



**INAOE**

# **Estimación de parámetros de un motor de inducción usando redes neuronales y sistemas neurodifusos**

por

**Ing. Miguel Antonio Cuautle Pat**

Tesis sometida como requisito parcial para obtener el grado de

**Maestro en Ciencias en el Área de Electrónica**

en el

**Instituto Nacional de Astrofísica, Óptica y Electrónica**

Tonantzintla, Puebla

Supervisada por:

**Dr. Israel Cruz Vega**

**Docente Catedrático CONACYT-INAOE**

©INAOE 2018

El autor otorga al INAOE el permiso de reproducir y distribuir copias en su totalidad o en parte de esta tesis





# Estimación de parámetros de un motor de inducción usando redes neuronales y sistemas neurodifusos

Tesis de Maestría

POR:

**Miguel Antonio Cuautle Pat**

ASESOR:

**Dr. Israel Cruz Vega**

Instituto Nacional de Astrofísica Óptica y Electrónica  
Coordinación de Electrónica



*Dedicado a mis padres, Antonio Cuautle Ixtlamati y Fatima Angelica Pat Avila.  
A mi esposa e hijo, Estefania Rosario Portillo Vera y Leonardo Antonio Cuautle  
Portillo; porque no es la carne y la sangre, sino el corazón lo que nos hace padres e  
hijos, gracias por todo.*



# Agradecimientos

---

A mis padres, por su inigualable apoyo para mi preparación profesional, siempre creyendo en mí en todo momento.

A mi asesor de tesis: el Dr. Israel Cruz Vega, que con su dedicación y enseñanza pudimos concluir con esta tesis.

A mis sinodales: por su experiencia en los comentarios y correcciones para concluir esta tesis.

A mis profesores, que durante los cursos nos impartieron consejos, apoyo y enseñanzas; que sin educación no hay esperanza.

A mis amigos y compañeros que hicieron de esta experiencia de vida algo inolvidable: Cris, Carlos Javier, Jorge, Carlos Ramos, Cesar, Bety, Carmona y a lo que me faltaron por nombrar; por su convivencia, ayuda y apoyo, hicieron más agradable este período de estudios.

A esta extraordinaria Institución, por brindarme el espacio y apoyo de mi desarrollo profesional, además de las facilidades otorgadas.

A CONACyT, por la beca otorgada para la realización de estos estudios.





# Resumen

---

En esta tesis se presenta la estimación de parámetros de un motor de inducción usando redes neuronales y sistemas neurodifusos; tales parámetros son el flujo del rotor, la posición del flujo del rotor y la velocidad del motor. Estos parámetros son indispensables para un esquema de control vectorial. La metodología se basa en la búsqueda de arquitecturas que puedan estimar estos parámetros mediante el entrenamiento de redes neuronales y sistemas neurodifusos. La elección de las arquitecturas se basa en un razonamiento de tipo heurístico, realizando pruebas y observando el comportamiento del error medio cuadrático. Una vez obtenidas las arquitecturas, se llevo a cabo la descripción de ellas en Síntesis de Alto Nivel, usando la herramienta *Vivado HLS* de *Xilinx*. Tal descripción se realizó en C++, donde se pudieron observar resultados del uso de recursos de hardware y de latencia, para una posterior implementación en FPGA. Se analizó la validación de las arquitecturas por medio de *System Generator*, que permite el modelado en Simulink con el fin de explorar el funcionamiento sin el uso de trasladar el diseño a hardware.



# Abstract

---

This thesis presents the estimation of parameters of an induction motor using neural networks and neurofuzzy systems; such parameters are the rotor flux, the position of the rotor flux and the motor speed. These parameters are indispensable for a vector control scheme. The methodology is based on the search of architectures that can estimate these parameters through the training of neural networks and neurofuzzy systems. The selection of the architectures is based on a heuristic reasoning, making tests and observing the behavior of the root mean square error. Once the architectures were obtained, they were described in High-Level Synthesis, using *Vivado HLS*, a tool of *Xilinx*. This description was accomplished in C++, where results of the use of hardware utilization and latency are presented, for a later implementation on a FPGA. The validation of the architectures was analyzed by *System Generator*, which allows the modeling in Simulink in order to explore the operation without the use of tracing the design to hardware.



# Tabla de Contenido

---

Agradecimientos	III
Resumen	v
Abstract	VII
Lista de Figuras	XI
Lista de Tablas	XV
Nomenclatura	XVII
<b>1. Introducción</b>	<b>1</b>
1.1. Planteamiento del problema . . . . .	3
1.2. Justificación . . . . .	4
1.3. Objetivo . . . . .	4
1.3.1. Objetivos específicos . . . . .	4
1.4. Organización de la tesis . . . . .	5
<b>2. Motor de Inducción</b>	<b>7</b>
2.1. Construcción del motor de inducción . . . . .	7
2.2. Funcionamiento del motor de inducción . . . . .	9
2.2.1. Modelo estático de un motor de inducción . . . . .	10
2.3. Modelado matemático del motor de inducción . . . . .	13
2.3.1. Marco de referencia . . . . .	13
2.3.2. Ecuaciones del motor de inducción . . . . .	15
2.4. Control y Estimación en el motor de inducción . . . . .	21

2.4.1. Control directo orientado al rotor . . . . .	23
2.4.2. Control indirecto orientado al rotor . . . . .	24
2.5. Estado del arte en estimadores de parámetros usando redes neuronales y sistemas neurodifusos . . . . .	24
<b>3. Redes Neuronales y Sistemas Neurodifusos</b>	<b>29</b>
3.1. Redes Neuronales . . . . .	29
3.1.1. Fundamentos de las Redes Neuronales . . . . .	30
3.1.2. Arquitectura . . . . .	30
3.1.3. Diseño de Redes Neuronales . . . . .	33
3.1.4. Funciones de Activación . . . . .	34
3.1.5. Algoritmos de Entrenamiento . . . . .	36
3.2. Sistemas Neurodifusos . . . . .	37
3.2.1. Fundamentos de los Sistemas Neurodifusos . . . . .	38
3.2.2. Arquitectura . . . . .	39
3.2.3. Algoritmos de Entrenamiento . . . . .	41
<b>4. Estimación de Parámetros</b>	<b>43</b>
4.1. Adquisición de datos para entrenamiento . . . . .	43
4.2. Entrenamiento de Redes y ANFIS . . . . .	45
4.2.1. Redes Neuronales . . . . .	48
4.2.2. ANFIS . . . . .	51
4.3. Resumen del capítulo . . . . .	55
<b>5. Implementación en <i>Vivado HLS</i></b>	<b>57</b>
5.1. Descripción en <i>Síntesis de Alto Nivel</i> . . . . .	57
5.2. Validación en <i>System Generator</i> . . . . .	62
5.3. Resumen del capítulo . . . . .	62
<b>6. Conclusiones y trabajo a futuro</b>	<b>67</b>
6.1. Conclusiones . . . . .	67
6.2. Trabajo a futuro . . . . .	68
<b>Bibliografía</b>	<b>69</b>

# Lista de Figuras

---

2.1. Estator de un motor de inducción donde se observan los devanados [1].	8
2.2. Tipos de rotores de un motor de inducción [1]. . . . .	8
2.3. Modelo equivalente de un MI. . . . .	10
2.4. Circuito equivalente por fase de un motor de inducción. . . . .	12
2.5. Transformación para circuitos estacionarios representados por relaciones trigonométricas [2]. . . . .	15
2.6. Diagrama de bloques de un sistema electromecánico elemental [2]. . .	17
2.7. Transformación de circuitos rotativos representados por relaciones trigonométricas. . . . .	19
2.8. Circuitos equivalentes del marco de referencia arbitrario de un motor de inducción simétrico. . . . .	20
2.9. Generación del par en un motor de inducción. . . . .	22
2.10. Control genérico orientado al flujo del rotor. . . . .	23
2.11. Error en el flujo del rotor entre el controlador y el valor real en el motor causado por un desajuste paramétrico. . . . .	25
2.12. Clasificación de estimación de parámetros [3]. . . . .	26
2.13. Arquitectura de redes neuronales de Ref.[4]. . . . .	27
3.1. Modelo de una neurona [5]. . . . .	31
3.2. Red <i>Feedforward</i> de una sola capa de neuronas. . . . .	32
3.3. Red <i>Feedforward</i> con una capa oculta de neuronas. . . . .	32
3.4. Red recurrente sin lazos de realimentación y sin neuronas ocultas. . .	33
3.5. Red recurrente con neuronas ocultas. . . . .	33
3.6. Tangente Hiperbolica Sigmoidea. . . . .	35
3.7. Logarítmica Sigmoidea. . . . .	35
3.8. Lineal. . . . .	36

3.9. Sistema de Inferencia Difusa. . . . .	39
3.10. Arquitectura ANFIS de dos entradas. . . . .	40
4.1. Diagrama de control vectorial para el motor de inducción implementado en Simulink. . . . .	45
4.2. (a) Variación pseudoaleatoria del par de carga con valores al vacío (0 Nm), media carga (4.5 Nm) y plena carga (9 Nm); (b) Velocidad del Motor con el valor de referencia fijo nominal; (c) Par de carga constante; (d) Variación de la velocidad del motor. . . . .	46
4.3. (a) y (b) Corrientes de los ejes directo y cuadratura sin normalizar y con normalización del caso 1; (c) y (d) Corrientes de los ejes directo y cuadratura sin normalizar y con normalización del caso 2. . . . .	47
4.4. Metodología de entrenamiento. . . . .	48
4.5. Resultado del entrenamiento de las redes neuronales para el caso 1 en la estimación del (a) Flujo del rotor; (b) Error del entrenamiento del Flujo; (c) Posición $\sin(\theta)$ ; (d) Error del entrenamiento de la posición; (e) Velocidad del motor; (f) Error del entrenamiento de la velocidad. . . . .	50
4.6. Resultado del entrenamiento de las redes neuronales para el caso 2 en la estimación del (a) Flujo del rotor; (b) Error del entrenamiento del Flujo; (c) Posición $\sin(\theta)$ ; (d) Error del entrenamiento de la posición; (e) Velocidad del motor; (f) Error del entrenamiento de la velocidad. . . . .	52
4.7. Resultado del entrenamiento de los sistemas ANFIS para el caso 1 en la estimación del (a) Flujo del rotor; (b) Error del entrenamiento del Flujo; (c) Posición $\cos(\theta)$ ; (d) Error del entrenamiento de la posición. . . . .	53
4.8. Resultado del entrenamiento de los sistemas ANFIS para el caso 2 en la estimación del (a) Flujo del rotor; (b) Error del entrenamiento del Flujo; (c) Posición $\sin(\theta)$ ; (d) Error del entrenamiento de la posición. . . . .	54
5.1. Latencia de las redes neuronales del caso 1. . . . .	58
5.2. Latencia de redes neuronales del caso 2. . . . .	59
5.3. Latencia de ANFIS del caso 1. . . . .	60
5.4. Latencia de ANFIS del caso 2. . . . .	61
5.5. Esquemas de las redes neuronales caso 1 para estimación del (a) Flujo del rotor; (b) Posición del flujo del rotor; (c) Velocidad del motor. . . . .	63



---

5.6. Esquemas de los sistemas ANFIS caso 1 para estimación del (a) Flujo del rotor; (b) Posición del flujo del rotor. . . . .	63
5.7. Respuesta de los esquemas de las redes neuronales y los sistemas neurodifusos caso 1 para estimación del (a) Flujo del rotor con redes neuronales; (b) Flujo del rotor con ANFIS; (c) Posición del flujo del rotor con redes neuronales; (d) Posición del flujo del rotor con ANFIS; (e) Velocidad del motor con redes neuronales. . . . .	64
5.8. Esquemas de las redes neuronales caso 2 para estimación del (a) Flujo del rotor; (b) Posición del flujo del rotor; (c) Velocidad del motor. . .	65
5.9. Esquemas de las redes neuronales caso 2 para estimación del (a) Flujo del rotor; (b) Posición del flujo del rotor. . . . .	65
5.10. Respuesta de los esquemas de las redes neuronales y los sistemas neurodifusos del caso 2 para la estimación del (a) Flujo del rotor con redes neuronales; (b) Flujo del rotor con ANFIS; (c) Posición del flujo del rotor con redes neuronales; (d) Posición del flujo del rotor con ANFIS; (e) Velocidad del motor con redes neuronales. . . . .	66



# Lista de Tablas

---

2.1. Marcos de referencia comunmente usados. . . . .	16
3.1. Algoritmos de Entrenamiento. . . . .	37
4.1. Datos del Motor de Inducción. . . . .	43
4.2. Errores de los datos de entrenamiento de las redes neuronales del caso1.	49
4.3. Errores de los datos de entrenamiento de las redes neuronales del caso 2.	51
4.4. Errores del entrenamiento de los sistemas ANFIS del caso 1. . . . .	54
4.5. Errores del entrenamiento de los sistemas ANFIS del caso 2. . . . .	55
5.1. Comparativa del uso de recursos de las redes neuronales del caso 1. . .	58
5.2. Comparativa del uso de recursos de las redes neuronales del caso 2. . .	59
5.3. Comparativa del uso de recursos de los sistemas ANFIS del caso 1. . . .	60
5.4. Comparativa del uso de recursos de los sistemas ANFIS del caso 2. . . .	61



# Nomenclatura

---

$i_{abcs}$	Corrientes del estator Mod. Dinámico
$i_{ds}, i_{qs}$	Corriente del Estator Directo y Cuadratura
$s$	Deslizamiento en fracción de unidad
$\theta$	Desplazamiento Angular
$\theta_{rm}$	Desplazamiento Angular del Rotor
$f_e$	Frecuencia de Alimentación
$f_r$	Frecuencia del Rotor
$\lambda_{abcs}, \lambda_{abcr}$	Flujo de Acoplamiento del Estator y Rotor
$\lambda_{qs}, \lambda_{ds}$	Flujo del Estator de Cuadratura y Directo
$\lambda_{qr}, \lambda_{dr}$	Flujo del Rotor de Cuadratura y Directo
$L_{ls}, L_{lr}$	Inductancia de dispersión del Estator y Rotor
$L_{ms}, L_{mr}$	Inductancia de magnetización del Estator y Rotor
$J$	Inercia del Rotor
$K_s$	Matriz de Transformación
$P$	Número de Polos
$T_e$	Par Electromagnético
$f_{abcs}$	Parámetros a transformar
$P_{conv}$	Potencia Convencional o Mecánica
$r_s, r_r$	Resistencia del Estator y Rotor Mod. Dinámico
$f_{qd0s}$	Transformación <i>directo-cuadratura-cero</i>
$\omega$	Velocidad Angular
$n_{sinc}$	Velocidad de Sincronismo
$n_{des}$	Velocidad de Deslizamiento
$n_m$	Velocidad de Mecánica del eje del motor
$V_{abcs}$	Voltajes del estator Mod. Dinámico



---

## Capítulo 1

# Introducción

---

Parte de la importancia de la energía eléctrica reside en el uso de máquinas que facilitan de alguna manera la vida cotidiana del ser humano, por ejemplo en las zonas residenciales y en la industria. De esta manera, se introduce a la máquina rotativa que representa el mayor consumo de energía eléctrica en los países con mayor desarrollo económico e industrial: el motor eléctrico. Gran parte de los motores eléctricos se basan en el empleo de la corriente alterna y esto ha ido mejorando con el transcurso del tiempo debido a la facilidad de su fabricación, desde pequeñas fracciones hasta grandes dimensiones de potencia, además de que el suministro de energía eléctrica se lleva a cabo por medio de corriente alterna haciendo que de manera inmediata no se requiera algún tipo de convertidor para su utilización.

Existen una gran variedad de máquinas rotativas de corriente alterna, por mencionar algunos están: los motores de inducción (MI), los motores síncronos de rotor devanado y los motores síncronos de imanes permanentes, pero por preferencia está el MI; es superior a los otros debido a que prepondera su robustez, es de fácil mantenimiento, cuenta con fabricación de varios intervalos de potencia y tiene un costo relativamente bajo. Sin embargo, el modelado y el control de este tipo de motores resulta ser complejo comparado con los otros motores.

En el control de los MI, se manejan dos cuestiones principales: la primera es hacer que el controlador y la planta tengan robustez contra las variaciones de los parámetros y las perturbaciones. La segunda cuestión es hacer que el sistema en conjunto responda con cierta *inteligencia* y esto se logra mediante un ajuste automático del mismo frente a los ya mencionados cambios. Uno de los métodos para hacer que el sistema sea inteligente es mediante el empleo de técnicas de cómputo suave (soft computing), usando redes neuronales artificiales, lógica difusa o la combinación de estas dos, dando como resultado a los sistemas neurodifusos [6].

El control del MI que es ampliamente usado en la industria es el llamado control vectorial (CV) o control orientado al campo (FOC *por sus siglas en ingles*), debido a la semejanza de control en los motores de corriente continua. La dificultad de este tipo de control radica en que el modelado del MI se vuelve complejo al presentar multiples variables no lineales, dando como resultado características fuertemente acopladas [7], esto debido a la inducción del estator sobre el rotor. Para el control de los MI es ineludible conocer los parámetros de la máquina, tales como el flujo magnético, la constante de tiempo del rotor, y principalmente el valor de la resistencia del rotor, además de la velocidad y el torque [3]. Cabe mencionar que los parámetros son dependientes de varios factores que influyen en el rendimiento de la propia máquina, por nombrar algunos esta el tipo de operación, la temperatura, la frecuencia de operación y/o la saturación magnética.

Como los motores de inducción presentan características dinámicas no lineales, es necesario hacer uso de técnicas de inteligencia artificial dadas sus propiedades de aproximación universal de funciones no lineales, robustez, operar con información incierta o imprecisa, ruido, además de adaptabilidad, entre otras. Esto puede ser muy útil en problemas importantes tales como la estimación de parámetros y la aplicación efectiva de técnicas de control, porque asimismo se da soporte versado al sistema, funcionando como técnicas adaptivas ante cualquier cambio, mejorando así el rendimiento del controlador del MI. Los estimadores basados en inteligencia artificial se dividen en los siguientes grupos [8]:

- Sistemas basados en lógica difusa.
- Sistemas basados en redes neuronales.
- Sistemas híbridos.
- Sistemas basados en algoritmos evolutivos.

Las redes neuronales, la lógica difusa y los sistemas neurodifusos han sido parte de aplicaciones en los MI como por ejemplo: detección de barras rotas en el rotor, diagnóstico de fallas, control vectorial sin sensado, etc [3, 9, 10]. De manera evidente, este tipo de técnicas adaptivas prometen ser competitivas para diversas tareas tales como la estimación e identificación de parámetros, el procesamiento óptimo de señales, así como la aplicación de diferentes métodos de control, medición y monitoreo de estas,



entre otras [11]. La mayor ventaja de las técnicas basadas en inteligencia artificial es que pueden no requerir de un modelo matemático analítico del motor considerado y el tiempo que desarrolla el estimador y/o controlador puede ser considerablemente disminuido [12].

## 1.1. Planteamiento del problema

En la industria, el MI es la máquina rotativa de corriente alterna más empleada. Muchas de las aplicaciones industriales requieren del control de la posición, la velocidad o el torque; el control desarrollado para este tipo de motores es el control orientado al campo (FOC) [7, 3, 8, 12, 4, 13] y este a su vez está dividido por dos enfoques: el control orientado al campo directo (DFO) y el control orientado al campo indirecto (IFOC).

Sin embargo, el esquema directo es muy limitado debido al problema inherente a bajas velocidades y porque necesita la integración de señales de medición difíciles de obtener, como el flujo en el entrehierro. Por otro lado, el esquema indirecto ha tenido varias aplicaciones industriales en la última década. No obstante, este esquema representa un lazo de realimentación de velocidad, del cual también es necesario conocer la posición y magnitud del flujo del rotor [14].

Dado que el uso de sensores agrava los problemas de costo y fiabilidad, la eliminación de estos es una de la mayor característica en los sistemas de control moderno [15], abriendo camino a la inteligencia artificial. Se ha visto que tiene muchas ventajas en el uso de controladores y estimadores: su diseño no requiere de un modelo de la planta, mejoran el rendimiento, requieren menos esfuerzo en el ajuste y dado a que el motor de inducción presenta un modelo no lineal, ayudando a mejorar el accionamiento del MI; los sistemas inteligentes tienen la propiedad de aproximar cualquier función no lineal, la cual ha sido explotada por varios investigadores [6].

La implementación de estos sistemas en hardware, como es el caso de los FPGAs, permite verificar su funcionamiento y aprovechar el paralelismo que brinda el hardware, además de que permite la aplicación en tiempo real.

## 1.2. Justificación

La mejora en las técnicas de control que hacen más eficiente la operación de máquinas como el MI, requieren de diversas etapas para lograr este objetivo. Una de ellas, es disponer de métodos para la adecuada estimación de parámetros obteniendo la mejor aproximación al comportamiento complejo de estas. Técnicas modernas basadas en inteligencia computacional proveen la ventaja de no requerir el uso de una gran cantidad de sensores, además de ser adaptables y tolerantes a imprecisión, incertidumbre, entre otras. Debido a que se vuelve complicado medir con sensores el flujo del rotor y la velocidad del rotor (por el acceso al interior del motor y al costo elevado), la estimación de su magnitud y posición es una alternativa en el esquema indirecto, volviéndose indispensable para el control. Estas técnicas sirven de tal manera que no es necesario la aplicación de un modelo que explique el comportamiento del motor. La implementación de tales técnicas en hardware, específicamente en una plataforma FPGA, se espera que haya una mejora en el rendimiento del controlador, debido al paralelismo del hardware y la reconfigurabilidad.

## 1.3. Objetivo

Aplicar técnicas de inteligencia computacional para la estimación de parámetros y control de un motor de inducción en plataformas de procesamiento en tiempo real, como lo es un FPGA. Se pretende la realización de la estimación de los parámetros en condiciones cambiantes considerando las conocidas por variaciones del par de carga y la velocidad.

### 1.3.1. Objetivos específicos

- Estimación de las variables necesarias para el esquema de control, con las técnicas de inteligencia computacional basadas en redes neuronales artificiales y sistemas neurodifusos.
- Implementar estas técnicas de inteligencia computacional en un esquema de co-simulación para su futura implementación en hardware mediante FPGAs.
- Comprobar la estimación de parámetros del motor en condiciones variables para

futuras implementaciones de esquemas de control

## 1.4. Organización de la tesis

En el capítulo 2 se presenta una breve descripción del MI. Como esta construido y como es su funcionamiento. Se describe el modelado matemático analítico del motor de inducción basado en la conversión del marco de referencia *directo y de cuadratura*. También se explica el control y la estimación de parámetros en este tipo de motores. Finalmente se presenta una breve reseña del estado del arte en estimadores de parámetros de motores de inducción.

En el capítulo 3 se presenta una breve introducción de las redes neuronales artificiales y los sistemas neurodifusos, los fundamentos que presenta cada tipo de inteligencia artificial, la arquitectura funcional que pueden tener, su diseño y los diferentes algoritmos de entrenamiento, dando como resultado el aprendizaje supervisado desarrollado en esta tesis.

En el capítulo 4 se describe la metodología para obtener las arquitecturas de las redes neuronales y los sistemas neurodifusos que estimaran los parámetros del MI. Tales parámetros son el flujo del rotor, la posición del flujo del rotor y la velocidad del motor en dos casos diferentes. El primer caso es donde se mantiene la velocidad de referencia a un valor nominal constante y únicamente se varia el par de carga. En el segundo caso, se mantiene constante el par de carga y se varia la velocidad de referencia. Se describe la adquisición de datos para el entrenamiento de las redes neuronales y los sistemas ANFIS. La configuración de las redes se llevaron a cabo por medio de un razonamiento de tipo heurístico, observando resultados de los factores de correlación, el error medio absoluto, el error medio cuadrático y el porcentaje de error medio relativo para elegir la mejor arquitectura.

En el capítulo 5 se presenta el resultado de las descripciones de las redes neuronales y los sistemas neurodifusos en Síntesis de Alto Nivel, importante para el desarrollo de algoritmos basados en FPGAs, la herramienta utilizada para esta descripción fue *Vivado-HLS*. La validación de las redes se llevo a cabo mediante la simulación usando la herramienta *System Generator*. Ambas herramientas pertenecen a **Xilinx**.

Finalmente, en el capítulo 6 se presentan las conclusiones respecto a los resultados obtenidos y los posibles trabajos a futuro.



# Motor de Inducción

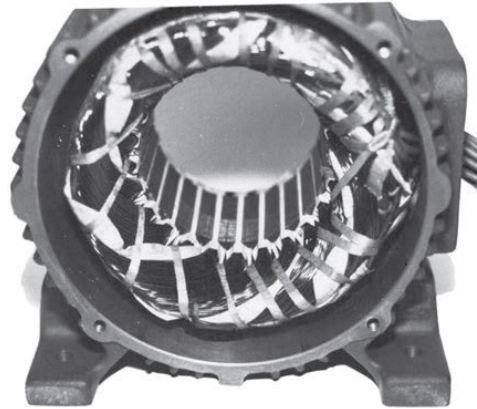
---

El MI es una de las máquinas más empleadas debido a la extensa variedad de aplicaciones industriales, con el fin de accionar cualquier tipo de carga mecánica. Es sin lugar a duda, el caballo de batalla en la industria eléctrica. Algunas de sus aplicaciones reside en: ascensores, bombas centrífugas, bandas transportadoras, trituradoras, ventiladores, compresoras, bombas de combustible, etc. Por tanto, su uso generalizado se debe a que representa un bajo costo, es de bajo mantenimiento, fácil de adquirir, tiene alto grado de protección, su funcionamiento se basa en pocos componentes y presenta una gran robustez.

## 2.1. Construcción del motor de inducción

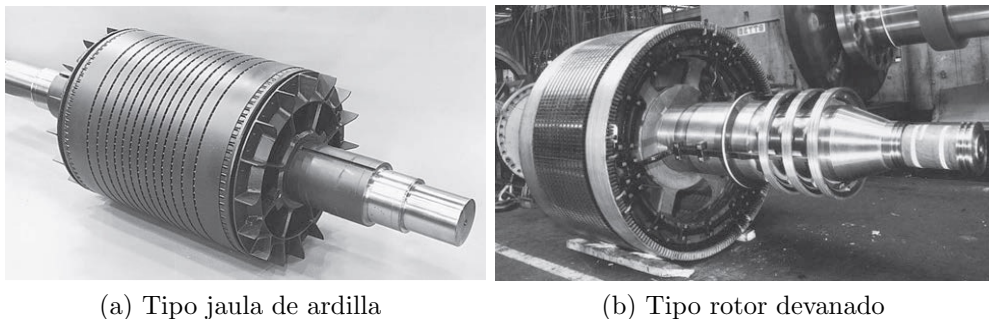
Para conocer el funcionamiento del MI, es necesario comprender la arquitectura que presenta esta máquina de corriente alterna. Los componentes fundamentales son el estator y el rotor. El estator es la parte estacionaria del motor y está constituido de laminaciones delgadas, ranuradas, de acero de alta permeabilidad, dentro de un bastidor o carcasa de acero o hierro de fundición [16]. En las ranuras es donde se colocan las bobinas para formar devanados monofásicos o polifásicos. En la Figura 2.1 se muestra el estator de un motor de inducción, en el cual se pueden apreciar los devanados del estator.

Por otra parte, el rotor está compuesto de laminaciones delgadas, ranuradas, de acero sumamente permeable que se fijan sobre un eje. El rotor de un MI puede ser de dos tipos: de jaula de ardilla y de rotor devanado. El rotor jaula de ardilla consta de barras conductoras insertadas en las ranuras y que están conectadas en corto circuito en los extremos, mediante anillos. Por lo general, este tipo de rotor, se utiliza



**Figura 2.1:** Estator de un motor de inducción donde se observan los devanados [1].

cuando la carga requiere de un par de arranque reducido [16]. El rotor devanado se construye con un devanado similar al del estator [17], fabricado con el mismo número de polos que el estator. Sus terminales van conectadas a anillos colectores aislados que están montados sobre el eje, haciendo que la corriente del rotor sea accesible. El rotor devanado se emplea cuando la carga requiere un par de arranque alto. Sin embargo, el rotor jaula de ardilla es el más empleado por la simplicidad y robustez en su construcción, de lo contrario, el de rotor devanado requiere de más mantenimiento debido al desgaste de los anillos colectores, por ello rara vez se utilizan. En la Figura 2.2 se muestra los dos tipos de rotores para el motor de inducción.



(a) Tipo jaula de ardilla

(b) Tipo rotor devanado

**Figura 2.2:** Tipos de rotores de un motor de inducción [1].

## 2.2. Funcionamiento del motor de inducción

El nombre del MI se debe a la forma en como trabaja. El voltaje aplicado a las terminales del estator es *inducido* al rotor, necesario para producir un par de torsión (inducción electromagnética), por ello, su semejanza al funcionamiento de un transformador. El principio de inducción esta relacionado con la Ley de Faraday. Cuando se aplica un voltaje trifásico balanceado en las terminales del estator, comenzara a fluir un conjunto trifásico de corrientes. Estas corrientes generadas produciran un campo magnético  $B_S$  que girara a cierta velocidad. La velocidad de rotación de este campo se conoce como velocidad de sincronismo, dada por la ecuación:

$$n_{sinc} = \frac{120f_e}{P} \quad (2.2.1)$$

donde  $f_e$  es la frecuencia del sistema de alimentación eléctrico en Herzt,  $P$  es el número de polos de la máquina. Es así cuando el campo magnético pasa a traves del entrehierro hacia las barras del rotor y se induce un voltaje en ellas. Este voltaje inducido en las barras del rotor, producira al igual corrientes que a su vez produciran un campo magnético relativo al campo magnético estatorico dando como resultado un par de fuerzas (Ley de Lorenz) haciendo que el rotor gire.

Sin embargo, la velocidad con la que gira el rotor no es la misma que la velocidad de sincronismo. Esto se debe a que si el rotor girara a la misma velocidad síncrona, las barras del rotor estarán estacionarias con respecto al campo magnético, y no habría un voltaje inducido, por lo tanto, no exisitiria corriente en el rotor, ni campo magnético que genere el par de fuerzas para su movimiento. Asi que el rotor tiene un limite superior finito.

El voltaje que se induce en las barras del rotor, va a depender de la velocidad del rotor en relación a los campos magnéticos que se generan en el estator [1], por ello se menciona una velocidad relativa, llamada comunmente *velocidad de deslizamiento*. Se define como la diferencia entre la velocidad síncrona y la velocidad del rotor:

$$n_{des} = n_{sinc} - n_m \quad (2.2.2)$$

donde  $n_{des}$  es la velocidad de deslizamiento,  $n_{sin}$  es la velocidad de sincronismo y  $n_m$  es la velocidad mecánica del eje del motor. Mayormente, el deslizamiento de una máquina, que es otro término para describir el movimiento relativo, esta dado en una

fracción de unidad o de porcentaje:

$$s = \frac{n_{des}}{n_{sinc}} \times 100 \quad (2.2.3)$$

$$s = \frac{n_{sinc} - n_m}{n_{sinc}} \times 100 \quad (2.2.4)$$

También se puede expresar la velocidad mecánica del eje del motor en términos de la velocidad síncrona y de deslizamiento:

$$n_m = (1 - s) \times n_{sinc} \quad (2.2.5)$$

Esta ecuación es muy útil para deducir el par del MI y las relaciones de potencia.

### 2.2.1. Modelo estático de un motor de inducción

Como ya se menciona, los MI dependen de la inducción electromagnética al circuito del rotor desde el circuito del estator. A partir de esta descripción se puede deducir un circuito equivalente por medio del circuito equivalente de un transformador. En la Figura 2.3 se muestra un circuito equivalente por fase que representa la operación de un motor de inducción en estado estable.

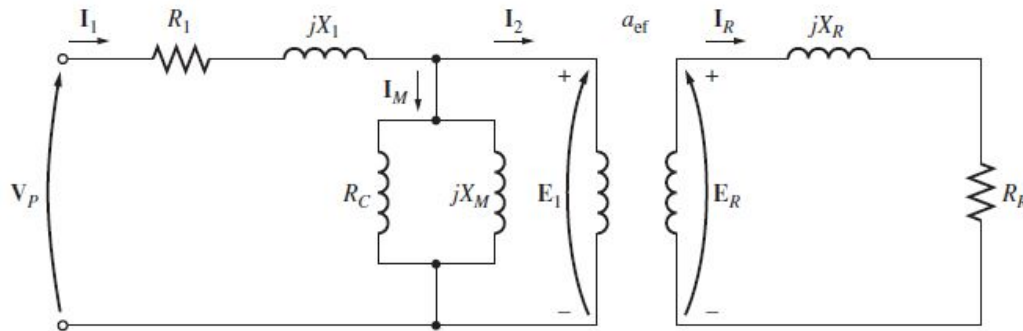


Figura 2.3: Modelo equivalente de un MI.

En el circuito se puede apreciar un voltaje aplicado  $V_p$  a la terminal del estator, existe cierta resistencia y una reactancia de dispersión en la entrada del modelo llamadas  $R_1$  y  $X_1$  respectivamente. Debido al entrehierro que existe en los MI, se provocara una reluctancia, y entre más grande sea esta reluctancia, mayor será la corriente de magnetización para obtener un cierto nivel de flujo magnético. La reactancia de magnetización  $X_M$  es mucho menor de lo que es un transformador ordinario. El voltaje  $E_1$



del estator se acopla al voltaje  $E_R$  por medio de una relación efectiva. El voltaje  $E_R$  que se produce en el rotor genera un flujo de corriente en cortocircuito de la máquina. Al igual que en el circuito del estator, el circuito del rotor presenta una resistencia y una reactancia de dispersión  $R_R$  y  $X_R$  respectivamente, sin embargo, esta impedancia se vera afectada por los efectos de variación de la frecuencia del rotor, la cual se puede expresar como:

$$f_r = sf_e \quad (2.2.6)$$

Mientras mayor sea el movimiento relativo entre los campos magnéticos del rotor y del estator, mayor será el voltaje resultante del rotor y la frecuencia del rotor. Un ejemplo a simple vista es cuando el rotor esta inmovil o en condición de *rotor bloqueado*, se induce el mayor voltaje y frecuencia en el rotor. Por otra parte, si no existiera movimiento relativo, es decir, que el rotor se mueva a la misma velocidad que el campo magnético del estator, el voltaje inducido será de 0 V y de frecuencia 0 Hz. Es así, que se define que la magnitud del voltaje y frecuencia inducidos en el rotor es directamente proporcional a su deslizamiento:

$$E_R = sE_{R0} \quad (2.2.7)$$

Como se observa en la Figura 2.3, la resistencia del rotor se mantiene constante, independiente del deslizamiento, mientras que la reactancia del rotor se ve afectada por el deslizamiento. La reactancia del rotor de un motor de inducción depende de la inductancia y de la frecuencia del rotor. Si se tiene una inductancia  $L_R$  entonces la reactancia viene dada por:

$$\begin{aligned} X_R &= 2\pi sf_e L_R \\ &= s(2\pi f_e L_R) \\ &= sX_{R0} \end{aligned} \quad (2.2.8)$$

La corriente resultante del rotor viene dada por:

$$I_R = \frac{E_R}{R_R + jsX_{R0}} \quad (2.2.9)$$

$$I_R = \frac{E_{R0}}{\frac{R_R}{s} + jX_{R0}} \quad (2.2.10)$$

Hasta ahora, la parte del rotor no ha sido referida al estator para obtener un circuito equivalente final. Al referir el rotor al lado del estator, se tiene que considerar los efectos de variación de velocidad concentrados en la impedancia del rotor como se observa en la Figura 2.4.

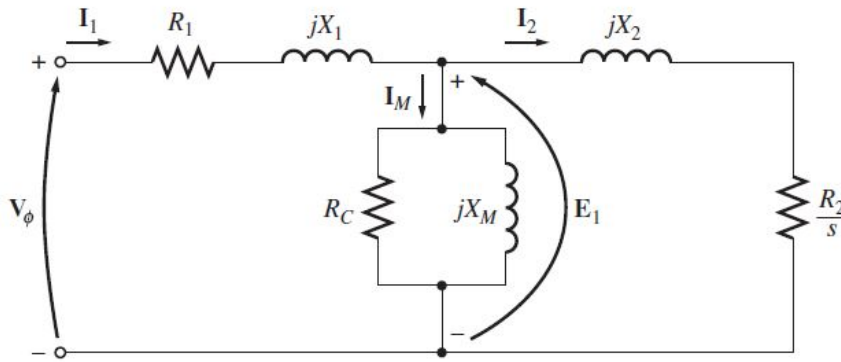


Figura 2.4: Circuito equivalente por fase de un motor de inducción.

Ya que los MI son de una sola excitación, la relación de potencia y par, viene dada por la potencia eléctrica en la entrada y potencia mecánica en la salida. Esta relación es a través de pérdidas que se presentan en el MI. Las primeras pérdidas en la máquina son las del cobre en el estator  $P_{PCE}$ . Luego viene la pérdida por histéresis y corrientes parásitas o también llamadas pérdidas en el núcleo  $P_{nucleo}$ . La potencia que resta, se transfiere al rotor a través del entrehierro  $P_{EH}$ . Esta potencia transferida se ve reducida por pérdidas en el cobre del rotor  $P_{PCR}$ , la resultante de potencia en este punto ya es mecánica  $P_{conv}$ . Sin embargo, se resta potencia en la salida por pérdidas de fricción y rozamiento  $P_{FyR}$  y pérdidas misceláneas  $P_{misc}$ . De esta manera, la potencia restante es la salida del motor  $P_{sal}$ .

El par inducido o par desarrollado por la máquina  $\tau_{ind}$  se define como el par generado por la conversión de potencia interna de energía eléctrica a mecánica. Este par está dado por:

$$\tau_{ind} = \frac{P_{conv}}{\omega_m} \quad (2.2.11)$$

## 2.3. Modelado matemático del motor de inducción

Una vez explicado el funcionamiento del MI, se plantea el modelado matemático. Debido al acoplamiento magnético entre el estator y el rotor, el modelado de su comportamiento dinámico en tres ejes es posible explicar con ecuaciones diferenciales de coeficientes variables que cambian con el tiempo dando una estructura del modelo compleja. Por esta razón, se presenta el modelado en un sistema de dos ejes llamado  $d-q$  [18], facilitando el modelado con un cambio de variable.

### 2.3.1. Marco de referencia

Existen varios cambios de variable que son utilizadas. Estas transformaciones en general refieren las variables de las máquinas a un marco de referencia rotativo a una velocidad angular arbitraria. El cambio de sistema a dos ejes es un enfoque que se introdujo en la década de 1920 por R.H. Park, donde se formula un cambio de variables (voltajes, corrientes, flujos de acoplamiento) asociados a los devanados del estator de una máquina síncrona con variables asociadas a devanados ficticios en rotación [2]. En otras palabras él transformó las variables del estator a un marco de referencia fijo del rotor, comúnmente llamada transformación Park.

Todas las transformaciones se pueden obtener asignando apropiadamente una velocidad de rotación, la cual puede ser cero, llamado comúnmente como *marco de referencia arbitrario*. A pesar de que el cambio de variables usado en el análisis de las máquinas de C.A. para eliminar las inductancias variables en el tiempo, el cambio de variables puede ser utilizado en sistemas eléctricos de potencia de parámetros constantes, sistemas de control asociados a accionamientos eléctricos, etc.

Un cambio de variable que formula la transformación de un sistema trifásico de un circuito estacionario a un marco de referencia arbitrario puede ser expresado como:

$$f_{qd0s} = K_s f_{abcs} \quad (2.3.1)$$

donde

$$(f_{qd0s})^T = [f_{qs} f_{ds} f_{0s}] \quad (2.3.2)$$

$$(f_{abc})^T = [f_{as} f_{bs} f_{cs}] \quad (2.3.3)$$

$$K_s = \frac{2}{3} \begin{bmatrix} \cos \theta & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ \sin \theta & \sin(\theta - \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (2.3.4)$$

$$\omega = \frac{d\theta}{dt} \quad (2.3.5)$$

Para la transformación inversa, se tiene la siguiente ecuación:

$$(K_s)^{-1} = \begin{bmatrix} \cos \theta & \sin \theta & 1 \\ \cos(\theta - \frac{2\pi}{3}) & \sin(\theta - \frac{2\pi}{3}) & 1 \\ \cos(\theta + \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) & 1 \end{bmatrix} \quad (2.3.6)$$

La velocidad angular  $\omega$ , y el desplazamiento angular  $\theta$ , de un marco de referencia arbitrario están relacionados por:

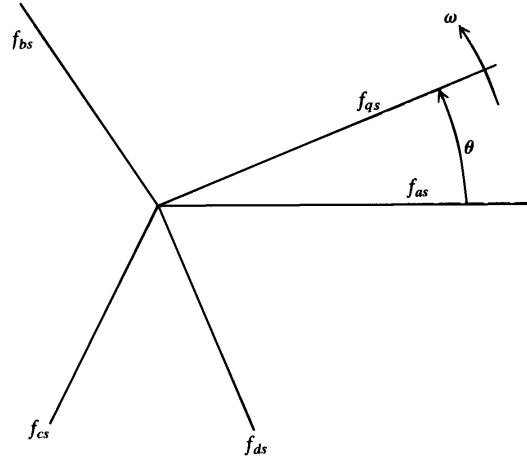
$$\theta = \int \omega dt \quad (2.3.7)$$

o por una integral definida:

$$\theta = \int_0^t \omega(\xi) d\xi + \theta(0) \quad (2.3.8)$$

En las ecuaciones anteriores, la letra  $f$  puede representar cualquier parámetro eléctrico, ya sea voltaje, corriente, flujo de acoplamiento o carga eléctrica. El superíndice  $T$  denota la transpuesta de una matriz. El subíndice  $s$  indica las variables, parámetros y transformación asociadas con el circuito estacionario. Es importante mencionar que, el marco de referencia puede rotar a cualquier constante o velocidad angular variable o permanecer estacionario. En particular, esta transformación es conocida como *transformación directo-cuadratura-cero*, como se aprecia en la Figura 2.5.

Como se ha mencionado, existen varios tipos de marcos de referencia. Los comúnmente usados en el análisis de máquinas eléctricas y componentes de sistemas de potencia se llaman: marco de referencia arbitrario, estacionario, rotor y síncrono. En la Tabla 2.1 se puede ver su aplicación y notación.



**Figura 2.5:** Transformación para circuitos estacionarios representados por relaciones trigonométricas [2].

Los marcos de referencia de rotación síncrona o de sincronismo son marcos que corresponden a una velocidad eléctrica angular de una frecuencia fundamental. La transformación de variables que se asocian a un marco de referencia estacionario fue desarrollado por E. Clarke, el cual, utilizó la notación  $f_\alpha$  y  $f_\beta$ . En la transformación Park, el marco de referencia esta en rotación.

### 2.3.2. Ecuaciones del motor de inducción

Se establecen ecuaciones de voltaje y par de un MI simétrico expresadas en variables de la máquina. Las ecuaciones pueden expresarse en cualquier marco de referencia asignando una velocidad apropiada en el marco de referencia arbitrario. Los marcos de referencia mas usados son el marco de referencia fijo en el rotor y el marco de referencia de rotación síncrona.

Para comenzar, se supone un MI con un arreglo trifásico de dos polos, conectado en estrella. Los devanados del estator son identicos, sinusoidalmente distribuidos  $120^\circ$ , con una resistencia del estator y del rotor  $r_s$  y  $r_r$  respectivamente. Las ecuaciones de voltaje en variables de la máquina son:

$$v_{abcs} = r_s i_{abcs} + p \lambda_{abcs} \quad (2.3.9)$$

Tabla 2.1: Marcos de referencia comunmente usados.

Marco de Referencia	Interpretación	Notación	
		Variables	Transformación
$\omega$	VARIABLES DEL CIRCUITO ESTACIONARIO referidos a un marco de ref arbitrario	$f_{qs}, f_{ds}, f_{0s}$	$K_s$
0	VARIABLES DEL CIRCUITO ESTACIONARIO referidos a un marco de ref estacionario	$f_{qs}^s, f_{ds}^s, f_{0s}^s$	$K_s^s$
$\omega_r$	VARIABLES DEL CIRCUITO ESTACIONARIO referidos a un marco de ref fijo en el rotor	$f_{qs}^r, f_{ds}^r, f_{0s}^r$	$K_s^r$
$\omega_s$	VARIABLES DEL CIRCUITO ESTACIONARIO referidos a un marco de ref de rotación síncrona	$f_{qs}^e, f_{ds}^e, f_{0s}^e$	$K_s^e$

$$v_{abcr} = r_r i_{abcr} + p \lambda_{abcr} \quad (2.3.10)$$

En las ecuaciones anteriores, el subíndice  $s$  esta asociada a las variables que pertenecen al circuito del estator, y el subíndice  $r$  a las variables que perteneces al circuito del rotor. Los valores  $r_s$  y  $r_r$  son matrices diagonales. Para un sistema lineal magnético, los flujos de acoplamiento se expresan como:

$$\begin{bmatrix} \lambda_{abcs} \\ \lambda_{abcr} \end{bmatrix} = \begin{bmatrix} L_s & L_{sr} \\ (L_{sr})^T & L_r \end{bmatrix} \begin{bmatrix} i_{abcs} \\ i_{abcr} \end{bmatrix} \quad (2.3.11)$$

Las inductancias de cada devanado son las siguientes:

$$L_s = \begin{bmatrix} L_{ls} + L_{ms} & \frac{-1}{2} L_{ms} & \frac{-1}{2} L_{ms} \\ \frac{-1}{2} L_{ms} & L_{ls} + L_{ms} & \frac{-1}{2} L_{ms} \\ \frac{-1}{2} L_{ms} & \frac{-1}{2} L_{ms} & L_{ls} + L_{ms} \end{bmatrix} \quad (2.3.12)$$

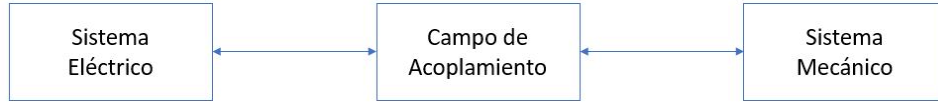
$$L_r = \begin{bmatrix} L_{lr} + L_{mr} & \frac{-1}{2} L_{mr} & \frac{-1}{2} L_{mr} \\ \frac{-1}{2} L_{mr} & L_{lr} + L_{mr} & \frac{-1}{2} L_{mr} \\ \frac{-1}{2} L_{mr} & \frac{-1}{2} L_{mr} & L_{lr} + L_{mr} \end{bmatrix} \quad (2.3.13)$$

$$L_{sr} = \begin{bmatrix} \cos \theta_r & \cos (\theta_r + \frac{2\pi}{3}) & \cos (\theta_r - \frac{2\pi}{3}) \\ \cos (\theta_r - \frac{2\pi}{3}) & \cos \theta_r & \cos (\theta_r + \frac{2\pi}{3}) \\ \cos (\theta_r + \frac{2\pi}{3}) & \cos (\theta_r - \frac{2\pi}{3}) & \cos \theta_r \end{bmatrix} \quad (2.3.14)$$

En las ecuaciones anteriores,  $L_{ls}$  y  $L_{ms}$  son las inductancias de dispersión y de magnetización de los devanados del estator respectivamente;  $L_{lr}$  y  $L_{mr}$  para los devanados del rotor. La inductancia mutua entre el estator y el rotor es  $L_{sr}$ . Cabe mencionar, que las variables del rotor deben ser referidas a las variables del estator, con una apropiada razón de cambio. Como las inductancias de magnetización y las inductancias mutuas están relacionados por la misma trayectoria de flujo magnético. La ecuación de voltaje referida al estator queda como:

$$\begin{bmatrix} v_{abcs} \\ v'_{abcr} \end{bmatrix} = \begin{bmatrix} r_s + pL_s & pL'_{sr} \\ p(L'_{sr})^T & r'_r + pL'_r \end{bmatrix} \begin{bmatrix} i_{abcs} \\ i'_{abcr} \end{bmatrix} \quad (2.3.15)$$

Para obtener la ecuación de par, es necesario comprender la energía almacenada en el campo de acoplamiento, que comprende la conversión de energía eléctrica a energía mecánica, como se observa en la figura 2.6.



**Figura 2.6:** Diagrama de bloques de un sistema electromecánico elemental [2].

El cambio de la energía mecánica en un sistema rotativo con una entrada mecánica se puede describir de la siguiente manera:

$$dW_m = -T_e d\theta_{rm} \quad (2.3.16)$$

donde  $T_e$  es el par electromagnético de valor positivo para el MI (el par es salida en el sistema) y  $\theta_{rm}$  es el desplazamiento angular del rotor. El desplazamiento angular eléctrico está en función del número de polos de la máquina:

$$\theta_r = \left(\frac{P}{2}\right)\theta_{rm} \quad (2.3.17)$$

por lo tanto,

$$dW_m = -T_e \left( \frac{P}{2} \right) d\theta_r \quad (2.3.18)$$

Haciendo ajustes matemáticos, se puede encontrar que existe una relación entre la velocidad del rotor y el par electromagnético:

$$T_e = J \left( \frac{2}{P} \right) p\omega_r + T_L \quad (2.3.19)$$

donde  $J$  es la inercia del rotor, recordando que el eje puede tener conectada carga, la cual también presenta inercia. Una vez que analizamos las ecuaciones del motor de inducción, es deseable transformar todas las variables asociadas a un marco de referencia arbitrario, este cambio de variables viene dado por:

$$f'_{qd0r} = K_r f'_{abcr} \quad (2.3.20)$$

donde

$$(f'_{qd0r})^T = \begin{bmatrix} f'_{qr} & f'_{dr} & f'_{0r} \end{bmatrix} \quad (2.3.21)$$

$$(f'_{abcr})^T = \begin{bmatrix} f'_{ar} & f'_{br} & f'_{cr} \end{bmatrix} \quad (2.3.22)$$

$$K_r = \frac{2}{3} \begin{bmatrix} \cos \beta & \cos \left( \beta - \frac{2\pi}{3} \right) & \cos \left( \beta + \frac{2\pi}{3} \right) \\ \sin \beta & \sin \left( \beta - \frac{2\pi}{3} \right) & \sin \left( \beta + \frac{2\pi}{3} \right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (2.3.23)$$

$$\beta = \theta - \theta_r \quad (2.3.24)$$

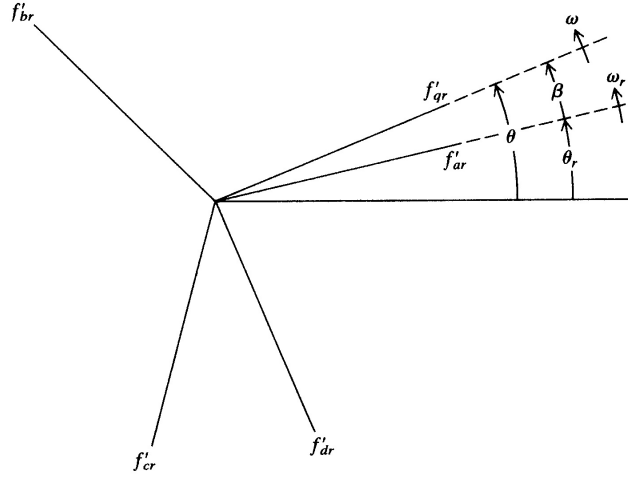
El subíndice  $r$  en las variables, parámetros y transformaciones indica una asociación con los circuitos giratorios. Es más fácil visualizar esta transformación mediante una relación trigonométrica como en la Figura 2.7.

Así, las ecuaciones de voltaje en el marco de referencia arbitrario, sin un análisis más estricto quedan como:

$$v_{qd0s} = r_s i_{qd0s} + \omega \lambda_{dqs} + p \lambda_{qd0s} \quad (2.3.25)$$

$$v_{qd0r} = r_r i_{qd0r} + (\omega - \omega_r) \lambda'_{dqr} + p \lambda'_{qd0r} \quad (2.3.26)$$





**Figura 2.7:** Transformación de circuitos rotativos representados por relaciones trigonométricas.

Con frecuencia, las ecuaciones de voltaje se representan en forma expandida, obteniendo:

$$v_{qs} = r_s i_{qs} + \omega \lambda_{ds} + p \lambda_{qs} \quad (2.3.27)$$

$$v_{ds} = r_s i_{ds} + \omega \lambda_{qs} + p \lambda_{ds} \quad (2.3.28)$$

$$v_{0s} = r_s i_{0s} + p \lambda_{0s} \quad (2.3.29)$$

$$v'_{qr} = r'_r i'_{qr} + (\omega - \omega_r) \lambda'_{dr} + p \lambda'_{qr} \quad (2.3.30)$$

$$v'_{dr} = r'_r i'_{dr} + (\omega - \omega_r) \lambda'_{qr} + p \lambda'_{dr} \quad (2.3.31)$$

$$v'_{0r} = r'_r i'_{0r} + p \lambda'_{0r} \quad (2.3.32)$$

Los flujos de acoplamiento en forma expandida quedan de la forma:

$$\lambda_{qs} = L_{ls} i_{qs} + L_M (i_{qs} + i'_{qr}) \quad (2.3.33)$$

$$\lambda_{ds} = L_{ls} i_{ds} + L_M (i_{ds} + i'_{dr}) \quad (2.3.34)$$

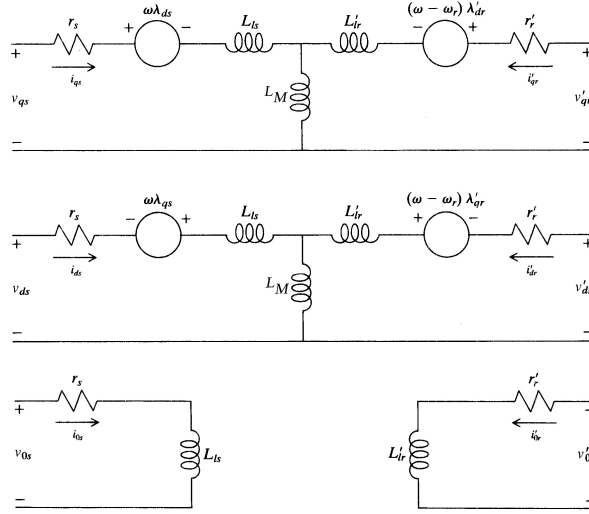
$$\lambda_{0s} = L_{ls} i_{0s} \quad (2.3.35)$$

$$\lambda'_{qr} = L'_{lr} i'_{qr} + L_M (i_{qs} + i'_{qr}) \quad (2.3.36)$$

$$\lambda'_{dr} = L'_{lr}i'_{dr} + L_M(i_{ds} + i'_{dr}) \quad (2.3.37)$$

$$\lambda'_{0r} = L'_{lr}i'_{0r} \quad (2.3.38)$$

De esta manera, las ecuaciones de voltaje y los flujos de acoplamiento se pueden observar en los circuitos equivalentes de la Figura 2.8



**Figura 2.8:** Circuitos equivalentes del marco de referencia arbitrario de un motor de inducción simétrico.

De esta manera, quedan expresadas las ecuaciones de voltaje en términos de corrientes y flujos de acoplamiento en el marco de referencia arbitrario. Las ecuaciones de voltaje están expresadas en los ejes *directo* y de *cuadratura*. Esta propiedad de conversión de 3 ejes a 2 ejes y que los flujos de acoplamiento pueden quedar como variables independientes, hace que se vuelva más fácil de implementar en simulación por computadora un MI, con los flujos de acoplamiento como variables de estado en lugar de las corrientes.

La ecuación del par en el marco de referencia arbitrario se obtiene al sustituir las ecuaciones de transformación, así el par expresado en términos de corrientes:

$$T_e = \left(\frac{3}{2}\right)\left(\frac{P}{2}\right)L_M(i_{qs}i'_{dr} - i_{ds}i'_{qr}) \quad (2.3.39)$$

Algunas otras expresiones del par electromagnético equivalentes son:

$$T_e = \left(\frac{3}{2}\right)\left(\frac{P}{2}\right)(\lambda'_{qr}i'_{dr} - \lambda'_{dr}i'_{qr}) \quad (2.3.40)$$

$$T_e = \left(\frac{3}{2}\right)\left(\frac{P}{2}\right)(\lambda_{ds}i_{qs} - \lambda_{qs}i_{ds}) \quad (2.3.41)$$

Es así, que el comportamiento del motor de inducción se puede describir en cualquier marco de referencia, los más empleados son el marco de referencia estacionario, el marco de referencia fijo al rotor y el marco de referencia de rotación síncrona. Generalmente, las condiciones de operación del motor son las que dictaminan que marco de referencia es el más conveniente para el análisis y/o simulación [2].

## 2.4. Control y Estimación en el motor de inducción

El control de los MI es un tema muy relevante, debido a que varios procesos industriales requieren de la variación de la velocidad. Existe una gran diversidad de estrategias en la industria [2]: el control por Volts-Hertz, donde se aplica cierta magnitud de frecuencia y voltaje a un inversor logrando una velocidad deseada; el control por deslizamiento constante donde se tiene una señal de entrada de par deseado y por lo tanto se requiere de un lazo de retroalimentación y un sensor de velocidad, limitando la rapidez de la respuesta en el sistema. La última estrategia de control a considerar es el control de campo orientado o control vectorial. A pesar de ser la estrategia de control más usada en la industria, cuenta con algunas desventajas. En la versión directa, las mediciones de flujo requieren de sensores, volviendo el sistema costoso; en su versión indirecta el sistema se vuelve sensible a variaciones paramétricas, a menos que se tenga una identificación de parámetros en línea.

El control y la estimación se vuelven más complejos en las máquinas de CA que en las máquinas de CD debido a que se requiere de un alto rendimiento, además de una frecuencia variable, fuentes de alimentación de convertidor, la propia dinámica de las máquinas de CA, las variaciones paramétricas y el procesar señales de retroalimentación con presencia de armónicos [19].

Existen varias permutaciones del control vectorial pero los que mayormente prevalecen son el control directo orientado al flujo del rotor y el control indirecto orientado al campo del rotor. El funcionamiento del control vectorial es una analogía del control de un motor DC excitado por separado. Se puede explicar mediante la ecuación de fuerza de Lorentz. Si se considera un lazo de corriente en un campo con flujo uniforme, el par desarrollado estará dado por:

$$T_e = -2BiNLr \sin \theta \quad (2.4.1)$$

Donde  $B$  es la densidad del flujo,  $i$  es la corriente,  $N$  es el número de vueltas de la bobina,  $L$  es la longitud de la bobina y  $r$  es el radio de la bobina. La magnitud del par será máxima cuando el vector de corriente se define perpendicular a la superficie del devanado. El mismo principio se aplica para los motores de inducción. Considera la Figura 2.9, donde se observan las corrientes del rotor del eje directo y de cuadratura, los vectores de flujo de acoplamiento en algún instante de tiempo.

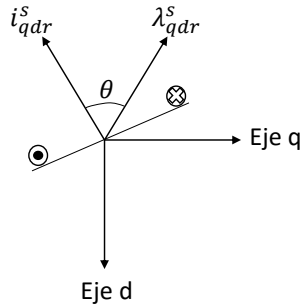


Figura 2.9: Generación del par en un motor de inducción.

Obteniendo el siguiente par desarrollado:

El par se maximiza cuando los vectores de flujo y de corriente se encuentran ortogonalmente. Por tanto, es deseable mantener el vector de flujo del rotor perpendicular al vector de corriente del rotor. Para lograr la perpendicularidad, en la implementación directa e indirecta, se requiere de dos condiciones, la primera es asegurar que:

$$\lambda_{qr} = 0 \quad (2.4.2)$$

Y la segunda es asegurar que:

$$i_{dr} = 0 \quad (2.4.3)$$

Para lograr que la corriente del rotor directa sea cero, se tiene que forzar a que la corriente del estator directa se mantenga constante. Es así, que el par desarrollado vendrá dado por:

$$T_e = \frac{3}{2} \frac{P}{2} \frac{L_M}{L_{rr}} \lambda_{dr} i_{qs} \quad (2.4.4)$$

La ecuación de par anterior sugieren un control genérico orientado al flujo del rotor. Existen variables de comando, de medición y de observación. Como se ilustra en la Figura 2.10, una fuente de DC alimenta a un inversor para el accionamiento del motor de inducción. Sin embargo, este control genérico no está completo por dos detalles importantes: la determinación del flujo del rotor y la determinación de la posición. La diferencia de la implementación entre el esquema directo y el indirecto radica en cómo estas dos variables se establecen.

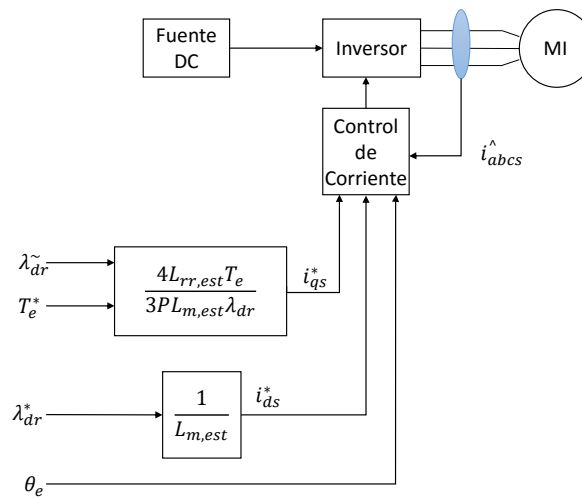


Figura 2.10: Control genérico orientado al flujo del rotor.

### 2.4.1. Control directo orientado al rotor

En este esquema llamado directo o método de retroalimentación (inventado por Blaschke) se tiene que definir la posición del marco de referencia síncrono. La dificultad de este enfoque es que los flujos de cuadratura y directo del rotor son cantidades de medición directa. Se pueden estimar usando mediciones directas en el flujo del entrehierro, utilizando sensores de efecto Hall. Este método se basa en la obtención de la magnitud y posición de flujo del vector de flujo, partiendo de la medición de los voltajes y corrientes del estator. Una desventaja es el uso de integradores, ya que funcionan de forma adecuada en velocidades altas pero a bajas velocidades, los integradores generan errores.

### 2.4.2. Control indirecto orientado al rotor

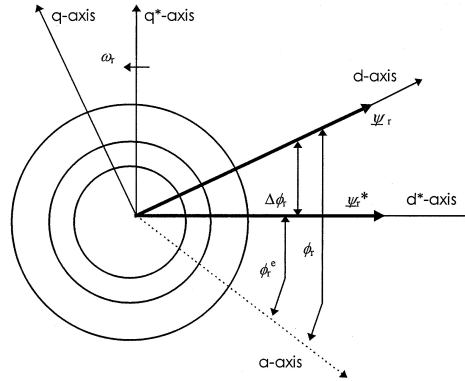
A pesar de que el esquema directo es robusto frente a variaciones paramétricas, el sensado del flujo del entrehierro lo vuelve algo problemático y costoso. Esto ha llevado a usar este enfoque indirecto o *feedforward* (inventado por Hasse), que es más sensible a conocer los parámetros del motor, pero no requiere de un sensado directo del flujo del rotor.

## 2.5. Estado del arte en estimadores de parámetros usando redes neuronales y sistemas neurodifusos

Para que se tenga un alto rendimiento en los distintos esquemas de control se requiere del conocimiento de algunos parámetros del motor de inducción. Los valores de los parámetros se usan dentro del controlador y pueden ser identificados de dos maneras: *fuera de línea*, durante la puesta en marcha de la máquina; sin embargo, la mayoría de los parámetros del motor de inducción varían durante su operación debido a la temperatura, frecuencia y saturación magnética [3], lo que conlleva a la segunda opción para los estimadores, identificación *en línea*.

La identificación *en línea* se realiza con el fin de evitar desajustes en el controlador; como consecuencia de algún error entre los valores paramétricos usados en el controlador y aquellos que el motor presenta se tendrá que la posición de flujo del rotor no coincida con la posición asumida por el controlador, llevándolo a un deterioro en su rendimiento [20]. En la Figura 2.11 se puede apreciar el error entre los ejes del flujo del rotor.

Para la técnica de estimación fuera de línea se ha desarrollado un procedimiento llamado *auto puesta en marcha*. La principal idea de este concepto es que el controlador determine automáticamente todos los parámetros necesarios para el control vectorial. Estos enfoques se describen en [21, 22, 23]. Las técnicas de este tipo varían respecto a la forma de excitación para la estimación de parámetros. En [24, 25] se aplica excitación a una sola fase en forma de DC, en forma de secuencia binaria pseudo aleatoria, y se requiere de un observador adaptivo. Por otra parte se tiene la excitación a una sola fase en forma de AC, donde se estudia la respuesta en frecuencia,



**Figura 2.11:** Error en el flujo del rotor entre el controlador y el valor real en el motor causado por un desajuste paramétrico.

tales trabajos se encuentran en [26, 27, 28].

Sin embargo, las técnicas de identificación en línea son mas variadas y se encuentran agrupadas. Se tienen técnicas por análisis espectral, donde la identificación se lleva a cabo por medio de mediciones de una respuesta de una señal inyectada de prueba, visualizando el espectro de corriente y/o voltaje. Estas señales son muestreadas y se derivan los parámetros del análisis espectral. Algunos trabajos recientes son [29, 30, 31]. Otra técnica es la que se basa en observadores en donde se desarrollan filtros Kalman extendidos y observadores Luenberger. El mayor inconveniente de este tipo de técnicas es el alto costo computacional. Algunos trabajos relacionados son [32, 33, 34]. También existe la técnica por medio de modelos de referencia basados en sistemas adaptivos, el cual es de facil implementación y la idea básica se centra en que una señal se mida de dos formas diferentes: la primera en referencias dentro del sistema de control y la segunda de mediciones directas. La diferencia de estas dos señales arrojará un error que puede ser usado en un mecanismo adaptivo (como un controlador PI). En esta técnica se utiliza el modelo de la máquina en el que su mayor inconveniente es que algunos parámetros son tratados como constantes. En las últimas tres décadas, los observadores y los modelos de adaptación del sistema de referencia han trazado un camino en el que es posible observar como la estimación de parámetros y el FOC han evolucionado [3] y al igual se ha estudiado el impacto de las variaciones paramétricas en varios esquemas de control [35, 36, 37, 38, 12].

El último grupo de identificación en línea esta aun en desarrollo, basada en el área de la inteligencia artificial, la aplicación de redes neuronales, la logica difusa y la combinación de estas, los sistemas neurodifusos. El porqué del uso de las redes neuronales

y los sistemas neurodifusos es debido a que pueden simular el comportamiento de sistemas dinámicos no lineales por su característica de aprendizaje y resolver problemas de reconocimiento de patrones, clasificación y ajuste de control. Basicamente, miran los patrones de datos y aprenden a clasificarlos. En la literatura se puede encontrar con varios tipos de estimación: estimación de flujo del rotor, estimación de la velocidad, estimación de la resistencia del estator y del rotor, estimación de la constante de tiempo del rotor [3].

La Figura 2.12 muestra como el control del motor se ha ido desarrollando y como por ejemplo, las redes neuronales han formado parte de la misma estimacion y control [39].

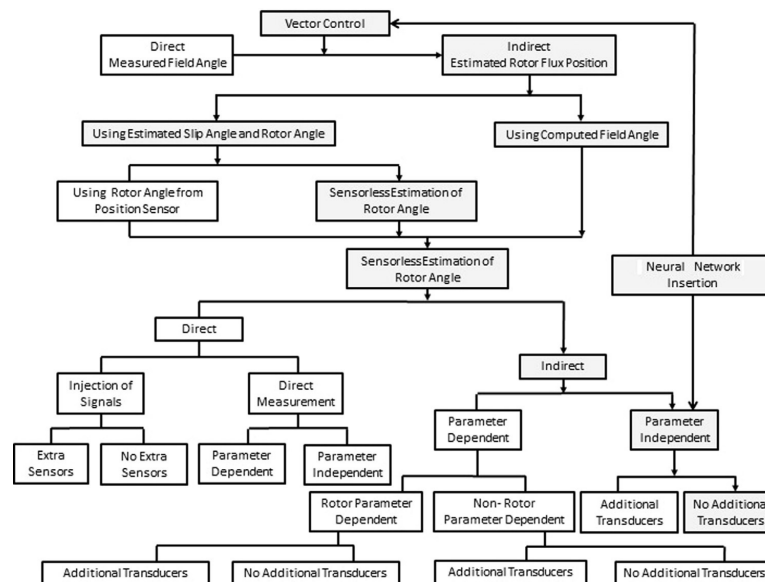


Figura 2.12: Clasificación de estimación de parámetros [3].

Para el control vectorial, como ya se mencionó, es necesario conocer la magnitud y posición del flujo del rotor. En [40] se presenta un estimador adaptivo para el FOC sin utilizar sensor de velocidad y mediante la medición de voltajes y corrientes de fases se describe una red neuronal basada en las componentes de flujo del rotor. En [41] la red neuronal propuesta para el control vectorial utiliza el marco de referencia orientado al flujo del rotor. Existe una variación del control vectorial, que es el control directo de par: para [42] se propone este tipo de control mediante un sistema neurodifuso usando la amplitud de flujo del estator y los errores del par electromagnético a través de ANFIS.



El artículo [43] presenta un estimador de flujo del rotor de dos redes neuronales conectados en cascada, las redes son capaces de estimar desde el punto de operación inicial. En [44] se utilizó un algoritmo genético para estimar los parámetros internos basado en el concepto del modelo de flujo de rotor. Para [8] y [45] se utilizaron redes neuronales como observadores de flujo y de la velocidad angular del motor; se sujeta a cambios de carga y de velocidad.

Las técnicas de inteligencia artificial ofrecen la ventaja de independencia de los parámetros y no se requiere un modelo exacto del sistema. Sin embargo, en algunos artículos prefieren usar modelos matemáticos del motor de inducción, como en [46] donde describe un modelo de corriente para estimar la resistencia del estator y un modelo de voltaje para la determinación del vector de flujo del rotor.

El principal tipo de red neuronal utilizado para la estimación de flujo del motor es la llamada *Feed-forward*, que viene siendo una versión de la *Multi-Layer Perceptron*. En [47], se entrenó una red *FNN*, implementada para estimar el flujo del rotor, donde el error de la estimación del flujo estuvo por debajo del 2%. En la ref.[14], el error entre el flujo y la red neuronal fue entre un 3% y 4%. La ref.[11] presenta una red con arquitectura 4 – 13 – 12 para el control del flujo con dos pruebas diferentes: una con cambios de carga súbitos y otro con aceleraciones y desaceleraciones únicamente con la inercia del motor (al vacío). Tomo cerca de 500,000 iteraciones para entrenar a la red. Trabajos más actuales como el [4] presentan la estimación del flujo en un par de redes neuronales conectadas en cascada y entrenadas con retropropagación, la arquitectura de las redes se presenta en la Fig. 2.13.

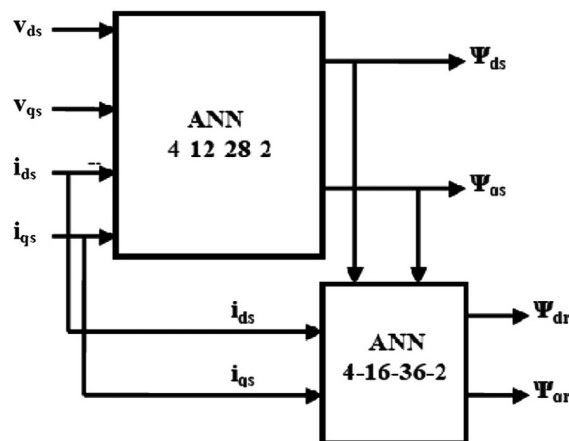


Figura 2.13: Arquitectura de redes neuronales de Ref.[4].



# Redes Neuronales y Sistemas Neurodifusos

---

Las redes neuronales y los sistemas neurodifusos son un conjunto de modelos que forman parte de la llamada *inteligencia artificial* ó *inteligencia computacional*. Estos modelos son elementales en la resolución de problemas practicos, tales como el reconocimiento de patrones, aproximación de funciones, predicción, regresiones, clasificadores, y también sistemas dinámicos.

## 3.1. Redes Neuronales

Las redes neuronales se basan en el cerebro humano, debido al comportamiento *computacional* que presentan. El cerebro humano trabaja como una computadora compleja, con procesamiento en paralelo no lineal, teniendo la capacidad de un almacenamiento estructural en forma de *neuronas*. Es por ello, que las redes neuronales tienen un comportamiento biologico con base en un razonamiento y en una toma de decisiones a partir de ciertos datos externos.

En su forma más general, una red neuronal es una máquina que esta diseñada para modelar el camino por el cual el cerebro lleva a cabo una tarea en particular o una función de interes. Se asemeja a las funciones del cerebro en dos aspectos: las redes neuronales adquieren conocimiento del medio ambiente a través de un proceso de aprendizaje y los pesos sinapticos funcionan como unidades de almacenamiento del conocimiento adquirido; las redes neuronales son usualmente implementadas por componentes electronicos o simulada en software [5]. Es por ello, que se puede decir que las redes neuronales estan inspiradas en la neurociencia como modelos ma-

temáticos inspirados en sistemas biológicos, adaptados y simulados en computadoras convencionales.

### 3.1.1. Fundamentos de las Redes Neuronales

Las redes neuronales tienen una fuerza de cómputo por la estructura distribuida masivamente paralela y por la capacidad de aprender, y por lo tanto, generalizar. La *generalización* de una red neuronal es la capacidad de dar resultados razonables ante los datos de entrada que no hayan aparecido durante el entrenamiento (aprendizaje). Por lo tanto, estas dos características hacen posible que las redes neuronales tengan la capacidad de solucionar problemas complejos. Las redes neuronales tienen las siguientes propiedades:

- No linealidad. Una neurona artificial puede ser lineal o no lineal. Una red neuronal puede estar compuesta de interconexiones de neuronas no lineales. La no linealidad es una propiedad particular ya que si el mecanismo responsable de la generación de la señal de entrada es inherentemente no lineal.
- Mapeo de entradas y salidas. El entrenamiento *supervisado* de las redes neuronales involucra la modificación de los pesos sinápticos al aplicar a la red ejemplos de entrenamiento y ejemplos de objetivos. El entrenamiento se lleva a cabo hasta que ya no existan cambios significativos en los pesos. Así es como la red neuronal aprende de ejemplos construyendo un mapeo de entradas y salidas.
- Adaptabilidad. Las redes neuronales tienen una capacidad incorporada para adaptar sus pesos sinápticos a los cambios en el ambiente circundante.

### 3.1.2. Arquitectura

Como se ha mencionado, las redes neuronales se basan en el comportamiento de las neuronas. Una neurona es una unidad que procesa información y es fundamental en la operación de una red neuronal. El modelo de una neurona se muestra en la Figura 3.1. La forma de esta neurona es la base de varias redes neuronales. La neurona por tanto, cuenta con tres elementos básicos: un conjunto de pesos sinápticos, cuya función es conectar una entrada con una neurona; un sumador para sumar las señales de entrada, ponderado por las respectivas fuerzas sinápticas de la neurona y una función de activación para limitar la salida de la neurona.

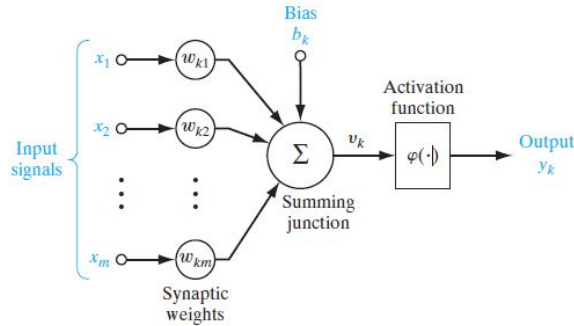


Figura 3.1: Modelo de una neurona [5].

El modelo de la neurona también incluye una entrada externa llamada *bias*, teniendo la función de incrementar o disminuir la entrada neta de la función de activación. En terminos matemáticos se puede describir al modelo de la neurona de la siguiente manera:

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (3.1.1)$$

y

$$y_k = \varphi(u_k + b_k) \quad (3.1.2)$$

donde  $x_1, x_2, \dots, x_m$  son señales de entrada;  $w_{k1}, w_{k2}, \dots, w_{km}$  son los respectivos pesos sinápticos de la neurona  $k$ ;  $b_k$  es el bias de la neurona  $k$ .

Una vez visualizado el principal elemento de las redes neuronales, se describe su arquitectura o la forma en que las neuronas llegan a estar interconectadas entre si, ligado también al tipo de algoritmo de entrenamiento de la red. La primera arquitectura que se presenta es la red de una sola capa. En esta arquitectura se tiene una *capa de entradas* que van interconectadas a una *capa de salida*. En la Figura 3.2 se aprecia este tipo de arquitectura.

El segundo tipo de arquitectura tipo *Feedforward* se distingue por la presencia de una o más capas ocultas. El termino *ocultas* se refiere a que estas partes de la red neuronal no es visible. La función principal de las capas ocultas es la interacción de las entradas externas y la salida de la red neuronal. En la Figura 3.3 se aprecia el diseño de una red con una capa oculta y una capa de salida.

El otro tipo de arquitectura es la red neuronal recurrente y se distingue de la

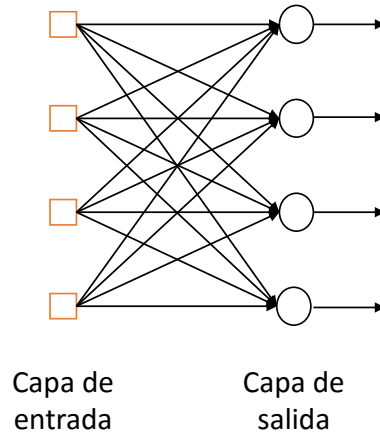


Figura 3.2: Red *Feedforward* de una sola capa de neuronas.

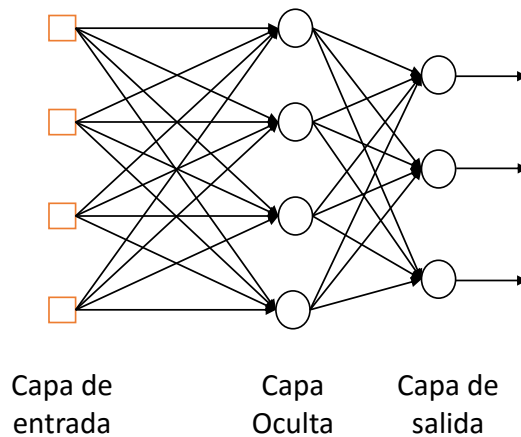


Figura 3.3: Red *Feedforward* con una capa oculta de neuronas.

*Feedforward* en que al menos tiene un lazo de realimentación. Se puede decir que este tipo de redes neuronales cuentan con *memoria* al introducir retrasos de tiempo en la estructura sináptica de la red. Este tipo de redes realizan el intercambio de información entre neuronas de una manera mucho más compleja y por sus características, pueden predecir varios eventos. En la Figura 3.4 se ilustra una red neuronal de 4 neuronas y se puede notar el uso de elementos de retraso llamados *delays*. La Figura 3.5 se ilustra otro tipo de red recurrente con neuronas ocultas. Las conexiones de realimentación se originan de las neuronas ocultas así como de las salidas.

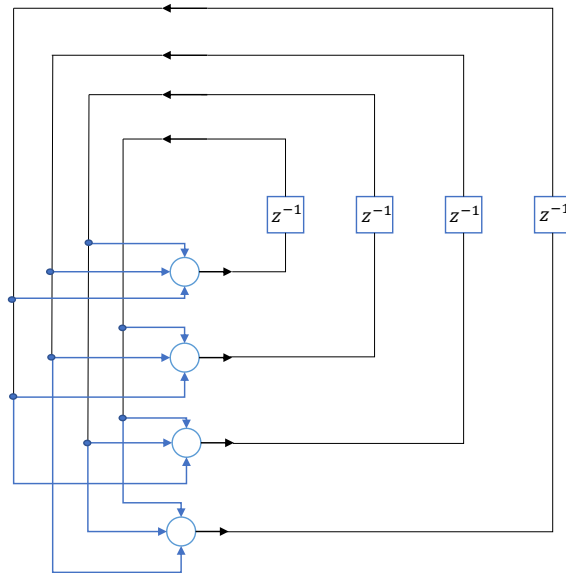


Figura 3.4: Red recurrente sin lazos de realimentación y sin neuronas ocultas.

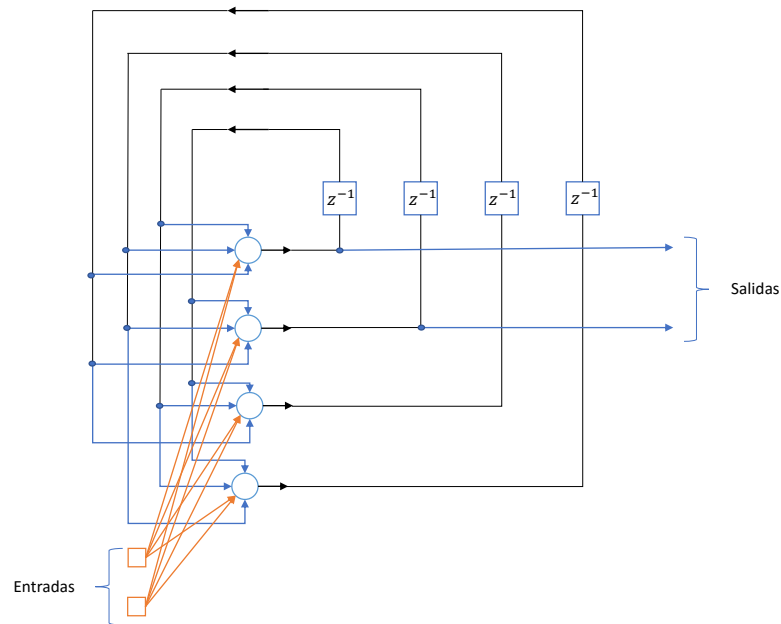


Figura 3.5: Red recurrente con neuronas ocultas.

### 3.1.3. Diseño de Redes Neuronales

En el diseño de redes neuronales no existe una fórmula, por lo tanto, se tiene una inherente experimentación y no se puede evitar una cierta experimentación de

prueba y error. Al diseñar una red neuronal se debe cumplir con cierta precisión y generalización; esto dependerá de la topología de la red, es decir, el número de neuronas y el número de capas ocultas.

Depende del problema a resolver para poder discernir la cantidad de neuronas, también dependen del tipo de problema a resolver y su complejidad. El número de neuronas debe ser suficiente para la correcta ejecución del problema pero al mismo tiempo debe ser bajo para permitir la generalización.

El uso de muy pocas neuronas puede provocar un *sub-ajuste*, es decir, la red neuronal no podrá dar una solución correcta al problema. Por lo contrario, si se utilizan demasiadas neuronas, el tiempo de entrenamiento se incrementará pudiendo causar un *sobre-ajuste*. Cuando ocurre esto, el error en el entrenamiento es muy pequeño, pero cuando se utilizan nuevos datos, el error se vuelve muy grande. Es decir, la red neuronal ha memorizado los datos de entrenamiento, pero no ha aprendido a generalizar otros valores.

Existe un método para calcular el número de neuronas ocultas llamado regla de la pirámide geométrica. En este método se tiene que las neuronas ocultas tendrán una forma piramidal desde la entrada hasta la salida. Para calcular el número de neuronas se tiene:

$$h = \sqrt{m * n} \quad (3.1.3)$$

donde  $h$  es el número de neuronas de la capa oculta,  $m$  es el número de neuronas de la salida,  $n$  es el número de entradas. Se debe cumplir con que  $n > h$ .

### 3.1.4. Funciones de Activación

El cálculo de  $\delta$  de cada neurona de una red neuronal requiere del conocimiento de la derivada de la función de activación asociado con esa neurona. Para eso, se requiere que la función de activación sea continua. En términos esenciales, se requiere que la función sea *diferenciable*. Como ejemplo se tienen las siguientes funciones de activación:

#### Tangente Hiperbólica Sigmoidea

Algunos procesos tienen a estabilizarse mediante curvas, debido a que presentan progresión temporal. Las funciones tipo sigmoide permiten describir este tipo de evo-



lución. La función tangente hiperbolica sigmoideal presenta la siguiente configuración. En la Figura 3.6 se ilustra la función de activación y el algoritmo para su implementación en la ecuación 3.1.4. Cabe mencionar que el rango de salida de esta función de activación esta entre  $[-1, 1]$ , siendo esta función la más empleada en entrenamientos supervisados.

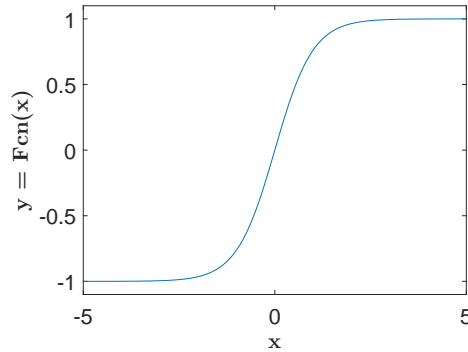


Figura 3.6: Tangente Hiperbolica Sigmoidea.

$$y = \frac{2}{1 + \exp(-2x)} - 1 \quad (3.1.4)$$

### Sigmoide Logarítmica

Esta función de activación se asemeja a la tangencial, con la diferencia que los valores de salida estan en el rango de  $[0, 1]$  como se aprecia en la Figura 3.7. Se pueden utilizar datos continuos o digitales proporcionando salidas exclusivamente continuas. El algoritmo de esta función de activación se describe en la ecuación 3.1.5.

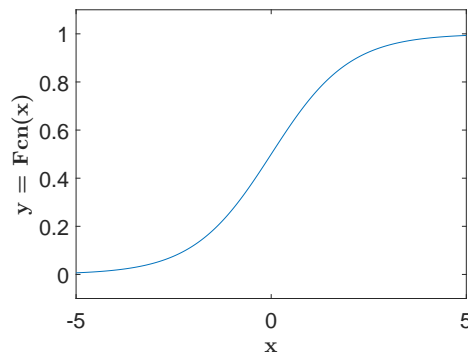


Figura 3.7: Logarítmica Sigmoidea.

$$y = \frac{1}{1 + \exp(-x)} \quad (3.1.5)$$

## Lineal

Al utilizar esta función de activación, la salida de las neuronas toman cualquier valor ya que se tiene una salida lineal. Se aprecia en la Figura 3.8.

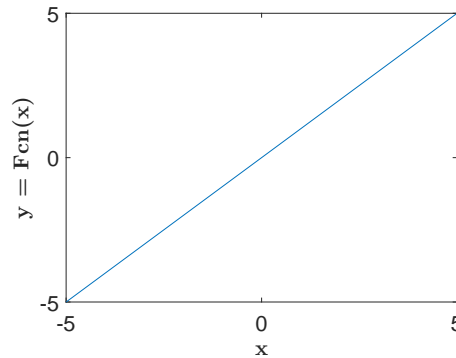


Figura 3.8: Lineal.

### 3.1.5. Algoritmos de Entrenamiento

Para hablar de algoritmos de entrenamiento se tiene que trata primero las formas de aprendizaje de una red neuronal. Al momento de entrenar una red neuronal se realiza un determinado tipo de procesamiento. Partiendo de un conjunto de pesos sinápticos aleatorios, que durante el proceso de aprendizaje se van refinando iterativamente hasta alcanzar un nivel de solución relativamente bueno. Existen tres tipos de paradigmas de aprendizaje:

- Aprendizaje Supervisado. Se presenta en la red neuronal un conjunto de patrones de entrada junto con un patron de salida esperado. Los pesos se van modificando de manera que se vaya reduciendo el error entre la salida real de la red neuronal y la salida esperada.
- Aprendizaje No Supervisado. Se presenta a la red un conjunto de patrones de entrada. No existe información disponible a la salida. el proceso de entrenamiento en este caso se ajustan los pesos con base a la correlación existente entre los datos de entrada.

- Aprendizaje por refuerzo. Este aprendizaje esta basado en los dos anteriores. La diferencia es que no se proporcionan valores de salida esperados. Este tipo de aprendizaje es ideal cuando no se conoce la salida exacta.

En Matlab se tienen varias opciones al momento de entrenar una red neuronal. Es algunas veces difícil saber que algoritmo de entrenamiento escoger debido a que se requiere rapidez para cierto problema. Esto dependerá de muchos factores, alguno de ellos son: la complejidad del problema, el número de datos para el entrenamiento, el número de pesos y bias de la red neuronal, el error objetivo. En la siguiente tabla se ilustran los algoritmos que presenta Matlab.

Tabla 3.1: Algoritmos de Entrenamiento.

Algoritmo	Descripción
trainlm	Levenberg-Marquardt
trainbfg	BFGS Quasi-Newton
trainrp	Resilient Backpropagation
trainscg	Scaled Conjugate Gradient
traincgb	Conjugate Gradient with Powell
traincgf	Fletcher-Powell Conjugate Gradient
traincgp	Polak-Ribière Conjugate Gradient
trainoss	One Step Secant
traingdx	Variable Learning Rate Backpropagation

## 3.2. Sistemas Neurodifusos

El modelado de sistemas basados en herramientas matemáticas convencionales, no es conveniente para sistemas que están mal definidos o son de naturaleza incierta. Por otro lado, los sistemas modelados en inferencia difusa emplean reglas *if-then* y de esta manera se pueden modelar cualitativamente por el comportamiento y/o conocimiento humano, sin emplear un análisis cuantitativo.

En 1993 Jyh-Shing Roger Jang presenta una arquitectura basada en un sistema de inferencia difuso adaptable (llamado *ANFIS*) en el que se construyen reglas difusas tipo *if-then* con funciones de membresía apropiadas para generar los pares de entrada-salida estipulados [48].

### 3.2.1. Fundamentos de los Sistemas Neurodifusos

#### Reglas difusas

Las reglas difusas *if-then* o declaraciones condicionales son expresiones del tipo *si A entonces B* donde *A* y *B* son etiquetas o conjuntos difusos, caracterizados por funciones de membresía apropiados. Estas reglas difusas con frecuencia se emplean debido que ejemplifican de manera adecuada la habilidad humana en la toma de decisiones en un ambiente complejo y/o incierto. Un ejemplo de regla difusa es el siguiente:

*Si la presión es **ALTA**, entonces el volumen es **BAJO**.*

donde la *presión* y el *volumen* son variables lingüísticas, *alta* y *bajo* son valores lingüísticos o etiquetas que son caracterizados por funciones de membresía.

Otra forma de las reglas difusas *if-then* propuesta por Takagi-Sugeno, donde solo se involucra la parte de la premisa. Como ejemplo se tiene:

*Si la velocidad es **ALTA**, entonces la fuerza =  $k * (velocidad)^2$*

donde *ALTA* es la parte de la premisa y es una etiqueta lingüística caracterizada por una función de membresía apropiada. Sin embargo, la parte consecuente se describe por una ecuación no difusa de la variable de entrada. Es así, que las reglas difusas son parte esencial del sistema de inferencia difuso.

#### Sistemas de Inferencia difusos

Los sistemas de inferencia difusos se conocen también como sistemas basados en reglas difusas, modelos difusos o controladores difusos cuando son usados como controladores. Básicamente estos sistemas se componen de cinco bloques elementales, representados en la siguiente Figura 3.9:

- Una base de Reglas que contiene un cierto número de reglas *if-then*.
- Una base de datos en la cual se definen las funciones de membresía de los conjuntos difusos usados en las reglas difusas.
- Una unidad de toma de decisiones que realiza las operaciones de inferencia sobre las reglas.
- Una interfaz *fusificadora* la cual transforma las entradas en grados de membresía.

- Una interfaz *desfusificadora* la cual transforma los resultados difusos en una salida nítida.

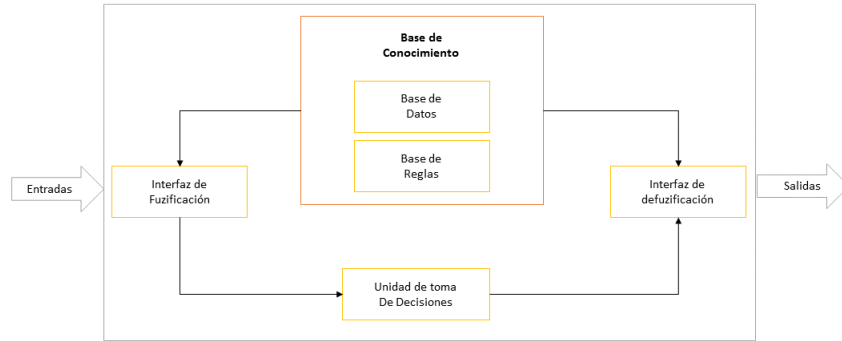


Figura 3.9: Sistema de Inferencia Difusa.

Cabe mencionar que cada proceso tiene un objetivo dentro del sistema de inferencia. Para la *Fusificación* se obtiene el grado de membresía de cada parámetro dentro de los conjuntos establecidos. Los grados de membresía que se seleccionan se combinan por medio de la *norma T* para obtener el peso  $\omega$  de cada regla. Se generan parámetros consecuentes a partir de los pesos y así la defusificación generará un grado de membresía para cada parámetro consecuente obtenido.

### 3.2.2. Arquitectura

La arquitectura de *ANFIS* (Adaptive Neuro-Fuzzy Inference System por sus en inglés) es una red multicapa tipo *Feedforward* adaptativa, la cual como se ha mencionado, permite la aplicación de la lógica difusa junto con una red neuronal. También se puede aplicar a una estructura recurrente cuando el algoritmo de aprendizaje lo requiera. La red ANFIS tiene cinco capas, pero solo dos capas tienen nodos adaptables.

En la Figura 3.10 se ilustra la arquitectura de una Red ANFIS. Para una comprensible explicación, solo se manejan dos entradas  $x$  y  $y$ , una salida  $f$  en la red y cada entrada cuenta con dos funciones de membresía tipo Takagi-Sugeno *if-then*. Cada capa de la arquitectura de la red cumple una función en general pero son dependientes una de otra. Entonces se tiene:

$$\text{Regla 1: Si } x \text{ es } A_1 \text{ y } y \text{ es } B_1, \text{ entonces } f_1 = p_1x + q_1y + r_1$$

$$\text{Regla 2: Si } x \text{ es } A_2 \text{ y } y \text{ es } B_2, \text{ entonces } f_2 = p_2x + q_2y + r_2$$

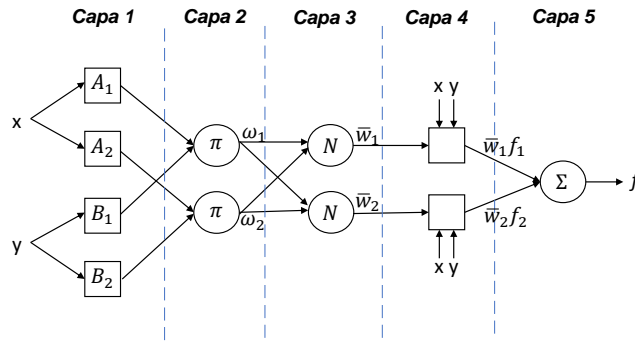


Figura 3.10: Arquitectura ANFIS de dos entradas.

### Capa 1

Esta capa es la de entrada. El objetivo de esta capa esencialmente es la de transformar los valores de entrada *crisp* en valores de magnitudes difusas a través de las funciones de membresía. En cada nodo  $i$  en esta capa se ejecuta la función de membresía:

$$O_i^1 = \mu_{A_i}(x) \quad (3.2.1)$$

donde  $x$  es la entrada al nodo  $i$ ,  $A_i$  es la etiqueta lingüística asociada con esta función del nodo.  $O_i^1$  es la función de membresía de  $A_i$  y especifica el grado en que  $x$  satisface a  $A_i$ .

### Capa 2

Cada nodo en esta capa se etiqueta con el símbolo  $\Pi$ , el cual multiplica las señales de entrada y como salida es el producto. La salida de cada nodo representa la *fuerza de disparo* de cada regla. Se tiene:

$$\omega_i = \mu_{A_i}(x) * \mu_{B_i}(y), i = 1, 2, \dots \quad (3.2.2)$$

### Capa 3

Los nodos en esta capa están etiquetados con  $N$ . El nodo  $i$  calcula la razón entre la *fuerza de disparo* y la sumatoria de todas las *fuerzas de disparo*. Los nodos de esta

capa normalizan el grado de membresía que adquirió  $\omega$  en la capa anterior y así se establece un rango adecuado entre los pesos asegurando la homogeneidad en las capas posteriores:

$$\bar{\omega}_i = \frac{\omega_i}{\omega_1 + \omega_2}, i = 1, 2, \dots \quad (3.2.3)$$

#### Capa 4

En esta capa, los nodos  $i$  se denotan con la función:

$$O_i^4 = \bar{\omega}_i f_i = \bar{\omega}_i (p_i x + q_i y + r_i) \quad (3.2.4)$$

donde  $p_i$ ,  $q_i$  y  $r_i$  son los parámetros consecuentes. Esta capa esta compuesta de nodos adaptivos y se obtienen funciones que dependen de los pesos de la red y parámetros consecuentes. Aqui se lleva a cabo el proceso de aprendizaje de la red, pues el valor que adopten los parámetros determina la similitud de salida con el patron de entrenamiento.

#### Capa 5

Esta capa ejerce la función de salida y solo hay un nodo en esta capa y calcula todas las salidas de la capa anterior mediante una sumatoria:

$$O_1^5 = \text{overall output} = \sum_i \bar{\omega}_i f_i = \frac{\sum_i \omega_i f_i}{\sum_i \omega_i} \quad (3.2.5)$$

De esta manera queda consolidada la arquitectura de una red adaptiva la cual tiene una funcionalidad por medio de un sistema de inferencia difuso.

### 3.2.3. Algoritmos de Entrenamiento

Existen diferentes reglas de aprendizaje para redes adaptivas en donde se sintonizan los parámetros primicia de la capa uno y los parámetros consecuentes de la capa cuatro.

Algunos algoritmos de aprendizaje que se aplican en ANFIS son los siguientes [49]:

- **Retropropagación.** Es necesario que la red tenga estructura recurrente ya que los nodos adaptables se ajustan por medio del método del gradiente descendente hasta que se obtenga el resultado deseado. Al inicio se fijan los parámetros primicia de la capa uno para determinar los parámetros consecuentes de la capa cuatro; se calcula el error entre la salida de la red y la deseada. Si el error es mayor a la tolerancia especificada, se modifican los parámetros primicia sin alterar los parámetros consecuentes y así obtener otro valor en la salida. Al aplicar este método se requiere que las funciones de membresía sean funciones continuas y derivables, como en las funciones de activación de las redes neuronales.
- **Hibrido.** Es la combinación del método de gradiente descendente con el estimador de mínimos cuadrados. En este tipo de aprendizaje existe el llamado *Fuera de línea* y *En línea*. En el primero se considera la matriz de la red y los datos de entrenamiento para la obtención de los parámetros consecuentes y en el segundo es necesario, cuando se tiene que actualizar cada elemento conforme se presenten cambios en el sistema.



# Estimación de Parámetros

---

En este capítulo se describe la metodología propuesta para la estimación de parámetros de un motor de inducción con velocidad constante y velocidad variable del motor, con cierto par de carga mecánica; usando redes neuronales recurrentes y sistemas neurodifusos tipo ANFIS.

## 4.1. Adquisición de datos para entrenamiento

En esta sección se describe cómo se llevo a cabo la adquisición de datos para el entrenamiento de las redes neuronales y los sistemas ANFIS que estiman el flujo del rotor, la posición de flujo del rotor y la velocidad del motor. Las señales para el entrenamiento (entradas-salidas) son adquiridas de la simulación del MI con el control vectorial en Simunlink, en el que se emplean bloques designados a cierta función.

Tabla 4.1: Datos del Motor de Inducción.

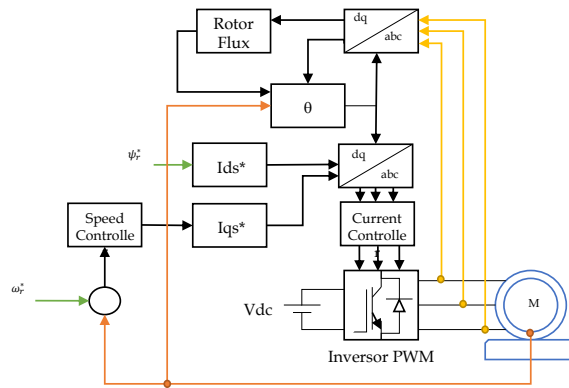
Motor de inducción Tipo Jaula de Ardilla	
Entrada Mecánica	Par (Tm)
Frecuencia Nominal	60 Hz
Potencia	3 H.P.
Pares de Polos	2
Resistencia Estator	0.435
Inductancia Estator	4e-3
Resistencia Rotor	0.816
Inductancia Rotor	2e-3
Inductancia Mutua	69.31e-3
Inercia	0.089J(Kg.m <sup>2</sup> )

Las señales adquiridas de corrientes son descompuestas en el marco de referencia utilizado en el sistema de control vectorial. La simulación que se realizó en MATLAB/Simulink permite tomar las señales y almacenarlas en el *Workspace*. En dicha implementación se utilizó un MI con las siguientes características mostradas en la Tabla 4.1.

Cada bloque es de vital importancia para la simulación del control vectorial. El motor de inducción es alimentado por un inversor PWM, el cual opera como una fuente de corriente sinusoidal trifásica. A continuación se presenta el funcionamiento de cada bloque de la simulación y en la Figura 4.1 se ilustra su implementación en Simulink:

- Uno de los bloques principales es el que modela la máquina asíncrona en el que se implementa una máquina trifásica asíncrona, ya sea de rotor devanado, de jaula de ardilla sencilla o doble. Puede operar en modo generador o en modo motor. El modo de operación está dado por la señal mecánica de par; si el par es positivo actuará como motor, si el par es negativo actuará como generador.
- Otro bloque de importancia es el bloque que funciona como inversor, el cual es un convertidor de potencia trifásico que consta de hasta seis interruptores de potencia conectados en una configuración de puente.
- El bloque de transformación  $abc$  a  $dq$  (*Park*) funciona para convertir el sistema trifásico convencional al sistema de referencia de dos ejes: *directo* y *de cuadratura*. También está el bloque de transformación inversa, donde el sistema de referencia de dos ejes  $dq$  se transforma al sistema trifásico  $abc$ .
- Existen bloques donde se calcula la magnitud del flujo del rotor y la posición. Para calcular la magnitud únicamente se requiere de la entrada de la corriente directa  $I_d$  y para el cálculo de la posición se requiere de la corriente de cuadratura  $I_q$ , la magnitud del flujo  $\psi_r$  y la velocidad del motor  $\omega_m$ .
- Los bloques donde se calculan las corrientes directa y de cuadratura de referencia que servirán para la transformación inversa Park. Las corrientes de referencia se obtienen a partir de referencias de flujo y del par.
- El bloque que controla la velocidad funciona por medio de una velocidad de

referencia  $\omega^*$  y la velocidad del motor  $\omega_m$  y la diferencia se integra en un controlador PI, que a su vez da como salida un par de referencia  $Te^*$ .



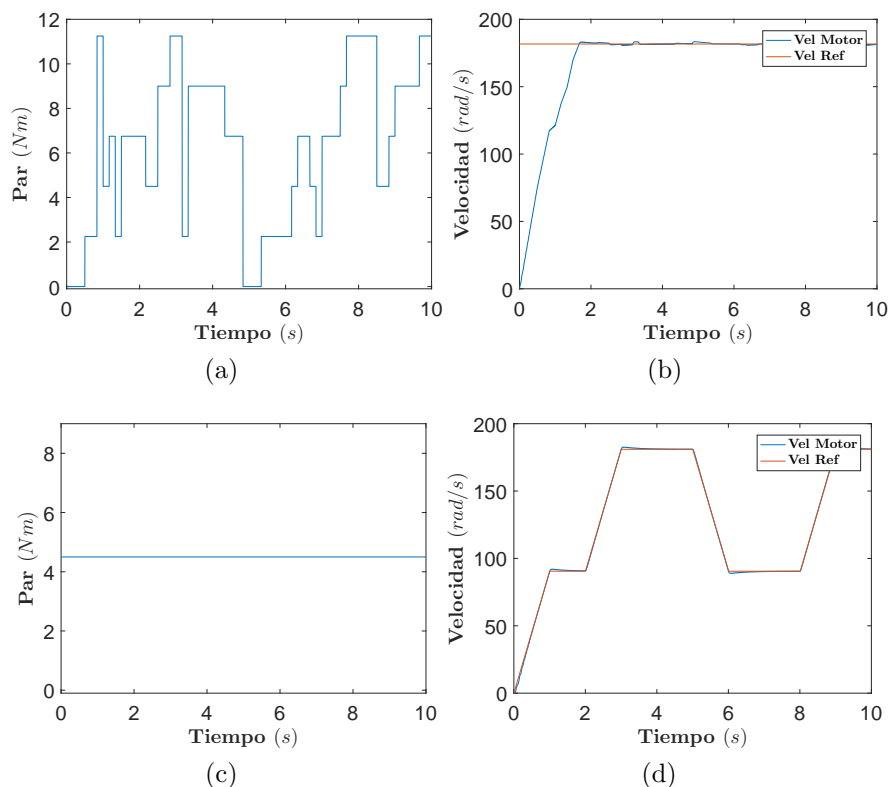
**Figura 4.1:** Diagrama de control vectorial para el motor de inducción implementado en Simulink.

Las señales de entrenamiento se obtuvieron mediante simulaciones de dos casos diferentes. En el primer caso, se evaluó el comportamiento del sistema mediante variaciones del par de carga a velocidad constante en la referencia del sistema de control del motor de inducción. En el segundo caso, de igual forma se evaluó el comportamiento del sistema mediante variaciones de la velocidad en la referencia, manteniendo el par de carga constante.; en la Figura 4.2 se muestran el par de carga y la velocidad para cada caso estudiado.

Posteriormente de la simulación, se obtienen las señales de corrientes directa y de cuadratura, seleccionadas como señales de entrada para las redes neuronales y los sistemas ANFIS. Como señales de salida para el entrenamiento, se obtienen las señales de flujo del rotor, la velocidad del motor y las posiciones  $\sin(\theta)$  y  $\cos(\theta)$ , y son almacenadas en el *Workspace* para su preprocesamiento.

## 4.2. Entrenamiento de Redes y ANFIS

En esta sección se describe como se lleva a cabo el entrenamiento de las redes neuronales y los sistemas ANFIS para la estimación del flujo del rotor, posición del

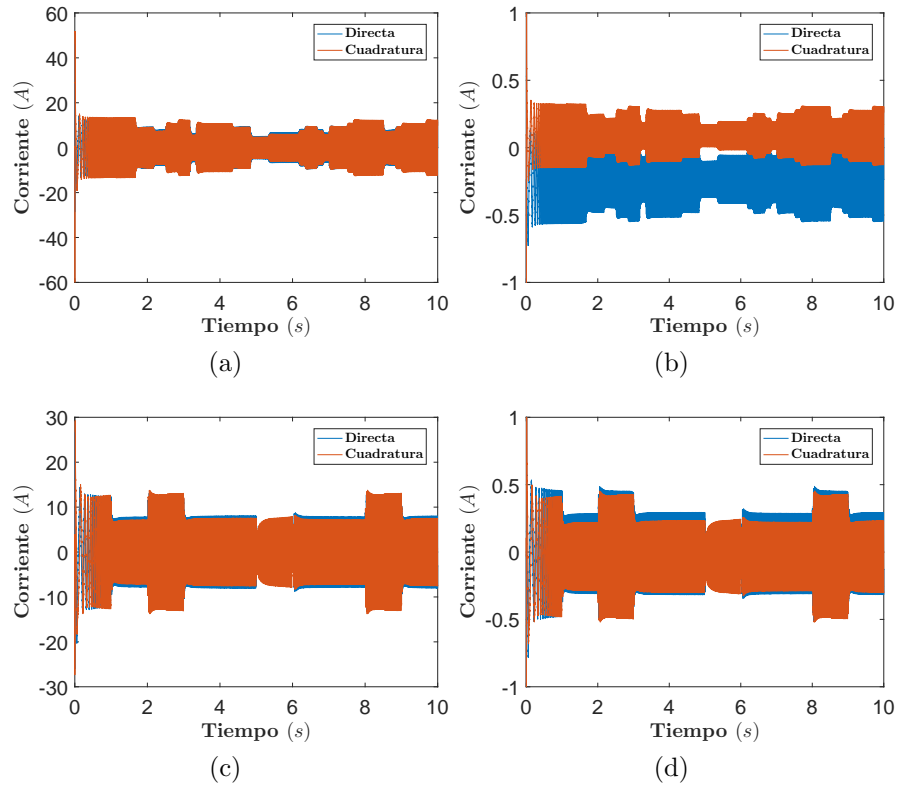


**Figura 4.2:** (a) Variación pseudoaleatoria del par de carga con valores al vacío (0 Nm), media carga (4.5 Nm) y plena carga (9 Nm); (b) Velocidad del Motor con el valor de referencia fijo nominal; (c) Par de carga constante; (d) Variación de la velocidad del motor.

rotor y la velocidad del motor. Como se mencionó anteriormente, se manejan dos casos: a) la velocidad de referencia se mantiene constante a un valor nominal y con la variación pseudoaleatoria del par de carga; y b) la velocidad de referencia es variable y el par de carga se mantiene constante.

El preprocesamiento de las señales para el entrenamiento es fundamental para el funcionamiento de las redes. En primera parte, se disminuye la tasa de muestreo por un factor entero  $n$ ; después se tiene que realizar la normalización de las entradas y de las salidas objetivos de manera que estén en un rango específico,  $[1, -1]$ , esto con el fin de evitar una *saturación* en las funciones de activación de las redes neuronales. En la Figura 4.3 se ilustra las señales de entrada para el entrenamiento de las redes neuronales y sistemas ANFIS del primer y segundo caso. Únicamente se utilizaron señales normalizadas para el entrenamiento de las redes neuronales, los sistemas ANFIS no

utilizan señales normalizadas.



**Figura 4.3:** (a) y (b) Corrientes de los ejes directo y cuadratura sin normalizar y con normalización del caso 1; (c) y (d) Corrientes de los ejes directo y cuadratura sin normalizar y con normalización del caso 2.

Para obtener todos los datos de las simulaciones en cada caso, se emplearon *10 segundos* de simulación. Esto con el proposito de obtener cada caso en un solo archivo. Se hicieron dos simulaciones para cada caso, en el que se obtuvieron los datos de entrenamiento y los datos de prueba. En la Figura 4.4 se tiene un diagrama donde se ilustra la metodología del proceso de entrenamiento de las redes, con el objetivo de obtener la mejor red neuronal o el mejor sistema ANFIS para la estimación del flujo del rotor, la posición del rotor y la velocidad del motor. Este proceso se repitió para cada caso a evaluación.

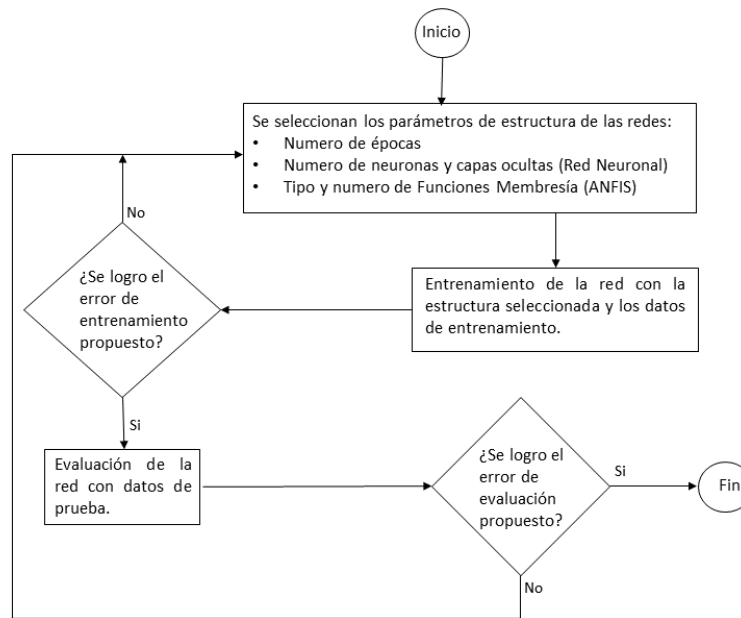


Figura 4.4: Metodología de entrenamiento.

### 4.2.1. Redes Neuronales

Una vez que se obtuvieron los datos de entrenamiento y hayan sido preprocesados, se someten a un solo conjunto para el entrenamiento con las redes neuronales. Matlab ofrece un toolbox para el entrenamiento de redes neuronales. Se utilizaron redes neuronales dinámicas ya que generalmente son más *potentes* que las redes estáticas y porque además tienen *memoria*. Pueden ser entrenadas para aprender secuencialmente o con patrones variables en el tiempo. En Matlab se utilizaron redes llamadas NARXNET, las cuales pueden predecir a partir de valores pasados. Matemáticamente se pueden describir de la siguiente manera:

$$y(t) = f(x(t), x(t-1), \dots, x(t-d), y(t-1), \dots, y(t-d)) \quad (4.2.1)$$

Como se menciona en la Figura 4.4, la configuración de las redes se llevaron a cabo por medio de un razonamiento de tipo heurístico, es decir, realizando pruebas y observando los resultados de los factores de correlación, el error medio absoluto y el error medio cuadrático. El total de número de datos obtenidos para este caso fue de 50001. Como se llevo a cabo un aprendizaje supervisado, también se generan los valores objetivo, es decir, los valores deseados. Para el primer caso, se obtuvieron

redes para la estimación de flujo del rotor, la posición  $\sin(\theta)$  y  $\cos(\theta)$ , y la velocidad del motor. En la Tabla 4.2 se muestran los errores obtenidos de la mejor arquitectura para cada estimación del primer caso. Los errores fueron obtenidos de las siguientes ecuaciones.

El Factor de Correlación:

$$FC = 1 - \frac{\sum_{i=1}^N [X_i(exp) - X_i(pred)]^2}{\sum_{i=1}^N [X(exp)]^2} \quad (4.2.2)$$

El Error Medio Absoluto:

$$MAE = \frac{1}{N} \sum_{i=1}^N |X_i(exp) - X_i(pred)| \quad (4.2.3)$$

El Error Medio Cuadrático:

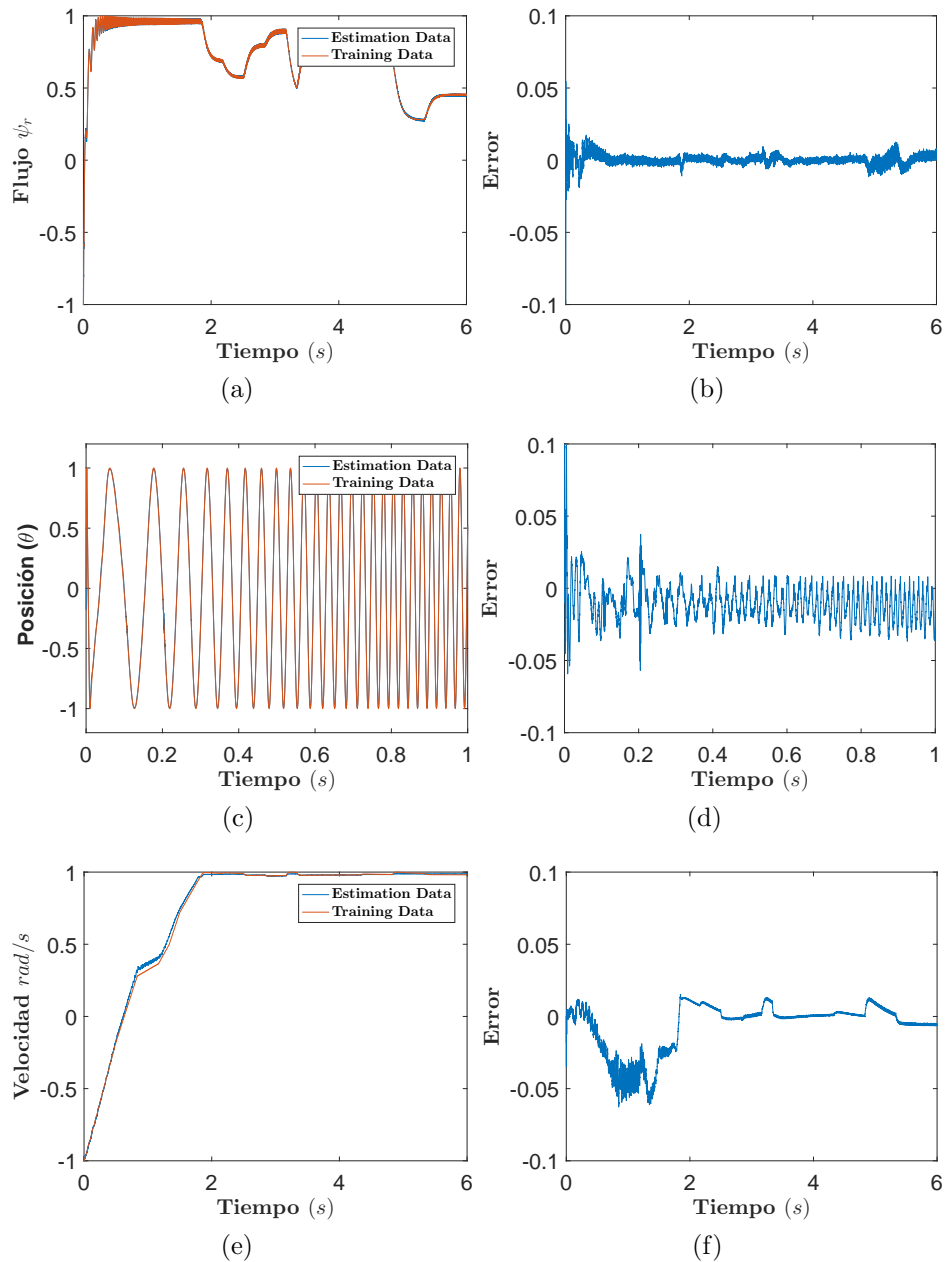
$$RMSE = \sqrt{\frac{\sum_{i=1}^N [X_i(exp) - X_i(pred)]^2}{N}} \quad (4.2.4)$$

**Tabla 4.2:** Errores de los datos de entrenamiento de las redes neuronales del caso1.

	<b>Flujo del Rotor</b>	<b>Posición del Rotor</b>	<b>Velocidad</b>
<b>Factor de Correlación</b>	0.99998	0.9991	0.9998
<b>Error Medio Absoluto</b>	0.002086	0.0175	0.0082
<b>Error Medio Cuadrático</b>	0.003137	0.0209	0.0144

En la Figura 4.5 se muestran las respuestas del entrenamiento de las redes neuronales del caso 1 para la estimación del flujo del rotor, la posición del flujo del rotor y la velocidad del rotor. Se visualiza el comportamiento real y el dato que estimó la red neuronal en el entrenamiento.

Para la red que estima el flujo del rotor se obtuvo una arquitectura de 4 capas ocultas y una capa de salida dando un número de neuronas en cada capa de la siguiente manera: 11 – 9 – 7 – 5 – 1; al inicio de la red neuronales se tienen líneas de retardo para las señales de entrada y para la señal de salida que se retroalimenta. Para cada entrada se tienen 8 retardos y para la salida también se tienen 8 retardos. La red que estima la posición del flujo del rotor se obtuvo una arquitectura más sencilla, en la que únicamente se necesitaron de 2 capas ocultas con el siguiente



**Figura 4.5:** Resultado del entrenamiento de las redes neuronales para el caso 1 en la estimación del (a) Flujo del rotor; (b) Error del entrenamiento del Flujo; (c) Posición  $\sin(\theta)$ ; (d) Error del entrenamiento de la posición; (e) Velocidad del motor; (f) Error del entrenamiento de la velocidad.

número de neuronas: 5 – 3 – 2; de igual forma, se tienen líneas de retardo, cada señal de entrada con 3 retardos y las señales de salidas con 3 retardos igualmen-



te. En la red donde se estima la velocidad, se obtuvo la siguiente arquitectura:  $8-6-1$  con 3 retardos para cada señal de entrada y 3 retardos para la señal de salida.

En el caso 2 de igual forma se entrenaron redes para la estimación del flujo del rotor, la posición  $\sin(\theta)$  y  $\cos(\theta)$  y la velocidad del motor. En la Tabla 4.3 se muestran los errores obtenidos de la mejor arquitectura para cada estimación.

**Tabla 4.3:** Errores de los datos de entrenamiento de las redes neuronales del caso 2.

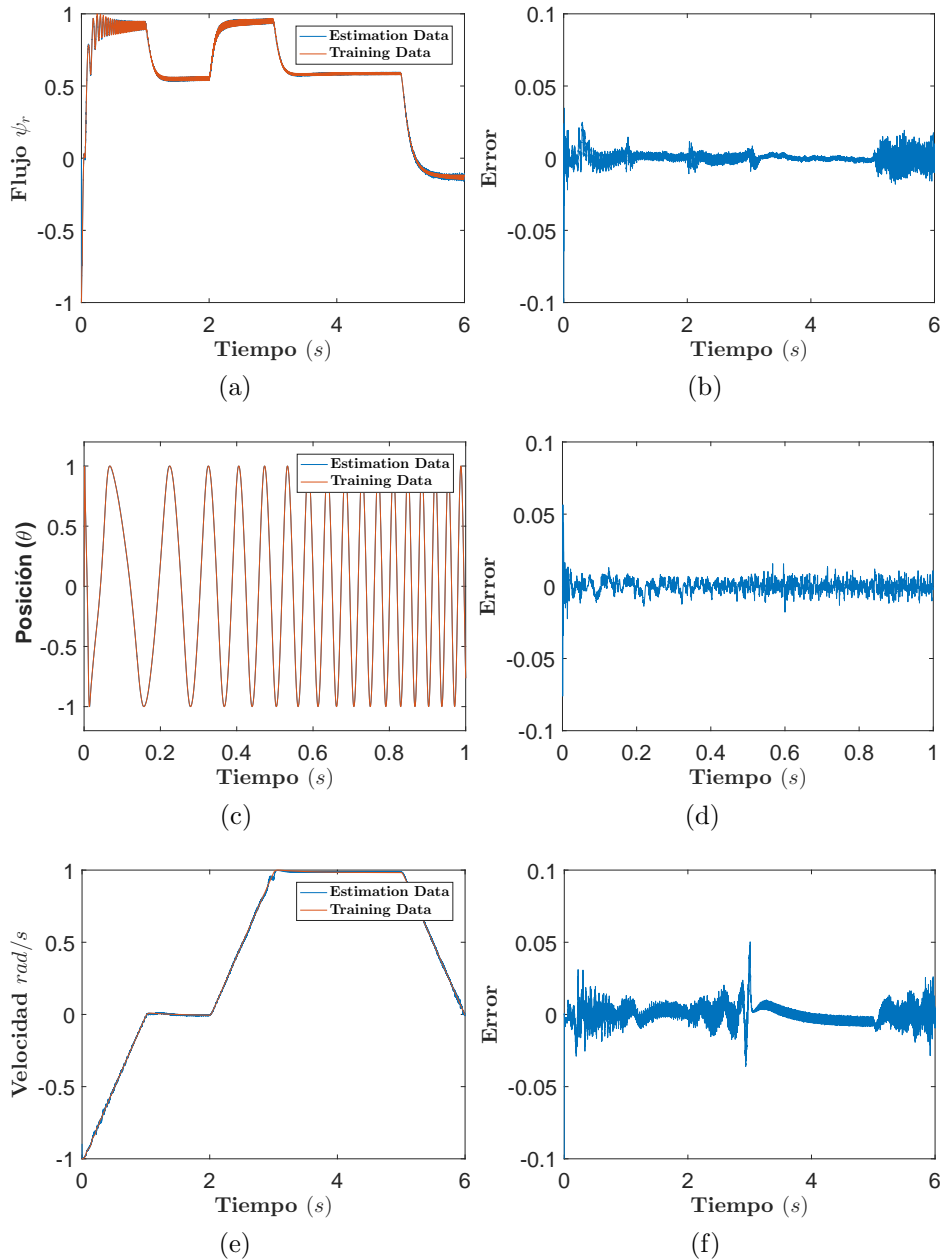
	<b>Flujo del Rotor</b>	<b>Posición del Rotor</b>	<b>Velocidad</b>
<b>Factor de Correlación</b>	0.99999	0.9999	0.9999
<b>Error Medio Absoluto</b>	0.0022	0.0042	0.0044
<b>Error Medio Cuadrático</b>	0.0058	0.0059	0.0062

En la Figura 4.6 se muestra el entrenamiento de las redes que estiman el flujo del rotor, la posición del rotor y la velocidad del motor para el caso 2. Igual en que el caso 1, se puede ver el comportamiento de los datos con los que se entrenó la red y los datos que estimó la red.

Para el flujo del rotor se obtuvo una arquitectura de dos capas ocultas, con un número de neuronas de la siguiente manera:  $9-7-1$ ; al inicio de la red se tienen líneas de retardo, para las señales de entrada y de salida se tienen 6 retardos. En la red neuronal que estima la posición se obtuvo una arquitectura de 3 capas ocultas con el siguiente número de neuronas:  $15-11-9-1$  con 8 retardos para cada señal de entrada y para la señal de salida. Por último, para la red que estima la velocidad se tiene la arquitectura  $10-9-8-1$ .

### 4.2.2. ANFIS

En Matlab se puede generar un sistema de inferencia difuso tipo Sugeno donde se ajustan parámetros usando los datos de entrenamiento de entrada/salida que se usaron de la misma manera en las redes neuronales. La estructura del sistema de inferencia difuso se genera de manera automática usando *grid partitioning*, esto a partir de cierto número de funciones de membresía y el tipo. El ajuste consiste esencialmente en determinar que sistema proporciona la mejor aproximación al formato entrada-salida con el que se entrena el sistema ANFIS. En Matlab existe la instrucción *anfis* para entrenar al sistema. El tipo de entrenamiento puede ser híbrido o

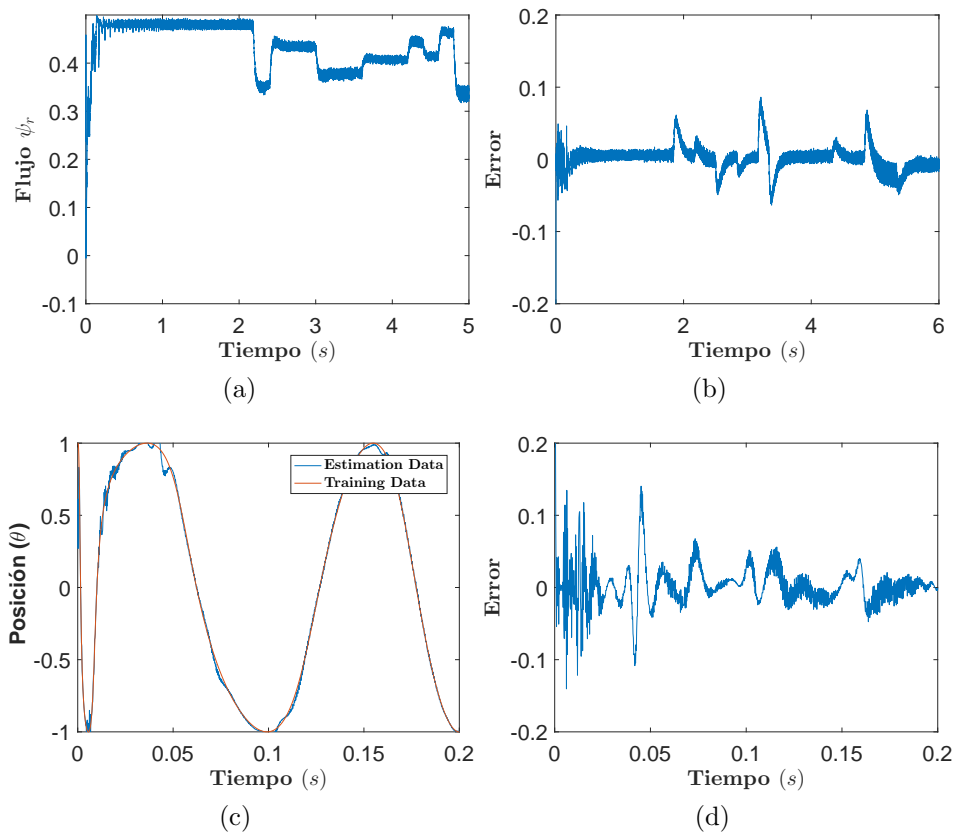


**Figura 4.6:** Resultado del entrenamiento de las redes neuronales para el caso 2 en la estimación del (a) Flujo del rotor; (b) Error del entrenamiento del Flujo; (c) Posición  $\sin(\theta)$ ; (d) Error del entrenamiento de la posición; (e) Velocidad del motor; (f) Error del entrenamiento de la velocidad.

únicamente utilizar *backpropagation*.

Para el caso 1, se obtuvieron arquitecturas para la estimación del flujo del rotor

y la posición del rotor  $\sin(\theta)$  y  $\cos(\theta)$ . Para el estimador de flujo del rotor se contó con 8 funciones de membresía para cada entrada del sistema. Para el estimador de la posición del flujo del rotor se contó con 6 funciones de membresía para cada entrada del sistema. En la Figura 4.7 se ilustra el comportamiento del entrenamiento de los sistemas ANFIS para la estimación del flujo y la posición del rotor del caso 1. En la Tabla 4.4 se muestran los errores obtenidos de la mejor arquitectura para las estimaciones ya mencionadas.



**Figura 4.7:** Resultado del entrenamiento de los sistemas ANFIS para el caso 1 en la estimación del (a) Flujo del rotor; (b) Error del entrenamiento del Flujo; (c) Posición  $\cos(\theta)$ ; (d) Error del entrenamiento de la posición.

Para el caso 2, se obtuvieron arquitectura para la estimación del flujo del rotor y la posición del rotor  $\sin(\theta)$  y  $\cos(\theta)$ . Para el estimador del flujo del rotor se tienen 7 funciones de membresía para cada entrada del sistema. Para el estimador de la posición del flujo del rotor se contó con 5 funciones de membresía para cada entrada del sistema. En la Figura 4.8 se muestra el comportamiento del entrenamiento de

Tabla 4.4: Errores del entrenamiento de los sistemas ANFIS del caso 1.

	Flujo del Rotor	Posición del Rotor
<b>Factor de Correlación</b>	0.9987	0.9989
<b>Error Medio Absoluto</b>	0.0101	0.0135
<b>Error Medio Cuadrático</b>	0.0155	0.0235

los sistemas ANFIS para la estimación del flujo y la posición del rotor del caso 2. En la Tabla 4.5 se muestran los errores obtenidos de las mejor arquitectura para las estimaciones ya mencionadas.

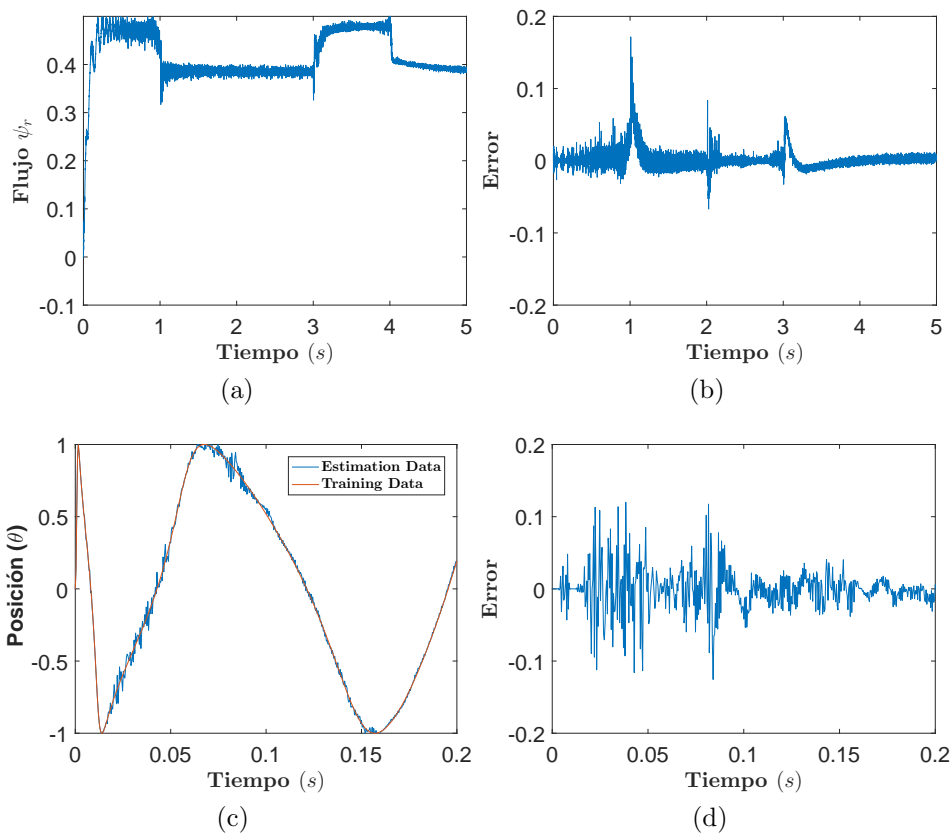


Figura 4.8: Resultado del entrenamiento de los sistemas ANFIS para el caso 2 en la estimación del (a) Flujo del rotor; (b) Error del entrenamiento del Flujo; (c) Posición  $\sin(\theta)$ ; (d) Error del entrenamiento de la posición.

Tabla 4.5: Errores del entrenamiento de los sistemas ANFIS del caso 2.

	<b>Flujo del Rotor</b>	<b>Posición del Rotor</b>
<b>Factor de Correlación</b>	0.9987	0.9971
<b>Error Medio Absoluto</b>	0.0072	0.0078
<b>Error Medio Cuadrático</b>	0.0142	0.0268

### 4.3. Resumen del capítulo

En este capítulo se presentó la metodología para realizar el entrenamiento de las redes neuronales y los sistemas neurodifusos mediante el uso de Matlab y obtener una arquitectura adecuada para la estimación de parámetros de un MI en un esquema de control vectorial. Tales parámetros fueron el flujo del rotor, la posición del flujo del rotor y la velocidad del motor. La configuración de cada arquitectura fue lograda por medio de un razonamiento de tipo heurístico, es decir, realizando pruebas y observando los resultados. El número de neuronas, capas ocultas o funciones de membresía está determinado por la complejidad del problema a resolver. Se observa que las redes neuronales presentaron un mejor comportamiento para la estimación en ambos casos presentados.



## Implementación en *Vivado HLS*

---

En este capítulo se presentan los resultados obtenidos de las redes neuronales y los sistemas neurodifusos para la estimación del flujo del rotor, la posición del flujo del rotor y la velocidad del motor simulados en un FPGA. Se especifica la descripción en *Síntesis de Alto Nivel* y la validación en *System Generator*. Se muestra a detalle el uso de recursos para su posible implementación en FPGA, así como los ciclos de reloj que dura cada red neuronal y cada sistema ANFIS para procesar los datos.

### 5.1. Descripción en *Síntesis de Alto Nivel*

El uso de la Síntesis de Alto Nivel para la descripción de algoritmos es fundamental para el desarrollo de sistemas basados en FPGAs. La definición se basa en la descripción del comportamiento, las restricciones y los objetivos del algoritmo a describir. Al momento de sintetizar se obtiene una estructura de un conjunto de componentes interconectados: registros, multiplexores, buses, etc., donde se puede obtener el *RTL* (Register-Transfer Level) para la simulación con *System Generator/Simulink*. La escritura en Síntesis de Alto Nivel puede llevarse a cabo en *C/C++* o *System C* y su principal objetivo es diseñar, construir y verificar hardware de manera eficiente.

En este trabajo, la descripción de las arquitecturas de las redes neuronales y los sistemas ANFIS se realizó en C++. El formato que se utilizó fue de tipo punto flotante *single precision*. Para las descripciones en Vivado HLS se utilizó un FPGA Kintex-7 XC7K325TFFG900-2 con un reloj que tiene una frecuencia de 100 MHz. Al momento de describir las redes neuronales, fue necesario la obtención de los *pesos* y *bias* para poder almacenarlas en matrices. Es importante mencionar que para la descripción de las redes neuronales, el empleo de recursos depende del número de

neuronas y el número de capas ocultas. En la Figura 5.1 se muestra la comparativa de latencias que necesita cada red neuronal para el caso 1 (descrito en el capítulo anterior) para procesar una iteración; en Vivado HLS existe la posibilidad de optimizar los rendimientos usando directivas en los diseños. En esta circunstancia, se utilizó la directiva *unroll* para la paralelización, lo que ocasiono la disminución de las latencias.

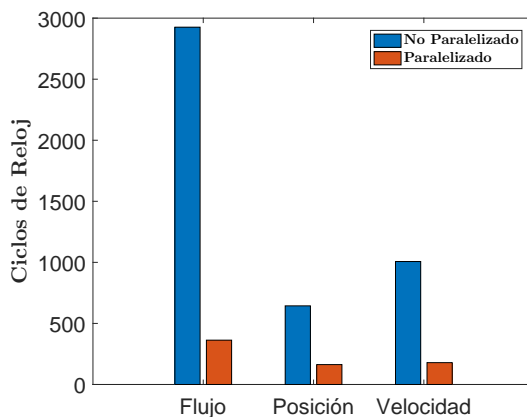


Figura 5.1: Latencia de las redes neuronales del caso 1.

Sin embargo, a pesar de que la paralelización disminuye considerablemente la latencia, el incremento del uso de recursos es considerable. En la Tabla 5.1 se muestran los recursos necesarios para las redes neuronales del caso 1, con paralelización y sin paralelización. Se puede notar que el recurso que se ve mas incrementado es el DSP ya que las redes neuronales forman multiplicadores-acumuladores. La latencia más baja fue la red neuronal que estima la posición del flujo del rotor con 163 ciclos de reloj, es decir,  $1.63 \mu s$ .

Tabla 5.1: Comparativa del uso de recursos de las redes neuronales del caso 1.

Recursos	Utilización					
	Flujo		Posición		Velocidad	
	<i>SP</i>	<i>P</i>	<i>SP</i>	<i>P</i>	<i>SP</i>	<i>P</i>
<b>LUT</b>	5020	18980	3898	8320	3464	10930
<b>Flip-Flop</b>	7298	7298	3398	4847	4847	6568
<b>DSP</b>	15	102	15	54	15	78
<b>BRAM</b>	0	0	0	0	0	0

Para las redes neuronales del caso 2, se procuró entrenar redes neuronales con una



arquitectura que tenga menos capas ocultas y menos neuronas en cada capa, por lo que se ve una disminución en la latencia en la red que estima el flujo del rotor. En la Figura 5.2 se muestran las latencias necesarias para cada red neuronal del caso 2. Aunque en las otras redes que estiman la posición del flujo del rotor y la velocidad del motor no hayan reducido su latencia comparado con las redes del caso 1, al momento de paralelizar, la disminución de ciclos de reloj se ve reflejada de manera considerable.

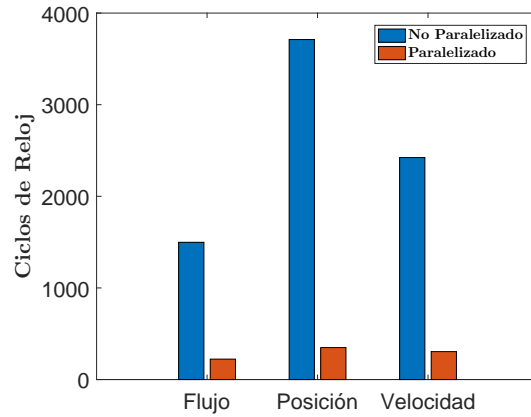


Figura 5.2: Latencia de redes neuronales del caso 2.

En la Tabla 5.2 se muestran los recursos necesarios para las redes del caso 2. Se percibe un incremento considerable en las LUTs debido a que se tiene una mayor cantidad de neuronas en cada capa oculta a comparación del caso 1. La latencia más baja fue la red neuronal que estima el flujo del rotor, con 225 ciclos de reloj, es decir,  $2.25 \mu s$ .

Tabla 5.2: Comparativa del uso de recursos de las redes neuronales del caso 2.

Recursos	Utilización					
	Flujo		Posición		Velocidad	
	<i>SP</i>	<i>P</i>	<i>SP</i>	<i>P</i>	<i>SP</i>	<i>P</i>
<b>LUT</b>	3835	18980	5074	23879	4593	16875
<b>Flip-Flop</b>	4561	7298	7050	11335	6042	8236
<b>DSP</b>	15	86	15	134	15	94
<b>BRAM</b>	0	0	0	0	0	0

Por otro lado, el uso de recursos para los sistemas ANFIS depende del número de funciones membresía por entrada, y que al mismo tiempo, determina el número de

parámetros consecuentes y que también repercuten en el uso de recursos del FPGA. Cabe mencionar, que sólo se logró la estimación del flujo del rotor y la posición del flujo del rotor. En la Figura 5.3 se muestra la comparativa de latencias que necesita cada sistema ANFIS del caso 1. De igual forma que con las redes neuronales, se usaron directivas para optimizar el algoritmo y obtener mejores tiempos de latencia.

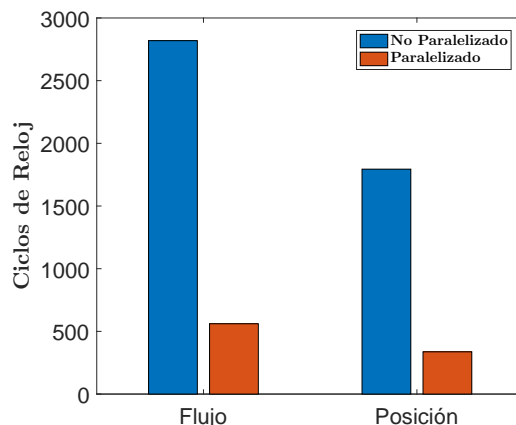


Figura 5.3: Latencia de ANFIS del caso 1.

En la Tabla 5.3 se muestran los recursos necesarios para los sistemas ANFIS del caso 1. Existe un incremento en las LUTs, ya que permiten ahorrar tiempo de procesamiento, porque sacar un valor es mucho más rápido que hacer una gran computación. También existe aumento de Flip-Flops, ya que permite tener una secuencia en el procesamiento de datos. La latencia más baja se tuvo en el sistema que estima la posición del flujo fue de 337 ciclos de reloj, es decir,  $3.37 \mu s$ .

Tabla 5.3: Comparativa del uso de recursos de los sistemas ANFIS del caso 1.

Recursos	Utilización			
	Flujo		Posición	
	<i>SP</i>	<i>P</i>	<i>SP</i>	<i>P</i>
<b>LUT</b>	6050	60988	11711	100014
<b>Flip-Flop</b>	3989	37129	7735	65244
<b>DSP</b>	50	226	100	500
<b>BRAM</b>	6	0	12	0

Para los sistemas ANFIS del caso 2 se aprecia un crecimiento en la latencia. En la Figura 5.4 se ilustra un incremento considerable comparado con los otros sistemas

ANFIS, inclusive con las redes neuronales. Para realizar la optimización en el rendimiento, se tuvo que disminuir el reloj del FPGA. Para el estimador del flujo del rotor se optó por un reloj de 80 MHz y para el estimador de la posición del flujo del rotor se optó por un reloj funcionando a 90 MHz.

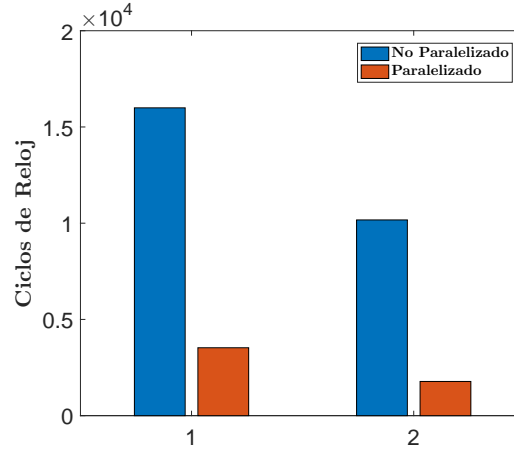


Figura 5.4: Latencia de ANFIS del caso 2.

No obstante, el uso de recursos creció de tal manera, que no podría ser implementado junto a un esquema de control. En la Tabla 5.4 se muestra la utilización de recursos sin la optimización y con el paralelizamiento. Es importante mencionar que al incrementar los recursos, se dieron los problemas de latencia y por lo tanto, se tuvo que disminuir la frecuencia de operación de estos sistemas ANFIS.

Tabla 5.4: Comparativa del uso de recursos de los sistemas ANFIS del caso 2.

Recursos	Utilización			
	Flujo		Posición	
	<i>SP</i>	<i>P</i>	<i>SP</i>	<i>P</i>
<b>LUT</b>	8943	64879	8914	133603
<b>Flip-Flop</b>	5820	20762	5805	75244
<b>DSP</b>	75	300	75	375
<b>BRAM</b>	7	7	7	4

## 5.2. Validación en *System Generator*

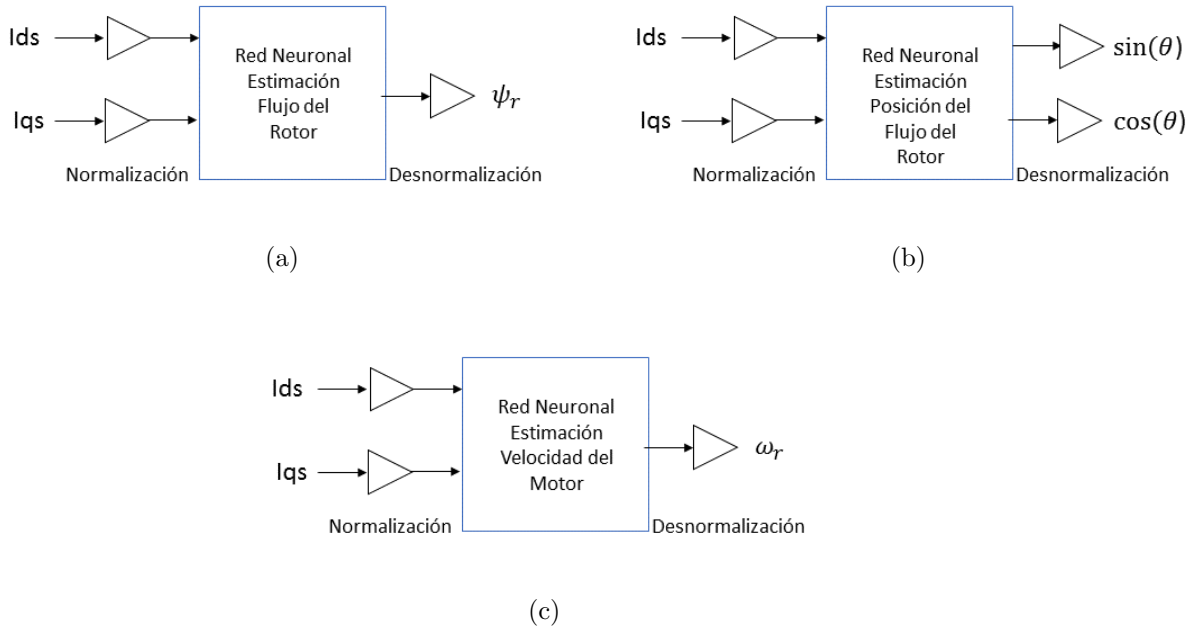
Como se mencionó, *System Generator* es una herramienta de diseño que permite el modelado en Simulink para el diseño en FPGAs. El diseño se captura en el entorno de Simulink utilizando un conjunto de bloques de *Xilinx*. En este trabajo, se utilizó esta herramienta con el fin de explorar las arquitecturas de las redes neuronales y los sistemas ANFIS sin el uso de trasladar el diseño a hardware. Lo grandioso de la herramienta es que provee abstracciones de alto nivel que son complicadas automáticamente en un FPGA.

En la Figura 5.5 se muestran la estructuras de las redes neuronales del caso 1 emuladas en *System Generator* que representan los estimadores como parte del conjunto del control vectorial implementado en Simulink descrito en el Capítulo 4. Las entradas para cada estimador son las corrientes  $I_{ds}$  e  $I_{qs}$ . De la misma manera, en los sistemas ANFIS se utilizaron las mismas entradas, las corrientes  $I_{ds}$  e  $I_{qs}$ . En la Figura 5.6 se muestran las estructuras de los sistemas ANFIS. En la Figura 5.7 se muestra las respuestas de los esquemas de las redes neuronales y los sistemas neurodifusos del caso 1.

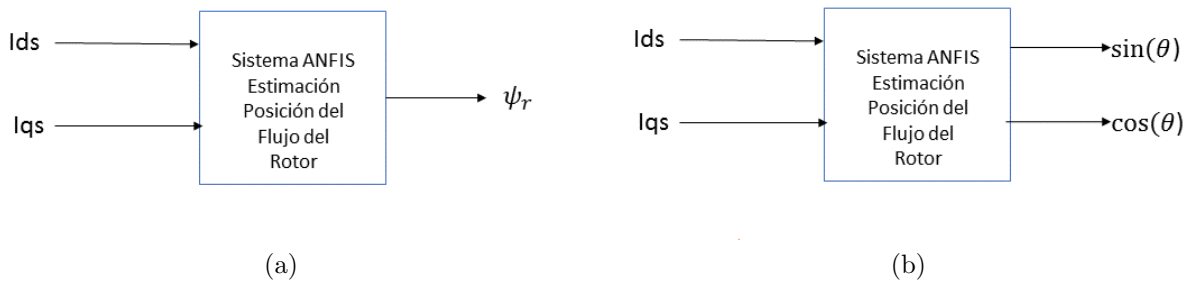
En la Figura 5.8 se muestran las estructuras de las redes neuronales del caso 2 emuladas en *System Generator*. En el estimador de la velocidad del motor se puede apreciar una entrada más. Las entradas para cada estimador son las corrientes  $I_{ds}$  e  $I_{qs}$ , además del flujo  $\psi_r$ . En la Figura 5.9 se tienen las mismas entradas de corriente que en las redes neuronales más una entrada extra. Las entradas fueron: la velocidad del motor y el flujo del rotor, respectivamente. En la Figura 5.10 se muestra las respuestas de los esquemas de las redes neuronales y los sistemas neurodifusos del caso 2.

## 5.3. Resumen del capítulo

En este capítulo se presentó la descripción de las redes neuronales y los sistemas neurodifusos en Síntesis de Alto Nivel, con la ayuda de *Vivado HLS*. La descripción de los algoritmos se llevó a cabo en C++. Se presentó el uso de recursos de cada red neuronal y cada sistema neurodifuso. También se muestra los ciclos de reloj que dura cada proceso y la paralelización para disminuirlos, funcionando con un reloj a una frecuencia de 100 MHz. Finalmente se realizó la validación en *System Generator*,

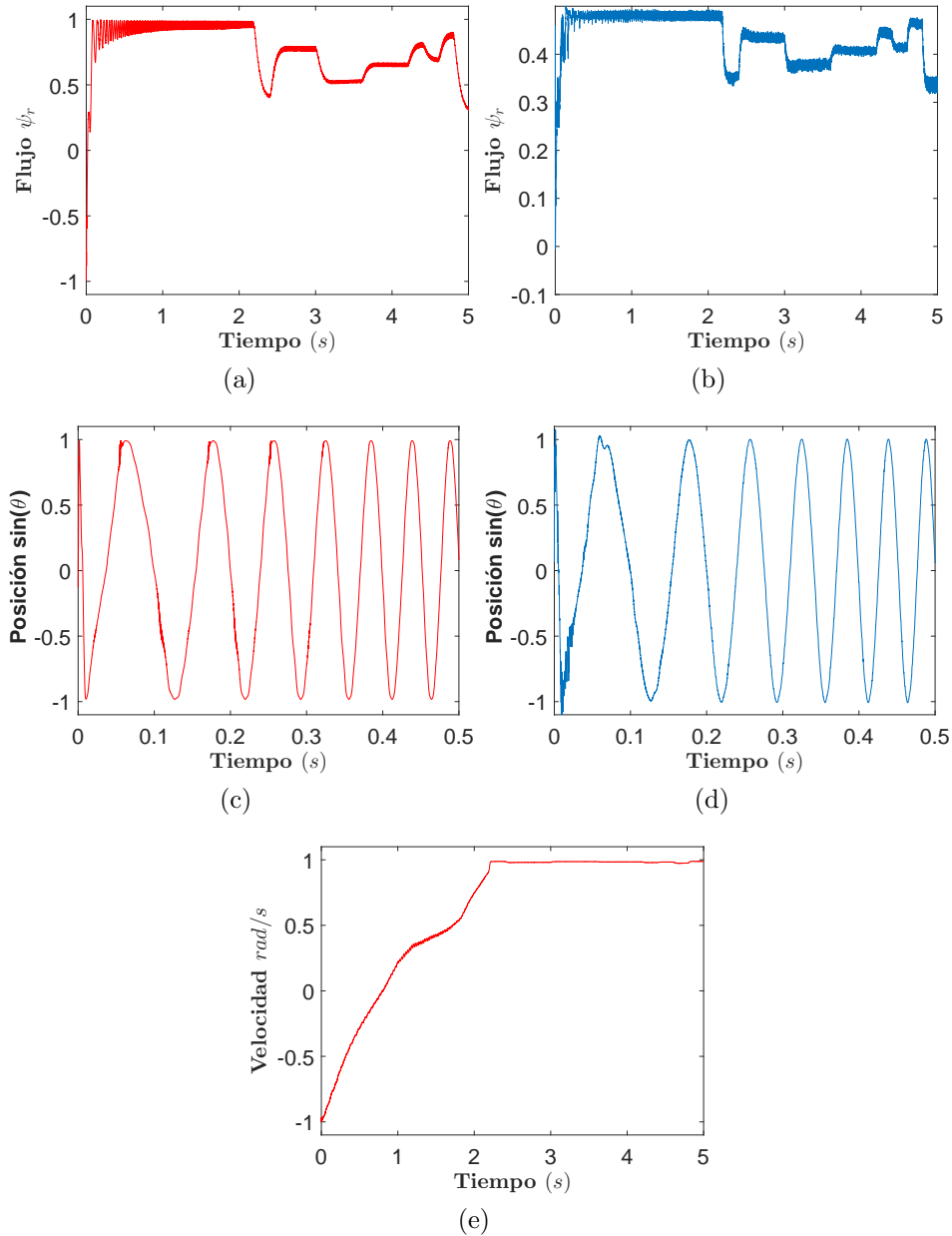


**Figura 5.5:** Esquemas de las redes neuronales caso 1 para estimación del (a) Flujo del rotor; (b) Posición del flujo del rotor; (c) Velocidad del motor.

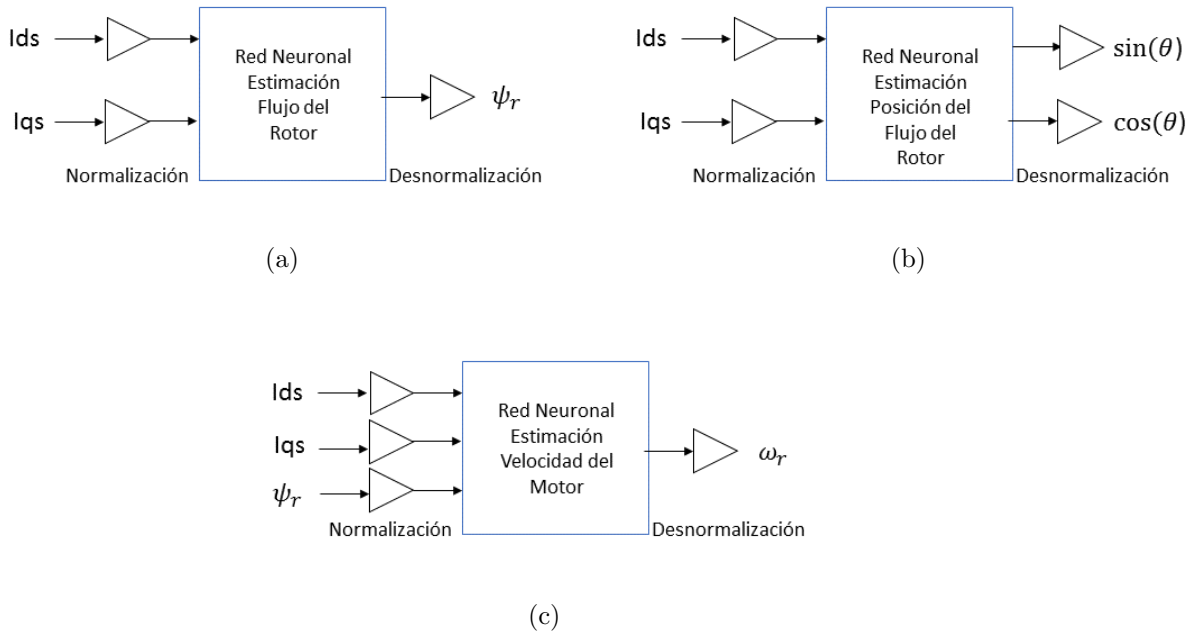


**Figura 5.6:** Esquemas de los sistemas ANFIS caso 1 para estimación del (a) Flujo del rotor; (b) Posición del flujo del rotor.

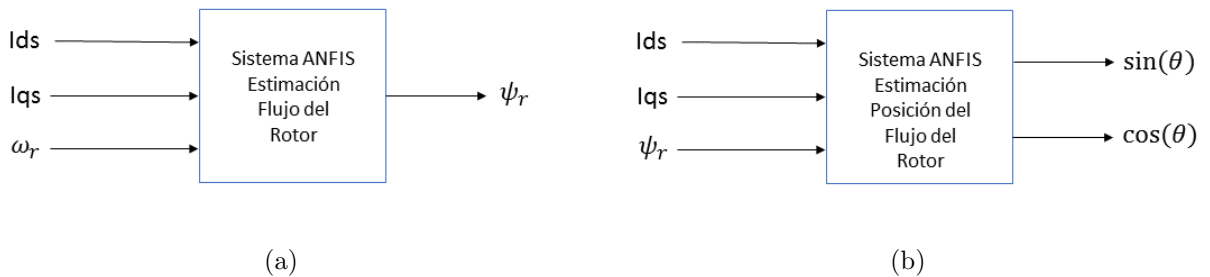
permitiendo la exploración de las arquitecturas con el esquema de control sin el uso de trasladar el diseño a hardware.



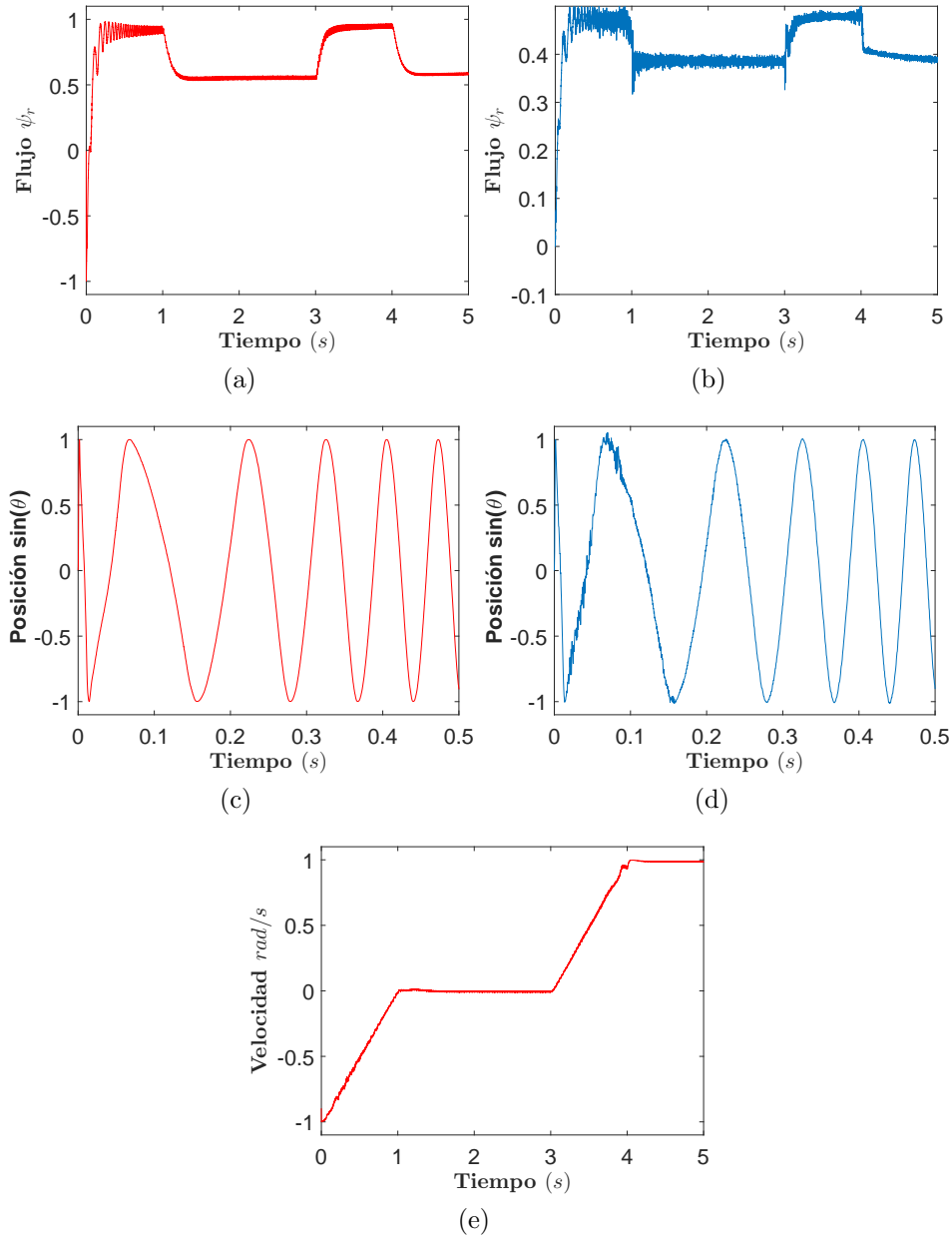
**Figura 5.7:** Respuesta de los esquemas de las redes neuronales y los sistemas neurodifusos caso 1 para estimación del (a) Flujo del rotor con redes neuronales; (b) Flujo del rotor con ANFIS; (c) Posición del flujo del rotor con redes neuronales; (d) Posición del flujo del rotor con ANFIS; (e) Velocidad del motor con redes neuronales.



**Figura 5.8:** Esquemas de las redes neuronales caso 2 para estimación del (a) Flujo del rotor; (b) Posición del flujo del rotor; (c) Velocidad del motor.



**Figura 5.9:** Esquemas de las redes neuronales caso 2 para estimación del (a) Flujo del rotor; (b) Posición del flujo del rotor.



**Figura 5.10:** Respuesta de los esquemas de las redes neuronales y los sistemas neurodifusos del caso 2 para la estimación del (a) Flujo del rotor con redes neuronales; (b) Flujo del rotor con ANFIS; (c) Posición del flujo del rotor con redes neuronales; (d) Posición del flujo del rotor con ANFIS; (e) Velocidad del motor con redes neuronales.



# Conclusiones y trabajo a futuro

---

## 6.1. Conclusiones

El uso de técnicas *inteligentes* como estimadores de parámetros en un MI para un esquema de control vectorial son una herramienta versátil dada su robustez frente a cambios, como la variación en el par de carga y la velocidad de referencia. Se aprecia una ventaja al usar estas técnicas debido a que no se requiere un modelo matemático del motor orientado al flujo del rotor. A partir de las simulaciones del esquema de control vectorial implementado en Simulink, se obtuvieron la base de datos para el entrenamiento de las redes neuronales y los sistemas neurodifusos. Sin embargo, la búsqueda de arquitecturas para los estimadores con redes neuronales y los sistemas neurodifusos es un proceso que conlleva tiempo, debido a los procesos de entrenamiento. A pesar de ello, al describir las arquitecturas en Síntesis de Alto Nivel, se puede encontrar una optimización de rendimiento y de área para cada una de ellas para poder implementar en FPGA. Con las redes neuronales que tienen una arquitectura con más neuronas y capas ocultas se observó una utilización de recursos mayor a las demás. Lo mismo ocurrió con los sistemas neurodifusos al aumentar el número de funciones membresía, lo que causó una repercusión en los recursos y la latencia. Con cada red neuronal se logró una operación de menos de 500 ciclos de reloj, es decir, a  $5\mu\text{s}$  con un reloj operando a 100 MHz. Por otro lado, los sistemas neurodifusos presentan una mayor operación en ciclos de reloj, la más baja de 561 ciclos de reloj, es decir, a  $7.012\mu\text{s}$  con un reloj operando a 80 MHz. Cabe mencionar, que la validación en *System Generator* ayudó a explorar el funcionamiento de las arquitecturas sin trasladar el diseño a hardware y visualizar que hubo un mejor comportamiento con las redes neuronales que con los sistemas neurodifusos.

## 6.2. Trabajo a futuro

- Implementar las redes neuronales y los sistemas neurodifusos obtenidos en este trabajo en un FPGA, comprobando resultados por medio de cosimulación con Simulink.
- Implementar por completo el esquema de control vectorial en *VHDL* o en Síntesis de Alto Nivel para el FPGA.
- Investigar otros esquemas de control como el DTC *Direct Torque Control* con la estimación de parámetros usando redes neuronales y sistemas neurodifusos.
- Investigar otras técnicas de estimación de parámetros para un MI como lo pueden ser los filtros Kalman o los algoritmos genéticos.

# Bibliografía

---

- [1] S. J. Chapman, *Máquinas eléctricas (5a.* McGraw Hill Mexico, 2012.
- [2] P. C. Krause, O. Wasynczuk, S. D. Sudhoff, and S. Pekarek, *Analysis of electric machinery and drive systems*, vol. 75. John Wiley & Sons, 2013.
- [3] J. Gutierrez-Villalobos, J. Rodríguez-Reséndiz, E. A. Rivas-Araiza, and V. Mucino, “A review of parameter estimators and controllers for induction motors based on artificial neural networks,” *Neurocomputing*, vol. 118, pp. 87–100, 2013.
- [4] D. Neema, R. Patel, and A. Thoke, “Rotor flux and torque estimator for vector controlled induction drive using ann,” in *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, pp. 2215–2220, IEEE, 2009.
- [5] S. S. Haykin, S. S. Haykin, S. S. Haykin, and S. S. Haykin, *Neural networks and learning machines*, vol. 3. Pearson Upper Saddle River, NJ, USA:, 2009.
- [6] B. Karanayil, *Parameter identification for induction motor drives using artificial neural networks and fuzzy logic*. VDM Verlag Dr. Müller, 2009.
- [7] M. H. Rashid, M. H. R. V. González, P. A. S. Fernández, V. González, *et al.*, *Electrónica de potencia: circuitos, dispositivos y aplicaciones*. Pearson Educación, 2004.
- [8] P. Girovsky, J. Timko, and J. Zilkova, “Shaft sensor-less foc control of an induction motor using neural estimators,” *Acta Polytechnica Hungarica*, vol. 9, no. 4, pp. 31–45, 2012.
- [9] C. Morales-Perez, J. Rangel-Magdaleno, H. Peregrina-Barreto, J. Ramirez-Cortes, and I. Cruz-Vega, “Fpga-based broken bar detection on im using

- omp algorithm,” in *Instrumentation and Measurement Technology Conference (I2MTC), 2017 IEEE International*, pp. 1–6, IEEE, 2017.
- [10] J. de Jesus Rangel-Magdaleno, H. Peregrina-Barreto, J. M. Ramirez-Cortes, P. Gomez-Gil, and R. Morales-Caporal, “Fpga-based broken bars detection on induction motors under different load using motor current signature analysis and mathematical morphology,” *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 5, pp. 1032–1040, 2014.
- [11] C.-H. Tsai, “Neural network application for flux and speed estimation in the sensorless decoupling induction motor drive,” in *Systems, Man and Cybernetics, 2006. SMC’06. IEEE International Conference on*, vol. 6, pp. 5297–5303, IEEE, 2006.
- [12] P. Vas, *Sensorless vector and direct torque control*. Oxford Univ. Press, 1998.
- [13] A. K. Toh, E. P. Nowicki, and F. Ashrafzadeh, “A flux estimator for field oriented control of an induction motor using an artificial neural network,” in *Industry Applications Society Annual Meeting, 1994., Conference Record of the 1994 IEEE*, vol. 1, pp. 585–592, IEEE, 1994.
- [14] T. Orłowska-Kowalska and C. Kowalski, “Neural network application for flux and speed estimation in the sensorless induction motor drive,” in *Industrial Electronics, 1997. ISIE’97., Proceedings of the IEEE International Symposium on*, pp. 1253–1258, IEEE, 1997.
- [15] S. V. Hule, R. Bindu, and D. Vincent, “Sensorless vector control of three phase induction motor,” in *Advances in Communication and Computing Technologies (ICACACT), 2014 International Conference on*, pp. 1–6, IEEE, 2014.
- [16] B. S. Guru, H. R. Hiziroglu, and J. E. Brito, *Máquinas eléctricas y transformadores*. Oxford University Press, 2003.
- [17] A. E. Fitzgerald, U. Charles, S. D. Y. Milanés, N. S. Jorge, *et al.*, *Electric machinery. Máquinas eléctricas*. 1992.
- [18] Y. Oguz and M. Dede, “Speed estimation of vector controlled squirrel cage asynchronous motor with artificial neural networks,” *Energy Conversion and Management*, vol. 52, no. 1, pp. 675–686, 2011.

- [19] B. K. Bose and B. K. Bose, *Power electronics and variable frequency drives: technology and applications*. Wiley Online Library, 1997.
- [20] H. A. Toliyat, E. Levi, and M. Raina, “A review of rfo induction motor parameter estimation techniques,” *IEEE transactions on Energy conversion*, vol. 18, no. 2, pp. 271–283, 2003.
- [21] M. Sumner and G. Asher, “Self-commissioning for voltage-referenced voltage fed vector controlled induction motor drives,” in *Power Electronics Specialists Conference, 1992. PESC’92 Record., 23rd Annual IEEE*, pp. 139–144, IEEE, 1992.
- [22] M. Ruff, A. Bunte, and H. Grotstollen, “A new self-commissioning scheme for an asynchronous motor drive system,” in *Industry Applications Society Annual Meeting, 1994., Conference Record of the 1994 IEEE*, vol. 1, pp. 616–623, IEEE, 1994.
- [23] A. Khambadkone and J. Holtz, “Vector controlled induction motor drive with a self-commissioning scheme,” in *Industrial Electronics Society, 1990. IECON’90., 16th Annual Conference of IEEE*, pp. 927–932, IEEE, 1990.
- [24] P. Chrzan and H. Klaassen, “Parameter identification of vector-controlled induction machines,” *Electrical Engineering (Archiv fur Elektrotechnik)*, vol. 79, no. 1, pp. 39–46, 1996.
- [25] S.-I. Moon and A. Keyhani, “Estimation of induction machine parameters from standstill time-domain data,” *IEEE Transactions on Industry Applications*, vol. 30, no. 6, pp. 1609–1615, 1994.
- [26] A. Bunte and H. Grotstollen, “Parameter identification of an inverter-fed induction motor at standstill with a correlation method,” in *Power Electronics and Applications, 1993., Fifth European Conference on*, pp. 97–102, IET, 1993.
- [27] N. R. Klaes, “Parameter identification of an induction machine with regard to dependencies on saturation,” *IEEE Transactions on Industry Applications*, vol. 29, no. 6, pp. 1135–1140, 1993.

- [28] A. Bünte and H. Grotstollen, “Offline parameter identification of an inverter-fed induction motor at standstill,” in *Proc. Europe. Conf. Power Electron. Applicat.*, pp. 3–492, 1995.
- [29] E. CERRUTO, A. CONSOLI, A. RACITI, and A. TESTA, “Slip gain tuning in indirect field oriented control drives,” *Electric machines and power systems*, vol. 23, no. 1, pp. 63–79, 1995.
- [30] J. Cilia, G. Asher, J. Shuli, M. Sumner, K. Bradley, and A. Ferrah, “The recursive maximum likelihood algorithm for tuning the rotor time constant in high-performance sensorless induction motor drives,” in *Proc. Int. Conf. Elect. Mach.*, pp. 926–930, 1998.
- [31] R. Beguenane and G. Capolino, “Induction motor rotor time constant measurement for vector control drives without rotary transducer,” in *Proc. IEEE Int. Symp. Power Eng. Power Technol.*, vol. 3, pp. 13–17, 1995.
- [32] A. Dell’Aquila, S. Papa, L. Salvatore, and S. Stasi, “A delayed state kalman filter for on-line estimation of induction motor parameters and rotor flux space vector position,” in *Electrotechnical Conference, 1996. MELECON’96., 8th Mediterranean*, vol. 1, pp. 269–273, IEEE, 1996.
- [33] S. Wade, M. Dunnigan, and B. Williams, “Improvements for induction machine vector control,” in *EUROPEAN CONFERENCE ON POWER ELECTRONICS AND APPLICATIONS*, vol. 1, pp. 1–542, PROCEEDINGS PUBLISHED BY VARIOUS PUBLISHERS, 1995.
- [34] R. Dadkhah, H. Givi, and A. Mehdipour, “Parameter estimation of the induction motor using extended kalman filter for wide range speed control,” in *Power Electronics, Drives Systems & Technologies Conference (PEDSTC), 2015 6th*, pp. 137–142, IEEE, 2015.
- [35] T. A. Lipo, *Vector control and dynamics of AC drives*, vol. 41. Oxford university press, 1996.
- [36] R. Krishnan, *Electric motor drives: modeling, analysis, and control*. Prentice Hall, 2001.

- [37] M. Kasmierkowski and H. Tunia, "Automatic control of converter fed drives," *ELECTRONIC ENGINEERING*, vol. 4, p. 6, 1994.
- [38] E. Levi, "Magnetic variables control," *Wiley Encyclopedia of Electrical and Electronics Engineering*, 1999.
- [39] R. Krishnan and A. S. Bharadwaj, "A review of parameter sensitivity and adaptation in indirect vector controlled induction motor drive systems," *IEEE Transactions on Power Electronics*, vol. 6, no. 4, pp. 695–703, 1991.
- [40] W. Dazhi, G. Shusheng, and W. Kenan, "A neural-network-base adaptive estimator for speed-sensorless control of induction motor," in *Intelligent Control and Automation, 2002. Proceedings of the 4th World Congress on*, vol. 4, pp. 2812–2816, IEEE, 2002.
- [41] X. Fu and S. Li, "A novel neural network vector control technique for induction motor drive," *IEEE Transactions on Energy Conversion*, vol. 30, no. 4, pp. 1428–1437, 2015.
- [42] F. G. Areed, A. Y. Haikal, and R. H. Mohammed, "Adaptive neuro-fuzzy control of an induction motor," *Ain Shams Engineering Journal*, vol. 1, no. 1, pp. 71–78, 2010.
- [43] D. Neema and P. Choudhary, "An ann based rotor flux estimator for vector controlled induction motor drive,"
- [44] G. Acevedo, N. Vargas, G. M. Hernando, C. Torres, J. Jairo, *et al.*, "Design of rotor flux oriented vector control systems for induction motor," in *Power Electronics and Motion Control Conference (IPEMC), 2012 7th International*, vol. 2, pp. 1384–1388, IEEE, 2012.
- [45] S. Hussain and M. A. Bazaz, "Neural predictive observer for sensorless-controlled induction motor drive," *Control and Intelligent Systems*, vol. 45, no. 2, 2017.
- [46] P. Brandstetter and M. Kuchar, "Rotor flux estimation using voltage model of induction motor," in *Electric Power Engineering (EPE), 2015 16th International Scientific Conference on*, pp. 246–250, IEEE, 2015.

- 
- [47] A. K. Toh, E. P. Nowicki, and F. Ashrafzadeh, “A flux estimator for field oriented control of an induction motor using an artificial neural network,” in *Industry Applications Society Annual Meeting, 1994., Conference Record of the 1994 IEEE*, vol. 1, pp. 585–592, IEEE, 1994.
- [48] J.-S. Jang, “Anfis: adaptive-network-based fuzzy inference system,” *IEEE transactions on systems, man, and cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.
- [49] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, “Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence,” 1997.