



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Análisis de un controlador difuso para el  
control de velocidad de un motor de  
corriente directa sin escobillas**

**TESIS**

Que para obtener el título de

**Ingeniero Eléctrico Electrónico**

**P R E S E N T A N**

Uvaldo Alonso Becerril Vega

Santiago Cruz Monjaraz

**DIRECTOR DE TESIS**

Ing. Francisco José Rodríguez Ramírez



Ciudad Universitaria, Cd. Mx., 2019



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

*Dedicatoria*

Le agradecemos profundamente al Dr. Jaime Guerra Pulido por sus consejos, paciencia y confianza.

Agradecemos a nuestras familias por su incansable labor en nuestra formación profesional y humana.

## ÍNDICE GENERAL

---

1	ANTECEDENTES	1
1.1	Planteamiento del problema . . . . .	1
1.2	Objetivo . . . . .	2
1.3	Introducción y Estado del Arte . . . . .	2
1.4	Motores de CD . . . . .	3
1.5	Motores Síncronos . . . . .	7
1.6	Motor de Inducción . . . . .	8
1.7	Motores BLDC . . . . .	11
1.8	Control con Sensores . . . . .	13
1.9	Control sin Sensores . . . . .	14
1.10	Sistemas de Arranque . . . . .	17
1.11	Control de Campo Orientado (FOC por sus siglas en inglés) . . . . .	19
1.12	Estimador FAST . . . . .	27
1.13	Lógica Difusa . . . . .	29
1.14	Control Difuso . . . . .	34
1.15	Ajuste de Parámetros . . . . .	36
2	METODOLOGÍA	39
2.1	Herramientas empleadas . . . . .	39
2.2	Caracterización del Motor . . . . .	42
2.2.1	Pruebas al sistema caracterizado con el control (PI) empleado por InstaSpin FOC . . . . .	45
2.3	Estudio del Control (PI) . . . . .	45
2.3.1	Diseño del Control Difuso . . . . .	50
2.4	Simulación con el Control Diseñado en Matlab . . . . .	56
2.5	Implementación del Control Difuso . . . . .	59
3	RESULTADOS Y CONCLUSIONES	65
3.1	Resultados de la Simulación - Difuso vs PI . . . . .	65
3.2	Resultados de la implementación - Difuso vs PI . . . . .	66
3.3	Control Difuso Adaptativo . . . . .	70
3.4	Conclusiones . . . . .	70
A	APÉNDICE	73

## ÍNDICE DE FIGURAS

---

Figura 1	Corte transversal de una máquina de corriente directa con embobinado de campo. . . . .	3
Figura 2	Circuito equivalente de un motor de CD con imanes permanentes . . . . .	4
Figura 3	Posibles conexiones del embobinado de campo de un motor de CD . . . . .	5
Figura 4	Curvas características de Torque-Velocidad de tres motores de CD con diferente conexión del embobinado de campo . . . . .	6
Figura 5	Máquina síncrona trifásica de dos polos . . . . .	7
Figura 6	Circuito equivalente de un motor síncrono . . . . .	8
Figura 7	Rotor de un motor de inducción de jaula de ardilla . . . . .	9
Figura 8	Curva característica Torque-Velocidad de un motor de inducción trifásico típico . . . . .	10
Figura 9	Circuito equivalente de un motor de inducción . . . . .	10
Figura 10	Corte transversal de un motor de CA con rotor de imanes permanentes. . . . .	11
Figura 11	a) Embobinados del estator por fase conectados en estrella (BLDC) b) Gráfica de excitación por fase para un ciclo mecánico completo . . . . .	12
Figura 12	a) Conexión en estrella de los embobinados de cada fase b) Construcción de un motor BLDC (corte transversal) . . . . .	12
Figura 13	Sección Longitudinal de un BLDC con sensores Hall montados sobre el eje. . . . .	14
Figura 14	Oscilograma de la FCE de cada una de las fases (cada seis segmentos representan un ciclo de conmutación completo: 360° eléctricos) . . . . .	15
Figura 15	Oscilograma de la señal de la fuerza contraelectromotriz pasada por un CAD (Convertidor Analógico Digital) . . . . .	16
Figura 16	Oscilograma de la señal de la fuerza contraelectromotriz real e ideal . . . . .	17
Figura 17	Diagrama de método de arranque sin sensores . . . . .	18
Figura 18	Modelo de un motor de CD de excitación separada (imanes permanentes) . . . . .	19
Figura 19	Esquema transversal de un motor BLDC con los ejes por fase y el vector de flujo del rotor . . . . .	21

Figura 20	Diagrama vectorial del vector espacial de corriente resultante a partir de las corrientes de fase . . . . .	22
Figura 21	Relación ángulo-torque de un motor BLDC . . . . .	22
Figura 22	Diagrama Vectorial del vector espacial de corriente resultante medido y esperado . . . . .	23
Figura 23	Diagrama vectorial del vector espacial de corriente a partir de las corrientes de fase . . . . .	24
Figura 24	Diagrama vectorial de la transformada directa de Park . . . . .	25
Figura 25	Esquema de las variables de error resultante a partir de los valores medidos y esperados . . . . .	26
Figura 26	Esquema de las variables de control resultantes al aplicarse las transformaciones sobre las corrientes de fase . . . . .	26
Figura 27	Esquema de las variables de control resultantes al aplicarse las transformaciones sobre las corrientes de fase . . . . .	27
Figura 28	El esquema FOC en diagrama de bloques . . . . .	28
Figura 29	Esquema básico de un observador . . . . .	29
Figura 30	Esquema de entradas/salidas del estimador FAST . . . . .	30
Figura 31	Experimento de Mamdani - Esquemático del sistema . . . . .	31
Figura 32	Ejemplos gráficos de funciones de pertenencia en un sistema difuso para la variable temperatura . . . . .	32
Figura 33	Esquema en bloques de un sistema difuso de dos entradas y una salida . . . . .	33
Figura 34	Esquema básico de un control difuso en un sistema realimentado . . . . .	35
Figura 35	Sistema básico de control realimentado . . . . .	40
Figura 36	Fotografía del estator del BLDC empleado en este trabajo . . . . .	42
Figura 37	Fotografía del sistema completo: Fuente de alimentación de CD, microcontrolador, driver, y el motor BLDC . . . . .	43
Figura 38	Ventana de aplicación MotorWare al cargar exitosamente a la memoria de la tarjeta el programa <i>InstaSpin-FOC</i> . . . . .	43
Figura 39	Casillas habilitadas para el cálculo de los parámetros del motor . . . . .	44
Figura 40	Ventana de Motorware dónde se muestran los parámetros del motor calculados. . . . .	44
Figura 41	Ventana con la librería del control PI cuyas variables de control se despliegan en la sub-ventana izquierda. . . . .	45

Figura 42	Todos los bloques del FOC incluidos en el módulo de control. . . . .	46
Figura 43	Diagrama de bloques del sistema de control difuso de un BLDC visto en Simulink (Matlab) .	57
Figura 44	Cuatro oscilogramas que muestran i) la corriente de estator, ii) la velocidad del rotor, iii) el torque electromagnético y iv) el voltaje de CD de bus. Se puede observar que la velocidad llega a la referencia (500 rpm) aproximadamente en 500 milisegundos. . . . .	58
Figura 45	Acercamiento de la <a href="#">Figura 46</a> . Se muestran las mismas variables pero se puede ver con mayor claridad la forma de las señales. . . . .	58
Figura 46	Acercamiento del oscilograma de cuatro variables al modificarse las reglas de pertenencia del primer sistema propuesto. El oscilograma es producto de la concentración de las reglas de pertenencia alrededor del origen. . . . .	59
Figura 47	Oscilogramas de cuatro variables del motor; i) la corriente de estator, ii) la velocidad del rotor, iii) el torque electromagnético y iv) el voltaje de CD de bus (de arriba hacia abajo) al implementarse el sistema con un control PI. . . . .	65
Figura 48	Resultados de la implementación. La figura muestra los valores de la referencia y de la velocidad medida a lo largo de 165 muestras. Durante este ejercicio se probaron 4 diferentes velocidades de referencia. . . . .	66
Figura 49	Sistema con control difuso 1 (anteriormente detallado) y constante de integración de 0.01 . . . . .	67
Figura 50	Sistema con control difuso 2 (anteriormente detallado) y constante de integración de 0.01 . . . . .	67
Figura 51	Sistema con control difuso 2 y constante de integración de 1. . . . .	68
Figura 52	Sistema con control difuso 2 y constante de integración de 100. . . . .	68
Figura 53	Sistema con control difuso 2 y constante de integración de 100, con el error multiplicado por dos. La técnica de multiplicar el error provoca un peso más fuerte, por lo que el sistema actúa de forma más rápida, ideal para diferencias muy grandes. . . . .	69
Figura 54	Sistema con control difuso 2 y constante de integración de 0.01, con el error multiplicado por dos. . . . .	69

Figura 55	Sistema con control difuso 2 y constante de integración de 0.0008, con el error multiplicado por dos. . . . .	70
Figura 56	Control difuso adaptativo, aplicando diferentes valores a la constante de integración dependiendo de la velocidad de referencia. . . . .	71

## ÍNDICE DE CUADROS

---

Cuadro 1	Conjunto de Reglas . . . . .	55
Cuadro 2	Tabla de 5x5 con la jerarquización de las reglas con base en los valores que toman las variables Error y $\frac{dError}{dt}$ . . . . .	56

## ANTECEDENTES

---

### 1.1 PLANTEAMIENTO DEL PROBLEMA

La industria de los vehículos aéreos no tripulados (UAV por sus siglas en inglés), o mejor conocidos cotidianamente como *drones*, ha tenido un gran auge durante la última década. Esta industria depende en gran medida del uso de los motores de corriente directa sin escobillas (BLDC por sus siglas en inglés). Por un lado, estos motores son empleados en la construcción de automóviles híbridos (Rippel, 2007), pero es en la industria de los UAVs donde tienen su mayor aplicación. Se estima que la industria de los drones supere los 12 mil millones de dólares para el año 2021, creciendo a una tasa compuesta de 7.6 puntos porcentuales anual desde 2016, además se espera que el crecimiento ocurra dentro de los tres segmentos de la industria: consumo, empresarial y militar (BI, 2018). Existen otras industrias que han adoptado el uso de BLDCs debido a sus características, como son: la de climatización y de aplicaciones residenciales (Durfee, 2015).

Debido a lo anterior, en el contexto de la industria de drones, el estudio y más en concreto, el control de los motores BLDC se ha vuelto un tema de mucho interés y de amplia relevancia comercial. Este trabajo a su vez propone una implementación de un controlador de bajo costo económico que use la lógica difusa para controlar la velocidad del motor a través del control difuso. Se buscará demostrar, en la medida de lo posible, la utilidad del control difuso en problemas de control actuales con aplicación en la industria y su implementación correspondiente usando componentes de alta disponibilidad en el mercado.

El control difuso se refiere a un conjunto de técnicas de control que utilizan la lógica difusa para el proceso de toma de decisiones dentro de un sistema de control. La lógica difusa nos permite construir sistemas con variables que toman valores “difusos”, de manera muy similar a la forma en la que las personas asignan valores a diferentes tipos de estímulos. Por ejemplo, ante la decisión de salir a la calle o quedarse en casa durante un día libre, uno podría tomar su decisión basándose en dos variables muy simples; clima y cantidad de trabajo pendiente. Ante esta conjetura, la manera humana de resolver el problema sería la siguiente: si tengo poco trabajo pendiente y el clima es agradable entonces saldré a la calle. En cambio, si tengo mucho trabajo y además está lloviendo, me quedaré en casa. Esta forma de pensar, sumamente difusa, sumada a un conjunto de experiencias pa-

sadas, nos permiten tomar decisiones de manera rápida y eficaz, sin tener que recurrir a un complejo cálculo que determine si saldré o no con base en una serie de valores cuantitativos sobre el clima y la cantidad de trabajo pendiente.

En este trabajo se busca aplicar la lógica difusa a un sistema de control de un motor eléctrico de corriente directa sin escobillas (BLDC). De manera análoga al caso de tener que decidir si uno saldrá a la calle o no, aquí se tomarán algunas variables para el control de velocidad del motor, y se construirá un sistema de inferencia difuso para determinar como tienen que cambiar las variables para lograr una velocidad deseada. Además del diseño del control, se implementará el sistema de control con un BLDC utilizando componentes disponibles en el mercado. Se buscará dar un panorama breve, sin llegar a mucho detalle, del marco teórico de los motores eléctricos, lógica difusa y sistemas de control de motores eléctricos. Se presentará una descripción de la metodología y una descripción de los componentes utilizados para la construcción del sistema de control. Finalmente, se discutirán y compararán los resultados y se llegará a las conclusiones correspondientes.

## 1.2 OBJETIVO

Este trabajo tiene como objetivo principal implementar un control de velocidad difuso para un motor de corriente directa sin escobillas. Como objetivos subyacentes, se proriza también lograr un control sin el uso de sensores físicos y además diseñar un sistema de inferencia difuso que pueda controlar el motor de manera eficiente. Adicionalmente se buscará realizar un análisis comparativo de desempeño con los controles tradicionales, de la familia PID, con la finalidad de llegar a una conclusión sobre la viabilidad y ventajas de la aplicación de la lógica difusa en el contexto del control de velocidad y motores BLDC.

## 1.3 INTRODUCCIÓN Y ESTADO DEL ARTE

En la actualidad existe una gran variedad de motores eléctricos, que por sus curvas características de velocidad-torque, precio, complejidad en su construcción y otras características, encuentran diferentes aplicaciones en la industria. Podemos definir de manera breve el concepto de motor de la siguiente manera; los motores son dispositivos que transforman energía eléctrica, ya sea en modalidad de corriente alterna o directa, en energía mecánica en modo de movimiento rotatorio (Perez-Amador, 1992). Es posible clasificar las máquinas eléctricas, ya sea generadores o motores, entre máquinas de CA (corriente alterna) y máquinas de CD (corriente directa). En el caso de los motores, dónde la energía electromagnética se convierte en energía mecánica,

la clasificación se puede entender a partir de la señal de alimentación; dónde la señal de alimentación es una señal alterna o directa. El motor síncrono y el motor de inducción son ejemplos de motores eléctricos y en ambos el voltaje de alimentación es un voltaje de CA.

Por otro lado, otra forma de hacer la clasificación de motores tiene que ver con el embobinado de armadura (A.E. Fitzgerald, 2003). En los motores de CA el embobinado de armadura se suele encontrar en el estator y la corriente de CA que se le aplica a este embobinado es la que produce el campo magnético rotatorio. En cambio, en los motores de CD, el embobinado de armadura se encuentra en el rotor. Actualmente, el motor más popular para uso industrial es el motor de inducción (en particular, el motor de inducción de jaula de ardilla) (A.E. Fitzgerald, 2003). El motor de inducción se utiliza comúnmente en aplicaciones domésticas e industriales debido a su robustez y fácil mantenimiento. Lo que hace al motor de inducción especial, es que se puede operar bajo casi cualquier condición ambiental. La falta de anillos, escobillas o cualquier otro tipo de conmutador mecánico hace que el mantenimiento de este tipo de motor sea casi nulo. Los motores síncronos y los motores de CD tienen también características particulares que encuentran diversas aplicaciones. A continuación se hará una revisión breve y concisa de los motores más populares.

#### 1.4 MOTORES DE CD

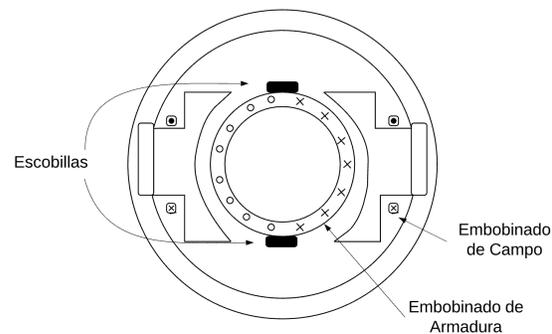


Figura 1: Corte transversal de una máquina de corriente directa con embobinado de campo.

Los motores de CD son ampliamente usados en aplicaciones que requieren un amplio rango de velocidades y control preciso sobre la salida del motor (A.E. Fitzgerald, 2003). Una ventaja de los motores de CD sobre los motores de CA es que el control de éstos es más simple. En contraparte, los motores de CA suelen ser más robustos y su construcción suele ser más sencilla. En los motores de CD la armadura es el rotor y se construye a partir de un núcleo cilíndrico de material laminado, ranurado en toda su periferia y en las mismas

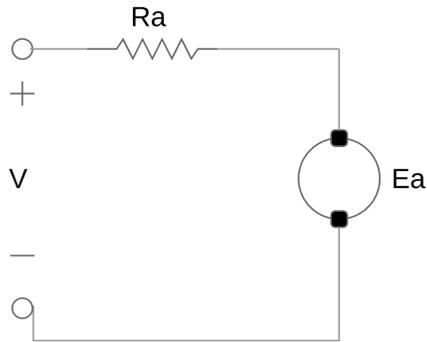


Figura 2: Circuito equivalente de un motor de CD con imanes permanentes

ranuras se aloja el embobinado de armadura, cuyas terminales se conectan al conmutador (Perez-Amador, 1992). La [Figura 1](#) muestra un corte transversal de la máquina de corriente directa.

El estator en estos motores puede ser un embobinado de campo o imanes permanentes. Cuando el imán es permanente se puede prescindir del circuito que excita al embobinado de campo ya que el campo magnético de los imanes permanentes está siempre presente. En la [Figura 2](#) se muestra un diagrama esquemático básico de un motor de CD. Las ecuaciones que rigen el comportamiento de los motores de CD son (A.E. Fitzgerald, 2003) :

$$E_a = k\phi\omega \quad (1)$$

$$V = E_a + R_a I_a \quad (2)$$

$$T = k\phi I_a \quad (3)$$

Dónde

- $k$  es la constante del motor
- $\phi$  es el flujo magnético por polo
- $\omega$  es la velocidad rotacional del motor
- $E_a$  es la fuerza contra electromotriz
- $V$  es la diferencia de potencial aplicada
- $R_a$  es la resistencia de armadura del motor
- $I_a$  es la corriente entregada a la armadura
- $T$  es el torque producido

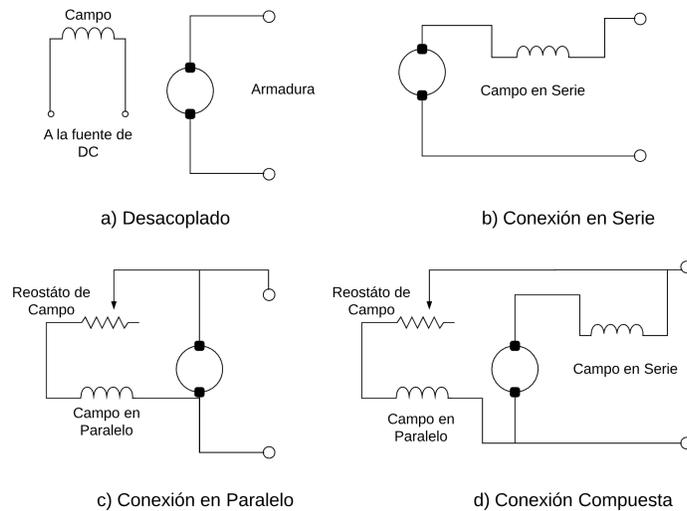


Figura 3: Posibles conexiones del embobinado de campo de un motor de CD

Todos los motores de CD hacen uso de las relaciones descritas previamente pero las variaciones en conexión hacen que las curvas características de torque-velocidad cambien de configuración a configuración. Por lo general, se consideran cuatro tipos de conexiones para los motores de CD que varían por el tipo de excitación (A.E. Fitzgerald, 2003) : las primeras dos configuraciones son las de excitación en serie y excitación en derivación (b y c). Los términos en serie y derivación se refieren a la conexión del embobinado de excitación (de campo) con la armadura. La configuración (a) es la de excitación separada, donde los embobinados están totalmente desacoplados. La [Figura 3](#) muestra los esquemas de las variaciones de conexión que puede tener una máquina de CD, en modo de motor en este caso. Existen además otras variaciones de las conexiones en serie y en derivación de estos motores, llamadas conexiones compuestas, donde hay un embobinado en serie y en paralelo con la carga. Estas conexiones buscan lograr relaciones torque-velocidad distintas, donde el torque aumenta con la velocidad.

La [Figura 4](#) muestra las diferentes curvas características torque-velocidad de tres de las primeras tres configuraciones mencionadas.

Cada conexión hace que el comportamiento del motor tenga variaciones importantes. En la conexión en derivación la velocidad del motor es independiente al torque. Esto hace que la configuración en derivación sea una buena opción para aplicaciones que exigen autorregulación, es decir, una velocidad constante a diferentes niveles de carga. Por otro lado, la conexión con el embobinado de excitación en serie tiene una curva característica de torque-velocidad no lineal. Esto hace que el torque de arranque en este tipo de motor sea muy alto y

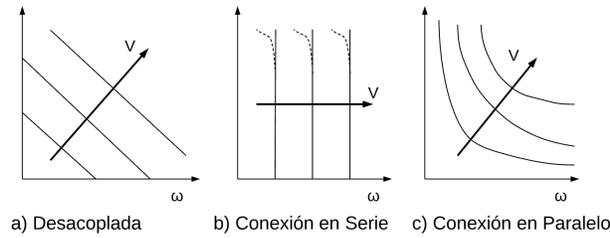


Figura 4: Curvas características de Torque-Velocidad de tres motores de CD con diferente conexión del embobinado de campo

naturalmente encuentra aplicaciones en donde se requiere esta condición durante el arranque, un ejemplo de esto son los ferrocarriles.

Existen otro tipo de motores de CD. El motor de imanes permanentes de CD es, tal vez, el motor más sencillo en cuanto a operación. En este motor el estator está compuesto de imanes permanentes montados y ordenados sobre una base que producen un flujo magnético constante  $\phi$ . Las ventajas de estos motores de CD tienen que ver con su fácil construcción y que no se requiere de una excitación adicional para energizar los embobinados de campo, ya que estos embobinados y su excitación son reemplazados por los imanes permanentes. Sin embargo, estos motores también tienen algunas desventajas, principalmente, el riesgo de desmagnetización de los imanes permanentes que haría que el motor se volviera inoperante.

En un motor con imanes permanentes el flujo magnético es constante y la fuente de voltaje que alimenta al rotor es fija. Como resultado, podemos plantear el circuito equivalente de la [Figura 2](#). Este circuito no tiene embobinado en el estator y está desacoplado con el embobinado de armadura (rotor).

En este caso, el producto  $k\phi$  se vuelve una constante, y se le denomina la constante de torque. Se vuelve constante ya que en el caso del motor con imanes permanentes  $\phi$  es constante y  $k$  es una constante que está en función de la geometría del motor. Considerando las tres ecuaciones principales llegamos a la conclusión que el torque en un motor de imanes permanentes está dado por el producto de la constante torque y de la corriente de armadura:

$$T = \frac{E_a I_a}{\omega} = k\phi I_a \quad (4)$$

A pesar de que los motores de CD muestran importantes ventajas en su operación y control, la principal desventaja de éstos que son menos robustos en comparación con los motores de CA debido a que los primero tienen elementos de conmutación mecánicos que suelen ser anillos conductores que entran en contacto con el embobinado de

armadura, que, con el paso del tiempo, se desgastan y deben reemplazarse periódicamente. La robustez está asociada con factores como durabilidad y costo de mantenimiento lo que los hace máquinas más caras. Estos mecanismos son en contraparte, el motor asíncrono y el motor de inducción no cuentan con este tipo de mecanismos. A continuación, se describen las principales características de los motores de CA y, cómo se ha resuelto el inconveniente de la conmutación mecánica a través del uso de la electrónica de potencia.

### 1.5 MOTORES SÍNCRONOS

El motor síncrono es uno de los dos motores de CA. Se le considera motor de CA ya que al embobinado del estator se le excita con una corriente de CA trifásica para producir un campo magnético rotatorio. El rotor de la máquina síncrona puede estar construido por imanes permanentes o por embobinados excitados con una fuente de CD para producir un campo magnético fijo. En este caso, la armadura, el órgano sobre el cual se inducen las fuerzas electromotrices, son bobinas que se alojan en las ranuras hechas en el núcleo del estator (Perez-Amador, 1992).

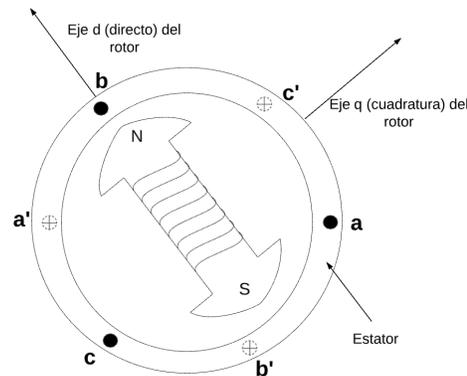


Figura 5: Máquina síncrona trifásica de dos polos

A diferencia de los motores de CD, el embobinado de excitación se encuentra en el rotor. Al producirse un campo magnético fijo en el rotor y al producirse un campo magnético rotatorio en el estator que gira a la velocidad que fija la frecuencia de la fuente de CA, que alimenta al embobinado de la armadura, el campo del rotor tratará de seguir al campo rotatorio y por consecuencia el rotor girará a la velocidad que gira el campo producido por la fuente de CA. El torque ejercido sobre el rotor en todo momento trata de alinear ambos campos, y de esta manera el rotor sigue al campo rotario del estator a la misma velocidad. Es importante notar que cuando la velocidad del motor se atrasa a la del campo, no hay torque total.

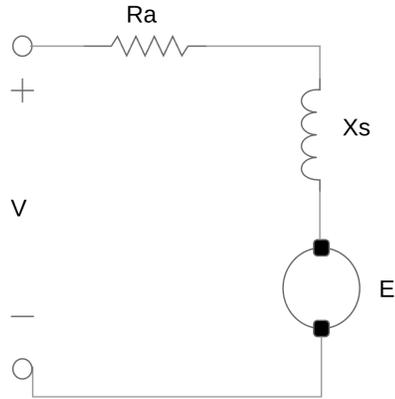


Figura 6: Circuito equivalente de un motor síncrono

Podemos proponer un modelo simplificado de una de las fases del motor. Su circuito equivalente se vería de la siguiente manera, como lo muestra la [Figura 6](#).

Si planteamos la ecuación de la malla, empleando fasores, obtenemos la siguiente relación:

$$V = E + R_a I_a + X_s I_a \quad (5)$$

Donde  $V$ ,  $I_a$  y  $E$  son fasores y  $X_s$  es la reactancia de la armadura. La ecuación del torque en este motor es proporcional al seno del ángulo entre el campo fijo y el campo rotatorio (A.E. Fitzgerald, 2003). Si el torque de carga excede el torque del motor, puede haber un deslizamiento de uno o más polos y causar una disrupción transitoria.

## 1.6 MOTOR DE INDUCCIÓN

Los motores de inducción pueden ser de dos tipos: el primero, es un motor cuyo rotor tiene un embobinado con el mismo número de polos que el embobinado del estator; este tipo de motores de inducción son muy poco comunes y son encontrados en aplicaciones muy específicas, por esta razón no se hablará de ellos a detalle en este trabajo y, el segundo es el motor de inducción con rotor de jaula de ardilla. Este segundo motor tiene la peculiar característica que su rotor no está construido a partir de embobinados sino a partir de barras conductoras que forman una estructura similar a la de una jaula de ardilla. La [Figura 7](#) muestra el rotor de un motor de jaula de ardilla.

Las principales ventajas de estos motores son su sencillez y robustez, que hacen de estos motores los más comunes y más usados en aplicaciones empezando desde una fracción de caballo de fuerza ( $P < 746\text{Watts}$ ) (A.E. Fitzgerald, 2003).

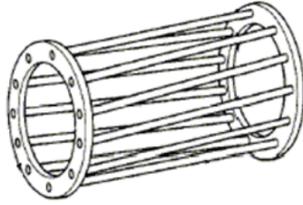


Figura 7: Rotor de un motor de inducción de jaula de ardilla

El funcionamiento del motor de inducción se puede describir brevemente de la siguiente manera: el estator tiene una configuración similar al de una máquina síncrona, es decir, tiene embobinados que crean un campo magnético de dos polos que rota a la velocidad de la fuente de alimentación o a la velocidad síncrona  $\omega_s$ . La superposición de los campos magnéticos creados por la corriente de CA que alimenta a cada uno de los embobinados del estator hace que se genere un campo magnético de dos polos que gira a la llamada velocidad síncrona, tal y como sucede en motor síncrono. Entonces, al existir un campo magnético rotatorio en la presencia de las barras conductoras que están conectadas en corto circuito, se genera una FCE (fuerza contraelectromotriz) que produce una corriente que circula a través de todas las barras. Estas acciones llevan a que exista una corriente y un campo y que a su vez, debido a la relación  $F = I \times B$ , se genere un par que haga girar al rotor.

Cabe notar que si la velocidad síncrona  $\omega_s$ , del campo del estator, es igual a la velocidad del rotor  $\omega_r$ , entonces no habrá corriente inducida en las barras del rotor ya que el campo magnético que verían las barras sería constante. Dado esto, siempre que  $\omega_s \neq \omega_r$ , habrá una corriente inducida en las barras y por consecuencia un torque. A esta diferencia de velocidades se le llama deslizamiento,  $s$ .

El deslizamiento está dado por la siguiente relación:

$$s = \frac{\omega_s - \omega_r}{\omega_s} \quad (6)$$

De la misma forma, la velocidad del rotor en rpm puede ser expresada en términos del deslizamiento y la velocidad síncrona:

$$\omega_r = (1 - s)\omega_s \quad (7)$$

Un deslizamiento de 0 significa que  $\omega_s = \omega_r$  y que el torque generado es cero ya que el campo magnético visto por el rotor es constante. Por otro lado, una razón de deslizamiento de 1 significa que el rotor se ha detenido. Considerando esto, la mayoría de los motores de inducción se diseñan para operar a una razón de deslizamiento entre 0,1 – 0,24.

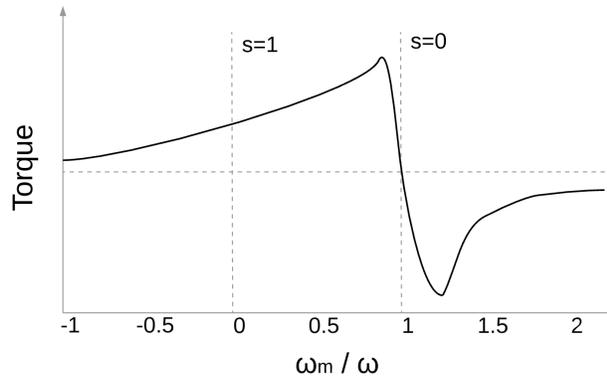


Figura 8: Curva característica Torque-Velocidad de un motor de inducción trifásico típico

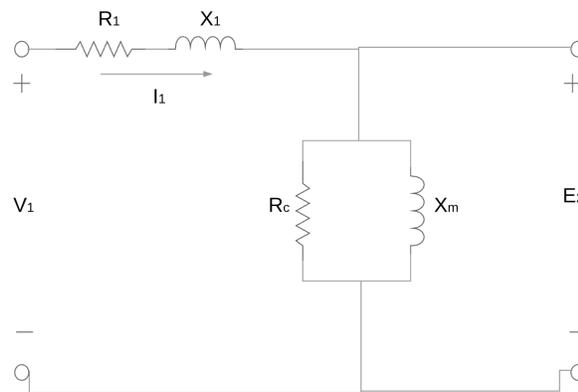


Figura 9: Circuito equivalente de un motor de inducción

La [Figura 8](#) muestra la curva característica torque-velocidad para un motor de inducción. Podemos observar que se obtiene torque máximo cuando se tiene un deslizamiento ligeramente mayor a cero, con valores menores al tener velocidades menores.

Si analizamos el circuito equivalente de [Figura 9](#) para una fase, en modo motor, considerando que los voltajes de alimentación son valores de línea a neutro, obtenemos la siguiente relación:

$$V_1 = E_2 + I_1(R_1 + jX_1) \quad (8)$$

dónde  $V_1$  es el voltaje de terminal de línea a neutro,  $E_2$  la FCE inducida por el flujo magnético,  $I_1$  es la corriente del estator,  $R_1$  resistencia efectiva del estator y  $X_1$  reactancia de fuga del estator.

En la [Figura 9](#) se observa el circuito equivalente utilizado para modelar al motor. Puede observarse que éste circuito equivalente corresponde con aquel utilizado para modelar el primario de un transformador (A.E. Fitzgerald, 2003).

## 1.7 MOTORES BLDC

En este trabajo es de especial interés un motor que tiene un funcionamiento muy similar al de una máquina síncrona pero que se alimenta con una fuente de corriente directa. Como primer paso es importante entender los motores de ac de imanes permanentes. Los motores de CA de imanes permanentes son motores síncronos de múltiples fases. Su rotor está constituido por los mismos imanes. Son muy similares a las máquinas síncronas, con la única diferencia siendo que el embobinado de excitación en el rotor es reemplazado por los imanes ya magnetizados.

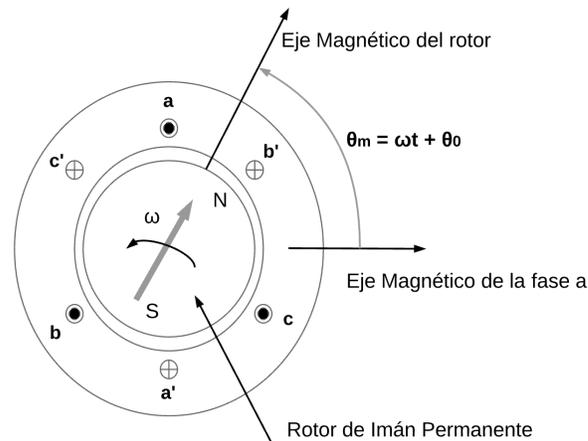


Figura 10: Corte transversal de un motor de CA con rotor de imanes permanentes.

En la [Figura 10](#) se muestra un motor de CA de imanes permanentes con el rotor dentro del estator. Cabe aclarar que existe otra tipo de estos motores en los que el imán permanente está por fuera, es decir, alrededor del estator.

A estos motores también se les conoce como motores de CD sin escobillas (BLDC brushless DC motors). Cuando se excita el estator de este motor con una fuente sinusoidal, el comportamiento se asemeja al de un motor síncrono. Sin embargo, al BLDC se le alimenta con una señal de CD (por ende su nombre) que pasa por un inversor que a su vez produce señales escalonadas en cada fase para ocasionar que el motor gire de una posición de equilibrio a otra. Este comportamiento es similar al del motor síncrono, y una de las diferencias radica en la señal de alimentación. En determinadas aplicaciones en las que es de interés controlar la velocidad y el torque del motor es posible implementar una retroalimentación de la posición del imán permanente para determinar el momento exacto y duración del pulso de excitación de cada fase para lograr la velocidad deseada (A.E. Fitzgerald, 2003)

Para estas aplicaciones es necesario contar con un circuito controlador de voltaje variable y un sistema de conmutación electrónico.

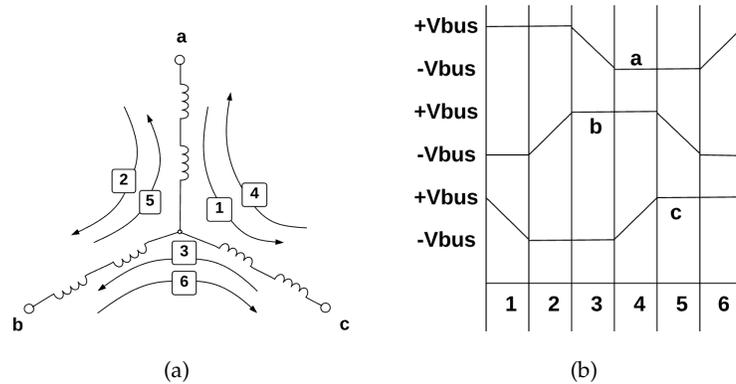


Figura 11: a) Embobinados del estator por fase conectados en estrella (BLDC) b) Gráfica de excitación por fase para un ciclo mecánico completo

La construcción básica de un BLDC se muestra en la [Figura 12](#).

La [Figura 12](#) muestra un BLDC. En su forma más simple un motor BLDC tiene dos embobinados por fase en el estator y un rotor con dos polos. Los embobinados están conectados en una configuración de estrella a un nodo común. En la práctica, hay un mayor número de embobinados en el estator y hay casos donde el rotor cuenta con más de un imán permanente. Sin embargo, el caso de la figura uno y dos se puede aplicar para motores con mayor número de polos ya que el sistema de seis pasos de conmutación va atado a un sistema de alimentación trifásico.

La [Figura 11](#) muestra un sistema de conmutación de 6 pasos, en el que en todo momento sólo hay corriente fluyendo en dos de las tres

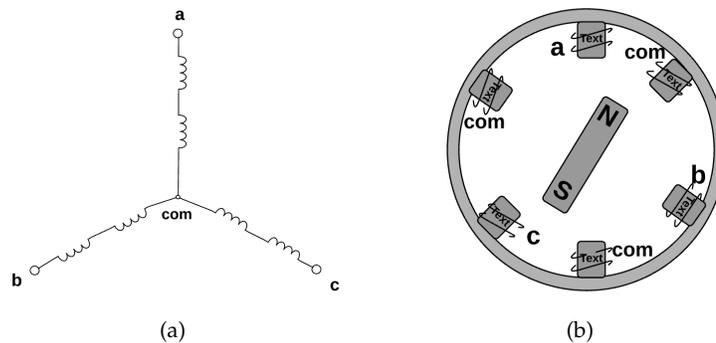


Figura 12: a) Conexión en estrella de los embobinados de cada fase b) Construcción de un motor BLDC (corte transversal)

fases. Si observamos la [Figura 12](#) y la posición donde se encuentra el polo norte del imán con referencia a las embobinados se puede concluir que en esa posición debe haber una corriente tal que haga que se produzca un polo sur en el embobinado a su derecha y un polo norte en el embobinado a su izquierda, y que además no exista corriente en la tercer fase. Esta combinación de voltajes en cada una de las fases produce movimiento en el rotor y cuando el rotor llega a la nueva posición, entonces, se activará el siguiente paso en la conmutación, y esta secuencia se repite indefinidamente durante la operación del motor. Para lograr que el campo magnético en el estator avance por delante del rotor, la transición entre un estado al otro debe ocurrir en posiciones precisas del rotor para lograr un torque óptimo. Las flechas en la [Figura 11](#) muestran la dirección de la corriente a través de cada uno de los embobinados en cada uno de los 6 pasos de conmutación (Condit, 2007). Por esta razón es importante conocer la posición del rotor de tal manera que la conmutación electrónica de cada una de las fases del motor se realice en instantes de tiempo bien determinados. Por esta razón, es fundamental entender las dos principales formas de control de este tipo de motores: control con sensores y control sin sensores.

## 1.8 CONTROL CON SENSORES

En la mayoría de las aplicaciones de control de un BLDC con sensores, se utilizan sensores de efecto Hall. Los sensores se montan en el estator para determinar la posición del rotor. Un esquema detallado de la salidas de cada uno de los sensores durante la operación de un motor BLDC puede encontrarse en (Condit, 2007). El funcionamiento básico de estos sensores se puede explicar de la siguiente manera: cada vez que uno de los polos magnéticos del rotor pasa cerca a uno de los sensores, estos generan una señal de salida que puede ser baja o alta (positiva o negativa), que indirectamente indican que un polo norte o un polo sur está pasando cerca de los sensores. Este comportamiento es producto del efecto Hall. La teoría detrás de este efecto es la siguiente: si un conductor que lleva un corriente eléctrica se encuentra dentro de un campo magnético entonces el campo ejerce una fuerza transversal sobre los portadores de carga y esto a su vez los empuja hacia un lado del conductor. La acumulación de carga de un lado del conductor produce una diferencia de potencial medible entre los dos lados del conductor (Yedamale, 2003). La siguiente figura muestra una sección transversal de un motor BLDC y cómo y dónde se montan los sensores de efecto Hall.

Se puede observar en la [Figura 13](#) que hay sensores adicionales montados en el eje. Esto se debe a que es más sencillo montar una réplica de los sensores que montarlos directamente sobre el estator.

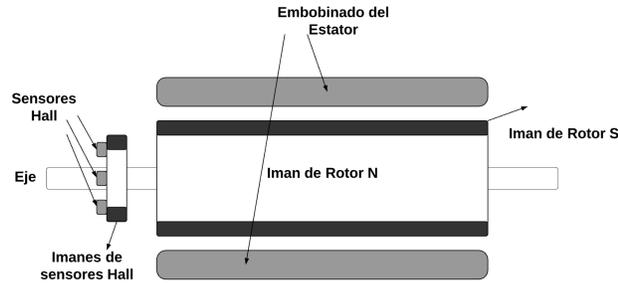


Figura 13: Sección Longitudinal de un BLDC con sensores Hall montados sobre el eje.

En las aplicaciones de control con sensor, el control mismo se vuelve sencillo ya que el procesador sólo necesita esperar a que los sensores Hall cambien de estado para saber dónde está el rotor y producir las señales de conmutación adecuadas. Sin embargo, este tipo de control con sensores tiene algunas desventajas (Condit, 2007). Algunas de ellas se enlistan a continuación:

- El costo adicional de los sensores, la montura en el proceso de manufactura y los cables de los sensores incrementa el costo total.
- La falla de los sensores produce una falla total del motor y del sistema.
- Es impráctico usar sensores en ambientes donde se busca robustez adicional, por ejemplo, en aplicaciones donde el motor se encuentra inundado.
- Los sensores son sensibles al calor. Su operación varía al presentarse variaciones de temperatura.
- Son sensibles a otras perturbaciones magnéticas.

#### 1.9 CONTROL SIN SENSORES

Con el uso de sensores *Hall* determinar la posición del rotor es sencillo, sin embargo, en el control sin sensores es necesario determinar la posición del rotor sin el uso de sensores adicionales al sistema. Una de las soluciones que existen se encuentra en el análisis de la fuerza contraelectromotriz (FCE). La FCE es la diferencia de potencial inducida en los bobinados del estator debido al imán permanente del rotor que gira. Esta FCE se produce ya que el campo magnético que observa el bobinado del estator es un campo que cambia en el tiempo. La FCE se opone al voltaje de alimentación y se opone a la corriente que fluye en el bobinado. De acuerdo con la *Ley de Faraday* aplicada al motor, la magnitud de la FCE es proporcional a la

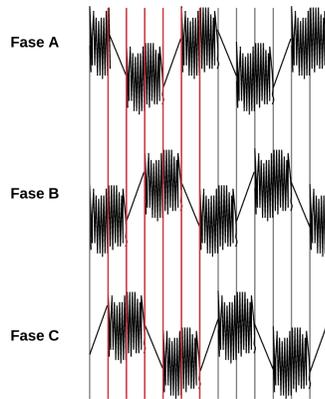


Figura 14: Oscilograma de la FCE de cada una de las fases (cada seis segmentos representan un ciclo de conmutación completo:  $360^\circ$  eléctricos)

velocidad del motor; cuando la magnitud del voltaje de alimentación aumenta, la velocidad aumenta y a su vez la FCE también aumenta en magnitud. Una descripción más detallada sobre cómo se genera la FCE en los BLDC se encuentra en (Yedamale, 2003).

La [Figura 14](#) muestra la FCE en cada uno de los embobinados correspondiente a cada una de las tres fases. Se puede observar que la forma de onda es muy similar a la forma trapezoidal y que cada fase está desfasada  $120^\circ$  con respecto a la otra. Por esta razón, es posible dividir el diagrama en seis intervalos que se repiten periódicamente y, además, se observa que cada tres tiempos no hay voltaje en alguna de las tres fases. Existen tres características principales de esta señal que resultan muy importantes para el motor, tal y como se indica en la referencia (Condit, 2007), que son:

- Cuando la velocidad del motor aumenta la magnitud de la FCE aumenta.
- Cuando la velocidad aumenta, la pendiente de la FCE también se hace más grande, siempre y cuando la FCE no sea constante
- La señal de la FCE es simétrica alrededor de  $0[V]$ , siempre y cuando los voltajes de alimentación al circuito de control sean  $+V$  y  $-V$ .

La [Figura 15](#) muestra un oscilograma real de una FCE. La forma de la señal difiere un poco de una señal trapezoidal perfecta ya que existe una cantidad importante de ruido generado por la conmutación montado en la señal del componente de modulación por ancho de pulso (PWM por sus siglas en inglés). En el caso de que no existiera ruido en la señal, la transición de la señal de positivo a negativo sería

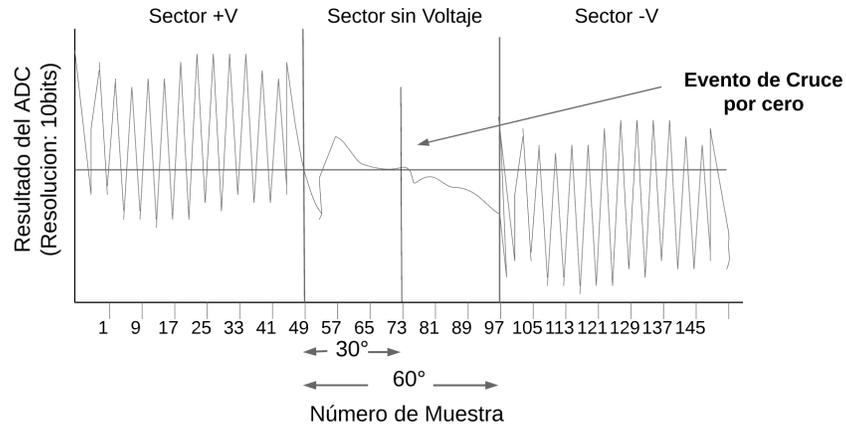


Figura 15: Oscilograma de la señal de la fuerza contraelectromotriz pasada por un CAD (Convertidor Analógico Digital)

una línea recta. Si esta parte de la señal fuera recta entonces la señal cruzaría  $0[V]$  exactamente a la mitad intervalo de no conducción, o  $30^\circ$  eléctricos a partir del inicio de la transición. Entonces el evento de cruce por cero se debería de dar  $30^\circ$  antes del siguiente paso de conmutación. Si es posible identificar el evento de cruce por cero de manera precisa entonces la posición del rotor se puede estimar y los embobinados se podrían conmutar de manera correcta para que el motor funcione sin necesidad de disponer de sensores externos al motor. Detectar este cruce por cero de una manera precisa es fundamental para lograr el control sin sensores que es una tarea compleja debido a que es imposible eliminar el ruido de conmutación. Si se implementara un algoritmo que esperara  $30^\circ$  para conmutar el sistema cuando se detectara el primer cruce por cero y tuviéramos una señal tan ruidosa como la que tenemos entonces es muy probable que el sistema falle ya que no es posible observar en la señal de la figura que debido al ruido, el primer cruce por cero no se da exactamente a los treinta grados. Esto resultaría en la falla total del sistema ya que los momentos de conmutación no serían los correctos y fallarían en cada ciclo de la señal. Esta problemática aumenta el grado de complejidad del sistema porque para solucionar este es necesario implementar un filtro para tratar la señal y eliminar parte del ruido.

La señal se puede filtrar con alguna implementación de hardware directamente o implementar un filtro digital para tratarla. El objetivo del filtro es eliminar las altas frecuencias de la señal que constituyen el ruido y así detectar el cruce por cero con una señal con menor nivel de ruido. La siguiente figura muestra una señal filtrada usando un DSP (*Digital Signal Processor* por sus siglas en inglés), que intenta aproximar la señal a una señal trapezoidal. La implementación de este filtro se detalla en (Condit, 2007).

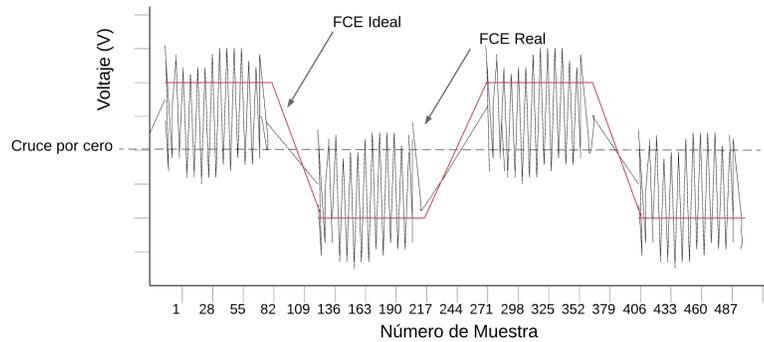


Figura 16: Oscilograma de la señal de la fuerza contraelectromotriz real e ideal

Los filtros solucionan la problemática del ruido presente durante la detección del cruce por cero. Sin embargo, los filtros también tienen una desventaja que debe ser considerada. Cualquier filtro, sea analógico o digital, introduce un retardo en el sistema. Dependiendo del diseño del filtro y de la frecuencia de operación del motor, el desfase que introduzca el filtro sobre la señal puede ser un problema en esta aplicación por que la detección precisa del instante de tiempo en el que ocurre el cruce por cero es crucial. Afortunadamente, el retardo de fase se puede calcular a partir de las especificaciones del filtro y se puede considerar en los algoritmos de conmutación.

#### 1.10 SISTEMAS DE ARRANQUE

A diferencia de un motor de inducción, los motores BLDC no cuentan con sistemas de arranque inherentes. Si se considera que un BLDC se va a controlar sin usar sensores de efecto *Hall*, entonces, surge el problema de que durante el arranque no es posible determinar la posición del rotor por medio de la fuerza contraelectromotriz porque su amplitud es nula o muy pequeña a velocidades bajas. Es por ello que los métodos de arranque de los BLDC son importantes y, a continuación, se tratará brevemente de ellos.

El algoritmo más sencillo que se emplea en este problema se llama pre-posicionamiento (M. Tursini, 2003). El pre-posicionamiento consiste en energizar dos fases para alinear el rotor a una posición conocida. A partir de esto el motor se acelera de acuerdo a una serie de secuencias de disparo con intervalos de tiempo decrecientes. La gran desventaja de este método es que en muchas ocasiones el rotor puede no seguir la secuencia de excitación y tiende a oscilar o a girar en reversa y así provocar una falla en el arranque, el detalle de este tipo de fallas se describe en el método propuesto por Tursini (M. Tursini, 2003). Otros autores han propuesto métodos que estiman la posición del rotor midiendo las variaciones de inductancia por efecto

del imán permanente. Vasudeven propone un método de arranque que utiliza la FCE en los embobinados del estator, medida de manera indirecta, restando los voltajes de fase a fase (Vasudeven, 2008). Esta implementación no requiere de instrumentos de medición adicionales. El método propuesto está definido en pasos de acuerdo al esquema mostrado en la siguiente figura.

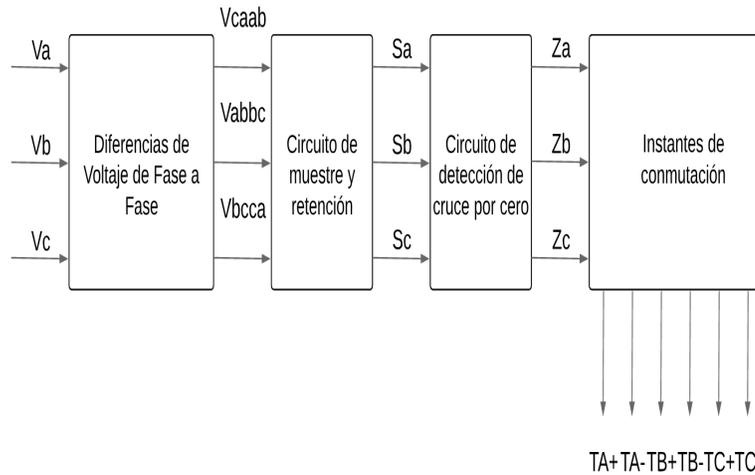


Figura 17: Diagrama de método de arranque sin sensores

Sin embargo, este método también requiere de un pre-posicionamiento previo al arranque de los bloques mostrados en la [Figura 17](#).

La mayoría de los sistemas de control de motores síncronos de imanes permanentes ((PMSM) por sus siglas en inglés) usan técnicas especiales para arrancar el motor y esperan a que la FCE sea de suficiente magnitud para entrar al control de lazo cerrado que depende del cruce por cero de la FCE, para energizar las fases correctas. Es decir, en estado estable, un BLDC sin sensores es controlado con la información de la FCE. Como se mencionó previamente, a bajas velocidades y en reposo, no hay señal de FCE o es muy pequeña. Debido a esta problemática, existen dos métodos que son comúnmente usados para el arranque y para la operación a muy bajas velocidades. La técnica más común es la de arranque de lazo abierto. En este método se aplica un voltaje o una secuencia aleatoria sin importar la posición del rotor. La principal desventaja de este método es su muy baja eficiencia. Al no conocer la posición del rotor, el torque de arranque será muy bajo y muy lejano al óptimo. En aplicaciones donde el torque de arranque es importante, este método presenta graves problemas. En muchos casos, el empleo de este método sí resulta en un arranque, pero en un arranque muy accidentado. Es común que se presenten rotaciones en dirección incorrecta. Un BLDC se caracteriza por su alta eficiencia y este método en particular no permite un arranque suave, rápido y con el máximo torque posible. Mucho más detalle de esta técnica de

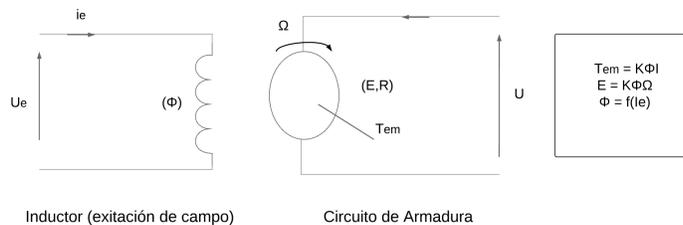


Figura 18: Modelo de un motor de CD de excitación separada (imanes permanentes)

arranque y referencias a otras técnicas, como la de inyección de alta frecuencia, se pueden encontrar en el trabajo de Yun (Yun, 2012).

### 1.11 CONTROL DE CAMPO ORIENTADO (FOC POR SUS SIGLAS EN INGLÉS)

Los BLDC son empleados en aplicaciones que demandan alto desempeño y alta eficiencia. Esto se traduce en la necesidad de tener un control de motor de alto desempeño. Las características generales de un control de alto desempeño son las siguientes (Bose, 2002):

- Rotación suave a lo largo de todo el rango de velocidades
- Control de Torque completo desde reposo (velocidad 0)
- Aceleración y desaceleración rápida

Para lograr este alto desempeño se emplean técnicas de control vectorial en el control del BLDC. A estas técnicas de control vectorial se les conoce como Control de Campo Orientado (FOC).

La idea u objetivo básico detrás del FOC es descomponer la corriente del estator en dos partes; el componente que genera campo magnético y el componente que genera torque. Al descomponer estas dos partes, podemos lograr el control independiente de cada uno. En otras palabras, el FOC, se vale de transformaciones matemáticas para desacoplar las funciones que generan torque y las funciones que generan magnetización dentro de un PMSM.

Para entender el racional detrás del desacoplamiento y la filosofía detrás del FOC es necesario primero entender cómo opera un motor de CD excitado de manera separada. En este tipo de motores, la excitación del rotor y del estator son controladas de manera independiente. La magnitud del campo de excitación determina el valor del flujo y la corriente del estator determina la cantidad de torque que se produce. La [Figura 18](#) nos muestra el modelo de un Motor de CD de excitación separada.

Uno de los primeros puntos a notar es que los circuitos de la armadura y del embobinado de campo están desacoplados y que al fijar  $I_e$ , la corriente de estator, sólo queda la constante  $K$ , y en ese momento el torque del motor,  $T_{em}$ , sólo depende y es directamente proporcional a la corriente de armadura,  $I$ . Estas ecuaciones nos muestran que podemos controlar el flujo y el torque de manera independiente.

Sabemos que el producto de dos vectores es máximo cuando están en cuadratura, es decir, que los separa un ángulo de  $90^\circ$ . En este tipo de motores, el conmutador mecánico juega un papel crucial en este sentido ya que su construcción está diseñada para conmutar en los embobinados del circuito que están alineados para producir el máximo torque. Este arreglo del conmutador y de los embobinados asegura que la producción del torque es en todo momento muy cercana a la producción óptima. El punto más importante a recalcar es que el motor está diseñado para que el *flujo producido por los embobinados del rotor sea siempre ortogonal al campo del estator*.

A diferencia de los motores de CD, los motores síncronos cuentan con una fuente de poder única, los voltajes de fase del estator. Por consecuencia el flujo y el torque dependen uno del otro. Aquí es justamente donde se vuelve relevante la idea del FOC. Esta técnica nos va a permitir desacoplar el torque y los componentes de flujo magnetizantes de la corriente de estator. Explicado de una manera ya más técnica, este desacoplamiento se logra de la siguiente manera: el FOC consiste en controlar las tres corrientes del estator, representadas por medio de un vector. Este control se basa en una proyección que transforma un sistema trifásico que depende del tiempo y la velocidad a un sistema de dos coordenadas invariante en el tiempo.

Podemos descomponer el proceso del FOC en 4 pasos conceptuales:

1. Medir la corriente que fluye hacia el motor.
2. Comparar la corriente medida contra la corriente deseada y generar una señal de error.
3. Amplificar la señal de error para generar un voltaje de corrección.
4. Modular el voltaje de corrección en las terminales del motor.

A continuación se explicara cada uno de los pasos con las transformaciones matemáticas correspondientes para convertir al sistema variante en el tiempo, en un sistema que se pueda controlar con cantidades escalares de CD.

**Paso 1:** La [Figura 19](#) muestra un esquema del motor, con el vector de flujo del rotor y los ejes de cada una de las tres fases:

Los ejes A, B y C están fijos con respecto a la construcción del motor. Ese marco de referencia esta fijo, por lo tanto nos podemos referir a

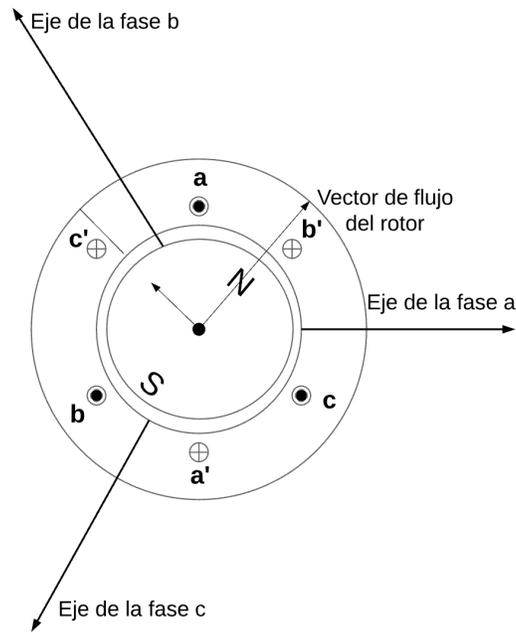


Figura 19: Esquema transversal de un motor BLDC con los ejes por fase y el vector de flujo del rotor

él como un marco estacionario. Debido a la conexión del motor en la figura  $i_a$  e  $i_b$  son positivos, por lo tanto  $i_c$  tiene que ser negativo. Al ajustar estas tres corrientes en fase y en magnitud, podemos obtener cualquier vector espacial de corriente que queramos. Podemos observar la construcción del vector espacial de corriente dentro del marco estacionario en la [Figura 20](#).

Ahora, el objetivo, como se planteó al principio de este apartado, es hacer que el vector espacial de corriente siempre se encuentre a  $90^\circ$  con respecto al vector de flujo. Si bien la razón de ser de esta condición se comentó previamente y se ligó con la producción máxima de torque y por ende eficiencia, la [Figura 21](#) muestra el ángulo entre el vector de flujo y corriente, y el torque por unidad de corriente (torque/corriente).

Se puede observar que el torque es máximo cuando el ángulo entre los dos vectores está a  $90^\circ$  o  $-90^\circ$ . A partir de esta gráfica se deriva el principio de mantener estos dos vectores a  $90^\circ$  en la medida de lo posible para ganar en términos de eficiencia. Una vez medido el vector espacial de corriente del motor y conociendo la posición del rotor y por consecuencia el ángulo del vector de flujo, podemos saber si los vectores son ortogonales o si es necesario realizar un ajuste.

**Paso 2:** En la [Figura 22](#) se puede observar que el vector espacial de corriente en ese momento no es ortogonal al eje del flujo del rotor. Esto causa que el vector espacial de corriente no sea el de referencia

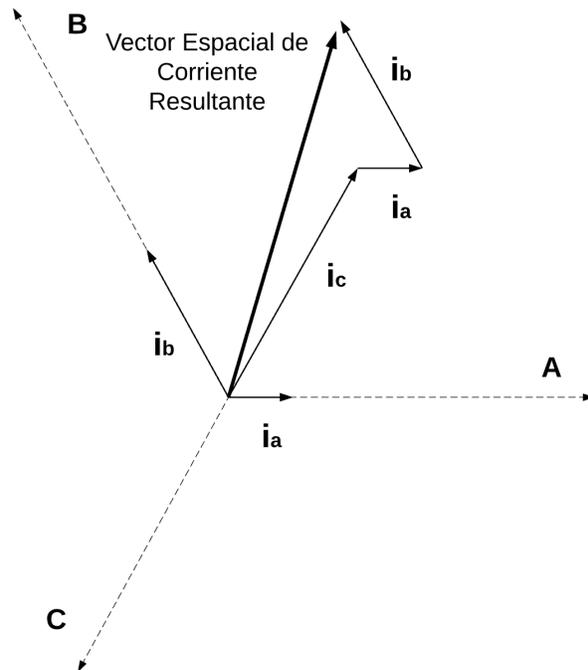


Figura 20: Diagrama vectorial del vector espacial de corriente resultante a partir de las corrientes de fase

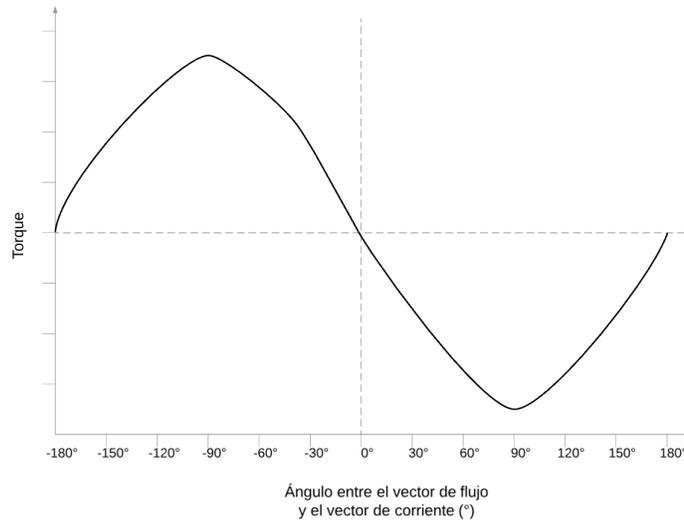


Figura 21: Relación ángulo-torque de un motor BLDC

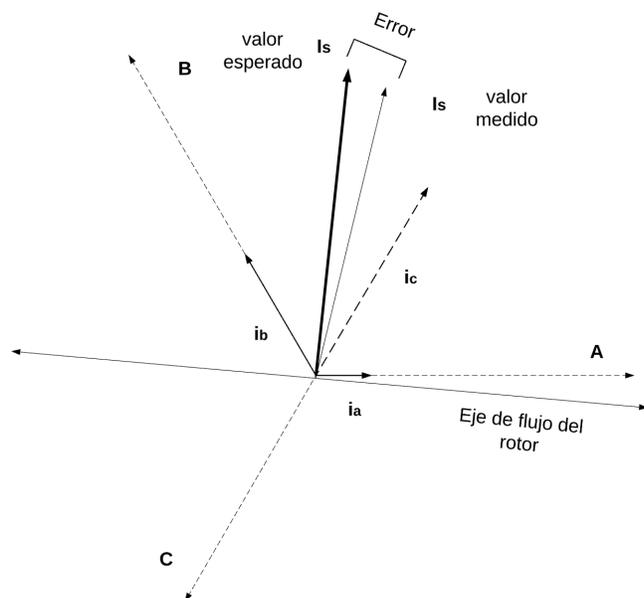


Figura 22: Diagrama Vectorial del vector espacial de corriente resultante medido y esperado

(deseado) y que exista en error entre ambos vectores. Ante esta situación debemos regular la magnitud y el ángulo de  $\hat{I}_s$  a través de la regulación de las corrientes  $i_a$ ,  $i_b$  e  $i_c$ .

Para conocer este error es necesario primero determinar si el vector espacial de corriente total,  $I_s$ , está efectivamente orientado a  $90^\circ$  con respecto al vector de flujo. Es evidente que conocer la orientación del vector de flujo es fundamental para la operación eficiente y ahora se hace muy obvia la importancia de conocer la posición del rotor, ya que conociendo la posición del rotor, la orientación del vector de flujo se hace evidente. Es ahora aún más evidente el papel crucial que juega el conocimiento de la posición del rotor en todo momento. No es un elemento crucial durante el arranque del motor, es un elemento crucial en todo momento de operación del motor. Además, es muy claro que la eficiencia de operación del motor está muy relacionada con la precisión de los métodos de medición de la posición del rotor. Es importante mencionar que por convención, el ángulo del vector de flujo del rotor se mide con respecto al eje magnético de la fase A.

En este punto del proceso podemos notar que el vector espacial de corriente está representado en un sistema sinusoidal trifásico. Este sistema puede ser representado en un marco de referencia con sólo dos ejes, ortogonales entre sí. La idea es convertir los vectores de corriente de las tres fases en dos vectores ortogonales que resulten en el mismo vector espacial de corriente. Esta transformación, de un marco de tres fases a otro marco de referencia con sólo dos ejes ortogonales llamados y es mejor conocida como la transformada directa de Clarke

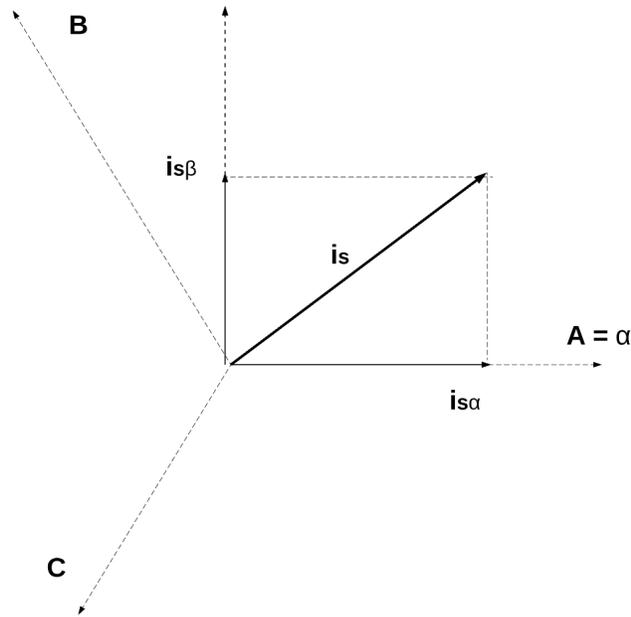


Figura 23: Diagrama vectorial del vector espacial de corriente a partir de las corrientes de fase

(A.E. Fitzgerald, 2003). La finalidad de esta transformada dentro del contexto de un motor trifásico es esencialmente convertir un motor de tres fases a uno de dos fases. Con esta transformación pasamos de tener tres corrientes que regular a sólo dos. La siguiente figura muestra la representación gráfica de la transformada.

Las proyecciones que modifican el sistema trifásico en un sistema ortogonal de dos dimensiones  $(\alpha, \beta)$  se realizan a través de las siguientes ecuaciones:

$$i_{s\alpha} = i_a \quad (9)$$

$$i_{s\beta} = \frac{1}{\sqrt{3}}i_a + \frac{2}{\sqrt{3}}i_b \quad (10)$$

La gran ventaja de esta transformación es que pasamos de tener tres variables para controlar a dos variables. Sin embargo persiste la problemática de tener aún dos corrientes de fase  $(i_{s\alpha}, i_{s\beta})$  que aún dependen del tiempo y de la velocidad. Resulta que aún podemos realizar algunas modificaciones al marco de referencia. La realidad es que estamos trabajando con un sistema sinusoidal que depende del tiempo. Hasta este momento estamos intentando trabajar con un sistema de ejes rotatorios y estamos intentando medirlo desde una referencia estacionaria, los ejes  $(\alpha, \beta)$  en este caso. La problemática se puede

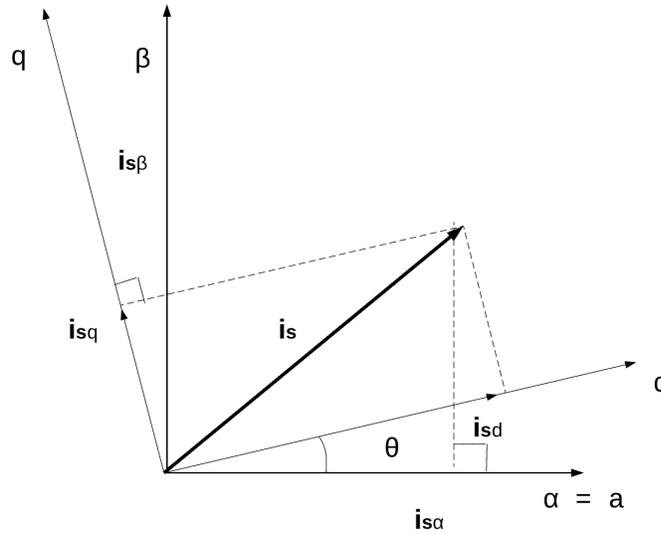


Figura 24: Diagrama vectorial de la transformada directa de Park

plantear con la siguiente analogía: rastrear un objeto que rota a partir de un marco estacionario se vuelve un tarea complicada. Sin embargo, si tratamos de rastrear al objeto desde el mismo marco de referencia rotatorio se vuelve una tarea mucho más sencilla. Esto nos lleva a la segunda transformación del FOC, que resulta ser la transformada más importante ya que esta segunda proyección modifica el sistema ortogonal de dos fases en un marco de referencia rotatorio  $(d, q)$ . A esta segunda transformación se le conoce como transformada directa de Park (A.E. Fitzgerald, 2003), tal y como se muestra en la [Figura 24](#). Este nuevo marco de referencia rotatorio gira de manera síncrona con el flujo del rotor. La siguiente figura considera que el eje  $d$  está alineado con el flujo del rotor y  $\theta$  define la posición del flujo del rotor con respecto al eje magnético de la fase A.

Los componentes de flujo y torque del vector espacial de corriente (sobre los ejes  $d, q$ ) son determinados a través de las siguientes ecuaciones:

$$i_{sd} = i_{s\alpha} \cos \theta + i_{s\beta} \sin \theta \quad (11)$$

$$i_{sq} = i_{s\alpha} \sin \theta + i_{s\beta} \cos \theta \quad (12)$$

Estos dos componentes dependen de los componentes del vector de corriente  $(\alpha, \beta)$  y de la posición del rotor, dada por  $\theta$ . Si conoces la posición del vector de flujo del rotor, y considerando las proyecciones, entonces los componentes  $d$  y  $q$  se hacen constantes. Con esta segunda transformación, dos corrientes de fase se convierten en cantidades

de CD, que no varían en el tiempo. Tal como se ve en la figura y en las ecuaciones,  $i_q$  e  $i_d$  se manejan de manera independiente. Debido a que la comparación se hace en el marco de referencia rotatorio, la frecuencia de CA no figura (se puede despreciar). Esto hace que tengamos únicamente cantidades de CD.

Ahora, con los resultados de estas dos transformaciones nos quedamos con el siguiente esquema en la [Figura 25](#).

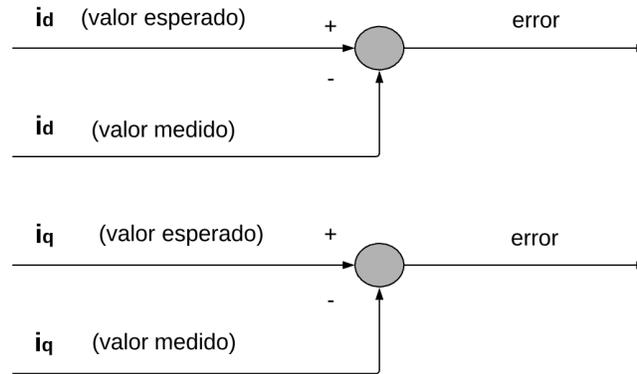


Figura 25: Esquema de las variables de error resultante a partir de los valores medidos y esperados

Cuando el motor opera bajo condiciones normales, todo el flujo sobre el eje d que necesitamos es suministrado por los imanes permanentes del rotor, por lo que la  $i_d$  deseada se fija con valor nulo, es decir, 0. Por otro lado, la  $i_q$  deseada, establece la cantidad de torque que queremos generar.

**Paso 3:** El paso tres lo podemos representar de manera muy simple a través del siguiente esquema de la [Figura 26](#):

Este esquema muestra los valores de CD, de  $i_q$  e  $i_d$  de referencia y los valores correspondientes medidos. La resta de estos genera la señal de error que entra a los bloques de control, PI, y generan los voltajes de corrección. La gran mayoría de las aplicaciones del FOC sobre

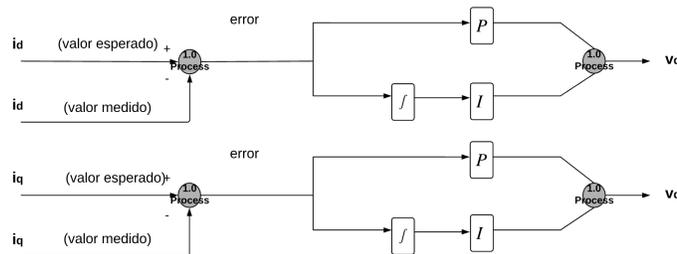


Figura 26: Esquema de las variables de control resultantes al aplicarse las transformaciones sobre las corrientes de fase

PMSM utilizan controles PI, tal y como se muestra en el reporte de aplicación de *Texas Instruments* (Akin y Bhardwaj, 2013).

**Paso 4:** Este último paso es el proceso de hacer las transformaciones de reversa sobre los voltajes de corrección de los ejes d y q. Primero es necesario obtener los voltajes en el marco de referencia fijo, a través de las siguientes ecuaciones. Se busca transformar a los vectores de voltaje de vuelta al sistema de coordenadas estacionario y rectangular.

$$v_{\alpha} = v_d \cos \theta_d - v_q \sin \theta_d \quad (13)$$

$$v_{\beta} = v_d \sin \theta_d + v_q \cos \theta_d \quad (14)$$

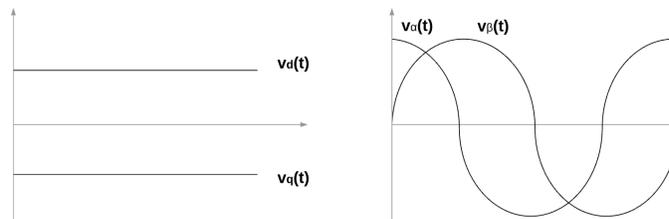


Figura 27: Esquema de las variables de control resultantes al aplicarse las transformaciones sobre las corrientes de fase

Por último transformamos los vectores de voltaje desde el sistema de coordenadas rectangular ( $d, q$ ) a un sistema de vectores de tres fases. En condiciones de estado estable  $v_a$ ,  $v_b$  y  $v_c$  son señales sinusoidales desfasadas  $120^\circ$  respecto a una con la otra.

Estos cuatro pasos son la esencia del FOC y se repiten miles de veces cada segundo para lograr un control de torque y por ende velocidad, de una manera eficiente. El siguiente esquema es un resumen de manera completa el control de torque de un PMSM empleando la técnica del FOC.

## 1.12 ESTIMADOR FAST

Uno de los componentes principales de la solución *InstaSpin-FOC*, mismo que está detrás de la decisión de utilizar *InstaSpin-FOC*, es el estimador FAST. Ya hemos hablado de la importancia de poder medir la magnitud y el flujo magnético en el control de un PMSM, y de las dos posibles soluciones que existen: usar sensores o no usar sensores. Una de las grandes ventajas de este software es que no usa sensores. En vez de usar sensores usa un estimador de alto desempeño, que tienen un gran número de ventajas contra otras soluciones en el mercado (SPRUHJ1G, 2013).

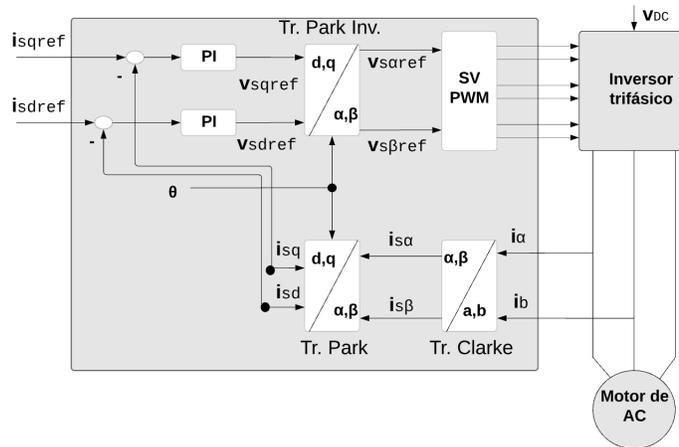


Figura 28: El esquema FOC en diagrama de bloques

El estimador FAST reemplaza al uso de sensores mecánicos. En esencia, el Estimador tiene como núcleo a un observador. Para tener una noción más clara de cómo FAST reemplaza el sensado mecánico para obtener los mismos valores de flujo, es necesario primero entender el concepto de un observador. El problema básico del sistema es que no hay una forma directa de acceder a los embobinados para medir el flujo magnético.

Se dice que un sistema es observable si es posible reconstruir el sistema a partir del uso de un estimador/observador de estado. El observador de estado es el algoritmo que estima el estado interno de un sistema dinámico y lineal a partir de medir las entradas y las salidas del mismo sistema.

Supongamos que queremos construir un control realimentado para el flujo de un motor de inducción. Debido a que el flujo no es medible podríamos estimar las variables de estado desconocidas (flujo) con base en variables a las que sí podemos acceder, como la corriente en este caso. El observador, que es una réplica no exacta del sistema, primero da una estimación de la corriente. A continuación las corrientes estimadas son truncadas a partir de las corrientes medidas. Entonces el error es multiplicado por una ganancia de corrección que corrige la corriente estimada del observador. Así, el sistema real y el observador son sometidos a las mismas entradas y se puede esperar que a medida que pase el tiempo, los dos sistemas comienzan a comportarse de manera similar debido a que sus estados internos tienden a ser más parecidos. Es a través de este proceso que el observador se puede usar como una buena aproximación del sistema original.

La clave detrás de esto es la señal de error que se realimenta al modelo aproximado del sistema. Es importante mencionar que ambos sistemas tienen que ser estables para asegurar el funcionamiento de

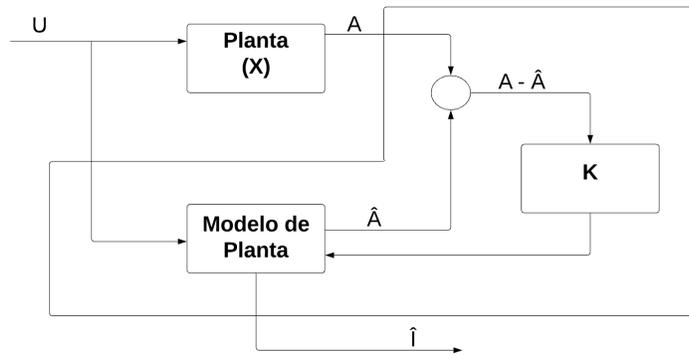


Figura 29: Esquema básico de un observador

esta técnica. El esquema básico de un observador se muestra en la [Figura 29](#).

Se puede observar que el error que se realimenta al modelo de la planta hace que el modelo se corrija y se ajuste para aproximarse a la planta real.

Las ventajas del estimador FAST de Texas Instruments y algunas de sus características se mencionan a continuación pero más detalle se puede encontrar en la referencia (SPRUHJ1G, 2013).

- FAST puede trabajar con las tres fases del motor y toma en consideración las diferencias entre motores y se ajusta.
- Mejora el desempeño y estabilidad del motor a lo largo de todo el rango de operación de frecuencia y carga para diferentes aplicaciones.
- Elimina el reto de afinar manualmente los sistemas FOC tradicionales.
- Elimina o reduce los efectos por la variación de los parámetros del motor.

La [Figura 30](#) muestra el esquema del FAST.

### 1.13 LÓGICA DIFUSA

Históricamente se ha pensado que el cerebro humano toma decisiones con base en una lógica limitada a dos resultados, sí/no. Tal es el caso que esta teoría fue condensada por George Boole en 1854, en su artículo, "An Investigation of the Laws of Thought" (Boole,2009). Sin embargo, las técnicas tradicionales de análisis de sistemas han estado por mucho tiempo atadas al análisis de sistemas físicos. Estos sistemas físicos a su vez están regidos por las leyes físicas mecánicas, de

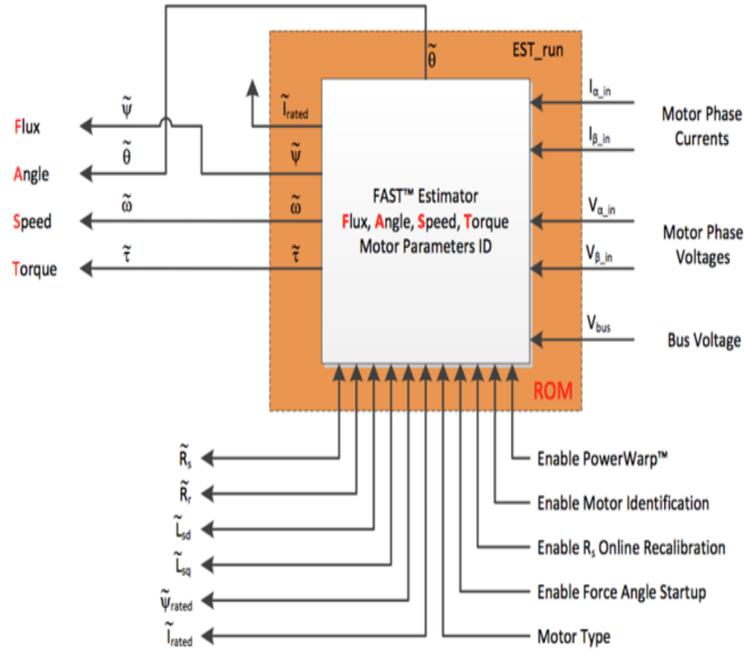


Figura 30: Esquema de entradas/salidas del estimador FAST

electromagnetismo y de termodinámica. El descubrimiento y entendimiento de estas leyes físicas nos ha llevado a desarrollar técnicas para resolver y controlar sistemas cada vez más complejos. Además, el avance en el poder computacional y las técnicas de simulación computacional han sido grandes impulsores de esta disciplina. Sin embargo, una de las condiciones de estas técnicas es el conocimiento de los modelos matemáticos que rigen a los sistemas. Los sistemas físicos modernos son tan complejos que intentar obtener sus modelos matemáticos resulta una tarea analítica no sólo impráctica sino imprecisa. A partir de estas observaciones y de analizar la manera empírica en que los seres humanos toman decisiones surgieron nuevas formas de plantear este problema.

Dada la tradición del pensamiento científico solemos medir nuestro entendimiento de un sistema de manera proporcional a nuestra capacidad de modelarlo. De la misma forma, hemos medido la capacidad de controlar un sistema siempre condicionado en el entendimiento del modelo matemático. Es sumamente complicado modelar sistemas no lineales, con muchas fuentes de ruido y con varios grados de acoplamiento.

En 1972 Lofti Zadeh propuso una nueva forma de analizar sistemas complejos. El problema surge a partir de la misma dificultad de querer controlar sistemas complejos de manera tradicional. La limitante consiste en que este tipo de control exige el conocimiento del modelo

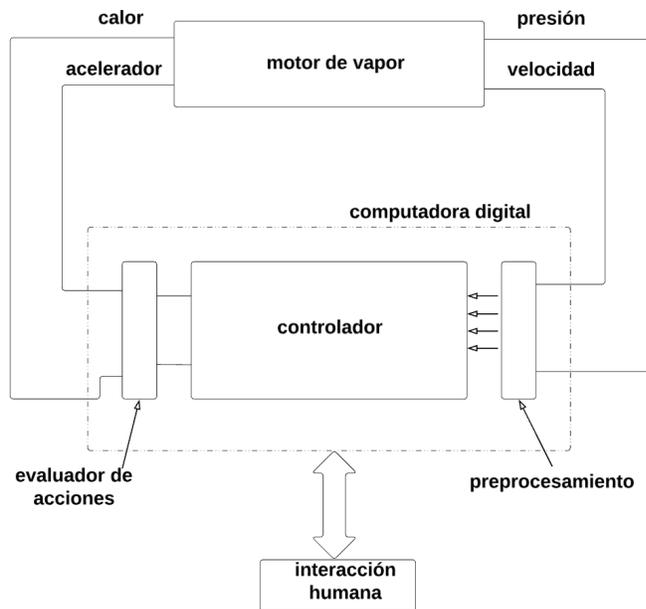


Figura 31: Experimento de Mamdani - Esquemático del sistema

matemático de la planta para poder controlar el sistema. Para esto propuso una nueva lógica cuyo principio establece lo siguiente; un elemento dentro de un conjunto puede tener varios estados, es decir, no limitado a 0 o 1 de acuerdo a la lógica Booleana. Zadeh propone el siguiente escenario; en el universo del discurso, es decir, todos los elementos dentro de un universo pueden, pertenecer a ciertos conjuntos en medida diferente (Zadeh, 1965). Este razonamiento se intenta acercar a la forma en que los humanos pensamos, clasificamos y tomamos decisiones.

Tomemos el siguiente ejemplo. Digamos que tenemos una piedra de cierto peso, 10kg. Ante el cuestionamiento, ¿cuánto crees que pesa esa piedra? La respuesta de una persona no sería intentar adivinar el peso real de la piedra, más bien el humano le asignaría variables lingüísticas al peso de esta piedra. Diría que es muy ligera, ligera, poco pesada, pesada o muy pesada. A falta de una herramienta para obtener un análisis cuantitativo y preciso de la piedra, el humano simplemente se basa en la experiencia y le asigna un valor a la variable lingüística peso. Bajo esta misma lógica, en 1974, Mamdani, desarrolló un algoritmo para controlar un motor de vapor construido en un laboratorio. Ante la dificultad de modelar este sistema complejo con una planta no lineal (Figura 31), se basó en la lógica difusa propuesta por Zadeh para controlar la planta. Esta planta que analiza Mamdani, es altamente interactiva y tiene dos entradas claramente cuantizadas: la entrada de ajuste de calor tiene 32 niveles y el acelerador tiene 10 niveles. Este algoritmo decide cuál es el cambio de ajuste en calor

y en el acelerador que debe haber para que el sistema llegue a una velocidad y a una presión de referencia. El algoritmo interpreta una serie de reglas que se definen con base en la experiencia de la operación de la planta. Por ejemplo, por simple experiencia de operación se sabe que si “el error en la presión del motor es alto positivo y si el cambio en el error es cero o cercano a cero entonces es necesario aplicar un ligero cambio en el calor”. Esta experiencia es la que se aplica en forma de control difuso. Primero se definen reglas de la forma “Si la temperatura es A y la presión es B entonces C”, y aquí es donde se condensa en conocimiento humano de la planta expresado en forma de lenguaje natural. A partir de estas reglas se definen las funciones de pertenencia que mapean los valores de las entradas y su correspondiente nivel de pertenencia a cierto conjunto, llámese CALIENTE o FRÍO para este caso. La forma de las funciones de pertenencia pueden tener diferentes formas pero la más tradicional es la forma triangular, que puede ser simétrica o asimétrica. La [Figura 32](#) representa la variable de temperatura usando conjuntos difusos.

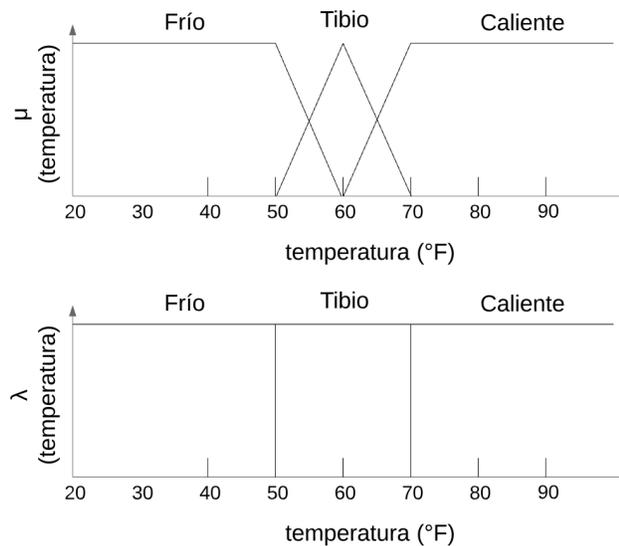


Figura 32: Ejemplos gráficos de funciones de pertenencia en un sistema difuso para la variable temperatura

Ahora, ya hemos hablado brevemente en qué consiste y cómo podemos aplicar la lógica difusa. La siguiente pregunta sería ¿Cómo se plantea el uso de la lógica difusa en un sistema? El siguiente paso ante este cuestionamiento es definir un sistema de inferencia difusa, o sistema difuso. Este problema se reduce a lo siguiente: formular el mapeo de un conjunto de variables de entrada a un conjunto de variables de salida, usando lógica difusa. Este mapeo es la base de un proceso de inferencia difuso y consiste de 5 pasos. Estos cinco pasos

son los pasos que se siguen en todos los procesos de inferencia, que realizan el mapeo entre entradas y salidas:

1. **Difusión** de las variables de entrada: a los valores absolutos de entrada se les asigna un valor difuso.
2. Aplicación de un **operador difuso** (AND, OR, NOT) en la parte condicional (IF) de las reglas. Este es el Antecedente.
3. **Implicación** del antecedente sobre la parte consecuente de la regla (THEN).
4. **Agregación** de las partes consecuentes de todas las reglas.
5. **Desdifusión** de la variable de salida para tener un valor absoluto.

Un ejemplo sencillo de este proceso se puede encontrar en (Bose,2002), el detalle de las funciones de pertenencia definidas se puede encontrar en la misma. Para fines ilustrativos se revisaran las generalidades de este ejemplo:

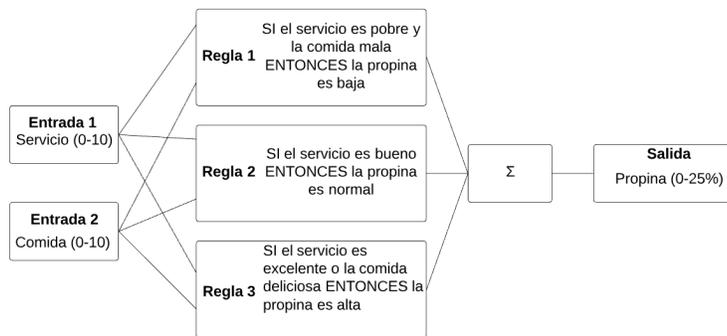


Figura 33: Esquema en bloques de un sistema difuso de dos entradas y una salida

El siguiente sistema difuso trata de resolver el problema de cuánta propina dejar un restaurante. Se tiene como entradas las variables "Servicio" y "Comida". La variable Comida a su vez está representada por los conjuntos "Mala" y "Deliciosa" y la variable Servicio tienen los conjuntos "Pobre", "Malo" y "Excelente". Por otro lado la variable de salida, "Propina" tienen los conjuntos "Baja", "Promedio" y "Generosa". Se definen los rangos de las entradas y salidas y se definen las reglas que definen al sistema. La Figura 33 es un resumen del sistema de inferencia difusa para dar propina en un restaurante.

Un punto importante que resulta una ventaja del control difuso es que el sistema de inferencia difuso y sus conjuntos difusos se pueden diseñar en términos de variables por unidad. Es decir, a cada variable se le aplica un factor de escala para normalizar el sistema. Es posible

posteriormente alterar los límites de las variables de entrada y salida simplemente normalizando a la entrada y desnormalizando a la salida. Este resulta conveniente ya que no hay necesidad de alterar el diseño del sistema difuso y el mismo diseño se puede aplicar a las plantas de la misma familia.

#### 1.14 CONTROL DIFUSO

Se puede hablar de control difuso cuando el algoritmo de control de algún proceso o sistema está basado en un sistema de inferencia difusa. En esencia entonces, un control difuso tiene embebida la experiencia e intuición humana como operador de una planta. Como ya se mencionó anteriormente, el diseño de un control tradicional está normalmente basado en un modelo matemático, concretamente en el modelo matemático de la planta. Una vez que se tienen un modelo suficientemente preciso se pueden realizar análisis para determinar la respuesta en frecuencia del modelo y entender a profundidad cómo se comporta para poder diseñar un control con ciertas especificaciones. La realidad es que para la mayoría de los sistemas en la práctica sus modelos son multi-variable, complejos y no lineales y suelen estar acoplados con muchas fuentes de ruido. Definir los modelos de estos sistemas es un proceso ineficiente y resulta en modelos muy alejados de los modelos reales. Muchas veces resulta el caso que a pesar de que es casi imposible obtener el modelo matemático del sistema sí se tiene el conocimiento de operación de la planta. Algunos ejemplos de estos sistemas son reactores nucleares y plantas de cemento. En este tipo de casos resulta mejor optar por otro tipo de control, y el control difuso es una de las opciones ante esta problemática.

Históricamente, el primero en poner en práctica la lógica difusa desarrollada por Zadeh fue Mamdani en 1974. Su objetivo era controlar un motor de vapor en el laboratorio. El problema consistía en regular la velocidad del motor y la presión del boiler a través del calor aplicado al boiler y al "throttle" del motor. El proceso para controlar este sistema resultaba complicado por ser no-lineal, ruidoso y altamente acoplado, además de no contar con el modelo matemático. Ante este problema Mamdani diseñó el primer control difuso únicamente utilizando el conocimiento empírico de operación de la planta (Mamdani, 1974).

La [Figura 34](#) muestra el esquema básico de un control difuso en un sistema realimentado. Los cuatro bloques del control difuso corresponden a los pasos definidos en la construcción de un sistema de inferencia difuso.

La metodología general de diseño de un control difuso se puede resumir en 10 pasos que se detallan en el libro de Bose (Mamdani, 1974).

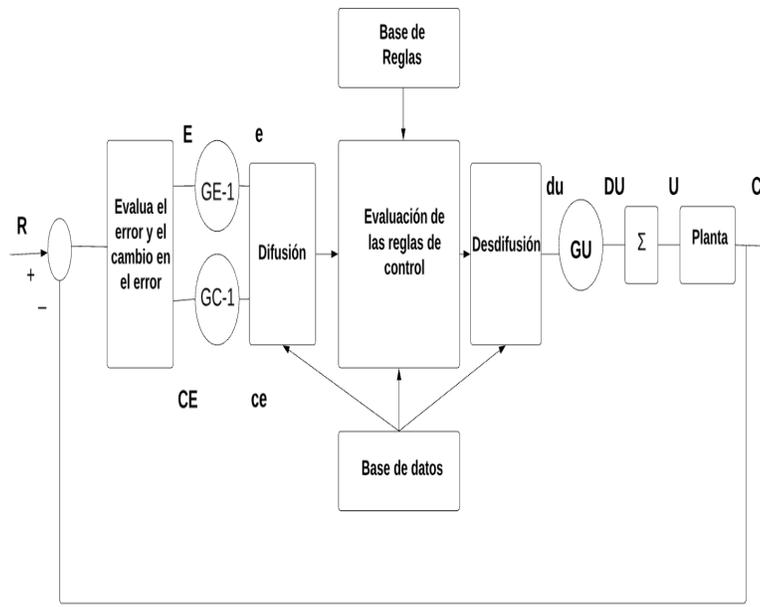


Figura 34: Esquema básico de un control difuso en un sistema realimentado

1. El primer paso es determinar si el control difuso es la mejor opción para un proceso y sistema determinado. En caso de que el modelo matemático se conozca y que las variaciones de los parámetros del sistema sean mínimos entonces el control difuso no tienen grandes ventajas.
2. Obtener toda la información del operador de la planta a controlar. De ser posible, la información de diseño y las características de operación también son importantes.
3. Si se cuenta con el modelo matemático de la planta realizar simulaciones con un control convencional para evaluar su desempeño.
4. Identificar los elementos funcionales donde se puede aplicar la lógica difusa.
5. Identificar las variables de entrada y salida para cada sistema difuso.
6. Definir el universo del discurso de cada una de las variables y convertirlas en variables por unidad una vez identificados los límites de cada variable involucradas.
7. Formular los conjuntos difusos y seleccionar las formas de las funciones de pertenencia correspondientes a cada conjunto. Para una variable sensible se necesita un mayor número de conjuntos. Si la variable requiere un mayor grado de precisión en

- estado estable entonces es necesario un mayor grado de amontonamiento de las funciones de pertenencia alrededor del origen.
8. Formular la tabla de reglas, con todas las combinaciones posibles. Es decir, mapear las combinaciones de todos los conjuntos de las variables de entrada con el correspondiente conjunto de salida. Este paso es el principal paso de diseño, donde se requiere la experiencia e intuición de los operadores de la planta.
  9. Simular el sistema con el control difuso si es que se tiene el modelo matemático de la planta e iterar los conjuntos y la tabla de reglas con el fin de optimizar el desempeño.
  10. Implementar el control en tiempo real e iterar para mejorar su desempeño.

Se puede encontrar un ejemplo de un control difuso que sigue la metodología presentada. El ejemplo se encuentra en el libro de Bose (Bose,2002) y se detalla todo el proceso de diseño para controlar la velocidad de un motor de inducción.

#### 1.15 AJUSTE DE PARÁMETROS

En el contexto de la familia de controladores PID el ajuste de parámetros consiste en ajustar las constantes (proporcional, integral y derivativa) del controlador para lograr cierto comportamiento en la respuesta del sistema. En muchos casos los ajustes en el control responden a un comportamiento esperado de la respuesta del sistema en términos de parámetros como sobrepaso máximo, tiempo de levantamiento y asentamiento. Existen muchos métodos que proponen ciertas reglas y procedimientos para ajustar un control. Muchos controladores se ajustan manualmente a través de pruebas y experiencia, sin embargo, existen algunos métodos que proponen una serie de pasos y reglas para realizar el ajuste; Ziegler-Nichols y Cohen-Coon son ejemplos de métodos de ajuste. El ajuste de parámetros en este tipo de controladores continua siendo un tema de estudio relevante.

Contrario a los controles PID, los difusos tienen características no-lineales que hace que resulte más complicado que existan reglas generales para ajuste de parámetros. Adicionalmente, los controladores difusos tienen muchos parámetros y reglas que se pueden ajustar y optimizar, cosa que complica aún más la optimización. A pesar de lo dicho, sí existen algunas metodologías de ajuste para controles difusos. Normalmente estas metodologías se basan en la aplicación de métodos y reglas lineales a partir del cumplimiento de ciertas condiciones (Strietzel,2000). El estudio de acondicionamientos, métodos de linealización y técnicas generales de ajustes de parámetros de controladores difusos continúan siendo motivo de estudio e interés en el campo del control difuso.

Cómo parte de los intentos de optimización de la respuesta de la variable de salida se decidió aplicar un método heurístico para afinar controles difusos con base en un algoritmo que modifica las funciones de pertenencia de las variables de entrada (Gómez, 2004). En este trabajo se modificaron las funciones de pertenencia, mayor concentración de reglas alrededor del origen, como ajuste para evaluar la respuesta. Sin embargo, como se mostrará a detalle en capítulos posteriores, este método de ajuste no represento mejoras visibles en la respuesta del sistema. Dados los resultados, y reconociendo que un estudio a fondo de ajustes de parámetros ameritaría todo un trabajo aparte, nos limitamos a realizar un ajustes manuales de la ganancia y la constante de integración del sistema.



A partir de los planteamientos teóricos del capítulo pasado, ahora es necesario describir todos los pasos en el proceso de la implementación, desde la selección de los componentes hasta la medición de los resultados. Este capítulo ofrece, con detalle, la selección, el acondicionamiento, modificación y el uso que se le da a todos los componentes que llevaron a la implementación y pruebas del sistema del control difuso.

Este capítulo busca también hacer constante referencia entre cada componente de la implementación, y su componente teórico ya discutido. Se hará especial énfasis en el sistema *InstaSpin-FOC* ya que es central para la implementación del sistema de control difuso en este trabajo. Este sistema, propiedad de *Texas Instruments* (TI), viene precargado en algunos microcontroladores y resuelve los dos problemas más importantes en la implementación de un control difuso para un BLDC. El primer problema tiene que ver con el sensado de las variables de interés del motor para fines de control. Este sistema implementa un control de campo orientado sin el uso de sensores. Básandonos en ese mismo sistema, parte del trabajo detallado en este capítulo consiste en la modificación de ciertas librerías de *InstaSpin-FOC* para sustituir, un sistema de control proporcional-integral (PI) por un sistema de control difuso. El segundo problema que resuelve el sistema de *TI* tiene que ver con que la tarjeta que lo integra se puede acoplar con ciertos sistemas de conmutación electrónica de la misma marca y así simplifica en gran parte el problema de alimentación al motor. Adicionalmente, se aprovecha el uso de varios laboratorios que ofrece *Motorware* (paquete de software para desarrollar aplicaciones con *InstaSpin-FOC*) para la caracterización del motor y la modificación de las librerías donde se encuentra el control.

Para fines comparativos y de resultados contra el control difuso propuesto, se considera el desempeño del control disponible en el sistema *InstaSpin-FOC* cuyo tipo de control es PI. Es por ello que otra parte de este capítulo se dedica al estudio del control PI disponible, para más adelante compararlo contra el control difuso que propone este trabajo.

## 2.1 HERRAMIENTAS EMPLEADAS

Como primer paso para realizar el sistema de control de velocidad de un motor sin escobillas de CD, planteamos los elementos necesarios para formarlo partiendo del siguiente diagrama:

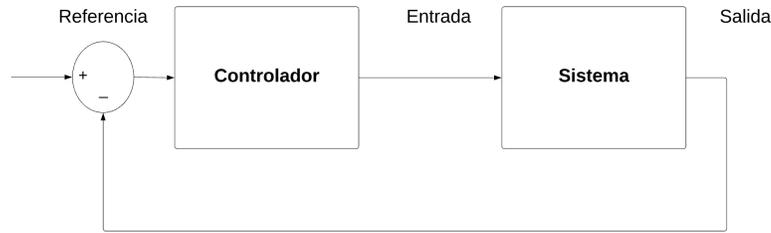


Figura 35: Sistema básico de control realimentado

De la [Figura 35](#) se observa que se necesita:

1. **Controlador:** Unidad de procesamiento para el control del sistema
2. **Sistema:** Motor BLDC
3. **Fuente de Alimentación:** no se indica explícitamente pero se requiere para que el sistema completo funcione.

Utilizamos un motor BLDC de 12[V] genérico con velocidad nominal de 12[krpm]. Este tipo de motor se puede encontrar comúnmente en aplicaciones para drones. En este caso, los motores empleados fueron los usados en los drones de tipo Quadcopter. Estos motores tienen como particularidad que el rotor, construido con imanes permanentes, se encuentra alrededor del estator. Adicional al motor, se emplean en este trabajo diferentes componentes de software y hardware para realizar la implementación completa del sistema. Se puede encontrar más información sobre estos componentes en el apéndice del trabajo.

Se decidió usar, para la parte de procesamiento, una tarjeta (microcontrolador) de *Texas Instruments*. Las tarjetas *Piccolo* de TI son tarjetas baratas (en relación a sus homólogas de otras marcas) de la familia *C2000*, familia de microcontroladores de 32 bits. Esta tarjeta tiene como una de sus aplicaciones principales el control de motores. Las virtudes principales de esta tarjeta son las integraciones que tienen con la oferta de Drivers de TI y software integrado para control de motores sin sensores que viene precargado en la memoria ROM de la tarjeta. La tarjeta que se seleccionó fue la *LaunchXL-F28027*. Esta tarjeta cuenta con el microprocesador *TMS320F28027* que a su vez tiene las siguientes características y capacidades de procesamiento:

- 60[MHz] (periodo: 16,67[ns])
- 16x16 y 32x32 Operaciones MAC

El siguiente elemento a considerar fue el driver que conforma la electrónica de potencia empleada para alimentar al motor. Se eligió el

driver *BOOSTXL-DRV8301* de TI. Este es un driver trifásico sin sensores de CD, de 10[A]. El módulo completo fue diseñado para trabajar en conjunto con cualquier *Launchpad XL* y con el software de *InstaSpin-FOC* que viene integrado con la tarjeta *LaunchXL-F28027F*. Estas características hacen a este driver ideal para trabajar con técnicas de control sin sensores. Algunas de las características técnicas más importantes de este módulo se muestran a continuación:

- De 6 a 24[V] de entrada
- Soporta hasta 14[A] de picos de corriente, y hasta 10[A] de corriente constante en la salida
- Soporta realimentación de voltaje y corriente para la solución sin sensores de corriente *InstaSpin-FOC*
- La etapa del driver cuenta con protección de corto circuito y protección térmica, entre otras.

Texas Instruments, dentro de su oferta de soluciones para el control de motores, ofrece una solución llamada *InstaSpin-FOC*. *InstaSpin-FOC* es software que emplea la técnica de control conocida como control de orientación de campo, que es una de las técnicas de control más usadas y más eficientes. *InstaSpin-FOC* te permite identificar, afinar y controlar cualquier tipo de motor síncrono o asíncrono trifásico, sin sensores, y de velocidad variable. Esta tecnología elimina la necesidad de usar sensores y reduce los costos del sistema a través del uso de un observador en forma de algoritmo de software llamado FAST (que mide el flujo, ángulo, velocidad y torque) para lograr un control sin sensores. Este software, FAST, está embebido en la memoria ROM de las tarjetas *Piccolo*. Los siguientes son los beneficios principales de usar *InstaSpin-FOC*:

- Se reemplazan los encoders mecánicos y solvers en un control FOC sin sensores. En lugar de estos encoders mecánicos se utiliza el encoder de software FAST, que resulta en un sensor de flujo del rotor superior.
- El diseño del control se hace más sencillo ya que *InstaSpin-FOC* tiene un sistema de identificación de parámetros del motor.
- El sistema de control completo *InstaSpin-FOC* se encuentra embebido en la ROM de ciertas tarjetas de TI.
- *InstaSpin-FOC* también cuenta con un sistema de arranque. El sistema arranca el motor desde un estado de reposo y se proporciona un torque máximo desde el mismo arranque y el control de lazo cerrado con FAST proporciona el ángulo en menos de un ciclo eléctrico. El control de lazo cerrado comienza desde



Figura 36: Fotografía del estator del BLDC empleado en este trabajo

el arranque, sólo se obtiene la posición del ángulo inicial a través de algoritmos de detección de posición inicial. Esto elimina la necesidad de tener un sistema de lazo abierto al arrancar el motor y perder eficiencia en el arranque, y de diseñar e implementar métodos de arranque aislados.

- No es necesario tener la hoja de especificaciones del motor.

Para la parte de la simulación se empleó el software *Matlab*, específicamente el módulo de *Simulink*. Las razones detrás de esta elección tienen que ver con el sencillo acceso a bloques y modelos ya construidos dentro de las librerías. Estas librerías contienen bloques de los modelos de los motores y los bloques de control.

Adicional a esto, *Matlab* cuenta con un módulo (*Fuzzy Logic Toolbox*) para el diseño de controles difusos empleando el método de Mamdani.

## 2.2 CARACTERIZACIÓN DEL MOTOR

Para simular el sistema se necesitaba la caracterización del motor de CD sin escobillas al que se refiere este trabajo. Para la caracterización del motor, el primer paso que se realizó fue observar el número de polos del motor, resultando en 12 polos.

Como parte del *driver 8301* y su uso con *InstaSpin-FOC*, se incluye *MotorWare*, la cual es un conjunto de bibliotecas y aplicaciones dirigidas al uso de *InstaSpin-FOC* y su programación con alguna *launchpad* para el control de motores.

No se contaba con ningún parámetro del motor, por lo que los pasos que siguieron, se tomaron con base en laboratorios de *MotorWare*

(SPRUHJ1G, 2013), específicamente tutoriales para el control de cualquier motor. Los tutoriales se acompañan con programas en lenguaje C diseñados para la familiarización con los motores BLDC y su control.

De los tutoriales se continuó con el uso de **InstaSpin\_UNIVERSAL.exe**, que es una aplicación incluida en MotorWare que calcula los parámetros de un motor, como son la resistencia en serie característica del estator por fase, la inductancia en serie y el offset de corriente. Su uso consiste en conectar las tarjetas de procesamiento y el driver de potencia (*Piccolo 2000* y *Driver 8301*) con la fuente de alimentación al motor y la aplicación arroja los resultados calculados.

La [Figura 37](#) muestra el sistema completo conectado a una fuente de alimentación.



Figura 37: Fotografía del sistema completo: Fuente de alimentación de CD, microcontrolador, driver, y el motor BLDC

Ya conectado todo el sistema, se procedió a enlazarlo a la computadora y a abrir la aplicación **InstaSpin\_UNIVERSAL.exe**. Se cargó el archivo **appProgram.out** a la memoria de la tarjeta *Piccolo* y se seleccionó inicializar, en caso de tener una inicialización correcta los mensajes desplegados deben ser como los que se muestran en la [Figura 38](#).

A continuación se procede a hacer el cálculo de los parámetros del motor, se seleccionan las casillas de:

- **Enable System:** Para habilitar el sistema.



Figura 38: Ventana de aplicación MotorWare al cargar exitosamente a la memoria de la tarjeta el programa *InstaSpin-FOC*



Figura 39: Casillas habilitadas para el cálculo de los parámetros del motor

- **Run:** Para ejecutar el programa.
- **RsRecalc:** Para el cálculo de la resistencia en serie.
- **OffsetCalc:** Para el cálculo del offset del CAD (convertidor analógico/digital).
- **ForceAngle:** No es necesario tener la hoja de especificaciones del motor.

La aplicación de **InstaSpin\_UNIVERSAL.exe** terminó de identificar el motor y arrojó los resultados que se observan a continuación:

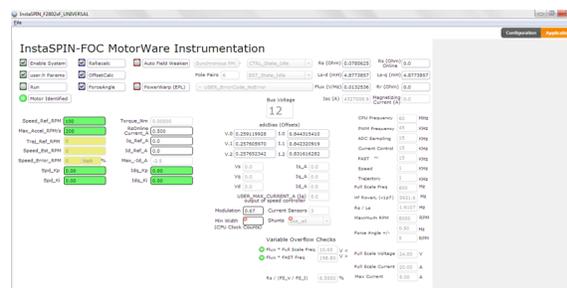


Figura 40: Ventana de Motorware dónde se muestran los parámetros del motor calculados.

De donde se obtienen los siguientes valores característicos:

- **USER\_MOTOR\_NUM\_POLE\_PAIRS:** (6)
- **USER\_MOTOR\_Rs:** (0,07758751511573792)
- **USER\_MOTOR\_Ls\_d:** (0,000011070956134062726)
- **USER\_MOTOR\_Ls\_q:** (0,000011070956134062726)
- **USER\_MOTOR\_RATED\_FLUX:** (0,00505314115434885)
- **USER\_MOTOR\_RES\_EST\_CURRENT:** (0,5)
- **USER\_MOTOR\_IND\_EST\_CURRENT:** (-0,5)
- **USER\_MOTOR\_MAX\_CURRENT:** (2,0)



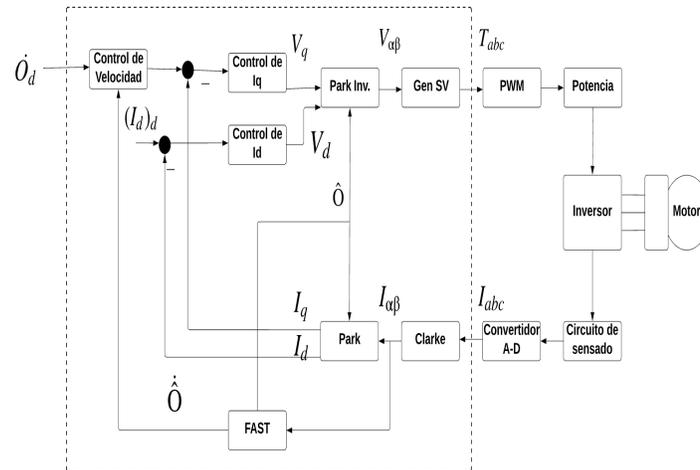


Figura 42: Todos los bloques del FOC incluidos en el módulo de control.

Se puede observar que el módulo tiene como entrada la velocidad del motor de referencia y los valores de corriente que salen del CAD. Como salida se tienen los voltajes de ajuste que alimentan el PWM.

El proceso que se explica a continuación se puede encontrar con más detalles en el laboratorio 11 (“InstaSPIN Projects and Labs User’s Guide”, s.f.) de la *instaSpin\_lab*. Ahora se buscará explicar cómo se puede acceder a los elementos del FOC y a las funciones alojadas en ROM que nos dan acceso al estimador FAST. Este proceso intenta ser un “approach” simple para exponer todo el FOC dentro de la ISR. El proceso se puede dividir en tres partes: Inicialización del estimador, el lazo que corre en el fondo y la ISR principal. Más detalle de cada uno de los pasos se puede ver a continuación:

1. **Inicialización del estimador** En esta etapa se configura el estimador y se prepara para ser habilitado, además se le pasan los parámetros del motor, previamente calculados.

---

```

1 // initialize the estimator
  estHandle = EST_init((void *)USER_EST_HANDLE_ADDRESS, 0x200)
  ;

  //initialize the user parameters
  USER_setParams(&gUserParams);

6 // set the hardware abstraction layer parameters
  HAL_setParams(hal_HANDLE, &gUserParams);

  //initialize the estimator
11 EST_setEstParams(estHandle, &gUserParams);

  EST_setupEstIdleState(estHandle);

```

---

2. **Lazo de Fondo** Este lazo conforma algunas líneas de código que habilitan o deshabilitan el estimador y además prenden y apagan el PWM.

---

```

// loop while the enable system flag is torque
2 while(gMotorVars.Flag_enableSys)
  {
    if(gMotorVars.Flag_Run_Identify)
    {
      // Update estimator state
7      EST_updateState(estHandle,0);

      // enable the PWM
      HAL_enablePwm(halHandle);
    }
12  else
    {
      //set estimator to Idle
      EST_estIdle(estHandle);

17      //disable PWM
      HAL_disablePwm(halHandle);

      //clear integral outputs
      PID_setUi(pidHandle[0],_IQ(0.0));
22      PID_setUi(pidHandle[1],_IQ(0.0));
      PID_setUi(pidHandle[2],_IQ(0.0));

      // clear Id and Iq references
      gIdq_ref_pu.value[0] = _IQ(0.0);
27      gIdq_ref_pu.value[1] = _IQ(0.0);

    }
    // end of while(gFlag_enableSys) loop
  }

```

---

### 3. ISR Principal

En esta ISR principal no hay objeto de control. Todo el FOC está expuesto y sigue la implementación estándar de un FOC usando archivos fuente de MotorWare, que pueden ser modificados por el usuario. Para fines de este trabajo, donde buscamos modificar el esquema de control, utilizamos esta implementación ya que nos permite hacer los cambios al bloque de una manera más sencilla. La ISR principal a su vez se divide en varias etapas:

- *FOC Directo*: En esta etapa se obtienen las conversiones del CAD, compensación de offsets y las transformadas de Clarke de corriente y voltaje.
-

```

//acknowledge the ADC interrupt
HAL_acqAdcInt(halHandle,ADC_IntNumber_1);

4 //convert the ADC data
HAL_readAdcDataWithOffsets(halHandle,&gAdcData);

//remove offsets
gAdcData.I.value[0] = gAdcData.I.value[0] -
    gOffsets_I_pu.value[0];
9 gAdcData.I.value[1] = gAdcData.I.value[1] -
    gOffsets_I_pu.value[1];
gAdcData.I.value[2] = gAdcData.I.value[2] -
    gOffsets_I_pu.value[2];
gAdcData.V.value[0] = gAdcData.V.value[0] -
    gOffsets_V_pu.value[0];
gAdcData.V.value[1] = gAdcData.V.value[1] -
    gOffsets_V_pu.value[1];
gAdcData.V.value[2] = gAdcData.V.value[2] -
    gOffsets_V_pu.value[2];
14
// run Clarke transform on current
CLARKE_run(clarkeHandle_I,&gAdcData.I,&Iab_pu);

// run Clarke transform on voltage
19 CLARKE_run(clarkeHandle_V,&gAdcData.V,&Iab_pu);

```

- *Ejecución del Estimador:* Esta es la función que llama a ROM para ejecutar el estimador.

```

// run the estimator
EST_run(estHandle,
        &Iab_pu,
        &Vab_pu,
5        gAdcData.dcBus,
        gMotorVars.speedRef_pu);

```

- *Extraer Variables Estimadas:* con estas funciones se obtienen las variables que se controlarán en el FOC. Las variables son ángulo, velocidad,  $I_d$  e  $I_q$ .

```

// generate the motor electrical angle
angle_pu = EST_getAngle_pu(estHandle);
3 speed_pu = EST_getFm_pu(estHandle);

//get Idq from estimator to avoid sin and cos
EST_getIdq_pu(estHandle,&gIdq_pu);

```

- *Control de Velocidad:* Aquí se hace una llamada al control de velocidad. Aquí se obtiene el valor de una variable global y un valor de la velocidad estimada

```

// when appropriate, run the PID speed controller
if(pidCntSpeed++ >= USER_NUM_CTRLTICKS_PER_SPEED_TICK)

```

```

3  {
    // clear counter
    pidCntSpeed = 0;

    // run speed controller
8  PID_run_spd(pidHandle[0],
              gMotorVars.SpeedRef_pu,
              speed_pu,
              &(gIdq_ref_pu.value[1]));
  }

```

---

- *Control de  $I_d$*

---

```

// get the reference value
2  refValue = gIdq_ref_pu.value[0];

// get the feedback value
    fbackValue = gIdq_pu.value[0];

7 //run the Id PID controller
    PID_run(pidHandle[1], refValue, fbackValue, &(gVdq_out_pu.
        value[0]));

```

---

- *Control de  $I_q$*

---

```

1 //get the Iq reference value
    refValue = gIdq_ref_pu.value[1];

//get the feedback value
    fbackValue = gIdq_pu.value[1];

6 //calculate Iq controller limits, and run Iq controller
    outMax_pu = _IQsqrt(_IQ(USER_MAX_VS_MAG_PU *
        USER_MAX_VS_MAG_PU) - _IQmpy(gVdq_out_pu.value[0],
        gVdq_out_pu.value[0]));
    PID_setMinMax(pidHandle[2], -outMax_pu, outMax_pu);
    PID_run(pidHandle[2], refValue, fbackValue, &(gVdq_out_pu.
        value[1]));

```

---

Cabe mencionar que previo a ejecutar el controlador de  $I_q$ , un límite de salida PID se calcula con base en el vector de voltaje de salida disponible y la velocidad máxima especificada por el usuario.

- *FOC Inverso* Una vez que se obtienen  $V_d$  y  $V_q$  se aplican las transformadas inversas, para esto primero se calcula el ángulo de salida.
- 

```

// compensate angle for the PWM delay
    angle_pu = angleDelayComp(speed_pu, angle_pu);
// compute the sin/cos phasor
    phasor.value[0] = _IQcosPU(angle_pu);
5  phasor.value[1] = _IQsinPU(angle_pu);

```

```

//set the phasor in the inverse Park transform
IPARK_setPhasor(iparkHandle,&phasor);
// run the space Vector Generator (SVGGEN) module
10 oneOverDcBus = EST_getOneOverDcBus_pu(estHandle);
Vab_pu.value[0] = _IQmpy(Vab_pu.value[0],oneOverDcBus);
Vab_pu.value[0] = _IQmpy(Vab_pu.value[0],oneOverDcBus);
SVGGEN_run(svggenHandle,&Vab_pu,&(gPwmData.Tabc));

```

Escribir los valores al PWM: Al terminar la ISR los valores calculados se escriben sobre el módulo PWM:

```

// write the PWM compare values
2 HAL_writePwmData(halHandle,&gPwmData);

```

### 2.3.1 Diseño del Control Difuso

En esta sección primero se buscara definir límites de las variables de entrada y salida, y la definición de las reglas de pertenencia.

Para diseñar el control difuso se establecieron las entradas del sistema que son el error y la derivada del error. El error es la velocidad de referencia menos la velocidad medida por el sistema.

Para tener un manejo más amigable de los datos, se optó por normalizar las entradas, siendo la constante para este sistema  $16198686$  y el doble de esta para la derivada del error. El motivo de que el rango de la derivada del error sea de  $-2$  a  $2$  se demuestra a continuación:

Si  $Error_{actual} = 1$  y  $Error_{previo} = -1$

$$\frac{dError}{dt} = Error_{actual} - Error_{previo} = 1 - (-1) = 2 \quad (15)$$

Si  $Error_{actual} = -1$  y  $Error_{previo} = 1$

$$\frac{dError}{dt} = Error_{actual} - Error_{previo} = -1 - 1 = -2 \quad (16)$$

Como salida del sistema se tiene la corriente  $I_q$  que para este sistema tiene valores de  $-1800000$  a  $1800000$  y de igual manera como se realizó para las entradas del sistema, se decidió normalizarlo.

Las constantes de normalización mencionadas previamente fueron obtenidas experimentalmente mediante el uso del control PI que incluye el laboratorio 11 (“InstaSPIN Projects and Labs User’s Guide”, s.f.) de los tutoriales que se incluyen con la tarjeta para el uso del FAST, se tomó lectura los valores medidos con ayuda de la interfaz gráfica de CCS (Code Composer Studio) y dado que estos valores no se alteran por el control difuso se decidió usarlos.

A continuación se muestran las entradas y salidas del sistema difuso, que es diferente a las entradas y salidas del sistema de control de velocidad, discrepando únicamente la salida del sistema difuso con la del sistema de control de velocidad, que es la derivada de la velocidad y la cual tiene una proporción directa en la modificación de  $I_q$ .

**Entradas:**

- $\text{Error} = \text{Velocidad}_{\text{referencia}} - \text{Velocidad}_{\text{medida}} [-1, 1]$
- $\frac{d\text{Error}}{dt} = \text{Error}_{\text{actual}} - \text{Error}_{\text{previo}} [-2, 2]$

**Salidas:**

- $I_q [-1, 1]$

Si el Error es negativo implica que el motor tiene una velocidad mayor a la deseada, si es positivo el motor tiene una velocidad menor a la deseada.

La derivada del error no es dividida entre el tiempo debido a que en cada iteración el tiempo es constante, si la derivada es positiva quiere decir que el motor está desacelerando, al contrario si es negativa el motor acelera.

La salida de corriente  $I_q$  representa que a corrientes positivas, el motor gira en sentido horario; a corrientes negativas el motor gira en sentido anti horario. Entre más alta sea la corriente mayor revoluciones por minuto tiene el motor. Cuando la corriente es cero el motor no gira.

Para el diseño del control difuso se optó por probar dos sistemas de reglas diferentes, con cinco conjuntos para el error cada uno, así mismo, para la derivada del error se tienen también cinco conjuntos:

- Ecuaciones de difusión para el sistema 1:

**[-1 , -0.9]**

$$a = 1 \quad (17)$$

**[-0.9 , -0.5]**

$$a = \frac{-5}{2}\text{Error} - \frac{5}{4} \quad (18)$$

**[-0.9 , -0.5]**

$$b = \frac{5}{2}\text{Error} + \frac{9}{4} \quad (19)$$

**[-0.5 , 0]**

$$b = -2\text{Error} \quad (20)$$

[-0.5 , 0]

$$c = 2\text{Error} + 1 \quad (21)$$

[0 , 0.5]

$$c = -2\text{Error} + 1 \quad (22)$$

[0 , 0.5]

$$d = 2\text{Error} \quad (23)$$

[0.5 , 0.9]

$$d = \frac{-5}{2}\text{Error} + \frac{9}{4} \quad (24)$$

[0.5 , 0.9]

$$e = \frac{5}{2}\text{Error} - \frac{5}{4} \quad (25)$$

[0.9 , 1]

$$e = 1 \quad (26)$$

[-2 , -1.8]

$$f = 1 \quad (27)$$

[-1.8 , -1]

$$f = \frac{-5}{4} \frac{d\text{Error}}{dt} - \frac{5}{4} \quad (28)$$

[-1.8 , -1]

$$g = \frac{5}{4} \frac{d\text{Error}}{dt} + \frac{9}{4} \quad (29)$$

[-1 , 0]

$$g = -\frac{d\text{Error}}{dt} \quad (30)$$

[-1 , 0]

$$h = \frac{d\text{Error}}{dt} + 1 \quad (31)$$

[0 , 1]

$$h = -\frac{d\text{Error}}{dt} + 1 \quad (32)$$

[0, 1]

$$i = \frac{d\text{Error}}{dt} \quad (33)$$

[1, 1.8]

$$i = \frac{-5}{4} \frac{d\text{Error}}{dt} + \frac{9}{4} \quad (34)$$

[1, 1.8]

$$j = \frac{5}{4} \frac{d\text{Error}}{dt} - \frac{9}{4} \quad (35)$$

[1.8, 2]

$$j = 1 \quad (36)$$

- Ecuaciones para el sistema 2:

[-1, -0.9]

$$a = 1 \quad (37)$$

[-0.9, -0.2]

$$a = -\frac{10}{7} \frac{d\text{Error}}{dt} - \frac{2}{7} \quad (38)$$

[-0.7, -0.2]

$$b = 2\text{Error} + \frac{7}{5} \quad (39)$$

[-0.2, 0]

$$b = -5\text{Error} \quad (40)$$

[-0.2, 0]

$$c = 5\text{Error} + 1 \quad (41)$$

[0, 0.2]

$$c = -5\text{Error} + 1 \quad (42)$$

[0, 0.2]

$$d = 5\text{Error} \quad (43)$$

[0.2 , 0.7]

$$d = -2\text{Error} + \frac{7}{5} \quad (44)$$

[0.2 , 0.9]

$$e = \frac{10}{7}\text{error} - \frac{2}{7} \quad (45)$$

[0.9 , 1]

$$e = 1 \quad (46)$$

[-2 , -1.8]

$$f = 1 \quad (47)$$

[-1.8 , -0.4]

$$f = \frac{-5}{7} \frac{d\text{Error}}{dt} - \frac{2}{7} \quad (48)$$

[-1.4 , -0.4]

$$g = \frac{d\text{Error}}{dt} + \frac{7}{5} \quad (49)$$

[-0.4 , 0]

$$g = \frac{-5}{2} \frac{d\text{Error}}{dt} \quad (50)$$

[-0.4 , 0]

$$h = \frac{5}{2} \frac{d\text{Error}}{dt} + 1 \quad (51)$$

[0 , 0.4]

$$h = \frac{-5}{2} \frac{d\text{Error}}{dt} + 1 \quad (52)$$

[0 , 0.4]

$$i = \frac{5}{2} \frac{d\text{Error}}{dt} \quad (53)$$

[0.4 , 1.4]

$$i = -\frac{d\text{Error}}{dt} + \frac{7}{5} \quad (54)$$

1-13	14-25
T1 trian(-1.09,-1,-0.91)	T14 trian(0,0.08,0.16)
T2 trian(-0.99,-0.91,-0.83)	T15 trian(0.08,0.16,0.24)
T3 trian(-0.91,-0.83,-0.75)	T16 trian(0.16,0.25,0.34)
T4 trian(-0.84,-0.75,-0.66)	T17 trian(0.25,0.33,0.41)
T5 trian(-0.74,-0.66,-0.58)	T18 trian(0.33,0.42,0.51)
T6 trian(-0.66,-0.58,-0.5)	T19 trian(0.42,0.5,0.58)
T7 trian(-0.58,-0.5,-0.42)	T20 trian(0.5,0.58,0.66)
T8 trian(-0.51,-0.42,-0.33)	T21 trian(0.58,0.66,0.74)
T9 trian(-0.41,-0.33,-0.25)	T22 trian(0.66,0.75,0.84)
T10 trian(-0.34,-0.25,-0.16)	T23 trian(0.75,0.83,0.91)
T11 trian(-0.24,-0.16,-0.08)	T24 trian(0.83,0.91,0.99)
T12 trian(-0.16,-0.08,0)	T25 trian(0.91,1,1.09)
T13 trian(-0.08,0,0.08)	-

Cuadro 1: Conjunto de Reglas

[0.4 , 1.8]

$$j = \frac{5}{7} \frac{dError}{dt} - \frac{2}{7} \tag{55}$$

[1.8 , 2]

$$j = 1 \tag{56}$$

Para el incremento del voltaje (la salida del sistema) se tienen 25 reglas, que es el resultado de la multiplicación del número conjuntos para cada entrada (5x5). En este conjunto de reglas se intentó tener un comportamiento diferente recorriendo el punto medio de las funciones, pero no representó ningún cambio notable por lo que no se muestra en este trabajo. Las reglas se muestran a continuación:

Siendo **trian** una función con forma triangular con las tres aristas indicadas como puntos verticales en una gráfica.

De esta forma se tienen 25 reglas de la forma IF X<sub>1</sub> es A AND X<sub>2</sub> es B THEN Y es C.

Para la jerarquización de las reglas se realizó un análisis en el que basado en nuestra experiencia decidimos los casos de mayor criticidad, así como los casos más estables y a partir de ahí se jugó con el orden, dando por resultado las reglas que pueden observarse en la [Tabla 2](#).

<i>error/Derror</i>	<b>MN</b>	<b>N</b>	<b>DCero</b>	<b>P</b>	<b>MP</b>
<b>MNG</b>	T1	T3	T7	T11	T9
<b>NG</b>	T2	T4	T8	T12	T10
<b>Cero</b>	T5	T6	T13	T20	T21
<b>PG</b>	T16	T14	T18	T22	T24
<b>MPG</b>	T17	T15	T19	T23	T25

Cuadro 2: Tabla de 5x5 con la jerarquización de las reglas con base en los valores que toman las variables Error y  $\frac{dError}{dt}$

Se utilizó la implicación de Mamdani así como la desdifusión de promedios ponderados, lo que facilitó el diseño.

Para entregar el valor de  $I_q$  es necesario multiplicarlo por una constante para que represente un valor que tenga peso en la variación del sistema, esta constante se eligió dependiendo de la velocidad deseada, siendo ,001 para valores de velocidad menores a 5[krpm] y 100 para velocidades superiores a esta.

El programa de control puede verse en el apéndice.

Es notorio que en la implementación del sistema de inferencia difuso se omitieron dos pasos: la *implicación* y la *agregación*, pero debido a que se usa la desdifusión de promedios ponderados estos dos pasos se realizan implícitamente.

#### 2.4 SIMULACIÓN CON EL CONTROL DISEÑADO EN MATLAB

El modelo de la simulación se construyó en Simulink y se insertó el archivo *.fis*, en el bloque del Controlador Difuso. El esquema del modelo se muestra en la [Figura 43](#).

Los resultados que se muestran en la [Figura 44](#) son de una corrida del modelo con velocidad de referencia de 500[Hz].

La [Figura 45](#) es un acercamiento de la figura anterior con el fin de ver con más detalle los rizados de los trazos de velocidad.

Se observa que la velocidad del rotor oscila alrededor de la velocidad de referencia  $\pm 5$ [rpm]. Esta oscilación por arriba y por debajo de la velocidad de referencia representa un error máximo del 1 % del valor de referencia.

Como segunda iteración se modificaron las reglas de pertenencia para poder lograr una disminución en el error de la velocidad del rotor.

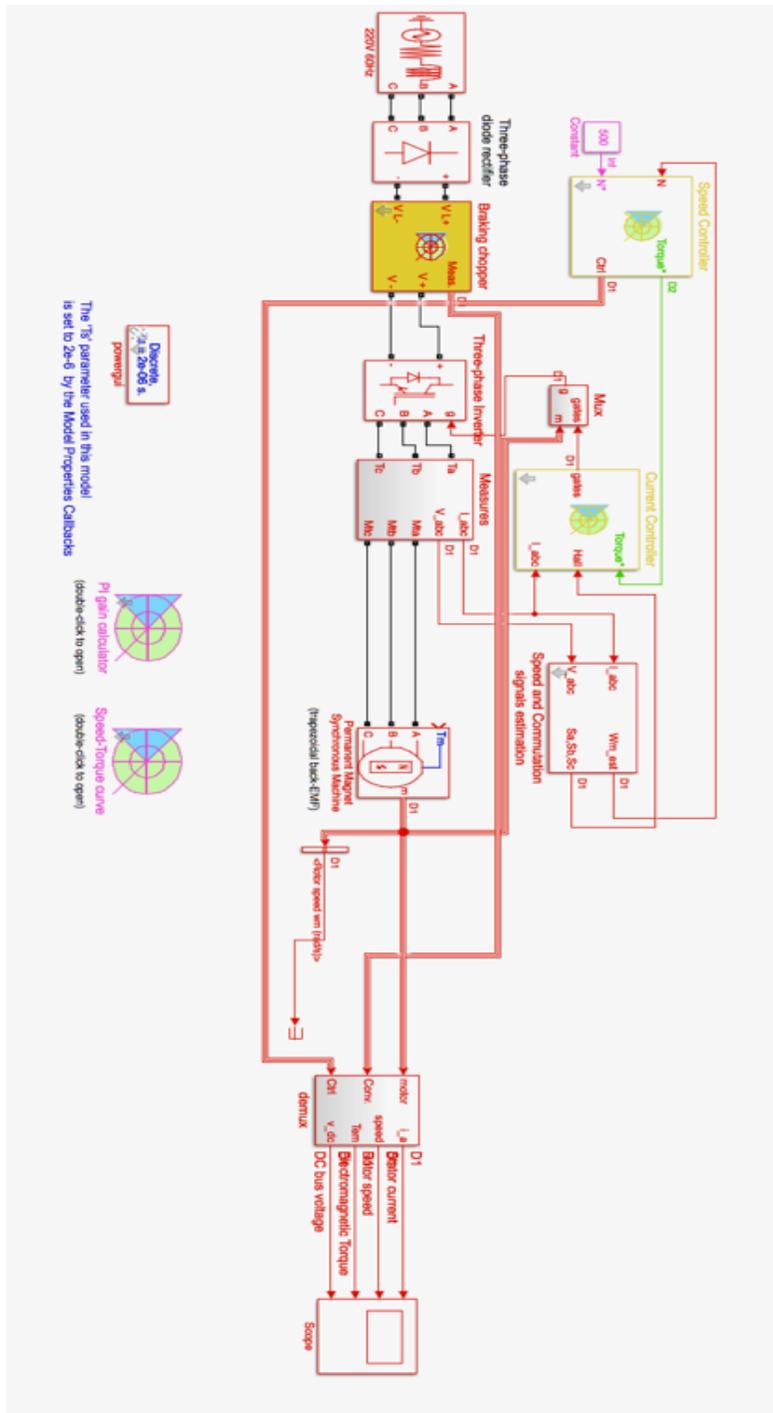


Figura 43: Diagrama de bloques del sistema de control difuso de un BLDC visto en Simulink (Matlab)

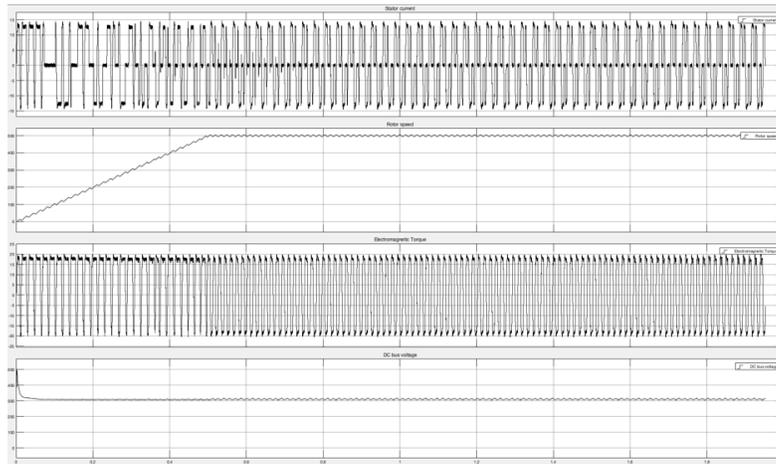


Figura 44: Cuatro oscilogramas que muestran i) la corriente de estator, ii) la velocidad del rotor, iii) el torque electromagnético y iv) el voltaje de CD de bus. Se puede observar que la velocidad llega a la referencia (500 rpm) aproximadamente en 500 milisegundos.

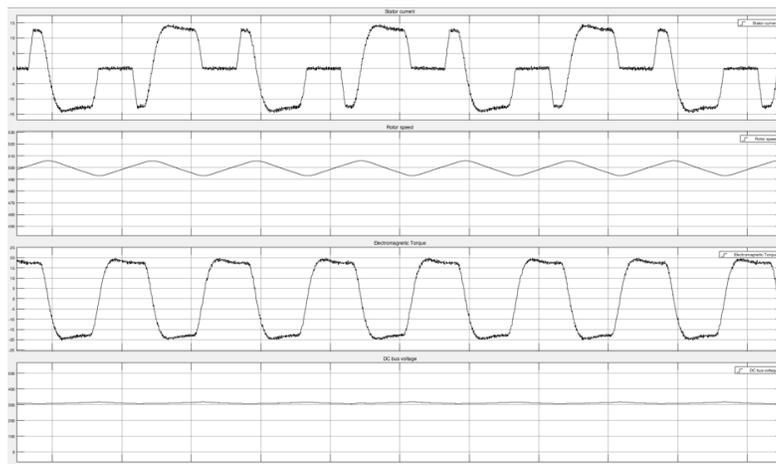


Figura 45: Acercamiento de la Figura 46. Se muestran las mismas variables pero se puede ver con mayor claridad la forma de las señales.

Una técnica para reducir el error máximo al tener una oscilación de menor magnitud alrededor del valor de referencia es concentrar las reglas de pertenencia del sistema de inferencia difuso alrededor del origen (Bose,2002). Los resultados de la simulación al haber modificado la forma de las reglas se puede observar en la [Figura 46](#).

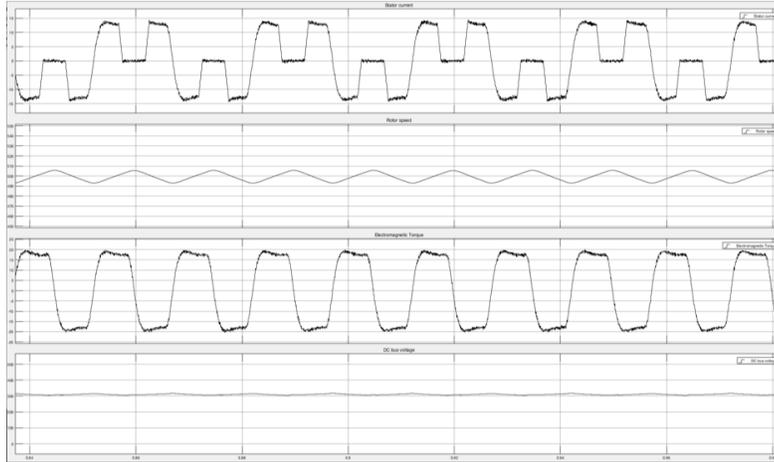


Figura 46: Acercamiento del oscilograma de cuatro variables al modificarse las reglas de pertenencia del primer sistema propuesto. El oscilograma es producto de la concentración de las reglas de pertenencia alrededor del origen.

:q!

Se observa que la oscilación alrededor del valor de referencia de la velocidad del rotor es muy similar al de la [Figura 45](#), que muestra los resultados de la primera versión del control difuso. No se observa una mejora en cuanto al error máximo.

## 2.5 IMPLEMENTACIÓN DEL CONTROL DIFUSO

Para implementar el control difuso modificamos las librerías del control y agregamos un bloque de código con el control difuso. El código se puede ver en el apéndice. En el código procesamos las entradas e implementamos los pasos correspondientes hasta pasar la salida al PWM.

Para el archivo difuso se usaron dos conjuntos de ecuaciones, mostradas a continuación:

- Ecuaciones de difusión para el sistema 1: [-1 , -0.9]

$$\alpha = 1 \quad (57)$$

$$[-0.9 , -0.5]$$

$$a = \frac{-5}{2}\text{Error} - \frac{5}{4} \quad (58)$$

**[-0.9 , -0.5]**

$$b = \frac{5}{2}\text{Error} + \frac{9}{4} \quad (59)$$

**[-0.5 , 0]**

$$b = -2\text{Error} \quad (60)$$

**[-0.5 , 0]**

$$c = 2\text{Error} + 1 \quad (61)$$

**[0 , 0.5]**

$$c = -2\text{Error} + 1 \quad (62)$$

**[0 , 0.5]**

$$d = 2\text{Error} \quad (63)$$

**[0.5 , 0.9]**

$$d = \frac{-5}{2}\text{Error} + \frac{9}{4} \quad (64)$$

**[0.5 , 0.9]**

$$e = \frac{5}{2}\text{Error} - \frac{5}{4} \quad (65)$$

**[0.9 , 1]**

$$e = 1 \quad (66)$$

**[-2 , -1.8]**

$$f = 1 \quad (67)$$

**[-1.8 , -1]**

$$f = -\frac{5}{4} \frac{d\text{Error}}{dt} - \frac{5}{4} \quad (68)$$

**[-1.8 , -1]**

$$g = \frac{5}{4} \frac{d\text{Error}}{dt} + \frac{9}{4} \quad (69)$$

[-1, 0]

$$g = -\frac{dError}{dt} \quad (70)$$

[-1, 0]

$$h = \frac{dError}{dt} + 1 \quad (71)$$

[0, 1]

$$h = -\frac{dError}{dt} + 1 \quad (72)$$

[0, 1]

$$i = \frac{dError}{dt} \quad (73)$$

[1, 1.8]

$$i = -\frac{5}{4} \frac{dError}{dt} + \frac{9}{4} \quad (74)$$

[1, 1.8]

$$j = \frac{5}{4} \frac{dError}{dt} - \frac{9}{4} \quad (75)$$

[1.8, 2]

$$j = 1 \quad (76)$$

- Ecuaciones para el sistema 2: [-1, -0.9]

$$a = 1 \quad (77)$$

[-0.9, -0.2]

$$a = -\frac{10}{7} \frac{dError}{dt} - \frac{2}{7} \quad (78)$$

[-0.7, -0.2]

$$b = 2Error + \frac{7}{5} \quad (79)$$

[-0.2, 0]

$$b = -5Error \quad (80)$$

[-0.2 , 0]

$$c = 5\text{Error} + 1 \quad (81)$$

[0 , 0.2]

$$c = -5\text{Error} + 1 \quad (82)$$

[0 , 0.2]

$$d = 5\text{Error} \quad (83)$$

[0.2 , 0.7]

$$d = -2\text{Error} + \frac{7}{5} \quad (84)$$

[0.2 , 0.9]

$$e = \frac{10}{7}\text{Error} - \frac{2}{7} \quad (85)$$

[0.9 , 1]

$$e = 1 \quad (86)$$

[-2 , -1.8]

$$f = 1 \quad (87)$$

[-1.8 , -0.4]

$$f = \frac{-5}{7} \frac{d\text{Error}}{dt} - \frac{2}{7} \quad (88)$$

[-1.4 , -0.4]

$$g = \frac{d\text{Error}}{dt} + \frac{7}{5} \quad (89)$$

[-0.4 , 0]

$$g = \frac{-5}{2} \frac{d\text{Error}}{dt} \quad (90)$$

[-0.4 , 0]

$$h = \frac{5}{2} \frac{d\text{Error}}{dt} + 1 \quad (91)$$

[0 , 0.4]

$$h = \frac{-5}{2} \frac{d\text{Error}}{dt} + 1 \quad (92)$$

**[0 , 0.4]**

$$i = \frac{5}{2} \frac{d\text{Error}}{dt} \quad (93)$$

**[0.4 , 1.4]**

$$i = -\frac{d\text{Error}}{dt} + \frac{7}{5} \quad (94)$$

**[0.4 , 1.8]**

$$j = \frac{5}{7} \frac{d\text{Error}}{dt} - \frac{2}{7} \quad (95)$$

**[1.8 , 2]**

$$j = 1 \quad (96)$$



## RESULTADOS Y CONCLUSIONES

A partir del proceso de diseño del control difuso y los resultados de las simulaciones, se llegó al sistema con el mejor desempeño dentro del marco de las pruebas y modificaciones que propusimos, un sistema de inferencia con cinco conjuntos para cada variable que tiene en la medición de la velocidad contra la velocidad de referencia un error de sobrepaso en su estado estable del 1 % o menos del valor de la señal de referencia.

Los resultados del control difuso en la simulación y en la implementación serán comparados a lo largo de este capítulo contra el comportamiento de un control PI. Esta referencia y los resultados correspondientes serán la base para la evaluación del diseño del control difuso, las ventajas y desventajas del uso de la lógica difusa en el contexto de la aplicación presentada en este trabajo y por último, las posibles mejoras a este trabajo.

### 3.1 RESULTADOS DE LA SIMULACIÓN - DIFUSO VS PI

Para fines de comparación, se realizó un ejercicio usando el mismo modelo y mismos bloques en Simulink para probar el desempeño de un control PI. El único bloque que se modificó para correr una simulación con control PI fue el controlador de velocidad. La siguiente figura muestra los resultados:

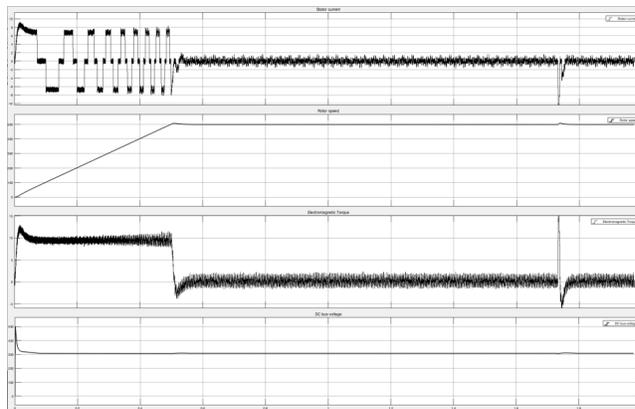


Figura 47: Oscilogramas de cuatro variables del motor; i) la corriente de estator, ii) la velocidad del rotor, iii) el torque electromagnético y iv) el voltaje de CD de bus (de arriba hacia abajo) al implementarse el sistema con un control PI.

El oscilograma de la velocidad del rotor muestra un sobrepaso que supera el 1 % de error con respecto a la velocidad de referencia. Sin embargo, al estabilizarse la velocidad, esta oscila alrededor de la referencia con un error máximo de 0.3 %. La velocidad máxima y mínima de las oscilaciones son 501.5[rpm] y 498.5 [rpm]. Es muy evidente la reducción del error con la implementación del control PI.

### 3.2 RESULTADOS DE LA IMPLEMENTACIÓN - DIFUSO VS PI

Con ayuda de CCS se realizó la medición de la velocidad real y la velocidad deseada del motor, tanto para el control PI, como para el control difuso. Se decidió ajustar la velocidad del motor a distintas velocidades, y extraer la información para procesarla y graficarla.

En la [Figura 48](#) se observa el comportamiento del control PI, teniendo una respuesta muy similar a la deseada.

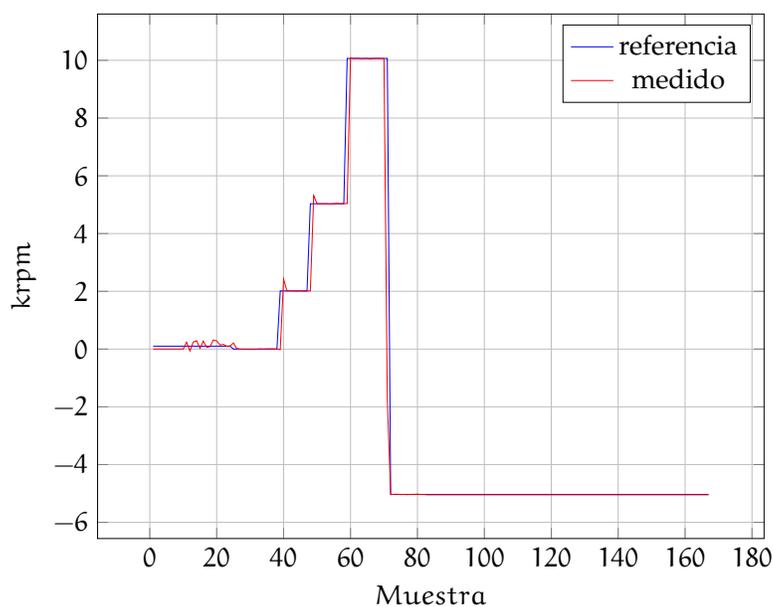


Figura 48: Resultados de la implementación. La figura muestra los valores de la referencia y de la velocidad medida a lo largo de 165 muestras. Durante este ejercicio se probaron 4 diferentes velocidades de referencia.

En la [Figura 49](#) se puede observar con detalle el comportamiento real del motor con el sistema de control difuso 1.

En este trabajo sólo se muestran los resultados con los dos conjuntos de reglas con los que se tuvo un mejor acercamiento al deseado. A partir del sistema de reglas 2 se decidió empezar a trabajar y a continuación se muestran distintos resultados variando constantes y errores.

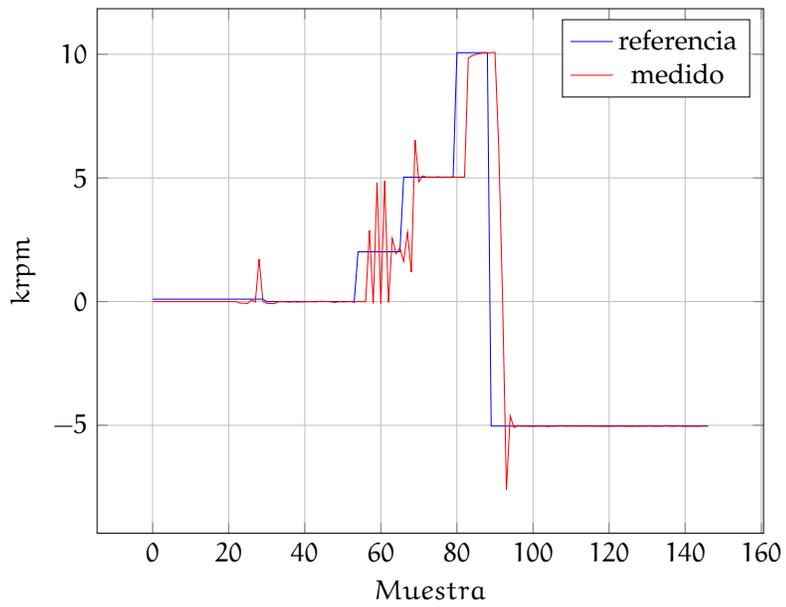


Figura 49: Sistema con control difuso 1 (anteriormente detallado) y constante de integración de 0.01

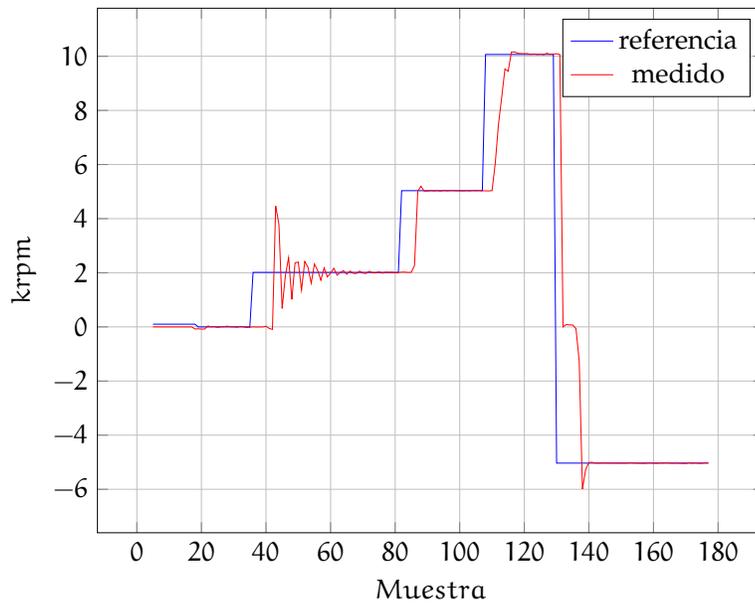


Figura 50: Sistema con control difuso 2 (anteriormente detallado) y constante de integración de 0.01

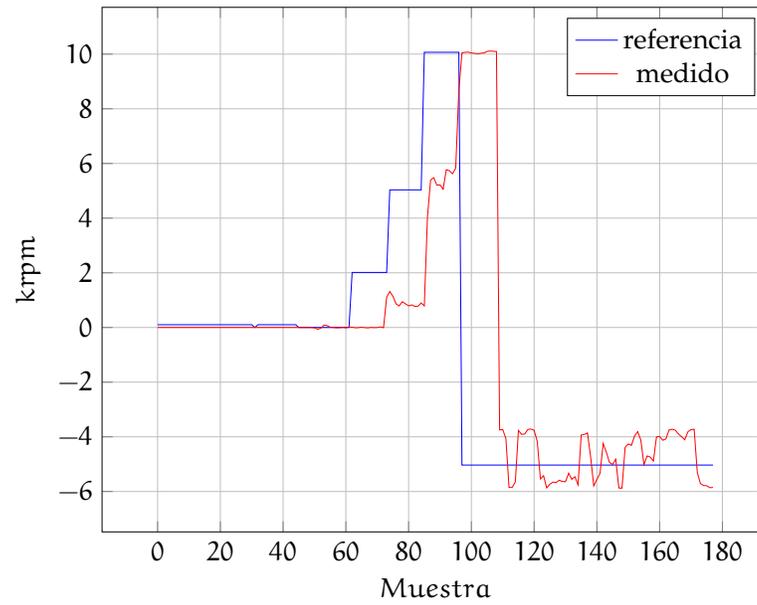


Figura 51: Sistema con control difuso 2 y constante de integración de 1.

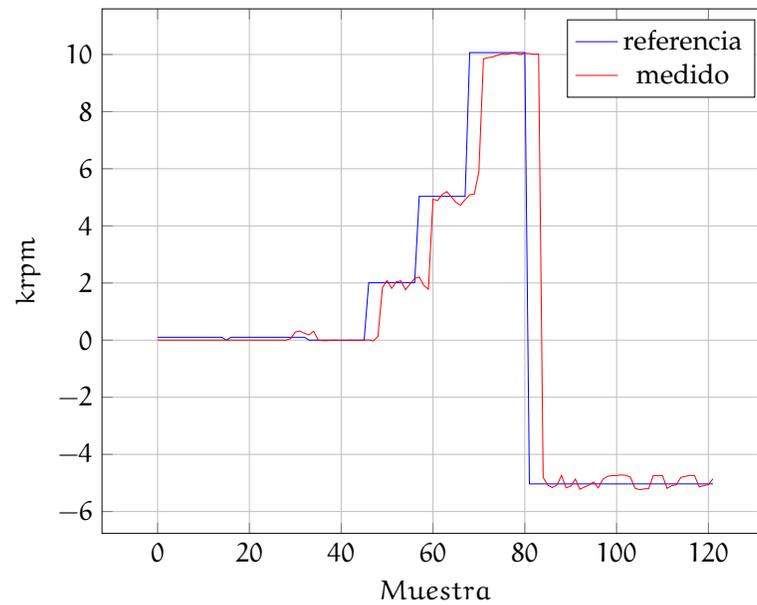


Figura 52: Sistema con control difuso 2 y constante de integración de 100.

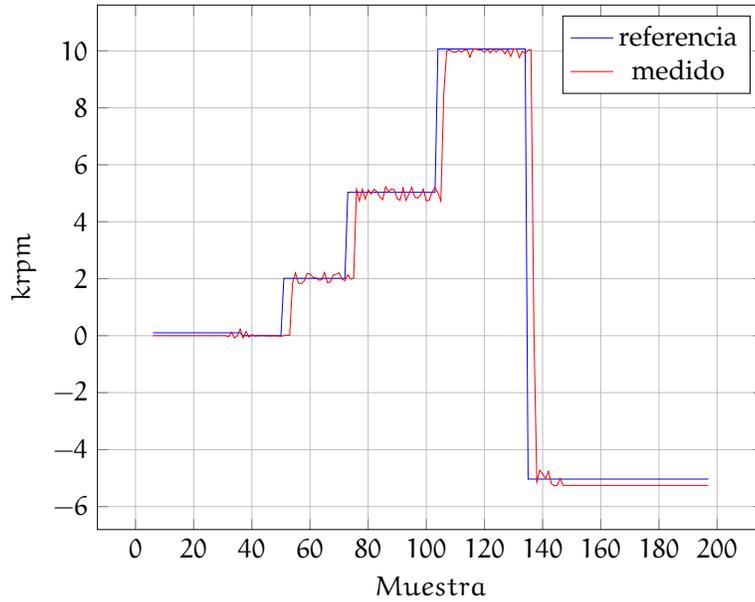


Figura 53: Sistema con control difuso 2 y constante de integración de 100, con el error multiplicado por dos. La técnica de multiplicar el error provoca un peso más fuerte, por lo que el sistema actúa de forma más rápida, ideal para diferencias muy grandes.

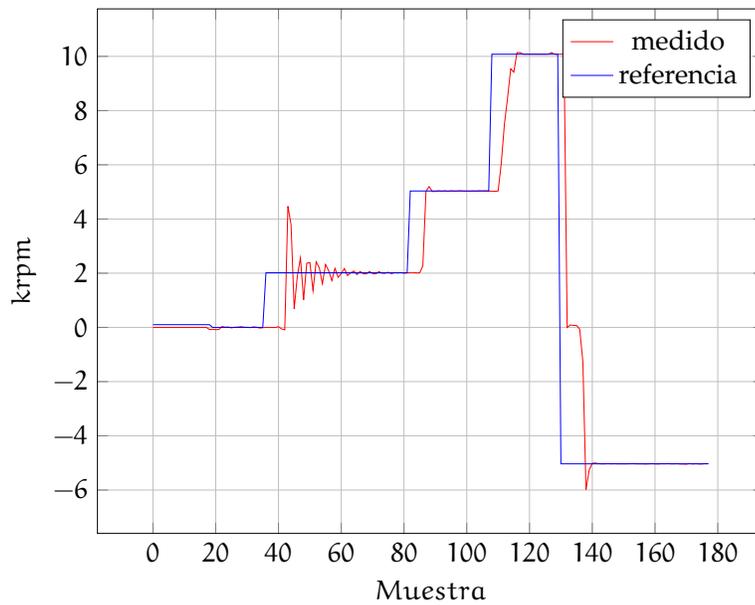


Figura 54: Sistema con control difuso 2 y constante de integración de 0.01, con el error multiplicado por dos.

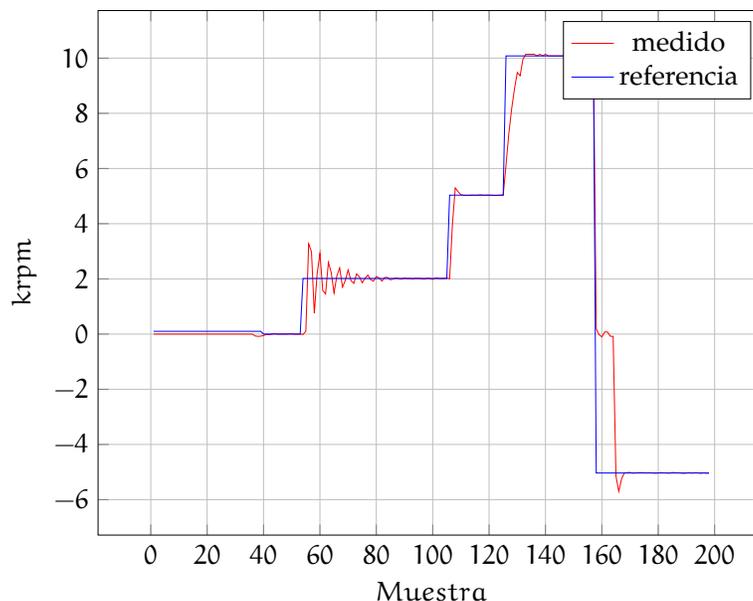


Figura 55: Sistema con control difuso 2 y constante de integración de 0.0008, con el error multiplicado por dos.

### 3.3 CONTROL DIFUSO ADAPTATIVO

Derivado de las implementaciones con los distintos sistemas de control difuso, es posible observar su comportamiento en las figuras previas ilustradas en este capítulo. De aquí se puede concluir que para velocidades bajas una constante de integración alta resulta muy eficiente; caso contrario para velocidades altas, en donde una constante pequeña muestra mejores resultados.

De esta forma, fue posible aplicar un sistema difuso adaptativo, en donde la constante de integración del sistema se define por la velocidad de referencia, manteniendo una constante de 100 para velocidades entre los intervalos de  $[-5, 5]$  krpm y una constante igual a 0,1 para velocidades entre los rangos de  $[-12, -5]$  o  $[5, 12]$  krpm.

El comportamiento se puede observar en la [Figura 56](#).

### 3.4 CONCLUSIONES

El uso de la lógica difusa para fines de control de un motor BLDC es una herramienta poderosa que permite la implementación rápida de un control con poco conocimiento del modelo matemático de la planta. El control difuso otorga la ventaja de poder diseñar un control mediante un modelo más amigable a comparación de un PI, usando sistemas de reglas que se asemejan más al proceso de razonamiento y entendimiento del cerebro humano.

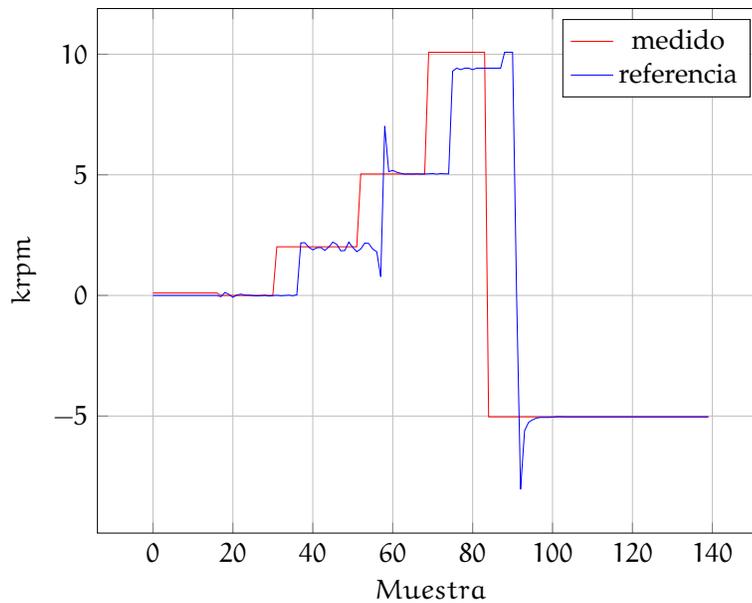


Figura 56: Control difuso adaptativo, aplicando diferentes valores a la constante de integración dependiendo de la velocidad de referencia.

Como se muestra en el trabajo, es posible controlar un motor genérico con un control difuso y obtener resultados positivos cuyo error máximo es 1%. Sin embargo, *MATLAB* e *InstaSpin-FOC* son herramientas especializadas que parametrizan el modelo del motor con alta precisión y el resultado de esto es que el control PI tiene mucho mejor desempeño ya que el modelo de la planta empleado se aproxima de buena manera a la realidad.

También cabe mencionar que el sistema difuso depende mucho de la jerarquización de las reglas, y entre más reglas se tenga más difícil es dicha jerarquización, porque se debe elegir qué comportamiento tiene más peso, para este caso se debía decidir, por ejemplo, si implicaba una mayor o menor variación un incremento en la aceleración que un incremento en la velocidad. De esta manera, se tienen que probar distintas jerarquizaciones y para este trabajo, para cada conjunto de reglas había 25 distintas reglas, por lo que la cantidad de jerarquizaciones posibles a elaborar es de  $25!$ . Resulta evidente que algunas de estas jerarquizaciones fueron omitidas en la elaboración de este proyecto.

El trabajo queda abierto a posibles mejoras tanto en el diseño de control (probando otro conjunto de reglas o jerarquización), como implementando otra solución al control difuso adaptativo, ya que aquí solo se intentó una pequeña mejora usando dos variables, pero puede incrementarse a más variables o a otro método de control adaptativo (Cox, 1993). Así mismo puede desarrollarse otro método de desdifusión, pero para este trabajo solo se ha intentado con promedios

ponderados por su practicidad y la eliminación de los pasos de implicación y agregación que conlleva implícitamente.

Este trabajo sirve para reafirmar la idea de que el control difuso encuentra sus mejores aplicaciones en casos donde en realidad se desconoce el modelo matemático de la planta, obteniendo resultados, que si bien para este trabajo no fueron igual de precisos como un PI, pueden llegar a tener mejor desempeño que un sistema de control convencional como se demuestra en el trabajo de Bouguerra (A. Bouguerra, 2017).

A continuación se hará una breve descripción de los componentes de software y hardware empleados en este trabajo. Se busca dar un esbozo general de su funcionamiento y se mencionan las especificaciones más relevantes de los componentes, en el contexto de este trabajo. Los componentes de hardware empleados son: la tarjeta de desarrollo *LaunchXL-F28027* y el driver para el motor BLDC *BOOSTXL-DRV8301*. *InstaSpin-FOC* es el software que se utiliza para controlar el motor. Todos los componentes empleados en este trabajo son propiedad de *Texas-Instruments*.

Adicionalmente, en este apartado se puede encontrar el archivo que contiene el código fuente del bloque de código que referenciamos a lo largo del trabajo. El archivo *fuzzy.h* es la implementación del sistema de inferencia difuso como bloque de control dentro de un controlador de lazo cerrado.

#### ***LaunchXL-F28027***

La tarjeta de desarrollo *LaunchXL-F28027* es parte de la familia de tarjetas C2000Piccolo de 32-bits, que a su vez cuenta con el procesador *TMS320F28027*. Esta familia de microcontroladores cuenta con altos niveles de integración analógica con componentes periféricos y capacidad de procesamiento. El diseño de este microcontrolador está orientado a aplicaciones de tiempo real y de bajo costo que requieren sensores de alta precisión y alta capacidad de cómputo.

Algunas de las características del hardware de esta tarjeta son las siguientes:

- Reloj de operación a 60[MHz] C28x CPU
- Memoria FLASH de 64KB
- Convertidor analógico-digital de 12 bits
- Conexión USB aislada *XDS100 JTAG*

Otra de las características importantes de la tarjeta, que se encuentra detrás de su elección para este trabajo es la integración con la que cuenta a drivers para motores, la compatibilidad de uso con el software para control de motores *InstaSpin-FOC* y el uso con la plataforma de desarrollo de programas *Code Composer Studio*, igualmente propiedad de *Texas Instruments*.

#### ***BOOSTXL-DRV8301***

Este componente es un driver de 10[A], de tres fases para motores de CD sin escobillas. El modulo cuenta con 3 sensores de corriente y con un convertidor de potencia Buck, además de protecciones contra circuitos cortos. Algunas de sus especificaciones se muestran a continuación:

- Entrada de alimentación de 6 a 24 [V]
- Soporta picos de hasta 14 [A], y corriente continua de salida de 10 [A]
- Soporte de corriente y voltaje de realimentación para el uso de la solución de control sin sensores *InstaSpin-FOC*
- Soporte para drivers de CD de motores sin escobillas

Este driver en particular soporta en su totalidad el software *InstaSpin-FOC*, solución de control sin sensores.

### *InstaSpin-FOC*

Este software en su totalidad representa una solución completa para el control de campo orientado sin sensores. Esta solución tienen la capacidad de controlar un motor de manera eficiente en cuestión de minutos sin la necesidad de emplear sensores de rotor mecánicos. En conjunto con los demás componentes del microcontralador, del estimador FAST (*firmware*) y de funciones adicionales para el control de un motor que implementa el software para lograr un eficiente control de campo orientado de tres fases. A continuación de enlistan algunas de las características más relevantes de la solución *InstaSpin-FOC*:

- Incluye el estimador FAST para medir el flujo del rotor (magnitud y ángulo) en un sistema de control de campo orientado
- Ajuste automático de torque de lazo (corriente indirectamente)
- Configuración automática de ganancias de lazo (proporcional e integral) que proveen la operación estable
- Permite el ajuste de parámetros para la respuesta transitoria por parte del usuario

El software *InstaSpin-FOC* es la herramienta principal empleada en este trabajo. Al poder implementar un control de campo orientado sin el uso de sensores, responde a la necesidad más importante en el contexto de como controlar un motor BLDC dejando a un lado el uso de cualquier tipo de sensor discreto.

Listing 1: Fuzzy.h 1 de 3

---

```

3  /*
   * fuzzy.h
   *
   * Created on: 10/04/2016
   * Author: Alonso
   */
8
   #ifndef FUZZY_H_
   #define FUZZY_H_

   #include ".././././modules/iqmath/src/float/IQmathLib.h"
13  #include ".././././modules/iqmath/src/32b/IQmathLib.h"
   #include "sw/modules/types/src/types.h"
   #include <math.h>

   #ifdef __cplusplus
18  extern "C" {
   #endif

   /*! \brief Runs the PID controller for speed
   /*! \param[in] handle The PID controller handle
23  /*! \param[in] refValue The reference value to the controller
   /*! \param[in] fbackValue The feedback value to the controller
   /*! \param[in] pOutValue The pointer to the controller output value
   float erroractd = 0;
   float errorantd = 0;
28  float Derrord = 0;
   float a=0;
   float c=0;
   float e=0;
   float f=0;
33  float g=0;
   float h=0;
   float b=0;
   float d=0;
   float i=0;
38  float j=0;
   float R,R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15,R16,R17,R18,R19,R20,R21,R22,R23,
       R24,R25;
   float centroide;
   double centro = 0;
   float Antecedente(float x, float y);
43  float MNG(float erroractd);
   float NG(float erroractd);
   float Cero(float erroractd);
   float PG(float erroractd);
   float MFG(float erroractd);
48  float MN(float Derrord);
   float N(float Derrord);
   float DCero(float Derrord);
   float P(float Derrord);
   float MP(float Derrord);
53  double Desdifusion(float R1,float R2,float R3,float R4,float R5,float R6,float R7,float R8,
       float R9,float R10,float R11,float R12,float R13,float R14,float R15,float R16,float
       R17,float R18,float R19,float R20,float R21,float R22,float R23,float R24,float R25);

   static inline void fuzzy(PID_Handle handle, _iq erroract, _iq Derrord, _iq errorant, _iq *
       pOutValue) {
       //erroract = referencia-medido;
       //Derrord = erroract - errorant;
58  //errorant = erroract;
       erroractd = (float)erroract / 16198686;
       Derrord = (float)Derrord / 16198686;
       a = MNG(erroractd);
       b = NG(erroractd);
63  c = Cero(erroractd);
       d = PG(erroractd);
       e = MFG(erroractd);
       f = MN(Derrord);
       g = N(Derrord);
68  h = DCero(Derrord);
       i = P(Derrord);
       j = MP(Derrord);
       R1 = Antecedente(a, f);
       R2 = Antecedente(b, f);
73  R3 = Antecedente(a, g);
       R4 = Antecedente(b, g);
       R5 = Antecedente(c, f);
       R6 = Antecedente(c, g);
       R7 = Antecedente(a, h);
78  R8 = Antecedente(b, h);
       R9 = Antecedente(a, j);
       R10 = Antecedente(b, j);
       R11 = Antecedente(a, i);
       R12 = Antecedente(b, i);
83  R13 = Antecedente(c, h);
       R14 = Antecedente(d, j);

```

---

Listing 2: Fuzzy.h 2 de 3

---

```

R15 = Antecedente(e,g);
R16 = Antecedente(d,f);
R17 = Antecedente(e,f);
5  R18 = Antecedente(d,h);
R19 = Antecedente(e,h);
R20 = Antecedente(c,i);
R21 = Antecedente(c,j);
R22 = Antecedente(d,i);
10 R23 = Antecedente(e,i);
R24 = Antecedente(d,j);
R25 = Antecedente(e,j);
centroide = 1677721 * Desdifusion (R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15,R16,
R17,R18,R19,R20,R21,R22,R23,R24,R25);
centroide = centroide * 0.001;
15 /*pOutValue = _IQsat(*pOutValue + centroide,1677721,-1677721);
*pOutValue = _IQsat(*pOutValue + centroide,180000,-180000);
return;
}

20 float MNG(float erroractd){
if (erroractd >= -1 && erroractd <= -0.9){
a = 1;
} else{
if (erroractd > -0.9 && erroractd <= -0.2){
25 a = ( - ( 10 / 7 ) * erroractd ) - ( 2 / 7 ) ;
} else{
a = 0;
}
}
30 return a;
}

float NG(float erroractd){
if (erroractd >= -0.7 && erroractd <= -0.2){
35 b = ( 2 * erroractd ) + 1.4;
} else{
if (erroractd > -0.2 && erroractd <= 0){
b = ( - 5 * erroractd );
} else{
40 b = 0;
}
}
return b;
}

45 float Cero(float erroractd){
if (erroractd >= -0.2 && erroractd <= 0){
c = ( 5 * erroractd ) + 1;
} else{
50 if (erroractd > 0 && erroractd <= 0.2){
c = ( - 5 * erroractd ) + 1;
} else{
c = 0;
}
}
55 return c;
}

float PG(float erroractd){
60 if (erroractd >= 0 && erroractd <= 0.2){
d = ( 5 * erroractd );
} else{
if (erroractd > 0.2 && erroractd <= 0.7){
65 d = ( - 2 * erroractd ) + 1.4;
} else{
d = 0;
}
}
return d;
70 }

float MCG(float erroractd){
if (erroractd >= 0.2 && erroractd <= 0.9){
75 e = ( ( 10 / 7 ) * erroractd ) - ( 2 / 7 ) ;
} else{
if (erroractd > 0.9 && erroractd <= 1){
e = 1;
} else{
80 e = 0;
}
}
return e;
}

```

---

## Listing 3: Fuzzy.h 3 de 3

```

float MN(float Derrord){
  if(Derrord >= -2 && Derrord <= -1.8){
    f = 1;
5  }else{
    if(Derrord > -1.8 && Derrord <= -0.4){
      f = ( - ( 5 / 7 ) * Derrord ) - ( 2 / 7 );
    }else{
      f = 0;
10  }
    }
    return f;
  }
}

15 float N(float Derrord){
  if(Derrord >= -1.4 && Derrord <= -0.4){
    g = ( Derrord ) + 1.4;
  }else{
    if(Derrord > -0.4 && Derrord <= 0){
20  g = - 2.5 * Derrord;
    }else{
      g = 0;
    }
  }
25  return g;
}

float DCero(float Derrord){
30  if(Derrord >= -0.4 && Derrord <= 0){
    h = 2.5 * Derrord + 1;
  }else{
    if(Derrord > 0 && Derrord <= 0.4){
      h = - 2.5 * Derrord + 1;
35  }else{
      h = 0;
    }
  }
  return h;
}

40 float P(float Derrord){
  if(Derrord >= 0 && Derrord <= 0.4){
    i = 2.5 * Derrord;
  }else{
45  if(Derrord > 0.4 && Derrord <= 1.4){
      i = ( - 1 * Derrord ) + 1.4;
    }else{
      i = 0;
    }
50  }
  return i;
}

float MP(float Derrord){
55  if(Derrord >= 0.4 && Derrord <= 1.8){
    j = ( ( 5 / 7 ) * Derrord ) - ( 2 / 7 );
  }else{
    if(Derrord > 1.8 && Derrord <= 2){
60  j = 1;
    }else{
      j = 0;
    }
  }
  return j;
65  }
}

float Antecedente(float x, float y){
  if(x >= y){
70  R = y;
  }else{
    R = x;
  }
  return R;
}

75 double Desdifusion(float R1, float R2, float R3, float R4, float R5, float R6, float R7, float R8,
float R9, float R10, float R11, float R12, float R13, float R14, float R15, float R16, float
R17, float R18, float R19, float R20, float R21, float R22, float R23, float R24, float R25){
  centro = ((R1*-1)+(R2*-0.91)+(R3*-0.83)+(R4*-0.75)+(R5*-0.66)+(R6*-0.58)+(R7*-0.5)+(R8
*-0.42)+(R9*-0.33)+(R10*-0.25)+(R11*-0.16)+(R12*-0.08)+(R13*0)+(R14*0.08)+(R15*0.16)
+(R16*0.25)+(R17*0.33)+(R18*0.42)+(R19*0.5)+(R20*0.58)+(R21*0.66)+(R22*0.75)+(R23
*0.83)+(R24*0.91)+(R25*1))/( R1 + R2 + R3 + R4 + R5 + R6 + R7 + R8 + R9 + R10 + R11 +
R12 + R13 + R14 + R15 + R16 + R17 + R18 + R19 + R20 + R21 + R22 + R23 + R24 + R25 );
  return centro;
80  }
}

#endif /* FUZZY_HL */

```

## BIBLIOGRAFÍA

---

- [1] S. Zeghlache A. Bouguerra, K. Loukal. Speed Control of a Brushless dc Motor (bldcm) Based on Fuzzy Gain-Adaptive pi. In *10th International Conference on Electrical and Electronics Engineering*, pages 216–221. IEEE, 2017.
- [2] C. Kingsley Jr. A.E. Fitzgerald. *Electrical Machinery*. McGraw-Hill, New York, 6a edition, 2003.
- [3] B. Akin and M. Bhardwaj. *Sensorless Field Oriented Control of 3-Phase Permanent Magnet Synchronous Motors*. Texas Instruments, Dallas TX, 2013.
- [4] George Boole. *An Investigation of the Laws of Thought: On Which Are Founded the Mathematical Theories of Logic and Probabilities*. Cambridge University Press, Cambridge UK, 2009.
- [5] Bimal K. Bose. *Modern Power Electronics and AC Drives*. Prentice Hall PTR, New Jersey, 1a edition, 2002.
- [6] R. Condit. *Sensorless Bldc Control with Back-EMF Filtering*. Microchip Technology Inc., 2007.
- [7] E. Cox. Adaptive fuzzy systems. *IEEE Spectrum*, 30(2):27–31, 1993.
- [8] S. Durfee. High-efficiency motor technology, 2015.
- [9] A. Chávez-Placencia E. Gómez-Ramírez. How to Tune Fuzzy Controllers. In *IEEE International Conference on Fuzzy Systems*, pages 1287–1292. IEEE, 2004.
- [10] Business Insider Intelligence. Drones 101: The future of drones for consumers, businesses, and the military, April 2018.
- [11] F. S. Hover M. S. Triantafyllou. Maneuvering and Control of Marine Vehicles. Technical report, Massachusetts Institute of Technology, Cambridge, MA, 2005.
- [12] F. Parasiliti M. Tursini, R. Petrella. Initial Rotor Position Estimation Method for Pm Motors. *IEEE Trans. Ind. Appl.*, 39(6), November/December 2003.
- [13] E.H. Mamdani. Application of fuzzy algorithms for control of simple dynamic plant. *PROC. IEEE*, 121(12), December 1974.
- [14] K. Vasudeven P. Damodharan, R. Sandeep. Simple position sensorless starting method for brushless Dc motor. *IET Electr. Power Appl*, 2(1):49–55, 2008.
- [15] V. Perez-Amador. *Generadores Motores y Transformadores Eléctricos*. FI UNAM, Ciudad de México, 1a edition, 1992.

- [16] W. Rippel. *Iduction Versus dc Brushless Motors*, 2007.
- [17] L.E. Unnewehr S.A. Nasar. *Electromecánica y Máquinas Eléctricas*. Editorial Limusa, Ciudad de México, 1a edition, 1990.
- [18] R. Strietzel. Tuning of Fuzzy Controllers - An Overview. In *IFAC Proceedings Volumes*, pages 313–318. ELSEVIER, 2000.
- [19] Texas Instruments, Dallas TX. *InstaSPIN Projects and Labs User's Guide*, 1a edition.
- [20] Texas Instruments, Dallas TX. *InstaSPIN-FOC and InstaSPIN-MOTION*, 2013.
- [21] P. Yedamale. *Brushless Dc (Bldc) Motor Fundamentals*. Microchip Technology Inc., 2003.
- [22] S.Y. Yun. Research on the Starting Methods for Initial Driving of Pmsm. In *15th International Conference on Electrical Machines and Systems*. IEEE, 2012.
- [23] L. A. Zadeh. Fuzzy Sets. *Information and Control*, 8(47):338–353, 1965.

