores de alta potencia son más sensibles a la desintonización de la constante de tiempo del rotor (T_r / T_r^*) que los de baja potencia.

3.3 Adaptación de los parámetros en el control vectorial indirecto

Como se mencionó en 3.2, un parámetro crítico para las condiciones de desacoplamiento de un sistema de control vectorial indirecto es la constante de tiempo del rotor T_r . Esta cambia principalmente bajo la influencia de los cambios de temperatura de la resistencia del rotor (r_r'). Se asume que T_r cambia en el rango de $0.75T_{r0} < T_r < 1.5T_{r0}$, donde T_{r0} es la constante de tiempo del rotor a la carga nominal y una temperatura de 75°C [32].

La corrección de la constante de tiempo del rotor se efectúa por adaptación en línea. A partir del diagrama de la Fig. 3.3, se obtiene la señal de corrección para los cambios de la constante de tiempo del rotor ($\Delta(1/T_r)$) a partir de los valores medidos reales del par, del flujo, o de cantidades conocidas como son las corrientes de par o de flujo. Sin embargo, estas cantidades son muy difíciles de medir o calcular en el esquema del control vectorial indirecto, su dificultad es comparable a la estimación vectorial del flujo en el sistema de control vectorial directo.

La Figura 3.3 muestra la representación básica de un esquema de adaptación de T_r que corresponde a la estructura del sistema adaptable al modelo de referencia [32].

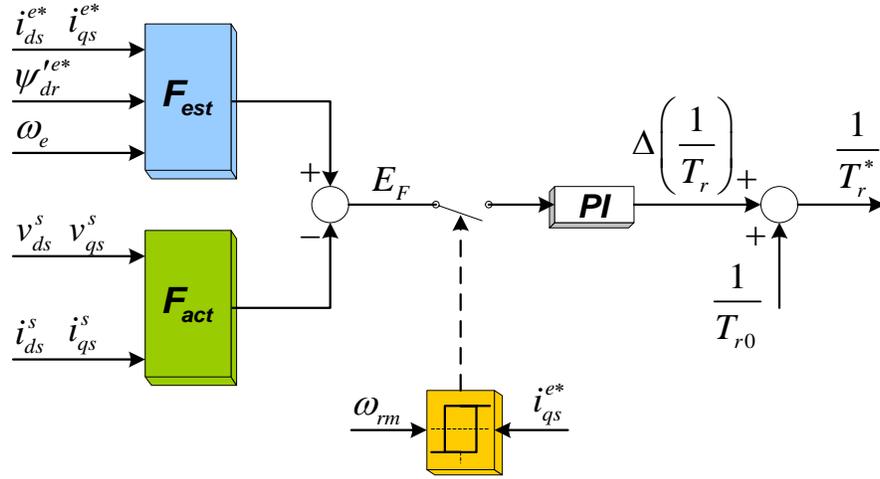


Figura 3.3. Principio de adaptación de la T_r basado en el esquema adaptable al modelo de referencia.

La función de referencia F_{est} se calcula de las cantidades de referencia en las coordenadas de campo. Por otro lado, la función estimada F_{act} se calcula de las cantidades medidas, las cuales normalmente son expresadas en las coordenadas orientadas del estator. El error $E_F = F_{est} - F_{act}$ es la señal de entrada al controlador PI, el cual genera la corrección necesaria de la constante de tiempo del rotor ($\Delta(1/T_r)$). Esta señal de corrección se adiciona al valor inicial ($1/T_{r0}$) dando la constante de tiempo actualizada ($1/T_r^*$), la cual finalmente se usa para el cálculo de la velocidad de deslizamiento ω_2^* . Es necesario resaltar que en estado estable, cuando el $E_F \rightarrow 0$ entonces $T_r^* \rightarrow T_r$. Para la identificación de los cambios de r_r' , se han sugerido varios criterios de la función, F , (ver la Tabla 3.1). La mayoría de estos criterios no trabajan para condiciones en la que no hay carga en el motor, ni para cuando la velocidad es cero. Por consiguiente, en la región cercana a la velocidad cero y operación sin carga, la señal de salida del error calculado debe bloquearse; y el último valor de $\Delta(1/T_r)$ se mantiene en el controlador PI de la constante de tiempo del rotor, T_r .

El método de correlación propuesto en [36] y que se desarrolla en [37] tiene la ventaja de que no se requiere ninguna medida adicional, pero necesita un incremento del 30–40% del tiempo computacional. El tiempo de respuesta del algoritmo de identificación también depende de la carga del motor de inducción. Las desventajas son, primero, bajo tiempo de actualización (debido al tiempo promedio requerido para construir una función de correlación significativa), y segundo, el rizo del par introducido por la secuencia binaria pseudo aleatoria (PRBS) usada como una señal de prueba, la cual no se permite en los accionamientos prácticos.

También se han propuesto otros métodos de identificación de parámetros en línea basados en la técnica del observador [38], [39].

F_{est}	F_{act}	Parámetro sensible	Autores	Observaciones
$-\left(\frac{X_M}{X_r}\right)\psi_r^*\omega_e^*i_{ds}^{e*}$	$\left(v_{ds}^s i_{qs}^s - v_{qs}^s i_{ds}^s\right) +$ $-\sigma X_s \left(p i_{ds}^s i_{qs}^s - p i_{ds}^s i_{qs}^s\right)$	σX_s	Garces [34]	Problema con la integración.
$-\psi_{dr}^{r*e}\omega_e i_{ds}^{e*}$	$\frac{L'_r}{L_M} \left[\left(i_{qs}^e v_{ds}^e - i_{ds}^e v_{qs}^e \right) \right.$ $\left. + \left(i_{qs}^{e2} + i_{ds}^{e2} \right) L_s \sigma \omega_e \right]$	σL_s	Koyama et al. [40]	v_{ds}^e y v_{qs}^e son salidas de los controladores de corriente.
$\left(\frac{X_M}{X_r}\right)\psi_r^*\omega_e^*i_{qs}^{e*}$	$\psi_{ds}^s i_{qs}^s - \psi_{qs}^s i_{ds}^s$	r_s	Lorenz and Lawson [44]	
i_{qs}^{e*}	$\frac{\psi_{ds}^s i_{qs}^s - \psi_{qs}^s i_{ds}^s}{\frac{X_r}{X_M} \sqrt{\psi_{ds}^s 2 + \psi_{qs}^s 2}}$	r_s	Rowan et al. [9]	Condición inicial y problema de saturación (integración pura).
0	$v_{ds}^e - r_s i_{ds}^e + \omega_e \sigma X_s i_{qs}^e$	$r_s, \sigma X_s$	Okuyama [46], Schumacher [47]	Simple, buena relación de convergencia.

Tabla 3.1 Algoritmos de adaptación de T_r (Fig. 3.3)

3.4 Descripción del método usado para la estimación y actualización de la resistencia del rotor

Para la operación del accionamiento y sin considerar los efectos derivados de la saturación, la constante de tiempo del rotor puede cambiar en función de la temperatura en el rango de $0.75T_{r0} < T_r < 1.5T_{r0}$. Para estimar y adaptar este parámetro sin realizar mediciones dentro del motor de inducción, se estableció una función en [34], la cual proporciona la información necesaria del estado del flujo del rotor (magnitud y posición). Esta función es determinada por medio de la expresión modificada de la potencia reactiva, la cual puede ser calculada usando las corrientes y los voltajes del estator, y no depende de la resistencia del estator r_s . La mayoría de los métodos propuesto para estimar el valor de la constante de tiempo del rotor, comparan el valor estimado del flujo del rotor en el circuito de control con el valor calculado del flujo del rotor. Estos métodos son inexactos para el cálculo del flujo en la región de baja velocidad.

Para resolver este problema en [40], se desarrolla un método de identificación, el cual se basa en el método descrito en [34]. Por lo tanto, de la referencia [40] y de la Tabla 3.1 [32] se utiliza la función F_{est} , la cual contribuye a la estimación de la constante de tiempo del rotor. Esta función esta expresada como:

$$F_{est} = -i_{ds}^{e*}\psi_{dr}^{r*e}\omega_e \quad (3.9)$$

Es importante resaltar que para el cálculo de F_{est} el valor de la resistencia del estator no se requiere. Por consiguiente, la función F_{est} puede calcularse con la exactitud suficiente, incluso en la región de baja velocidad.

Ahora, se analiza la disponibilidad de F_{est} para la identificación de la constante de tiempo del rotor. Por lo tanto, en estado estable la función F_{est} en el circuito de control se expresa como en la Ec. (3.9). De este modo, la función F_{est} correspondiente en el motor de inducción [40] se describe de la siguiente forma:

$$F_{est}^{mot} = -i_{ds}^{e\ mot} \psi_{dr}'^{e\ mot} \omega_e \quad (3.10)$$

donde las variables con un superíndice (mot) significan las variables reales en el motor de inducción.

En estado estable las siguientes ecuaciones son válidas [40]:

$$i_{ds}^{e\ 2} + i_{qs}^{e\ 2} = i_{ds}^{e(mot)2} + i_{qs}^{e(mot)2} \quad (3.11)$$

$$\frac{i_{qs}^e}{i_{ds}^e} \frac{1}{T_r^{con}} = \frac{i_{qs}^{e\ mot}}{i_{ds}^{e\ mot}} \frac{1}{T_r} \quad (\omega_e = \omega_e^{mot}) \quad (3.12)$$

donde T_r^{con} es la constante de tiempo del rotor en el circuito de control.

De las Ecs. (3.9) – (3.12) se obtiene la siguiente ecuación:

$$\frac{F_{est}}{F_{est}^{mot}} = \frac{\frac{L_M^{con}}{L_M} \left[\left(\frac{T_r^{con}}{T_r} \right)^2 + \left(\frac{i_{qs}^e}{i_{ds}^e} \right)^2 \right]}{\left(\frac{T_r^{con}}{T_r} \right)^2 \left[1 + \left(\frac{i_{qs}^e}{i_{ds}^e} \right)^2 \right]} = \frac{m(k^2 + x^2)}{k^2(1 + x^2)} \quad (3.13)$$

donde

$$m = \frac{L_M^{con}}{L_M} \quad k = \frac{T_r^{con}}{T_r} \quad x = \frac{i_{qs}^e}{i_{ds}^e}$$

La Fig. 3.4 muestra la relación entre el valor de F_{est}/F_{est}^{mot} y x . Como se observa, existe una relación monótona entre estas dos funciones (F_{est}/F_{est}^{mot}), por lo tanto, la función F_{est} es adecuada para la identificación. Las curvas de la Fig. 3.4 se obtuvieron considerando la misma máquina de inducción de 20 HP, 220 V, 4 polos del apartado 2.2.2.

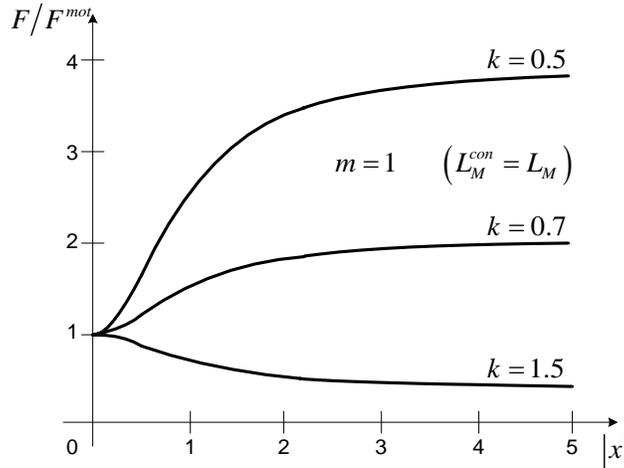


Figura 3.4 Relación entre el valor de F/F^{mot} y x .

Nuevamente de la referencia [40] y de la Tabla 3.1 [32] se utiliza la segunda expresión de la función, llamada F_{act} , para identificar la constante de tiempo del rotor. Esta función se expresa como:

$$F_{act} = \frac{L'_r}{L_M} \left[\left(i_{qs}^e v_{ds}^e - i_{ds}^e v_{qs}^e \right) + \left(i_{qs}^{e2} + i_{ds}^{e2} \right) L_s \sigma \omega_e \right] \quad (3.14)$$

3.5 Estimación de la resistencia del rotor basado en el uso de las funciones F_{est} , F_{act} y en un controlador difuso

3.5.1 Análisis de la señal de entrada al estimador

La estructura del estimador basado en el uso de las funciones F_{est} y F_{act} se realiza de forma similar que el controlador descrito en las referencias [42], [43]. El primer paso en el diseño de este estimador es determinar sus variables de entrada. Dado que la constante de tiempo del rotor en el circuito de control es mayor que la constante mecánica del rotor, debido a variaciones de la resistencia del rotor, el proceso de estimación de la resistencia del rotor puede ejecutarse bajo condiciones de estado estable, lo cual significa que no hay ningún cambio del par de carga y de la velocidad de referencia.

Para estimar la resistencia del rotor, se considera que las variables de flujo no son accesibles a partir de una medida directa, por lo tanto, la función F_{est} [40], [42], [43]

mostrada en la Ec. (3.9) se calcula por medio del flujo del rotor, ψ_{dr}^{e*} , es decir la señal de referencia del flujo (ver Fig. 3.1).

De las consideraciones proporcionadas en el apartado 3.3, al comparar la Ec. (3.9) con la Ec. (3.14), se nota que la función F_{est} difiere de la función F_{act} por el efecto del cambio de la constante de tiempo del rotor, la cual se expresa como $T_r = L'_r / r'_r$, que a su vez se ve afectada por el cambio de la resistencia del rotor, r'_r . De hecho, la resistencia del rotor usada en el bloque de *Campo Orientado* no es el valor real de r'_r a menos que se realice una actualización de la resistencia del rotor. Por lo tanto, para realizar la estimación de la resistencia del rotor, a partir de las funciones F_{est} y F_{act} , se utilizan las Ecs. (3.9) y (3.14) respectivamente, que también son utilizadas en las referencias [40], [42], [43], las cuales se vuelven a escribir:

$$F_{est} = -i_{ds}^{e*} \psi_{dr}^{e*} \omega_e$$

$$F_{act} = \frac{L'_r}{L_M} \left[(i_{qs}^e v_{ds}^e - i_{ds}^e v_{qs}^e) + (i_{qs}^{e2} + i_{ds}^{e2}) L_s \sigma \omega_e \right]$$

La función F_{act} puede calcularse por medio de las corrientes y voltajes del estator, los cuales son obtenidos de las terminales del motor de inducción; así como F_{est} se obtiene de la configuración del control vectorial indirecto.

En la Fig. 3.5 se muestra el método de estimación de la resistencia del rotor, en donde el bloque del controlador de lógica difusa sustituye a un controlador PI [42], [43]. Este controlador consiste en obtener el error entre F_{est} y F_{act} , produciendo la compensación de la resistencia del rotor de referencia en el circuito de control, según el modo de operación del motor de inducción. En esta figura también se muestra que el $\Delta r'_r$ se adiciona a la resistencia del rotor inicial para obtener el valor actualizado de esta resistencia.

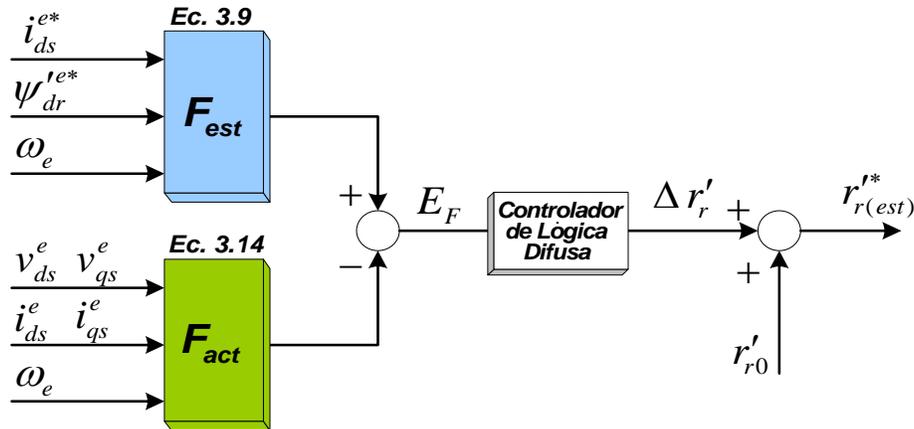


Figura 3.5 Configuración del bloque de estimación de la resistencia del rotor.

3.5.2 Estimación de la resistencia del rotor

En la Fig. 3.6 se muestra la configuración del estimador de la resistencia del rotor, en donde también se observa el interior del bloque del controlador difuso. Asimismo, las funciones F_{est} y F_{act} se calculan a partir de los valores de referencia i_{ds}^{e*} , ψ_{dr}^{e*} y de las variables medidas v_{ds}^e , v_{qs}^e , i_{qs}^e , i_{ds}^e , ω_e . Además se calcula el error, E_F , entre las funciones F_{est} y F_{act} así como el cambio de error, ΔE_F a partir de las siguientes expresiones:

$$E_F(k) = F_{est}(k) - F_{act}(k) \quad (3.15)$$

$$\Delta E_F(k) = E_F(k) - E_F(k-1) \quad (3.16)$$

Por lo tanto, el error, E_F , y el cambio del error, ΔE_F se usan como entradas para el estimador de la resistencia del rotor.

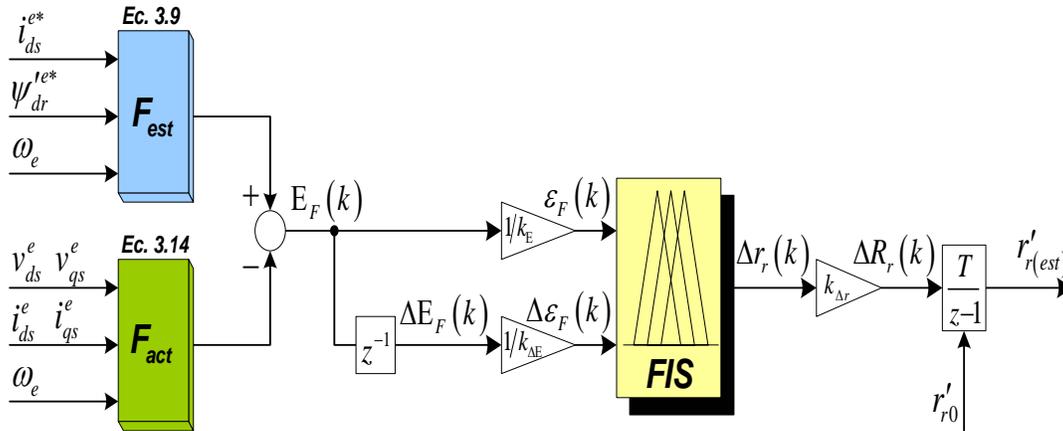


Figura 3.6 Configuración de la estimación de la resistencia del rotor basado en las funciones F_{est} , F_{act} y en un controlador difuso.

De acuerdo a las referencias [42], [43], la estructura interna del estimador de la resistencia del rotor se elige de forma similar al de un controlador de lógica difusa, que consiste de tres fases: fuzzificación, inferencia, y defuzzificación.

Las señales de entrada $\varepsilon_F(k)$ y $\Delta\varepsilon_F(k)$ de la etapa de *fuzzificación* son señales expresadas en por unidad, derivadas de las señales reales $E_F(k)$ y $\Delta E_F(k)$ las cuales se dividen por su respectiva ganancia, k_E y $k_{\Delta E}$, con el objetivo de normalizar los valores de entrada al estimador. Las variables no difusas (reales) $\varepsilon_F(k)$ y $\Delta\varepsilon_F(k)$ se convierten en las variables difusas ε_F y $\Delta\varepsilon_F$, en esta etapa se usan funciones de membresía triangulares, tomadas de la referencias [42], [43].

$\Delta\varepsilon_F$	\mathcal{NG}	\mathcal{NM}	\mathcal{NP}	\mathcal{Z}	\mathcal{PP}	\mathcal{PM}	\mathcal{PG}
ε_F							
\mathcal{NG}	NG	NM	NP	NG	PP	PM	PG
\mathcal{NM}	NG	NM	NP	NM	PP	PM	PG
\mathcal{NP}	NG	NM	NP	NP	PP	PM	PG
\mathcal{Z}	NG	NM	NP	Z	PP	PM	PG
\mathcal{PP}	NG	NM	NP	PP	PP	PM	PG
\mathcal{PM}	NG	NM	NP	PM	PP	PM	PG
\mathcal{PG}	NG	NM	NP	PG	PP	PM	PG

Tabla 3.2. Base de reglas para la estimación de la resistencia del rotor.

En la segunda etapa del estimador, las variables ε_F y $\Delta\varepsilon_F$ son procesados por una máquina de inferencia que ejecuta 49 reglas (7x7) como se muestra en la Tabla 3.2 (obtenidas de la referencia [42] y [43]), dónde \mathcal{NG} , \mathcal{NM} , \mathcal{NP} , \mathcal{Z} , \mathcal{PP} , \mathcal{PM} , \mathcal{PG} corresponden para Negativo Grande, Negativo Mediano, Negativo Pequeño, Cero, Positivo Pequeño, Positivo Mediano, Positivo Grande respectivamente.

En la etapa de defuzzificación, se obtiene el valor no difuso (real) para la variable de salida $\Delta r_r(k)$ en donde se usa el método del centro de gravedad. El valor calculado del incremento de la resistencia $\Delta R_r(k)$ se obtiene multiplicando a $\Delta r_r(k)$ por el factor de ganancia $k_{\Delta r}$. El valor estimado de la resistencia del rotor, $r'_{r(est)}$, se obtiene integrando la señal del incremento de la resistencia $\Delta R_r(k)$, en donde el valor de la resistencia del rotor se toma como el valor inicial para esta integral:

$$r'_{r(est)} = R_r(k-1) + k_{\Delta r} \Delta r_r(k) \quad (3.17)$$

Este valor estimado de r'_r es usado en el cálculo de la velocidad de deslizamiento (Ec. (2.17)) y en el bloque del *Campo Orientado* como se muestra en la Fig. 2.6 para asegurar la operación correcta del control vectorial indirecto.

En la Fig. 3.7 se muestra el sistema general del control vectorial indirecto del motor de inducción, en donde se ilustra el bloque del estimador de la resistencia del rotor. Como se observa, las señales de entrada son las variables de referencia y las variables medidas en las terminales del motor de inducción. La señal de salida del estimador es la resistencia del rotor estimada ($r'_{r(est)}$).

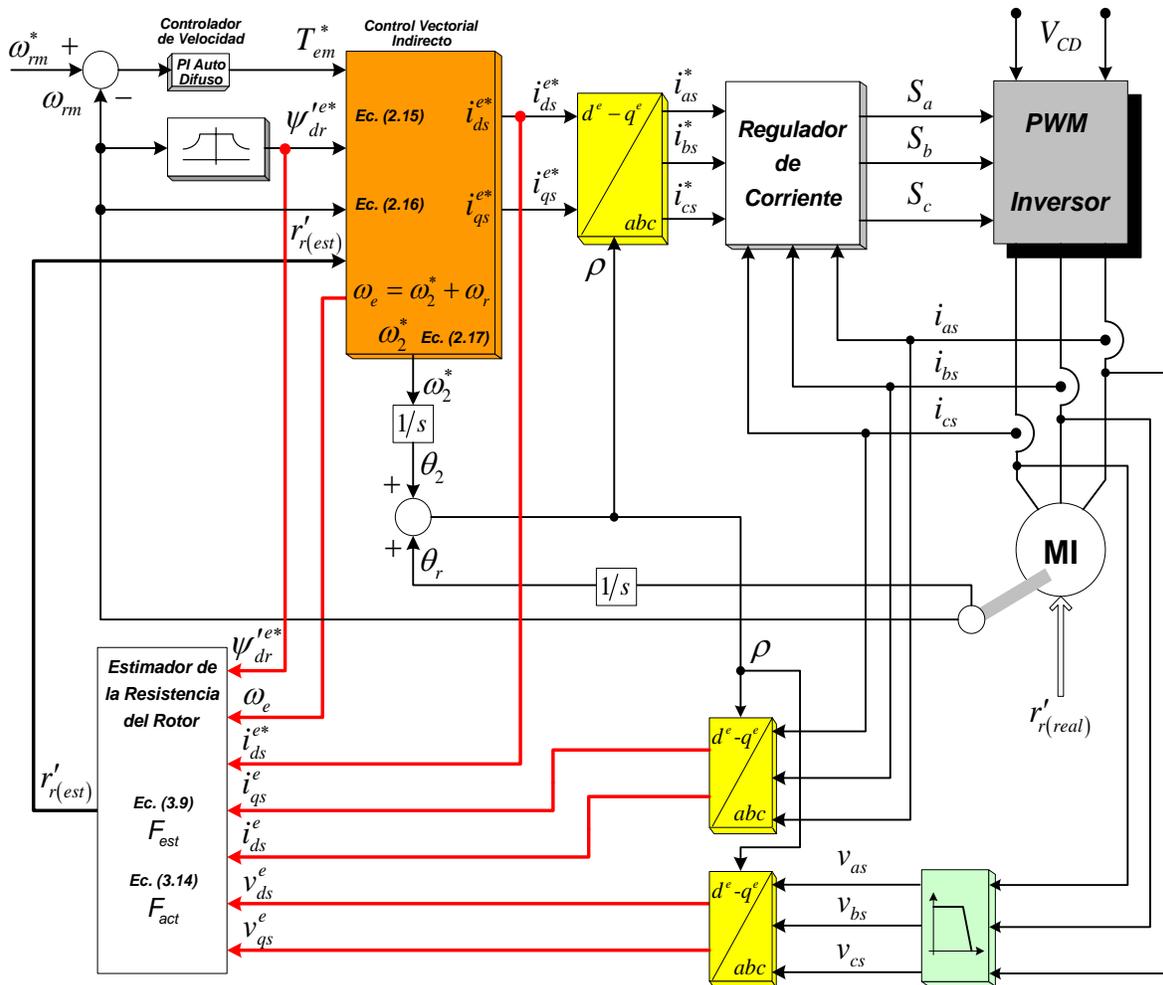


Figura 3.7 Control vectorial indirecto del motor de inducción y configuración del estimador de la resistencia del rotor basado en las funciones F_{est} , F_{act} y en un controlador difuso.

3.6 Estimación de la resistencia del rotor basado en una red neuronal artificial

3.6.1 Introducción

En este apartado se presenta un método para la estimación de la resistencia del rotor sin requerir procesamiento complejo de los datos medidos y en general se utilizan las variables generadas en el control vectorial. Este método es simple y es aplicable a cualquier accionamiento del motor de inducción con control vectorial indirecto. Por lo tanto, para realizar este estimador se utiliza una red neuronal artificial multicapa de realimentación de tres capas, la cual se entrena usando el algoritmo de retropropagación de Levenberg–Marquardt para obtener la estimación de la resistencia del rotor. Este tipo de red es de aprendizaje supervisado, que emplea un ciclo de propagación–adaptación. Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas superiores de la red, hasta generar una salida (resistencia del

rotor). La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas. Las salidas de error se propagan hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida. Sin embargo las neuronas de la capa oculta solo reciben una fracción de la señal total del error, basándose aproximadamente en la contribución relativa que haya aportado cada neurona a la salida original. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total. Basándose en la señal de error percibida, se actualizan los pesos de conexión de cada neurona, para hacer que la red converja hacia un estado que permita clasificar correctamente todos los patrones de entrenamiento (ver Apéndice C).

El objetivo es entrenar una red neuronal artificial a través de los valores de referencia i_{ds}^{e*} , ψ_{dr}^{e*} y de las variables medidas v_{ds}^e , v_{qs}^e , i_{qs}^e , i_{ds}^e , ω_e , que fueron utilizadas en el cálculo de las funciones F_{est} y F_{act} . Al realizar este entrenamiento se consigue un estimador neuronal que tiene la capacidad de estimar la resistencia del rotor para cualquier variación de su valor, ya sea por efecto de la temperatura del rotor y para cuando se presentan cambios del par de carga y de la velocidad mecánica del rotor, durante la operación del motor. La decisión de realizar al estimador neuronal y sustituirlo por el estimador descrito en 3.5 es para obtener un mejor desempeño en estado transitorio, como se mostrara en el siguiente capítulo.

3.6.2 Descripción general de la red neuronal artificial

Una red neuronal artificial es una malla de nodos y conexiones altamente interconectados [48]. En la Fig. 3.8 se muestra una red neuronal de realimentación, donde el flujo de la señal es unidireccional de la entrada a la salida, a través de conjuntos de neuronas agrupadas en capas.

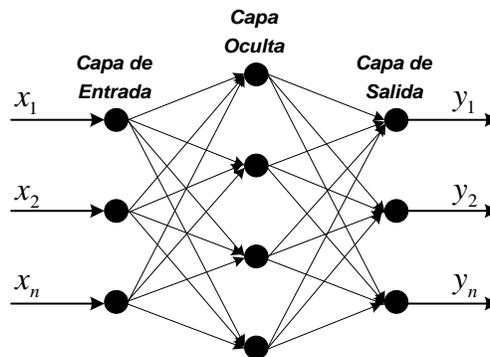


Figura 3.8. Arquitectura de una red neuronal de realimentación.

El número de capas es variable, desde un mínimo de dos, es decir solo capas de entrada y salida, y crecerá dependiendo del número de capas ocultas. En general, es diferente el número de neuronas por capas. En la entrada, habrán tantas neuronas como entradas tenga el sistema a modelar. El número de neuronas en las capas ocultas no está explícitamente determinado y varía dependiendo del problema, no conociéndose formulación precisa para obtenerlo, aunque algunos autores han trabajado en la generación

de algunos métodos para establecer esta cifra [51]. Cada nodo se conoce como una neurona y representa la unidad básica de procesamiento, las líneas de interconexión se conocen como sinapsis en analogía a la topología del cerebro humano y tienen asociado un factor de peso (W_{ji}). Una neurona se comporta como una función traduciendo una entrada (x_i) generalmente normalizada (entre 0 a 1 o -1 a 1) en una señal de salida (y_j).

El procesamiento de la señal de una neurona se ilustra en la Fig. 3.9. En donde se observa que las entradas x_i ponderadas a través de los correspondientes pesos sinápticos W_{ji} se suman, como se muestra en la siguiente ecuación:

$$y_j = f \left(\sum_{i=1}^n W_{ji} x_i + \theta_j \right) \quad (3.18)$$

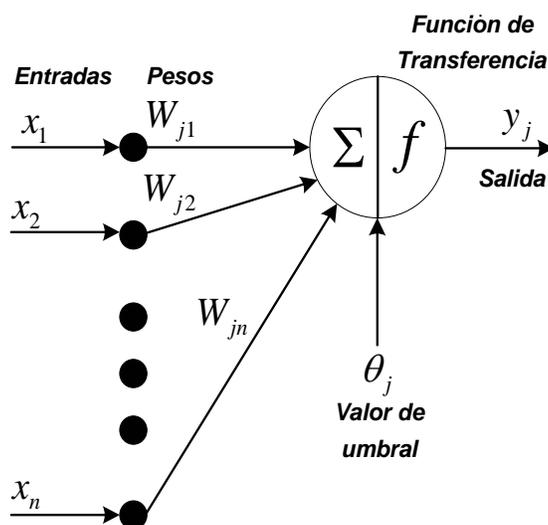


Figura 3.9 Procesamiento en una neurona.

A la sumatoria de $W_{ji}x_i$ se aplica la función de transferencia, entre las más utilizadas están la función escalón, tangente sigmoideal, senoidal y logarítmica. En forma general, la función de transferencia es no lineal, la cual asociada a las neuronas provee a la red la propiedad de transformación no lineal, lo que permite ejecutar cálculos altamente no lineales.

La red neuronal artificial se entrena por un algoritmo de aprendizaje, el cual realiza la adaptación de los pesos de la red de forma iterativa hasta que el error entre el vector objetivo y la salida de la red neuronal es menor que un error establecido. El algoritmo de aprendizaje más popular para las redes multicapa es el algoritmo de retropropagación y sus variantes [52]. Una descripción más detallada sobre la red neuronal se describe en el Apéndice C.

3.6.3 Estructura y normalización de la red neuronal artificial

La red neuronal artificial se requiere para sustituir al estimador de la resistencia del rotor basado en las funciones F_{est} , F_{act} y en un controlador difuso. Solamente se utilizan los valores de referencia i_{ds}^{e*} , ψ_{dr}^{e*} y las variables medidas v_{ds}^e , v_{qs}^e , i_{qs}^e , i_{ds}^e , ω_e , estos valores y variables son el patrón de entrada de la red neuronal artificial. De esta manera el patrón de salida es la resistencia del rotor obtenida en el estimador descrito en 3.5, la cual fue finalmente calculada en la Ec. (3.17). Para realizar adecuadamente la estimación, es necesario proporcionar muestreos anteriores del patrón de entrada, además de los valores de muestreo presentes. La Fig. 3.10, muestra en forma general el patrón de entradas y salidas de la red neuronal artificial, dónde k denota el intervalo de muestreo presente y $(k-i)$ denota la medición de retardo en los intervalos de i -muestreo.

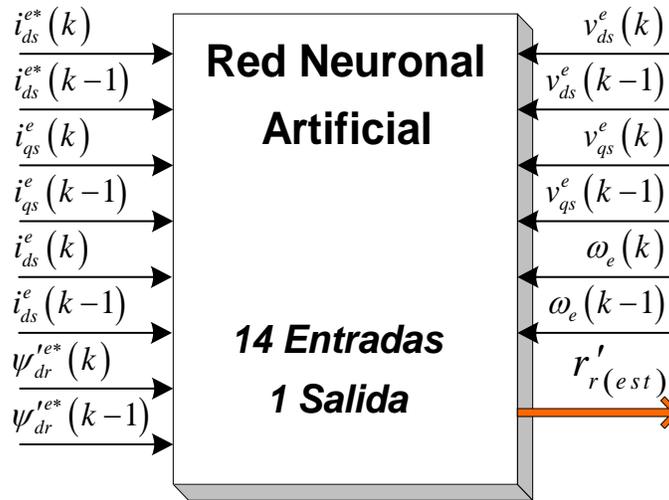


Figura 3.10 Diagrama general del patrón de entradas y salidas del estimador neuronal de la resistencia del rotor.

Así, la red neuronal artificial tiene 14 entradas y 1 salida. Falta definir el número de capas y número de neuronas en cada capa, lo cual dependerá de los resultados que se obtengan en el entrenamiento.

Cualquier cambio que se realice en los patrones de entrenamiento exige una codificación diferente del vector de entrada y a su vez cambian las condiciones generales de la red, pero el proceso de entrenamiento sigue siendo igual. En esta aplicación se utilizó la configuración de la Fig. 3.10, en donde se muestra un diagrama descriptivo de las entradas y salidas de la red neuronal artificial, con el objetivo de obtener la resistencia del rotor de acuerdo a su capacidad para predecir comportamientos futuros de patrones nunca antes presentados en la etapa de aprendizaje [48].

Los datos de entrada a la red neuronal artificial deben normalizarse entre $[-1, 1]$. En general, en la mayoría de los accionamientos de CA alimentados con inversor, la corriente de línea se limita a dos veces la corriente nominal. Para el marco de referencia d^e , q^e , la

corriente base usada para la normalización es de dos veces el valor máximo de la corriente de línea trifásica. El valor máximo del voltaje base se usa para normalizar los voltajes medidos y la velocidad se normaliza como un porcentaje de su valor nominal. Para evitar problemas con la saturación de la neurona de salida, los datos se normalizan en el rango de [0, 1]. [53]

3.6.4 Generación de datos y entrenamiento de la red

Para la generación de los datos de entrenamiento de la red neuronal artificial se realizará la simulación del control vectorial indirecto del motor de inducción con el estimador de la resistencia del rotor basado en un controlador de lógica difusa. Con los datos obtenidos de la simulación se realizará el entrenamiento de la red neuronal artificial la cual sustituirá al estimador descrito en 3.5. De esta manera se genera al estimador neuronal de la resistencia del rotor, el cual actuará junto con el control vectorial indirecto del motor de inducción.

En la Fig. 3.11 se muestra la forma en que se entrenará la red neuronal artificial por medio del algoritmo de entrenamiento de retropropagación. Dicha red será entrenada usando los patrones de datos generados para condiciones de operación en estado estable, es decir, el motor se accionará con el control vectorial indirecto y se considera al esquemas de regulación de corriente síncrono descrito en 2.4, de esta manera el motor tomará su velocidad mecánica del rotor y su par de carga nominal, y no será necesario obtener un patrón de datos para diferentes condiciones de operación. Sin embargo, se considerará la condición transitoria que se presenta durante el arranque del motor de inducción aunque después siga operando en estado estable, ya que la estimación se realizará desde el arranque.

Como se describió en 3.1, con variaciones en la resistencia del rotor, el funcionamiento del control vectorial indirecto no se realiza adecuadamente. Por lo tanto, la red neuronal se entrenará usando un conjunto de datos de forma aleatoria para tres diferentes valores de la resistencia del rotor cubriendo el 100% del rango de variación.

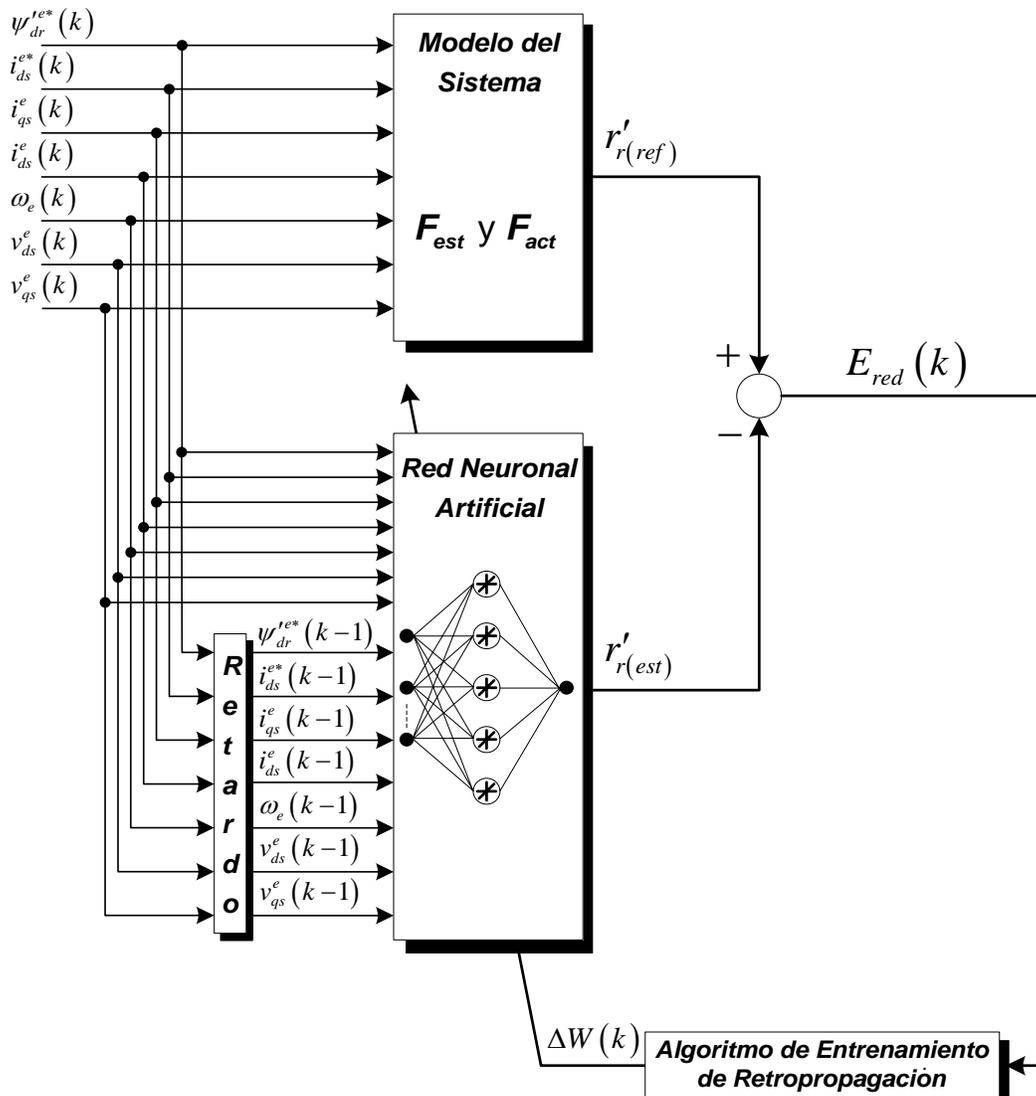


Figura 3.11 Fase de entrenamiento del estimador neuronal de la resistencia del rotor.

A partir de lo descrito y de las consideraciones tomadas, el procedimiento de entrenamiento de retropropagación usado para la red neuronal artificial se lleva a cabo como sigue:

1. Se realiza la simulación del sistema del accionamiento del motor de inducción como se muestra en la Fig. 3.7, usando el software de MATLAB-SIMULINK y se generan los datos de Entrada/Salida para condiciones de operación con velocidad mecánica nominal y con par de carga nominal. En el esquema propuesto se deben considerar diferentes valores de la resistencia del rotor cubriendo un rango del 100% de variación.

2. Los datos de entrada a la red neuronal son: los valores estimados del control, i_{ds}^{e*} , ψ_{dr}^{te*} y los valores medidos v_{ds}^e , v_{qs}^e , i_{qs}^e , i_{ds}^e , ω_e , y sus respectivas señales de retardo. Esto suma 14 entradas, teniendo como salida a la resistencia del rotor.
3. Se normalizan los datos de entrenamiento, dividiéndolos por sus valores máximos. Normalmente las funciones de activación de la red se limitan entre [-1, 1]. De esta forma, si el dato no se normaliza, ocurrirá una saturación en la salida de la función de transferencia.
4. Se cargan los datos de las Entrada/Salida en los archivos de datos y se suministran a la red neuronal, con el programa escrito en MATLAB según el algoritmo de retropropagación.
5. El programa de MATLAB asigna inicialmente pesos pequeños y aleatorios a la red.
6. Se selecciona un patrón de Entrada/Salida en forma aleatoria de los archivos de datos. Para un patrón de datos de entrada dados, la red calcula las salidas y los compara con los datos de salida deseados para obtener el patrón del error, $E_{red}(k)$.
7. Del patrón del error, $E_{red}(k)$, se calculan y se reajustan los pesos de la red por el algoritmo de retropropagación para que el nuevo error sea menor que el anterior. En este algoritmo se usan los diferentes valores del coeficiente de aprendizaje η y el término de momentum α .
8. Se repiten los pasos anteriores para cada conjunto de patrones de datos de Entrada/Salida, hasta que la raíz cuadrática media (*rms*) del error converja por debajo del valor de umbral deseado, de acuerdo al error del entrenamiento establecido.
9. Después de completar el entrenamiento, se prueba el desempeño de la red con los diferentes patrones de entrada para asegurar el entrenamiento exitoso y la generalización de la red.

En la Fig. 3.11 se muestra una representación esquemática de este algoritmo. Para encontrar el error, $E_{red}(k)$, que se usa en el algoritmo de retropropagación, a la salida de la red neuronal, se comparan la resistencia del rotor de referencia, $r'_{r(ref)}$, con la resistencia del rotor calculada, $r'_{r(est)}$, para actualizar los pesos de la red neuronal, de manera que el error se minimice. Esto se repite hasta que el error entre ambas resistencias sea igual que el error especificado.

Finalmente en la Fig. 3.12 se muestra el sistema general del control vectorial indirecto del motor de inducción con el estimador neuronal de la resistencia del rotor, en donde se observa que la configuración no cambia y los datos se siguen obteniendo de la misma manera que para el estimador descrito en 3.5.

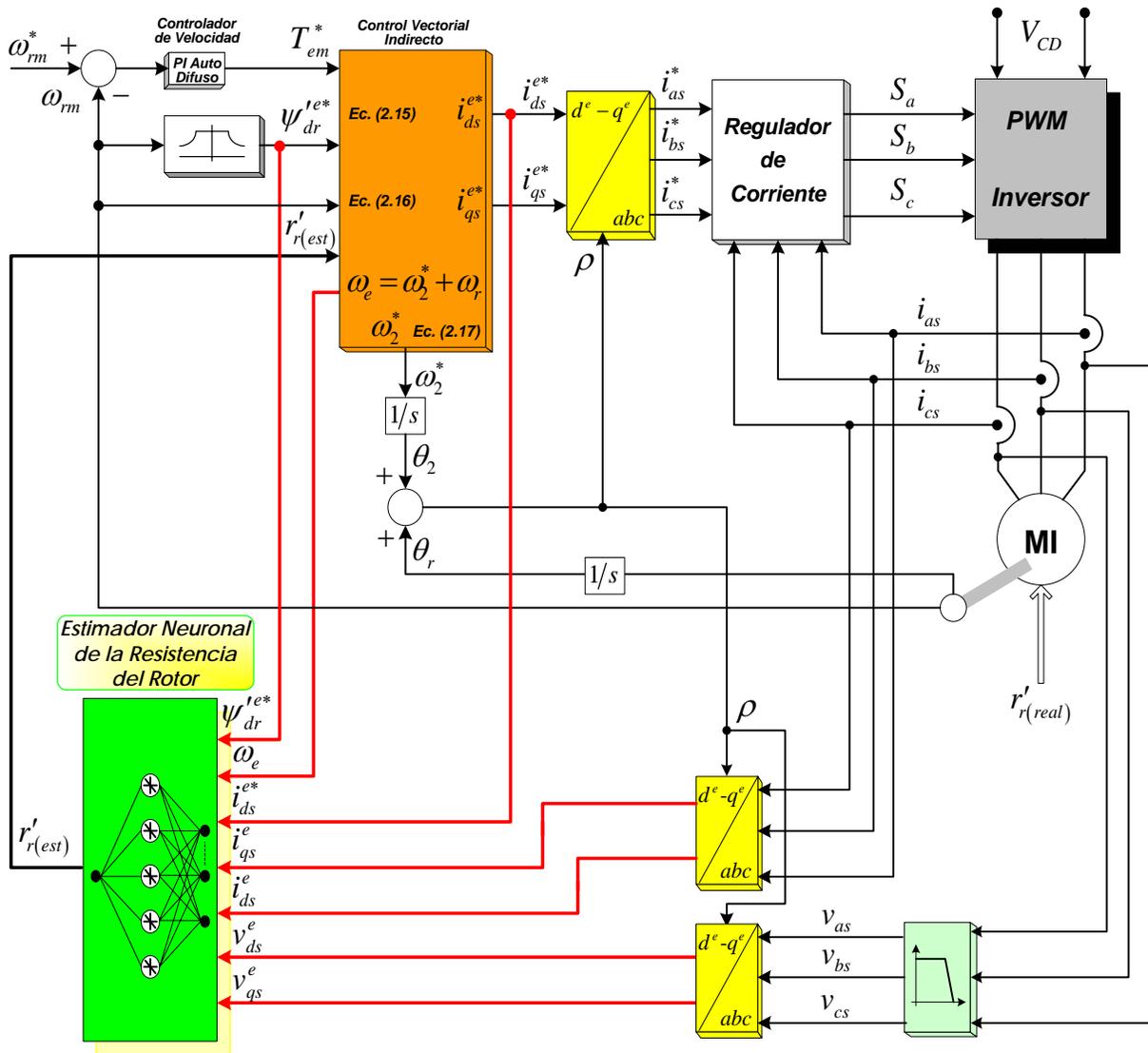


Figura 3.12. Control vectorial indirecto del motor de inducción con el estimador neuronal de la resistencia del rotor.

Capítulo 4

RESULTADOS DE LA SIMULACIÓN

4.1 Introducción

En el capítulo anterior, se presentaron dos métodos de estimación de la resistencia del rotor en el control vectorial indirecto del motor de inducción, donde se mencionó que la estimación de la resistencia del rotor influye en el desempeño del sistema de control. Este capítulo se ha reservado para presentar el comportamiento real del control vectorial indirecto del motor de inducción con el regulador de corriente síncrono, al momento de realizar la estimación de la resistencia del rotor. Estos estimadores se basaron en: (a) las funciones F_{est} , F_{act} y en un controlador difuso y (b) en una red neuronal artificial.

El objetivo de este capítulo es analizar el desempeño estático y dinámico del estimador con un controlador difuso y el estimador basado en una red neuronal artificial, la cual fue entrenada utilizando como patrón al estimador con un controlador difuso (ver Capítulo 3), los resultados se obtuvieron usando el paquete de programación de MATLAB-SIMULINK.

Se realizó la simulación del estimador con las funciones F_{est} , F_{act} y un controlador difuso lográndose resultados considerando el estado estable (velocidad mecánica del rotor y par de carga nominales) y el estado transitorio (velocidad mecánica del rotor y/o par de carga variables) del motor de inducción de 20 HP, en donde se generaron las variables del sistema de control, así como la estimación de la resistencia del rotor.

Posteriormente, se realizó el entrenamiento de la red neuronal artificial considerando diferentes números de capas ocultas y de neuronas en dichas capas, con la finalidad de seleccionar a la configuración de la red neuronal artificial más adecuada para utilizarla en el bloque del estimador neuronal. Con este estimador se realizó la simulación en estado estable y transitorio de los motores de inducción de 20, 100, 1 y 5 HP.

En ambas simulaciones de los estimadores, se consideró que la resistencia del rotor se mantenía sin cambios y en otros casos esta resistencia variaba en diferentes puntos de operación del motor de inducción.

La finalidad de estas simulaciones es mostrar el mejor desempeño del estimador neuronal frente al estimador con un controlador difuso. Así mismo, para tener un punto de comparación se simuló el control vectorial indirecto del motor de inducción sin realizar la estimación de la resistencia del rotor.

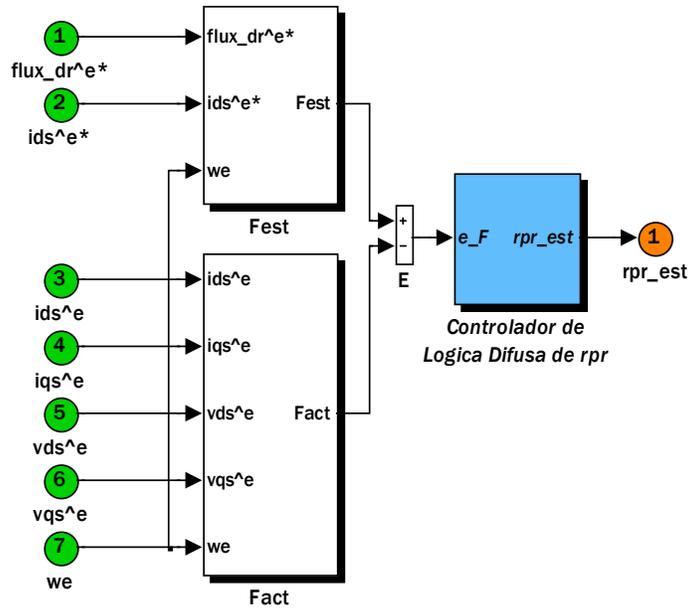
4.2 Simulación de la estimación de la resistencia del rotor basado en las funciones F_{est} , F_{act} y en un controlador difuso

A continuación se mostrará la simulación del control vectorial indirecto del motor de inducción con PWM y regulador de corriente síncrono, realizándose la estimación de la resistencia del rotor por medio del estimador basado en un controlador difuso. Se utiliza una máquina de inducción de 20 HP, 220 V, 4 polos para realizar la simulación (apartado 2.2.2).

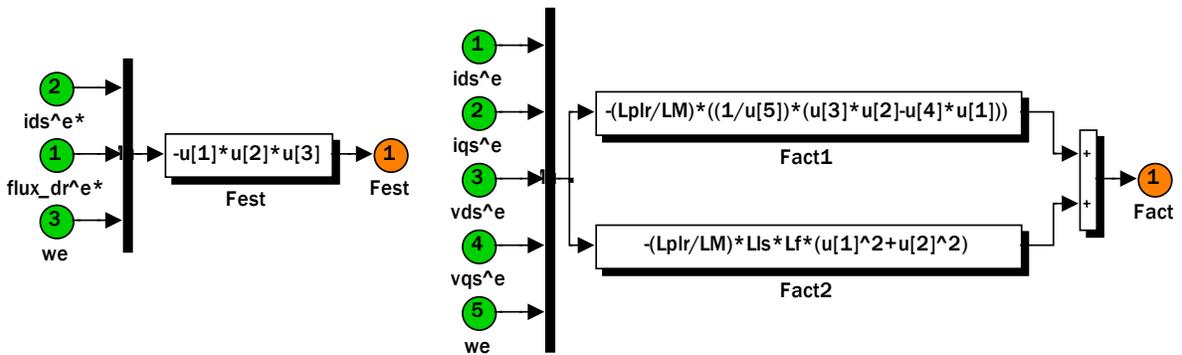
A partir de la Fig. 3.7, se realiza la simulación en SIMULINK del sistema general del control vectorial indirecto del motor de inducción con PWM y regulador de corriente síncrono, como se muestra en la Fig. 4.1(a), en la cual se realiza la estimación y la actualización de la resistencia del rotor.

El bloque del estimador de la resistencia del rotor (Fig. 4.1(b)), utiliza como señales de entrada al flujo del rotor en el eje d , Ψ_{dr}^{e*} , la cual se obtiene del bloque de campo debilitado; a la corriente bifásica síncrona de referencia, i_{ds}^{e*} , y la frecuencia del estator, ω_e , que provienen del bloque de campo orientado; y al voltaje bifásico síncrono, v_{ds}^e , v_{qs}^e , así como a la corriente bifásica síncrona i_{ds}^e , i_{qs}^e , que son obtenidos de los bloques $vqds2qde$ e $iqds2qde$ respectivamente, cuyos voltajes y corrientes se toman de las terminales del motor de inducción como se muestra en la Fig. 4.1(a). En la Fig. 4.1(g) se muestra el interior del bloque $vqds2qde$ y en la Fig. 4.1(h) el interior del bloque $iqds2qde$.

La Fig. 4.1(b) muestra el bloque del *estimador de la resistencia del rotor*. Este estimador se realiza a partir de la Fig. 3.6, en donde la Ec. (3.9) determina a la función estimada, F_{est} (Fig. 4.1(c)), y la Ec. (3.14) a la función real, F_{act} (Fig. 4.1(d)), las cuales se usan para generar una señal de error que es utilizada por el estimador (Fig. 4.1(e)), cuyo interior se ve en la Fig. 4.1(f). El estimador convierte el error entre las funciones a un incremento de la resistencia del rotor, $\Delta R_r(k)$, cuyo valor estimado de la resistencia del rotor se obtiene entonces integrando la señal $\Delta R_r(k)$ tomando como valor inicial a la resistencia del rotor, la cual es $r_r' = 0.0764 \Omega$. Este valor estimado de $r_r'(est)$ se usa en el cálculo del deslizamiento (Ec. 3.3) y de la corriente i_{ds}^{e*} (Ec. 3.1) la cual se utiliza en el bloque de campo orientado.

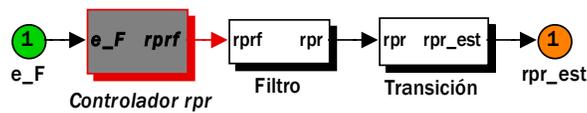


(b) Bloque del estimador de la resistencia del rotor.

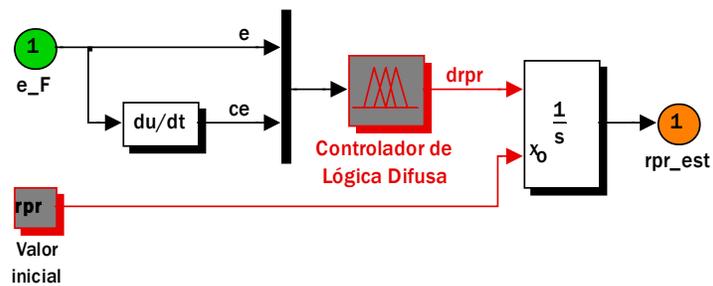


(c) Función estimada, F_{est} .

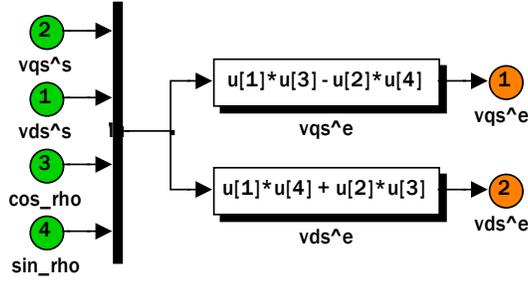
(d) Función real, F_{act} .



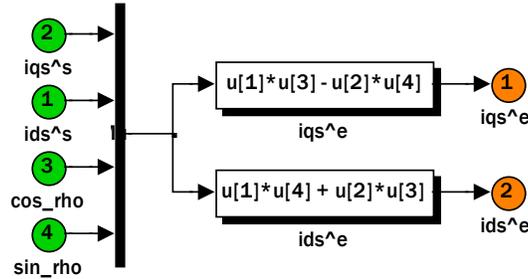
(e) Interior del bloque FLE_Rr de la Fig. 4.1(b).



(f) Estimador con un controlador difuso de la resistencia del rotor.



(g) Voltajes bifásicos síncronos, v_{ds}^e , v_{qs}^e .



(h) Corrientes bifásicas síncronas, i_{ds}^e , i_{qs}^e .

Figura. 4.1. Diagrama de bloques del programa en SIMULINK utilizado para la simulación del control vectorial indirecto del motor de inducción con PWM y regulador de corriente síncrono, en el cual se realiza la estimación y actualización de la resistencia del rotor usando un estimador basado en las funciones F_{est} , F_{act} y en un controlador difuso.

4.3 Simulación del estimador neuronal de la resistencia del rotor en el control vectorial indirecto con regulador de corriente síncrono

4.3.1 Capa de entradas

Los valores de entrada a la red neuronal artificial se agruparon en el vector de catorce entradas, x . Como se describió en 3.5.1, estas entradas son los valores de referencia obtenidos del bloque de campo orientado: i_{qs}^{e*} , ψ_{dr}^{e*} , ω_e y de las variables medidas en las terminales del motor de inducción: v_{ds}^e , v_{qs}^e , i_{ds}^e , i_{qs}^e , asimismo es necesario proporcionar valores de *un muestreo* anterior de las cantidades del patrón de entrada, además de los valores de muestreo presentes (Fig. 3.8). Se sigue utilizando la máquina de inducción de 20 HP, 220 V, 4 polos, utilizada en 4.2. Para cada una de las entradas se generó un vector columna, del cual se obtuvo un vector fila por medio de la transpuesta y posteriormente se obtuvo el vector fila de retardo, lo anterior se muestra a continuación (Tabla 4.1) en donde se emplea la codificación para el archivo *m3.m* de MATLAB:

Entrada	Código
ψ_{dr}^{e*}	ψ_{dr} ; $\psi_{dr1} = \psi_{dr}'$; $\psi_{dr2} = \text{delaysig}(\psi_{dr1}, 1, 1)$;
i_{ds}^{e*}	ids_ref ; $ids1_ref = ids_ref'$; $ids2_ref = \text{delaysig}(ids1_ref, 1, 1)$;
i_{qs}^e	iqs ; $iqs1 = iqs'$; $iqs2 = \text{delaysig}(iqs1, 1, 1)$;
i_{ds}^e	ids ; $ids1 = ids'$; $ids2 = \text{delaysig}(ids1, 1, 1)$;
v_{ds}^e	vds ; $vds1 = vds'$; $vds2 = \text{delaysig}(vds1, 1, 1)$;
v_{qs}^e	vqs ; $vqs1 = vqs'$; $vqs2 = \text{delaysig}(vqs1, 1, 1)$;
ω_e	we ; $we1 = we'$; $we2 = \text{delaysig}(we1, 1, 1)$;

Tabla 4.1 Generación de las entradas de la red neuronal artificial.

Finalmente el vector de entradas \mathbf{x} (Tabla 4.2) se genera de la siguiente manera:

Código
$\mathbf{x} = [\psi_{dr1}; \psi_{dr2}; ids1_ref; ids2_ref; iqs1; iqs2; ids1; ids2; vds1; vds2; vqs1; vqs2; we1; we2]$;

Tabla 4.2 Vector de entradas.

La red neuronal artificial se entrenó con 2500 valores en cada una de las entradas, cuyos valores fueron normalizados; el proceso de normalización consiste en dividir cada uno de los valores de las entradas por su valor máximo, de tal forma que el mayor valor de entrada a la red este en el rango de [-1 1]:

$$\psi_{dr(\text{normalizado})}^{e*} = \frac{\text{valor del flujo}}{\psi_{dr(\text{max})}^{e*}} \quad (4.1)$$

$$i_{ds(\text{normalizado})}^{e*} = \frac{\text{valor de la corriente en el eje d}}{i_{ds(\text{max})}^{e*}} \quad (4.2)$$

$$i_{qs(normalizado)}^e = \frac{\text{valor de la corriente en el eje q}}{i_{qs(max)}^e} \quad (4.3)$$

$$i_{ds(normalizado)}^e = \frac{\text{valor de la corriente en el eje d}}{i_{ds(max)}^e} \quad (4.4)$$

$$v_{ds(normalizado)}^e = \frac{\text{valor del voltaje en el eje d}}{v_{ds(max)}^e} \quad (4.5)$$

$$v_{qs(normalizado)}^e = \frac{\text{valor del voltaje en el eje q}}{v_{qs(max)}^e} \quad (4.6)$$

$$\omega_{e(normalizado)} = \frac{\text{valor de la velocidad}}{\omega_{e(max)}} \quad (4.7)$$

4.3.2 Capa de salida

El valor esperado (Tabla 4.3), es decir el valor de la resistencia del rotor obtenido en el estimador descrito en 4.2, es representado por *rpr_ref*:

Salida	Código
r_r^{r*}	<i>rpr_ref</i> ; <i>rpr_ref1 = rpr_ref</i> ;

Tabla 4.3 Vector de salidas.

Para evitar problemas con la saturación de la neurona de salida, los datos de la resistencia del rotor se normalizan como en el caso de las entradas, para que el valor máximo que proporcione la salida este en el rango de [0 1].

$$r_{r(normalizado)}^{r*} = \frac{\text{valor de la resistencia del rotor}}{r_{r(max)}^r} \quad (4.8)$$

En el denominador de las relaciones anteriores, se consideran los valores de las variables y del parámetro durante la simulación del esquema mostrado en la Fig. 4.1.

4.3.3 Entrenamiento de la red neuronal artificial

El código de entrenamiento para esta red (*m3.m*, ver Apéndice A), es desarrollado con base en la herramienta de redes neuronales artificiales de MATLAB. La red es creada mediante el comando *newff* para creación de redes multicapa de realimentación conocida como de Retropropagación.

4.3.3.1 Entrenamiento con 1 capa oculta con 3 neuronas

La red neuronal artificial creada tiene las siguientes características (Tabla 4.4):

Código
<code>net=newff([minmax(x)],[3 1],{'tansig','purelin'},'trainlm');</code>

Tabla 4.4 Creación de una red con 1 capa oculta.

Por lo tanto, se crea una red de tres capas. En la primera capa se tiene un vector de entradas con 14 elementos, en la segunda se tiene 1 capa oculta con 3 neuronas y en la capa de salida solamente hay 1 neurona. La función de transferencia en la capa oculta es la tan-sigmoidal y en la capa de salida la función de transferencia es lineal. El algoritmo de entrenamiento es `trainlm` (Levenberg–Marquardt), el cual se describe en el Apéndice C.

A partir de las siguientes instrucciones (Tabla 4.5) se obtiene el entrenamiento de la red para 4000 iteraciones, la cual se simula con los valores obtenidos en el entrenamiento y se realiza la gráfica de la salida de la red.

Código
<code>net = train(net, x, r_dif1)</code>
<code>y = sim(net, x)</code>
<code>plot(x, r_dif1, x, y, 'o')</code>

Tabla 4.5 Entrenamiento de la red neuronal artificial.

Los valores de iniciación de la matriz de pesos se generaron de manera aleatoria; después de varias pruebas, los parámetros que determinan el entrenamiento se fijaron en los siguientes valores (Tabla 4.6):

Código
<code>net.trainParam.show = 50;</code>
<code>net.trainParam.lr = 0.01;</code>
<code>net.trainParam.epochs = 4000;</code>
<code>net.trainParam.goal = 0;</code>
<code>net.trainParam.max_fail = 5;</code>
<code>net.trainParam.mem_reduc = 1;</code>
<code>net.trainParam.min_grad = 1e-12;</code>
<code>net.trainParam.time = inf;</code>

Tabla 4.6 Parámetros para realizar el entrenamiento.

4.3.3.2 Entrenamiento con 2 capas ocultas

Se sigue el mismo procedimiento que en el caso anterior, en donde se utiliza el mismo algoritmo de entrenamiento `trainlm` y se crea una red neuronal artificial (Tabla 4.7), cuyas características se muestran en la Tabla 4.8.

Código	
<code>net=newff([minmax(x)],[5 5 1],{'tansig','tansig','purelin'},'trainlm');</code>	

Tabla 4.7 Creación de una red con 2 capas ocultas.

No capa	Tipo	No neuronas	Función de transferencia
1	Entradas	14	
2	Ocultas	5	tan-sigmoidal
3	Ocultas	5	tan-sigmoidal
4	Salida	1	lineal

Tabla 4.8 Características de la red con 2 capas ocultas.

4.3.3.3 Entrenamiento con 1 capa oculta con 5 neuronas

Al igual que en los casos anteriores, se sigue el mismo procedimiento, en donde se utiliza el mismo algoritmo de entrenamiento `trainlm` y se crea una red neuronal artificial (Tabla 4.9), cuyas características se muestran en la Tabla 4.10.

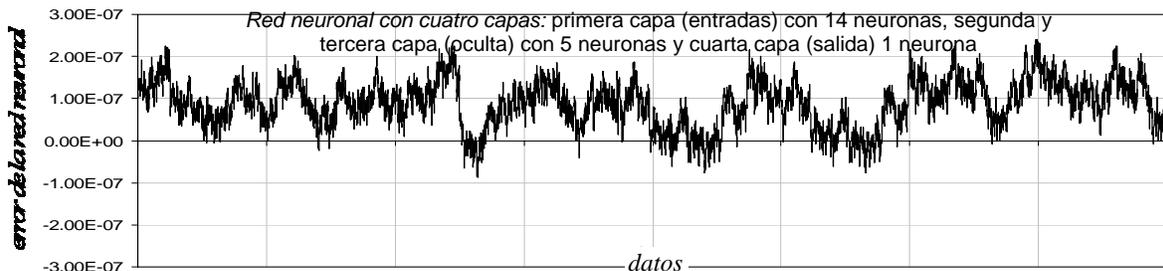
Código	
<code>net=newff([minmax(x)],[5 5 1],{'tansig','tansig','purelin'},'trainlm');</code>	

Tabla 4.9 Creación de una red con 2 capas ocultas.

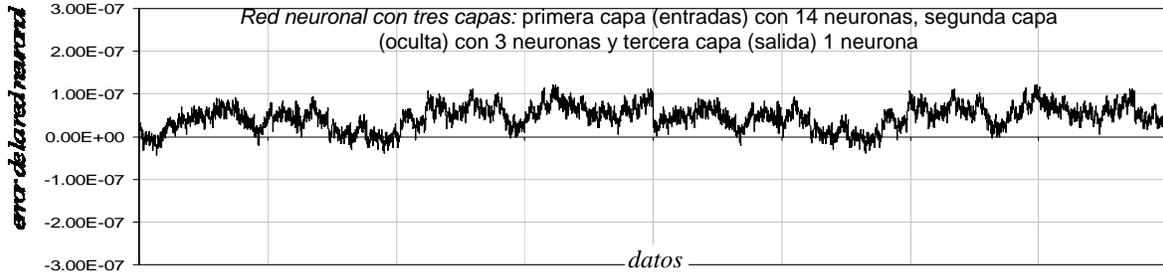
No. capa	Tipo	No. neuronas	Función de transferencia
1	Entradas	14	
2	Ocultas	5	tan-sigmoidal
3	Salida	1	lineal

Tabla 4.10 Características de la red con 1 capa oculta.

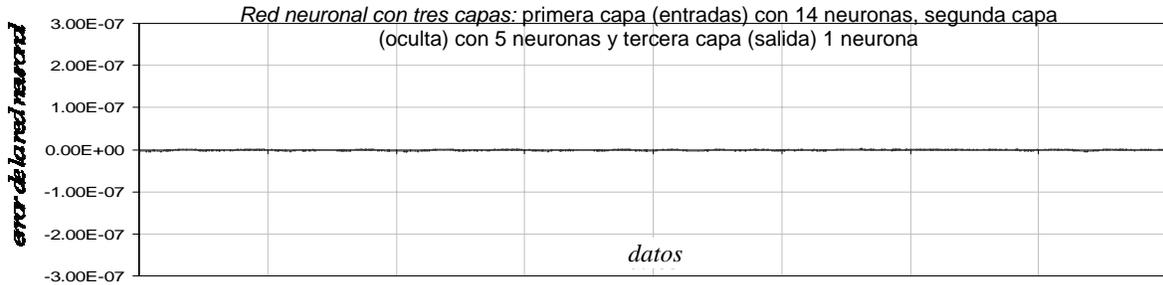
Después de crear las redes anteriores, y de haber obtenido el error entre la resistencia del rotor de referencia y la resistencia del rotor que proporciona la red neuronal artificial durante el entrenamiento, se grafican los errores de las tres redes (Fig. 4.2), en donde se observa que el error en la red neuronal artificial descrita en la Tabla 4.10 (Fig. 4.2(c)) es inferior que las redes descritas en la Tabla 4.8 (Fig. 4.2(a)) y en 4.3.3.1 (Fig. 4.2 (b)).



(a) Error típico = 2.449×10^{-7}



(b) Error típico = 1.134×10^{-7}



(c) Error típico = 2.63541×10^{-9}

Figura. 4.2. Error entre la resistencia del rotor de referencia y la resistencia del rotor que proporciona la red neuronal artificial.

De acuerdo a los resultados proporcionados por la red neuronal artificial de tres capas, es decir la que tiene en la primera capa un vector de entradas con 14 elementos, en la segunda capa 5 neuronas y en la capa de salida solamente 1 neurona, se obtiene la configuración de la red (Fig. 4.3) por medio de la siguiente instrucción (Tabla 4.11):

<i>Código</i>
<i>gensim(net, 1)</i>

Tabla 4.11 Obtención de la red neuronal en SIMULINK después del entrenamiento.

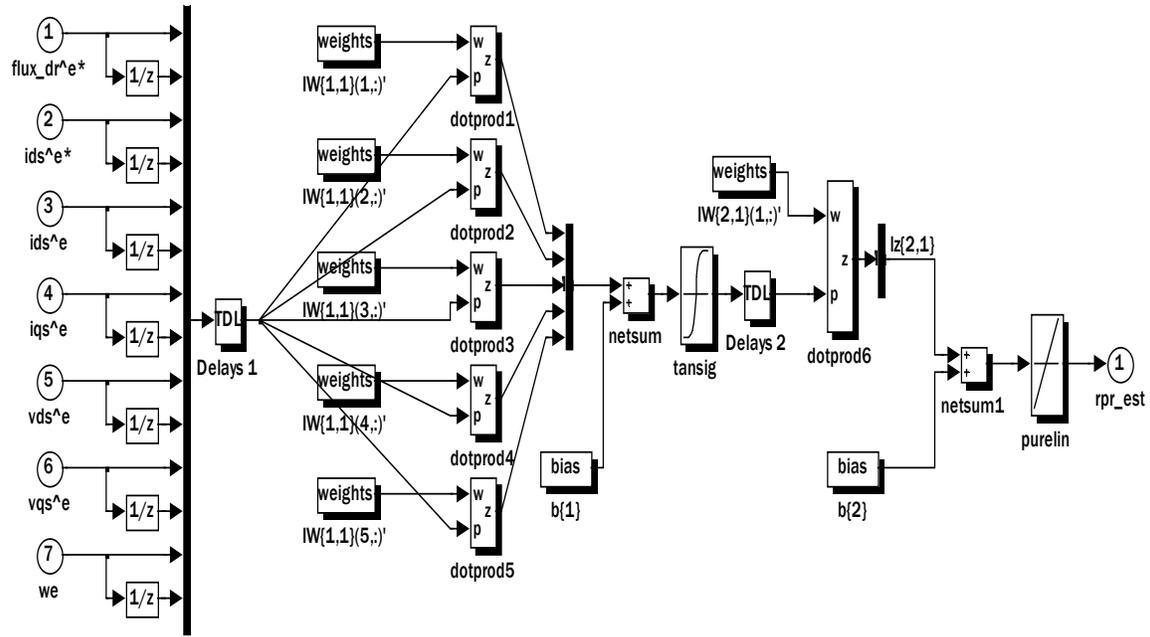


Figura 4.3. Configuración de la red neuronal artificial para la estimación de la resistencia del rotor.

Asimismo, los parámetros más importantes de la red neuronal artificial entrenada (Fig. 4.3) se presentan en la Tabla 4.12.

	$W1=net.IW\{1,1\}$					
	1	2	3	4	5	6
$\mathcal{N}1$	-4.998249e-4	2.976732	-2.341576e-2	-6.045707e-2	-2.0576341e-2	-7.570407e-2
$\mathcal{N}2$	-5.994826e-4	-2.439296	-3.474525e-2	1.163915e-2	-3.2745245e-2	4.391615e-2
$\mathcal{N}3$	-8.792362e-5	-0.780268	-2.107064e-2	2.499103e-3	-2.6070614e-2	2.103499e-3
$\mathcal{N}4$	7.843468e-4	4.916252	5.082147e-2	-1.494302e-2	5.5021487e-2	-4.340492e-2
$\mathcal{N}5$	-1.437146e-3	2.141252	-9.286891e-2	-0.145953	-9.8868921e-2	-1.951453
	$W1=net.IW\{1,1\}$					
	7	8	9	10	11	12
$\mathcal{N}1$	3.237735e-6	-1.419993e-5	-2.404466e-6	3.624657e-4	-2.804127e-5	9.873704e-5
$\mathcal{N}2$	-2.583264e-3	-1.729591e-3	-2.011230e-4	-5.589837e-2	5.313939e-3	-1.621455e-2
$\mathcal{N}3$	-2.577876e-3	-1.721725e-3	-1.999790e-4	-5.585683e-2	5.3082814e-3	-1.620079e-2
$\mathcal{N}4$	1.260627e-4	1.568888e-3	2.390726e-4	-2.410727e-2	1.722115e-3	-6.425117e-3
$\mathcal{N}5$	8.127052e-3	4.900219e-2	-7.404489e-2	-0.180426	-3.631721e-2	2.072177e-2
	$W1=net.IW\{1,1\}$					
	13	14				
$\mathcal{N}1$	1.048027e-6	7.248138e-5				
$\mathcal{N}2$	8.956699e-3	6.197595e-3				
$\mathcal{N}3$	8.934872e-3	6.163813e-3				
$\mathcal{N}4$	-1.564068e-3	-7.588158e-3				
$\mathcal{N}5$	-6.917724e-3	5.688699e-3				

(a) Pesos para la red neuronal artificial (W1).

$W2=net.IW[2,1]$					
	1	2	3	4	5
$N1$	0.566039	-0.545737	0.547272	0.430302	6.527549e-10

(b) Pesos para la red neuronal artificial ($W2$).

$b1=net.b\{1\}$		$b2=net.b\{2\}$	
$N1$	-0.223953	$N1$	0.08107518
$N2$	1.866728		
$N3$	1.231688		
$N4$	1.690453		
$N5$	4.432494		

(c) Bias o sesgos de las neuronas para la red neuronal artificial.

Tabla 4.12 Resultados obtenidos con la red neuronal artificial entrenada.

En la Tabla 4.13, se muestran algunos de los valores de la simulación de la red; los valores de a_1 corresponden a los valores entregados por la red y los valores de t_1 corresponden a los valores esperados.

$Error$	
$emáx =$	1.6854e-05
$emin =$	0

(a) Errores calculados.

	a_1	t_1	e
100% rpr	0.07640001	0.07640	1.0000e-08
	0.07640001	0.07640	1.0000e-08
	0.07640000	0.07640	0
	0.07640000	0.07640	0
	0.07640001	0.07640	1.0000e-08
120% rpr	0.09166311	0.09168	1.6894e-05
	0.09166313	0.09168	1.6875e-05
	0.09166314	0.09168	1.6860e-05
	0.09166315	0.09168	1.6854e-05
	0.09166314	0.09168	1.6856e-05
150% rpr	0.11458899	0.11460	1.1010e-05
	0.11458894	0.11460	1.1060e-05
	0.11458891	0.11460	1.1090e-05
	0.11458881	0.11460	1.1190e-05
	0.11458882	0.11460	1.1180e-05

(b) Valores obtenidos por la red y valores esperados.

Tabla 4.13 Valores de la simulación.

En la Tabla 4.13 puede observarse el trabajo final de la red neuronal artificial y la labor adecuada de aproximación que realiza. Sí se mostrara el resultado para todos los patrones de prueba resultaría extenso, por lo tanto se escogieron los patrones más representativos, donde puede verse que sin ningún problema la red ha aproximado patrones

que no se le habían presentado durante el entrenamiento, garantizando el éxito de la red neuronal artificial para realizar estimaciones de la resistencia del rotor.

Durante el entrenamiento de la red neuronal artificial se comprobó que el algoritmo *trainlm*, correspondiente al método de Levenberg Marquardt garantiza una alta velocidad de aprendizaje y se obtuvo una generalización de los patrones de entrenamiento que no se le habían presentado inicialmente, y por eso se escogió para completar el proceso de entrenamiento.

El proceso de entrenamiento involucró experimentar con varias configuraciones en la red neuronal artificial, tratando de encontrar no solo una configuración adecuada, sino también un algoritmo que además de rapidez garantizara estabilidad en el resultado final. Por lo tanto, la red neuronal artificial mostrada en la Fig. 4.3 se selecciona debido a que tiene un error menor como se mostró en la Fig. 4.2(c), de esta manera se busca que para condiciones de operación transitoria del control vectorial indirecto se obtenga un desempeño satisfactorio, como se verá mas adelante en este capítulo.

De esta forma se confirma que no existe un procedimiento establecido para determinar el modelo de la red que debe emplearse en cada aplicación y que sólo con la práctica puede determinarse cual es la configuración de la red que garantiza mejores resultados, esta aparente dificultad de las redes neuronales artificiales puede aprovecharse para resaltar una de sus grandes ventajas: la facilidad con que se pueden implementar en cualquier lenguaje de programación, de tal forma que con unas pocas instrucciones los parámetros de la red pueden ser transformados sin mayor problema, adoptando una configuración totalmente diferente [41].

4.3.4 Estimador neuronal de la resistencia del rotor

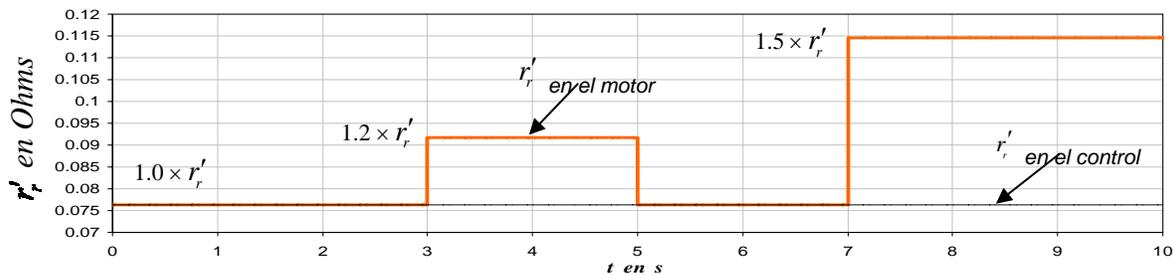
A partir del esquema presentado en la Fig. 3.12, en la Fig. 4.4 se muestra el diagrama general de la simulación en SIMULINK para el control vectorial indirecto del motor de inducción con PWM y regulador de corriente síncrono, en el cual se realiza la estimación y actualización de la resistencia del rotor, donde el estimador utilizado en este caso está basado en una red neuronal artificial.

Como se ve en el bloque del estimador de la resistencia del rotor de la Fig. 4.4, básicamente se utilizan las mismas señales de entrada que en el caso del estimador descrito en la Fig. 4.1, las cuales son: el flujo del rotor en el eje d , Ψ_{dr}^{re*} , la cual es obtenida del bloque de campo debilitado; la corriente bifásica síncrona de referencia, i_{ds}^{e*} , y la frecuencia del estator, ω_e , que provienen del bloque de campo orientado; y al voltaje bifásico síncrono, v_{ds}^e , v_{qs}^e , así como a la corriente bifásica síncrona i_{ds}^e , i_{qs}^e , que se obtienen de los bloques *vqds2qde* e *iqds2qde* respectivamente, cuyos voltajes y corrientes se toman de las terminales del motor de inducción como se muestra en la Fig. 4.4. El interior del bloque del *estimador neuronal* es la red neuronal artificial entrenada, cuya configuración y entrenamiento se obtuvo en 4.3.3.3 y se muestra en la Fig. 4.3.

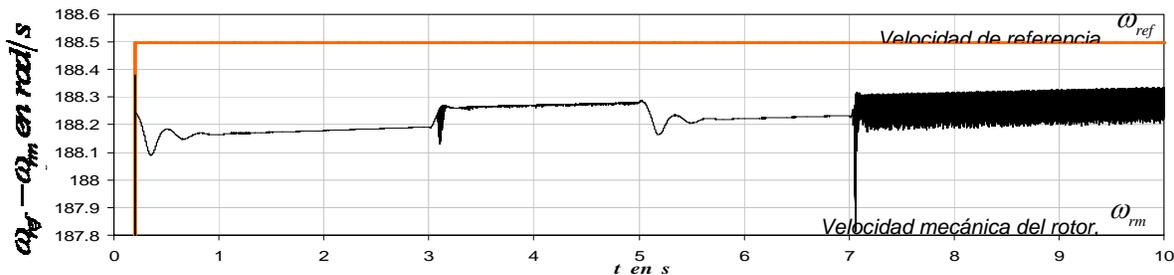
4.4 Resultados de la simulación del control vectorial indirecto sin la estimación de la resistencia del rotor

La Fig. 4.5 tiene la finalidad de mostrar el comportamiento del control vectorial indirecto cuando no se realiza la estimación de la resistencia del rotor. En la Fig. 4.5(a) se muestran cambios de la resistencia del rotor en el motor de inducción manteniendo constante la resistencia del rotor en el control vectorial indirecto.

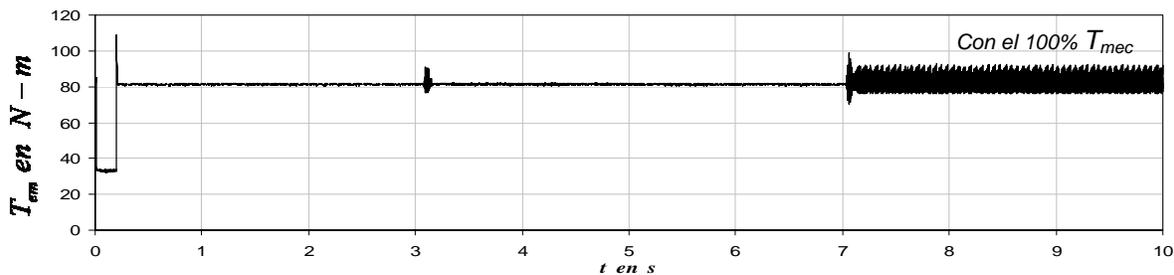
Como se observa el desempeño del control vectorial indirecto se ve perturbado. En la Fig. 4.5(b) cuando la resistencia del rotor varía hasta el 150% de su valor en $t = 7$ s, la velocidad mecánica del rotor presenta oscilaciones modificando el valor del par electromagnético (Fig. 4.5(c)) y del flujo del rotor (Fig. 4.5(d)) el cual toma valores por arriba de los esperados. Así mismo, se observa que la velocidad mecánica del rotor cae abruptamente (Fig. 4.5(e)), esto se presenta en el momento en que el motor de inducción retoma su velocidad nominal en $t = 8$ s, el cual se somete al par de carga nominal y se varía la resistencia del rotor hasta el 150%. Estos resultados demuestran la necesidad de realizar la estimación de la resistencia del rotor y su adaptación en el algoritmo de control.



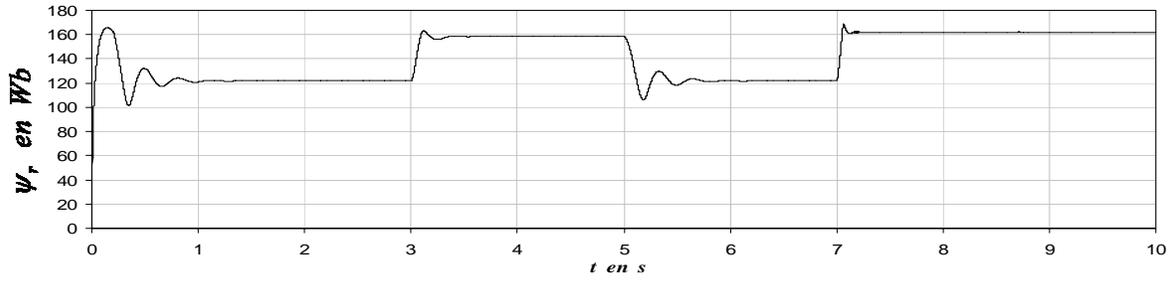
(a) Variación de la resistencia del rotor.



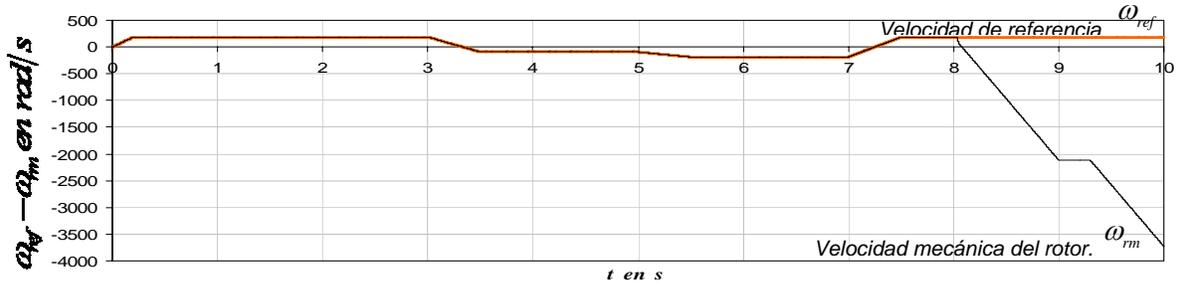
(b) Velocidad mecánica del rotor cuando el motor opera con el par de carga nominal.



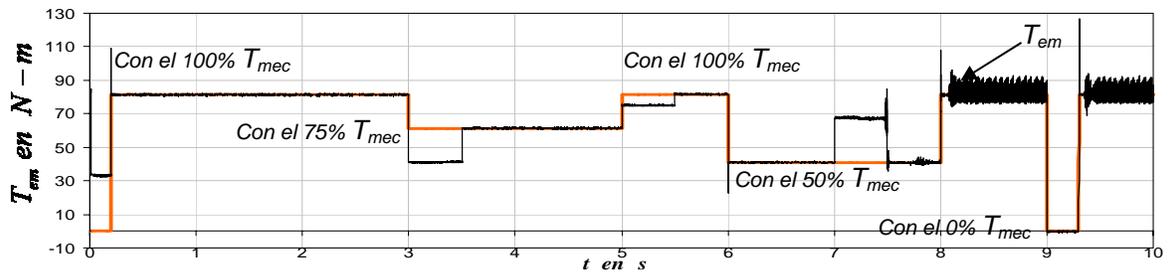
(c) Par electromagnético con velocidad mecánica del rotor y par de carga nominales.



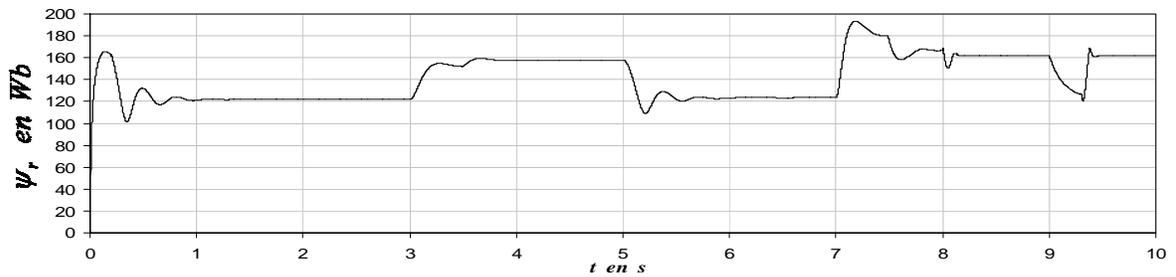
(d) Flujo del rotor cuando la velocidad mecánica del rotor y el par de carga son nominales.



(e) Velocidad mecánica del rotor cuando con velocidad de referencia y par de carga variables.



(f) Par electromagnético con velocidad mecánica del rotor y par de carga variables.



(g) Flujo del rotor cuando la velocidad mecánica del rotor y el par de carga son variables.

Figura 4.5. Efecto de la variación de la resistencia del rotor sin realizar la estimación y la actualización de su valor en el algoritmo de control.

4.5 Resultados de la simulación de la estimación de la resistencia del rotor en estado estable con el motor de inducción de 20 HP

En condiciones de funcionamiento real, la resistencia del rotor varía lentamente parecido a un sistema de primer orden, pero en el estudio de la simulación, se consideran solamente 10 s. Por lo tanto, se evalúa el desempeño de los estimadores basados en un controlador difuso y en una red neuronal artificial, considerando el estado estable.

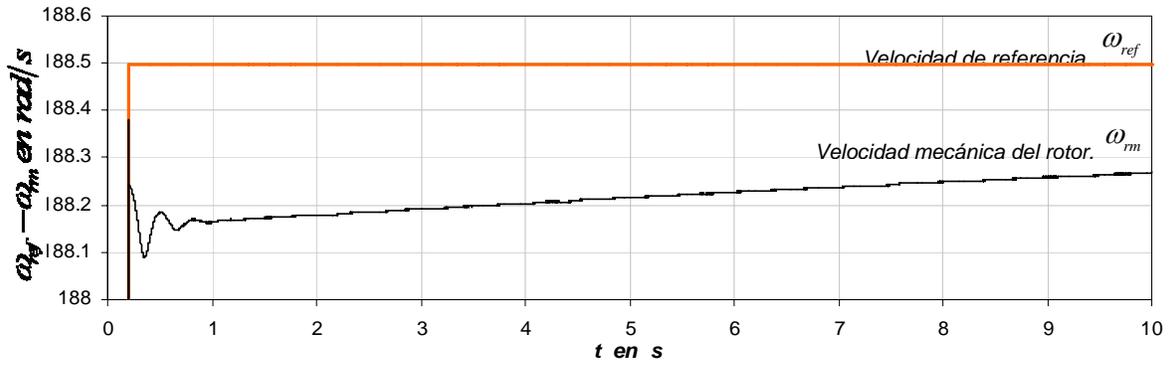
Para la operación de los estimadores se comienza a realizar la estimación de la resistencia del rotor desde los 0 s (aunque se tiene un transitorio debido al arranque del motor), pero a los 2 s es cuando se realiza la actualización de la resistencia del rotor en el bloque de *campo orientado*, ya que se establece la variación de la misma resistencia en el motor de inducción. Así los estimadores compensan la variación de la resistencia del rotor con diferentes condiciones de funcionamiento del motor.

4.5.1 Primer caso estable

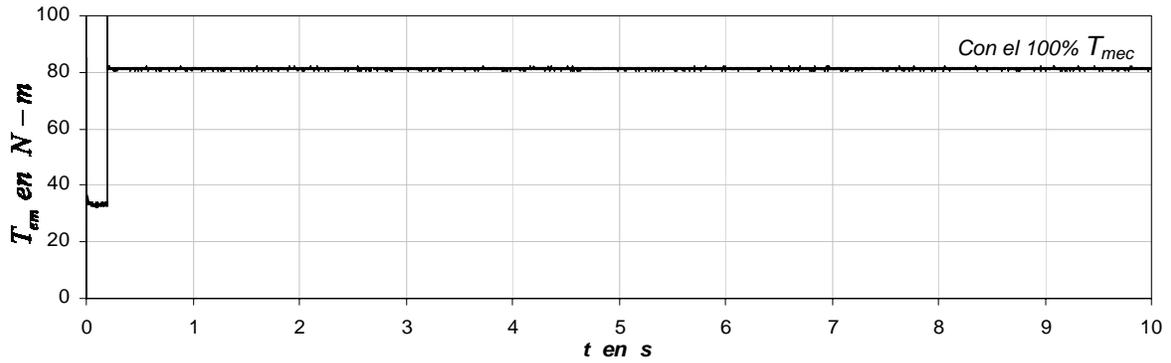
Par de carga y velocidad mecánica del rotor nominales, sin variación en el valor de la resistencia del rotor. Por lo tanto, el archivo *m4.m* de MATLAB (ver Apéndice A), establece las siguientes condiciones de funcionamiento:

t_paro	El tiempo de estudio a 10 segundos.
$tiempo_wref$	Lapso de tiempo de la secuencia de la velocidad de referencia: $[0 \ 0.20 \ t_paro]$.
vel_wref	Secuencia de la velocidad de referencia: $[0 \ \omega_{ref} \ \omega_{ref}]$.
$tiempo_tmec$	Lapso de tiempo para la secuencia del par de carga: $[0 \ 0.20 \ 10]$.
$tmec_tmec$	Secuencia del par de carga: $[0 \ 0 \ -T_{nom}]$.
$Rrot_Rrot$	Secuencia de la resistencia del rotor: $[r'_r \ r'_r \ r'_r]$.

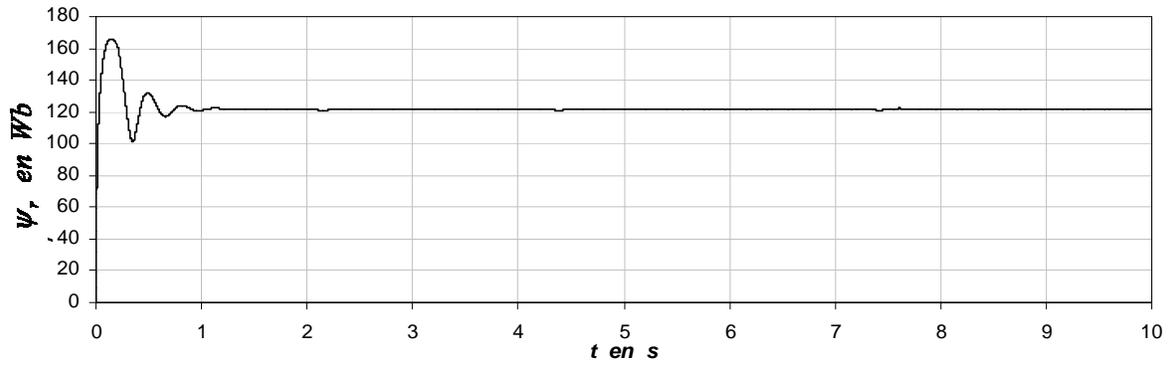
Las Figs. 4.6 y 4.7 muestran el desempeño del estimador basado en un controlador difuso y en una red neuronal artificial en el control vectorial indirecto, respectivamente. En estas figuras se muestran los valores de la velocidad mecánica del rotor, ω_{rm} , el par electromagnético, T_{em} , la magnitud del flujo del rotor, ψ_r , la corriente de la fase *a* del estator, i_{as} , la resistencia del rotor estimada, $r'_{r(est)}$ y la trayectoria del vector de corriente del estator. Los resultados de la simulación son considerando las condiciones del primer caso estable.



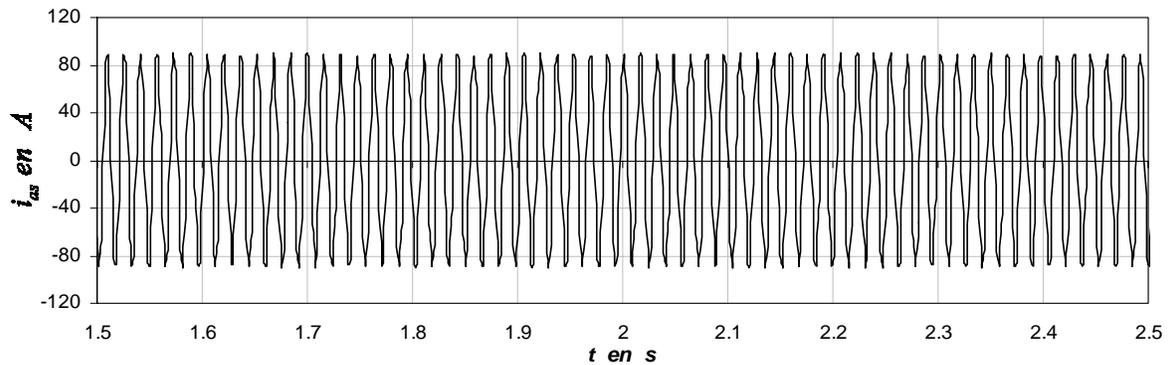
(a) Velocidad mecánica del rotor, ω_m .



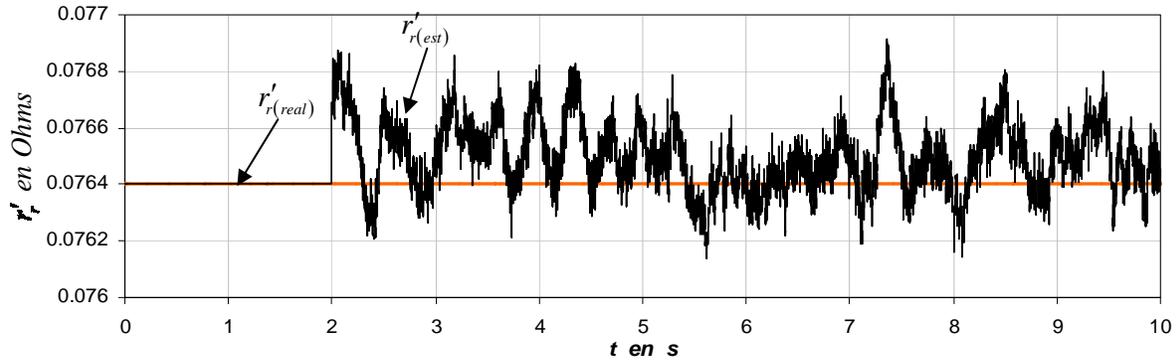
(b) Par electromagnético, T_{em} .



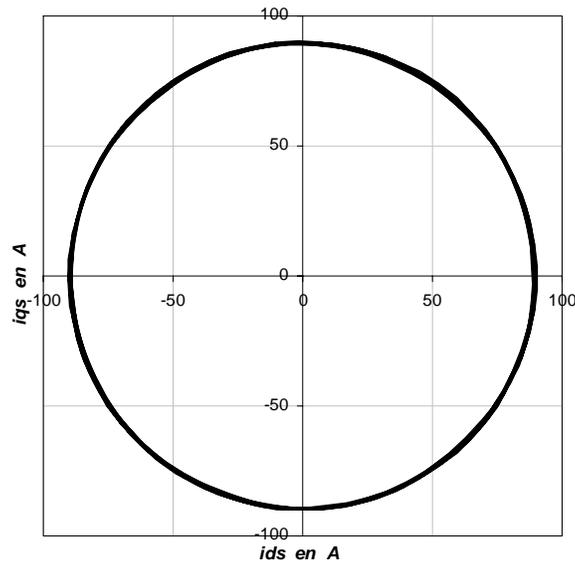
(c) Flujo del rotor.



(d) Corriente de la fase a del estator, i_{as} .



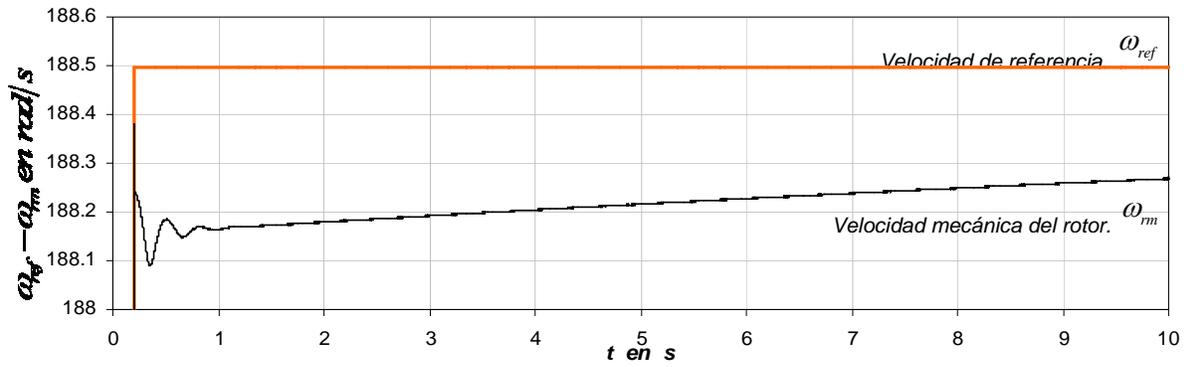
(e) Resistencia del rotor estimada.



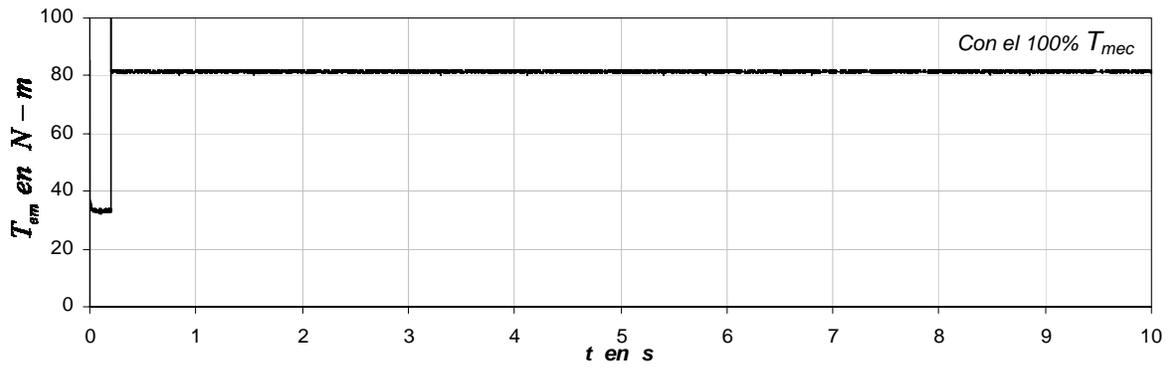
(f) Trayectoria del vector de corriente del estator.

Figura 4.6 Estimación de la resistencia del rotor en estado estable con el estimador basado en un controlador difuso y el comportamiento de las variables del control vectorial indirecto del motor de inducción. *Primer caso estable.*

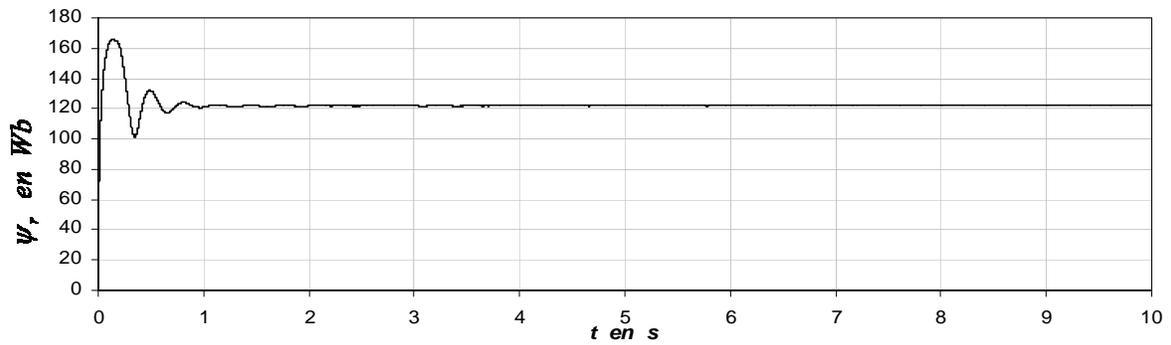
Para propósitos de comparación, se consideran los resultados de la Fig. 4.6(e) donde se muestra la resistencia del rotor estimada con el estimador basado en un controlador difuso, en la cual la resistencia estimada sigue a la resistencia del rotor real, obteniéndose un error de 2.42946×10^{-07} , pero comparada con la resistencia del rotor estimada con el estimador neuronal de la Fig. 4.7(e) se reduce el error tomando un valor de 9.58855×10^{-09} . Por lo tanto, al mantenerse prácticamente constante el valor de la resistencia del rotor, se obtiene que el flujo del rotor se mantenga sin distorsión como se puede observar en la Fig. 4.6(c). Finalmente, se ilustra la trayectoria del vector de corriente del estator en el intervalo de 1.5 s–2.5 s (Figs. 4.6(f) y 4.7(f)), donde se obtiene una trayectoria circular para ambos estimadores, que para este caso ambas trayectorias son similares, puesto que el control vectorial indirecto opera en estado estable.



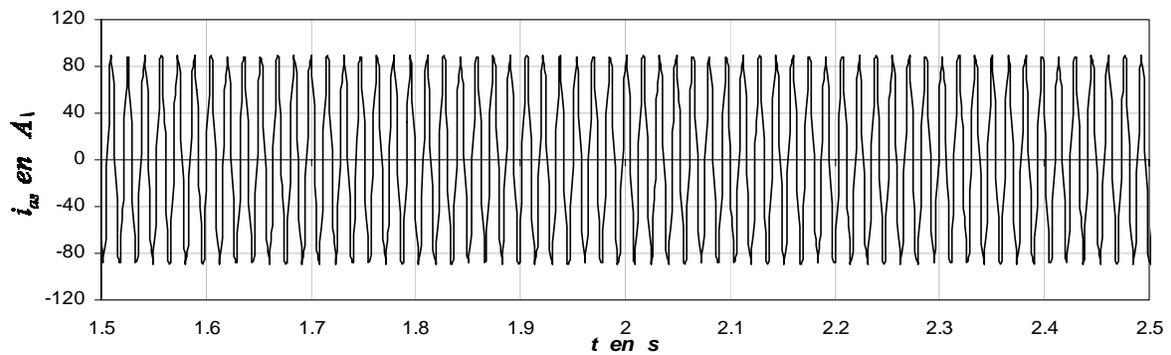
(a) Velocidad mecánica del rotor, ω_m .



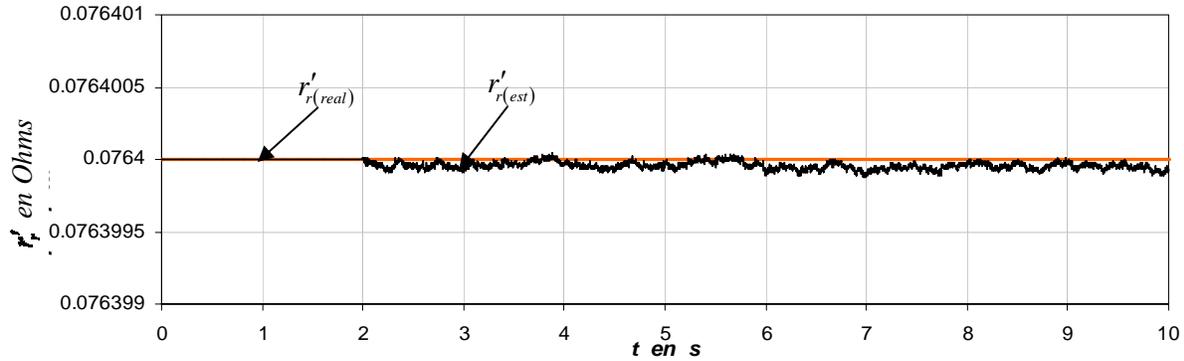
(b) Par electromagnético, T_{em} .



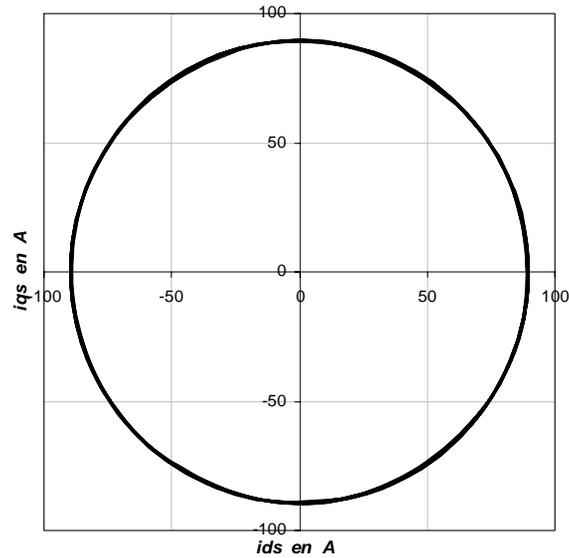
(c) Flujo del rotor.



(d) Corriente de la fase a del estator, i_{as} , en $t=2$ s se realiza la actualización de la r'_r .



(e) Resistencia del rotor estimada.



(f) Trayectoria del vector de corriente del estator.

Figura 4.7 Estimación de la resistencia del rotor en estado estable con el estimador neuronal y el comportamiento de las variables del control vectorial indirecto del motor de inducción. *Primer caso estable.*

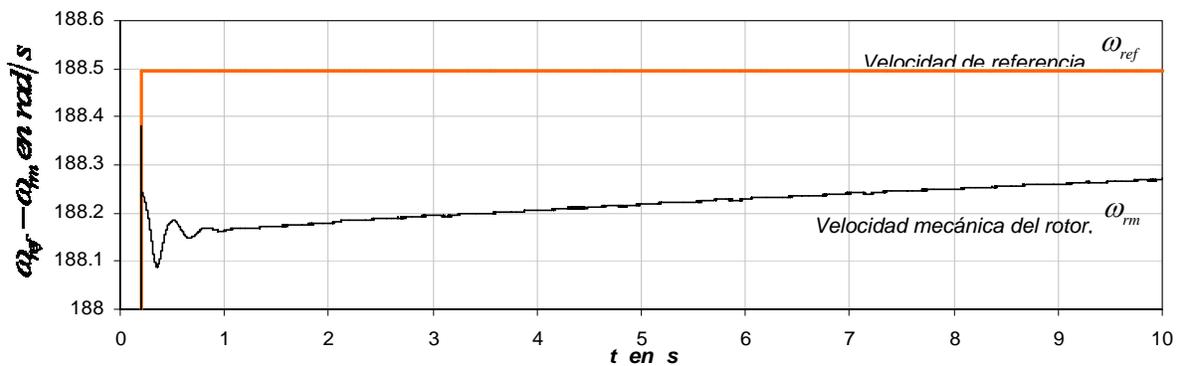
4.5.2 Segundo caso estable

Par de carga y velocidad mecánica del rotor nominales, se establece una variación desde el 100% al 150% en el valor de la resistencia del rotor. Por lo tanto, el archivo *m4.m* de MATLAB (ver Apéndice A), establece las siguientes condiciones de funcionamiento:

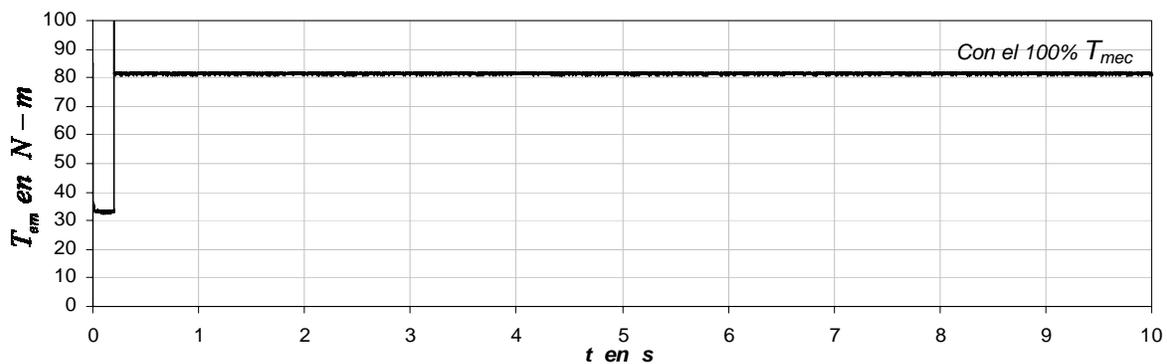
<i>t_paro</i>	El tiempo de estudio a 10 segundos.
<i>tiempo_wref</i>	Lapso de tiempo para la velocidad de referencia: $[0 \ 0.20 \ t_par0]$.
<i>vel_wref</i>	Secuencia de la velocidad de referencia: $[0 \ \omega_{ref} \ \omega_{ref}]$.
<i>tiempo_tmec</i>	Lapso de tiempo para el par de carga: $[0 \ 0.20 \ 10]$.
<i>tmec_tmec</i>	Secuencia del par de carga: $[0 \ 0 \ -T_{nom}]$.
<i>tiempo_Rrot</i>	Lapso de tiempo para la resistencia del rotor en el motor de inducción:

	[0 3.0 3.0 5.0 5.0 7.0 7.0 10]
R_{rot_Rrot}	Secuencia de la resistencia del rotor: $[r'_r \ r'_r \ r'_r \ 1.2 \times r'_r \ 1.2 \times r'_r \ r'_r \ 1.5 \times r'_r \ 1.5 \times r'_r]$

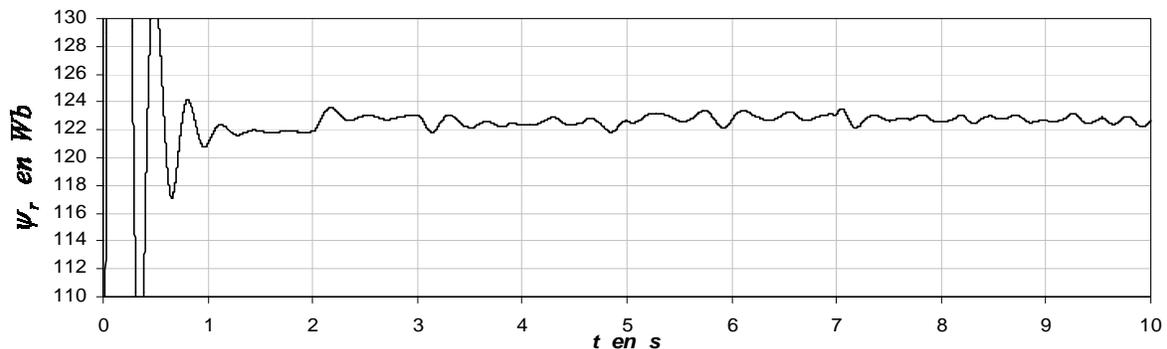
Las Figs. 4.8 y 4.9 muestran el desempeño del estimador basado en un controlador difuso y en una red neuronal artificial en el control vectorial indirecto, respectivamente. Se grafican los valores de la velocidad mecánica del rotor, ω_{rm} , el par electromagnético, T_{em} , la magnitud del flujo del rotor, ψ_r , la corriente de la fase a del estator, i_{as} , la resistencia estimada del rotor, $r'_{r(est)}$ y la trayectoria del vector de corriente del estator. Los resultados de la simulación son considerando las condiciones del segundo caso estable.



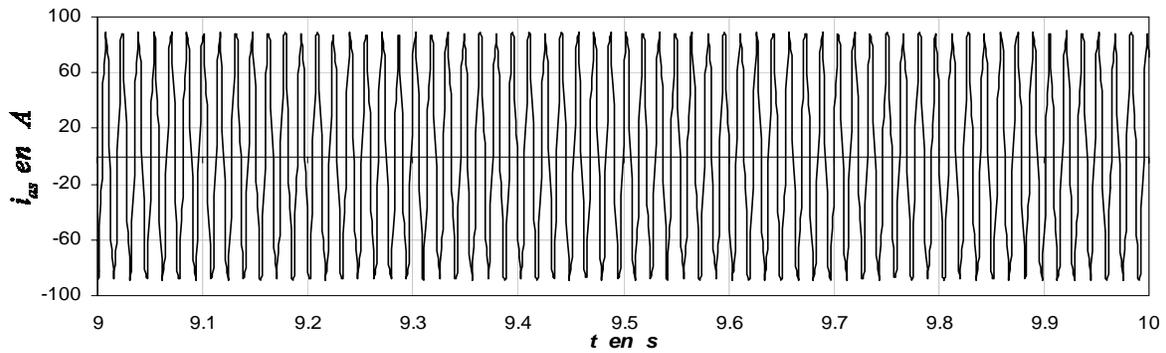
(a) Velocidad mecánica del rotor, ω_{rm} .



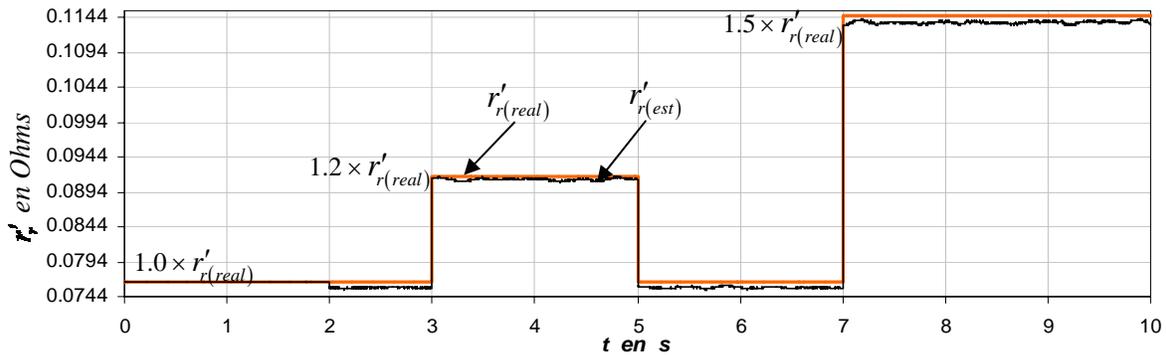
(b) Par electromagnético, T_{em} .



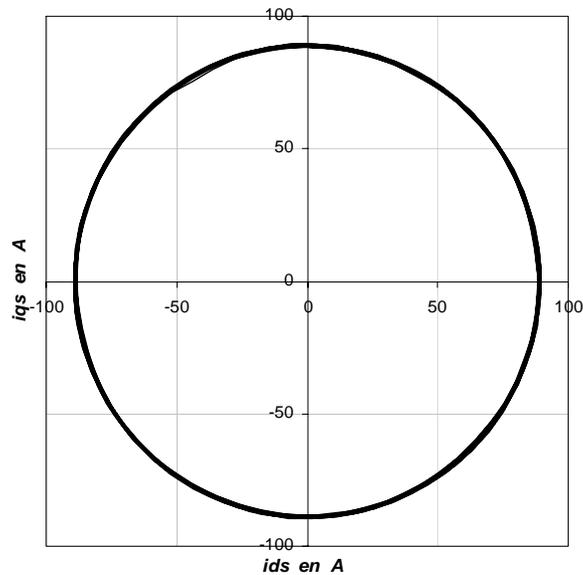
(c) Flujo del rotor.



(d) Corriente de la fase a del estator, i_{as} .



(e) Resistencia del rotor estimada.

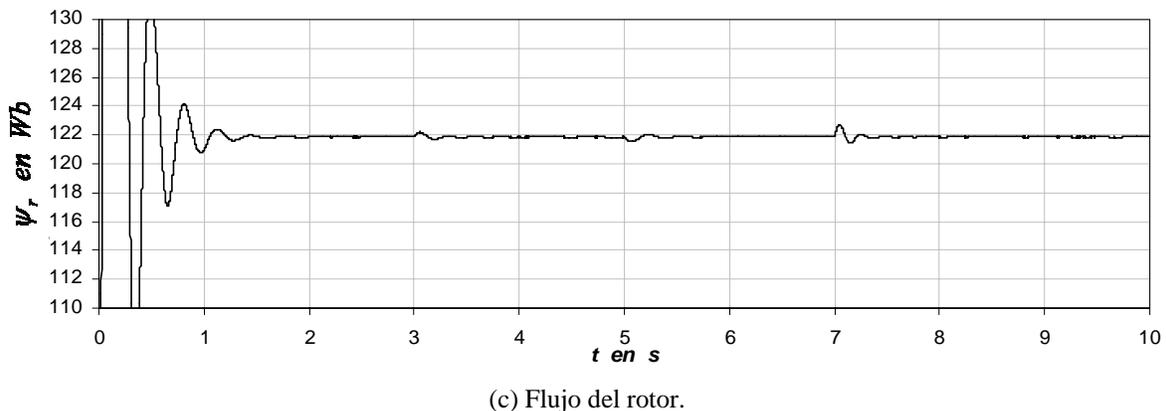
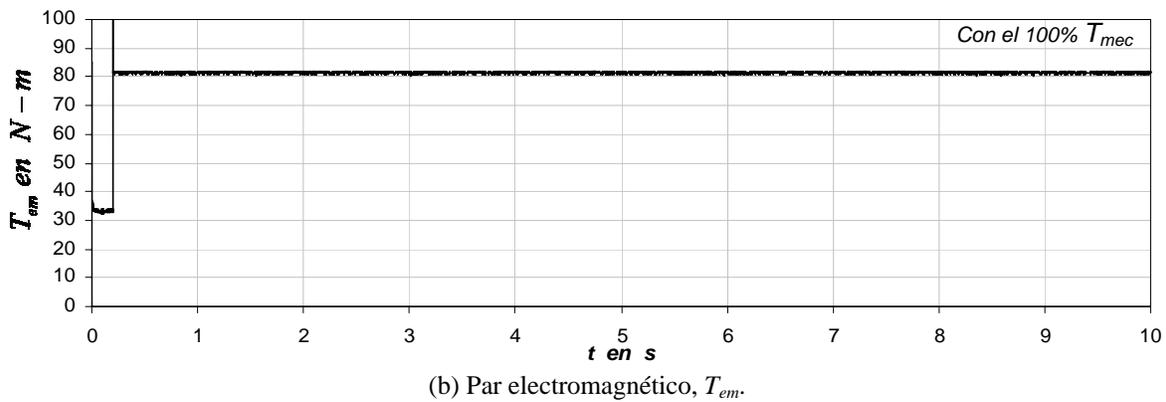
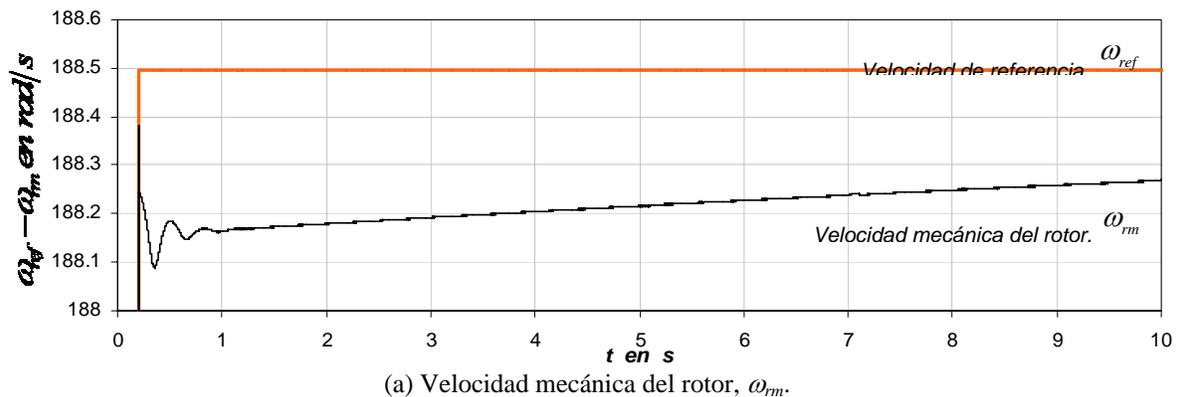


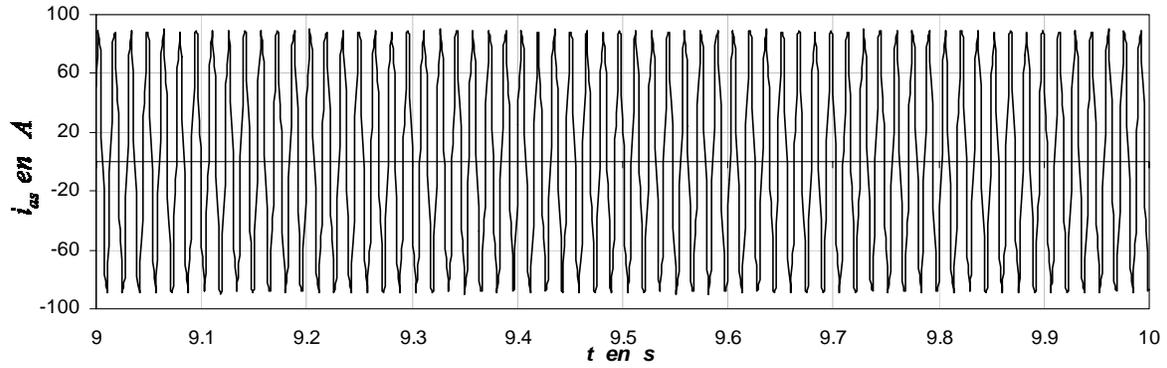
(f) Trayectoria del vector de corriente del estator.

Figura 4.8 Estimación de la resistencia del rotor en estado estable con el estimador basado en un controlador difuso y comportamiento de las variables del control vectorial indirecto del motor de inducción. Se realizan cambios de paso en la resistencia del rotor. *Segundo caso estable.*

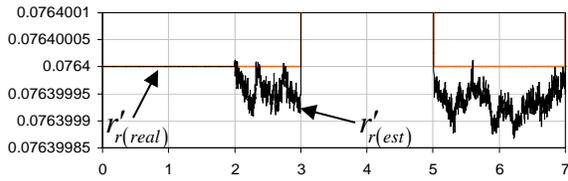
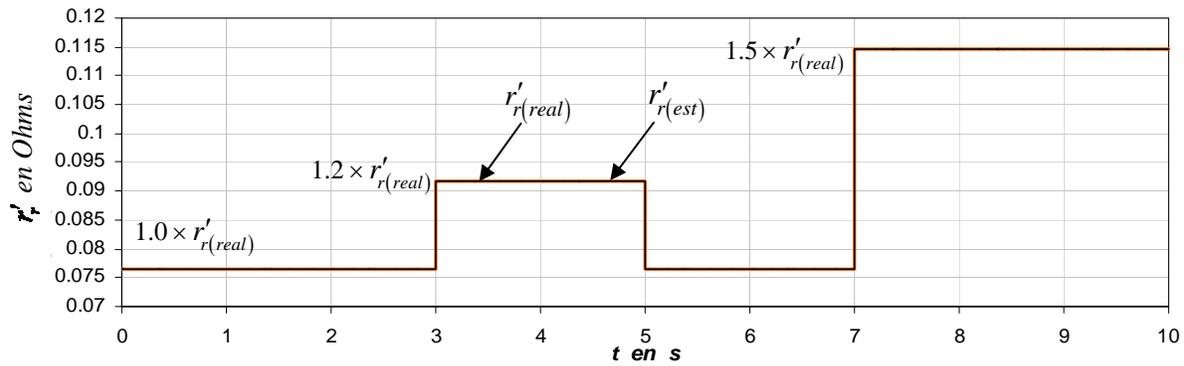
Se consideran los resultados de la Fig. 4.8(e) donde se muestra la resistencia del rotor estimada con el estimador basado en un controlador difuso, en la cual la resistencia

estimada tiende a seguir a la resistencia del rotor real, en donde el error es de 6.93784×10^{-04} . No obstante, con el estimador neuronal la resistencia del rotor estimada de la Fig. 4.9(e), sigue adecuadamente a la resistencia del rotor real, como se observa en los acercamientos que se realizaron para los tres intervalos de variación de la resistencia del rotor, por lo tanto, disminuye el error cuyo valor es de 1.41325×10^{-07} . De esta manera, al mantenerse prácticamente constante el valor de la resistencia del rotor, se obtiene que el flujo del rotor se mantenga sin distorsión como se puede observar en la Fig. 4.9(c) en contraste con el flujo del rotor de la Fig. 4.8(c). Finalmente, se ilustra la trayectoria del vector de corriente del estator en el intervalo de 9.0 s – 10.0 s (Figs. 4.8(f) y 4.9(f)), donde se obtiene una trayectoria circular para ambos estimadores, que para este caso ambas trayectorias son similares.



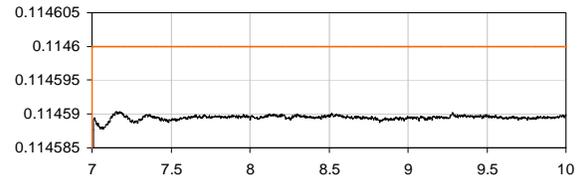


(d) Corriente de la fase *a* del estator, i_{as} .



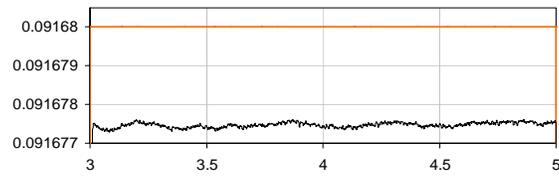
Resistencia del rotor en el intervalo de 0 - 2 s y de 5 - 6 s, se

realiza la estimación considerando que vale $1.0 \times r'_r$.



Resistencia del rotor en el intervalo de 7 - 10 s, se realiza la

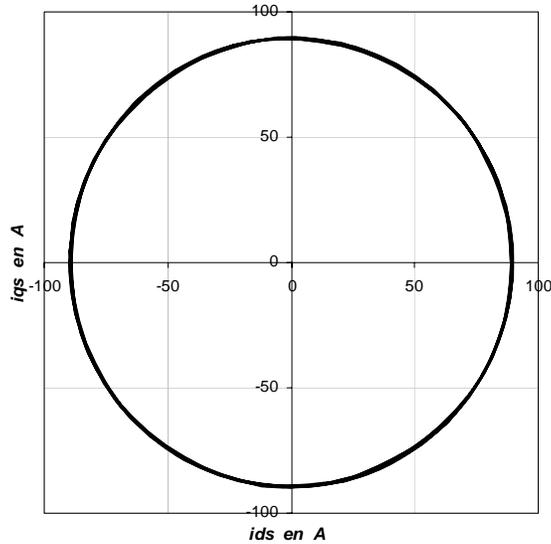
estimación considerando que vale $1.5 \times r'_r$.



Resistencia del rotor en el intervalo de 3 - 7 s, se realiza la

estimación considerando que vale $1.2 \times r'_r$.

(e) Resistencia del rotor estimada.



(f) Trayectoria del vector de corriente del estator.

Figura 4.9 Estimación de la resistencia del rotor en estado estable con el estimador neuronal y comportamiento de las variables del control vectorial indirecto del motor de inducción. Se realizan cambios de paso en la resistencia del rotor. *Segundo caso estable.*

4.6 Resultados de la simulación de la estimación de la resistencia del rotor en estado transitorio con el motor de inducción de 20 HP

Para la operación del estimador basado en un controlador difuso y en una red neuronal artificial en estado transitorio, se consideran diferentes condiciones de operación del motor, tanto para variaciones del par de carga como de la velocidad mecánica del rotor. También se realizan las simulaciones considerando que la resistencia del rotor se mantiene constante y posteriormente se ejecutan cambios de paso en la resistencia.

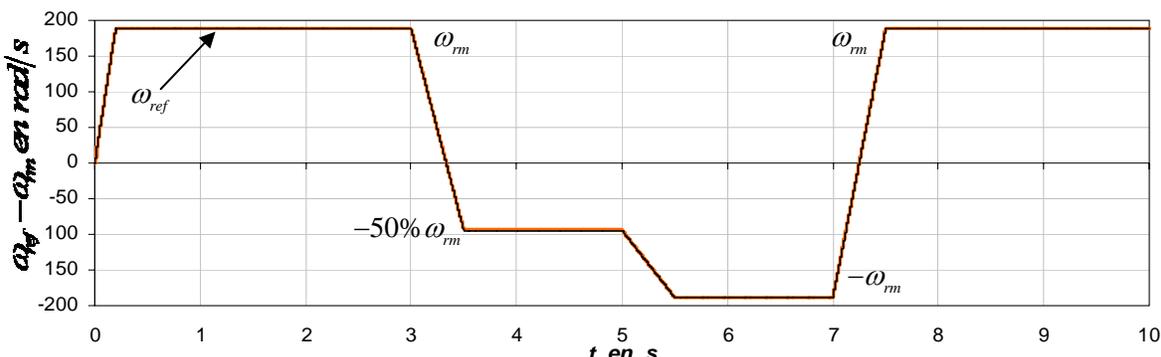
4.6.1 Primer caso transitorio

Par de carga nominal y velocidad mecánica del rotor variable, sin variación en el valor de la resistencia del rotor. Por lo tanto, el archivo *m5.m* de MATLAB (ver Apéndice A), establece las siguientes condiciones de funcionamiento:

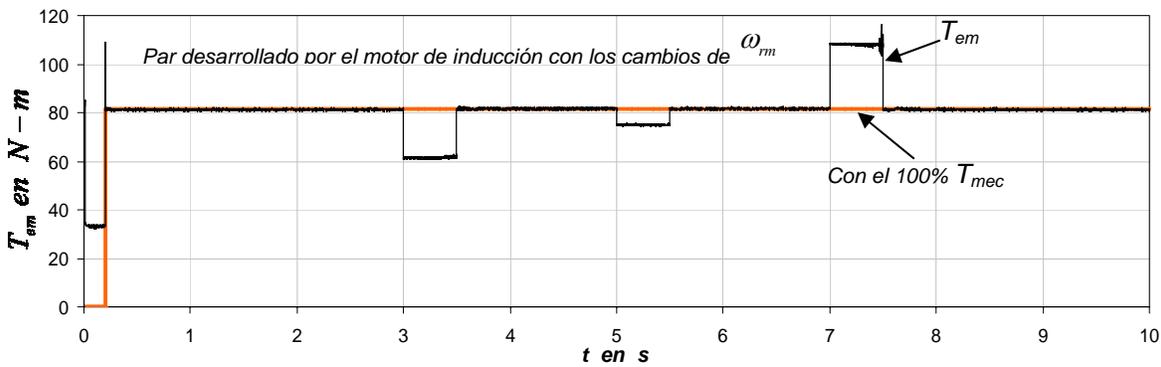
<i>t_paro</i>	El tiempo de estudio a 10 segundos.
<i>tiempo_wref</i>	Lapso de tiempo para la velocidad de referencia: [0 0.20 3.0 3.5 5.0 5.5 7.0 7.5 10.0]
<i>vel_wref</i>	Secuencia de la velocidad de referencia: [0 ω_{ref} ω_{ref} $-0.5 \times \omega_{ref}$ $-0.5 \times \omega_{ref}$ $-1.0 \times \omega_{ref}$ $-1.0 \times \omega_{ref}$ ω_{ref} ω_{ref}].
<i>tiempo_tmec</i>	Lapso de tiempo para el par de carga: [0 0.20 10].

t_{mec_tmech}	Secuencia del par de carga: $[0 \ 0 \ -T_{nom}]$.
R_{rot_Rrot}	Secuencia de la resistencia del rotor: $[r'_r \ r'_r \ r'_r]$.

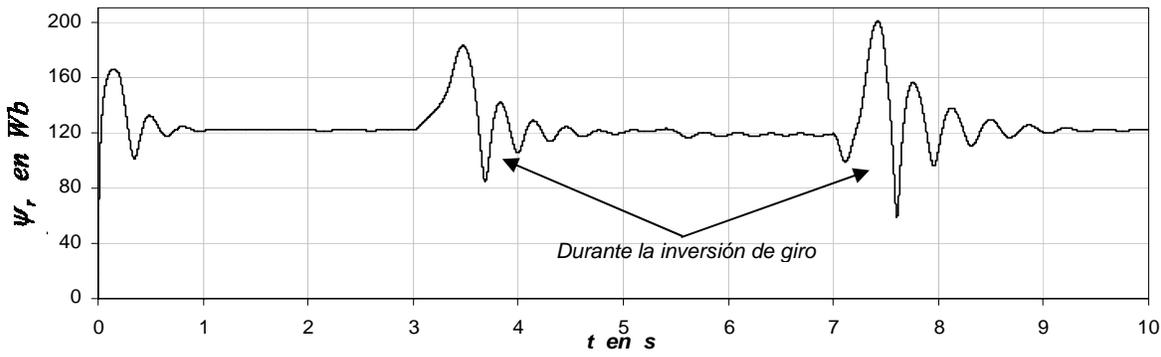
Se muestra el desempeño del estimador basado en un controlador difuso (Fig. 4.10) y del estimador neuronal (Fig. 4.11), en el control vectorial indirecto. Se grafican los valores de la velocidad mecánica del rotor, ω_{rm} , el par electromagnético, T_{em} , la magnitud del flujo del rotor, ψ_r , la corriente de la fase a del estator, i_{as} , la resistencia estimada del rotor, $r'_{r(est)}$ y la trayectoria del vector de corriente del estator. Los resultados de la simulación son considerando las condiciones del primer caso transitorio.



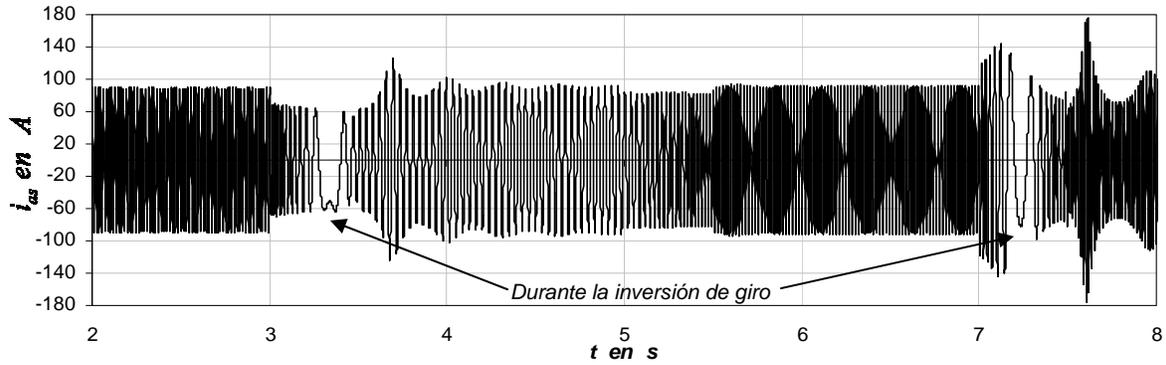
(a) Velocidad mecánica del rotor, ω_{rm} .



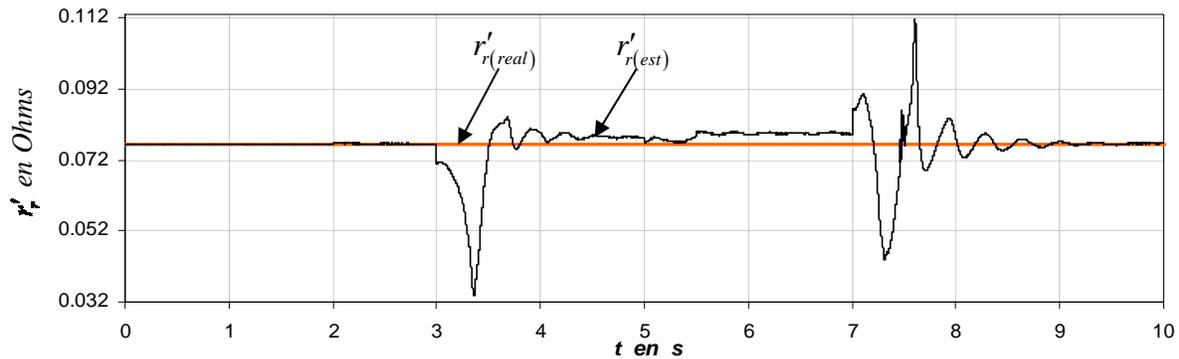
(b) Par electromagnético, T_{em} .



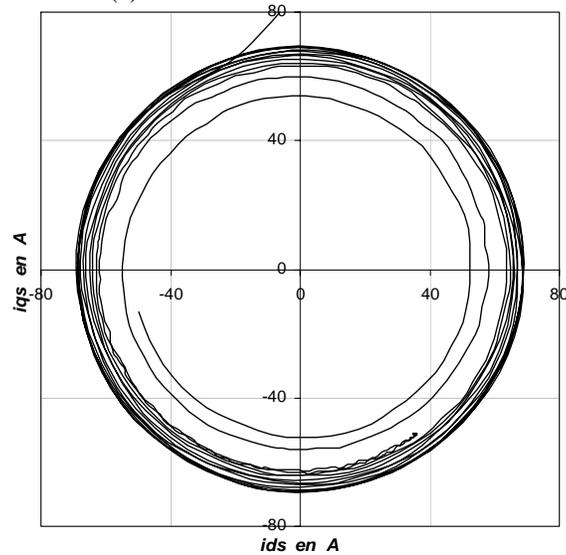
(c) Flujo del rotor.



(d) Corriente de la fase a del estator, i_{as} .



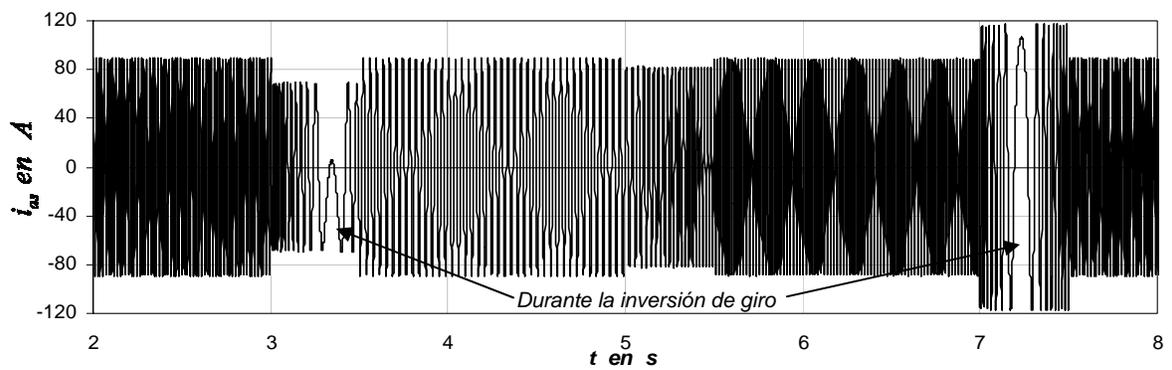
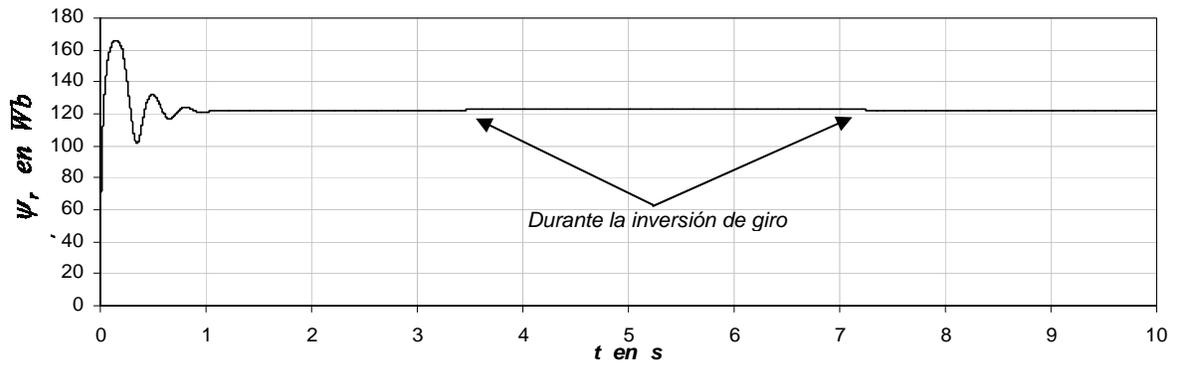
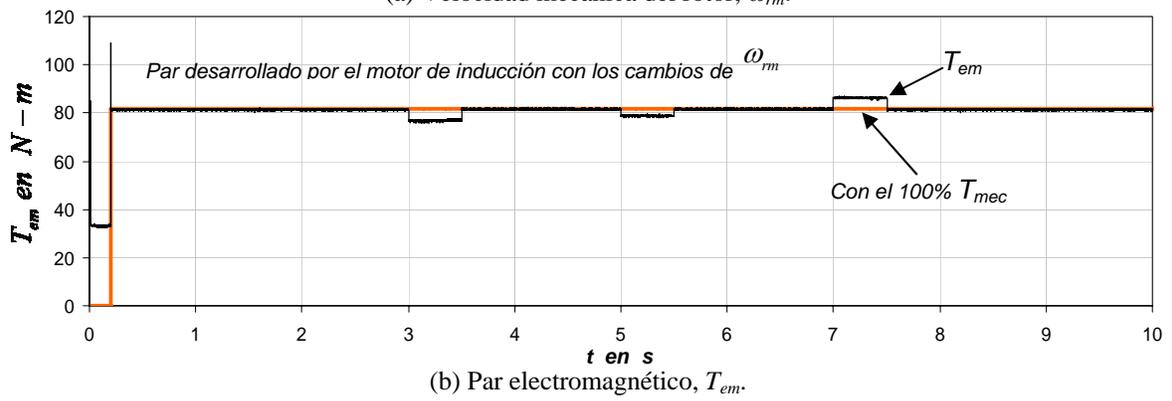
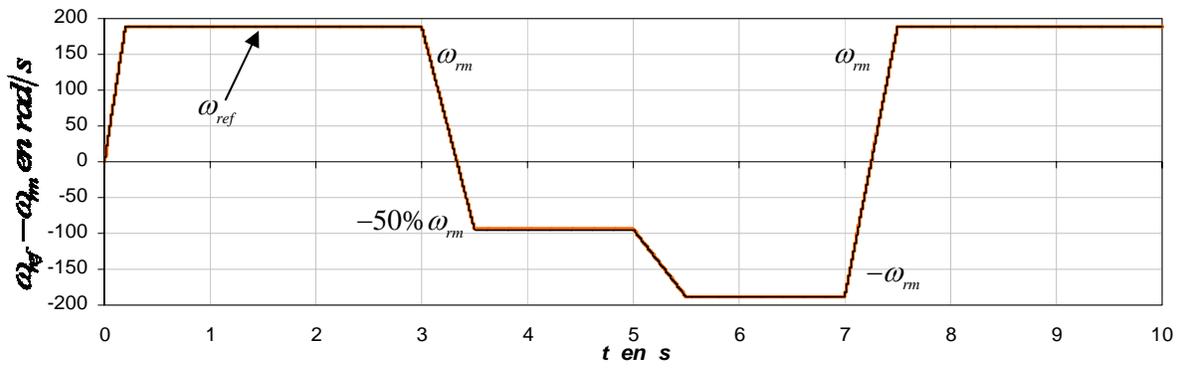
(e) Resistencia del rotor estimada.

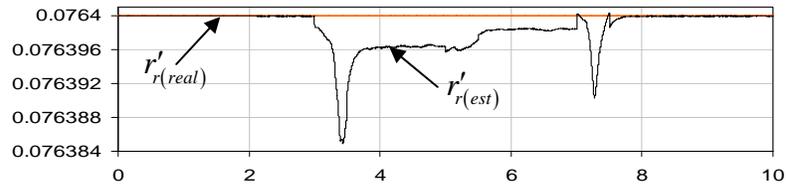
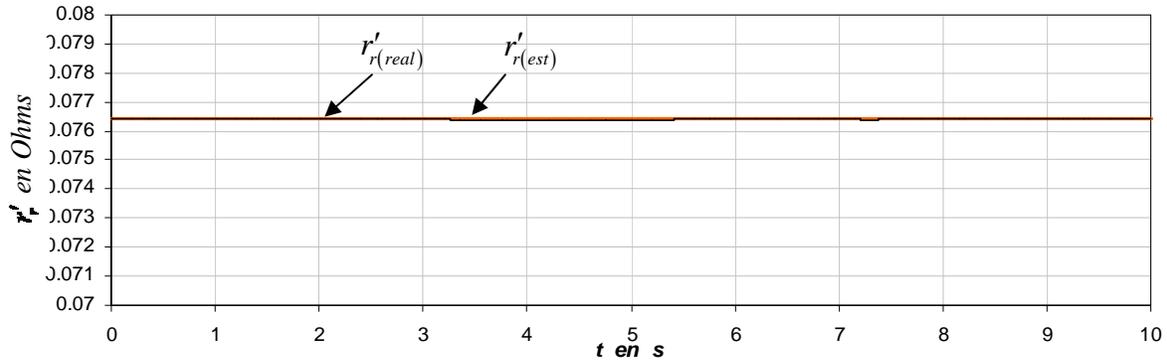


(f) Trayectoria del vector de corriente del estator.

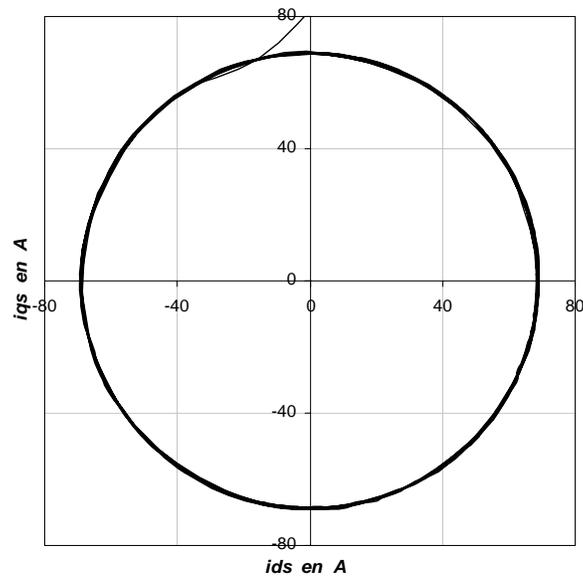
Figura 4.10 Estimación de la resistencia del rotor en estado transitorio con el estimador basado en un controlador difuso y comportamiento de las variables del control vectorial indirecto del motor de inducción. *Primer caso transitorio.*

La Fig. 4.11 muestra el resultado de la simulación del estimador neuronal frente a una inversión de giro. A fin de poder comparar este resultado con el estimador basado en un controlador difuso (Fig. 4.10), en la cual se han utilizado las mismas condiciones de operación.





(e) Resistencia del rotor estimada.



(f) Trayectoria del vector de corriente del estator.

Figura 4.11 Estimación de la resistencia del rotor en estado transitorio con el estimador neuronal y comportamiento de las variables del control vectorial indirecto del motor de inducción. *Primer caso transitorio.*

En la Fig. 4.10(e) se muestra el resultado de la resistencia del rotor estimada con el estimador basado en un controlador difuso, en la cual la resistencia estimada varía al establecerse los cambios de giro del motor, se observa que se realiza una estimación adecuada solamente cuando el motor gira a la velocidad nominal, aunque el error tiene un valor de 2.26032×10^{-04} . No obstante, con el estimador neuronal la resistencia del rotor estimada de la Fig. 4.11(e), sigue a la resistencia del rotor real, como se observa en el acercamiento que se hace a la resistencia del rotor estimada, por lo tanto, se tiene un error de 1.51313×10^{-08} . De esta manera, al mantenerse prácticamente constante el valor de la

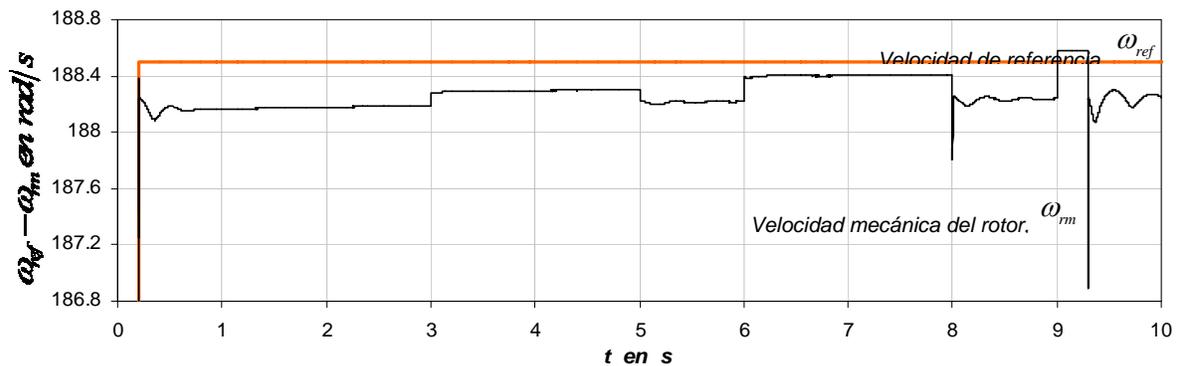
resistencia del rotor, se obtiene que el flujo del rotor se mantenga sin distorsión como se puede observar en la Fig. 4.11(c) en contraste con el flujo del rotor de la Fig. 4.10(c). Finalmente, se ilustra la trayectoria del vector de corriente del estator en el intervalo de 3.0 s – 3.5 s, observándose que en la Fig. 4.10(f), el vector no sigue una sola trayectoria, como si sucede en la Fig. 4.11(f), donde se obtiene una trayectoria circular, indicando que la corriente trifásica es senoidal con menos distorsión (Fig. 4.11(d)), con respecto a la obtenida en la Fig. 4.10(d).

4.6.2 Segundo caso transitorio

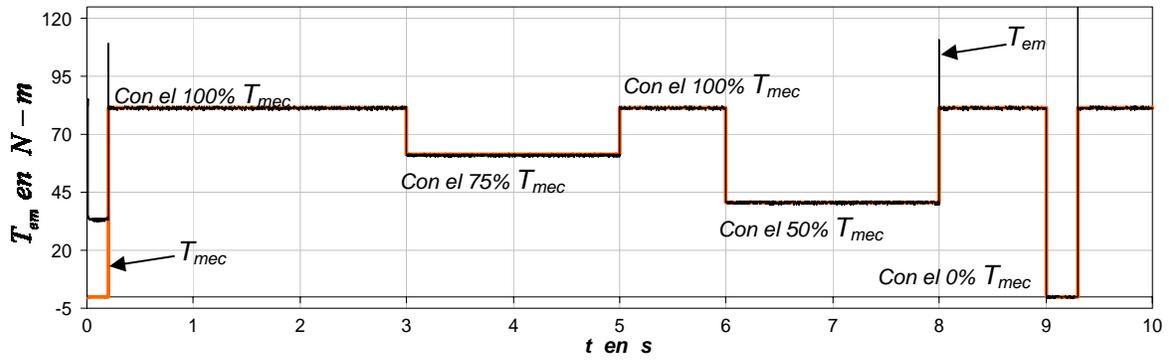
Par de carga variable y velocidad mecánica del rotor nominal, sin variación en el valor de la resistencia del rotor. Por lo tanto, el archivo *m5.m* de MATLAB (ver Apéndice A), establece las siguientes condiciones de funcionamiento:

t_paro	El tiempo de estudio a 10 segundos.
$tiempo_wref$	Lapso de tiempo para la velocidad de referencia: $[0 \ 0.20 \ t_paro]$.
vel_wref	Secuencia de la velocidad de referencia: $[0 \ \omega_{ref} \ \omega_{ref}]$.
$tiempo_tmec$	Lapso de tiempo para el par de carga: $[0 \ 0.20 \ 0.20 \ 3.0 \ 3.0 \ 5.0 \ 5.0 \ 6.0 \ 6.0 \ 8.0 \ 8.0 \ 9.0 \ 9.0 \ 9.3 \ 9.3 \ 10]$.
$tmec_tmec$	Secuencia del par de carga: $[0 \ 0 \ -T_{nom} \ -T_{nom} \ -0.75 \times T_{nom} \ -0.75 \times T_{nom} \ -T_{nom} \ -T_{nom} \ -0.5 \times T_{nom} \ -0.5 \times T_{nom} \ -T_{nom} \ -T_{nom} \ 0 \ 0 \ -T_{nom} \ -T_{nom}]$.
$Rrot_Rrot$	Secuencia de la resistencia del rotor: $[r'_r \ r' \ r']$.

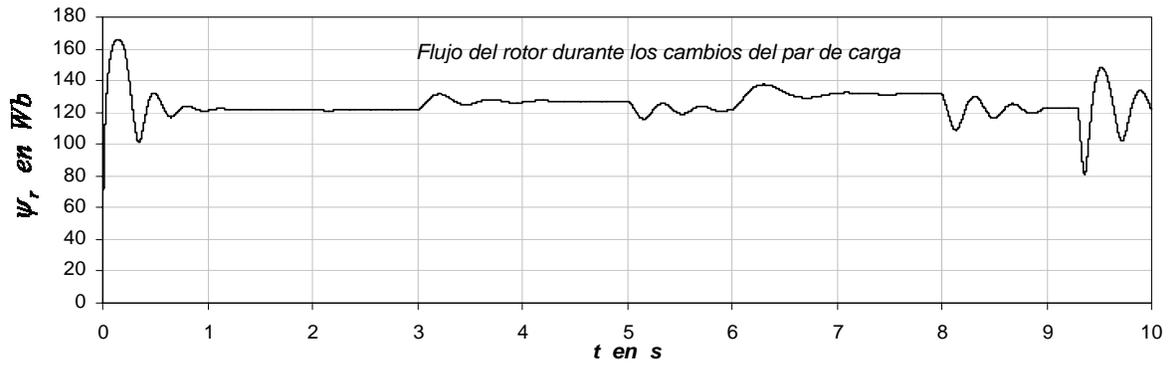
Las Figs. 4.12 y 4.13 muestran el desempeño del estimador basado en un controlador difuso y en una red neuronal artificial en el control vectorial indirecto, respectivamente. Se grafican los valores de la velocidad mecánica del rotor, ω_{rm} , el par electromagnético, T_{em} , la magnitud del flujo del rotor, ψ_r , la corriente de la fase *a* del estator, i_{as} , la resistencia estimada del rotor, $r'_{r(est)}$ y la trayectoria del vector de corriente del estator. Los resultados de la simulación son considerando las condiciones del segundo caso transitorio.



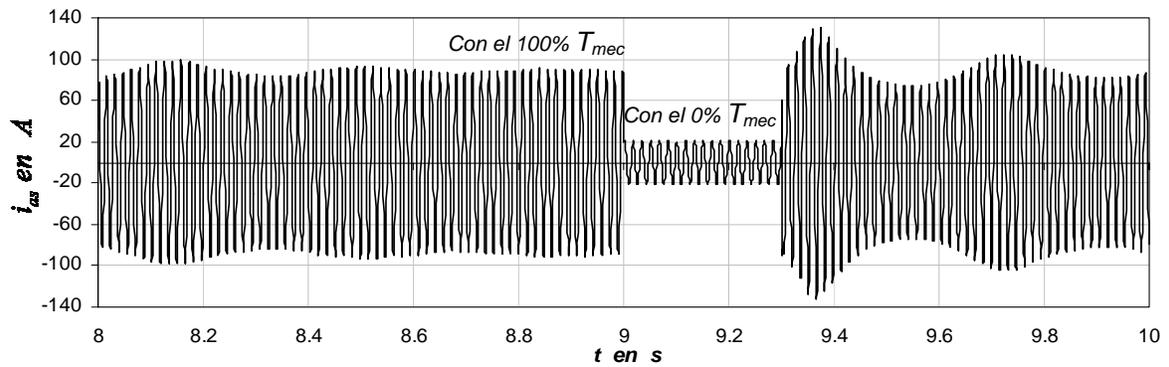
(a) Velocidad mecánica del rotor, ω_{rm} .



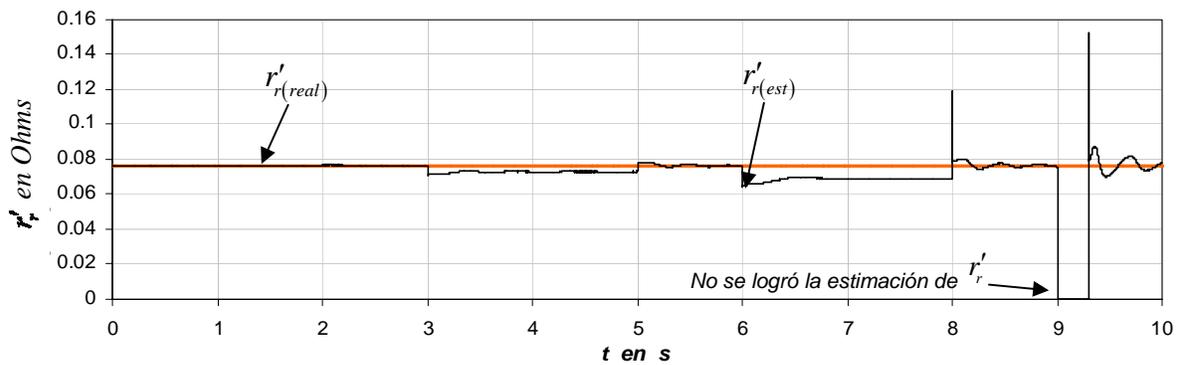
(b) Par electromagnético, T_{em} .

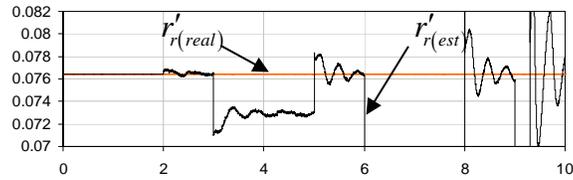


(c) Flujo del rotor.

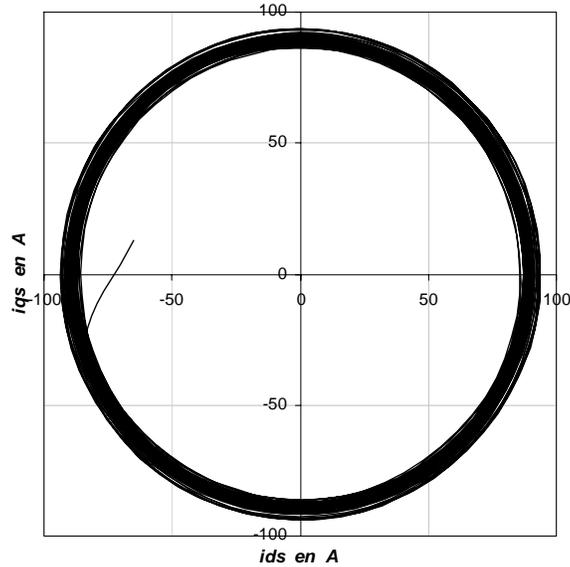


(d) Corriente de la fase a del estator, i_{as} .





(e) Resistencia del rotor estimada.

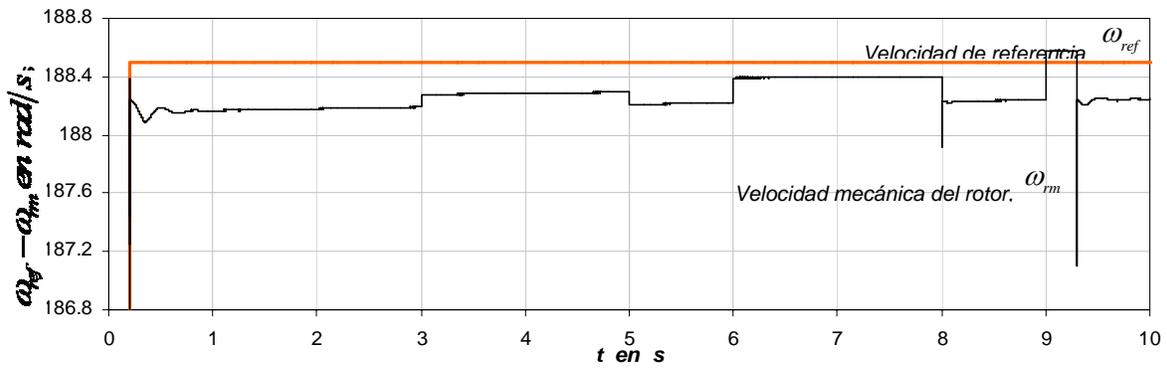


(f) Trayectoria del vector de corriente del estator.

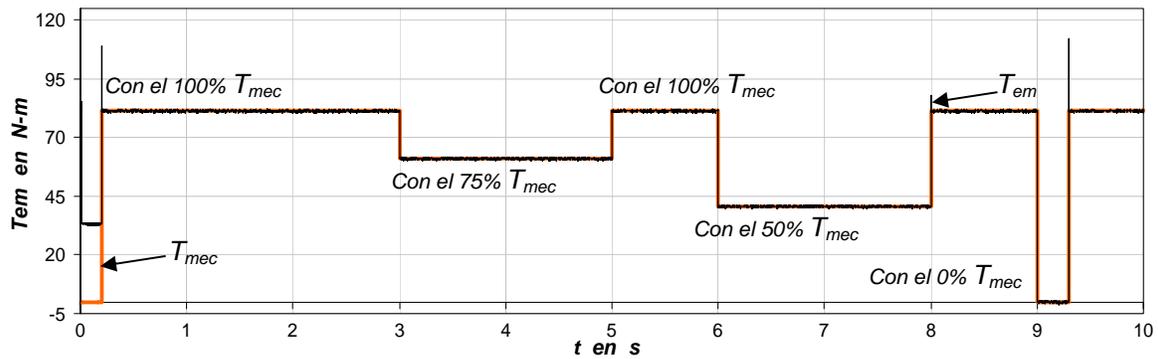
Figura 4.12. Estimación de la resistencia del rotor en estado transitorio con el estimador basado en un controlador difuso y comportamiento de las variables del control vectorial indirecto del motor de inducción. *Segundo caso transitorio.*

La Fig. 4.13 muestra el resultado de la simulación del estimador neuronal frente a una variación del par de carga, manteniendo la velocidad constante. A fin de poder comparar con los resultados obtenidos en la Fig. 4.12, en la cual se han utilizado las mismas condiciones de operación.

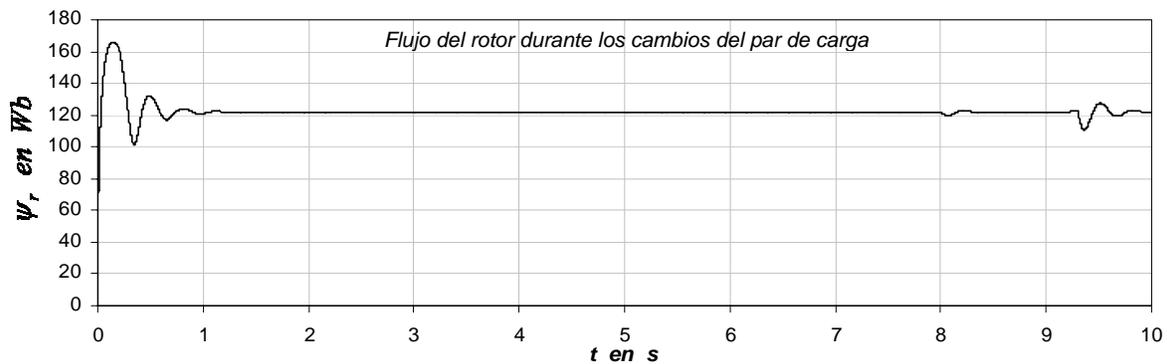
Se consideran los resultados de la Fig. 4.12(e) donde se muestra la resistencia del rotor estimada con el estimador basado en un controlador difuso, en la cual la resistencia estimada varía al establecerse los cambios en el par de carga, observándose que se realiza una estimación adecuada solamente cuando el motor tiene el par de carga nominal y el error toma un valor de 4.77897×10^{-03} . De hecho, cuando el par de carga es cero (en $t = 9.0 \text{ s} - 9.3 \text{ s}$), se pierde el control, ya que en el transitorio que se genera la resistencia del rotor, toma valores muy altos o llega a ser negativa, lo cual no puede ser posible.



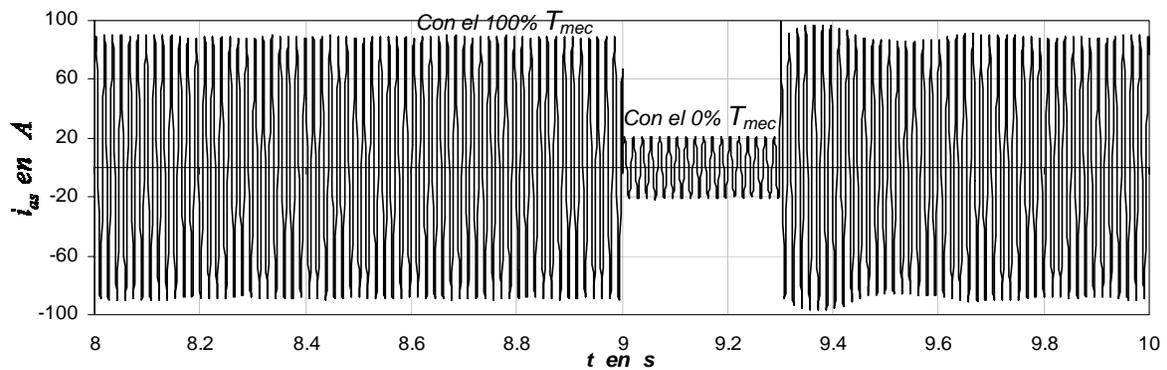
(a) Velocidad mecánica del rotor, ω_m .



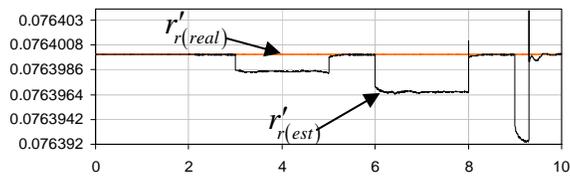
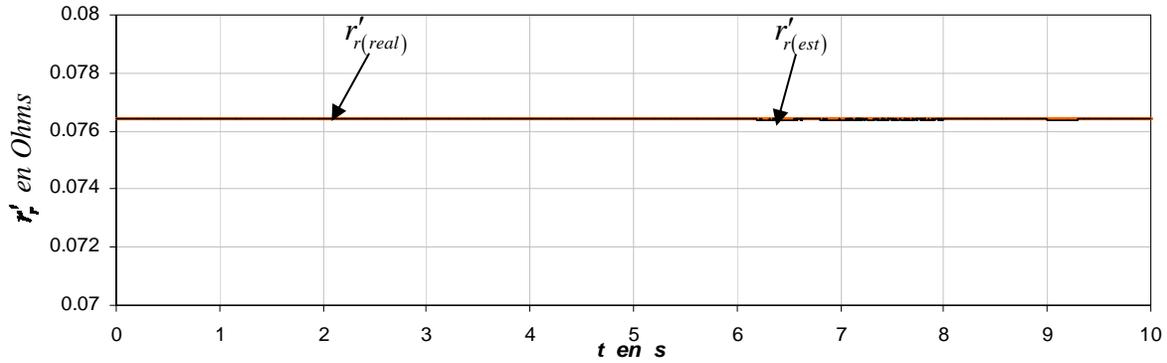
(b) Par electromagnético, T_{em} .



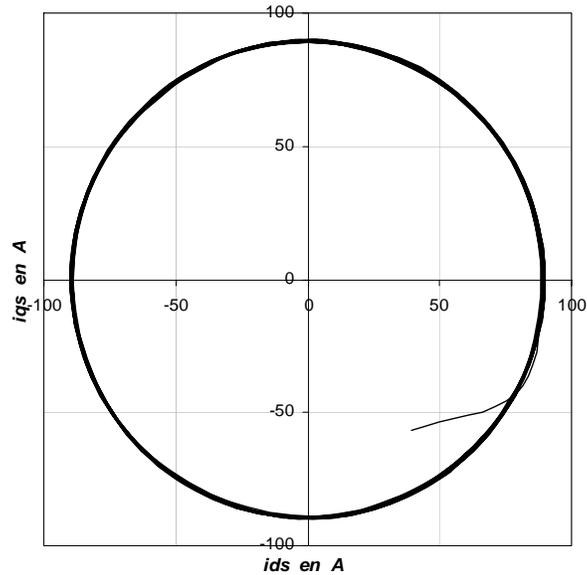
(c) Flujo del rotor.



(d) Corriente de la fase a del estator, i_{as} .



(e) Resistencia del rotor estimada.



(f) Trayectoria del vector de corriente del estator.

Figura 4.13. Estimación de la resistencia del rotor en estado transitorio con el estimador neuronal y comportamiento de las variables del control vectorial indirecto del motor de inducción. *Segundo caso transitorio.*

Por otra parte, con el estimador neuronal la resistencia del rotor estimada de la Fig. 4.13(e), sigue a la resistencia del rotor real, como se observa en el acercamiento que se hace a la resistencia del rotor estimada, por lo tanto, se minimiza el error con un valor de 1.23150×10^{-08} . De esta manera, al mantenerse prácticamente constante el valor de la resistencia del rotor, se obtiene que el flujo del rotor se mantenga sin distorsión como se puede observar en la Fig. 4.13(c) en contraste con el flujo del rotor de la Fig. 4.12(c). Finalmente, se ilustra la trayectoria del vector de corriente del estator en el intervalo de 5.0 s – 6.0 s, observándose que en la Fig. 4.12(f), el vector no sigue una sola trayectoria, como si sucede en la Fig. 4.13(f), donde se obtiene una trayectoria circular, indicando que

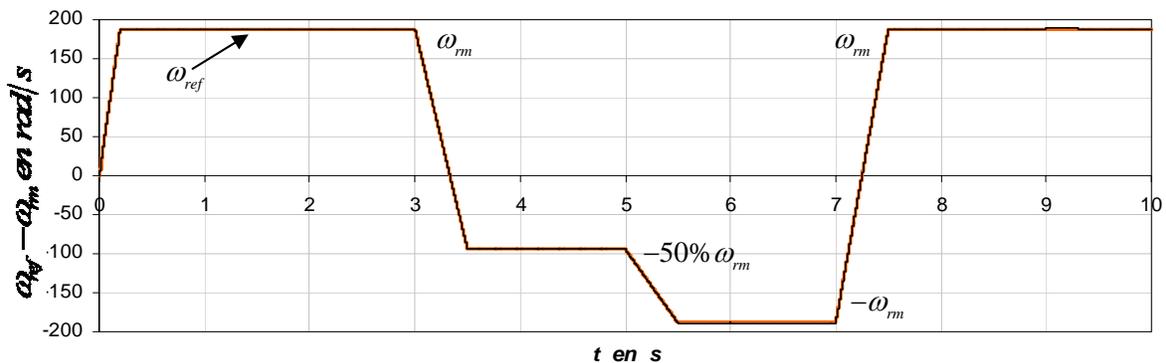
la corriente trifásica es senoidal con menos distorsión (Fig. 4.13(d)), con respecto a la obtenida en la Fig. 4.12(d).

4.6.3 Tercer caso transitorio

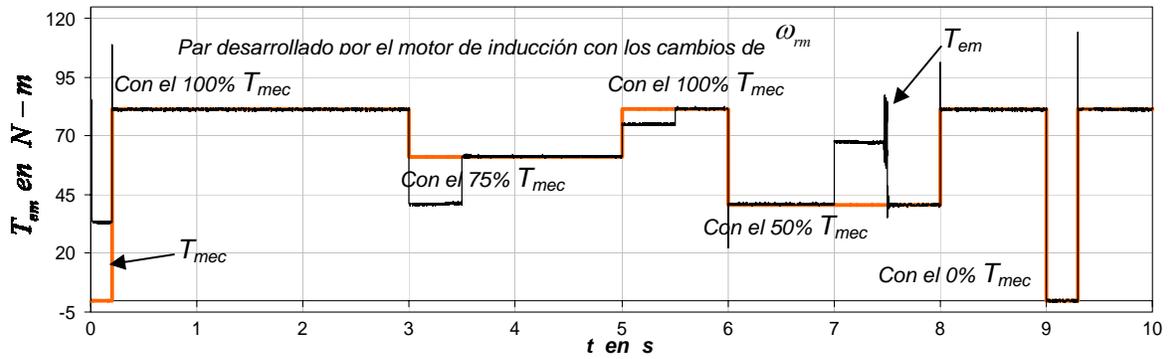
Par de carga y velocidad mecánica del rotor variables, sin variación en el valor de la resistencia del rotor. Por lo tanto, el archivo *m5.m* de MATLAB (ver Apéndice A), establece las siguientes condiciones de funcionamiento:

<i>t_paro</i>	El tiempo de estudio a 10 segundos.
<i>tiempo_wref</i>	Lapso de tiempo para la velocidad de referencia: [0 0.20 3.0 3.5 5.0 5.5 7.0 7.5 10.0].
<i>vel_wref</i>	Secuencia de la velocidad de referencia: [0 ω_{ref} ω_{ref} $-0.5 \times \omega_{ref}$ $-0.5 \times \omega_{ref}$ $-1.0 \times \omega_{ref}$ $-1.0 \times \omega_{ref}$ ω_{ref} ω_{ref}].
<i>tiempo_tmec</i>	Lapso de tiempo para el par de carga: [0 0.20 0.20 3.0 3.0 5.0 5.0 6.0 6.0 8.0 8.0 9.0 9.0 9.3 9.3 10].
<i>tmec_tmec</i>	Secuencia del par de carga: [0 0 $-T_{nom}$ $-T_{nom}$ $-0.75 \times T_{nom}$ $-0.75 \times T_{nom}$ $-T_{nom}$ $-T_{nom}$ $-0.5 \times T_{nom}$ $-0.5 \times T_{nom}$ $-T_{nom}$ $-T_{nom}$ 0 0 $-T_{nom}$ $-T_{nom}$].
<i>Rrot_Rrot</i>	Secuencia de la resistencia del rotor: [r'_r r'_r r'_r].

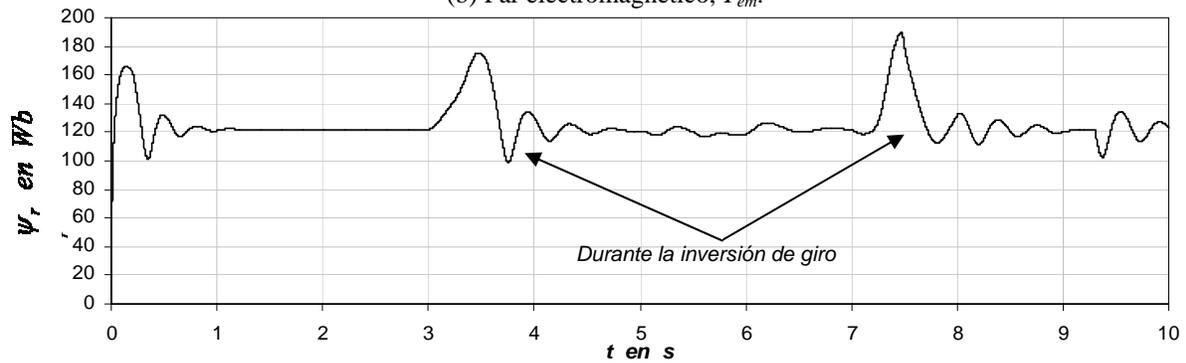
Las Figs. 4.14 y 4.15 muestran el desempeño de ambos estimadores en el control vectorial indirecto. En estas figuras se muestran los valores de la velocidad mecánica del rotor, ω_{rm} , el par electromagnético, T_{em} , la magnitud del flujo del rotor, Ψ_r , la corriente de la fase *a* del estator, i_{as} , la resistencia del rotor estimada, $r'_{r(est)}$ y la trayectoria del vector de corriente del estator. Los resultados de la simulación son considerando las condiciones del tercer caso transitorio.



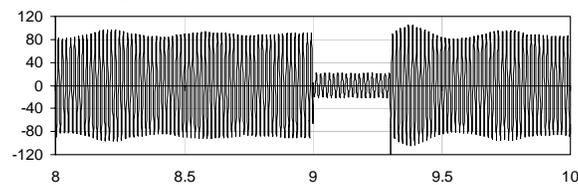
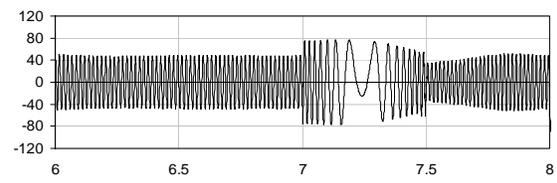
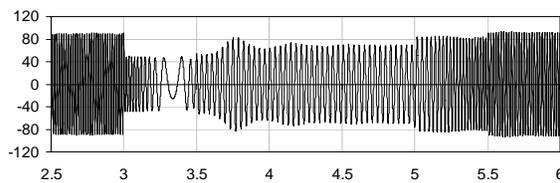
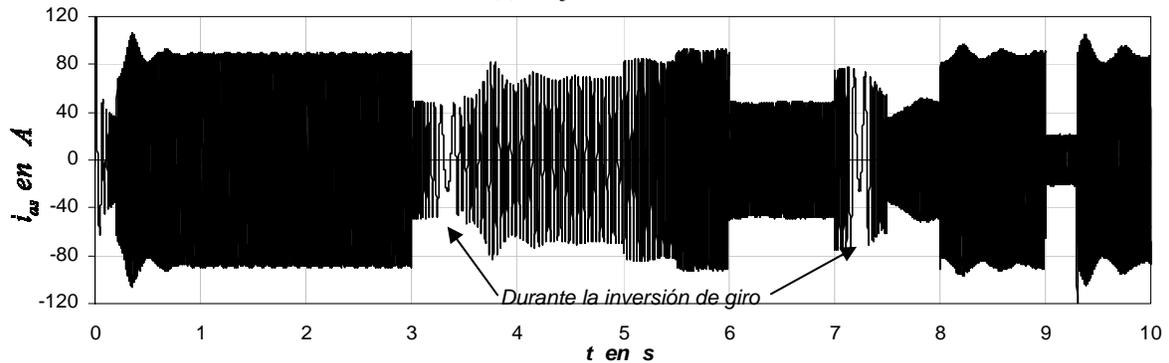
(a) Velocidad mecánica del rotor, ω_{rm} .



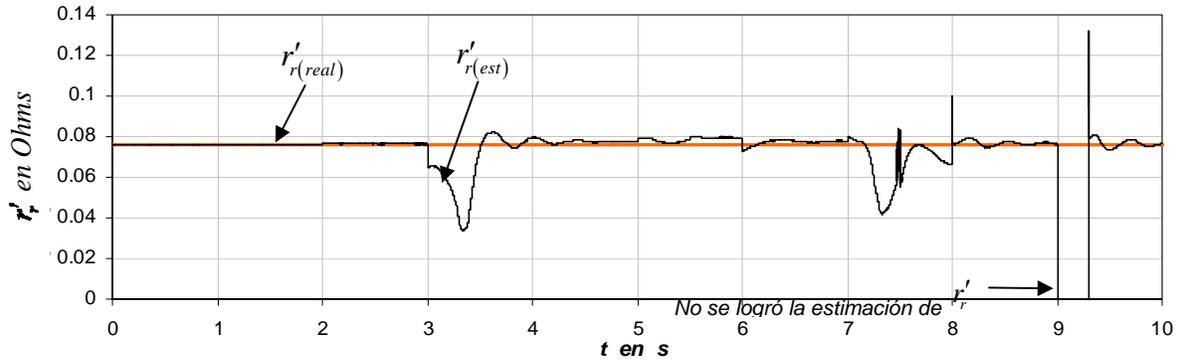
(b) Par electromagnético, T_{em} .



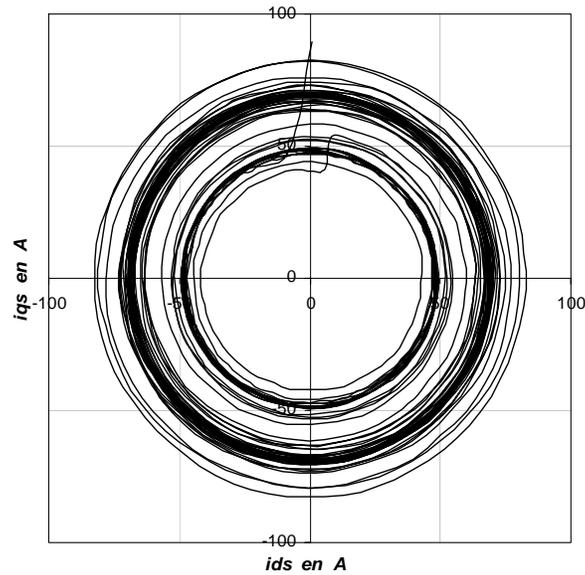
(c) Flujo del rotor.



(d) Corriente de la fase a del estator, i_{as} .

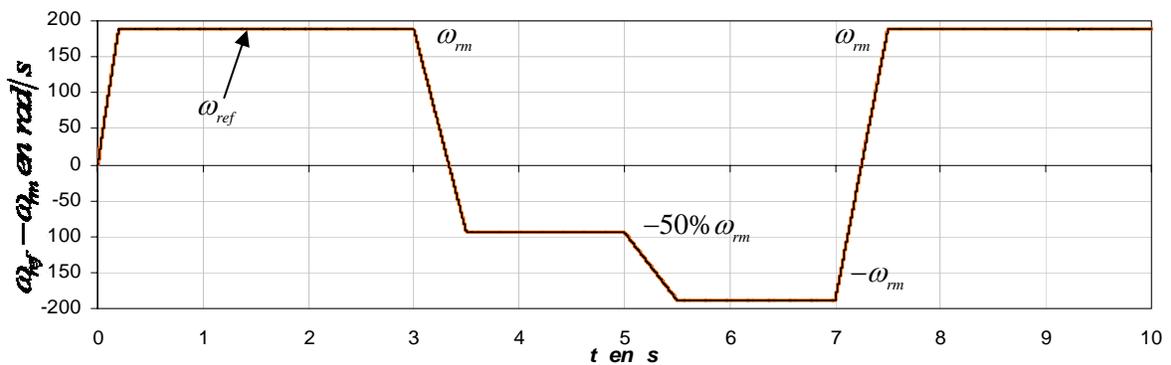


(e) Resistencia del rotor estimada.

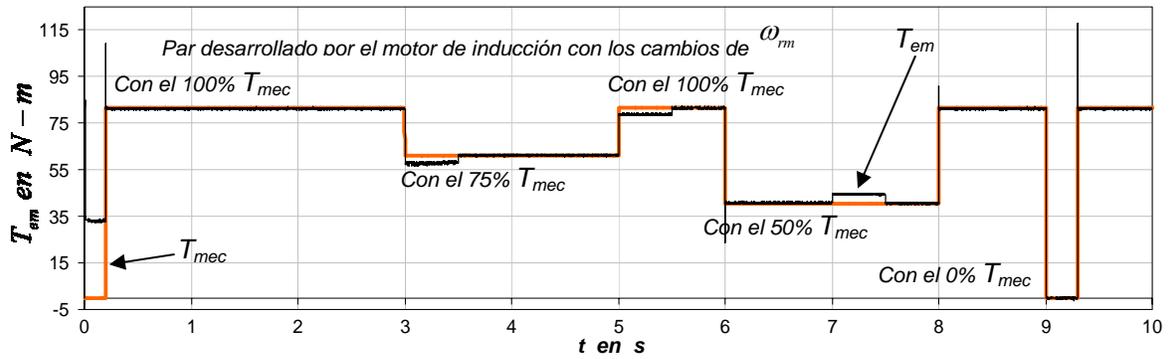


(f) Trayectoria del vector de corriente del estator.

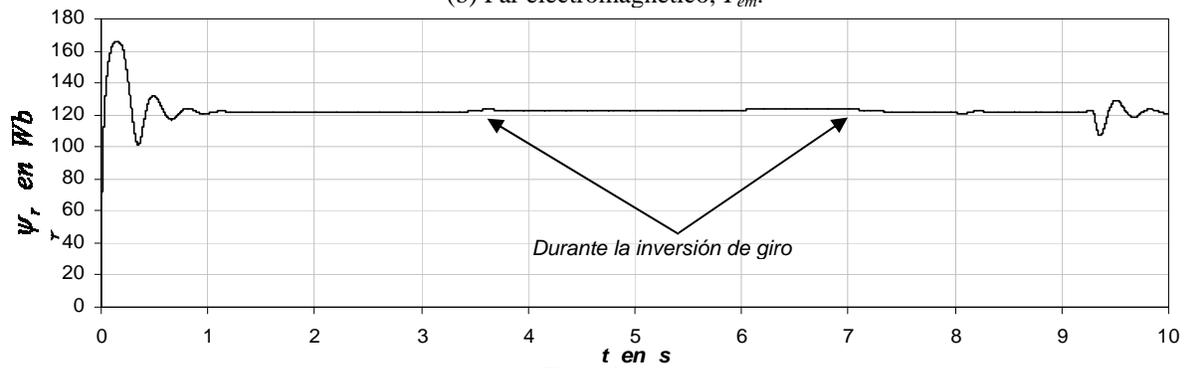
Figura 4.14 Estimación de la resistencia del rotor en estado transitorio con el estimador basado en un controlador difuso y comportamiento de las variables del control vectorial indirecto del motor de inducción. Tercer caso transitorio.



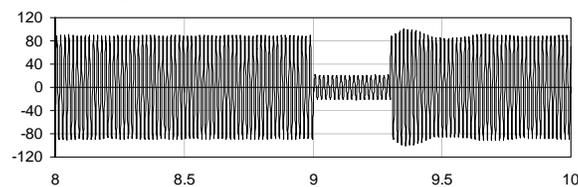
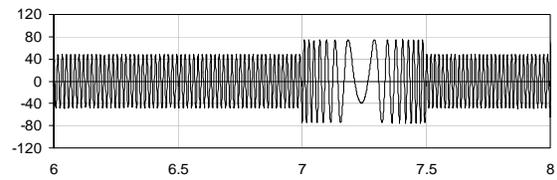
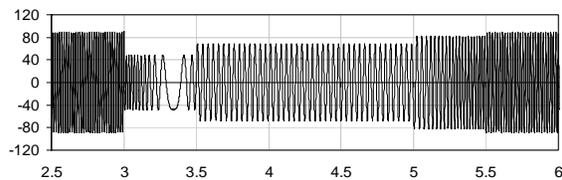
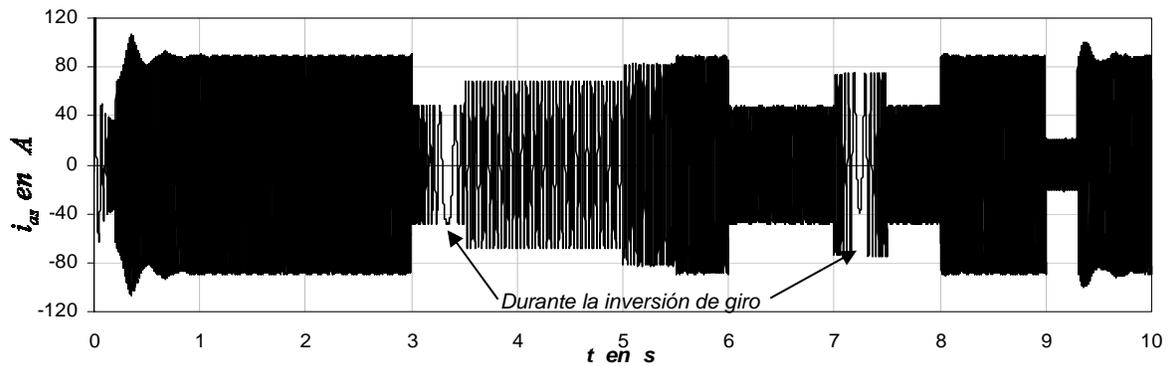
(a) Velocidad mecánica del rotor, ω_m .



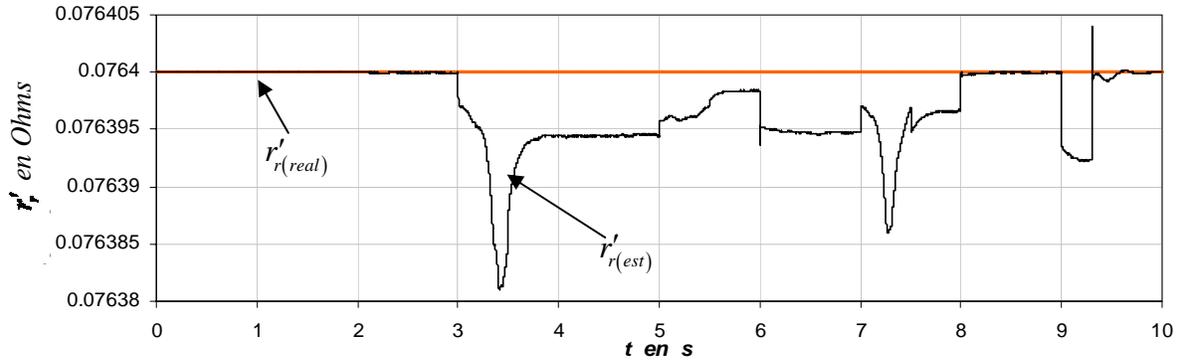
(b) Par electromagnético, T_{em} .



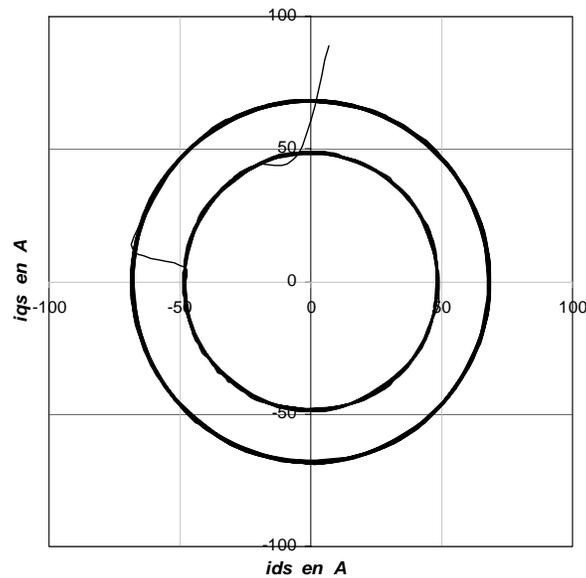
(c) Flujo del rotor.



(d) Corriente de la fase a del estator, i_{as} .



(e) Resistencia del rotor estimada.



(f) Trayectoria del vector de corriente del estator.

Figura 4.15 Estimación de la resistencia del rotor en estado transitorio con el estimador neuronal y comportamiento de las variables del control vectorial indirecto del motor de inducción. Tercer caso transitorio.

Con el fin de poder establecer diferencias entre el estimador basado en un controlador difuso (Fig. 4.14) y el estimador neuronal (Fig. 4.15) se presenta una inversión de marcha (Fig. 4.14(a) y Fig. 4.15(a)) frente a una variación del par de carga, manteniendo a la resistencia del rotor constante.

En la Fig. 4.14(e) se muestra el resultado de la resistencia del rotor estimada con el estimador basado en un controlador difuso, en la cual la resistencia estimada varía al establecerse los cambios en el par de carga y la inversión de giro, el error de la resistencia es de 4.69032×10^{-03} . Al igual que en el caso anterior, cuando el par de carga es cero (en $t = 9.0 \text{ s} - 9.3 \text{ s}$), se pierde el control, ya que en el transitorio que se genera la resistencia del rotor, toma valores muy altos o llega a ser negativa, lo cual no puede ser posible.

Por otra parte, con el estimador neuronal la resistencia del rotor estimada de la Fig. 4.15(e), sigue a la resistencia del rotor real, por lo tanto, disminuye el error ya que su valor es de 2.89801×10^{-07} . De esta manera, al mantenerse prácticamente constante el valor de la

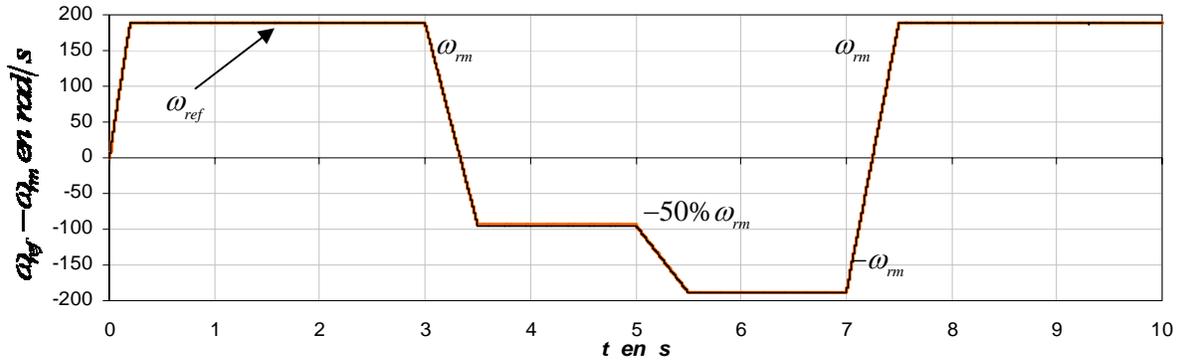
resistencia del rotor, se obtiene que el flujo del rotor se mantenga sin distorsión como se puede observar en la Fig. 4.15(c) en contraste con el flujo del rotor de la Fig. 4.14(c). Finalmente, se ilustra la trayectoria del vector de corriente del estator en el intervalo de 3.0 s – 5.0 s, observándose que en la Fig. 4.14(f), el vector no sigue una sola trayectoria, como si sucede en la Fig. 4.15(f), donde se obtiene una trayectoria circular, indicando que la corriente trifásica es senoidal con menos distorsión (Fig. 4.15(d)), con respecto a la obtenida en la Fig. 4.14(d).

4.6.4 Cuarto caso transitorio

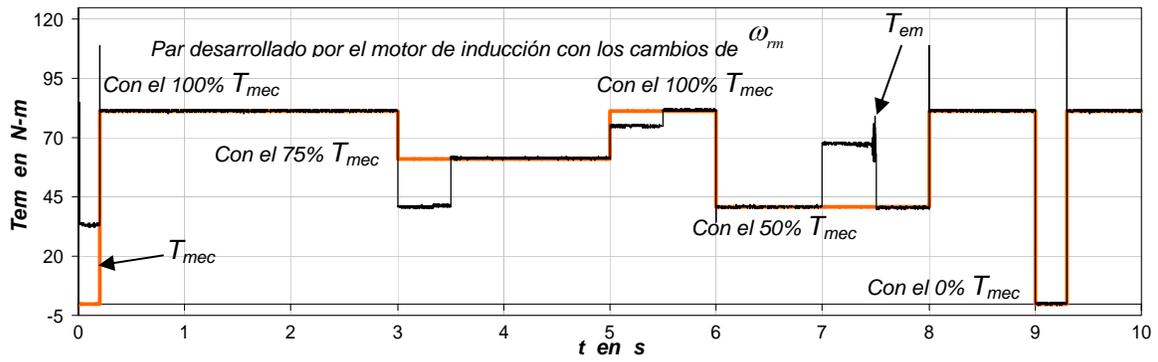
Par de carga y velocidad mecánica del rotor variables, se establece una variación desde el 100% al 150% en el valor de la resistencia del rotor. Por lo tanto, el archivo *m5.m* de MATLAB, establece las siguientes condiciones de funcionamiento:

<i>t_paro</i>	El tiempo de estudio a 10 segundos.
<i>tiempo_wref</i>	Lapso de tiempo para la velocidad de referencia: [0 0.20 3.0 3.5 5.0 5.5 7.0 7.5 10.0].
<i>vel_wref</i>	Secuencia de la velocidad de referencia: [0 ω_{ref} ω_{ref} $-0.5 \times \omega_{ref}$ $-0.5 \times \omega_{ref}$ $-1.0 \times \omega_{ref}$ $-1.0 \times \omega_{ref}$ ω_{ref} ω_{ref}].
<i>tiempo_tmec</i>	Lapso de tiempo para el par de carga: [0 0.20 0.20 3.0 3.0 5.0 5.0 6.0 6.0 8.0 8.0 9.0 9.0 9.3 9.3 10].
<i>tmec_tmec</i>	secuencia del par de carga: [0 0 $-T_{nom}$ $-T_{nom}$ $-0.75 \times T_{nom}$ $-0.75 \times T_{nom}$ $-T_{nom}$ $-T_{nom}$ $-0.5 \times T_{nom}$ $-0.5 \times T_{nom}$ $-T_{nom}$ $-T_{nom}$ 0 0 $-T_{nom}$ $-T_{nom}$].
<i>tiempo_Rrot</i>	Lapso de tiempo para la resistencia del rotor en el motor de inducción: [0 3.0 3.0 5.0 5.0 7.0 7.0 10].
<i>Rrot_Rrot</i>	Secuencia de la resistencia del rotor: [r'_r r'_r r'_r $1.2 \times r'_r$ $1.2 \times r'_r$ r'_r $1.5 \times r'_r$ $1.5 \times r'_r$].

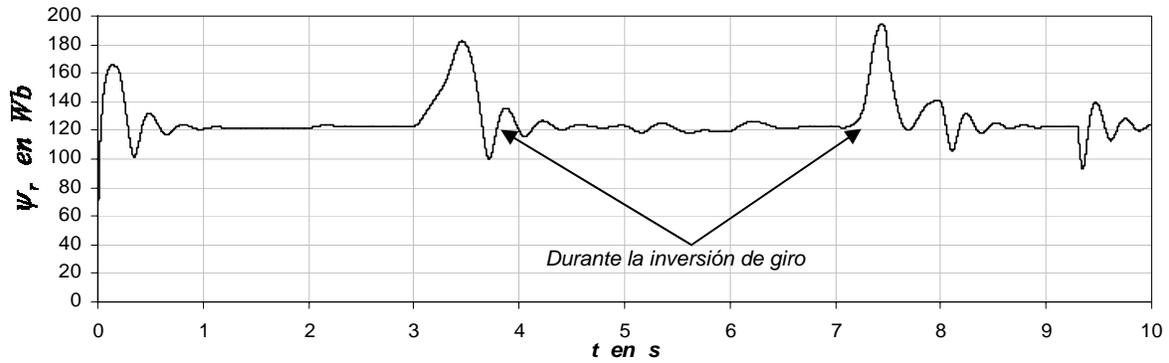
Las Figs. 4.16 y 4.17 muestran el desempeño de ambos estimadores en el control vectorial indirecto. En estas figuras se muestran los valores de la velocidad mecánica del rotor, ω_{rm} , el par electromagnético, T_{em} , la magnitud del flujo del rotor, Ψ_r , la corriente de la fase *a* del estator, i_{as} , la resistencia del rotor estimada, $r'_{r(est)}$ y la trayectoria del vector de corriente del estator. Los resultados de la simulación son considerando las condiciones del cuarto caso transitorio.



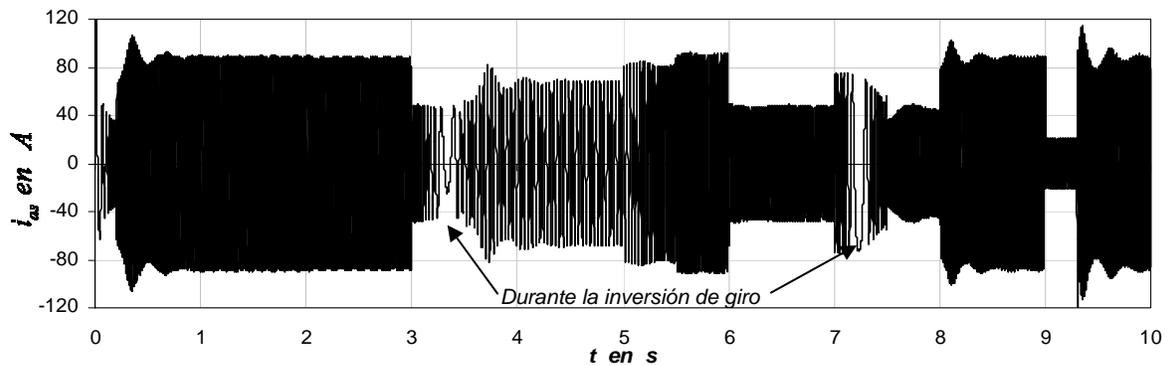
(a) Velocidad mecánica del rotor, ω_{rm} .

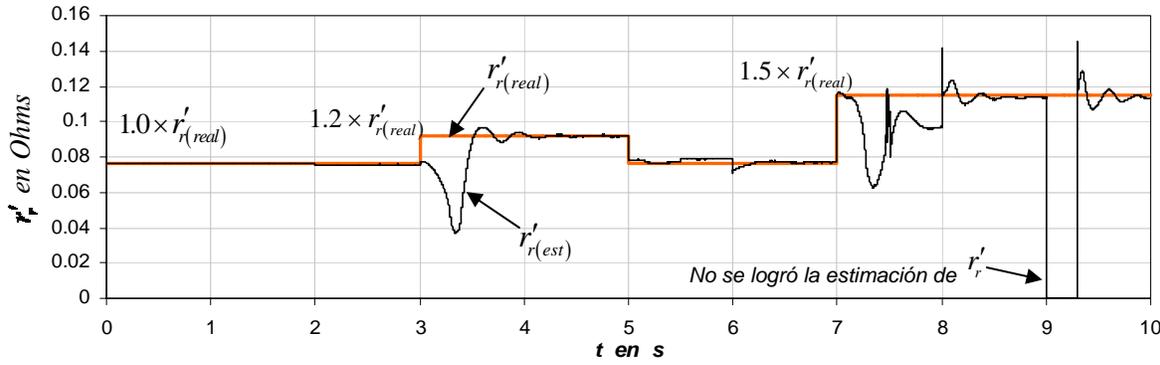
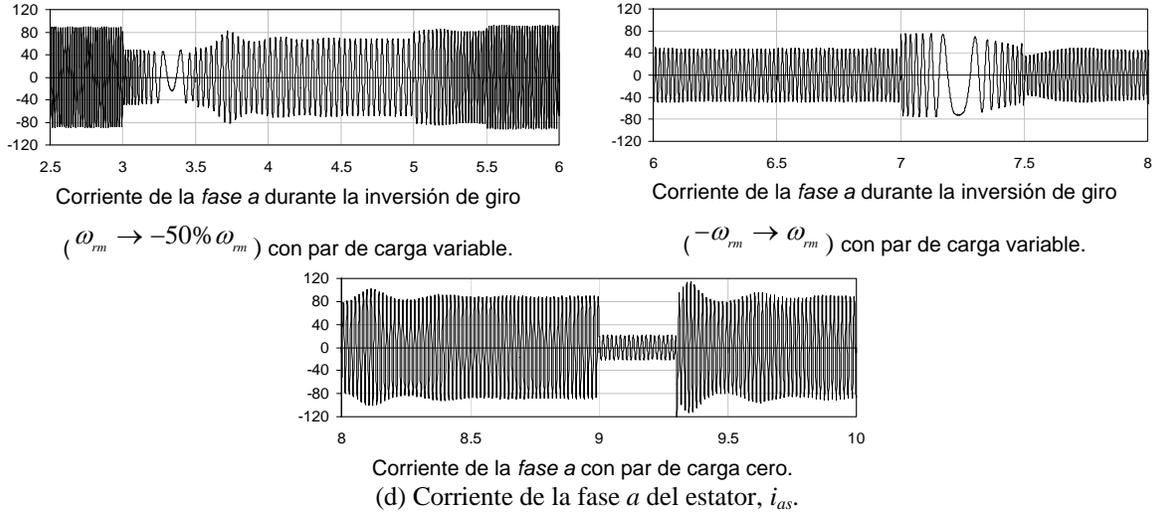


(b) Par electromagnético, T_{em} .

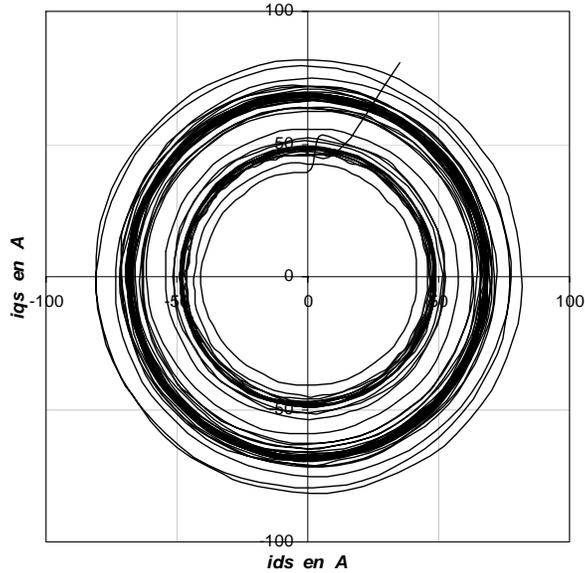


(c) Flujo del rotor.



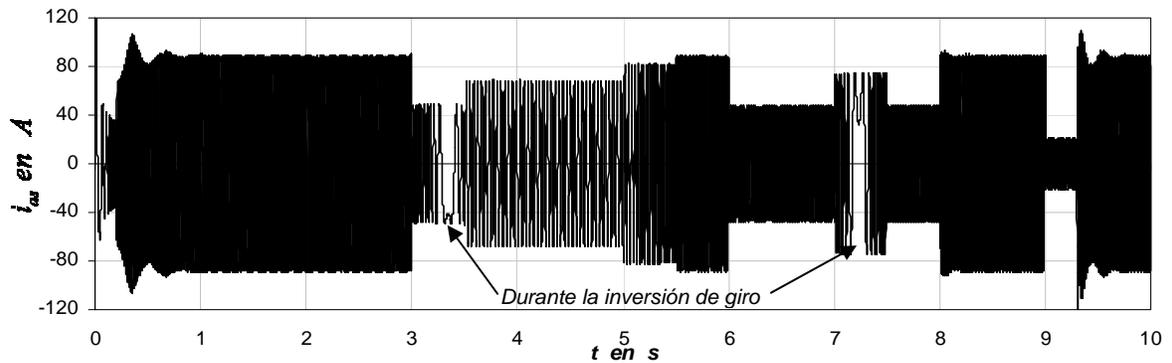
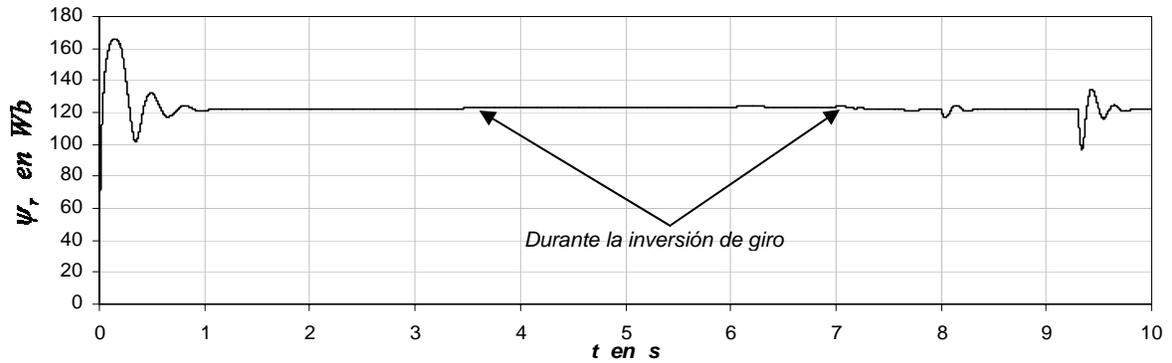
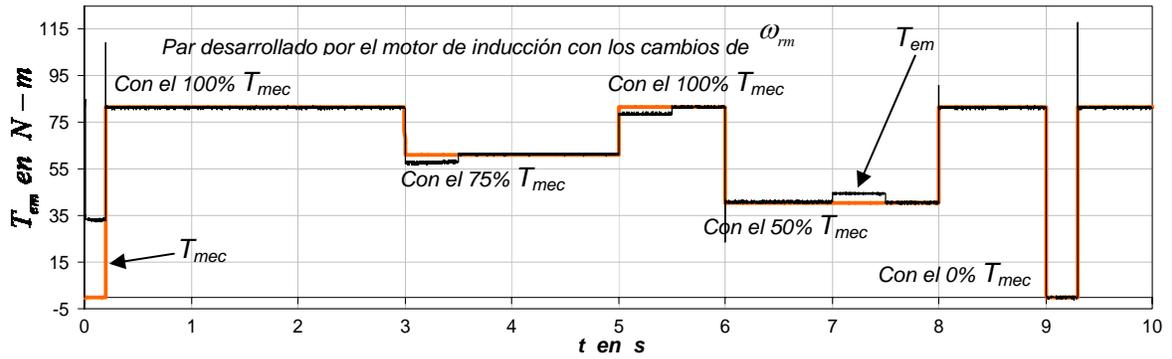
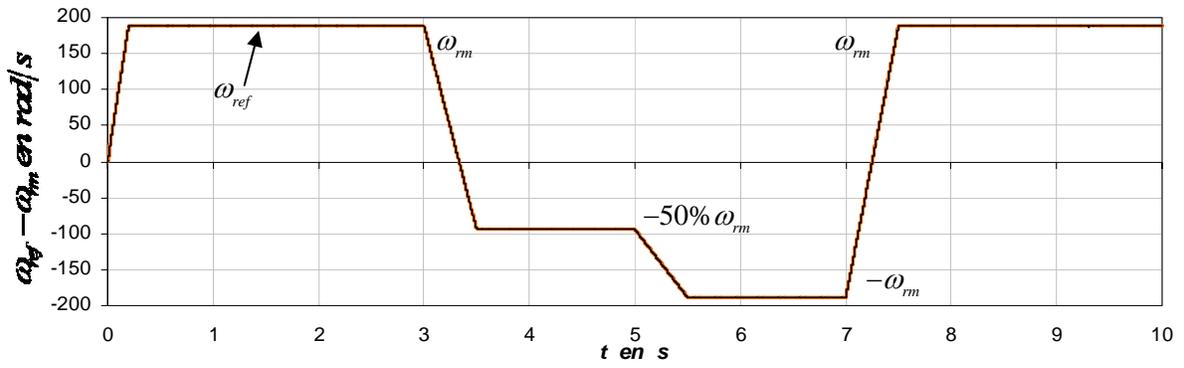


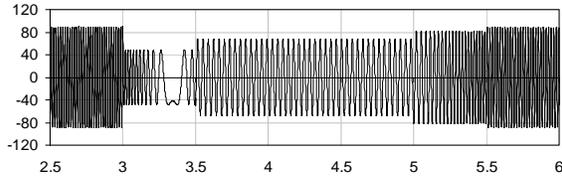
(e) Resistencia del rotor estimada.



(f) Trayectoria del vector de corriente del estator.

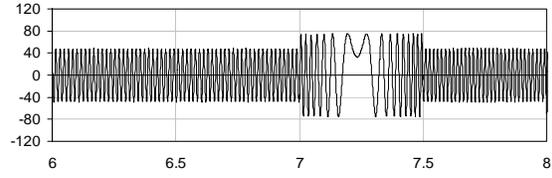
Figura 4.16. Estimación de la resistencia del rotor en estado transitorio con el estimador basado en un controlador difuso y comportamiento de las variables del control vectorial indirecto del motor de inducción. *Cuarto caso transitorio.*





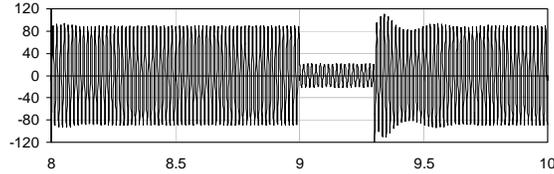
Corriente de la fase a durante la inversión de giro

($\omega_{rm} \rightarrow -50\% \omega_{rm}$) con par de carga variable.



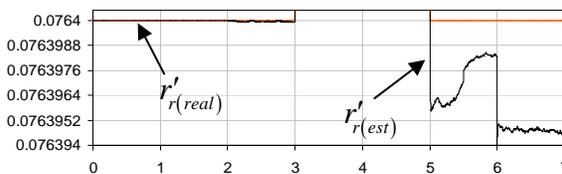
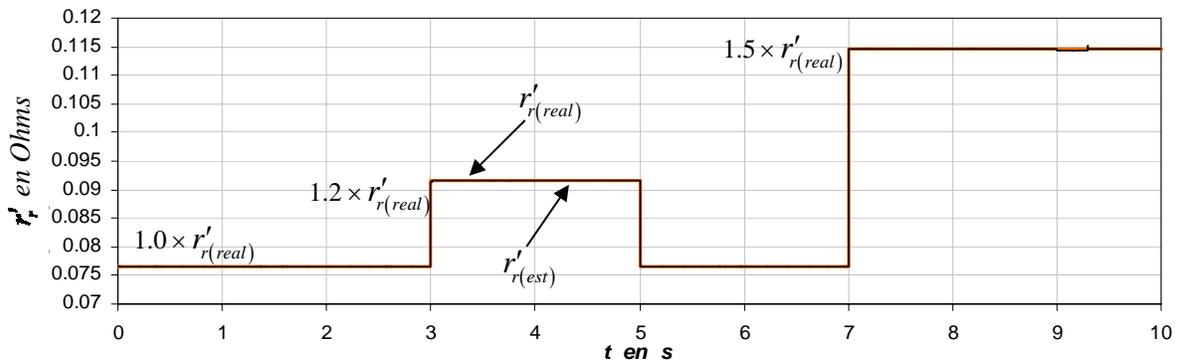
Corriente de la fase a durante la inversión de giro

($-\omega_{rm} \rightarrow \omega_{rm}$) con par de carga variable.



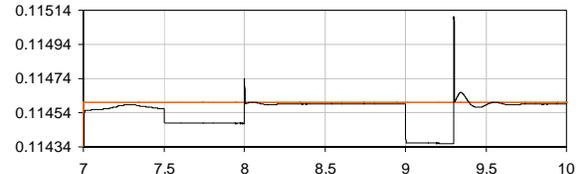
Corriente de la fase a con par de carga cero.

(d) Corriente de la fase a del estator, i_{as} .



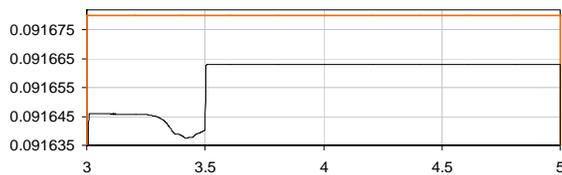
Resistencia del rotor en el intervalo de 0-2 s y de 5-6 s, se

realiza la estimación considerando que vale $1.0 \times r_r'$.



Resistencia del rotor en el intervalo de 7-10 s, se realiza la

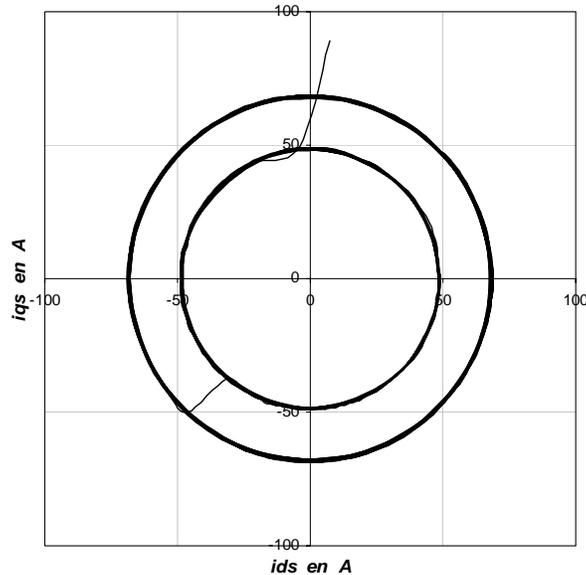
estimación considerando que vale $1.5 \times r_r'$.



Resistencia del rotor en el intervalo de 3-7 s, se realiza la

estimación considerando que vale $1.2 \times r_r'$.

(e) Resistencia estimada del rotor.



(f) Trayectoria del vector de corriente del estator.

Figura 4.17 Estimación de la resistencia del rotor en estado transitorio con el estimador neuronal y comportamiento de las variables del control vectorial indirecto del motor de inducción. *Cuarto caso transitorio.*

Finalmente, se presenta una inversión de giro (Fig. 4.16(a) y Fig. 4.17(a)) frente a una variación del par de carga, realizando cambios de paso en la resistencia del rotor para tres condiciones: 100%, 120% y 150% (Fig. 4.16(e) y Fig. 4.17(e)), para establecer diferencias entre el Estimador basado en un controlador difuso (Fig. 4.16) y el estimador neuronal (Fig. 4.17).

Se considera el resultado de la Fig. 4.15(e) donde se muestra la resistencia del rotor estimada con el Estimador basado en un controlador difuso, se observa que la resistencia estimada varía al establecerse los cambios en el par de carga y la inversión de giro, obteniendo un error de 8.27818×10^{-03} . De hecho, cuando el par de carga es cero (en $t = 9.0 \text{ s} - 9.3 \text{ s}$), se pierde el control, ya que en el transitorio que se genera la resistencia del rotor toma valores muy altos o llega a ser negativa, lo cual no puede ser posible.

Por otra parte, con el estimador neuronal la resistencia del rotor estimada de la Fig. 4.17(e), sigue a la resistencia del rotor real, como se observa en el acercamiento que se hace a la resistencia del rotor estimada, por lo tanto, se minimiza el error siendo su valor igual a 3.57323×10^{-07} . De esta manera, al mantenerse prácticamente constante el valor de la resistencia del rotor, se obtiene que el flujo del rotor se mantenga sin distorsión como se puede observar en la Fig. 4.17(c) en contraste con el flujo del rotor de la Fig. 4.16(c). Finalmente, se ilustra la trayectoria del vector de corriente del estator en el intervalo de $3.0 \text{ s} - 5.0 \text{ s}$, observándose que en la Fig. 4.16(f), el vector no sigue una sola trayectoria, como si sucede en la Fig. 4.17(f), donde se obtiene una trayectoria circular, indicando que la corriente trifásica es senoidal con menos distorsión (Fig. 4.17(d)), con respecto a la obtenida en la Fig. 4.16(d).

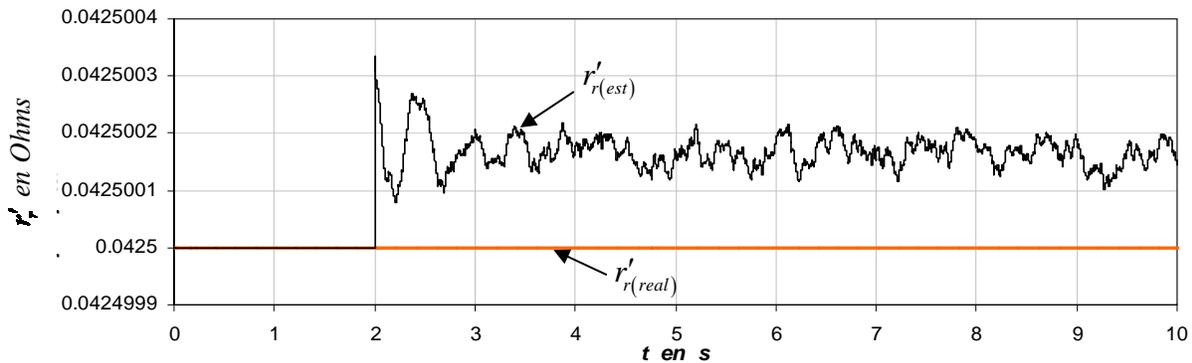
4.7 Resultados de la simulación del estimador neuronal con otros motores de inducción

La simulación se realiza con el mismo estimador neuronal que se utilizó con el motor de inducción de 20 HP, solamente que se lleva a cabo la normalización de los datos de entrada a la red neuronal artificial de la Fig. 4.3, en donde se consideran los nuevos valores máximos del motor de inducción de 100 HP, 1 HP y 5 HP. A continuación solamente se muestran los resultados de la resistencia del rotor estimada con diferentes condiciones de operación del motor de inducción.

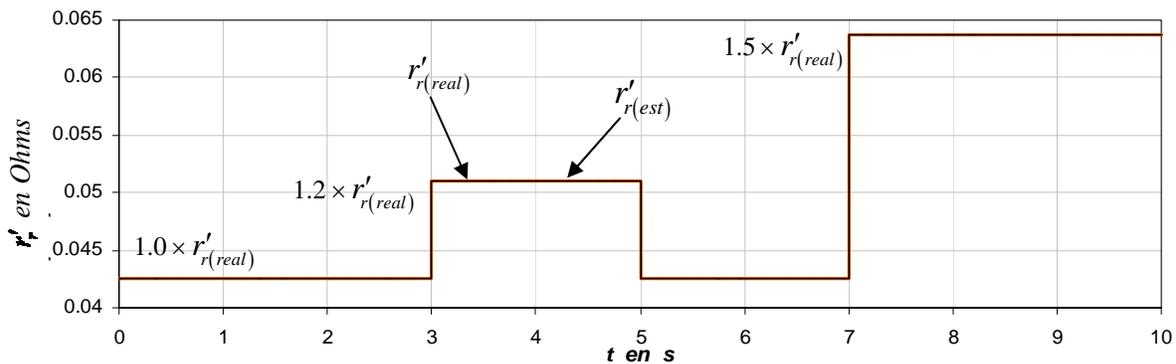
4.7.1 Simulación con un motor de inducción de 100 HP

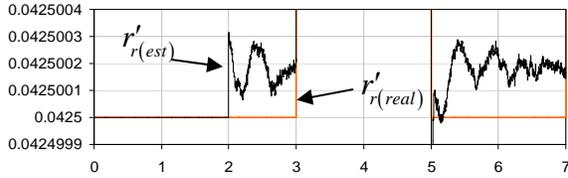
Se realiza la simulación del control vectorial indirecto del motor de inducción con PWM y regulador de corriente síncrono que usa el estimador neuronal para la estimación de la resistencia del rotor, en donde se utiliza un motor de 100 HP, trifásico, 60 Hz, 4 polos, 460 V; se declaran los parámetros en el archivo *m6.m* de MATLAB (ver Apéndice A), los cuales son:

$$\begin{aligned} r_s &= 0.0425 \Omega & X_{ls} &= 0.284 \Omega \\ r_r' &= 0.0425 \Omega & X_{lr}' &= 0.284 \Omega \\ X_m &= 8.51 \Omega & J &= 2.0 \text{ kgm}^2 \end{aligned}$$

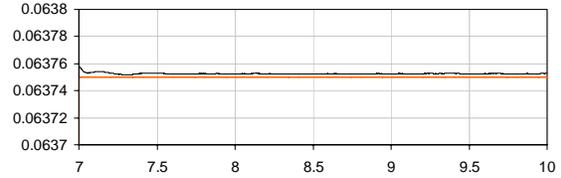


(a) Resistencia del rotor estimada, con velocidad mecánica del rotor y par de carga nominal, sin variación en el valor de la resistencia del rotor.

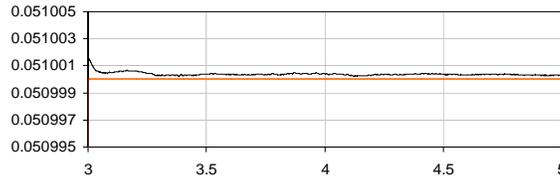




Resistencia del rotor en el intervalo de 0-2 s y de 5-6 s, se realiza la estimación considerando que vale $1.0 \times r'_r$.

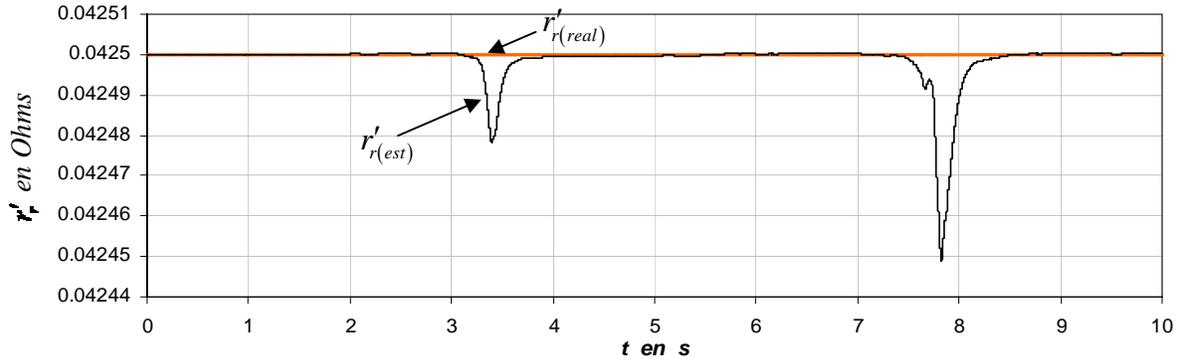


Resistencia del rotor en el intervalo de 7-10 s, se realiza la estimación considerando que vale $1.5 \times r'_r$.

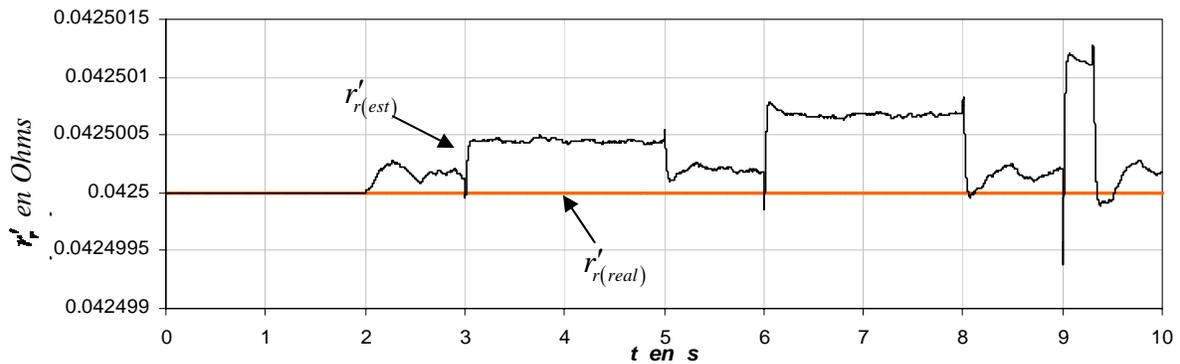


Resistencia del rotor en el intervalo de 3-7 s, se realiza la estimación considerando que vale $1.2 \times r'_r$.

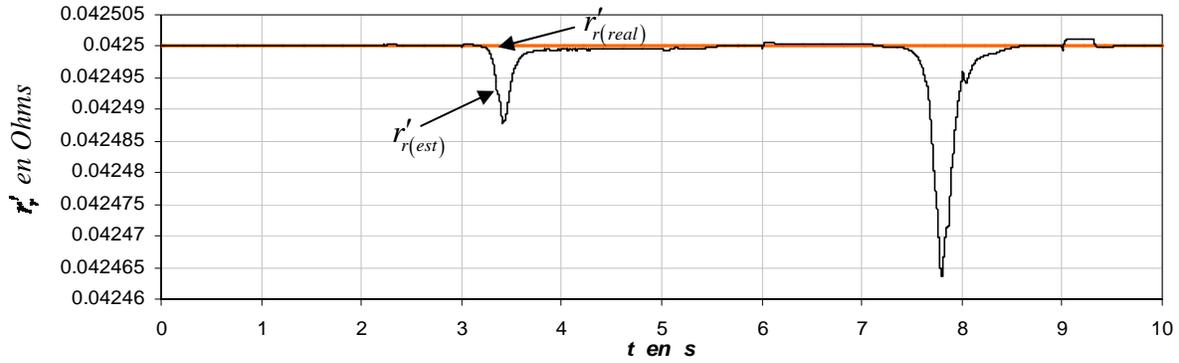
(b) Resistencia del rotor estimada, con velocidad mecánica del rotor y par de carga nominal, se establece una variación desde el 100% al 150% en el valor de la resistencia del rotor.



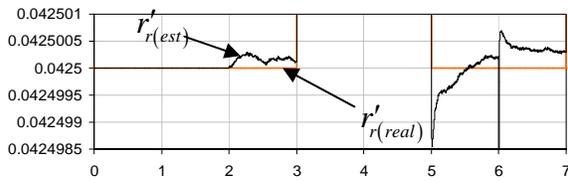
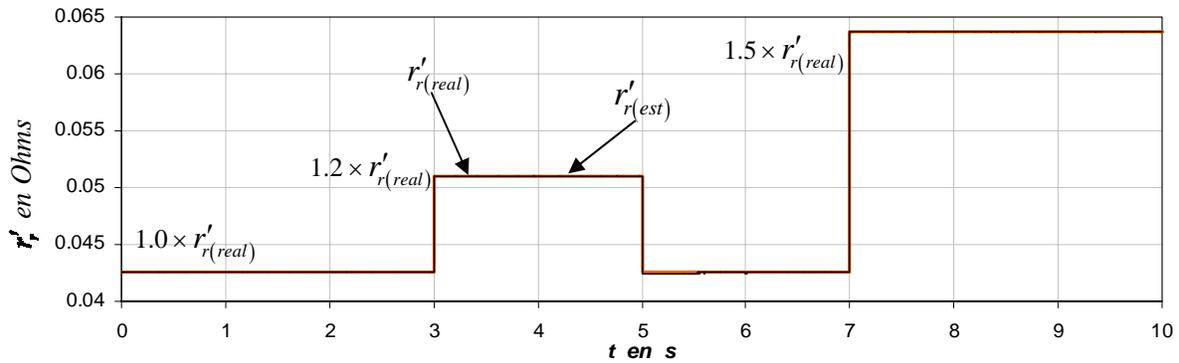
(c) Resistencia del rotor estimada, con velocidad mecánica del rotor variable y par de carga nominal sin variación en el valor de la resistencia del rotor.



(d) Resistencia del rotor estimada, con velocidad mecánica del rotor nominal y par de carga variable sin variación en el valor de la resistencia del rotor.

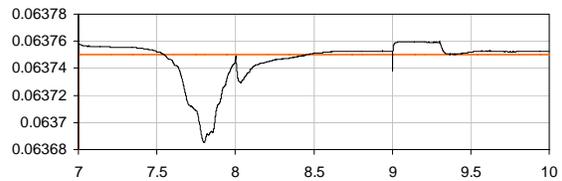


(e) Resistencia del rotor estimada, con velocidad mecánica del rotor y par de carga variable, sin variación en el valor de la resistencia del rotor.



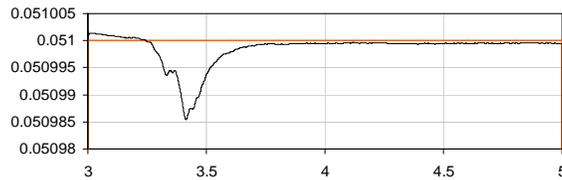
Resistencia del rotor en el intervalo de 0 - 2 s y de 5 - 6 s, se

realiza la estimación considerando que vale $1.0 \times r_r'$.



Resistencia del rotor en el intervalo de 7 - 10 s, se realiza

la estimación considerando que vale $1.5 \times r_r'$.



Resistencia del rotor en el intervalo de 3 - 7 s, se realiza la

estimación considerando que vale $1.2 \times r_r'$.

(f) Resistencia del rotor estimada, con velocidad mecánica del rotor y par de carga variable, se establece una variación desde el 100% al 150% en el valor de la resistencia del rotor.

Figura 4.18 Estimación de la resistencia del rotor con el estimador neuronal bajo diferentes condiciones de operación en el motor de inducción de 100 HP.

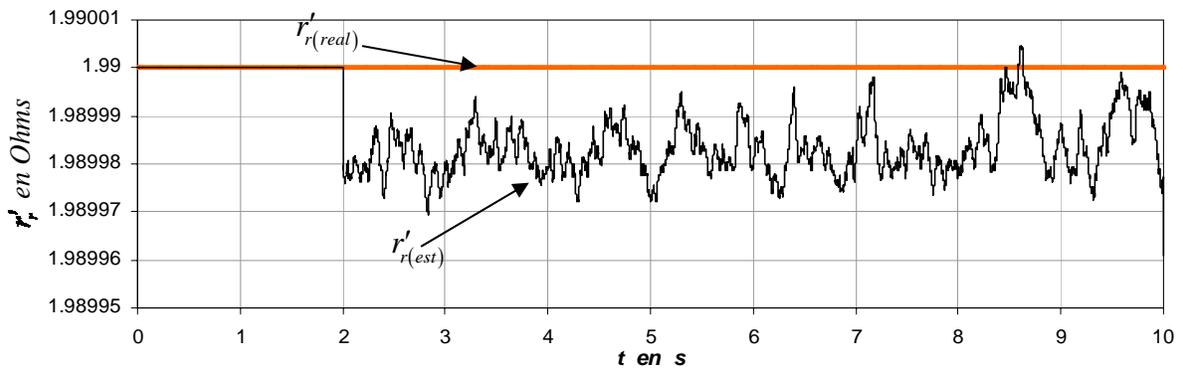
Como se puede observar en la Fig. 4.18(a) - (b), el comportamiento del estimador neuronal durante el proceso de estimación de la resistencia del rotor, proporcione resultados satisfactorios. Hay que resaltar que se utilizó la misma red neuronal artificial entrenada para el motor de inducción de 20 HP, por lo tanto, al cambiar el vector de

entradas, se cambian las condiciones generales de la red, pero realizando la normalización del vector de entradas, el proceso de estimación sigue siendo igual.

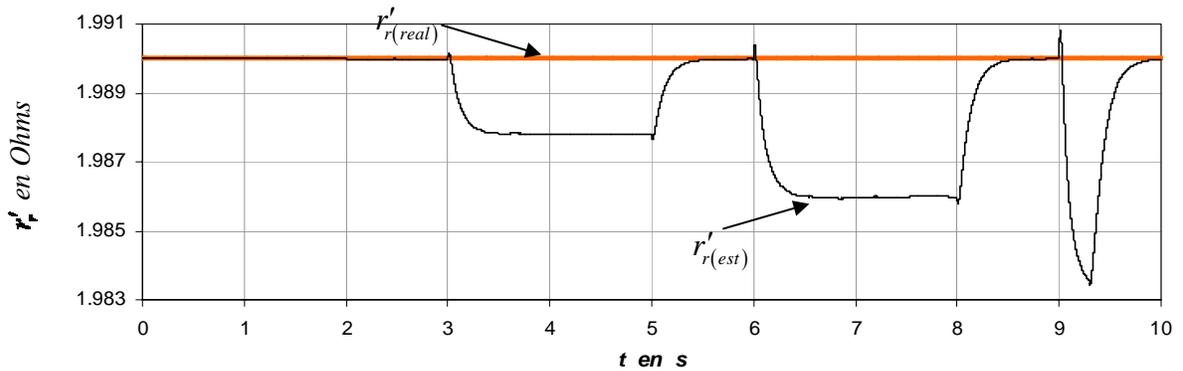
4.7.2 Simulación con un motor de inducción de 1 HP

Ahora se realiza la simulación para la estimación de la resistencia del rotor utilizando un motor de inducción de 1 HP, trifásico, 60 Hz, 4 polos, 200 V, en donde se usa el estimador neuronal. Los parámetros de este motor son:

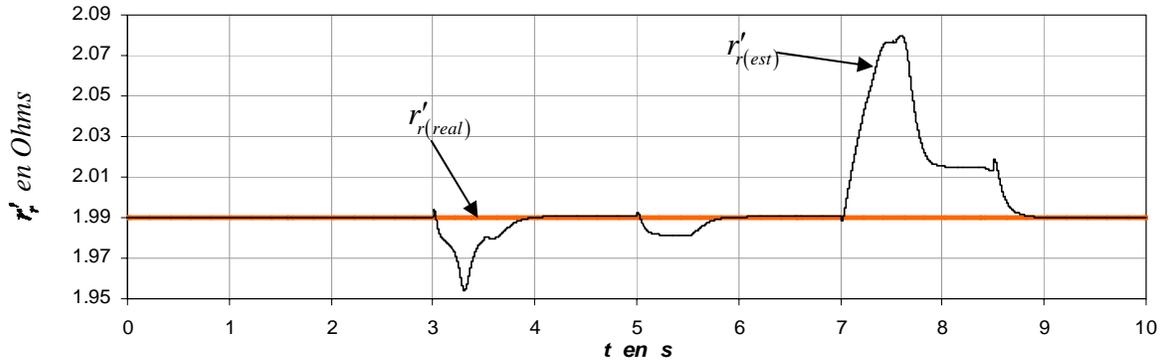
$$\begin{aligned} r_s &= 3.35 \, \Omega & X_{ls} &= 6.94 \times 10^{-3} \, \Omega \\ r_r' &= 1.99 \, \Omega & X_{lr}' &= 6.94 \times 10^{-3} \, \Omega \\ X_m &= 163.73 \times 10^{-3} \, \Omega & J &= 0.1 \, \text{kgm}^2 \end{aligned}$$



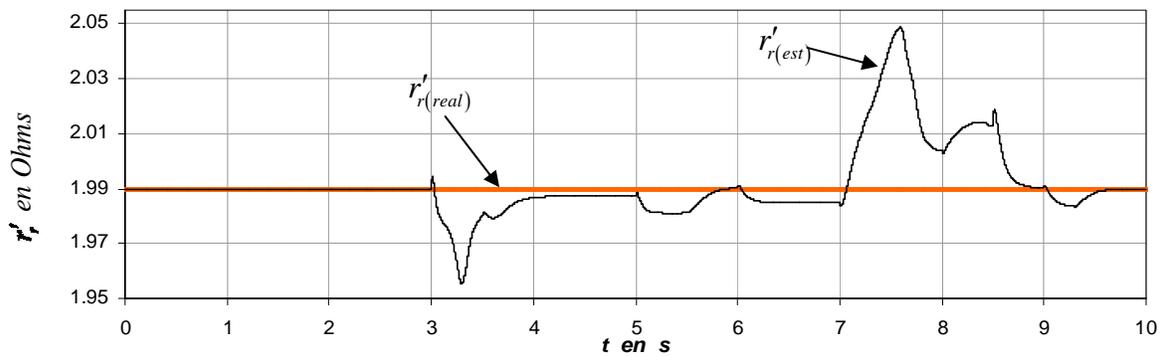
(a) Resistencia del rotor estimada, con velocidad mecánica del rotor y par de carga nominal.



(b) Resistencia del rotor estimada, con velocidad mecánica del rotor nominal y par de carga variable.



(c) Resistencia del rotor estimada, con velocidad mecánica del rotor variable y par de carga nominal.



(d) Resistencia del rotor estimada, con velocidad mecánica del rotor y par de carga variable.

Figura 4.19 Estimación de la resistencia del rotor con el estimador neuronal bajo diferentes condiciones de operación en el motor de inducción de 1 HP.

4.7.3 Simulación con un motor de inducción de 5 HP

Finalmente se realiza la simulación para la estimación de la resistencia del rotor utilizando un motor de inducción de 5 HP, trifásico, 60 Hz, 4 polos, 230 V, en donde se usa el estimador neuronal. Los parámetros de este motor son:

$$r_s = 0.531 \Omega$$

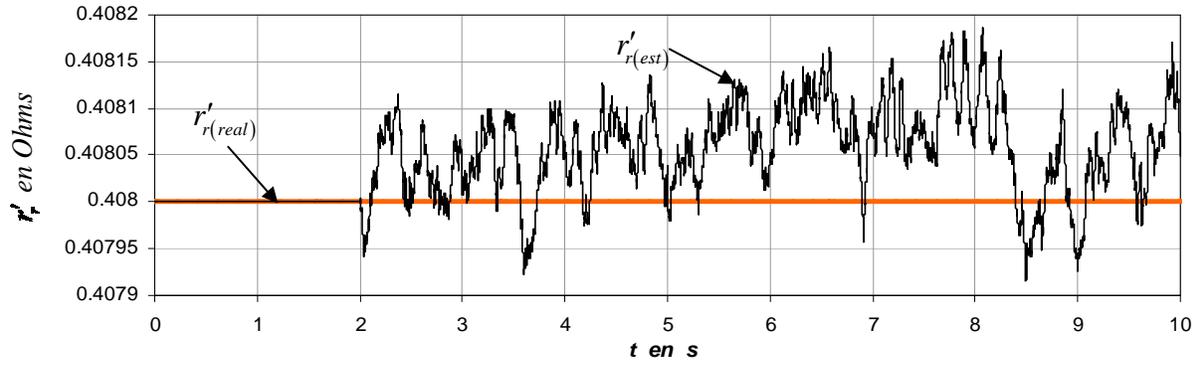
$$X_{ls} = 0.95 \Omega$$

$$r'_r = 0.408 \Omega$$

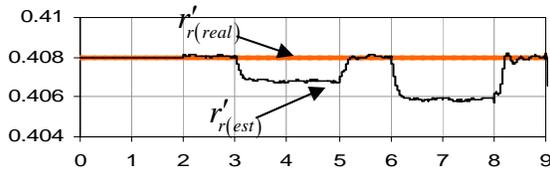
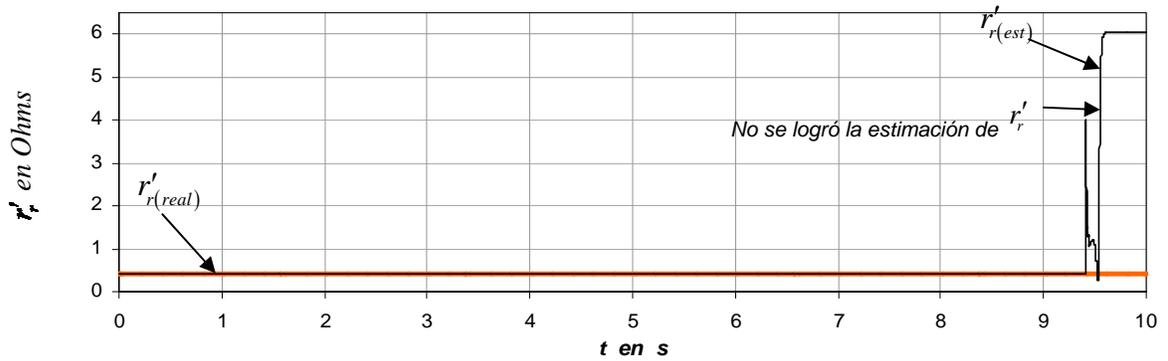
$$X'_{lr} = 0.95 \Omega$$

$$X_m = 31.931 \Omega$$

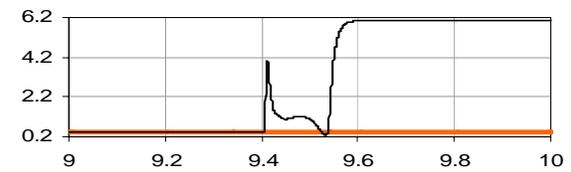
$$J = 0.1 \text{ kgm}^2$$



(a) Resistencia del rotor estimada, con velocidad mecánica del rotor y par de carga nominal.

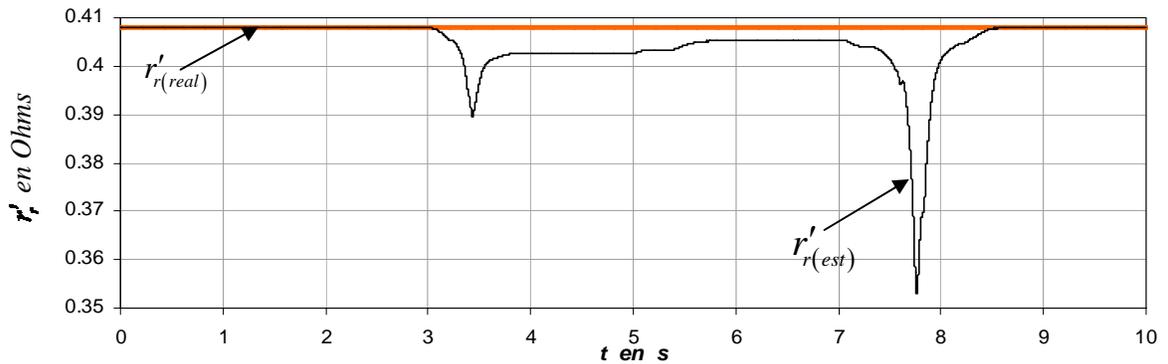


Resistencia del rotor en el intervalo de 0 - 9 s, se realiza la estimación con par de carga variable.

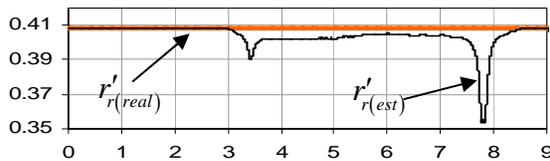
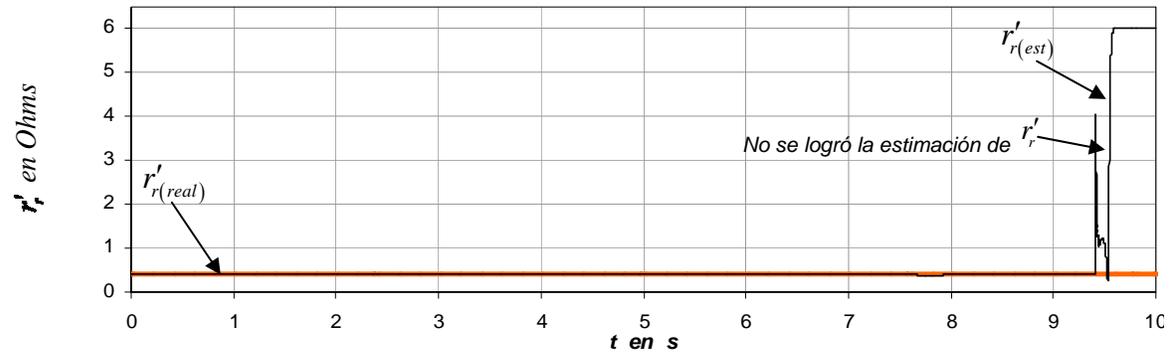


Resistencia del rotor en el intervalo de 9 - 10 s, se realiza la estimación con par de carga variable.

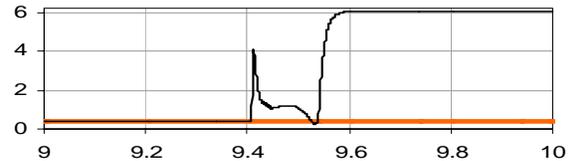
(b) Resistencia del rotor estimada, con velocidad mecánica del rotor nominal y par de carga variable.



(c) Resistencia del rotor estimada, con velocidad mecánica del rotor variable y par de carga nominal.



Resistencia del rotor en el intervalo de 0 - 9 s, se realiza la estimación con par de carga y velocidad variables.



Resistencia del rotor en el intervalo de 9 - 10 s, se realiza la estimación con par de carga y velocidad variables.

(d) Resistencia del rotor estimada, con velocidad mecánica del rotor y par de carga variable.

Figura 4.20 Estimación de la resistencia del rotor con el estimador neuronal bajo diferentes condiciones de operación.

La estimación de la resistencia del rotor en los motores de inducción de 1 HP y 5 HP, como se observa, se realiza de manera adecuada cuando el motor opera en estado estable (Figs. 4.19(a) y 4.20 (a)), obteniéndose por lo tanto, el valor correcto de la resistencia del rotor y de esta manera no se altera el funcionamiento del motor. Sin embargo, cuando se varia la velocidad mecánica del rotor y/o el par de carga, la resistencia del rotor estimada tiende a variar con los cambios que se realizan en las condiciones de operación del motor (Figs. 4.19(b)-(d) y 4.20 (b)-(d)). Una muestra de lo anterior, se observa en las Figs. 4.20(b) y (d), en donde el motor de inducción opera con la velocidad nominal y se quita la carga en 9.0 s, la cual se vuelve aplicar a los 9.3 s estimándose una resistencia del rotor de aproximadamente 6.0Ω , alterando el funcionamiento del motor y el comportamiento de las variables en el control vectorial indirecto.

Para estos dos motores, también se efectuó la estimación de la resistencia del rotor con la misma red neuronal entrenada para el motor de inducción de 20 HP, en donde como ya se mencionó, al cambiar el vector de entradas, se cambian las condiciones generales de la red, pero realizando la normalización del vector de entradas, el proceso de entrenamiento sigue siendo el mismo, pero para estos motores el estimador neuronal no proporciona los valores adecuados de la resistencia del rotor.

Capítulo 5

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

La aportación principal de esta tesis es el desarrollo de un estimador de la resistencia del rotor basado en una red neuronal artificial, el cual opera con el control vectorial indirecto en donde se actualiza este parámetro una vez estimado, dependiendo de la operación del motor de inducción.

Como se observa en el Capítulo 4, la estimación de la resistencia del rotor mediante el estimador basado en las funciones F_{est} , F_{act} y en un controlador difuso y en el estimador neuronal, se inicia desde los 0 s, pero es hasta los 2 s cuando se realiza la adaptación de la resistencia del rotor en el bloque de campo orientado, ya que a partir de los 2 s es cuando se varía la resistencia en el motor de inducción.

En base a los resultados de las simulaciones obtenidas con el estimador neuronal, se demuestra la efectividad de las redes neuronales artificiales para aplicarlas como estimadores en modelos no lineales. A su vez se concluye que la resistencia del rotor estimada sigue a la resistencia del rotor real, con lo cual el flujo del rotor se mantiene sin distorsión y en forma general el comportamiento de las variables del control vectorial indirecto es favorable para ambos estimadores ya que el motor de inducción de 20 HP trabaja en estado estable, pero en base a los errores calculados, el desempeño del estimador neuronal es sobresaliente. Asimismo, cuando se mantiene al motor de inducción en estado estable y se varía la resistencia del rotor del 100% al 150% de su valor, también se observa que la resistencia del rotor estimada se conserva constante, solamente cambiando en función al escalón de la resistencia del rotor real. Cabe destacar que la variación real de la resistencia del rotor no varía en forma de escalón, lo cual significa que lo hace de forma lenta, pero en la simulación se considera que la resistencia del rotor cambia en forma de escalón, es decir lo hace abruptamente, y aun así el estimador neuronal proporciona un valor adecuado de la resistencia del rotor.

Con respecto a los resultados obtenidos con las simulaciones de ambos estimadores, en donde se considera que el motor de inducción de 20 HP opera en estado transitorio, presentándose diferentes comportamientos de la velocidad mecánica del rotor y del par de carga, se demuestra nuevamente que el desempeño del estimador neuronal proporciona una resistencia del rotor con un error menor en relación a la resistencia del rotor real. Además, para obtener a la resistencia del rotor estimada, se toma en cuenta que la resistencia del rotor real se mantiene constante y posteriormente se realizan cambios en forma de escalón. De acuerdo a lo anterior, el comportamiento de las variables del control vectorial indirecto es satisfactorio.

Con el objetivo de observar el desempeño del estimador neuronal utilizado para el motor de inducción de 20 HP, se emplean otros motores de inducción con otras capacidades. Por lo tanto, de los resultados obtenidos con la simulación del estimador neuronal con el motor de inducción de 100 HP en estado estable y transitorio; en general se concluye que el comportamiento de las variables del control vectorial indirecto es adecuada, presentándose variaciones en la velocidad mecánica del rotor y en el flujo del rotor durante el momento en el cual se aplica la carga (0.5 s), asimismo, la resistencia del rotor estimada mantiene un error menor con respecto a la resistencia del rotor real, causando que la resistencia del rotor en el bloque de campo orientado se este actualizando al momento de presentarse otras condiciones de operación. Como la estimación se efectuó con la misma red neuronal artificial entrenada con el patrón de Entradas/Salidas del motor de inducción de 20 HP, es necesario realizar la normalización del vector de entradas, para que el proceso de estimación siga manteniéndose igual. De lo contrario, si se mantiene el mismo vector de entradas, se cambian las condiciones generales de la red.

También se realizó la estimación de la resistencia del rotor en los motores de inducción de 1 HP y 5 HP. Con lo cual se comprueba que la estimación se realiza de manera adecuada cuando los motores de inducción operan en estado estable, consiguiendo que la resistencia del rotor estimada siga a la resistencia del rotor real y de esta manera no se altera el funcionamiento del motor. Sin embargo, cuando se varía la velocidad mecánica del rotor y/o el par de carga, la resistencia del rotor estimada tiende a variar con los cambios que se realizan en las condiciones de operación del motor, alterando el funcionamiento del motor y el comportamiento de las variables en el control vectorial indirecto. Para estos dos motores, también se efectuó la estimación de la resistencia del rotor con la misma red neuronal artificial entrenada para el motor de inducción de 20 HP, sin embargo, el estimador neuronal no proporciono valores adecuados de la resistencia del rotor.

Finalmente, se concluye que la estructura del estimador neuronal, puede emplearse de manera sobresaliente en la estimación de la resistencia del rotor de otros motores de inducción que en el caso del motor de inducción de 100 HP solo fue necesario generar la normalización de los valores de entrada, como son: el flujo del rotor en el eje d , Ψ_{dr}^{e*} , la corriente bifásica síncrona de referencia, i_{ds}^{e*} , la velocidad del estator, ω_e , y los voltajes bifásicos síncronos, v_{ds}^e , v_{qs}^e , así como las corrientes bifásicas síncronas, i_{ds}^e , i_{qs}^e . Pero para otros motores de inducción, por ejemplo el de 1 HP y 5 HP, aunque se realizó la normalización de los valores de entrada, no fue posible obtener un desempeño adecuado del estimador neuronal, por lo tanto, es indispensable obtener distintos patrones de datos ya sea de entrada y de salida, considerando diferentes condiciones de operación del motor de inducción, para realizar el entrenamiento de la red neuronal artificial y de esta manera obtener la estimación de la resistencia del rotor de manera satisfactoria.

Un resultado importante del trabajo fue su presentación en el congreso internacional: The 40th International Universities Power Engineering Conference (UPEC-2005), en Cork, Irlanda. En el Apéndice D se muestra la ponencia presentada en el congreso.

5.2 Recomendaciones a trabajos futuros

Este trabajo constituye un acercamiento al desarrollo de técnicas de estimación de la resistencia del rotor en el control vectorial indirecto de motores de inducción mediante un estimador neuronal. Como el objetivo de la tesis es desarrollar la simulación del estimador, entonces no se ha podido abarcar hasta la implementación de esta técnica de estimación en el control vectorial indirecto. Por lo tanto, la continuidad y profundización de este trabajo en el futuro constituye un interés de los profesores, investigadores y estudiantes vinculados a este tema, por lo que se proponen las siguientes recomendaciones para darle un seguimiento al mismo:

- 1- En la literatura existen varias técnicas de regulación de corriente, en esta tesis solamente se considero el regulador síncrono en el control vectorial indirecto y posteriormente se realizo la estimación de la resistencia del rotor, por lo tanto, es conveniente realizar la simulación del estimador neuronal con otras técnicas de regulación de corriente para comprobar su desempeño.
- 2- Se recomienda realizar la implementación de los algoritmos del control vectorial indirecto y del estimador neuronal de la resistencia del rotor.
- 3- Asimismo, se recomienda seleccionar el software que permita pasar de los programas desarrollados en SIMULINK al ambiente con el que trabajará el Procesador Digital de Señales (DSP), como puede ser el DSPACE.
- 4- Continuar integrando al sistema otros algoritmos relacionadas a nuevas técnicas de generación de PWM y al desarrollo de estimadores de otros parámetros que afectan el desempeño del control vectorial indirecto, mediante adaptación de parámetros en línea, entre otros.

Apéndice A

PROGRAMAS EN MATLAB

A.1 Programa *m1.m*: Parámetros de la maquina de inducción trifásica de 20 HP

```
Ts=10e-6; %Tamaño del paso fijo

% ***** VALORES NOMINALES *****
Sb = 20*746; % Potencia nominal en VA
Vrated = 220; % Voltaje nominal de línea-línea en V
pf = 0.853; % Factor de potencia nominal
Irated = Sb/(sqrt(3)*Vrated*pf); % Corriente nominal rms en A
P= 4; % Numero de polos
frated = 60; % Frecuencia nominal en Hz

% ***** VALORES BASE *****
wb = 2*pi*frated; % Frecuencia eléctrica base en rad/s
we = wb;
wbm = 2*wb/P; % Frecuencia mecánica base en rad/s
Tb = Sb/wbm; % Par base
Zb = Vrated*Vrated/Sb; % Impedancia base en Ohms
Vm = Vrated*sqrt(2/3); % Magnitud del voltaje de fase en V
Vb = Vm;
Tfactor = (3*P)/(4*wb); % Factor para la expresión del par

% ***** ESPECIFICACIONES *****
srated = 0.0287; % Deslizamiento nominal
Nrated = 1748.3; % Velocidad nominal en rev/min
wmrated = 2*pi*Nrated/60; % Velocidad nominal en rad/s
Trated = Sb/wmrated; % Par nominal
iasb = 49.68; % Corriente de fase nominal rms en A

% ***** PARÁMETROS *****
rs = 0.1062; % Resistencia del devanado del estator en Ohms
Rs=rs;
xls = 0.2145; % Reactancia de enlace del estator en Ohms
xplr = xls; % Reactancia de enlace del rotor en Ohms
xm = 5.8339; % Reactancia de magnetización del estator en Ohms
rpr = 0.0764; % Resistencia referida al devanado del rotor en Ohms
xM = 1/(1/xm + 1/xls + 1/xplr);
```

```

J = 2.8; % Inercia del rotor en kg-m2
H = J*wbm*wbm/(2*Sb); % Constante de inercia en s
Domega = 0; % Coeficiente de amortiguamiento del rotor
Lls=xls/wb;
Lplr=Lls;
Lm=xm/wb;
LM=xM/wb;
Lf=1-((LM*LM)/(Lls*Lplr));

```

A.2 Programa *m2.m*: Parámetros de la maquina de inducción trifásica de 20 HP para el control vectorial indirecto.

% m2.m fija los parámetros de la maquina, los disturbios y se grafican los resultados.

```

clear all % Limpiar a workspace

% ***** PARÁMETROS DE LA MAQUINA DE 20 HP *****
m1.m

% ***** CÁLCULOS DE LAS CURVAS DE PAR VELOCIDAD *****
vas = Vrated/sqrt(3); % Especificar el voltaje fasorial rms
we = wb; % Especificar la frecuencia de excitación
xls = (we/wb)*xls; % Reactancias a la frecuencia de excitación
xplr = (we/wb)*xplr; % Reactancias a la frecuencia de excitación
xm = (we/wb)*xm; % Reactancias a la frecuencia de excitación

xM = 1/(1/xm + 1/xls + 1/xplr);
xs = xls + xm; % Auto-reactancia del estator
xr = xplr + xm; % Auto-reactancia del rotor
xsprime = xs - xm*xm/xr; % Reactancia transitoria del estator

% ***** Equivalente de Thevenin's *****
vth = abs((j*xm/(rs + j*(xls + xm)))*vas);
zth = (j*xm*(rs + j*xls)/(rs + j*(xls + xm)));
rth = real(zth);
xth = imag(zth);

% ***** Computar la resistencia del rotor *****
% Resistencia del rotor para Par max a s=1
rpr1 = sqrt(rth^2 + (xth + xplr)^2);
% rprm = 0.4*sqrt(rth^2 + (xth + xplr)^2);

% Estructuración vectorial de las resistencias del rotor
rprv = [rpr];
Nrr = length(rprv);
sx = (1:-0.01:0.02);

```

```

N = length(sx);
for n=1:N
    sn = sx(n);
    wr(n) = 2*we*(1-sn)/P;
    for nrr = 1:Nrr
        rrn = rprv(nrr);
        zin = (rs +j*xls) + j*xm*(rrn/sn + j*xplr)/(rrn/sn + j*(xm + xplr));
        ias = vas/zin;
        Sin = 3*vas*conj(ias);
        pin = real(Sin);
        pfin(nrr,n) = cos(-angle(ias));
        iin(nrr,n) = abs(ias);
        te(nrr,n) = (3*P/(2*we))*(vth^2*rrn/sn)/((rth + rrn/sn)^2 + (xth + xplr)^2);
        pe(nrr,n) = te(nrr,n)*wr(n);
        eff(nrr,n) = 100*pe(nrr,n)/pin;
    end % lazo para nrr
end % lazo para n

% Sumar en los valores de la velocidad síncrona
size(te);
z = [0];
inl = vas/(rs +j*(xls+xm));
inlm = abs(inl);
inla = cos(-angle(inl));
iin = [iin [inlm]'];
pfin = [pfin [inla]'];
eff = [eff z'];
te = [te z'];
pe = [pe z'];
sx = [sx 0];
wr = [wr 2*we/P];

% Determinar el flujo del rotor sin carga
eprime = vas - (rs +j*xsprime)*inl; % Voltaje entre la reactancia transitoria del estator
lambdadr = real(eprime)*(xr/xm)/we;

% ***** Estructurar los vectores de velocidad y flujo en la tabla lookup de simulación *****
speed = [-2: 0.1: 2];

mask = abs(speed)>1.1;
notmask = ~mask;
i = find(speed==0); % Encontrar el índice del elemento de velocidad cero y
speed(i) = realmin; % replácelo con un numero positivo utilizable mas pequeño
invspeed = abs(1./speed); % antes de dividirlo para evitar una división por cero

```

```

% Escala de los ejes de flujo y velocidad
lambdadre = lambdadr.*(invspeed.*mask + notmask);
speed = wbm*speed;
N = size(te);
M = size(te);
subplot(2,2,1)
plot(wr,te(1,:),'-')
xlabel('Velocidad del Rotor en rad/seg')
ylabel('Par en Nm')
subplot(2,2,2)
plot(wr,pe(1,:),'-')
xlabel('Velocidad del Rotor en rad/seg')
ylabel('Potencia Desarrollada en Watts')
subplot(2,2,3)
plot(wr,iin(1,:),'-')
xlabel('Velocidad del Rotor en rad/seg')
ylabel('Corriente del Estator en Amps')
subplot(2,2,4)
plot(wr,eff(1,:),'-')
xlabel('Velocidad del Rotor en rad/seg')
ylabel('Eficiencia en porciento')

% ***** Fijando todas las condiciones iniciales en la simulación de SIMULINK a cero *****
Psiqso = 0;
Psidso = 0;
Psipqro = 0;
Psipdro = 0;
wrbywbo = 0;

% ++++ CAMBIOS DE PASO EN EL PAR A UNA VELOCIDAD DE REFERENCIA FIJA ++++

tstop = 5; % Duración de la simulación en segundos

% Estructuración de la velocidad de referencia para un ciclo con carga
time_wref = [0 0.25 tstop];
speed_wref = [0 wbm wbm ];

% Estructuración de la señal de Tmech para un ciclo con carga
time_tmech = [0 0.25 0.25 1.5 1.5 2.5 2.5 3.5 3.5 5];
tmech_tmech= [0 0 -Trated -Trated -0.75*Trated -0.75*Trated -0.5*Trated -0.5*Trated
-0.25*Trated -0.25*Trated];

disp('Ejecute la Simulación y después teclee return para graficar.')

% Transferir al teclado para la simulación
keyboard

```

```

disp('Guardar las graficas antes de teclear return para seguir adelante con el siguiente estudio.')
keyboard

% ++++++ CAMBIO CÍCLICO DE LA VELOCIDAD DE REFERENCIA ++++++

% Estructuración de la velocidad de referencia variable para un ciclo con carga
time_wref = [0 0.25 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 5.0];
speed_wref = [0 wbm wbm 0.75*wbm 0.75*wbm 0.5*wbm 0.5*wbm -0.5*wbm
-0.5*wbm wbm wbm];

% Estructuración de la señal de Tmech para un ciclo con carga
time_tmech = [0 0.25 0.25 tstop ];
tmech_tmech = [0 0 -Trated -Trated];

disp('Ejecute la Simulación y después teclee return para graficar.')

% Transferir al teclado para la simulación
keyboard
disp('Guardar las graficas antes de teclear return para salir')
keyboard

```

A.3 Programa *m3.m*: Entrenamiento de la red neuronal para el estimador de la resistencia del rotor.

```

% **** ESTRUCTURACIÓN DE LAS SEÑALES DE ENTRADA A LA RED NEURONAL ****

% Los vectores se generaron con el estimador difuso, se omiten el total de los puntos
lam_dr;
lam_dr1 = lam_dr';           % Flujo del rotor en el eje d
lam_dr2 = delaysig(lam_dr1,1,1); % Muestreo anterior del flujo del rotor en el eje d
ids;
ids1 = ids';                % Corriente d en coordenadas síncronas
ids2 = delaysig(ids1,1,1);  % Muestreo anterior de la corriente d en coordenadas
                             % síncronas

iqs;
iqs1 = iqs';                % Corriente q en coordenadas síncronas
iqs2 = delaysig(iqs1,1,1);  % Muestreo anterior de la corriente q en coordenadas
                             % síncronas

vd;
vd1 = vd';                  % Voltaje d en coordenadas síncronas
vd2 = delaysig(vd1,1,1);    % Muestreo anterior del voltaje d en coordenadas
                             % síncronas

vq;
vq1 = vq';                  % Voltaje q en coordenadas síncronas
vq2 = delaysig(vq1,1,1);    % Muestreo anterior del voltaje q en coordenadas
                             % síncronas

```

```

we;
we1 = we'; % Velocidad del estator
we2 = delaysig(we1,1,1); % Muestreo anterior de la velocidad del estator

% Vector de las entradas para el entrenamiento de la red
x = [lam_dr1;lam_dr2;ids1;ids2;iqs1;iqs2;vd1;vd2;vq1;vq2;we1;we2];

% Vector de salida
r_dif;
r_dif1 = r_dif'; % Resistencia del rotor

% Generación de la red de Retropropagación
net = newff([minmax(x)],[5 1],{'tansig' 'purelin'});

net.trainParam.show = 50;
net.trainParam.lr = 0.01;
net.trainParam.epochs = 4000;
net.trainParam.goal = 0;
net.trainParam.max_fail = 5;
net.trainParam.mem_reduc = 1;
net.trainParam.min_grad = 1e-12;
net.trainParam.time = inf;

% Entrenamiento y simulación de la red de Retropropagación
net = train(net, x, r_dif1)
y = sim(net, x)
plot(x, r_dif1, x, y, 'o')
gensim(net, -1) % Generación de la red neuronal en Simulink

```

A.4 Programa *m4.m*: Simulación en estado estable de los estimadores de la resistencia del rotor

% Se utiliza el archivo *m2.m*, solamente se cambian las condiciones de operación

```

% ++++++ PRIMER CASO ++++++
% *****PAR DE CARGA Y VELOCIDAD MECÁNICA DEL ROTOR NOMINALES*****
% ***** SIN VARIACIÓN EN EL VALOR DE LA RESISTENCIA DEL ROTOR *****

```

```

tstop = 10; % Duración de la simulación en segundos

```

```

% Estructuración de la velocidad de referencia para un ciclo con carga
time_wref = [0 0.20 tstop];
speed_wref = [0 wbm wbm ];

```

```

% Estructuración de Tmech para un ciclo con carga
time_tmech = [0 0.20 tstop];
tmech_tmech= [0 0 -Trated];

```

```

% Estructuración de la resistencia del rotor
Rrot_Rrot = [rpr rpr rpr];

disp('Ejecute la Simulación y después teclee return para graficar.')
% Transferir al teclado para la simulación
keyboard
disp('Guardar las graficas antes de teclear return para seguir adelante con el siguiente estudio.')
keyboard

% ++++++ SEGUNDO CASO ++++++
% ***** PAR DE CARGA Y VELOCIDAD MECÁNICA DEL ROTOR NOMINALES *****
% ***** VARIACIÓN DESDE EL 100% AL 150% EN EL VALOR DE LA *****
% ***** RESISTENCIA DEL ROTOR *****

% Estructuración de la velocidad de referencia para un ciclo con carga
time_wref = [0 0.20 tstop];
speed_wref = [0 wbm wbm ];

% Estructuración de Tmech para un ciclo con carga
time_tmech = [0 0.20 tstop];
tmech_tmech= [0 0 -Trated];

% Estructuración de la resistencia del rotor
Time_Rrot = [0 3.0 3.0 5.0 5.0 7.0 7.0 tstop];
Rrot_Rrot = [rpr rpr rpr 1.2*rpr 1.2*rpr rpr 1.5*rpr 1.5*rpr];

disp('Ejecute la Simulación y después teclee return para graficar.')

% Transferir al teclado para la simulación
keyboard
disp('Guardar las graficas antes de teclear return para salir')
keyboard

```

A.5 Programa *m5.m*: Simulación en estado transitorio de los estimadores de la resistencia del rotor

```

% Se utiliza el archivo m2.m, solamente se cambian las condiciones de operación

% ++++++ PRIMER CASO ++++++
% *** PAR DE CARGA NOMINAL Y VELOCIDAD MECÁNICA DEL ROTOR VARIABLE **
% ***** SIN VARIACIÓN EN EL VALOR DE LA RESISTENCIA DEL ROTOR *****

tstop = 10; % Duración de la simulación en segundos

% Estructuración de la velocidad de referencia para un ciclo con carga

```

```

time_wref = [0 0.20 3.0 3.5 5.0 5.5 7.0 7.5 tstop];
speed_wref = [0 wbm wbm -0.5*wbm -0.5*wbm -1.0*wbm -1.0*wbm wbm wbm];

% Estructuración de Tmech para un ciclo con carga
time_tmech = [0 0.20 tstop];
tmech_tmech= [0 0 -Trated];
% Estructuración de la resistencia del rotor
Rrot_Rrot = [rpr rpr rpr];

disp('Ejecute la Simulación y después teclee return para graficar.')
% Transferir al teclado para la simulación
keyboard
disp('Guardar las graficas antes de teclear return para seguir adelante con el siguiente estudio.')
keyboard

% ++++++ SEGUNDO CASO ++++++
% *** PAR DE CARGA VARIABLE Y VELOCIDAD MECÁNICA DEL ROTOR NOMINAL ***
% ***** SIN VARIACIÓN EN EL VALOR DE LA RESISTENCIA DEL ROTOR *****

% Estructuración de la velocidad de referencia para un ciclo con carga
time_wref = [0 0.20 tstop];
speed_wref = [0 wbm wbm];

% Estructuración de Tmech para un ciclo con carga
time_tmech = [0 0.20 0.20 3.0 3.0 5.0 5.0 6.0 6.0 8.0 8.0 9.0 9.0 9.3 9.3 tstop];
tmech_tmech = [0 0 -Trated -Trated -0.75*Trated -0.75*Trated -Trated -Trated
-0.5*Trated -0.5*Trated -Trated -Trated 0 0 -Trated -Trated];

% Estructuración de la resistencia del rotor
Rrot_Rrot = [rpr rpr rpr];

disp('Ejecute la Simulación y después teclee return para graficar.')
% Transferir al teclado para la simulación
keyboard
disp('Guardar las graficas antes de teclear return para seguir adelante con el siguiente estudio.')
keyboard

% ++++++ TERCER CASO ++++++
% ***** PAR DE CARGA Y VELOCIDAD MECÁNICA DEL ROTOR VARIABLE *****
% ***** SIN VARIACIÓN EN EL VALOR DE LA RESISTENCIA DEL ROTOR *****

% Estructuración de la velocidad de referencia para un ciclo con carga
time_wref = [0 0.20 3.0 3.5 5.0 5.5 7.0 7.5 tstop];
speed_wref = [0 wbm wbm -0.5*wbm -0.5*wbm -1.0*wbm -1.0*wbm wbm wbm];

% Estructuración de Tmech para un ciclo con carga
time_tmech = [0 0.20 0.20 3.0 3.0 5.0 5.0 6.0 6.0 8.0 8.0 9.0 9.0 9.3 9.3 tstop];

```

```

tmech_tmech = [0 0 -Trated -Trated -0.75*Trated -0.75*Trated -Trated -Trated
-0.5*Trated -0.5*Trated -Trated -Trated 0 0 -Trated -Trated];

% Estructuración de la resistencia del rotor
Rrot_Rrot = [rpr rpr rpr];

disp('Ejecute la Simulación y después teclee return para graficar.')
% Transferir al teclado para la simulación
keyboard
disp('Guardar las graficas antes de teclear return para seguir adelante con el siguiente
estudio.')
keyboard

% ++++++ CUARTO CASO ++++++
% ***** PAR DE CARGA Y VELOCIDAD MECÁNICA DEL ROTOR VARIABLE *****
% ***** VARIACIÓN DESDE EL 100% AL 150% EN EL VALOR DE LA *****
% ***** RESISTENCIA DEL ROTOR *****

% Estructuración de la velocidad de referencia para un ciclo con carga
time_wref = [0 0.20 3.0 3.5 5.0 5.5 7.0 7.5 tstop];
speed_wref = [0 wbm wbm -0.5*wbm -0.5*wbm -1.0*wbm -1.0*wbm wbm wbm];

% Estructuración de Tmech para un ciclo con carga
time_tmech = [0 0.20 0.20 3.0 3.0 5.0 5.0 6.0 6.0 8.0 8.0 9.0 9.0 9.3 9.3 tstop];
tmech_tmech = [0 0 -Trated -Trated -0.75*Trated -0.75*Trated -Trated -Trated
-0.5*Trated -0.5*Trated -Trated -Trated 0 0 -Trated -Trated];

% Estructuración de la resistencia del rotor
Time_Rrot = [0 3.0 3.0 5.0 5.0 7.0 7.0 tstop];
Rrot_Rrot = [rpr rpr rpr 1.2*rpr 1.2*rpr rpr 1.5*rpr 1.5*rpr];

disp('Ejecute la Simulación y después teclee return para graficar.')

% Transferir al teclado para la simulación
keyboard
disp('Guardar las graficas antes de teclear return para salir')
keyboard

```

A.6 Programa *m6.m*: Parámetros de la maquina de inducción trifásica de 100 HP

```

Ts=10e-6; %Tamaño del paso fijo

% ***** VALORES NOMINALES *****
Sb = 100*746; % Potencia nominal en VA
Vrated = 460; % Voltaje nominal de línea-línea en V
pf = 0.853; % Factor de potencia nominal

```

```

Irated = Sb/(sqrt(3)*Vrated*pf); % Corriente nominal rms en A
P= 4; % Numero de polos
frated = 60; % Frecuencia nominal en Hz

% ***** VALORES BASE *****
wb = 2*pi*frated; % Frecuencia eléctrica base en rad/s
we = wb;
wbm = 2*wb/P; % Frecuencia mecánica base en rad/s
Tb = Sb/wbm; % Par base
Zb = Vrated*Vrated/Sb; % Impedancia base en Ohms
Vm = Vrated*sqrt(2/3); % Magnitud del voltaje de fase en V
Vb = Vm;
Tfactor = (3*P)/(4*wb); % Factor para la expresión del par

% ***** ESPECIFICACIONES *****
srated = 0.0248; % Deslizamiento nominal
Nrated = 1748.3; % Velocidad nominal en rev/min
wmrated = 2*pi*Nrated/60; % Velocidad nominal en rad/s
Trated = Sb/wmrated; % Par nominal
iasb = 109.897; % Corriente de fase nominal rms en A

% ***** PARÁMETROS *****
rs = 0.0425; % Resistencia del devanado del estator en Ohms
Rs=rs;
xls = 0.284; % Reactancia de enlace del estator en Ohms
xplr = xls; % Reactancia de enlace del rotor en Ohms
xm = 8.51; % Reactancia de magnetización del estator en Ohms
rpr = 0.0425; % Resistencia referida al devanado del rotor en Ohms
xM = 1/(1/xm + 1/xls + 1/xplr);
J = 2.0; % Inercia del rotor en kg-m2
H = J*wbm*wbm/(2*Sb); % Constante de inercia en s
Domega = 0; % Coeficiente de amortiguamiento del rotor
Lls=xls/wb;
Lplr=Lls;
Lm=xm/wb;
LM=xM/wb;
Lf=1-((LM*LM)/(Lls*Lplr));

```

Apéndice B

OPERACIÓN CON CAMPO DEBILITADO

B.1 Introducción

En un accionamiento del motor de inducción, el par electromagnético y el par de carga deben ser iguales en magnitud y opuesto en dirección durante la operación en estado estable. En general se pueden separar las cargas en cuatro categorías, 1) cargas con velocidad constante y par variable o constante, 2) cargas con velocidad ajustable y par constante, 3) cargas con velocidad ajustable y par variable y 4) cargas con velocidad variable y par variable [33].

Debe notarse que una vez que la corriente de armadura y de campo se mantienen constantes, la restricción establecida por la curva #1 en la Figura B.1 produce un límite del par constante sobre el plano velocidad-par mientras que la restricción de la corriente de armadura constante y la fuerza contraelectromotriz constante de la curva #2 imponen un límite de la potencia constante. Despreciando la resistencia de armadura, la curva #3 impone una característica del inverso del par al cuadrado. La aceptación de estas limitaciones de la capacidad de una máquina de CD, ha causado que se definan las mismas características con respecto a los requerimientos del par, tanto para la máquina de CD como para cualquier tipo de accionamiento de motores. Si las curvas de capacidad máxima de una máquina de inducción son consistentes con los requerimientos del par, esto hace que se adopten las curvas de capacidad máxima de una máquina de CD [33].

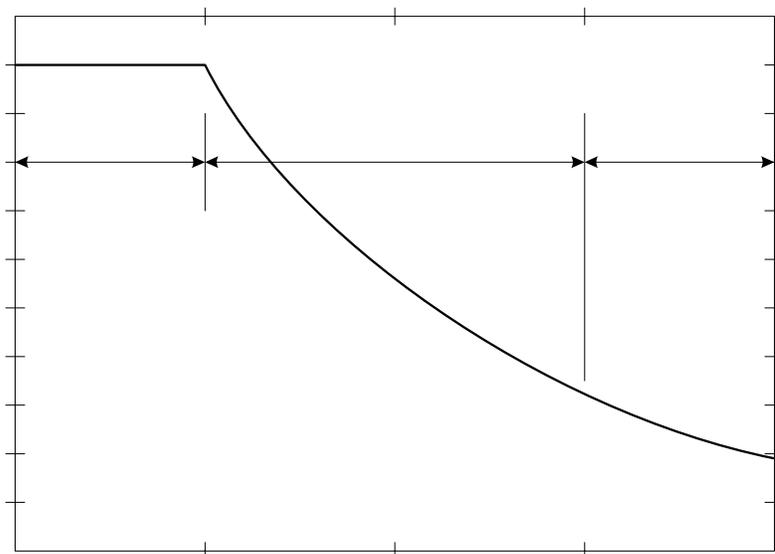


Figura B.1 Curva del par contra la velocidad para un accionamiento del motor de CD.

El rango de operación con campo debilitado se divide en dos regiones distintas. La primera región empieza a la velocidad base y finaliza a la velocidad dónde el límite de la corriente no se alcanza debido al límite del voltaje. La segunda región se extiende más allá de la primera región con el límite de voltaje anulando el límite de la corriente y forzando una reducción en la corriente del estator por debajo del límite de la corriente [58].

B.2 Estrategia de control de la máquina de inducción para la operación de potencia constante

Un requerimiento del par máximo establecido por la curva #1 de la Figura B.1 se cumple para el caso del control vectorial del motor de inducción. En particular, cuando la frecuencia de deslizamiento se mantiene constante ocasiona que la reactancia $j(L_m/L_r')X_m$ y la resistencia $(L_m/L_r')^2[r_r'/s\omega_e]\omega_e$ sean proporcionales a la frecuencia aplicada. Por lo tanto, si la amplitud de la corriente del estator, i_s , se mantiene constante por medio del regulador de corriente, entonces la división de i_s entre la componente de magnetización, i_{ds} y la componente productora del par, i_{qs} permanece sin cambios y así el par es constante, como se demuestra en la Ec. 2.16. Excepto para efectos ligeramente diferente del estator, las curvas de las variables eléctricas son notablemente similares a la máquina de CD [33].

En el rango de alta velocidad, la capacidad del par de salida de la máquina de inducción depende de la estrategia del campo debilitado. El método convencional para el campo debilitado es variar el flujo del rotor de referencia en proporción a la inversa de la velocidad del rotor como se muestra en la Figura B.2. Por consiguiente, el par disminuirá inversamente de acuerdo con la Ec. 2.16 [33].

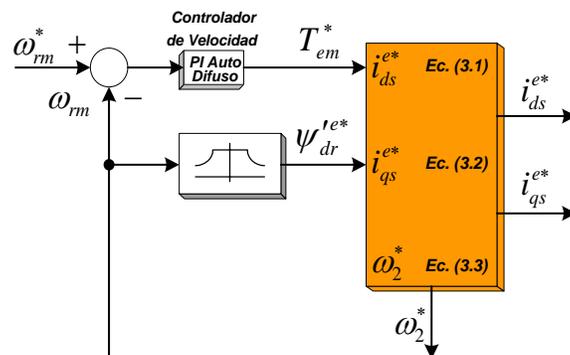


Figura B.2 Diagrama de bloques del campo debilitado usado en el debilitamiento del flujo del rotor.

B.3 Obtención del flujo del rotor $\psi_{dr}'^e$ de referencia y la velocidad mecánica ω_{rm} del rotor

El flujo del rotor $\psi_{dr}'^e$ de referencia se calcula a partir de las siguientes ecuaciones:

$$\psi_{dr}'^e = \left(\frac{\text{real}(E') \frac{X_r}{X_m}}{\omega_e} \right) \left(\frac{1}{\omega_{bm}} \right) \quad (\text{B.1})$$

En donde el voltaje presente en la reactancia transitoria del estator se denota por E' , el cual es:

$$E' = v_{as} - (r_s + jX')i_{as} \quad (\text{B.2})$$

Siendo X' la reactancia transitoria del estator:

$$X' = X_s - \frac{X_m^2}{X_r} \quad (\text{B.3})$$

Y de la Ec. B.2, i_{as} es la corriente de la fase a:

$$i_{as} = \frac{v_{as}}{r_s + j(X_s + X_m)} \quad (\text{B.4})$$

Para obtener la velocidad mecánica del rotor, ω_{rm} , se utilizan las siguientes relaciones:

$$\omega_{bm} = \omega_{rm} = \frac{2\omega_b}{P} \quad (\text{B.5})$$

donde ω_{bm} es la velocidad mecánica base y ω_b es la velocidad eléctrica base:

$$\omega_b = 2\pi f_N \quad (\text{B.6})$$

Apéndice C

PRINCIPIOS DE LAS REDES NEURONALES ARTIFICIALES

C.1 Introducción

El modelo de una neurona artificial [48] es una imitación del proceso de una neurona biológica, puede también asemejarse a un sumador hecho con un amplificador operacional tal como se ve en la Fig. C.1.

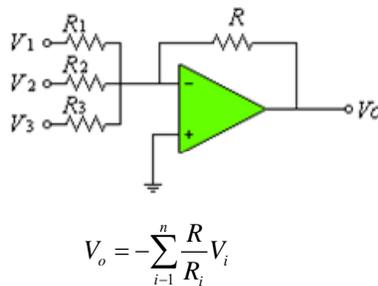


Figura C.1. Neurona Artificial.

Existen varias formas de nombrar una neurona artificial, es conocida como nodo, neuronodo, celda, unidad o elemento de procesamiento (EP); En la Fig. C.2 se observa un EP en forma general y su similitud con una neurona biológica

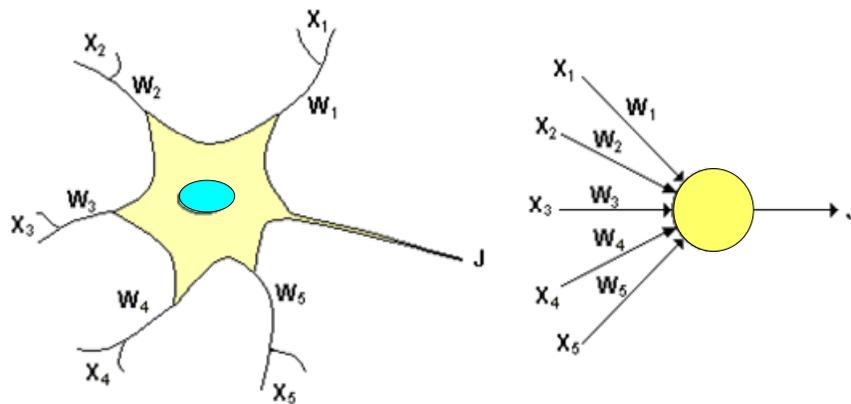


Figura C.2. De la neurona biológica a la neurona artificial.

De la observación detallada del proceso biológico se han hallado los siguientes análogos con el sistema artificial:

- Las entradas X_i representan las señales que provienen de otras neuronas y que son capturadas por las dendritas.

- Los pesos W_i son la intensidad de la sinápsis que conecta dos neuronas; tanto X_i como W_i son valores reales.
- θ es la función umbral que la neurona debe superar para activarse; este proceso ocurre biológicamente en el cuerpo de la célula.

Las señales de entrada a una neurona artificial X_1, X_2, \dots, X_n son variables continuas en lugar de pulsos discretos, como se presentan en una neurona biológica. Cada señal de entrada pasa a través de una ganancia o peso, llamado peso sináptico o fortaleza de la conexión cuya función es análoga a la de la función sináptica de la neurona biológica. Los pesos pueden ser positivos (excitatorios), o negativos (inhibitorios), el nodo sumatoria acumula todas las señales de entradas multiplicadas por los pesos o ponderadas y las pasa a la salida a través de una función umbral o función de transferencia. La entrada neta a cada unidad puede escribirse de la siguiente manera

$$neta_i = \sum_{i=1}^n W_i X_i = \vec{X} \vec{W} \quad (C.1)$$

Una idea clara de este proceso se muestra en la Fig. C.3, en donde puede observarse el recorrido de un conjunto de señales que entran a la red.

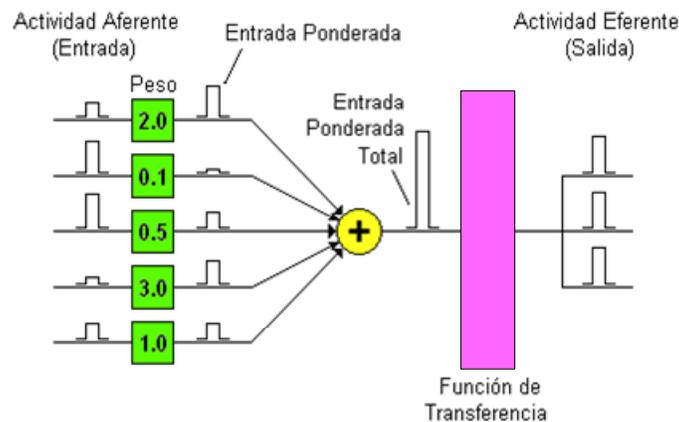


Figura C.3. Proceso de una red neuronal.

Una vez que se ha calculado la activación del nodo, el valor de salida equivale a

$$x_i = f_i(neta_i) \quad (C.2)$$

donde f_i representa la función de activación para esa unidad, que corresponde a la función escogida para transformar la entrada $neta_i$ en el valor de salida x_i y que depende de las características específicas de cada red.

C.2 Red neuronal artificial multicapa

La regla de aprendizaje del Perceptrón de Rosenblatt y el algoritmo LMS (Least Mean Square) de Widrow y Hoff fueron diseñados para entrenar redes de una sola capa. En general, estas redes tienen la desventaja que solo pueden resolver problemas linealmente separables, fue esto lo que llevó al surgimiento de las redes multicapa para superar esta dificultad en las redes hasta entonces conocidas.

El primer algoritmo de entrenamiento para redes multicapa fue desarrollado por Paul Werbos en 1974, este se desarrolló en un contexto general, para cualquier tipo de redes, siendo las redes neuronales una aplicación especial, razón por la cual el algoritmo no fue aceptado dentro de la comunidad de desarrolladores de redes neuronales. Fue solo hasta mediados de los años 80 cuando el algoritmo de Retropropagación o algoritmo de propagación inversa fue redescubierto al mismo tiempo por varios investigadores, David Rumelhart, Geoffrey Hinton y Ronald Williams, David Parker y Yann Le Cun. El algoritmo se popularizó cuando fue incluido en el libro “Parallel Distributed Processing Group” por los psicólogos David Rumelhart y James McClelland. La publicación de este libro trajo consigo un auge en las investigaciones con redes neuronales, siendo la de Retropropagación una de las redes más ampliamente empleadas, aun en nuestros días.

Uno de los grandes avances logrados con la Retropropagación es que esta red aprovecha la naturaleza paralela de las redes neuronales para reducir el tiempo requerido por un procesador secuencial para determinar la correspondencia entre unos patrones dados. Además el tiempo de desarrollo de cualquier sistema que se este tratando de analizar se puede reducir como consecuencia de que la red puede aprender el algoritmo correcto sin que alguien tenga que deducir por anticipado el algoritmo en cuestión.

La mayoría de los sistemas actuales de cómputo se han diseñado para llevar a cabo funciones matemáticas y lógicas a una velocidad que resulta asombrosamente alta para el ser humano. Sin embargo la destreza matemática no es lo que se necesita para solucionar problemas de reconocimiento de patrones en entornos ruidosos, característica que incluso dentro de un espacio de entrada relativamente pequeño, puede llegar a consumir mucho tiempo. El problema es la naturaleza secuencial del propio computador; el ciclo *tomar-ejecutar* de la naturaleza Von Neumann solo permite que la máquina realice una operación a la vez. En la mayoría de los casos, el tiempo que necesita la máquina para llevar a cabo cada instrucción es tan breve (típicamente una millonésima de segundo) que el tiempo necesario para un programa, así sea muy grande, es insignificante para los usuarios. Sin embargo, para aquellas aplicaciones que deban explorar un gran espacio de entrada o que intentan correlacionar todas las permutaciones posibles de un conjunto de patrones muy complejo, el tiempo de computación necesario se hace bastante grande.

Lo que se necesita es un nuevo sistema de procesamiento que sea capaz de examinar todos los patrones en paralelo. Idealmente ese sistema no tendría que ser programado explícitamente, lo que haría es adaptarse a sí mismo para aprender la relación entre un

conjunto de patrones dado como ejemplo y ser capaz de aplicar la misma relación a nuevos patrones de entrada. Este sistema debe estar en capacidad de concentrarse en las características de una entrada arbitraria que se asemeje a otros patrones vistos previamente, sin que ninguna señal de ruido lo afecte. Este sistema fue el gran aporte de la red de Retropropagación.

La Retropropagación es un tipo de red de aprendizaje supervisado, que emplea un ciclo *propagación–adaptación* de dos fases. Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas superiores de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas.

Las salidas de error se propagan hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida. Sin embargo las neuronas de la capa oculta solo reciben una fracción de la señal total del error, basándose aproximadamente en la contribución relativa que haya aportado cada neurona a la salida original. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total. Basándose en la señal de error percibida, se actualizan los pesos de conexión de cada neurona, para hacer que la red converja hacia un estado que permita clasificar correctamente todos los patrones de entrenamiento.

La importancia de este proceso consiste en que, a medida que se entrena la red, las neuronas de las capas intermedias se organizan a sí mismas de tal modo que las distintas neuronas aprenden a reconocer distintas características del espacio total de entrada. Después del entrenamiento, cuando se les presente un patrón arbitrario de entrada que contenga ruido o que esté incompleto, las neuronas de la capa oculta de la red responderán con una salida activa si la nueva entrada contiene un patrón que se asemeje a aquella característica que las neuronas individuales hayan aprendido a reconocer durante su entrenamiento. Y a la inversa, las unidades de las capas ocultas tienen una tendencia a inhibir su salida si el patrón de entrada no contiene la característica para reconocer, para la cual han sido entrenadas.

Varias investigaciones han demostrado que, durante el proceso de entrenamiento, la red de Retropropagación tiende a desarrollar relaciones internas entre neuronas con el fin de organizar los datos de entrenamiento en clases. Esta tendencia se puede extrapolar, para llegar a la hipótesis consistente en que todas las unidades de la capa oculta de una Retropropagación son asociadas de alguna manera a características específicas del patrón de entrada como consecuencia del entrenamiento. Lo que sea o no exactamente la asociación puede no resultar evidente para el observador humano, lo importante es que la red ha encontrado una representación interna que le permite generar las salidas deseadas cuando se le dan las entradas, en el proceso de entrenamiento. Esta misma representación interna se puede aplicar a entradas que la red no haya visto antes, y la red clasificará estas entradas según las características que compartan con los ejemplos de entrenamiento.

C.3 Estructura de la red neuronal artificial multicapa

La estructura típica de una red multicapa se observa en la Fig. C.4.

Puede notarse que esta red de tres capas equivale a tener tres redes tipo Perceptron en cascada; la salida de la primera red, es la entrada a la segunda y la salida de la segunda red es la entrada a la tercera. Cada capa puede tener diferente número de neuronas, e incluso distinta función de transferencia.

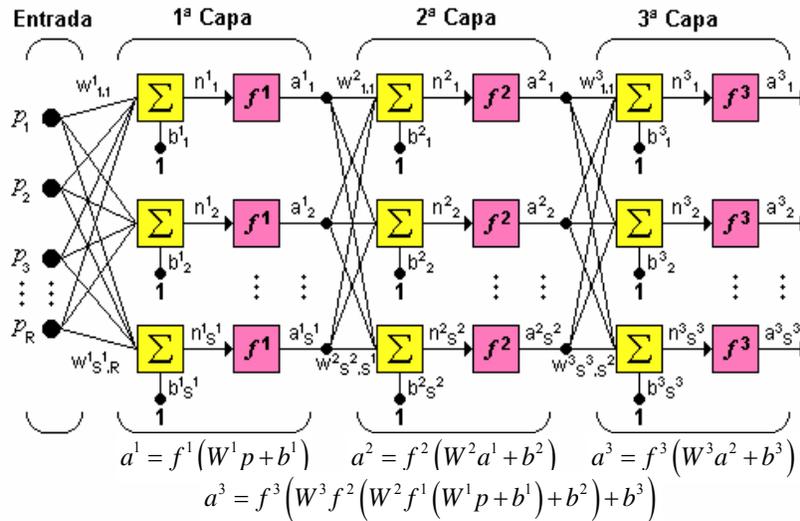


Figura C.4. Red de tres capas.

En la Fig. C.4, W_1 representa la matriz de pesos para la primera capa, W_2 los pesos de la segunda y así similarmente para todas las capas que incluya una red. Para identificar la estructura de una red multicapa, se empleará una notación abreviada, donde el número de entradas va seguido del número de neuronas en cada capa:

$$R : S^1 : S^2 : S^3 \quad (C.3)$$

donde S representa el número de neuronas y el exponente representa la capa a la cual la neurona corresponde.

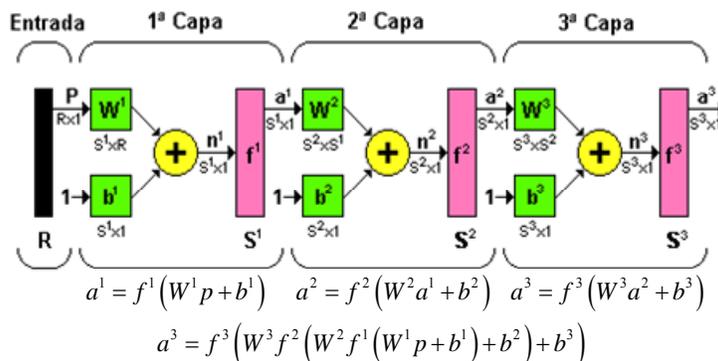


Figura C.5. Notación compacta de una red de tres capas.

La notación de la Fig. C.4 es bastante clara cuando se desea conocer la estructura detallada de la red, e identificar cada una de las conexiones, pero cuando la red es muy grande, el proceso de conexión se torna muy complejo y es bastante útil utilizar el esquema de la Fig. C.5.

C.4 Regla de aprendizaje de la red neuronal artificial multicapa

El algoritmo de Retropropagación para redes multicapa es una generalización del algoritmo LMS, ambos algoritmos realizan su labor de actualización de pesos y ganancias con base en el error medio cuadrático. La red de Retropropagación trabaja bajo aprendizaje supervisado y por tanto necesita un conjunto de entrenamiento que le describa cada salida y su valor de salida esperado de la siguiente forma:

$$\{p_1, t_1\}, \{p_2, t_2\}, \dots, \{p_Q, t_Q\} \quad (C.4)$$

donde p_Q es una entrada a la red y t_Q es la correspondiente salida deseada para el patrón q -ésimo. El algoritmo debe ajustar los parámetros de la red para minimizar el error medio cuadrático.

El entrenamiento de una red neuronal multicapa se realiza mediante un proceso de aprendizaje, para realizar este proceso se debe inicialmente tener definida la topología de la red esto es: número de neuronas en la capa de entrada el cual depende del número de componentes del vector de entrada, cantidad de capas ocultas y número de neuronas de cada una de ellas, número de neuronas en la capa de la salida el cual depende del número de componentes del vector de salida o patrones objetivo y funciones de transferencia requeridas en cada capa, con base en la topología escogida se asignan valores iniciales a cada uno de los parámetros que conforma la red.

Es importante recalcar que no existe una técnica para determinar el número de capas ocultas, ni el número de neuronas que debe contener cada una de ellas para un problema específico, esta elección es determinada por la experiencia del diseñador, el cual debe cumplir con las limitaciones de tipo computacional.

Cada patrón de entrenamiento se propaga a través de la red y sus parámetros para producir una respuesta en la capa de salida, la cual se compara con los patrones objetivo o salidas deseadas para calcular el error en el aprendizaje, este error marca el camino mas adecuado para la actualización de los pesos y ganancias que al final del entrenamiento producirán una respuesta satisfactoria a todos los patrones de entrenamiento, esto se logra minimizando el error medio cuadrático en cada iteración del proceso de aprendizaje.

La deducción matemática de este procedimiento se realizará para una red con una capa de entrada, una capa oculta y una capa de salida y luego se generalizará para redes que tengan más de una capa oculta.

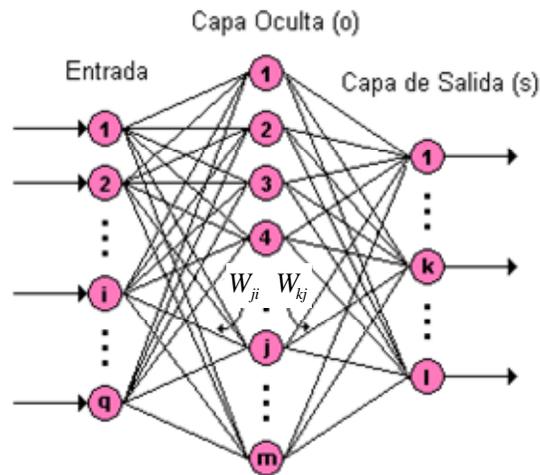


Figura C.6. Disposición de una red sencilla de 3 capas.

Es importante aclarar que en la Fig. C.6:

- q : Equivale al número de componentes del vector de entrada.
- m : Número de neuronas de la capa oculta.
- l : Número de neuronas de la capa de salida.

Para iniciar el entrenamiento se le presenta a la red un patrón de entrenamiento, el cual tiene q componentes como se describe en la Ec. (C.5),

$$P = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_i \\ \vdots \\ p_q \end{bmatrix} \quad (C.5)$$

Cuando se le presenta a la red un patrón de entrenamiento, este se propaga a través de las conexiones existentes produciendo una entrada neta n en cada una de las neuronas de la siguiente capa, la entrada neta a la neurona j de la siguiente capa debido a la presencia de un patrón de entrenamiento en la entrada está dada por la Ec. (C.6), nótese que la entrada neta es el valor justo antes de pasar por la función de transferencia:

$$n_j^o = \sum_{i=1}^q W_{ji}^o p_i + b_j^o \quad (C.6)$$

donde:

- W_{ji}^o : Peso que une la componente i de la entrada con la neurona j de la capa oculta.
- p_i : Componente i del vector p que contiene el patrón de entrenamiento de q componentes.
- b_j^o : Ganancia de la neurona j de la capa oculta.

donde el superíndice (o) representa la capa a la que pertenece cada parámetro, es este caso la capa oculta.

Cada una de las neuronas de la capa oculta tiene como salida a_j^o , que esta dada por la Ec. (C.7)

$$a_j^o = f^o \left(\sum_{i=1}^q W_{ji}^o p_i + b_j^o \right) \quad (C.7)$$

donde:

f^o : Función de transferencia de las neuronas de la capa oculta.

Las salidas a_j^o de las neuronas de la capa oculta (de l componentes) son las entradas a los pesos de conexión de la capa de salida, este comportamiento esta descrito por la siguiente expresión:

$$n_k^s = \sum_{j=1}^m W_{kj}^s a_j^o + b_k^s \quad (C.8)$$

donde:

- W_{kj}^s : Peso que une la neurona j de la capa oculta con la neurona k de la capa de salida, la cual cuenta con s neuronas.
- a_j^o : Salida de la neurona j de la capa oculta, la cual cuenta con m neuronas.
- b_k^s : Ganancia de la neurona k de la capa de salida.
- n_k^s : Entrada neta a la neurona k de la capa de salida

La red produce una salida final descrita por la siguiente expresión:

$$a_k^s = f^s \left(n_k^s \right) \quad (C.9)$$

donde:

f^s : Función de transferencia de las neuronas de la capa de salida.

Reemplazando (C.8) en (C.9) se obtiene la salida de la red en función de la entrada neta y de los pesos de conexión con la última capa oculta:

$$a_k^s = f^s \left(\sum_{j=1}^m W_{kj}^s a_j^o + b_k^s \right) \quad (C.10)$$

La salida de la red de cada neurona a_k^s se compara con la salida deseada t_k para calcular el error en cada unidad de salida,

$$\delta_k = (t_k - a_k^s) \quad (C.11)$$

El error debido a cada patrón p propagado esta dado por:

$$ep^2 = \frac{1}{2} \sum_{k=1}^s (\delta_k)^2 \quad (C.12)$$

donde:

ep^2 : Error medio cuadrático para cada patrón de entrada p .

δ_k : Error en la neurona k de la capa de salida con l neuronas.

Este proceso se repite para el número total de patrones de entrenamiento (r), para un proceso de aprendizaje exitoso el objetivo del algoritmo es actualizar todos los pesos y ganancias de la red minimizando el error medio cuadrático total descrito en:

$$e^2 = \sum_{p=1}^r ep^2 \quad (C.13)$$

donde:

e^2 : Error total en el proceso de aprendizaje en una iteración luego de haber presentado a la red los r patrones de entrenamiento.

El error que genera una red neuronal en función de sus pesos, genera un espacio de n dimensiones, donde n es el número de pesos de conexión de la red, al evaluar el gradiente del error en un punto de esta superficie se obtendrá la dirección en la cual la función del error tendrá un mayor crecimiento, como el objetivo del proceso de aprendizaje es minimizar el error debe tomarse la dirección negativa del gradiente para obtener el mayor decremento del error y de esta forma su minimización, condición requerida para realizar la actualización de la matriz de pesos en el algoritmo de Retropropagación:

$$W_{k+1} = W_k - \alpha \nabla ep^2 \quad (\text{C.14})$$

El gradiente negativo de ep^2 se denotara como $-\nabla ep^2$ y se calcula como la derivada del error respecto a todos los pesos de la red.

En la capa de salida el gradiente negativo del error con respecto a los pesos es:

$$-\frac{\partial ep^2}{\partial W_{kj}^s} = -\frac{\partial}{\partial W_{kj}^s} \left(\frac{1}{2} \sum_{k=1}^l (t_k - a_k^s)^2 \right) = (t_k - a_k^s) \times \frac{\partial a_k^s}{\partial W_{kj}^s} \quad (\text{C.15})$$

donde:

$\frac{\partial ep^2}{\partial W_{kj}^s}$: Componente del gradiente $-\nabla ep^2$ respecto al peso de la conexión de la neurona de la capa de salida y la neurona j de la capa oculta W_{kj}^s .

$\frac{\partial a_k^s}{\partial W_{kj}^s}$: Derivada de la salida de la neurona k de la capa de salida respecto, al peso W_{kj}^s .

Para calcular $\partial a_k^s / \partial W_{kj}^s$ se debe utilizar la regla de la cadena, pues el error no es una función explícita de los pesos de la red, de la Ec. (C.9) puede verse que la salida de la red a_k^s esta explícitamente en función de n_k^s y de la Ec. (C.8) puede verse que n_k^s esta explícitamente en función de W_{kj}^s , considerando esto se genera la Ec. (C.16):

$$\frac{\partial a_k^s}{\partial W_{kj}^s} = \frac{\partial a_k^s}{\partial n_k^s} \times \frac{\partial n_k^s}{\partial W_{kj}^s} \quad (\text{C.16})$$

Tomando la Ec. (C.16) y reemplazándola en la Ec. (C.15) se obtiene:

$$-\frac{\partial ep^2}{\partial W_{kj}^s} = (t_k - a_k^s) \times \frac{\partial a_k^s}{\partial n_k^s} \times \frac{\partial n_k^s}{\partial W_{kj}^s} \quad (\text{C.17})$$

donde:

$\frac{\partial n_k^s}{\partial W_{kj}^s}$: Derivada de la entrada neta a la neurona k de la capa de salida respecto a los pesos de la conexión entre las neuronas de la capa oculta y la capa de salida.

$\frac{\partial a_k^s}{\partial n_k^s}$: Derivada de la salida de la neurona k de la capa de salida respecto a su entrada neta.

Reemplazando en la ecuación (C.17) las derivadas de las Ecs. (C.8) y (C.9) se obtiene:

$$-\frac{\partial ep^2}{\partial W_{kj}^s} = (t_k - a_k^s) \times f'^s(n_k^s) \times a_j^o \quad (C.18)$$

Como se observa en la Ec. (C.18) las funciones de transferencia utilizadas en este tipo de red deben ser continuas para que su derivada exista en todo el intervalo, ya que el término $f'^s(n_k^s)$ es requerido para el cálculo del error.

Las funciones de transferencia f más utilizadas y sus respectivas derivadas son las siguientes:

$$\text{logsig} \quad f(n) = \frac{1}{1 + e^{-n}} \quad f'(n) = f(n)[1 - f(n)] \quad f'(n) = a(1 - a) \quad (C.19)$$

$$\text{tansig} \quad f(n) = \frac{e^n - e^{-n}}{e^n + e^{-n}} \quad f'(n) = 1 - [f(n)]^2 \quad f'(n) = (1 - a^2) \quad (C.20)$$

$$\text{purelin} \quad f(n) = n \quad f'(n) = 1 \quad (C.21)$$

De la Ec. (C.18), los términos del error para las neuronas de la capa de salida están dados por la Ec. (C.22), la cual se le denomina comúnmente sensibilidad de la capa de salida.

$$\delta_k^s = (t_k - a_k^s) f'^s(n_k^s) \quad (C.22)$$

Este algoritmo se denomina Retropropagación o de propagación inversa debido a que el error se propaga de manera inversa al funcionamiento normal de la red, de esta forma, el algoritmo encuentra el error en el proceso de aprendizaje desde las capas más internas hasta llegar a la entrada; con base en el cálculo de este error se actualizan los pesos y ganancias de cada capa.

Después de conocer (C.22) se procede a encontrar el error en la capa oculta el cual esta dado por:

$$-\frac{\partial ep^2}{\partial W_{ji}^o} = -\frac{\partial}{\partial W_{ji}^o} \left(\frac{1}{2} \sum_{k=1}^l (t_k - a_k^s)^2 \right) = \sum_{k=1}^l (t_k - a_k^s) \times \frac{\partial a_k^s}{\partial W_{ji}^o} \quad (C.23)$$

Para calcular el último término de la Ec. (C.23) se debe aplicar la regla de la cadena en varias ocasiones como se observa en la Ec. (C.24) puesto que la salida de la red no es una función explícita de los pesos de la conexión entre la capa de entrada y la capa oculta.

$$\frac{\partial a_k^s}{\partial W_{ji}^o} = \frac{\partial a_k^s}{\partial n_k^s} \times \frac{\partial n_k^s}{\partial a_k^o} \times \frac{\partial a_k^o}{\partial n_j^o} \times \frac{\partial n_j^o}{\partial W_{ji}^o} \quad (C.24)$$

Todos los términos de la Ec. (C.25) son derivados respecto a variables de las que dependen explícitamente, reemplazando (C.24) en (C.23) tenemos:

$$-\frac{\partial ep^2}{\partial W_{ji}^o} = \sum_{k=1}^l (t_k - a_k^s) \times \frac{\partial a_k^s}{\partial n_k^s} \times \frac{\partial n_k^s}{\partial a_k^o} \times \frac{\partial a_k^o}{\partial n_j^o} \times \frac{\partial n_j^o}{\partial W_{ji}^o} \quad (C.25)$$

Tomando las derivadas de las Ecs. (C.6), (C.7), (C.8), (C.9) y reemplazándolas en la Ec. (C.25) se obtiene la expresión del gradiente del error en la capa oculta

$$-\frac{\partial ep^2}{\partial W_{ji}^o} = \sum_{k=1}^l (t_k - a_k^s) \times f'^s(n_k^s) \times \mathbf{W}_{kj}^s \times f'^o(n_j^o) \times \mathbf{p}_i \quad (C.26)$$

Reemplazando la Ec. (C.22) en la Ec. (C.26) se tiene:

$$-\frac{\partial ep^2}{\partial W_{ji}^o} = \sum_{k=1}^l \delta_k^s \times \mathbf{W}_{kj}^s \times f'^o(n_j^o) \times \mathbf{p}_i \quad (C.27)$$

Los términos del error para cada neurona de la capa oculta esta dado por la Ec. (C.28), este término también se denomina sensibilidad de la capa oculta.

$$\delta_j^o = f'^o(n_j^o) \times \sum_{k=1}^l \delta_k^s \mathbf{W}_{kj}^s \quad (C.28)$$

Luego de encontrar el valor del gradiente del error se procede a actualizar los pesos de todas las capas empezando por la de salida, para la capa de salida la actualización de pesos y ganancias esta dada por (C.29) y (C.30):

$$\mathbf{W}_{kj}(t+1) = \mathbf{W}_{kj}(t) - 2\alpha \delta_k^s \quad (C.29)$$

$$\mathbf{b}_k(t+1) = \mathbf{b}_k(t) - 2\alpha \delta_k^s \quad (C.30)$$

donde:

α : Índice de aprendizaje que varía entre 0 y 1 dependiendo de las características del problema a solucionar.

Luego de actualizar los pesos y ganancias de la capa de salida se procede a actualizar los pesos y ganancias de la capa oculta mediante las Ecs. (C.31) y (C.32):

$$\mathbf{W}_{ji}(t+1) = \mathbf{W}_{ji}(t) - 2\alpha \delta_j^o p_i \quad (C.31)$$

$$\mathbf{b}_j(t+1) = \mathbf{b}_j(t) - 2\alpha\delta_j^o \quad (\text{C.32})$$

Esta deducción fue realizada para una red de tres capas, si se requiere realizar el análisis para una red con dos o más capas ocultas, las expresiones pueden derivarse de la Ec. (C.28) donde los términos que se encuentran dentro de la sumatoria pertenecen a la capa inmediatamente superior, este algoritmo es conocido como la regla Delta Generalizada desarrollada por Rumelhart D. [32], la cual es una extensión de la regla delta desarrollada por Widrow [34] en 1930

Para algunos autores las sensibilidades de las capas están denotadas por la letra S , rescribiendo las Ecs. (C.22) y (C.28) con esta notación se obtienen las Ecs. (C.33) y (C.34):

$$S^M = -2f^M(n^M)[t-a] \quad (\text{C.33})$$

$$s^m \equiv f^m(n^m)(W^{m+1})^T s^{m+1}, \text{ para } m = M-1, \dots, 2, 1 \quad (\text{C.34})$$

En la Ec. (C.33) M representa la última capa y S^M la sensibilidad para esta capa, la Ec. (C.34) expresa el cálculo de la sensibilidad capa por capa comenzando desde la última capa oculta, cada uno de estos términos involucra que el término para la sensibilidad de la capa siguiente ya este calculado.

Como se ve el algoritmo de Retropropagación utiliza la misma técnica de aproximación en pasos descendientes que emplea el algoritmo LMS, la única complicación está en el cálculo del gradiente, el cual es un término indispensable para realizar la propagación de la sensibilidad.

En las técnicas de gradiente descendiente es conveniente avanzar por la superficie de error con incrementos pequeños de los pesos; esto se debe a que tenemos una información local de la superficie y no se sabe lo lejos o lo cerca que se está del punto mínimo, con incrementos grandes, se corre el riesgo de pasar por encima del punto mínimo, con incrementos pequeños, aunque se tarde más en llegar, se evita que esto ocurra. El elegir un incremento adecuado influye en la velocidad de convergencia del algoritmo, esta velocidad se controla a través del Índice de aprendizaje α , la que por lo general se escoge como un número pequeño, para asegurar que la red encuentre una solución. Un valor pequeño de α significa que la red tendrá que hacer un gran número de iteraciones, si se toma un valor muy grande, los cambios en los pesos serán muy grandes, avanzando muy rápidamente por la superficie de error, con el riesgo de saltar el valor mínimo del error y estar oscilando alrededor de él, pero sin poder alcanzarlo.

Es recomendable aumentar el valor de α a medida que disminuye el error de la red durante la fase de entrenamiento, para garantizar así una rápida convergencia, teniendo la precaución de no tomar valores demasiado grandes que hagan que la red oscile alejándose demasiado del valor mínimo. Algo importante que debe tenerse en cuenta, es la posibilidad de convergencia hacia alguno de los mínimos locales que pueden existir en la superficie del error del espacio de pesos como se ve en la Fig. C.7.

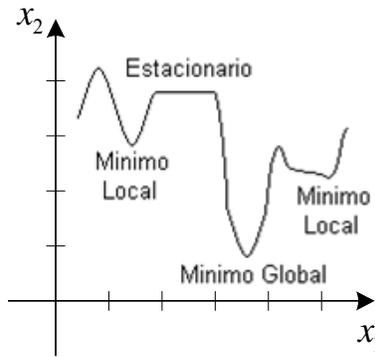


Figura C.7. Superficie típica de error

En el desarrollo matemático que se ha realizado para llegar al algoritmo de Retropropagación, no se asegura en ningún momento que el mínimo que se encuentre sea global, una vez la red se asiente en un mínimo sea local o global cesa el aprendizaje, aunque el error siga siendo alto. En todo caso, si la solución es admisible desde el punto de vista del error, no importa si el mínimo es local o global o si se ha detenido en algún momento previo a alcanzar un verdadero mínimo.

C.5 Algoritmo de Levenberg - Marquardt

Este algoritmo es una modificación del método de Newton, el que fue diseñado para minimizar funciones que sean la suma de los cuadrados de otras funciones no lineales; es por ello que el algoritmo de Levenberg-Marquardt [50], tiene un excelente desempeño en el entrenamiento de redes neuronales donde el rendimiento de la red esté determinado por el error medio cuadrático.

El método de Newton para optimizar el rendimiento $e(x)$ es:

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \mathbf{A}_k^{-1} \mathbf{g}_k \quad (\text{C.35})$$

$$\mathbf{A}_k \equiv \nabla^2 e(x) \Big|_{x=x_k} \quad \mathbf{g}_k \equiv \nabla e(x) \Big|_{x=x_k} \quad (\text{C.36})$$

Si asumimos que $e(x)$ es una suma de funciones cuadráticas:

$$e(x) = \sum_{i=1}^n v_i^2(x) = \mathbf{v}^T(x) \mathbf{v}(x) \quad (\text{C.37})$$

El gradiente puede ser escrito entonces en forma matricial:

$$\nabla e(x) = 2\mathbf{J}^T(x) \mathbf{v}(x) \quad (\text{C.38})$$

donde $\mathbf{J}(x)$ es la matriz Jacobiana.

Ajustando el método de Newton, obtenemos el algoritmo de Levenberg–Marquardt expresado de las dos formas siguientes:

$$x_{k+1} = x_k - [\mathbf{J}^T(x_k)\mathbf{J}(x_k) + \mu_k\mathbf{I}]^{-1} \mathbf{J}^T(x_k) \mathbf{v}(x_k) \quad (\text{C.39})$$

$$\Delta x_k = -[\mathbf{J}^T(x_k)\mathbf{J}(x_k) + \mu_k\mathbf{I}]^{-1} \mathbf{J}^T(x_k) \mathbf{v}(x_k) \quad (\text{C.40})$$

La nueva constante μ determina la tendencia del algoritmo, cuando μ_k se incrementa, este algoritmo se aproxima al algoritmo de pasos descendientes para índices de aprendizaje muy pequeños; cuando μ_k se minimiza este algoritmo se convierte en el método de Gauss–Newton.

El algoritmo comienza con un valor pequeño para μ_k , por lo general 0.01, si en ese paso no se alcanza el valor para $e(x)$ entonces el paso es repetido con μ_k multiplicado por un factor $\mathcal{G} > 1$. Si se ha escogido un valor pequeño de paso en la dirección de paso descendiente, $e(x)$ debería decrecer. Si un paso produce un pequeño valor para $e(x)$, entonces el algoritmo tiende al método de Gauss–Newton, el que se supone garantiza una rápida convergencia. Este algoritmo genera un compromiso entre la velocidad del método de Gauss–Newton y la garantía de convergencia del método de paso descendiente.

Los elementos de la matriz Jacobiana necesarios en el algoritmo de Levenberg–Marquardt son de este estilo:

$$[\mathbf{J}]_{h,l} = \frac{\partial e_{k,q}}{\partial x_l} \quad (\text{C.41})$$

donde x es el vector de parámetros de la red, que tiene la siguiente forma:

$$\mathbf{x}^T = [x_1, x_2, \dots, x_n] = [w_{1,1}^1, w_{1,2}^1, \dots, w_{S^1,R}^1, b_1^1, \dots, b_{S^1}^1] \quad (\text{C.42})$$

Para utilizar este algoritmo en las aplicaciones para redes multicapa, se redefinirá el término sensibilidad de forma que sea más simple hallarlo en cada iteración.

$$s_{i,h}^m \equiv \frac{\partial e_{k,q}}{\partial n_{i,q}^m} \quad (\text{C.43})$$

donde $h = (q-1)S^M + k$

Con la sensibilidad definida de esta manera, los términos de la matriz Jacobiana pueden ser calculados más fácilmente:

$$[\mathbf{J}]_{h,l} = \frac{\partial e_{k,q}}{\partial w_{i,j}^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} * \frac{\partial n_{i,q}}{\partial w_{i,j}^m} = s_{i,h}^m * \frac{\partial n_{i,q}}{\partial w_{i,j}^m} = s_{i,h}^m * a_{j,q}^{m-1} \quad (\text{C.44})$$

y para las ganancias:

$$[\mathbf{J}]_{h,l} = \frac{\partial e_{k,q}}{\partial b_i^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} * \frac{\partial n_{i,q}}{\partial b_i^m} = s_{i,h}^m * \frac{\partial n_{i,q}}{\partial b_i^m} = s_{i,h}^m \quad (\text{C.45})$$

De esta forma, cuando la entrada P_Q ha sido aplicada a la red y su correspondiente salida a_Q^M ha sido computada, el algoritmo de Retropropagación de Levenberg–Marquardt es inicializado con:

$$\mathbf{S}_Q^M = -f^M(\mathbf{n}_Q^M) \quad (\text{C.46})$$

Cada columna de la matriz \mathbf{S}_Q^M debe ser propagada inversamente a través de la red para producir una fila de la matriz Jacobiana. Las columnas pueden también ser propagadas conjuntamente de la siguiente manera:

$$\mathbf{S}_q^m = f^m(\mathbf{n}_q^m) (\mathbf{W}^{m+1})^T \mathbf{S}_q^{m+1} \quad (\text{C.47})$$

La matriz de sensibilidad total para cada capa en el algoritmo de Levenberg–Marquardt es formada por la extensión de las matrices computadas para cada entrada:

$$\mathbf{S}^m = [\mathbf{S}_1^m][\mathbf{S}_2^m] \dots [\mathbf{S}_q^m] \quad (\text{C.48})$$

Para cada nueva entrada que es presentada a la red, los vectores de sensibilidad son propagados hacia atrás, esto se debe a que se ha calculado cada error en forma individual, en lugar de derivar la suma al cuadrado de los errores. Para cada entrada aplicada a la red habrá S^M errores, uno por cada elemento de salida de la red y por cada error se generara una fila de la matriz Jacobiana.

Este algoritmo puede resumirse de la siguiente manera:

1. Se presentan todas las entradas a la red, se calculan las correspondientes salidas y cada uno de los errores según

$$e_q = t_q - a_q^M \quad (\text{C.49})$$

Se calcula después, la suma de los errores cuadrados para cada entrada $e(x)$.

2. Se calculan las sensibilidades individuales y la matriz de sensibilidad total y con estas, se calculan los elementos de la matriz Jacobiana.
3. Se obtiene Δx_k .
4. Se recalcula la suma de los errores cuadrados usando $x_k + \Delta x_k$. Si esta nueva suma es más pequeña que el valor calculado en el paso 1 entonces se divide μ por \mathcal{G} , se calcula $x_{k+1} = x_k + \Delta x_k$ y se regresa al paso 1. Si la suma no se reduce entonces se multiplica μ por \mathcal{G} y se regresa al paso 3.

El algoritmo debe alcanzar convergencia cuando la norma del gradiente de

$$\nabla e(x) = 2\mathbf{J}^T(x)\mathbf{v}(x) \quad (\text{C.50})$$

sea menor que algún valor predeterminado, o cuando la suma de los errores cuadrados ha sido reducida a un error que se haya trazado como meta.

El comportamiento de este algoritmo se visualiza en la Fig. C.8, la cual muestra la trayectoria de convergencia con $\mu = 0.01$ y $\mathcal{G} = 5$.

Este algoritmo converge en menos iteraciones que cualquier otro método, por supuesto requiere mucha más computación por iteración, debido a que implica el cálculo de matrices inversas. A pesar de su gran esfuerzo computacional sigue siendo el algoritmo de entrenamiento más rápido para redes neuronales cuando se trabaja con un moderado número de parámetros en la red, si el número de parámetros es muy grande utilizarlo resulta poco práctico.



Figura C.8. Trayectoria del algoritmo Levenberg-Marquardt

C.6 Descripción de las funciones utilizadas en MATLAB

C.6.1 Red tipo Perceptrón

Las herramientas de redes neuronales de Matlab 6.0 [49] utilizadas en el entrenamiento de las redes neuronales tipo Perceptrón son las siguientes:

- *newp*: Crea una red tipo Perceptrón, que requiere las siguientes entradas:

NET = NEWP(PR,S,TF,LF)

PR Rx2 matriz de valores máximos y mínimos para los *R* elementos de entrada.

S Número de neuronas.

TF Función de Transferencia, en este caso 'hardlims'.

LF Función de aprendizaje, para este caso 'learnp'.

- *rands*: Función simétrica que inicializa aleatoriamente los valores de pesos y ganancias de una red con valores entre -1 y 1; requiere de la estructura:

rands(S,PR), generando una matriz de dimensiones S x PR.

- *adapt*: Permite a una red neuronal adaptarse a los patrones de entrada, esta función tiene la siguiente sintaxis:

[net,Y,E,Pf,Af] = adapt(NET,P,T,Pi,Ai)

net Red que va a crearse

P Entradas a la red; deben aparecer en forma de un arreglo de matrices.

T Salidas esperadas de la red, si no se especifican son ceros por defecto.

Pi Condiciones de retardo para la entrada inicial, por defecto son ceros.

Ai Condiciones de retardo para la capa inicial, por defecto ceros.

- *net.adaptParam.passes*: Número de iteraciones que utiliza el programa.

C.6.2 Red tipo Adaline

Las siguientes son las funciones de las herramientas de Redes Neuronales de Matlab 6.0 utilizadas en el entrenamiento de una red Adaline.

- *newlin*: Función para crea una red tipo Adaline, que requiere las siguientes entradas:

NEWLIN(PR,S,ID,LR)

- R Matriz de Rx2 que contiene los valores máximos y mínimos de cada uno de los R elementos de entrada.
- S Número de neuronas
- ID Arreglo que contiene los valores de los retardos, por defecto todos sus valores son cero.
- LR Rata de aprendizaje, por defecto = 0.01

- *net.inputWeights{1,1}.delays*: Especifica los retardos iniciales
- *net.adaptParam.passes*: Número máximo de iteraciones
- *[net,y,E,pf,af]=adapt(net,p,T,pi)*: Comando de entrenamiento de la red; requiere como entradas la red creada anteriormente, los patrones de entrada, las salidas esperadas y los retardos iniciales, retorna el estado final de la red, los valores obtenidos para cada patrón de entrada con sus correspondientes errores así como los valores finales de los retardos.

C.6.3 Red tipo Retropropagación

La red neuronal de Retropropagación presenta una gran variedad de opciones de configuración, dependiendo de la necesidad de aprendizaje y de la aplicación que se este desarrollando.

- *newff*: Crea una red tipo Retropropagación, requiere que le sean especificados los siguientes parámetros:

newff: (PR,[S1 S2...SNI},{TF1 TF2...TFNI},BTF,BLF,PF)

PR Rx2 Matriz de valores máximos y mínimos de cada uno de las R neuronas de entrada.

Si Número de neuronas para cada una de las capas.

TFi Función de transferencia a utilizar en cada una de las capas, por defecto utiliza *tansig*

BTF Algoritmo de entrenamiento a utilizar, por defecto utiliza *trainlm*

BLF Función de actualización de los pesos, por defecto utiliza *learnngdm*.

PF Función para evaluar el desempeño de la red, por defecto utiliza *mse*.

C.6.4 Algoritmos de entrenamiento para la red de retropropagación

C.6.4.1 Traingd

Algoritmo de pasos descendientes, que actualiza pesos y ganancias variándolos en la dirección negativa del gradiente de la función del error. Es un algoritmo de aprendizaje muy lento, que requiere de la siguiente sintaxis:

- *net.trainParam.epochs*: Máximo número de iteraciones para obtener convergencia

- *net.trainParam.goal*: Error máximo permitido
- *net.trainParam.lr*: Rata de aprendizaje
- *net.trainParam.max_fail*: Máximo número de fallas
- *net.trainParam.min_grad*: Mínimo rendimiento del gradiente
- *net.trainParam.show*: Intervalo de visualización de los resultados
- *net.trainParam.time*: Máximo tiempo de entrenamiento en segundos

Con este algoritmo el aprendizaje de la red se detendrá si el número de iteraciones excede el comando *net.trainParam.epochs*, si se alcanzó el valor del error propuesto como meta, si la magnitud del gradiente es menor que *net.trainParam.min_grad*, o si el tiempo de entrenamiento supera el valor de *net.trainParam.time*.

C.6.4.2 Traingdm

Equivale al algoritmo tradicional, más un nuevo coeficiente de momentum, que interviene en el proceso de actualización de los pesos. Si el error de la red en una iteración dada, excede el valor del error en la iteración anterior, en un valor mayor al definido por un radio de cobertura dado el que puede determinarse por medio de la función *max_perf_inc* y que está típicamente alrededor de 1.04, los nuevos pesos y ganancias son descartados y el coeficiente de momentum *mc* es fijado en cero.

La sintaxis de este algoritmo es igual a la utilizada para el algoritmo *traingd*, más un nuevo comando que permite modificar el coeficiente de momentum:

- *net.trainParam.mc*: Valor fijado para el coeficiente de momentum

C.6.4.3 Traingda

Algoritmo de Gradiente Descendiente, que emplea una rata de aprendizaje adaptiva durante el proceso de entrenamiento. La rata de aprendizaje varía entre 0.01 y 1, una rata de aprendizaje muy pequeña torna lento el aprendizaje, pero si se incrementa demasiado el aprendizaje puede tornarse inestable y crear divergencia, por esto la función *traingda* varía la rata de aprendizaje tratando de sacar provecho de la inclinación del gradiente en cada momento; su gran desventaja es que los pesos iniciales varían muy poco así se encuentren distantes de los valores de convergencia. La sintaxis de este el algoritmo es la siguiente:

- *net.trainParam.epochs*: Máximo número de iteraciones para obtener convergencia
- *net.trainParam.goal*: Error máximo permitido
- *net.trainParam.lr*: Rata de aprendizaje inicial
- *net.trainParam.lr_inc*: Porcentaje que incrementa la rata de aprendizaje cuando el error disminuye
- *net.trainParam.lr_dec*: Porcentaje en que es decrementada la rata de aprendizaje cuando el error aumenta
- *net.trainParam.max_fail*: Máximo número de fallas
- *net.trainParam.max_perf_inc*: Máximo incremento del rendimiento
- *net.trainParam.min_grad*: Mínimo rendimiento del gradiente

- *net.trainParam.show*: Los resultados son visualizados siempre que transcurre este número de iteraciones.
- *net.trainParam.time*: Máximo tiempo de entrenamiento en segundos

C.6.4.4 Trainrp

Las redes multicapa, utilizan típicamente una función de transferencia sigmoïdal en las capas ocultas, estas funciones comprimen un infinito rango de entradas, dentro de un finito rango de salidas, además se caracterizan porque su pendiente tendera cada vez más a cero, mientras más grande sea la entrada que se le presenta a la red, esto ocasiona problemas cuando se usa un algoritmo de entrenamiento de pasos descendientes, porque el gradiente empieza a tomar valores muy pequeños y por lo tanto no habrán cambios representativos en los pesos y las ganancias, así se encuentren bastante lejos de sus valores óptimos. El propósito del algoritmo Backpropagation Resilient (RPROP) es eliminar este efecto en la magnitud de las derivadas parciales. En este algoritmo solamente el signo de la derivada es utilizado para determinar la dirección de actualización de los parámetros, la magnitud de las derivadas no tiene efecto en la actualización. La magnitud en el cambio de cada peso es determinada por separado; el valor del incremento de pesos y ganancias es determinado por el factor *delt_inc*, así la derivada parcial del error con respecto a los pesos tenga el mismo signo durante dos iteraciones sucesivas; el valor de decremento está determinado por el factor *delt_dec* así la derivada del error con respecto a los pesos haya cambiado de signo con respecto a la anterior iteración; si la derivada es cero, entonces el valor actualizado se conserva; si los pesos continúan cambiando en la misma dirección durante varias iteraciones, la magnitud de cambios de los pesos se decrementa.

La sintaxis de este algoritmo se resume a continuación:

- *net.trainParam.epochs*: Máximo número de iteraciones del entrenamiento
- *net.trainParam.show*: Intervalo de visualización de los resultados
- *net.trainParam.goal*: Error deseado
- *net.trainParam.time=inf*: Máximo tiempo de entrenamiento en segundos
- *net.trainParam.min_grad*: Mínimo rendimiento del gradiente
- *net.trainParam.max_fail*: Máximo número de fallas
- *net.trainParam.lr*: Rata de aprendizaje
- *net.trainParam.delt_inc*: Incremento en la actualización de pesos
- *net.trainParam.delt_dec*: Decremento en la actualización de pesos
- *net.trainParam.delta0*: Incremento inicial en la actualización de pesos
- *net.trainParam.deltamax*: Máximo cambio en los pesos

C.6.4.5 Trainbfg

Algoritmo alternativo que emplea la técnica del gradiente conjugado, su expresión matemática se deriva del método de Newton, con la ventaja de que no es necesario computar las segundas derivadas; este algoritmo requiere mas capacidad de almacenamiento que el algoritmo tradicional, pero generalmente converge en menos iteraciones. Requiere de un cálculo aproximado de la matriz Hessiana, la cual es de dimensiones $n^2 \times n^2$, donde n la cantidad de pesos y ganancias de la red; para redes que involucren una gran cantidad de parámetros es preferible emplear el algoritmo trainrp.

- *net.trainParam.epochs*: Máximo número de iteraciones del entrenamiento
- *net.trainParam.show*: Número de iteraciones entre las cuales se muestran resultados
- *net.trainParam.goal*: Error deseado
- *net.trainParam.time=inf*: Máximo tiempo de entrenamiento en segundos
- *net.trainParam.min_grad*: Mínimo rendimiento del gradiente
- *net.trainParam.max_fail=5*: Máximo número de fallas
- *net.trainParam.searchFcn* 'srchcha' Nombre de la rutina de búsqueda lineal a utilizar.
- *net.trainParam.scal_tol*: Se divide entre el valor de Delta para determinar la tolerancia para la búsqueda lineal.
- *net.trainParam.alpha*: Factor de escala que determina una reducción suficiente en el desempeño.
- *net.trainParam.beta*: Factor de escala que determina un tamaño de paso suficientemente grande.
- *net.trainParam.delta*: Tamaño de paso inicial en el intervalo de localización de paso.
- *net.trainParam.gama*: Parámetro para evitar pequeñas reducciones en el desempeño.
- *net.trainParam.low_lim*: Límite inferior en el cambio del tamaño del paso.
- *net.trainParam.up_lim*: Límite superior en el cambio del tamaño del paso.
- *net.trainParam.maxstep*: Máximo longitud de paso.
- *net.trainParam.minstep*: Mínima longitud de paso; por defecto es 1.0e-6
- *net.trainParam.bmax*: Máximo tamaño de paso.

C.6.4.6 Trainlm

Algoritmo que actualiza los pesos y las ganancias de acuerdo a la optimización de Levenberg-Marquardt. Es el algoritmo más rápido para redes Backpropagation; tiene la desventaja de requerir de un set de entrenamiento lo más estándar posible, pues de otra forma solo aproximará correctamente valores que se encuentren dentro de los patrones de aprendizaje. Si el set de entrenamiento es muy extenso, se recomienda reducir el Jacobiano.

La sintaxis de este algoritmo es la siguiente:

- *net.trainParam.epochs*: Máximo número de iteraciones del entrenamiento
- *net.trainParam.goal*: Error deseado
- *net.trainParam.lr*: Rata de aprendizaje
- *net.trainParam.max_fail*: Máximo número de veces que falla el valor de Mu
- *net.trainParam.mem_reduc*: Factor de fraccionamiento de Jacobiano para ahorrar memoria
- *net.trainParam.min_grad*: Mínimo rendimiento del gradiente
- *net.trainParam.show*: Intervalo de visualización de los resultados.
- *net.trainParam.time*: Máximo tiempo de entrenamiento en segundos
- *tr.mu*: Valor del Mu adaptivo

Apéndice D

Artículo presentado en The 40th International Universities Power Engineering Conference (UPEC-2005) en Cork, Irlanda, dentro del área de *Máquinas Eléctricas y Accionamientos*.

USING AN ARTIFICIAL NEURAL NETWORK AS A ROTOR RESISTANCE ESTIMATOR IN THE INDIRECT VECTOR CONTROL OF AN INDUCTION MOTOR

P. Huerta González⁽¹⁾ J. Rodríguez Rivas⁽¹⁾ E. Guzmán Serrano⁽²⁾
 (1) National Polytechnic Institute, Mexico (2) Metropolitan Autonomous University, Mexico.

ABSTRACT

This paper presents a rotor resistance estimator based on an artificial neural network (ANN) used in the indirect vector control (IVC) of an induction motor (IM). Attention is focused on the dynamic performance of ANN rotor estimator, which gives superior performance over the fuzzy logic based rotor estimator reported in technical literature. The simulation was done using a 20 HP induction motor. The same ANN rotor estimator was proved with other IM having different rated powers. The use of the same ANN was possible because the scaling and descaling (normalization) of the input and output of ANN was property done for each motor. The ANN training was done offline using the Levenberg-Marquardt algorithm. The neuronal network is a three-layer network; the first layer has fourteen neurons (or nodes), the hidden layer has five neurons and the output layer has only one neuron because the unique output signal is the rotor resistance value.

Keywords: Induction motor vector control, rotor resistance estimation, artificial neural network.

INTRODUCTION

Vector control or field-oriented control is the most popular method of obtaining high performance in induction motor drives. There are essentially two general methods of vector control. One, called the direct or feedback method, and the other, the indirect or feedforward method. Indirect vector-controlled (IVC) induction motor (IM) drives used in high-performance systems are very popular in industrial applications due to their relative simple configuration, as compared to the direct method which require flux and torque estimators [1]-[2]. IVC eliminates the need for a flux model but require an accurate measurement of shaft position in order to determine the precise location of the rotor flux space vector or phasor. In an IVC induction motor drive, the flux, torque, and slip commands are calculated from the IM variables based on machine's parameters. It is desirable that these parameters match the actual parameters of the machine at all operating conditions to achieve decoupling control of the machine. The control performance is thus sensitive to the system parameters, in particular to the rotor resistance which changes significantly with temperature and skin effect. The estimation of rotor resistance is done in order to update its value into the control and keep the IVC tuned.

The estimation of rotor resistance in an IVC induction motor drive has been subject of several research works [3]-[7]. Despite all these efforts, rotor resistance estimation remains a difficult problem. In this paper, a rotor resistance estimator based on artificial neural networks (ANN) is presented. The ANN simulation results are compared with the ones obtained by a fuzzy logic estimator presented in the literature. The

effectiveness of the proposed rotor resistance estimator for a 20 hp IM is then demonstrated by simulation. The use of the ANN rotor estimation for other rated power IM is also shown.

IVC USING AN ANN ROTOR RESISTANCE ESTIMATOR

Figure 1 shows IVC scheme with an ANN rotor resistance estimator. The implementation of IVC is based in the following equations [2]:

$$i_{ds}^{e*} = \psi_{dr}^{e*} \frac{r_r' + L_r' P}{r_r' L_m} \quad (1)$$

$$i_{qs}^{e*} = \frac{2}{3} \frac{2}{P} \frac{L_r'}{L_m} \frac{T_{em}^*}{\psi_{dr}^{e*}} \quad (2)$$

$$\omega_2^* = \omega_e - \omega_r = \frac{r_r' i_{qs}^{e*}}{L_r' i_{ds}^{e*}} \quad (3)$$

where r_r' , L_r' , L_m are the rotor resistance and the rotor and the magnetizing inductances respectively; T_{em}^* , ψ_{dr}^{e*} are the electromagnetic torque and rotor flux reference values respectively, i_{ds}^{e*} , i_{qs}^{e*} are the reference values of stator current d-q components in the synchronous reference frame; P is the number of pole pairs; ω_2^* , ω_e , ω_r are the slip (reference value), synchronous and rotor frequency respectively and P is the Laplace operator.

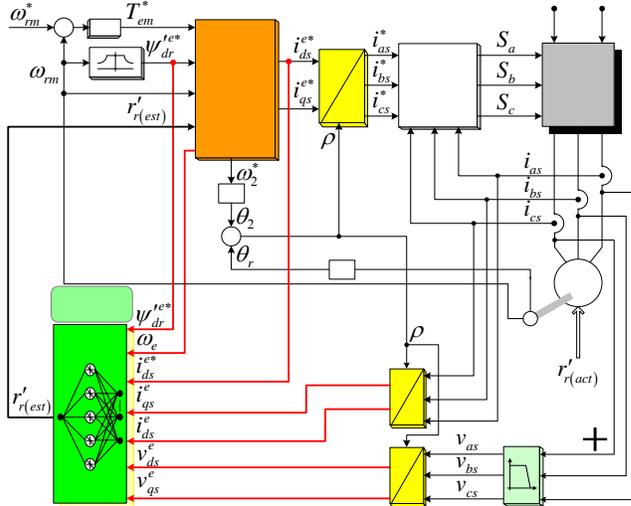


Fig.1. Indirect vector-controlled IM drive with an ANN rotor resistance estimator.

ANN rotor estimator

The rotor estimator shown in figure 1 is described in more detailed in figure 2. As it is observed in this figure, the ANN has fourteen inputs because each one of the seven input variables is entered to ANN together with a one-step time delayed variable value.

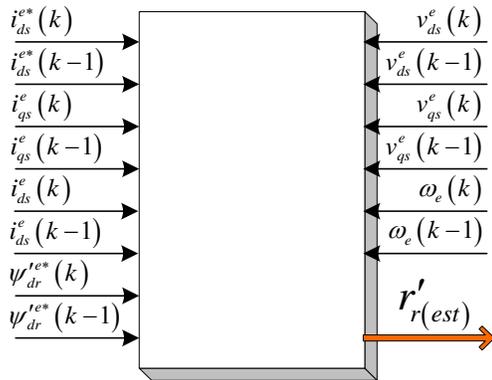


Fig.2. Input variables in ANN rotor resistance estimator.

Figure 3 shows the training method of the ANN. The input/output data patterns were generated from a system model based on fuzzy logic [5], [8]-[9]. The training was done considering both the starting and steady state operating condition of IM vector control. Three different values of rotor resistances were employed during the training procedure. The inputs of ANN rotor resistance estimator are: stator voltages and currents components in synchronous reference frame, reference stator flux, d-axis component stator reference current and synchronous electrical speed. Besides these seven inputs seven delayed inputs were also considered.

The neuronal network is a three-layer network; the first layer has fourteen neurons (or nodes), the hidden layer

has five neurons and the output layer has only one neuron because the unique output signal is the rotor resistance value. The training procedure used was the back propagation algorithm and it was carried on using the Matlab/Simulink package. The input/output example data patterns are gathered from the simulated system because a system model is available as it can be seen in the top side of the figure 3. The network is initialized with random positive and negative weights to avoid saturation before training starts. With one input pattern, the output is calculated and compared with the desired output pattern. The weights are then changed until the error between the calculated pattern and the desired pattern is very small and acceptable. A similar training is done with all the patterns, in order to make them match. At this point, the network is said to have been trained satisfactorily.

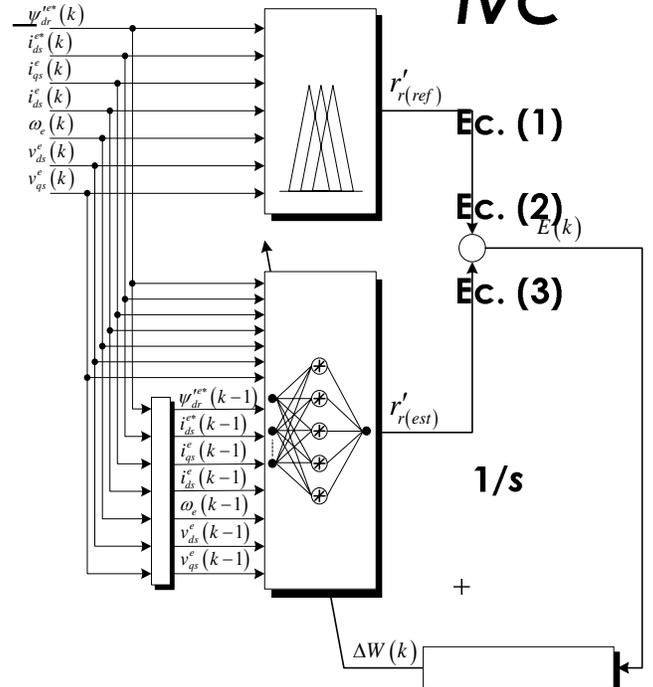


Fig.3. Training scheme of the ANN rotor resistance estimator.

System model based on fuzzy logic

Figure 4 shows the system model used in the training procedure of ANN. The difference between the functions

$$F_{est} = -i_{ds}^{e*} \psi_{dr}^{e*} \omega_e \quad (4)$$

$$F_{act} = \frac{L'_r}{L_M} \left[(i_{qs}^e v_{ds}^e - i_{ds}^e v_{qs}^e) + (i_{qs}^{e2} + i_{ds}^{e2}) L_s \sigma \omega_e \right] \quad (5)$$

reflects the variation of rotor resistance [5], [8]-[9]. The superscript * means reference values instead of measured values.

The system model shown in figure 4 is used in [9] as rotor resistance estimator. In the present paper this

scheme is used as model in the training procedure of an ANN. The simulating results show that the rotor resistance based on ANN has a better dynamic performance than the rotor resistance estimator shown in [9] (figure 4) even when different rated power of IM are used with the same ANN estimator. In this case only one training procedure was needed. The use of the same ANN was possible for different motors because the scaling and descaling (normalization) of the ANN input and output was properly done for each motor. The error $E_F(k)$

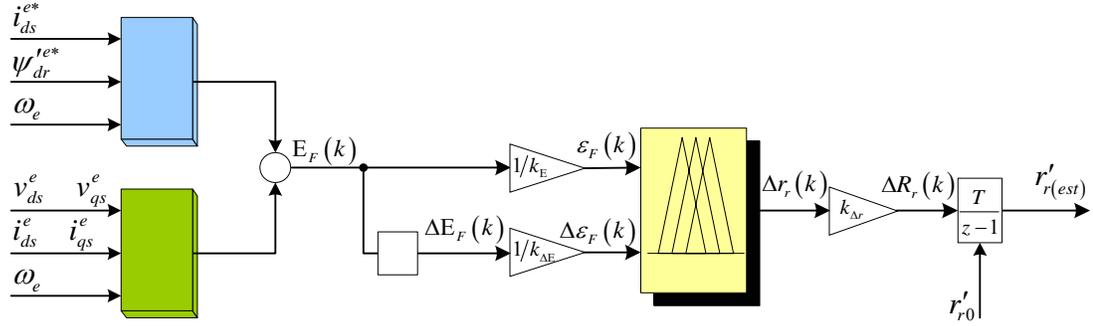


Fig.4. System model based on fuzzy logic

fuzzy logic controller, which consists of tree stages:

fuzzification, inference, and defuzzification. The $\varepsilon_F(k)$

and $\Delta\varepsilon_F(k)$ fuzzification stage inputs signals are per unit (p.u.) signals computed from the actual $E_F(k)$ and $\Delta E_F(k)$ by dividing them by the respective gain factors k_E

and Δk_E . The crisp variables $\varepsilon_F(k)$ and $\Delta\varepsilon_F(k)$ are converted into fuzzy variables ε_F and $\Delta\varepsilon_F$ using triangular membership functions.

In the second stage of the estimator, variables $\varepsilon_F(k)$

and $\Delta\varepsilon_F(k)$ are processed by an inference engine (block FIS in fig. 4) that executes 49 rules (7x7) as shown in table 1, where NL, NM, NS, ZE, PS, PM, PL correspond to Negative Large, Negative Medium, Negative short, Zero, Positive Short, Positive Medium, and Positive Large respectively.

In the defuzzification stage, a crisp value for the output variable $\Delta r_r(k)$ is obtained by the height method. The calculated value of the incremental resistance $\Delta R_r(k)$ is then obtained by multiplying $\Delta r_r(k)$ by the gain factor $k_{\Delta r}$. The value of estimated rotor resistance is then obtained by integrating the output signal:

$$r'_{r(est)} = R_r(k-1) + k_{\Delta r} \Delta r_r(k) \quad (8)$$

between F_{est} and F_{act} and its time variation $\Delta E_F(k)$ are then calculated as:

$$E_F(k) = F_{est}(k) - F_{act}(k) \quad (6)$$

$$\Delta E_F(k) = E_F(k) - E_F(k-1) \quad (7)$$

These variables are used as input for the estimator based on fuzzy logic. The internal structure of the fuzzy logic rotor resistance estimator is chosen similar to that of a

Note that the rated value of rotor resistance is taken as initial value for this integral.

The estimate valor $r'_{r(est)}$ is used in the slip calculator (equation 3) and rotor flux (d-component stator current) estimator to ensure the correct field orientation operation of the drive.

Table1. Rule base for rotor resistance determination.

$\Delta\varepsilon_F$	NL	NM	NS	ZE	PS	PM	PL
ε_F							
NL	NL	NM	NS	NL	PS	PM	PL
NM	NL	NM	NS	NM	PS	PM	PL
NS	NL	NM	NS	NS	PS	PM	PL
ZE	NL	NM	NS	ZE	PS	PM	PL
PS	NL	NM	NS	PS	PS	PM	PL
PM	NL	NM	NS	PM	PS	PM	PL
PL	NL	NM	NS	PL	PS	PM	PL

SIMULATION RESULTS

The simulation was done using the Matlab/Simulink package. Figure 5 shows the simulated transient response of both fuzzy logic based and ANN based rotor resistance estimator using 20 hp rated power motor drive (see appendix). The transient variation of reference and actual speed, and electromagnetic and load torque are shown in

the figures 5 (a) and (b) respectively. Figures 5 (c), (e), (g), and (i) show the rotor flux responses when speed and torque vary as they are shown in figure 5 (a) and (b). The responses of fuzzy logic based and ANN based rotor resistance estimator for keeping the actual rotor resistance constant are observed in the figures 5 (d) and (f) respectively. The response of both estimators, when the actual rotor resistance varies, is shown in figures 5 (h) and (j). Regardless the ANN was training for the 20 hp rated power IM using the rotor resistance fuzzy logic based estimator as system model, the reached performance of ANN rotor resistance estimator is much better than fuzzy logic based estimator as it can be observed in figure 5. Figure 5 shows the influence of rotor resistance estimation on the IVC performance (see

rotor flux response); from this figure it is clear that when the estimation of rotor resistance is improved, the vector control detuning due to rotor resistance mismatch is almost eliminated.

Figure 6 shows the transient responses of IVC and ANN based rotor resistance estimator using 100 hp rated power MI (see appendix). The same ANN can be used in the rotor resistance estimator by properly changing the scaling and descaling (normalization) of the input and output of ANN. As it is observed in this figure, an excellent performance of rotor estimator is obtained regardless that a different motor is used. This means that the ANN has a good generalization behavior. Figure 7

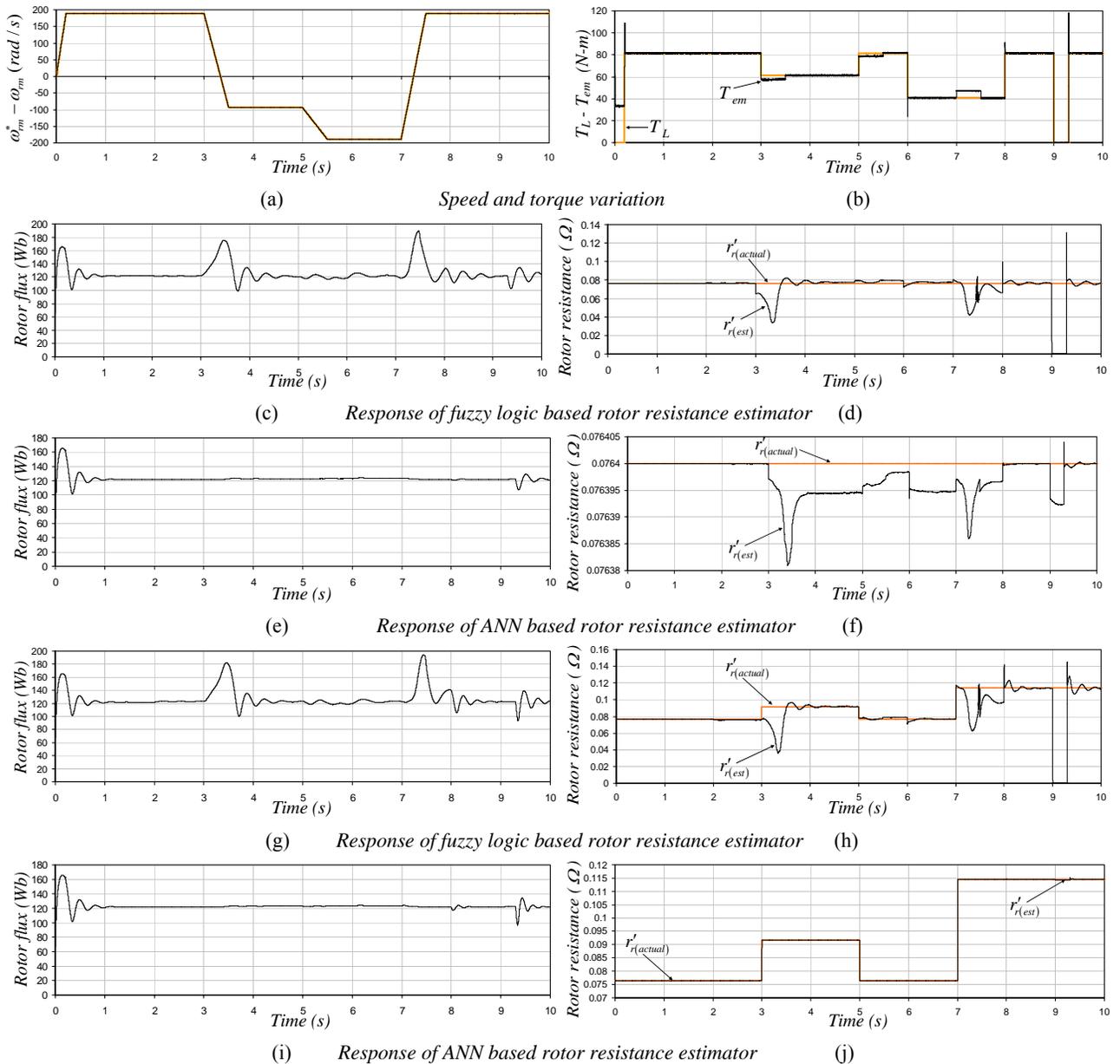


Fig. 5. Simulation transient response of fuzzy logic based and ANN based rotor resistance estimator for a 20 hp rated power IM

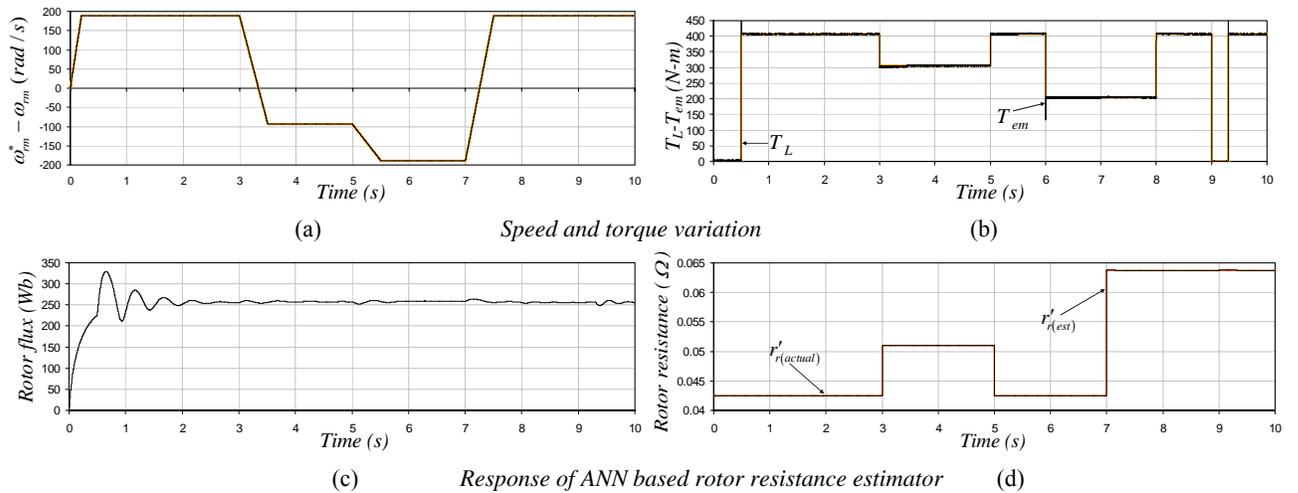


Fig. 6. Simulation transient response of ANN based rotor resistance estimator for a 100 hp rated power IM

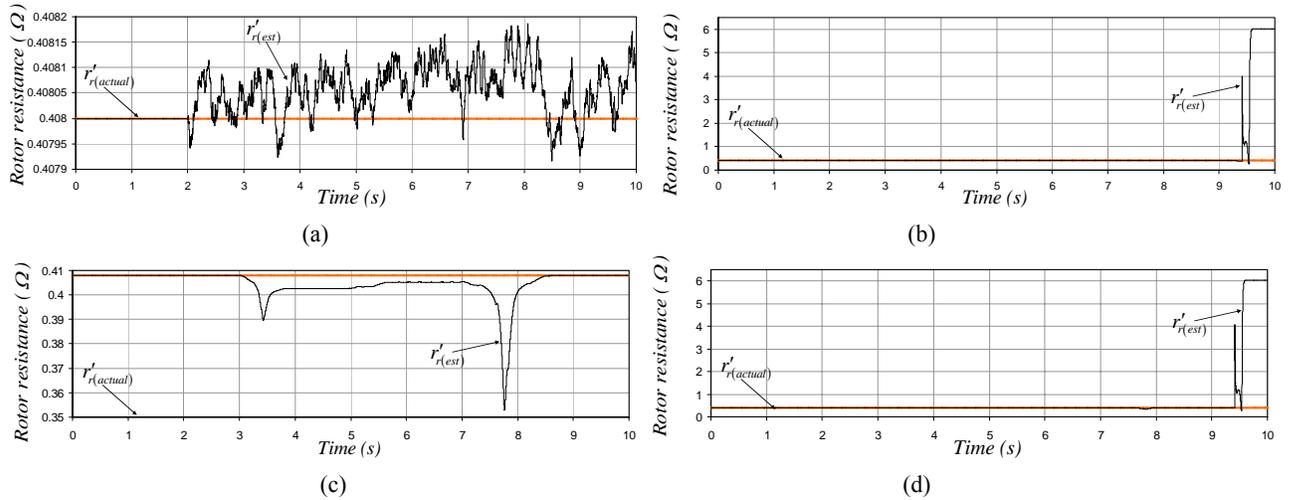


Fig. 7. Simulation transient response of ANN based rotor resistance estimator for a 5 hp rated power IM

shows the steady-state ANN rotor resistance estimation response considering a 5 hp rated power IM drive (see appendix). In figure 7 (a) rated speed and electromagnetic torque is considered. In figure 7 (b) the rotor speeds is kept constant and equal to rated rotor speed, while load torque varies in the same rate shown in figure 5 (b). The estimation of rotor resistance is lost at 9.3 s, when the load torque varies from zero to rated value instantaneously. Figure 7 (c) shows the response when the electromagnetic torque is equal to its rated value and the rotor speed varies as in figure 5 (a). In figure 7 (d) the rotor speed and electromagnetic torque varies at the same time, again the estimation fails at 9.3 s when the load torque varies from zero to rated value.

CONCLUSIONS

This paper has presented an ANN based rotor resistance estimator used to update its value in indirect-vector control of induction motor drives. Base on obtained

results, the proposed estimator has much better dynamic performance than fuzzy based rotor resistance estimator presented in the technical literature though the training of ANN was done using this fuzzy based estimator as system model. The obtained simulation results show that the ANN based estimator can be extended to other IM getting an acceptable performance, even in dynamic conditions.

APPENDIX

Induction motor parameters

20 hp, 60 Hz, 220 V, 1750 r/min, $r_s = 0.1062 \Omega$, $r_r' = 0.0764 \Omega$, $x_m = 5.834 \Omega$, $x_{ls} = x_{lr} = 0.2145 \Omega$, $J = 2.8 \text{ kgm}^2$.

100 hp, 60 Hz, 460 V, $r_s = 0.0425 \Omega$, $r_r' = 0.0425 \Omega$, $x_m = 8.51 \Omega$, $x_{ls} = x_{lr} = 0.284 \Omega$, $J = 2.0 \text{ kgm}^2$.

5 hp, 60 Hz, 230 V, $r_s = 0.531 \Omega$, $r_r' = 0.408 \Omega$, $x_m = 31.931 \Omega$, $x_{ls} = x_{lr} = 0.95 \Omega$, $J = 0.1 \text{ kgm}^2$.

REFERENCES

1. Bose, B.K., Modern power electronics and ac drives, *Prentice Hall PTR*, 2002.
2. Chee-Mun, O., Dynamic Simulation of Electric Machinery, *Prentice Hall PTR*, 1998.
3. Matsuo, T., and Lipo, T.A., A rotor parameter identification scheme for vector-controlled in induction motor drives, *IEEE Trans. Ind. Appl.*, Vol. IA-21, No. 4, pp. 624-632, May/June 1985.
4. Nordin, K.B., Novotny D.W., and Zinger, D.S., The influence of motor parameter in feedforward field orientation drive systems, *IEEE Trans. Ind. Appl.*, Vol. IA-21, No. 4, pp. 1009-1015, July/Aug. 1985.
5. Koyama, M., Yano, I., Kamiyama, I., and Yano, S., Microprocessor-based vector control system for induction motor drives with rotor time constant identification function, *IEEE/IAS Conf. Record*, pp. 564-569, 1985.
6. Krishnan, R., and Bharadwaj, A.S., A review of parameter sensitivity and adaptation in indirect vector controlled induction motor drive systems, *IEEE Trans. on Power Elect.*, Vol. 6, No. 4, pp. 695-703, Oct. 1991.
7. Krishnan, R., and Doran, F.C., Study of parameter sensitivity in high performance inverter-fed induction motor drive systems, *IEEE Trans. on Industry Appl.*, Vol. IA_23, No. 4, pp. 623-635, July/Aug. 1987.
8. Kazmierkowski, M.P., Krishnan, R., and Blaabjerg, F., Control in power electronics, *Academic Press*, 2002.
9. Ta-Kao, M., and Le-Huy, H., Rotor resistance estimation using fuzzy logic for high performance induction motor drives, *IECON'98, Proceeding of the 24th Annual Conference of IEEE*, Vol. 1, pp. 303-308, 31 Aug.- 4 Sept., 1998.

AUTHOR'S ADDRESS

The authors can be contacted to:

National Polytechnic Institute
Mechanic and Electric Engineering School
Graduate Program in Engineering Electrical
Mexico, D.F.

phuertaglez@hotmail.com

jjrodriguezr@ipn.mx

gse@correo.azc.uam.mx

BIBLIOGRAFÍA

- [1] A. Ba-razzouk, A. Cheriti and G. Olivier. "Artificial Neural Networks Rotor Time Constant Adaptation in Indirect Field Oriented Control Drives", *IEEE-PESC, Baveno*, pp. 701-707, June 1996.
- [2] A. Ba-razzouk, A. Cheriti and G. Olivier. "A Neural Networks Based Field Oriented Control Scheme for Induction Motors", *IEEE-IAS proceedings, New Orleans, Louisiana*, pp. 804-811, October 1997.
- [3] E.Y.Y. Ho and P.C. Sen. "Decoupling Control of Induction Motor Drives", *IEEE Trans. on Industrial Electronics*, Vol. 35, No. 2, pp. 253-262, May 1988.
- [4] L. Kreindler, J.C. Moreira, A. Testa, T. A. Lipo, "Direct Field Orientation Controller Using the Stator Phase Voltage Third Harmonic", *IEEE-IAS proceedings*, pp. 508-514, October 1992.
- [5] A. Ba-razzouk, A. Cheriti and G. Olivier and P. Sicard. "Field Oriented Control of Induction Motors Using Neural Networks Decouplers", *IEEE-IECON, Orlando*, pp. 1428-1433, November 1995.
- [6] T. Matsuo, V. Blasko. J.C. Moreira and T.A. Lipo, "Field Oriented Control of Induction Machines Employing Rotor End Ring Current Detection", *Trans. on Power Electronics*. Vol. 9, No. 6, pp. 638-645. November 1994.
- [7] L. Umanand, and S.R. Bhat, "Adaptation of the Rotor Time Constant for Variation in the Rotor Resistance of an Induction Motor", *IEEE, Annual Meeting*, pp. 738-743. 1994.
- [8] W. Li and R. Venkatatesan. "A New Adaptive Control Scheme for Indirect Vector Control System", *IEEE-IAS, Conference Record*, pp. 524-530, October 1992.
- [9] T. Rowan, R. Kerkman and D. Leggate, "A Simple On-Line Adaptation for Indirect Field Orientation of an Induction Machine", *IEEE Transaction on Industry Applications*, Vol. IA-37, pp. 720-727, July/August 1991.
- [10] S. Wade, M.W. Dunnigan and B.W. Williams, "Parameter Identification for Vector Controlled Induction Machine", *Conf. Rec. IEE Control '94*, Vol. 2, No. 389, pp. 1187-1192.
- [11] T. Du, P. Vas and F. Stronach, "Design and Applications of Extended Observers for joint State and Parameter Estimation in High-Performance AC Drives", *Proc. IEE Electr. Power appl.*, Vol. 142, No. 2, pp. 71-78. 1995.
- [12] K. Ohnishi, Y. Ueda and K. Miyachi, "Model Reference Adaptive System against Rotor Resistance Variation in Induction Motor Drive", *IEEE Trans. Ind. Electron.*, Vol. IE-33, pp. 217-223. June 1986.
- [13] M. Boussak, G.A. Capolino, "Recursive Least Squares Rotor Time Constant Identification for Vector Controlled Induction Machine", *Elec. Mach. Power Syst.*, Vol. 20, No. 2, pp. 137-147. 1992.
- [14] L. R. Valdenebro, E. Bim, "Fuzzy Optimization for Rotor Time Constant Identification of an Indirect Vector-Controlled Induction Motor Drive", in *Conf. IEEE-ISIE'99*. Bled, Slovenia, pp. 504-509.

- [15] M. P. Kazmierkowski, L. Malesani, "Current Control Techniques for Three-Phase Voltage-Source PWM Converters: A Survey", *IEEE Trans. Industrial Electronics*, vol. 45, no. 5, 1998.
- [16] Chee-Mun Ong. *Modern Dynamic Simulation of Electric Machinery*, Prentice Hall, N.Y., 1998.
- [17] Tseng, H.C. and Hwang, V.H., "Servocontroller tuning with fuzzy logic", *IEEE Trans. Ctrl. Syst Tech* 1-4, pp. 262-269, 1993.
- [18] Hyung-Soo Hwang, Jeong-Nae Choi, Won-Hyok Lee, Jin-Kwon Kim "A Tuning Algorithm for the PI Controller Utilizing Fuzzy Theory", *IEEE*, 1999.
- [19] Y. Nishikawa, N. Sannomiya, T. Ohta, and H. Tanaka, "A method for auto-tuning of PID control parameters", *Automatica*, vol. 20, no. 3, pp. 321-332, 1984.
- [20] Su, C.Y. and Stepanenko, Y., "Adaptive control of a class of nonlinear systems with fuzzy logic", *IEEE Trans. Fuzzy Syst* 2-4, pp. 285-294, 1994.
- [21] M. Kawakami, M. Miyajima, K. Ishihara, and H. Naitoh, "Automatic gain tuning of speed and vector controls for induction motor", 1990 National Convention Record *IEEE IAS*, no. 132, 1990.
- [22] K. S. Martin, R. A. Chavarriaga, R. A. Nuñez, C. A. Peña "Controladores Adaptables basados en Mecanismos de Inferencia Difusa", *Pontificia Universidad Javeriana Cali, Ingeniería Electrónica, Control Difuso*, 1999.
- [23] H. J. Zimmerman, *Fuzzy Set Theory and Its Application*. Kluwer Academic Publishers, Boston, 1991.
- [24] C. Wang, D. W. Novotny, and T. A. Lipo, "An automated rotor time constant measurement system for indirect field oriented drives", *IEEE IAS Conf. Rec.*, pp. 140-146, 1986.
- [25] R. D. Lorenz, "Tuning of field oriented Induction Motor controllers for high performance applications", *IEEE IAS Conf. Rec.*, pp. 607-612, Oct. 1985.
- [26] J. W. Choi, S. K. Sul, "New Current Control Concept-Minimum Time Current Control in 3-Phase PWM Converter", in *Conf. Rec. IEEE-Pesc'95*, 1995, pp. 332-338.
- [27] J. Holtz, B. Bayer, "Optimal Pulsewidth Modulation for AC Servos and Low-Cost Industrial Drives", in *Conf. Rec. IEEE-IAS Annu. Meeting*, Houston, Tx, 1992, pp. 1010-1017.
- [28] J. O. Krah, J. Holtz, "High Performance Current Regulation for Low Inductance Servo Motors", in *Conf. Rec. IEEE-IAS Annu. Meeting*, 1998, pp. 490-499.
- [29] R. D. Lorenz and D. B. Lawson, "A simplified approach to continuous, on-line Tuning of field oriented Induction Machine drives", *IEEE IAS Conf. Rec.*, pp. 444-449, Oct. 1988.
- [30] S. N. Vukosavic and M. R. Stojic, "On-line Tuning of the rotor time constant for vector-controlled induction motor in position control applications", *IEEE Transactions on Industrial Electronics*, vol. 40, No. 1, pp. 130-138, Feb. 1993.
- [31] T. Rowan and R. Kerkman, "A New Synchronous Current Regulator and an Analysis of Current-regulated PWM Inverters", *IEEE Trans. Industrial Applications*, vol. IA-22, No. 4, pp. 678-690, July/August 1986.
- [32] M. P. Kazmierkowski, R. Krishnan, Frede Blaabjerg, *Control in Power Electronics*. Academic Press, San Diego, California, 2002.
- [33] D. W. Novotny and T. A. Lipo, *Vector Control and Dynamics of AC Drives*. Clarendon Press Oxford, New York, 2000.

- [34] L. Garces, "Parameter adaptation for the speed controlled static AC drive with a squirrel cage induction motor". IEEE Trans. Industrial Applications, IA-16, pp. 173-178, 1980.
- [35] R. Krishnan and F. C. Doran, "Study of parameter sensitivity in high performance inverter-fed Induction Motor Drive system". Proc. IEEE/IAS'84, pp. 510-524, 1984.
- [36] W. Leonhard, *Control of Electrical Drives*, 2nd ed. Springer-Verlag, Berlin, Germany, 1996.
- [37] P. J. Costa Branco, "A simple adaptive scheme for indirect field orientation of an induction motor". ETEP 7, pp. 243-249, 1997.
- [38] R. Nilsen and M. P. Kazmierkowski, "Reduced-order observer with parameter adaptation for fast rotor flux estimation in induction machines". Proc. IEE, Pt. D, Vol. 136, No. 1, pp. 35-43, 1989.
- [39] T. Orłowska-Kowalska, "Application of extended Luenberger observer for flux and rotor time constant estimation in induction Motor Drives". IEEE Proc. 136, Pt. D, pp. 324-330, 1989.
- [40] M. Koyama, M. Yano, I. Kamiyama, and S. Yano, "Microprocessor-based vector control system for induction motor drives with rotor time constant identification function" in IEEE/IAS Conf. Record, 1985, pp. 564-569.
- [41] Bimal K. Bose. *Modern Power Electronics and AC Drives*, Prentice Hall, N.Y., 2002.
- [42] M. Ta-Cao, and H. Le-Huy, "Model reference adaptive fuzzy controller and fuzzy estimator for high performance induction motor drives", in IEEE/IAS Conf. Record, San Diego, Oct. 1996, pp. 380-387.
- [43] M. Ta-Cao, and H. Le-Huy, "Rotor resistance estimation using fuzzy logic for high performance induction motor drives", in IEEE, 1998, pp. 303-308.
- [44] R. D. Lorenz, and D. B. Lawson, "Simplified approach to continuous on-line tuning of field-oriented induction machine drives", Proc. IEEE/IAS Annual Meeting, pp. 444-449, Oct. 1988.
- [45] Kung, Y. S. and Liaw, C. M., "A fuzzy controller improving a linear model following controller for motor drives", IEEE Transactions on fuzzy systems **2-3**, pp. 194-202, 1994.
- [46] T. Okuyama, H. Nagase, Y. Kubota, H. Horiuchi, K. Miyasaki and S. Ibori, "High performance AC speed control system using GTO converters", Proc. IPEC-Tokyo, pp. 720-731, Oct. 1983.
- [47] J. D. M. Murphy, F. G. Turnbull, *Control Power Electronics of Ac Motors*, Pergamon, N.Y., 1988.
- [48] Hilerá José R, Martínez Víctor J. "Redes neuronales artificiales. Fundamentos, Modelos y Aplicaciones". Ra-ma editorial. Madrid, 1995.
- [49] Matlab 6.0 Help Neural Network pdf.
- [50] Freeman James, Skapura David. "Redes Neuronales: Algoritmos, Aplicaciones y Técnicas de Programación". EUA. Addison Wesley Iberoamericana SA, 1993.
- [51] KOSKO, B., "Neural Networks and Fuzzy Systems", Prentice Hall International Editions, USA, 1992.
- [52] S. Haykin, *Neural Networks, A Comprehensive Foundation*, Macmillan College Publishing Company, 1994
- [53] Arnol S. Kulkarni, M. A. El-Sharkawi. "Speed Estimator for Induction Motor Drives Using and Artificial Neural Network", IEEE, 0-7803-3946-0/97, 1997.

- [54] F. Blaschke, "The Principle of Field Orientation as applied to the new TRANSVECTOR closed loop control system for rotating field machines", Siemens Rev., vol. 34, pp. 217-220, 1972.
- [55] S. Ogasawara, H. Akagi, and A. Nabac, "The Generalized theory of indirect vector control for AC machines", IEEE Trans. Ind. Appl., vol. 24, no. 3, pp. 470-478, May/June 1988.
- [56] R. Krishnan, and A. S. Bharadwaj, "A review of parameter sensitivity and adaptation in indirect vector controlled induction motor drive systems", IEEE Trans. Power Electronics, vol. 6, No. 4, pp. 695-703, 1991.
- [57] Ponce Cruz Pedro. "Simulación Digital del Control Vectorial de Velocidad de un Motor de Inducción". Tesis de Maestría en Ingeniería Eléctrica, SEPI-ESIME-IPN, 1998.
- [58] B. J. Seibel, T. M. Rowan, and R. J. Kerkman, "Field Oriented Control of an Induction Machine with DC Link and Load Disturbance Rejection", Rockwell Automation, Allen-Bradley Standard Drives Business, Nov, 1996.