



UNIVERSIDAD DEL BÍO-BÍO

FACULTAD DE INGENIERÍA
DEPTO. INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

**“Implementación de un Controlador FOC para Motores
Brushless con Encoder Utilizando STM32”**

AUTOR:
RODRIGO FUENTES PEDREROS

SEMINARIO PARA OPTAR AL TÍTULO DE
INGENIERO DE EJECUCIÓN EN ELECTRÓNICA

CONCEPCIÓN - CHILE
AÑO 2024



UNIVERSIDAD DEL BÍO-BÍO

FACULTAD DE INGENIERÍA
DEPTO. INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

**“Implementación de un Controlador FOC para Motores
Brushless con Encoder Utilizando STM32”**

AUTOR
RODRIGO FUENTES PEDREROS

PROFESOR GUÍA:
ANGEL ERNESTO RUBIO

PROFESORES GUÍA ADJUNTO:
PEDRO MELIN COLINA

Índice

Resumen	5
1 Introducción	6
1.1 Introducción general	6
1.2 Marco teórico	7
1.2.1 Motores <i>brushless</i>	7
1.2.2 Control Orientado de Campo (FOC)	8
1.2.3 Transformada de Clarke	8
1.2.4 Transformada de Park	9
1.2.5 Transformada inversa de Park	9
1.2.6 Modulación de Espacio Vectorial (SVM)	10
1.3 Motivación	11
1.4 Objetivos	11
1.4.1 Objetivo General	11
1.4.2 Objetivos Específicos	11
1.4.3 Alcances	12
2 Desarrollo del controlador	13
2.1 Introducción	13
2.2 Desarrollo del hardware	13
2.2.1 Motor	14
2.2.2 Encoder	15
2.2.3 Conjunto motor encoder	16
2.2.4 Puente Mosfet	17
2.2.5 Controladores de Puerta	19
2.2.6 Sensores de Corriente	21
2.3 Implementación del hardware	22
2.4 Microcontrolador y periféricos	23
2.4.1 Configuración de Timers	23
2.4.2 Configuración de ADCs	26
2.5 Firmware	26
3 Validación	27

3.1	Validación de la adquisición y transformación de las mediciones de corriente	28
3.1.1	Validación de las mediciones de corriente	28
3.1.2	Validación de la transformada de Clarke	29
3.1.3	Validación de la transformada de Park	30
3.2	Validación de los Controladores PI	31
3.2.1	Validación del controlador de velocidad	31
3.2.2	Validación del controlador de corriente	32
3.3	Validación señales del SVM	33
Comentarios y Conclusiones		34
Bibliografía		35
Anexos		36
Anexo A		36

Resumen

Capítulo 1

Introducción

1.1. Introducción general

Cuando se habla de motores eléctricos de corriente continua, los motores sin escobillas destacan por su alto desempeño, siendo la opción preferida en aplicaciones como vehículos eléctricos, drones, robótica avanzada y robótica industrial.

Para los motores *brushless*, existe una gran variedad de técnicas de control. Algunas de las más relevantes son el control trapezoidal, utilizado mayormente en drones; el control directo de torque (DTC), empleado principalmente en motores de media y alta potencia; y el control FOC, que se aplica mayormente en robótica. En este documento, se tratará únicamente esta última técnica.

Existen tres aspectos clave que diferencian cada técnica: el rizado de torque, el costo computacional y la complejidad del hardware necesario para su ejecución. En estos aspectos, el control FOC es uno de los que produce menor rizado de torque, aunque requiere un mayor coste computacional y hardware especializado.

Este proyecto presenta la implementación y validación de un controlador de campo orientado (FOC) para motores sin escobillas de corriente continua (BLDC). Se desarrolló la placa controladora utilizando STM32 y se implementó el algoritmo de control FOC en lenguaje C.

1.2. Marco teórico

1.2.1. Motores *brushless*

Los motores *brushless*, o motores sin escobillas de corriente continua (BLDC), son un tipo de motores trifásicos síncronos que utilizan imanes permanentes en el rotor y bobinas en el estator, el cual está formado por un núcleo de hierro laminado. A diferencia de los motores de corriente continua con escobillas, los BLDC no emplean elementos mecánicos para conmutar la corriente en las bobinas. En su lugar, esta conmutación se realiza mediante un controlador electrónico [1], eliminando el rozamiento y el desgaste asociados a las escobillas, lo que mejora la eficiencia del sistema.

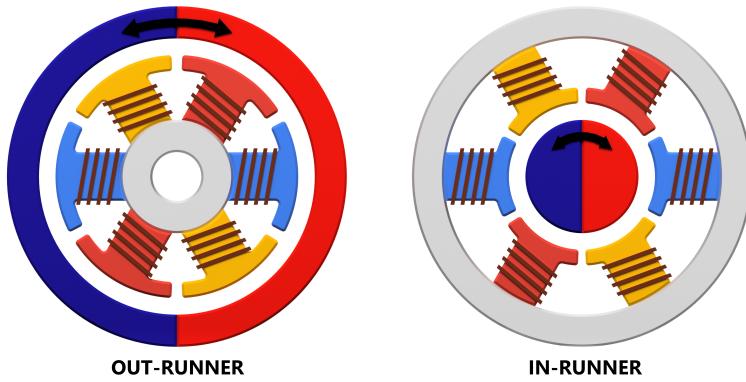


Figura 1.1: Esquema de motores sin escobillas (*BLDC*).

Similar a los motores trifásicos de corriente alterna, estos pueden tener configuraciones de bobinado en delta o estrella [2]. Además, pueden presentar configuraciones *in-runner*, donde los imanes están en el centro del eje con las bobinas en el exterior, o configuraciones *out-runner*, donde los imanes se encuentran por fuera del motor, mientras que las bobinas están en el centro [3]. Estos motores, al requerir una sincronización entre el campo magnético del estator y el campo magnético del rotor, suelen utilizar encoders o sensores de efecto Hall para obtener información sobre la posición del eje y, de este modo, mantener una conmutación adecuada. No obstante, también es posible emplear técnicas *sensorless* (sin sensor) para estimar la posición del rotor sin necesidad de sensores físicos. [4]

Los *BLDC* se destacan por ofrecer una mayor densidad de potencia, mayor torque, mejor eficiencias y pueden llegar a velocidades mas altas. Sin embargo, al requerir de un controlador electrónico que se encargue de la conmutación para generar el movimiento del rotor, la complejidad y los costos asociados a su implementación son mayores. [5]

1.2.2. Control Orientado de Campo (FOC)

El control orientado de campo se basa en desacoplar el flujo electromagnético del par motor, permitiendo el control independiente de cada uno. Este desacoplamiento se logra mediante la transformación de las variables trifásicas del motor, que están en un marco de referencia estacionario ABC , hacia un marco de referencia rotatorio dq que gira sincrónamente con el rotor. [6]

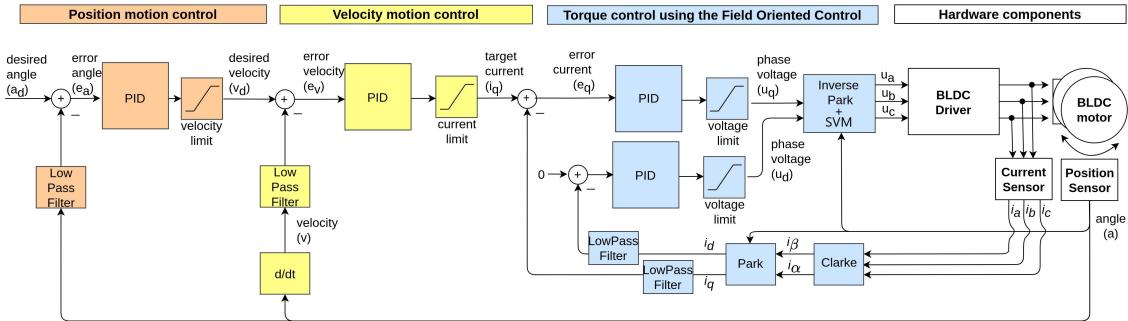


Figura 1.2: Diagrama de flujo del control FOC.[7]

La transformación entre los marcos de referencia se realiza aplicando la transformada de Clarke y la transformada de Park. De esta forma, las variables que presentan un comportamiento oscilatorio en el tiempo se convierten en variables de corriente continua, lo que permite emular, en cierta medida, el funcionamiento de un motor con escobillas a nivel de controladores. [6]

1.2.3. Transformada de Clarke

La transformada de Clarke convierte un sistema trifásico de corrientes ABC en un sistema bifásico $\alpha\beta$, proyectando las corrientes en un sistema de coordenadas bidimensional estacionario. Esta transformación simplifica el análisis al reducir las tres corrientes de fase a dos componentes ortogonales, manteniendo la información esencial del sistema original. [8]

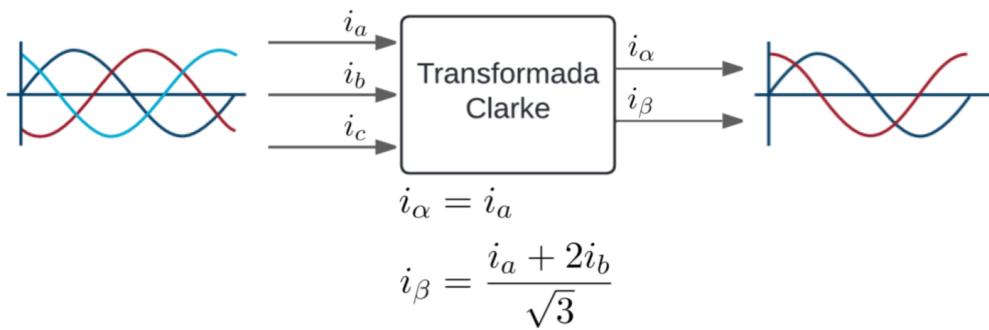


Figura 1.3: Esquema transformada de Clarke.

1.2.4. Transformada de Park

La transformada de Park convierte el sistema en el marco de referencia estacionario $\alpha\beta$ en el sistema en el marco de referencia rotatorio dq sincronizado con el rotor del motor. [8]

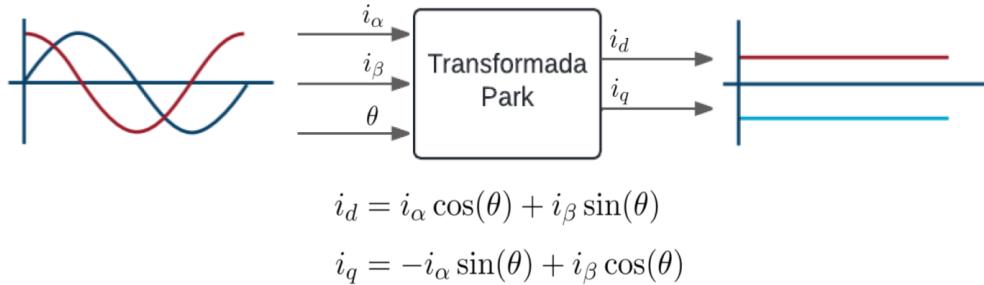


Figura 1.4: Esquema transformada de Park.

1.2.5. Transformada inversa de Park

La transformada inversa de Park convierte el sistema en el marco de referencia rotatorio dq de vuelta al marco de referencia estacionario $\alpha\beta$. [8].

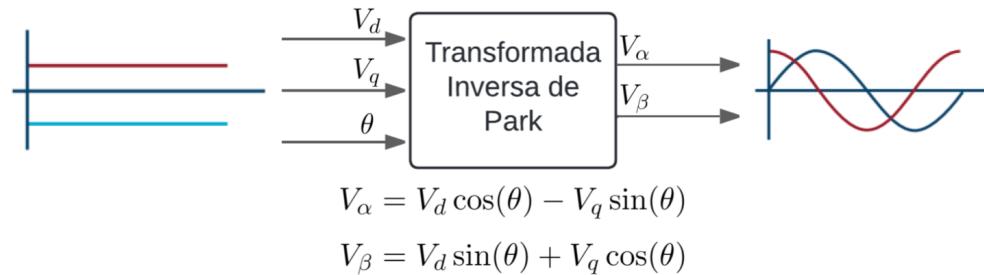


Figura 1.5: Esquema transformada inversa de Park.

1.2.6. Modulación de Espacio Vectorial (SVM)

La Modulación por Vector Espacial (SVM) es una técnica utilizada para el control digital de inversores de voltaje. Representa los estados de conmutación del inversor como vectores de voltaje en el plano α - β , formando un hexágono regular dividido en seis sectores o sextantes.



Figura 1.6: Esquema SVM.

Esta técnica busca sintetizar el vector de referencia $\vec{V}_{\text{ref}} = (V_\alpha, V_\beta)$ mediante una suma ponderada de los vectores de voltaje adyacentes en el sextante donde se encuentre, logrando una aproximación más precisa de la señal deseada.

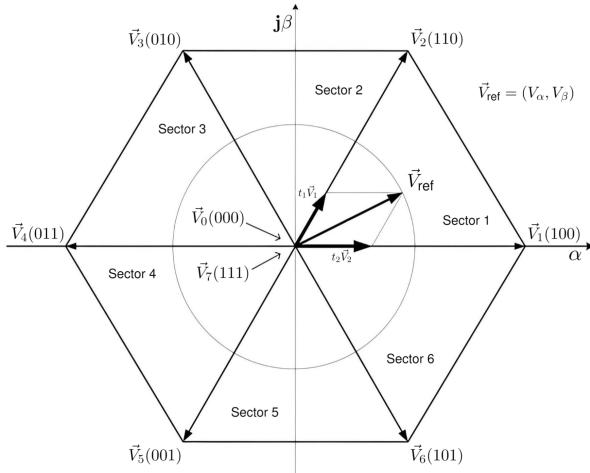


Figura 1.7: Sextantes en SVM.

Se utiliza un patrón de activación específico que permite minimizar la frecuencia de conmutación de los transistores, aprovechando estados redundantes para reducir las pérdidas asociadas a la conmutación y mejorar la eficiencia del sistema. [9]

Tabla 1.1: Secuencia de Conmutación en SVM

Sector	1	2	3	4	5	6	7
I	$\vec{V}_0(000)$	$\vec{V}_1(100)$	$\vec{V}_2(110)$	$\vec{V}_7(111)$	$\vec{V}_2(110)$	$\vec{V}_1(100)$	$\vec{V}_0(000)$
II	$\vec{V}_0(000)$	$\vec{V}_3(010)$	$\vec{V}_2(110)$	$\vec{V}_7(111)$	$\vec{V}_2(110)$	$\vec{V}_3(010)$	$\vec{V}_0(000)$
III	$\vec{V}_0(000)$	$\vec{V}_3(010)$	$\vec{V}_4(011)$	$\vec{V}_7(111)$	$\vec{V}_4(011)$	$\vec{V}_3(010)$	$\vec{V}_0(000)$
IV	$\vec{V}_0(000)$	$\vec{V}_5(001)$	$\vec{V}_4(011)$	$\vec{V}_7(111)$	$\vec{V}_4(011)$	$\vec{V}_5(001)$	$\vec{V}_0(000)$
V	$\vec{V}_0(000)$	$\vec{V}_5(001)$	$\vec{V}_6(101)$	$\vec{V}_7(111)$	$\vec{V}_6(101)$	$\vec{V}_5(001)$	$\vec{V}_0(000)$
VI	$\vec{V}_0(000)$	$\vec{V}_1(100)$	$\vec{V}_6(101)$	$\vec{V}_7(111)$	$\vec{V}_6(101)$	$\vec{V}_1(100)$	$\vec{V}_0(000)$

1.3. Motivación

Este proyecto nace de la necesidad de un controlador para motores brushless adecuado para su uso en robots de competencia en la categoría de robot sumo autónomo. En esta categoría, este tipo de motores no son normalmente utilizados debido a las limitaciones de los controladores comerciales y los riesgos que conlleva la sobrecarga de los motores en periodos cortos de acción, ya que los controladores comerciales no están pensados para esto.

1.4. Objetivos

1.4.1. Objetivo General

Implementar un controlador de tipo FOC (Control de Campo Orientado) para motores brushless con encoder, utilizando un microcontrolador STM32, que sirva de base para un driver especializado en la robótica competitiva.

1.4.2. Objetivos Específicos

- Estudiar los principios del Control de Campo Orientado (FOC) y la modulación de espacio vectorial (SVM) para aplicarlos en el diseño del controlador.
- Diseñar el hardware para el controlador FOC, con los componentes mínimos necesarios para validar el funcionamiento.
- Configurar y programar el microcontrolador para el algoritmo FOC, utilizando las librerías HAL de STM32
- Validar el funcionamiento del controlador y proponer posibles mejoras para su aplicación en robótica competitiva.

1.4.3. Alcances

- Se configurará el microcontrolador utilizando STM32CubeMX.
- Se implementará el controlador de velocidad y corriente.
- No se implementará el control de posición.
- Se programará el firmware utilizando VScode y el lenguaje C.
- El firmware se compilará utilizando un makefile y GCC, junto con la extensión STM32 en VScode.
- Se cargará el firmware utilizando STM32CubeProgrammer y ST-link.
- Se diseñará la PCB en Eagle.
- Se diseñarán las piezas adicionales en Autodesk Inventor, para su posterior fabricación mediante impresión 3D.
- La PCB se fabricará y semi ensamblará utilizando JLCPCB.
- Se obtendrán datos detallados durante el tiempo de ejecución.
- Los datos se graficarán utilizando Plotly en Python.
- Se identificarán posibles mejoras.

Capítulo 2

Desarrollo del controlador

2.1. Introducción

Este capítulo busca documentar el desarrollo e implementación del controlador FOC, partiendo la implementación de hardware revisando los parámetros y componentes utilizados, posteriormente una revisión a grandes rasgos de las configuraciones más importantes de los periféricos del STM32 en CubeMX para finalizar con la implementación del algoritmo del control FOC.

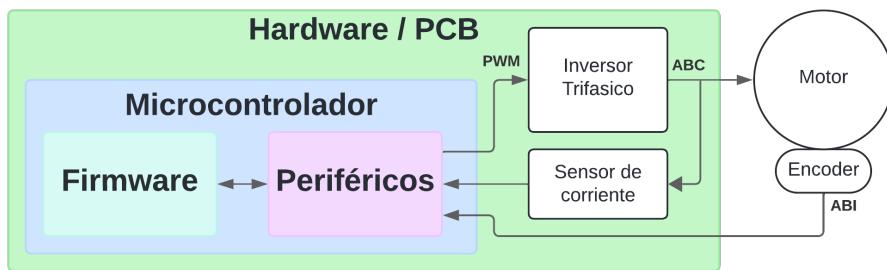


Figura 2.1: Diagrama Resumido del sistema.

2.2. Desarrollo del hardware

El hardware del controlador está principalmente conformado por el inversor trifásico, sensores de corriente y el motor.

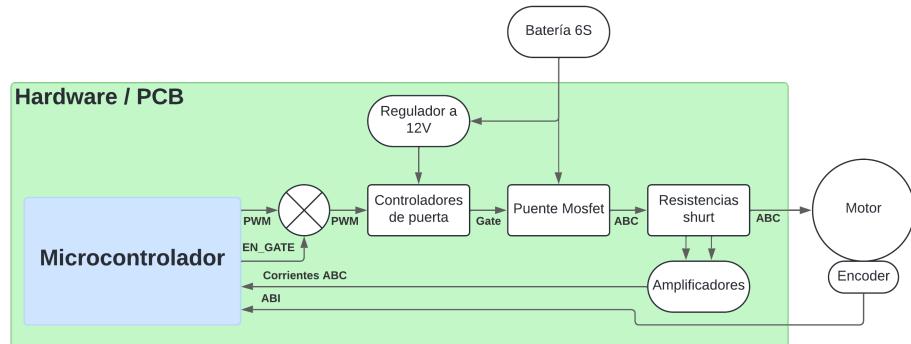


Figura 2.2: Diagrama Resumido del hardware.

2.2.1. Motor

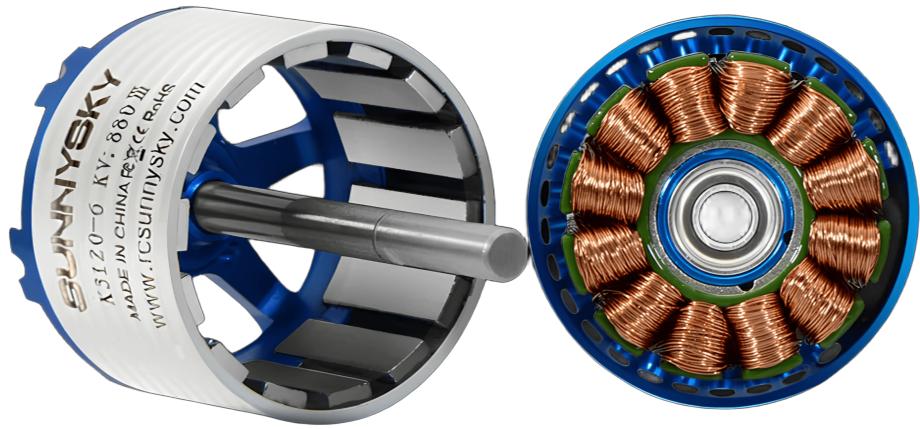


Figura 2.3: SunnySky X3120 V3.

Se ha optado por utilizar el motor **SunnySky X3120 V3** de 585Kv para el diseño del controlador debido a su disponibilidad inmediata. este es un motor principalmente orientado al uso en drones de tipo aeroplano, pero su gran potencia y bajo peso lo hace ideal para su uso en robots sumo.

Tabla 2.1: Características del motor SunnySky X3120 V3 585Kv

Propiedad	Valor
Número de polos del rotor	14
Número de ranuras del estátor	12
Máxima celda Lipo	6S (22.2 V)
Constante de velocidad del motor	585 K_v
Resistencia del motor	43.5 mΩ
Corriente continua máxima	65 A/30 s
Potencia continua máxima	1625 W
Corriente en vacío	1.0 A/10 V

Se toma como punto de partida el motor, ya que este es el que principalmente define los parámetros necesarios para el inversor trifásico y sensores de corriente. algunos de los parámetros mas relevantes para el diseño del hardware son el voltaje de funcionamiento, que esta marcado en 6S (celdas de lipo), las celdas de lipo varian su voltaje en función de su carga, teniendo un voltaje mínimo de 3V, nominal de 3.7V y máximo de 4.2V, por esto, una batería de 6 celdas en serie corresponde a un voltaje mínimo de 18V, nominal de 22.2V y máximo de 25.2V. otro parámetro importante es la corriente continua maxima de 65A.

2.2.2. Encoder

El encoder es el encargado de proveer la información sobre la posición angular del rotor en el motor, se utilizo el encoder magnético AS5047P, el cual utiliza un imán especial con una polarización diametral en el eje del motor, para poder medir la posición angular de este.



Figura 2.4: AS5047P.

En esencia este es un encoder absoluto que funciona principalmente a través de una comunicación SPI, pero opcionalmente incorpora una salida que imita el comportamiento de un encoder incremental con señal index (ABI), el microcontrolador STM32 tiene periféricos que pueden trabajar de forma nativa con estas señales sin mayor intervención del procesador, lo que simplifica en gran medida su implementación en comparación con el uso de SPI, aun que en consecuencia se pierden las capacidades de encoder absoluto.

Tabla 2.2: Características del encoder AS5047P

Parámetro	Mín	Típ	Máx	Unidad
Tensión de alimentación (LDO)	4.5	5.0	5.5	V
Tensión de alimentación	3.0	3.3	3.6	V
Corriente de suministro		15		mA
Velocidad máxima			28000	RPM
Resolución del núcleo		14		bits
Resolución ABI	25	512	1024	PPR
Resolución ABI (X4)	100	2048	4096	Pasos/rev

2.2.3. Conjunto motor encoder

El encoder requiere estar fijo junto al motor y centrado respecto al eje del motor, por esto se diseño en Autodesk Inventor y posteriormente se fabrico con impresión 3D, una montura hecha a medida para el conjunto motor-encoder.

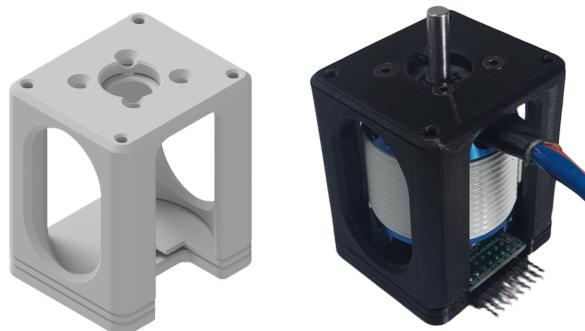


Figura 2.5: Montura motor-encoder.



Figura 2.6: Socket Encoder.

También se diseño un socket con el fin de mantener el imán necesario para el funcionamiento del en coder en el centro del eje del motor.



Figura 2.7: Socket imán axial.

2.2.4. Puente Mosfet

El puente mosfet es una de las partes del hardware mas critica, este se encarga de realizar las conmutaciones en las fases del motor, se divide en 3 piernas (o fases) iguales, donde cada pierna esta conformada por 2 mosfets de tipo N en configuración push-pull, es decir, se tiene un mosfet alto conectado a $V+$ y un mosfet bajo conectado a $V-$. cada mosfet tiene su propia señal de gate las cuales se deben controlar de forma adecuada para evitar cortocircuitos. es importante considerar que cada pierna del puente debe soportar por lo menos la corriente maxima del motor, aun que idealmente se deja un margen de seguridad de 2 a 3 veces la corriente maxima, considerando la corriente de 65A del sunnysky X3120, los mosfets deberán de soportar al menos 195A continuos y un voltaje Drain-source sobre los 25.2V para poder trabajar con la batería 6S.

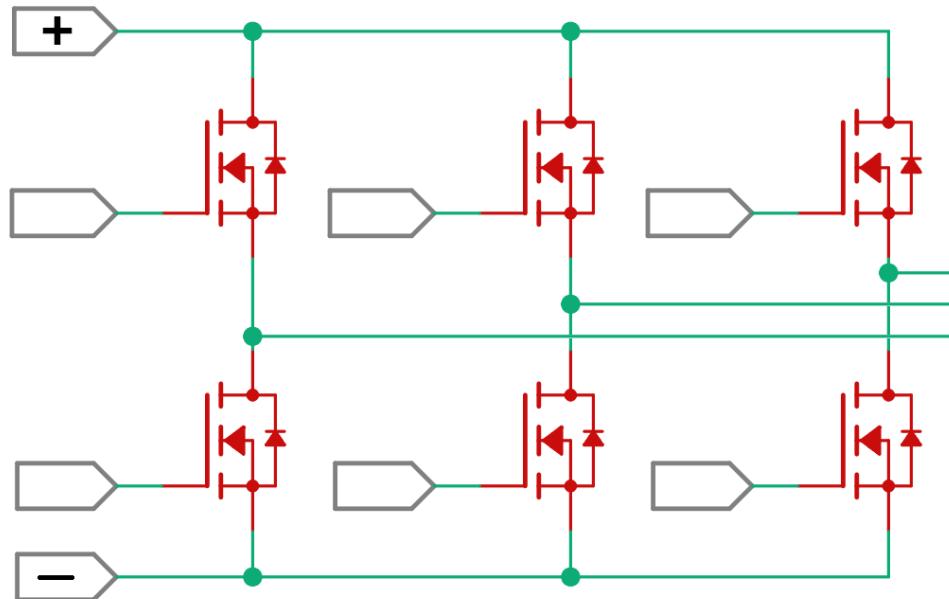


Figura 2.8: Esquema puente MOSFET.

Se utilizo el mosfet **BSC011N03LSI** de la marca Infineon, es un Mosfet de tipo N con un voltaje Drain-source de 30V, estos mosfets soportan una corriente continua de 100A.

Tabla 2.3: Características del MOSFET BSC011N03LSI.

Parámetro	Variable	Min	Typ	Max	Unidad
Tensión de drenaje-fuente	V_{DS}			30	V
Resistencia encendido	$R_{DS(on)}$	0.9	1.1	1.5	mΩ
Corriente continua de drenaje	I_D			100	A
Corriente de pulso drenaje	$I_{D,pulse}$			400	A
Tensión umbral del gate	$V_{GS(th)}$	1.2		2	V
Carga de puerta total	Q_g		34	45	nC
Potencia de dissipación	P_{tot}			96	W

Basado en las características del mosfet, utilizar solo 2 mosfets por pierna seria insuficiente si se quiere tener buen margen respecto a la corriente del motor, por esto se utilizo una configuración de doble mosfet en paralelo, de esta forma, aun que despreciando las pequeñas diferencias entre mosfets del mismo modelo, se podría considerar un capacidad total de 200A continuos.

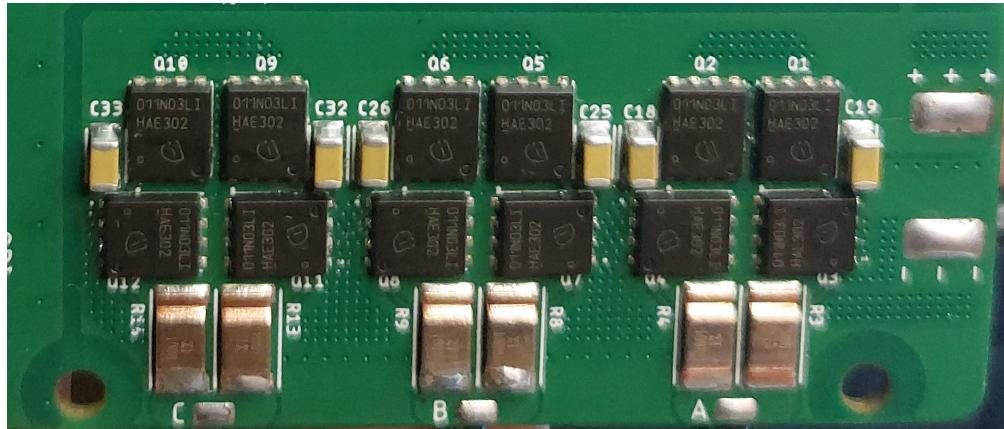


Figura 2.9: Puente MOSFET.

Por el reverso de la placa se agregaron capacitores electrolíticos de $470\mu F$ y capacitores cerámicos de $22\mu F$ por ambos lados para suplir en los picos de corriente que la batería no pueda, ademas de mejorar la estabilidad general de la alimentación del puente

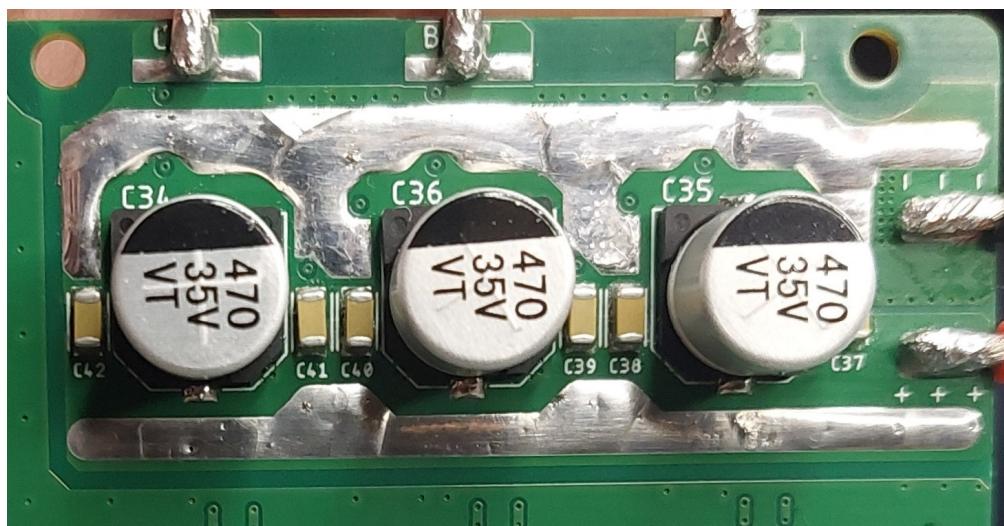


Figura 2.10: alimentación del Puente MOSFET.

2.2.5. Controladores de Puerta

El controlador de puerta (gate driver) es un componente esencial, ya que se encarga de recibir las señales PWM provenientes del microcontrolador y, en consecuencia, activar o desactivar los MOSFETs en cada pierna del puente. Su función principal es proporcionar la corriente y el voltaje adecuados para encender y apagar rápidamente los MOSFETs, garantizando una conmutación eficiente y minimizando las pérdidas de energía.

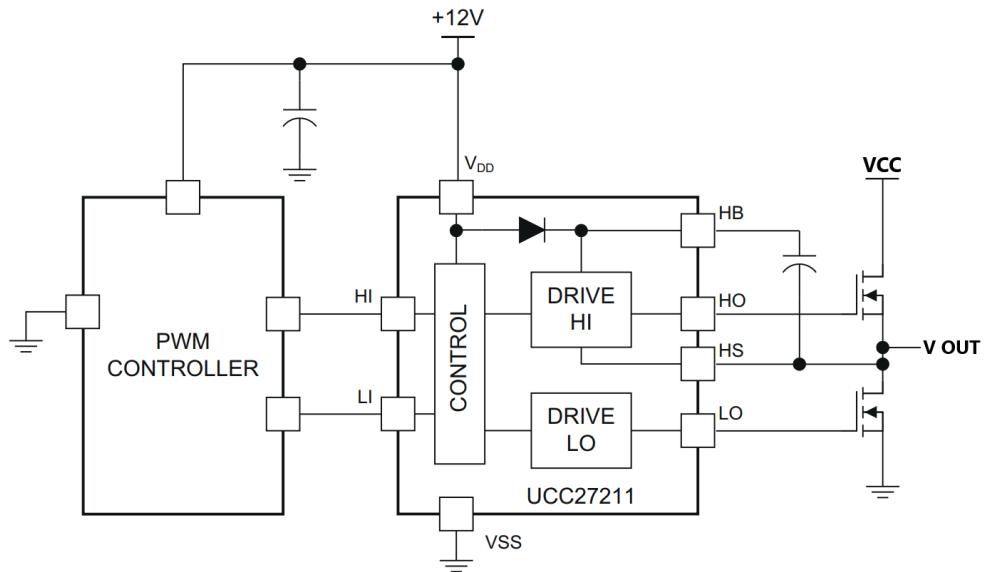


Figura 2.11: Esquema UCC27211.

Se utilizó el controlador de puerta **UCC27211** de Texas Instruments, el cual es un driver de doble canal diseñado específicamente para controlar pares de MOSFETs. Este dispositivo ofrece entradas de control independientes para los MOSFETs de alta (H_{In}) y baja (L_{In}), lo que permite un control preciso y flexible de cada transistor de potencia.

Tabla 2.4: Tabla de Lógica del UCC27211

H In	L In	H Out	L Out	V Out
L	L	L	L	Z
L	H	L	H	GND
H	L	H	L	VCC
H	H	H	H	Corto

Es importante destacar que el UCC2711 no incorpora protección interna contra la conducción simultánea de ambos MOSFETs (conocido como *shoot-through*). Este fenómeno ocurre ambos MOSFETs de una pierna conducen al mismo tiempo, generando un cortocircuito entre el suministro de voltaje y tierra. Para evitar esta situación es importante agregar un tiempo entre la conmutación de las señales de entrada conocido como *Dead-Time*, el cual se estima considerando los tiempos de conmutación del controlador de puerta junto al tiempo de conmutación del mosfet.

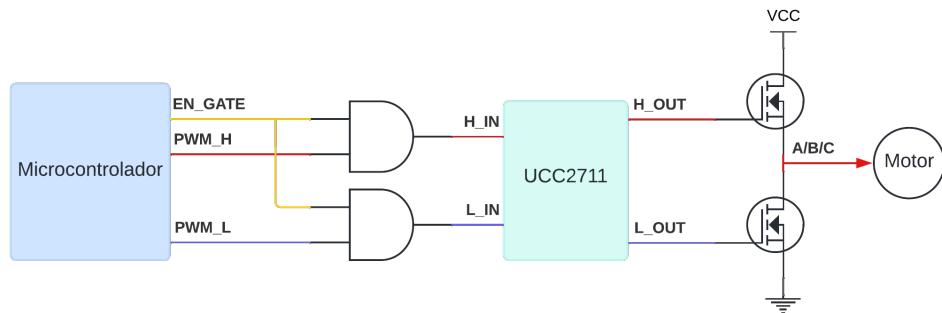


Figura 2.12: Esquema de control PWM con las compuertas AND.

El UCC27211 no dispone de un pin para activar o desactivar el control de los mosfets, por su parte, la única forma de desactivar el puente mosfet en su totalidad es llevar las señales de entrada alta (H In) y baja (L In) a un estado bajo (L), ademas el microcontrolador al realizar esta acción puede llegar a perder el control sobre el estado de las señales PWM y asi generar por un breve momento un estado de doble activación de los mosfet. para evitar esto se agregaron externamente comportas AND modelo SN74LVC1G08DBVR en cada señal PWM, con el fin de poder cortar las señales desde el microcontrolador de forma segura.

Tabla 2.5: Tabla de Lógica del UCC27211 con AND

EN GATE	PWM H	PWM L	H In	L In	H Out	L Out	V Out
H	L	L	L	L	L	L	Z
H	L	H	L	H	L	H	GND
H	H	L	H	L	H	L	VCC
H	H	H	H	H	H	H	Corto
L	X	X	L	L	L	L	Z

2.2.6. Sensores de Corriente

Para el sensor de corriente se opto por una configuración de resistencia shunt en la salida de los mosfet como se puede ver en la figura 2.9, la cual se combina con un amplificador instrumental especializado en esta aplicación INA240A1 que tiene una ganancia de 20V/V y se le aplica un offset equivalente a la mitad voltaje de referencia de 3.3V.

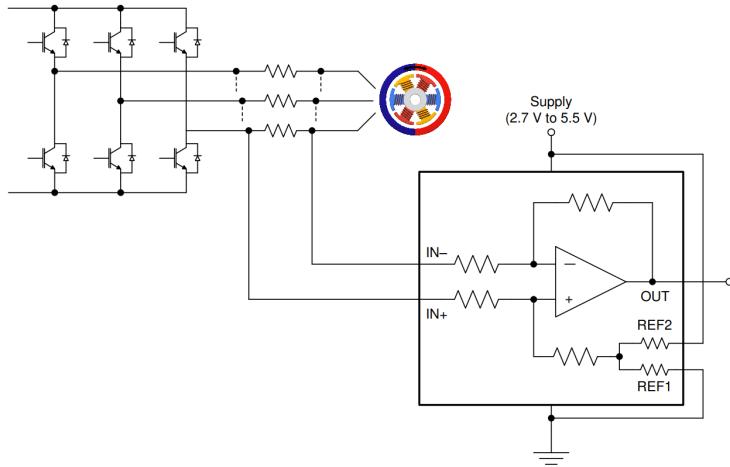


Figura 2.13: Esquema INA240.

Para la resistencia shunt se utilizo una doble resistencia en paralelo de $1\text{m}\Omega$ con una potencia de disipación maxima de 5W, dando asi en total, una resistencia equivalente de $500\mu\Omega$ y 10W de disipación, para determinar la corriente máxima que puede pasar la resistencia conociendo su valor en ohmios R y la potencia P en vatios, se puede calcular utilizando la siguiente fórmula:

$$I_{\max} = \sqrt{\frac{P}{R}}$$

Donde:

- I_{\max} es la corriente máxima,
- P es la potencia en vatios (W),
- R es el valor de la resistencia en ohmios (Ω).

De esta forma se determina que el máximo de corriente continua que puede circular por las resistencias es de 141.42A.

2.3. Implementación del hardware

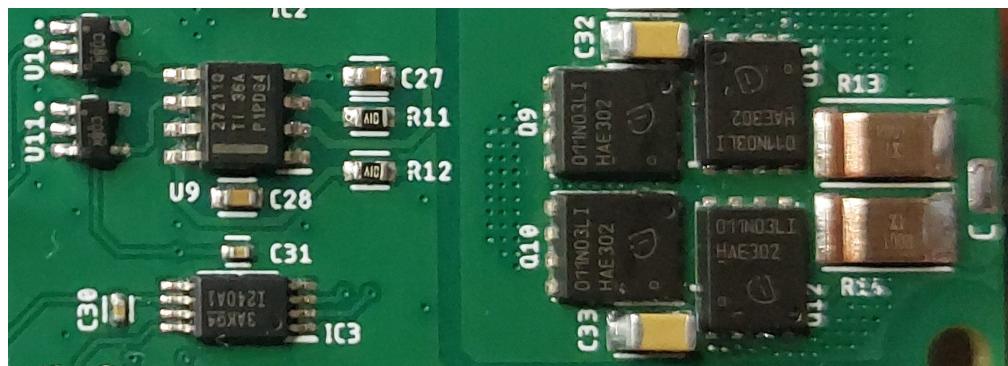


Figura 2.14: Pierna C del inversor trifásico.

2.4. Microcontrolador y periféricos

Se utilizo un microcontrolador de la serie H7 de STM32, esta es la serie de alto performance de STM32 que incorpora un núcleo ARM Cortex M7 con unidad de calculo para punto flotante de doble precision (FPU). el modelo especifico es STM32H743VIT6, su núcleo trabaja a 480MHz, tiene de 1MB de memoria RAM y 2MB de memoria flash, donde asi una potencia de computo y capacidad bastante elevada para un microcontrolador.

- **Core:** Arm Cortex-M7 de 32bits a 480 MHz, FPU de doble precisión.
- **Memoria Flash:** 2 MB.
- **RAM:** 1 MB total (192 KB de RAM DTCM, 864 KB de SRAM).
- **Voltaje de operación:** 1.62 V - 3.6 V.
- **Empaquetado:** LQFP-100 (14 x 14 mm).
- **Entradas/Salidas:** 82 I/O.
- **Periféricos de comunicación:** USART, UART, SPI, CAN FD, USB, Ethernet.
- **Timers:** Hasta 22 timers (PWM, encoder, control de motores, etc.).
- **ADC:** 3x ADC tipo SAR de 16 bits (hasta 3.6 MSPS).
- **Programación y Depuración:** Interfaz SWD y JTAG.

Los periféricos en un microcontrolador son piezas de hardware especializados en una función que van integrados en el microcontrolador y permiten interactuar con el mundo exterior a traves de sus señales eléctricas, ya sean entradas o salidas, digitales o análogas. en el ecosistema de STM32 existe la herramienta **STM32CubeMX** que permite configurar todos los periféricos del microcontrolador desde una interfaz gráficas fácil de entender para posteriormente poder generar un código C que implemente todas las configuraciones en el microcontrolador utilizando las librerías HAL de STM32, de esta forma **STM32CubeMX** genera un código base desde el que empezar a programar el firmware con todos los periféricos ya configurados.

2.4.1. Configuración de Timers

Los timers en los microcontroladores son en esencia módulos que permiten contar eventos y controlar pines en función de estos eventos, estos eventos pueden tener un origen interno al microcontrolador, como lo pueden ser las señales de reloj o pueden tener orígenes externos. STM32 tiene una gran diversidad de tipos de times con a su ves, una gran flexibilidad para su configuración.

Timer 1

El timer 1 es un timer de tipo **Advanced-control** de 16bits, este tipo de times son especializados en el control de las señales para puentes mosfets ya que permiten generar señales PWM con complementaria y dead time, ademas de configurar el contador de modo triangular (center-aligned), lo que facilita en gran medida su implementación para este propósito. este timer ademas se aprovecha para generar la interrupción donde se ejecute la rutinas para el control del motor.

Parámetro en CubeMX	Valor
Clock Source	Internal Clock (240MHz)
Counter Mode	Center Aligned mode 3
Counter Period (16bits)	5000
Trigger Event Selection TRGO	Update Event
Dead Time	50
Channel 1	PWM Generation CH1 CH1N
PWM Generation Channel 1 and 1N Mode	PWM mode 1
Channel 2	PWM Generation CH2 CH2N
PWM Generation Channel 2 and 2N Mode	PWM mode 1
Channel 3	PWM Generation CH3 CH3N
PWM Generation Channel 3 and 3N Mode	PWM mode 1

Tabla 2.6: Configuración del timer 1 en CubeMX

algunos aspectos claves en la configuración del timer 1 es por una parte el uso de el modo de conteo en **Center Aligned mode 3**, este modo genera un contador acendente-descendente o triangular y esta configurado para contar los cilos del reloj interndo de 240MHz, en este modo al alcanzar el valor conteo configurado en 5000, el contador pasa al ciclo descendente hasta llegar a 0 para volver a repetir, en cada uno de estos eventos de desborde, el timer ejecuta una interrupcion la cual se aprovecha para realizar todos los cálculos del sistema.

Timer 3

El timer 3 es un timer de uso general de 16bits, este es utilizado en modo encoder, lo que permite contar los pulsos AB de un encoder incremental sin intervención del firmware, lo que optimiza el uso de recursos para la obtención de la posición del encoder.

Parámetro en CubeMX	Valor
Combined Channels	Encoder Mode
Counter Period (16bits)	65535
Encoder Mode	Encoder Mode TI1 and TI2
Input Filter	4

Tabla 2.7: Configuración del timer 3 en CubeMX

El modo TI1 and TI2 permite recibir las señales del puerto incremental en X4, es decir, 4 pulsos por paso, ademas de fijar un filtro de entrada de 4 ciclos de reloj (de 240MHz), esto permite tener

un contra ruido y pulsos no deseados. el periodo del contador se fijo en el valor máximo para un registro de 16bits, es decir $2^{16} - 1$.

Timer 2

el timer 2 es un timer de propósito general de 32bits, este se utiliza para obtener tiempos precisos de las señales, principalmente para obtener el delta tiempo entre pulsos del encoder para un mejor calculo de la velocidad del motor, esto se logra agregando una compuerta Xor entre las señales AB del encoder

Parámetro en CubeMX	Valor
Trigger Source	ITR0 (Timer 1 trigger out)
Clock Source	Internal Clock (240MHz)
Counter Mode	Up
Counter Period (32 bits)	4294967295
Channel 1 IC Selection	Input Capture triggered by TRC TRC
Channel 2 Polarity Selection Input Filter	Input Capture direct mode Both Edges 4
Channel 3 Polarity Selection Input Filter	Input Capture direct mode Rising Edge 4

Tabla 2.8: Configuración del timer 2 en CubeMX

el modo Input Capture permite registrar el valor del contador en el momento de tener una señal, el canal 1 esta vinculado a la señal trigger out del timer 1, el canal 2 esta vinculado a la compuerta XOR en el encoder, esta configurado en Both Edges, para que de esta forma se registre el tiempo al tener un cambio de estado en esta señal. el canal 3 esta conectado a la señal index del encoder y solo registra el tiempo en el flanco ascendente (Rising Edge).

Timer 15

el timer 15 es un timer básico de 16 bits, hace de intermediario entre el timer 1 y el ADC, este timer mantiene una sincronización con el timer 1 y esta configurado para generar una señal de salida en su trigger out en algunos ciclos antes del desborde del contador en el timer 1, esta señal es utilizada para activar la lectura de las lecturas de corriente en el ADC.

Parámetro en CubeMX	Valor
Slave Mode	Combined Reset Trigger Mode
Trigger Source	ITR0 (Timer 1 trigger out)
Clock Source	Internal Clock (240MHz)
Counter Mode	Up
Counter Period (16 bits)	5000
Master/Slave Mode	Enable
Trigger Event Selection TRGO	Compare Pulse (OC1)
Channel 1	Output Compare No Output
Mode	Toggle on match
Pulse (16 bits)	4975

Tabla 2.9: Configuración del timer 15 en CubeMX

2.4.2. Configuración de ADCs

Los conversores analogo digital (ADC), son los encargados transformas voltajes analogos en un rango especifico a un valor digital. En los microcontroladores STM32 de la serie H7 se incorporan tres modulos ADC con una Resolución de 16bits. en estos ADC existen 2 modos leer los canales, siendo la principal la conversion regular la cual permite leer hasta 16 canales en una secuencia configurable y se puede combinar con el modulo DMA, que permite transferir registros de los perifericos a la memoria RAM sin intervencion del microcontrolador. tambien esta el modo injectado, el cual es una especie de interrupcion en el ADC, el cual interrumpe la conversion regular para realizar hasta 4 conversiones tambien en una secuencia configurable.

ADC 1

El ADC 1 se utiliza para la lectura de los sensores de corriente en las fases del motor, esta configurado para que dada la señal del trigger out del timer 15 se realicen las 2 lecturas de sensores

ADC 3

el ADC 3 esta configurado con una lectura regular del voltaje de bus,

2.5. Firmware

Capítulo 3

Validación

Para las validaciones, se realizó una adquisición de datos, donde estos se recopilan ciclo a ciclo en la memoria RAM del microcontrolador para su posterior envío de forma asincrónica por el puerto USB, como se muestra en la figura 3.1. Los datos son recibidos en una computadora a través de un código en Python, donde se almacenan en archivos CSV. Esta recopilación de información se realiza al final de cada ejecución de la interrupción del sistema, la cual se ejecuta a una frecuencia de 48 kHz. Debido a las limitaciones de memoria, la recopilación de datos solo puede llevarse a cabo durante un período de aproximadamente 100 ms, hasta alcanzar la capacidad máxima de la memoria.

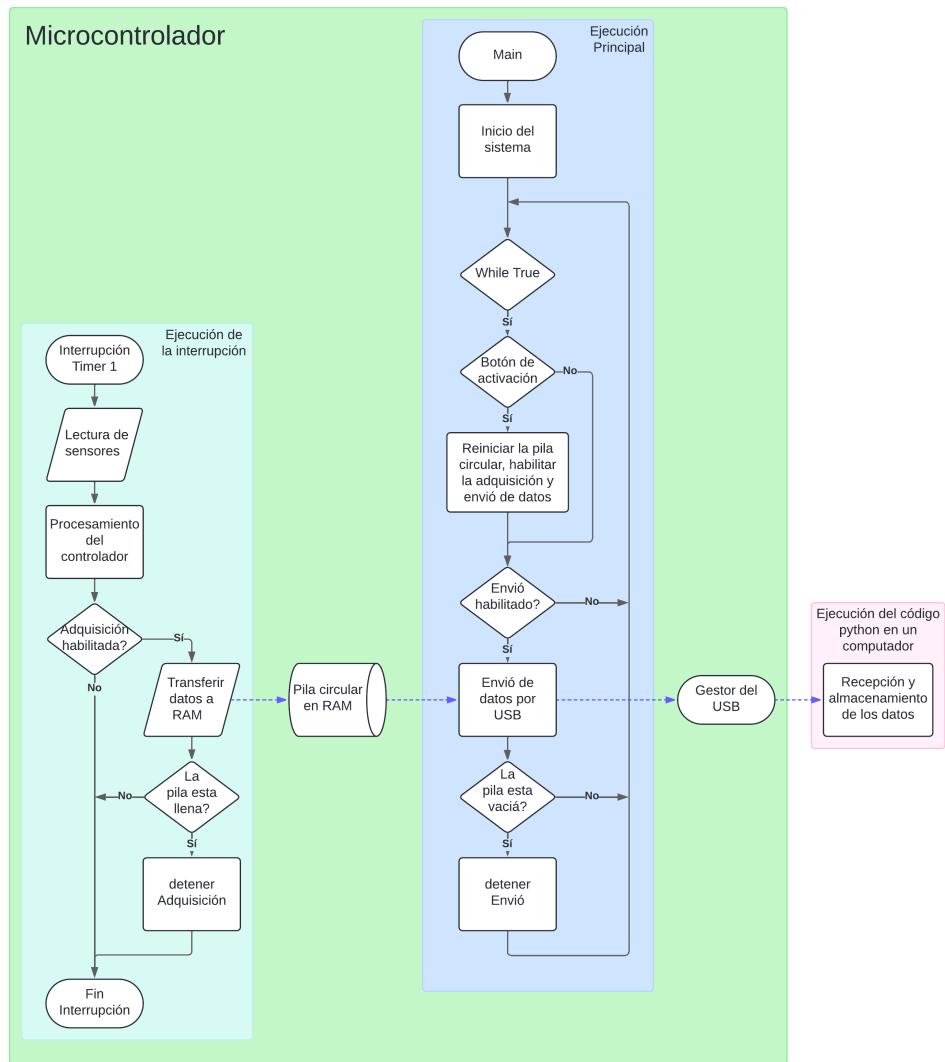


Figura 3.1: Diagrama de flujo adquisición de datos.

3.1. Validación de la adquisición y transformación de las mediciones de corriente

3.1.1. Validación de las mediciones de corriente

Se validarán las mediciones de corriente adquiridas desde el microcontrolador.

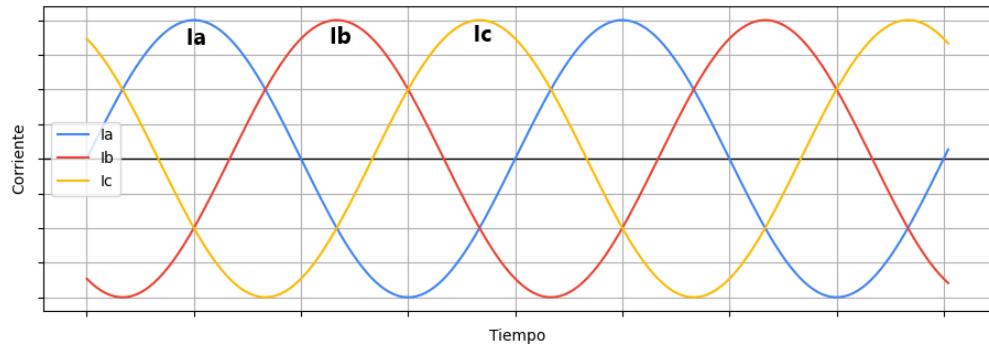


Figura 3.2: Corrientes ideales en un sistema trifásico equilibrado.

En la figura 3.2 se representan las señales ideales de un sistema trifásico equilibrado, con las tres corrientes de fase I_a , I_b e I_c , cada una de ellas con una forma senoidal pura y un desfase de 120° entre sí.

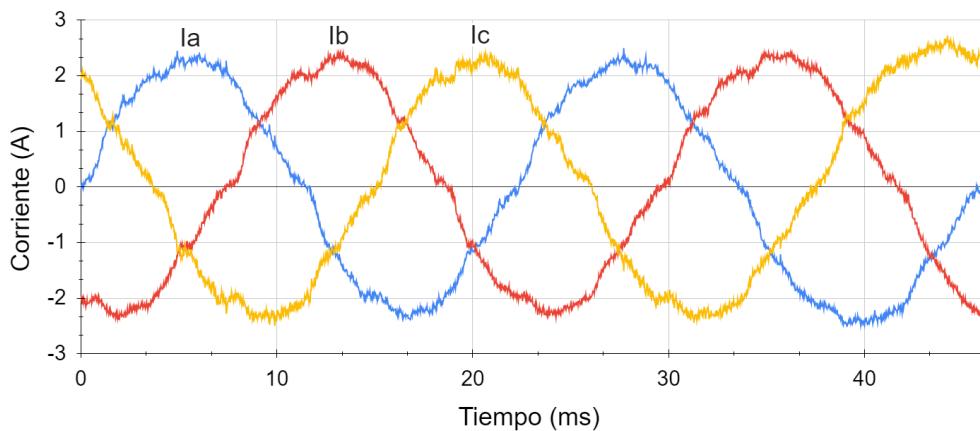


Figura 3.3: Corrientes medidas en el sistema trifásico.

En la figura 3.3 se pueden apreciar las corrientes de fase, las cuales presentan una cantidad significativa de ruido, aun cuando internamente pasan por un filtro complementario con una frecuencia de corte $f_W = 12000\text{Hz}$, pero mantienen un comportamiento sinusoidal con el desfase de 120° entre señales, como es característico de un sistema trifásico equilibrado. Pero la cantidad de ruido podría indicar que sería necesario disminuir la frecuencia de corte en el filtro complementario o agregar un filtro pasivo en el circuito.

3.1.2. Validación de la transformada de Clarke

Se validara los resultado a la salida de la transformada de Clarke en el microcontrolador.

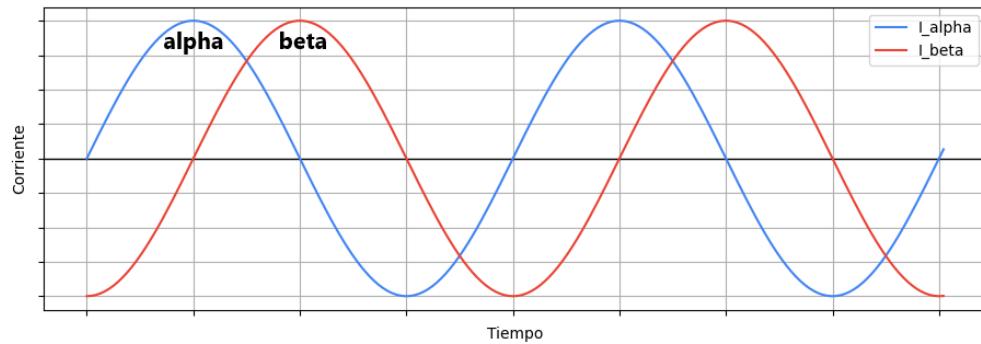


Figura 3.4: Corrientes ideales en el plano $\alpha\beta$.

En la figura 3.4 se representan las corrientes ideales en el plano $\alpha\beta$, con las formas senoidales puras con un desfase de 90° entre sí.

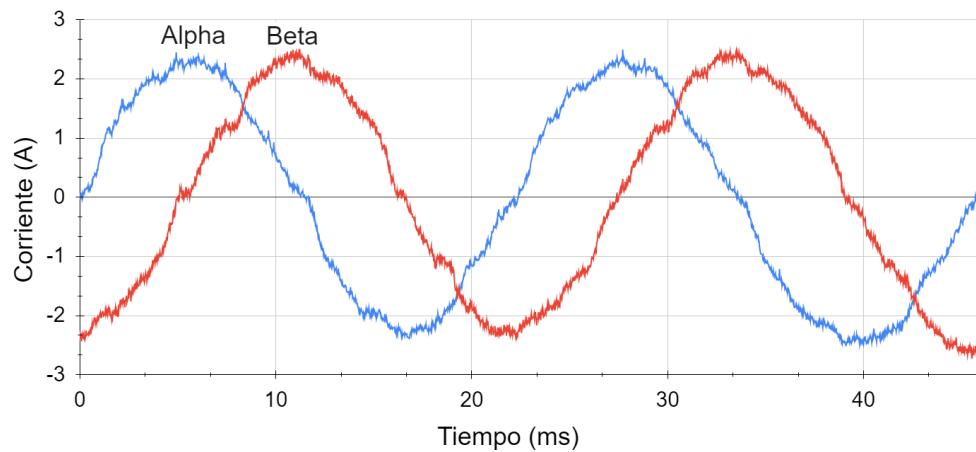


Figura 3.5: Corrientes medidas en el plano $\alpha\beta$.

En la figura 3.5 se pueden apreciar como las corrientes $\alpha\beta$ reflejan el ruido presente en las mediciones de los sensores de corriente, pero mantienen un comportamiento esperado a la salida de la transformada de Clarke con las dos señales sinusoidal con el desfase de 90° entre señales, como es característico.

3.1.3. Validación de la transformada de Park

Se validara los resultado a la salida de la transformada de Park en el microcontrolador.

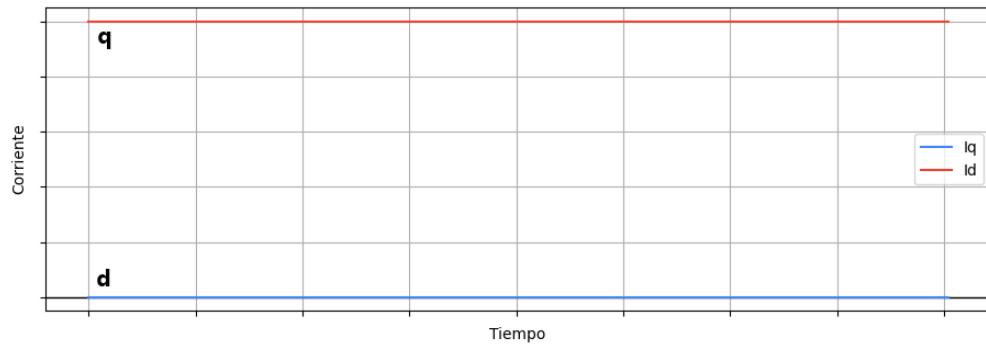


Figura 3.6: Corrientes ideales en el plano dq .

En la figura 3.6 se representan las corrientes ideales en el plano dq , donde lo ideal, es que la corriente directa tenga un valor de cero, para mantener la eficiencia del sistema, mientras que solo la corriente de cuadratura tiene un valor distinto a cero.

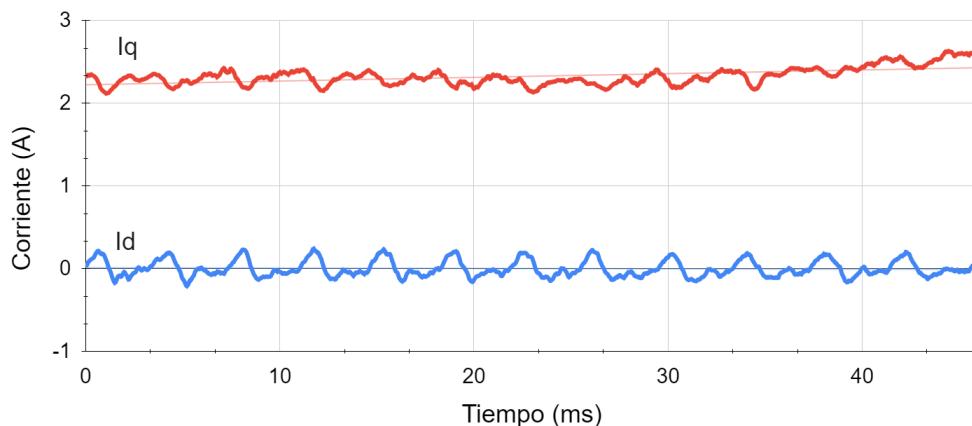


Figura 3.7: Corrientes medidas en el plano dq .

En la figura 3.7 se pueden apreciar como las corrientes dq presenta una menor cantidad de ruido gracias a que su filtro complementario esta ajustado para una frecuencia de corte de $f_W = 800Hz$, aunque igualmente presentan ciertas deformaciones y e inestabilidad con un patron aparentemente constante, pero en términos generales mantienen aproximadamente el comportamiento esperado a la salida de la transformada de Park.

3.2. Validación de los Controladores PI

En esta validación se busca comprobar si los controladores PI de velocidad y corriente son capaces de mantener sus setpoints. Las pruebas se realizaron de forma estática, aplicando una carga ligera sobre el motor y capturando datos durante este proceso.

3.2.1. Validación del controlador de velocidad

Para la validación, se aplicó un setpoint de 116.8 RPM utilizando uno de los potenciómetros disponibles.

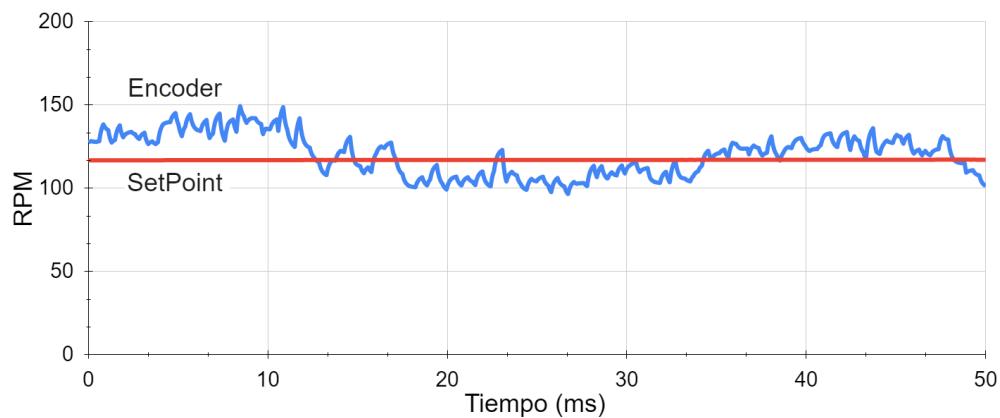


Figura 3.8: Velocidad medida por el encoder y setpoint de velocidad.

En la Figura 3.8, se observa que la velocidad medida por el encoder sigue adecuadamente el setpoint establecido de 116.8 RPM. A pesar de ligeras oscilaciones, el controlador de velocidad mantiene el régimen deseado, demostrando su capacidad para alcanzar y mantener el setpoint bajo condiciones de carga estática.

3.2.2. Validación del controlador de corriente

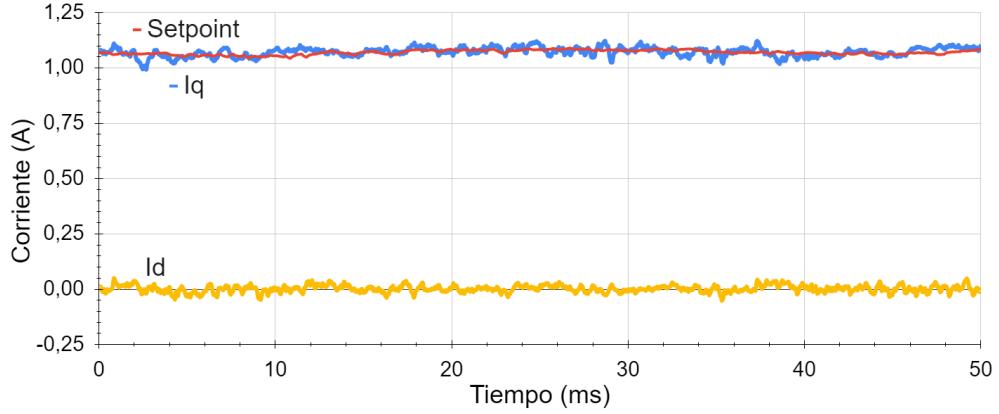


Figura 3.9: Corrientes medidas en el plano dq .

Como se muestra en la Figura 3.9, las corrientes en el plano dq indican que el controlador de corriente logra mantener la corriente de cuadratura (I_q) cercana al valor de referencia proporcionado por el controlador de velocidad, mientras que la corriente directa (I_d) se mantiene próxima a cero. Esto evidencia que el controlador de corriente regula eficazmente las corrientes según los setpoints establecidos.

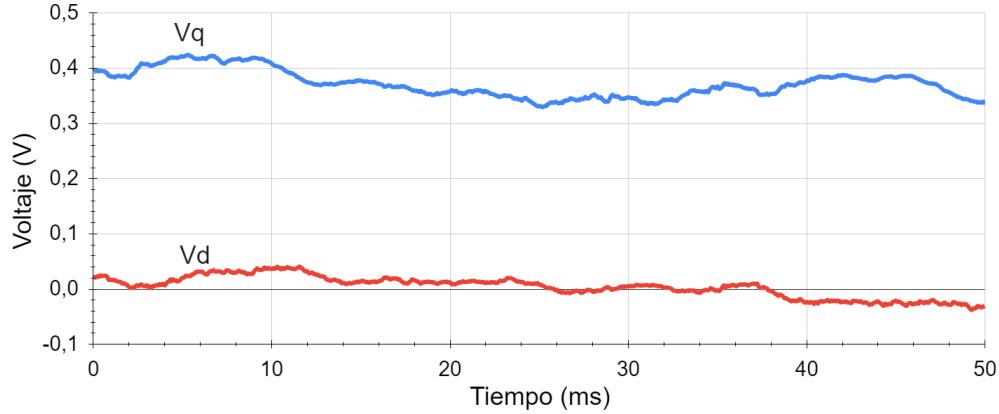


Figura 3.10: Voltajes en el plano dq .

Además, la Figura 3.10 presenta los voltajes en el plano dq , donde se aprecia que las tensiones generadas están dentro de los valores esperados para mantener las corrientes deseadas. Esto corrobora que el controlador de corriente responde adecuadamente a las demandas del sistema, contribuyendo al correcto desempeño del motor bajo condiciones de carga.

3.3. Validación señales del SVM

Para validar el funcionamiento del modulador por vector espacial (SVM), se realizaron pruebas virtuales aplicando diferentes valores de tensión de referencia V_{ref} al SVM para observar su comportamiento. Durante estas pruebas, se utilizó un analizador lógico para capturar las señales PWM generadas por el microcontrolador. En la captura presentada, se aisló un ciclo completo del PWM donde se puede apreciar de forma clara la secuencia de activación correspondiente al sector 1.

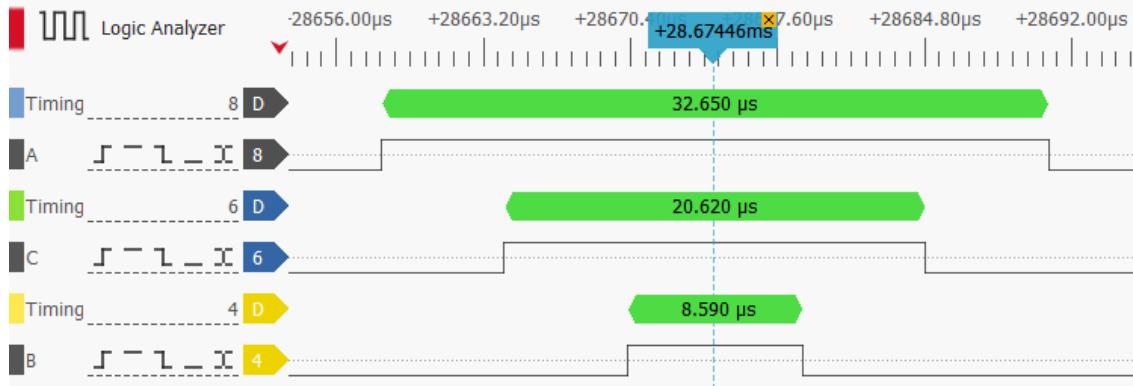


Figura 3.11: Señales del timer.

En la Figura 3.11, se observan las señales PWM correspondientes a las tres fases generadas por el microcontrolador. La secuencia de activación muestra que el SVM implementado sigue correctamente el patrón teórico esperado para el sector 1, evidenciando una conmutación precisa y sincronizada de los transistores del inversor. Esto confirma que el SVM está modulando adecuadamente las señales PWM para generar los vectores de tensión requeridos.

Comentarios y Conclusiones

Bibliografía

- [1] Charles Frick. *Brushless DC Motors Introduction for Next-Generation Missile Actuation Systems*. Technical Article TA20788-0-10/18. Norwood, MA, USA: Analog Devices, Inc., 2018. URL: <https://www.analog.com/media/en/technical-documentation/technical-articles/brushless-dc-motors-introduction-for-next-generation-missile-actuation-systems.pdf>.
- [2] Pete Millet. *Brushless vs. Brushed DC Motors: When and Why to Choose One Over the Other*. MonolithicPower.com. Jun. de 2022. URL: <https://MonolithicPower.com>.
- [3] Deepak Mohanraj et al. «A Review of BLDC Motor: State of Art, Advanced Control Techniques, and Applications». En: (2022). DOI: 10.1109/ACCESS.2022.3175011. URL: <https://ieeexplore.ieee.org/abstract/document/9774372>.
- [4] Matias Gualtieri Lara. «Sensorless Brushless DC Motors. Development and comparison of different fault tolerant control algorithms». Relatore: Prof. Paolo Maggiore. Tesi di Laurea Magistrale. Corso di laurea magistrale in Ingegneria Aerospaziale: Politecnico di Torino, mar. de 2018, págs. 1-49. URL: <http://webthesis.biblio.polito.it/id/eprint/6891>.
- [5] Padmaraja Yedamale. *Brushless DC (BLDC) Motor Fundamentals*. AN885. Jul. de 2003. URL: <https://www.microchip.com/en-us/application-notes/an885>.
- [6] Bin Wu. *High-Power Converters and AC Drives*. 1.^a ed. Hoboken, NJ: Wiley-IEEE Press, 2006, págs. 296-306. ISBN: 978-0471731719. URL: <https://ieeexplore.ieee.org/servlet/opac?bknumber=5237895>.
- [7] Antun Skuric et al. «SimpleFOC: A Field Oriented Control (FOC) Library for Controlling Brushless Direct Current (BLDC) and Stepper Motors». En: *Journal of Open Source Software* 7.74 (jun. de 2022), pág. 4232. DOI: 10.21105/joss.04232. URL: <https://doi.org/10.21105/joss.04232>.
- [8] Jorge Zambada y Debraj Deb. *Sensorless Field Oriented Control of a PMSM using a Sliding Mode Observer (SMO)*. AN1078. Mayo de 2010. URL: <https://www.microchip.com/en-us/application-notes/an1078>.
- [9] Bin Wu. *High-Power Converters and AC Drives*. 1.^a ed. Hoboken, NJ: Wiley-IEEE Press, 2006, págs. 101-109. ISBN: 978-0471731719. URL: <https://ieeexplore.ieee.org/servlet/opac?bknumber=5237895>.

Anexo A