

Prototipo de inversor trifásico para motor BLDC: modelado analítico y validación experimental del diseño.

Dryan López
dryan.lopez@outlook.com

I. INTRODUCCIÓN

Los motores BLDC (Brushless Direct Current) se utilizan ampliamente en aplicaciones industriales, automotrices y domésticas debido a su alta eficiencia energética, densidad de potencia, baja necesidad de mantenimiento y capacidad para ofrecer un control preciso de velocidad y par. A diferencia de los motores de corriente continua con escobillas, los BLDC sustituyen el conmutador mecánico por un sistema electrónico, lo que elimina el desgaste por fricción y mejora la confiabilidad operativa.

No obstante, esta ausencia de conmutación mecánica implica la necesidad de un sistema de conmutación electrónica capaz de generar las tres fases de excitación del estator de manera sincronizada con la posición del rotor. Dicho sistema se implementa mediante un inversor trifásico, que convierte una fuente de corriente continua en tensiones alternas desfasadas 120° entre sí. Este subsistema es esencial para accionar el motor, pero introduce una capa adicional de complejidad en el diseño.

El presente documento se enfoca en el diseño y análisis de un inversor trifásico básico para el accionamiento de un motor BLDC en configuración de lazo abierto, construido con componentes discretos. Se desarrolla un modelo matemático que describe la dinámica de conmutación de los MOSFET y sirve como base para estimar los tiempos característicos del circuito, los cuales se comparan posteriormente con mediciones obtenidas en el prototipo.

II. ARQUITECTURA

El sistema desarrollado tiene como propósito accionar un motor BLDC mediante un inversor trifásico en lazo abierto, empleando componentes discretos y control digital básico. La arquitectura general se divide en tres bloques principales: fuente de poder, bloque de potencia y bloque de control.

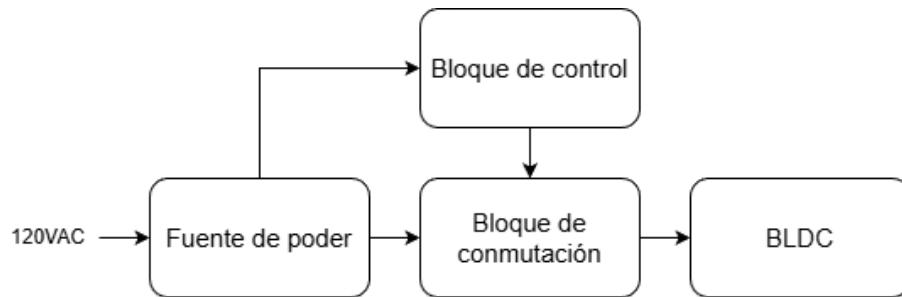


Figura 1: Arquitectura del inversor trifásico. Fuente: elaboración propia.

III. DESCRIPCIÓN DEL SISTEMA

III-A. Fuente de poder

La fuente de alimentación utilizada es una unidad con salida de $12 V_{DC}$ y una corriente máxima de 14.2 A (modelo PB-2171-02M1, revisión A). Esta tensión se encuentra dentro del rango operativo típico para motores BLDC de baja potencia y es adecuada para ensayos experimentales o de laboratorio.

Aunque la corriente del motor supera la capacidad de la fuente bajo condiciones de carga completa, su uso resulta apropiado para pruebas en vacío o con cargas parciales, donde la demanda de corriente se mantiene moderada. En estas condiciones, se garantiza la integridad del sistema y la estabilidad del suministro sin riesgo de disparo por sobrecorriente.

La salida de la fuente se conecta directamente al bus de corriente continua del inversor, el cual alimenta las tres ramas del puente trifásico. Se incluye un capacitor electrolítico de filtrado en el bus DC para reducir el rizado de tensión producido por las conmutaciones de los transistores.

III-B. Bloque de potencia

El bloque de potencia está conformado por un puente trifásico compuesto por seis transistores MOSFET de canal N, dispuestos en tres ramas correspondientes a las fases del motor. Cada rama contiene un transistor en la parte superior (alto lado) y otro en la parte inferior (bajo lado), los cuales conmutan de forma complementaria para generar las señales trifásicas necesarias para el estator.

La conmutación de los MOSFET se realiza en pares, de modo que en cada instante solo dos fases del motor están energizadas mientras la tercera permanece flotante, siguiendo la secuencia de seis pasos típica de los motores BLDC de conmutación trapezoidal.

III-C. Bloque de control

El bloque de control tiene la función de generar la secuencia de pulsos que define la conmutación de las fases del motor. Esta secuencia se implementa mediante un microcontrolador o un generador de señales programable, que produce los pulsos PWM correspondientes a los seis transistores del inversor.

El circuito de control también incluye las etapas de acondicionamiento necesarias para el manejo de las compuertas, utilizando controladores de compuerta (gate drivers) capaces de proporcionar la corriente de carga y descarga adecuada a los MOSFET de alto y bajo lado.

En la figura 2 se muestra el diagrama de la implementación con el puente trifásico en detalle.

III-D. Especificaciones del motor

En el cuadro I se brindan las especificaciones del motor a utilizar.

Cuadro I: Especificaciones del motor

Especificación	Valor
Constante de velocidad (K_v)	2650
Configuración del devanado	Y
Tensión máxima de entrada	16.8 V
Corriente nominal	120 A
Peso	13.1 oz (370 g)
Diámetro del motor	39.98 / 41.78 mm con aletas
Longitud del motor	68 mm
Diámetro del eje	5 mm
Rodamientos	ABEC-1 de gran tamaño frontal y trasero
Conecotor	Oro de 6.5 mm

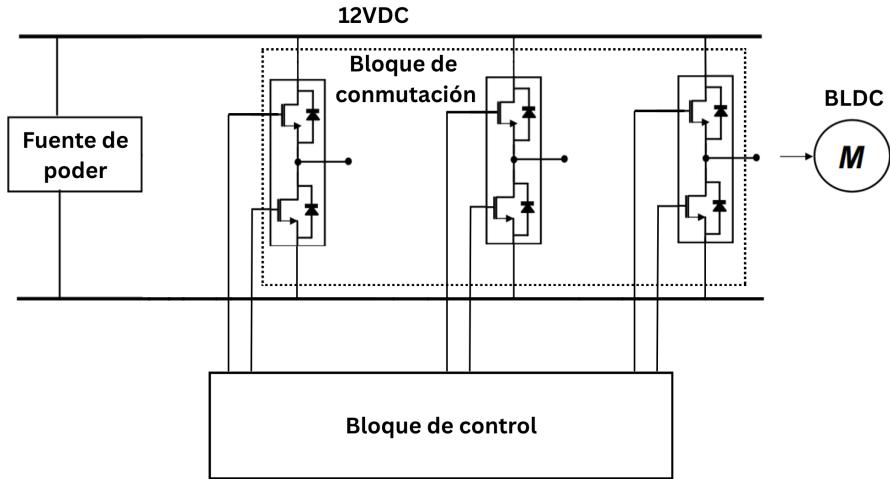


Figura 2: Diagrama de la implementación.

IV. SELECCIÓN DE COMPONENTES

IV-1. Microcontrolador: Se selecciona el microcontrolador ESP32 debido a su disponibilidad y a sus prestaciones adecuadas para aplicaciones de control de potencia. Este dispositivo incorpora un conjunto de periféricos de control de motores, entre ellos el módulo MCPWM (Motor Control Pulse Width Modulator), accesible mediante la librería `mcpwm.h`. Dicho módulo permite generar señales PWM con resolución programable, control de frecuencia y ciclo de trabajo independientes por canal, sincronización de fases, inserción de tiempos muertos (dead time) y modos complementarios, características esenciales para la conmutación segura y eficiente de un puente trifásico. La programación para este inversor se puede encontrar en el apéndice o en la página de GitHub del proyecto.

IV-A. Dispositivo de conmutación

Dado que el sistema opera con tensiones inferiores a 20 V, se selecciona un transistor MOSFET como dispositivo de conmutación en el puente trifásico. Los MOSFET resultan más adecuados que los IGBT en este rango de tensión, ya que presentan una menor caída de tensión en conducción, conmutación más rápida y menores pérdidas dinámicas, lo cual se traduce en una mayor eficiencia general del inversor.

Para garantizar un margen de seguridad frente a transitorios de conmutación y posibles sobretensiones en el bus de continua, se elige un MOSFET con un voltaje drenaje-fuente (V_{DS}) nominal al menos dos veces superior al voltaje de la fuente de alimentación. Además, la corriente máxima del dispositivo debe superar la corriente de salida máxima esperada de la fuente, de modo que el transistor pueda operar dentro de su región segura sin riesgo de sobrecalentamiento o degradación.

En este diseño se utiliza el modelo IRFZ44N, seleccionado principalmente por su disponibilidad y características adecuadas para aplicaciones de baja tensión. Este dispositivo posee un (V_{DS}) máximo

de 55 V y una corriente de drenaje continua de hasta 49 A, valores suficientes para las condiciones de operación establecidas. Su resistencia en conducción ($R_{DS(on)}$) es de aproximadamente 17.5 mΩ. En aplicaciones donde el motor opera de forma continua o bajo carga elevada, se recomienda emplear MOSFETs con ($R_{DS(on)}$) en el orden de pocos miliohmios para optimizar la eficiencia térmica. En este caso, dado que el sistema se utiliza de manera intermitente y con cargas mecánicas ligeras, la temperatura de los transistores se mantiene dentro de rangos seguros sin requerir disipación adicional significativa.

Según la hoja de datos para el IRFZ44N, la carga Q_{gd} tiene un valor máximo de 23 nC. Por otro lado, según la figura 3, la capacitancia de entrada (C_{iss}) es de aproximadamente 1625 pF.

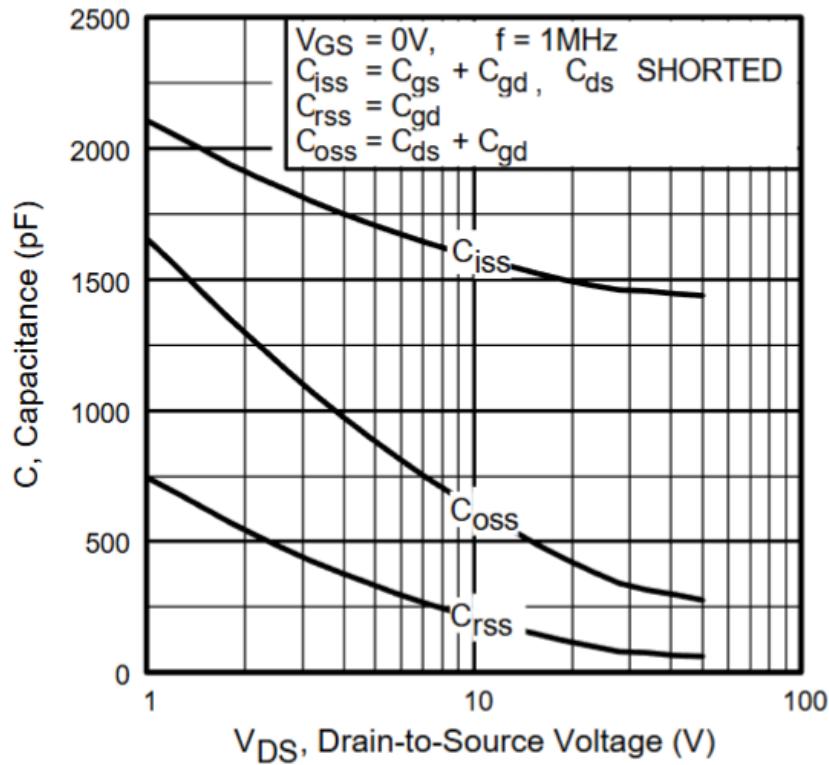


Figura 3: Capacitancia típica según V_{ds} . Fuente: .

El voltaje de meseta de compuerta (V_{plt}) se determina mediante pruebas en laboratorio, de donde se obtiene un valor de 3.6 V.

IV-B. Controlador de compuerta

IV-B1. Convertidor de nivel: Un convertidor de nivel es necesario para elevar las señales de 3.3V brindadas por el microcontrolador a un voltaje adecuado para la activación de los dispositivos de commutación. El transistor elegido para el convertidor de nivel corresponde al BC547B, debido a su amplia disponibilidad, bajo costo y características adecuadas para la commutación de señales de control. Este transistor bipolar de propósito general soporta tensiones colector-emisor de hasta 45 V y corrientes

de colector de hasta 100 mA, valores suficientes para el manejo de señales lógicas provenientes del microcontrolador.

El circuito implementado se muestra en la figura 4. Este circuito corresponde a un convertidor de nivel no inversor con una baja cantidad de componentes. Las resistencias R_c y R_b se eligen de forma que el transistor opere en la zona activa, se elige la resistencia R_c más pequeña que no supere la corriente de colector máxima del BJT, en este caso 470 Ω se considera un valor adecuado, para este valor de resistencia R_c , la resistencia R_b que hace al BJT entrar en zona activa se determina en 24.5 kΩ según las pruebas en laboratorio. El capacitor C_b tiene la función de aumentar la velocidad de conmutación del transistor, se elige el valor más pequeño de capacitancia disponible que produzca un tiempo de conmutación adecuado, en este caso la capacitancia más pequeña con la que se cuenta en el laboratorio es de 1 nF. El valor del voltaje entre colector y emisor (V_b) para esta configuración se determina en aproximadamente 1.2 V según las pruebas de laboratorio.

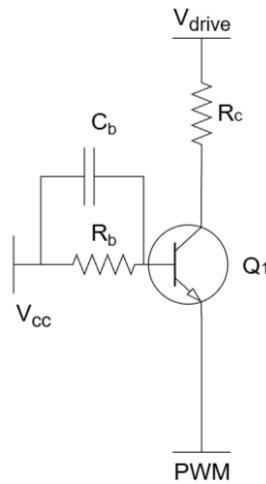


Figura 4: Convertidor de nivel. Fuente: elaboración propia.

IV-B2. Controlador de compuerta del lado bajo: Se elige una frecuencia de conmutación de 25 kHz para evitar ruido audible, para esta frecuencia de conmutación y un ciclo de trabajo mínimo de 5 % se obtiene un tiempo mínimo de conmutación según la ecuación 1. Se elige un tiempo de conmutación (t_{sw}) 10 veces menor al tiempo mínimo de ciclo de trabajo, según esto, t_{sw} tendría un valor de 200 ns.

$$t_{sw} = 5\% \cdot \frac{1}{25 \text{ kHz} \cdot 10} = 200 \text{ ns} \quad (1)$$

Para el controlador de compuerta se utiliza una configuración totem-pole de transistores bipolares en conjunto con el convertidor de nivel, esta configuración proporciona la corriente necesaria para la conmutación rápida de las compuertas. El circuito utilizado se muestra en la figura 5.

La resistencia ($R_{g(off)}$) entra en funcionamiento únicamente durante la descarga debido a que el diodo en serie (D_1) solo se polariza durante las descargas. Cuando el diodo se polariza, las resistencias $R_{g(off)}$ y $R_{g(on)}$ se encuentran en una configuración en paralelo, lo que reduce la resistencia, y por lo tanto, el tiempo de descarga de la compuerta. Un diodo de señal sencillo puede usarse, en este caso se selecciona el 1N4148.

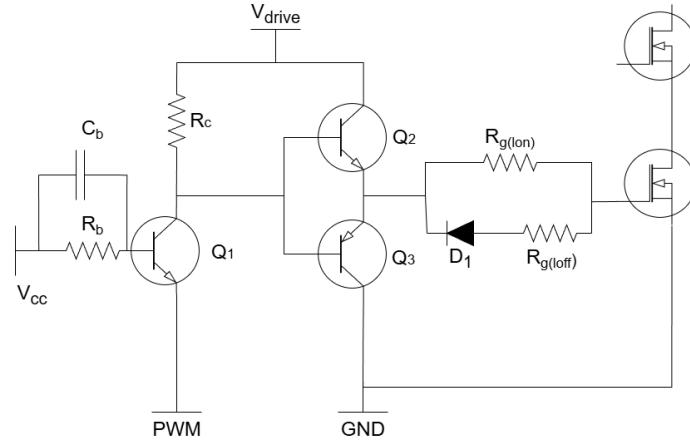


Figura 5: Controlador del lado bajo. Fuente: elaboración propia.

Para el encendido del MOSFET bajo se puede simplificar el circuito anterior al apreciado de la figura 5 en el de la figura 6. Este circuito corresponde a una configuración de colector común.

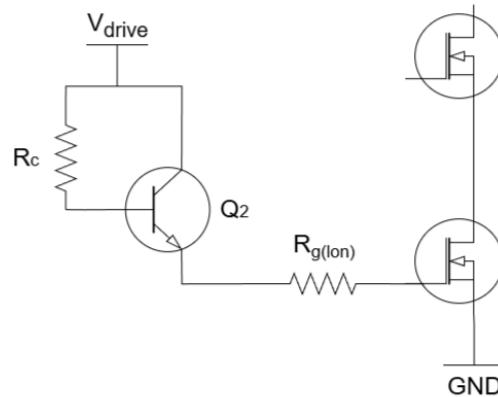


Figura 6: Controlador del lado bajo durante el encendido. Fuente: elaboración propia.

Para la etapa totem-pole se selecciona la pareja complementaria de transistores BC327 (tipo PNP) y BC337 (tipo NPN). Cuenta con una tensión máxima colector-emisor de 45 V y una corriente pico en el colector de aproximadamente 1 A, la cual es suficiente para suministrar la corriente de carga de compuerta. Se seleccionan los modelos con una ganancia (B_2, B_3) de 250 (BC327-25, BC337-25,). El voltaje entre emisor y base (V_{be2}, V_{be3}), se estima en 0.7 V.

A partir de la figura 7 se determina la ecuación 2 para la resistencia de encendido del lado bajo ($R_{g(lon)}$).

$$R_{g(lon)} = \frac{-t_{sw}}{C_{iss} \cdot \ln \left(1 - \frac{V_{plt}}{V_{drv} + V_{be3} + V_b - V_{be2}} \right) + \frac{Q_{gd}}{V_{plt} + V_{be2} - V_{drv}}} - \frac{R_c}{B_2 + 1} \quad (2)$$

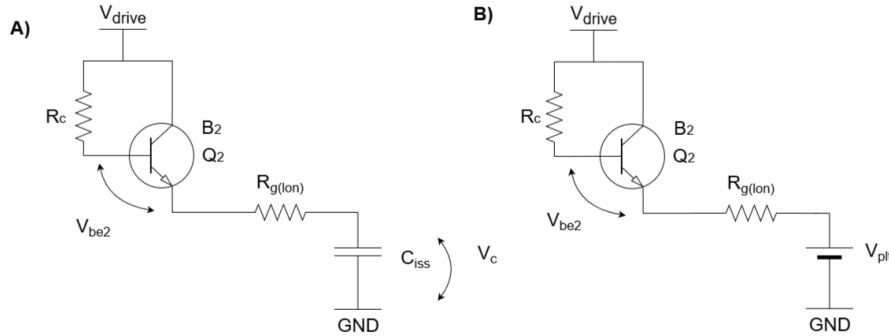


Figura 7: Circuitos simplificados para la carga del capacitor (A) y del voltaje de meseta (B). Fuente: elaboración propia.

Con esto se determina el valor de $R_{g(on)}$ como aproximadamente 55Ω . Con lo que se elige una resistencia de 47Ω , y se calcula el tiempo t_{sw} nuevamente, lo cual da aproximadamente 172 ns.

Para el apagado del MOSFET bajo se tiene el circuito de la figura 8. Este circuito corresponde a una configuración de colector común.

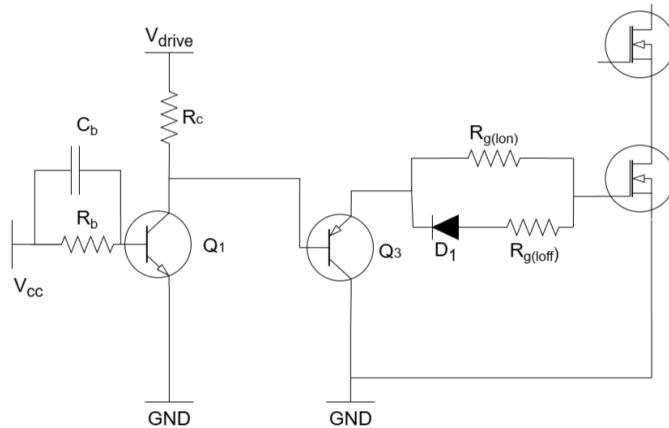


Figura 8: Controlador del lado bajo durante el apagado. Fuente: elaboración propia.

A partir de la figura del circuito simplificado de apagado 9 se determina la ecuación 3 para la resistencia de encendido ($R_{g(offeq)}$). Para resolver las ecuaciones diferenciales de estos circuitos se utilizó MATLAB, más información se puede encontrar en el apéndice o en la página de GitHub del proyecto.

$$R_{g(offeq)} = \frac{-t_{sw}}{C_{iss} \cdot \ln \left(1 - \frac{V_{plt}}{V_{drv} - V_{be2} - V_{be3} - V_b} \right) + \frac{Q_{gd}}{V_b + V_{be3} - V_{plt}}} \quad (3)$$

Con esto se determina el valor de $R_{g(offeq)}$ como aproximadamente 14Ω , a partir de este valor se determina $R_{g(off)}$ según la ecuación 4.

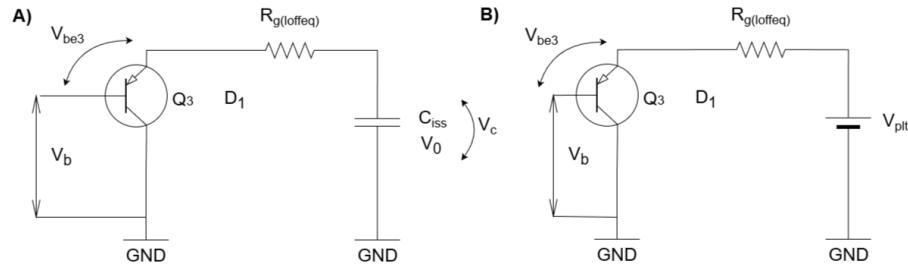


Figura 9: Circuitos simplificados para la descarga del capacitor (A) y del voltaje de meseta (B). Fuente: elaboración propia.

$$R_{g(loff)} = \frac{R_{g(loffeq)} \cdot R_{g(lon)}}{R_{g(lon)} - R_{g(loffeq)}} \quad (4)$$

El valor de $R_{g(loff)}$, por lo tanto, es de aproximadamente 19.9Ω , con lo que se elige una resistencia de 20Ω .

IV-B3. Controlador de compuerta del lado alto: En el caso de los transistores del lado alto, se emplea además un circuito en configuración bootstrap que permite elevar el potencial de la compuerta por encima del voltaje de alimentación del motor. Dado que un ciclo de trabajo del 100 % no es estrictamente necesario en este diseño, esta técnica resulta suficiente para accionar de manera adecuada los MOSFET flotantes sin necesidad de una fuente de voltaje dedicada para el control de las compuertas.

En la figura 10 se aprecia el circuito utilizado para el accionamiento del mosfet del lado alto.

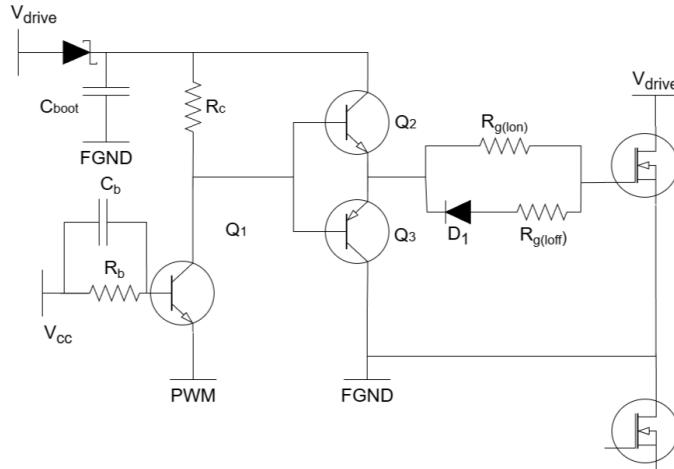


Figura 10: Controlador del lado alto. Fuente: elaboración propia.

Para el circuito de bootstrap se selecciona el diodo Schottky 1N5819 debido a su bajo voltaje en conducción (V_f) de aproximadamente 0.3 V y su rápida frecuencia de conmutación, este diodo cuenta

con una corriente máxima de 1A y un voltaje de operación de 40V, suficiente para esta aplicación. El valor de la capacitancia de bootstrap se elige de forma que se garantice que el nodo elevado mantenga su voltaje en un rango adecuado. Es importante elegir un capacitor con baja ESR (resistencia en serie equivalente), en el laboratorio el tipo de capacitor con la menor ESR que se cuenta es el cerámico multicapa.

El circuito de encendido del mosfet de lado alto se puede apreciar en la figura 11.

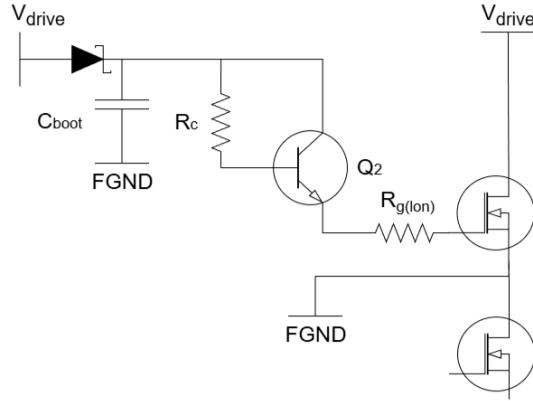


Figura 11: Controlador del lado alto durante el encendido. Fuente: elaboración propia.

A partir de la figura 12 se determina la ecuación 5 para la resistencia de encendido del lado bajo ($R_{g(hon)}$).

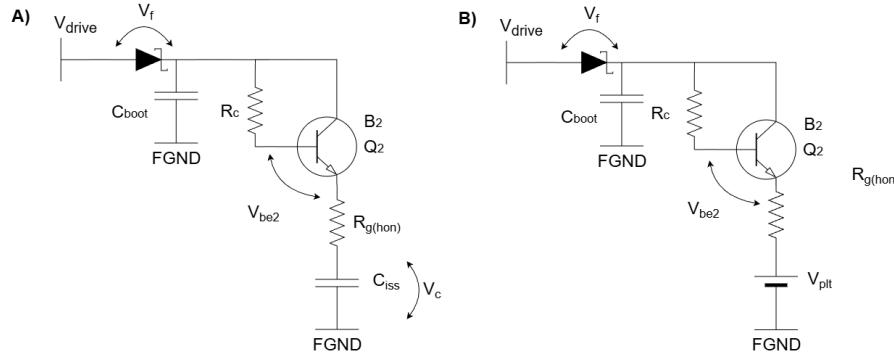


Figura 12: Circuitos simplificados para la carga del capacitor (A) y del voltaje de meseta (B). Fuente: elaboración propia.

$$R_{g(hon)} = \frac{-t_{sw}}{C_{iss} \cdot \ln \left(1 - \frac{V_{plt}}{V_{drv} + V_b + V_{be3} - V_{be2} - V_f} \right) + \frac{Q_{gd}}{V_f + V_{plt} + V_{be2} - V_{drv}}} - \frac{R_c}{B_2 + 1} \quad (5)$$

Con esto se determina el valor de $R_{g(hon)}$ como aproximadamente 55 Ω. Con lo que se elige una resistencia de 47 Ω, y se calcula el tiempo t_{sw} nuevamente, lo cual da aproximadamente 172 ns.

Para el apagado del MOSFET bajo se tiene el circuito de la figura 13.

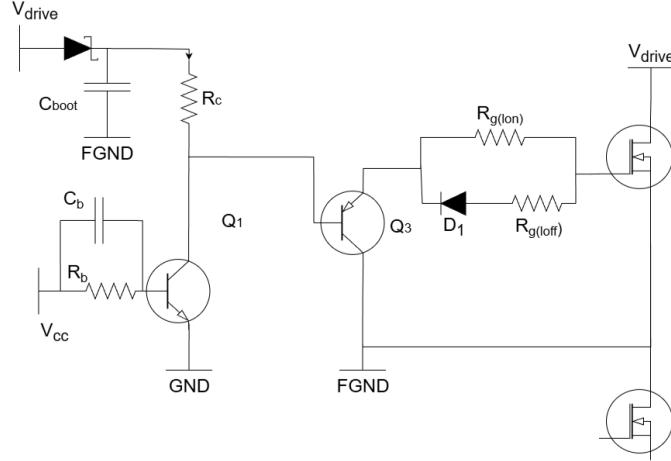


Figura 13: Controlador del lado alto durante el apagado. Fuente: elaboración propia.

A partir de la figura del circuito simplificado de apagado 14 se determina la ecuación 6 para la resistencia de encendido ($R_{g(hoffeq)}$).

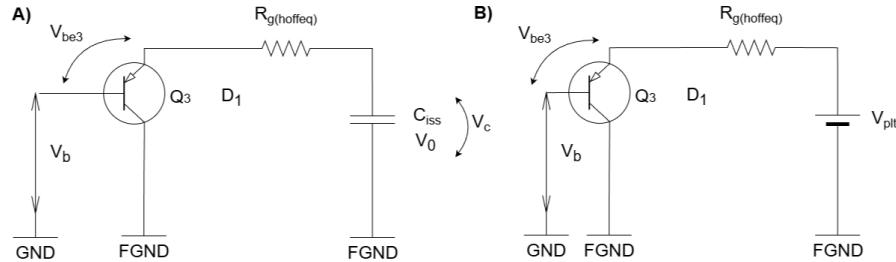


Figura 14: Circuitos simplificados para la descarga del capacitor (A) y del voltaje de meseta (B). Fuente: elaboración propia.

$$R_{g(hoffeq)} = \frac{-t_{sw}}{C_{iss} \cdot \ln \left(1 - \frac{V_{pt}}{V_{drv} - V_f - V_{be3} - V_b} \right) + \frac{Q_{gd}}{V_b + V_{be3} - V_{pt}}} \quad (6)$$

Con esto se determina el valor de $R_{g(hoffeq)}$ como aproximadamente 14 Ω, a partir de este valor se determina $R_{g(hoff)}$ según la ecuación 7.

$$R_{g(hoff)} = \frac{R_{g(hoffeq)} \cdot R_{g(hon)}}{R_{g(hon)} - R_{g(hoffeq)}} \quad (7)$$

El valor de $R_{g(hoff)}$, por lo tanto, es de aproximadamente 20 Ω, con lo que se elige una resistencia de 20 Ω.

Finalmente el circuito de medio puente se muestra en la figura 15.

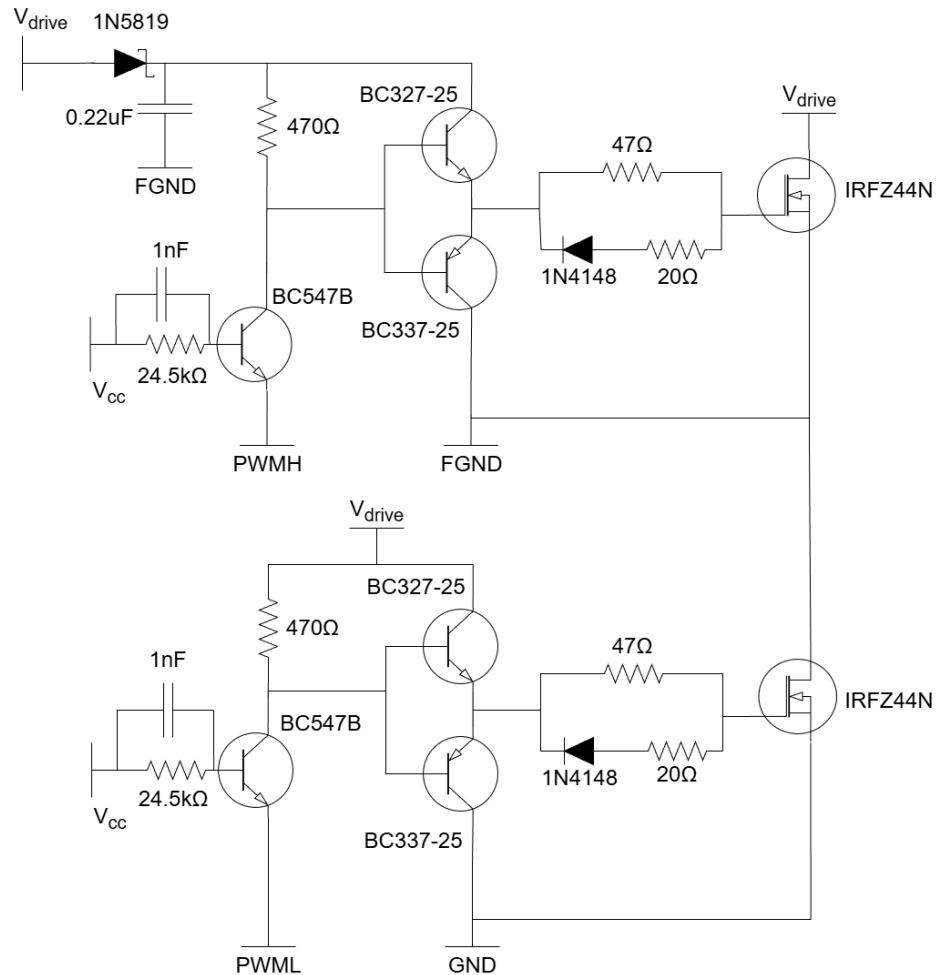


Figura 15: Circuito final de medio puente del inversor. Fuente: elaboración propia.

V. RESULTADOS

En la figura 16 se muestra el prototipo del inversor trifásico.

Como resultados se muestra las figuras 17 a 20 con los tiempos de encendido y apagado tanto para el lado bajo como el alto.

Para el lado bajo se mide un tiempo de encendido de aproximadamente 112 ns, mientras que 224 ns para el tiempo de apagado. Por otro lado, se midió 156 ns para el tiempo de encendido del lado alto y 176 ns para el tiempo de apagado.

VI. ANÁLISIS DE RESULTADOS

En el cuadro II se pueden apreciar los errores en los cálculos obtenidos.

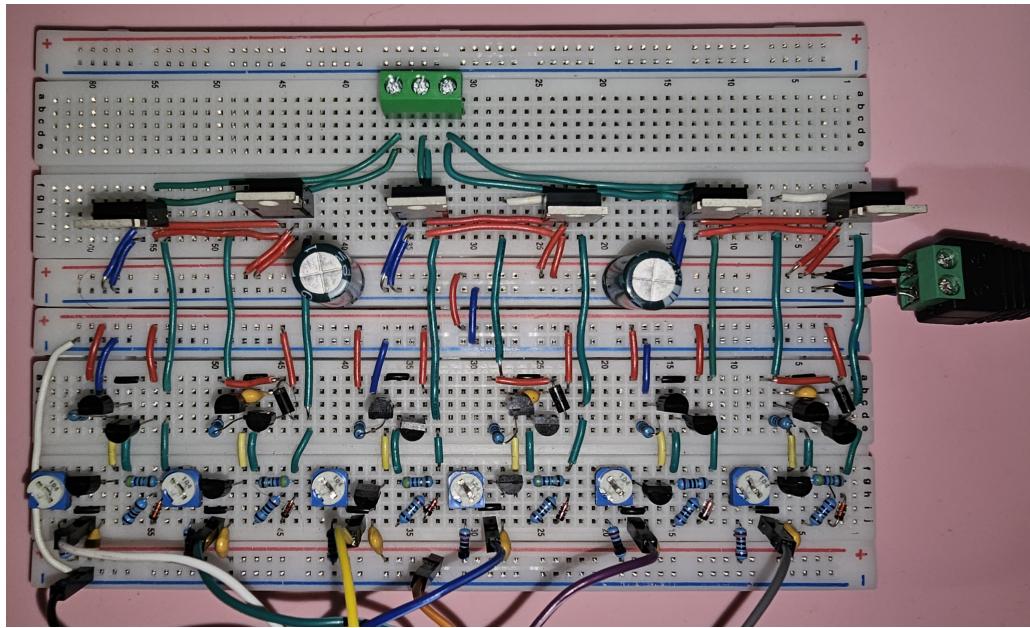


Figura 16: Prototipo del inversor trifásico. Fuente: elaboración propia.



Figura 17: Tiempo de encendido para el lado bajo (amarillo: voltaje de compuerta, verde: voltaje del pin drenaje). Fuente: elaboración propia.

En los resultados obtenidos con el osciloscopio se puede apreciar ciertas oscilaciones en la mayoría de

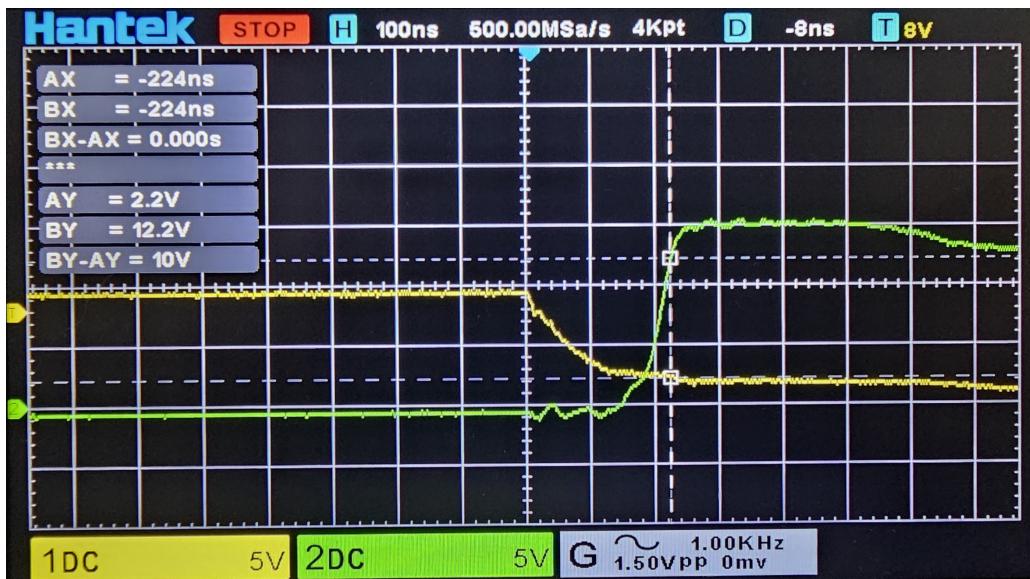


Figura 18: Tiempo de apagado para el lado bajo (amarillo: voltaje de compuerta, verde: voltaje del pin drenaje). Fuente: elaboración propia.

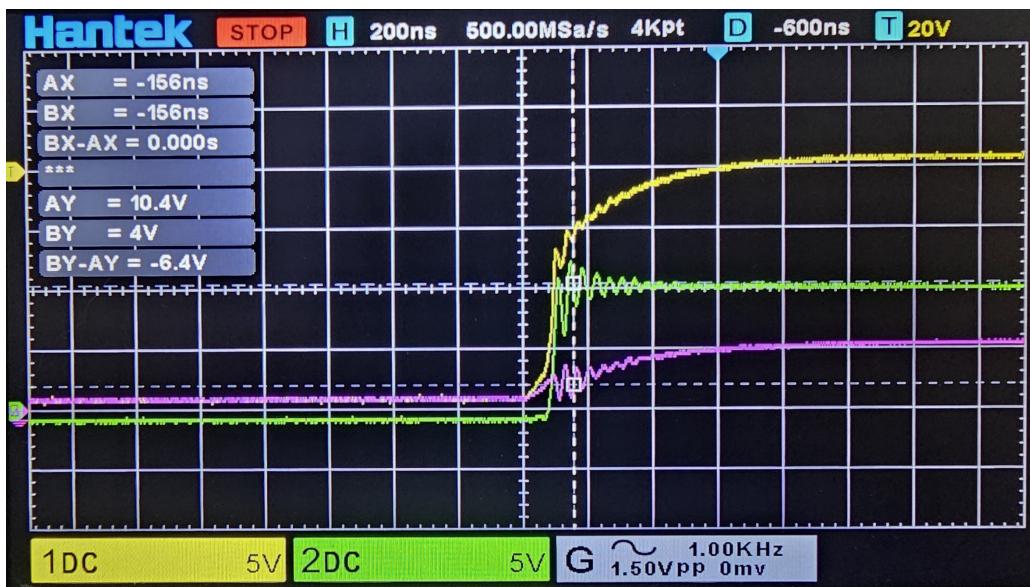


Figura 19: Tiempo de encendido para el lado alto (amarillo: voltaje de compuerta, verde: voltaje del pin fuente, morado: voltaje de compuerta - voltaje del pin fuente). Fuente: elaboración propia.

mediciones, esto podría ser debido a la fuente de voltaje commutada y puede llegar a afectar los tiempos de

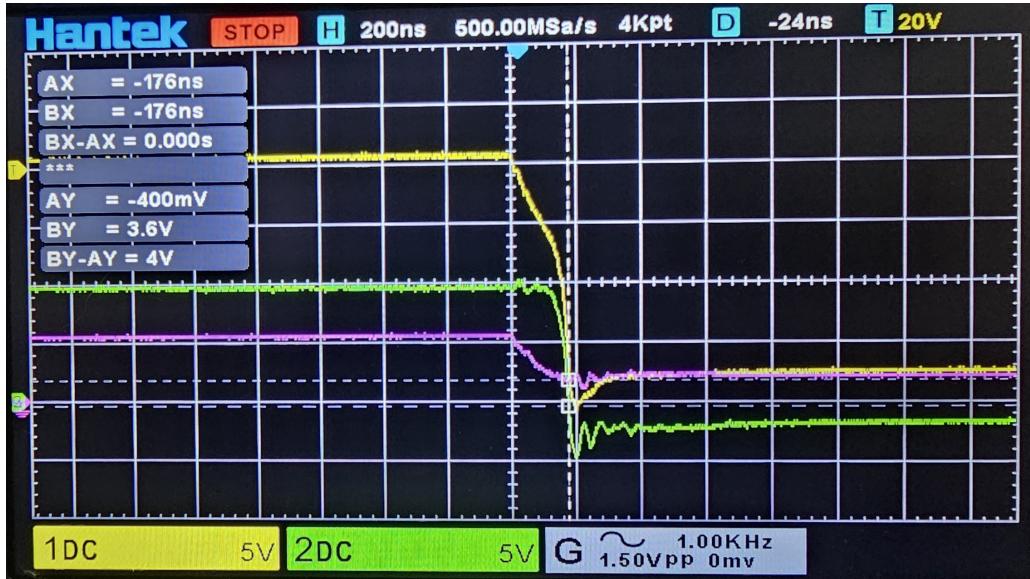


Figura 20: Tiempo de apagado para el lado alto (amarillo: voltaje de compuerta, verde: voltaje del pin fuente, morado: voltaje de compuerta - voltaje del pin fuente). Fuente: elaboración propia.

Cuadro II: Error en el tiempo calculado para los tiempos de apagado y encendido de la compuerta del mosfet.

Modo	Tiempo analítico (ns)	Tiempo experimental (ns)	Tiempo error (ns)
Encendido bajo	172	112	-60
Apagado bajo	200	224	24
Encendido alto	172	156	-16
Apagado alto	200	176	-24

carga y descarga de la compuerta. Por otro lado, el tiempo de encendido y apagado del convertidor de nivel puede afectar la actuación de la compuerta y extender su tiempo de encendido o apagado. Sin embargo, en este prototipo se intenta mantener una cantidad de componentes pequeña, y el comportamiento del convertidor de nivel se considera adecuado.

VII. CONCLUSIONES

- El prototipo permitió contrastar los tiempos teóricos de conmutación con mediciones directas sobre el hardware. Las discrepancias observadas entre los valores analíticos y experimentales se mantuvieron en un rango reducido, lo que indica que el modelo utilizado para estimar la dinámica de compuerta describe de forma razonable el comportamiento real del MOSFET dentro de la topología implementada.
- La comparación sistemática entre cálculos y mediciones evidencia que un diseño basado en un modelo analítico sencillo es suficiente para predecir el orden de magnitud de los tiempos de conmutación y para orientar la selección de componentes del driver. Esto respalda el enfoque adoptado en el proyecto, en el cual el modelado físico guía el diseño y la etapa experimental confirma su validez.

- En conjunto, el prototipo cumple con el objetivo de establecer una relación clara entre la formulación analítica y el comportamiento real del circuito, y constituye una base adecuada para versiones posteriores que requieran mayor precisión dinámica o control de pérdidas de conmutación.

VIII. RECOMENDACIONES

- El uso de una fuente de laboratorio conmutada introdujo oscilaciones en varios registros. Para reducir su influencia en los tiempos de carga y descarga de la compuerta sería preferible alimentar el prototipo con una batería o una fuente lineal. Esto permitiría obtener mediciones más limpias y disminuir la incertidumbre asociada al ruido de conmutación.
- Resultaría útil incorporar una caracterización más completa del convertidor de nivel. Aunque su desempeño fue adecuado para el prototipo, la medición explícita de sus tiempos de propagación y de su capacidad para suministrar corriente transitoria permitiría aislar con mayor claridad su efecto sobre la dinámica del MOSFET. Esta información facilitaría la comparación con alternativas de mayor velocidad si se requiere mejorar el rendimiento en versiones futuras.
- Resultaría útil un análisis más preciso del lado alto del inversor que permita caracterizar con mayor fidelidad la evolución de la tensión de compuerta respecto a su referencia flotante.

REFERENCIAS

[1]

APÉNDICE

A. Código para determinar la resistencia de encendido ($R_{g(lon)}$) del lado bajo.

Listing 1: CargaBajo.m

```

1 clear; clc; close all;
2 syms t tau ie(t) Rgloff Rglon Vplt Ci ss Vf Vdrv Vbe2 B1 B2 Rc Rb ie2 Qgd t_on
   tsw Vbe3 Vb
3
4
5 % ---- Definir ecuaci n de carga del capacitor
6
7 eqn = -(Vbe3+Vb)+(1/Ci ss)*int(ie(tau), tau, 0, t)== - ie(t)*(Rc/(B2+1)+Rglon) +
   Vdrv-Vbe2;
8
9 % ---- Pasar a ecuaci n diferencial: derivar LHS y RHS por separado ----
10
11 odeEq = diff(lhs(eqn), t) == diff(rhs(eqn), t);
12 odeEq = simplify(odeEq);
13
14 % ---- Condici n inicial i_e(0) ----
15 eqn0 = subs(eqn, t, 0);
16
17 % Crear un s mbolo para ie(0) y sustituirlo
18 syms ie0;
19 eqn0_ie0 = subs(eqn0, ie(0), ie0);
20
21 % Resolver simb licamente para ie0
22 ie0_sol = solve(eqn0_ie0, ie0);
23 ie0_sol = simplify(ie0_sol);

```

```

24 % ---- Condicion inicial para dsolve ----
25 icEq = ie(0) == ie0_sol;
26
27 % ---- Resolver la ODE ----
28 ie_sol = dsolve(odeEq, icEq);
29 ie_sol = simplify(ie_sol);
30
31 % ---- Definir el voltaje en el capacitor ----
32 ie_tau = subs(ie_sol, t, tau);
33 Vcharge = 1/Ciss * int(ie_tau, tau, 0, t);
34
35 % ---- Despejar tpo ----
36 tpo = solve(Vcharge == Vplt, t);
37
38 % ---- Definir ecuacion de carga de plateau
39 eqn = (Vdrv-Vplt-Vbe2)*t_on== Qgd*(Rc/(B2+1)+Rglon);
40
41 % ---- Despejar ton ----
42 ton = solve(eqn, t_on);
43
44 % ---- Determinar Rglon ----
45 eqn = tsw==ton+tpo;
46
47 valorRglon = solve(eqn, Rglon);
48
49 % ---- Evaluar las ecuaciones ----
50
51 % Valores utilizados
52 B2_val = 230;
53 tsw_val = 200e-9;
54 Ciss_val = 1625e-12;
55 Rc_val = 470;
56 Vplt_val = 3.6;
57 Vbe2_val = 0.7;
58 Vbe3_val = 0.7;
59 Vdrv_val = 12;
60 Qgd_val = 23e-9;
61 Vb_val = 1.2;
62
63 sol_num2 = subs(valorRglon, ...
64 [B2, tsw, Ciss, Rc, Vplt, Vbe2, Vdrv, Qgd
65 , Vb, Vbe3], ...
66 [B2_val, tsw_val, Ciss_val, Rc_val, Vplt_val, Vbe2_val, Vdrv_val Qgd_val
67 , Vb_val,Vbe3_val]);
68
69 % ---- Valor calculado para la resistencia ----
70 Rglon_val = double(sol_num2);
71
72
73
74
75
76

```

```

77 % ---- Determinar el tiempo total ----
78
79 Rglon_val = 47;
80 sol_num1 = subs(ton, ...
81     [B2, Ciss, Rc, Vplt, Vbe2, Vdrv, Qgd, Rglon,
82      Vb, Vbe3], ...
83     [B2_val, Ciss_val, Rc_val, Vplt_val, Vbe2_val, Vdrv_val Qgd_val,
84      Rglon_val, Vb_val, Vbe3_val]);
85 ton_val=double (sol_num1);
86
87 sol_num1 = subs(tpo, ...
88     [B2, Ciss, Rc, Vplt, Vbe2, Vdrv, Rglon, Vb,
89      Vbe3], ...
90     [B2_val, Ciss_val, Rc_val, Vplt_val, Vbe2_val, Vdrv_val, Rglon_val,
91      Vb_val, Vbe3_val]);
92 tpo_val=double (sol_num1);
93 t_total=ton_val+tpo_val;

```

B. Código para determinar la resistencia de apagado ($R_{g(l_{off})}$) del lado bajo.

Listing 2: DescargaBajo.m

```

1 clear; clc; close all;
2 syms t tau ie(t) Rgloff Rglon Vplt Ciss Vf Vdrv Vbe2 B1 B2 Rc Rb ie2 Qgd t_on
   tsw Vbe3 Vb
3
4
5 % ---- Definir ecuaci n de carga del capacitor
6
7 eqn = -(Vbe3+Vb)+(1/Ciss)*int(ie(tau), tau, 0, t)== - ie(t)*(Rc/(B2+1)+Rglon) +
   Vdrv-Vbe2;
8
9 % ---- Pasar a ecuaci n diferencial: derivar LHS y RHS por separado ----
10
11 odeEq = diff(lhs(eqn), t) == diff(rhs(eqn), t);
12 odeEq = simplify(odeEq);
13
14 % ---- Condici n inicial i_e(0) ----
15 eqn0 = subs(eqn, t, 0);
16
17 % Crear un smbolo para ie(0) y sustituirlo
18 syms ie0;
19 eqn0_ie0 = subs(eqn0, ie(0), ie0);
20
21 % Resolver simblicamente para ie0
22 ie0_sol = solve(eqn0_ie0, ie0);
23 ie0_sol = simplify(ie0_sol);
24
25 % ---- Condici n inicial para dsolve ----
26 icEq = ie(0) == ie0_sol;

```

```

27 % ---- Resolver la ODE ----
28 ie_sol = dsolve(odeEq, icEq);
29 ie_sol = simplify(ie_sol);
30
31 % ---- Definir el voltaje en el capacitor ----
32 ie_tau = subs(ie_sol, t, tau);
33 Vcharge = 1/Ciss * int(ie_tau, tau, 0, t);
34
35 % ---- Despejar tpo ----
36
37 tpo = solve(Vcharge == Vplt, t);
38
39 % ---- Definir ecuaci n de carga de plateau
40 eqn = (Vdrv-Vplt-Vbe2)*t_on== Qgd*(Rc/(B2+1)+Rglon);
41
42 % ---- Despejar ton ----
43
44 ton = solve(eqn, t_on);
45
46 % ---- Determinar Rglon ----
47
48 eqn = tsw==ton+tpo;
49
50 valorRglon = solve(eqn, Rglon);
51
52
53 % ---- Evaluar las ecuaciones ----
54
55 % Valores utilizados
56 B2_val = 230;
57 tsw_val = 200e-9;
58 Ciss_val = 1625e-12;
59 Rc_val = 470;
60 Vplt_val = 3.6;
61 Vbe2_val = 0.7;
62 Vbe3_val = 0.7;
63 Vdrv_val = 12;
64 Qgd_val = 23e-9;
65 Vb_val = 1.2;
66
67
68 sol_num2 = subs(valorRglon, ...
69     [B2,           tsw,          Ciss,        Rc,       Vplt,       Vbe2,       Vdrv,       Qgd
70      , Vb, Vbe3], ...
71     [B2_val, tsw_val, Ciss_val, Rc_val, Vplt_val, Vbe2_val, Vdrv_val Qgd_val
72      , Vb_val,Vbe3_val]);
73
74 % ---- Valor calculado para la resistencia ----
75 Rglon_val = double(sol_num2);
76
77
78 % ---- Determinar el tiempo total ----
79

```

```

80 Rglon_val = 47;
81 sol_num1 = subs(ton, ...
82     [B2,           Ciss,      Rc,       Vplt,       Vbe2,       Vdrv,       Qgd,       Rglon,
83      Vb,           Vbe3], ...
84     [B2_val,      Ciss_val, Rc_val, Vplt_val, Vbe2_val, Vdrv_val, Qgd_val,
85      Rglon_val, Vb_val, Vbe3_val]);
86 ton_val=double (sol_num1);
87
88 sol_num1 = subs(tpo, ...
89     [B2,           Ciss,      Rc,       Vplt,       Vbe2,       Vdrv,       Rglon,      Vb,
90      Vbe3], ...
91     [B2_val,      Ciss_val, Rc_val, Vplt_val, Vbe2_val, Vdrv_val, Rglon_val,
92      Vb_val, Vbe3_val]);
93 tpo_val=double (sol_num1);
94
95 t_total=ton_val+tpo_val;

```

C. Código para determinar la resistencia de encendido ($R_{g(hon)}$) del lado alto.

Listing 3: CargaAlto.m

```

1 clear; clc; close all;
2 syms t tau ie(t) Rgloff Rglon Vplt Ciss Vf Vdrv Vbe2 B1 B2 Rc Rb ie2 Qgd t_on
3 tsw Vbe3 Vb
4
5 % ---- Definir ecuaci n de carga del capacitor
6
7 eqn = -(Vbe3+Vb)+(1/Ciss)*int(ie(tau), tau, 0, t)== - ie(t)*(Rc/(B2+1)+Rglon) +
8     Vdrv-Vbe2;
9
10 % ---- Pasar a ecuaci n diferencial: derivar LHS y RHS por separado ----
11 odeEq = diff(lhs(eqn), t) == diff(rhs(eqn), t);
12 odeEq = simplify(odeEq);
13
14 % ---- Condici n inicial i_e(0) ----
15 eqn0 = subs(eqn, t, 0);
16
17 % Crear un smbolo para ie(0) y sustituirlo
18 syms ie0;
19 eqn0_ie0 = subs(eqn0, ie(0), ie0);
20
21 % Resolver simblicamente para ie0
22 ie0_sol = solve(eqn0_ie0, ie0);
23 ie0_sol = simplify(ie0_sol);
24
25 % ---- Condici n inicial para dsolve ----
26 icEq = ie(0) == ie0_sol;
27
28 % ---- Resolver la ODE ----
29 ie_sol = dsolve(odeEq, icEq);

```

```

30 ie_sol = simplify(ie_sol);
31 % ---- Definir el voltaje en el capacitor ----
32 ie_tau = subs(ie_sol, t, tau);
33 Vcharge = 1/Ciss * int(ie_tau, tau, 0, t);
34
35 % ---- Despejar tpo ----
36 tpo = solve(Vcharge == Vplt, t);
37
38 % ---- Definir ecuaci n de carga de plateau
39 eqn = (Vdrv-Vplt-Vbe2)*t_on== Qgd*(Rc/(B2+1)+Rglon);
40
41 % ---- Despejar ton ----
42 ton = solve(eqn, t_on);
43
44 % ---- Determinar Rglon ----
45 eqn = tsw==ton+tpo;
46
47 valorRglon = solve(eqn, Rglon);
48
49 % ---- Evaluar las ecuaciones ----
50
51 % Valores utilizados
52 B2_val = 230;
53 tsw_val = 200e-9;
54 Ciss_val = 1625e-12;
55 Rc_val = 470;
56 Vplt_val = 3.6;
57 Vbe2_val = 0.7;
58 Vbe3_val = 0.7;
59 Vdrv_val = 12;
60 Qgd_val = 23e-9;
61 Vb_val = 1.2;
62
63 sol_num2 = subs(valorRglon, ...
64 [B2, tsw, Ciss, Rc, Vplt, Vbe2, Vdrv, Qgd
65 , Vb, Vbe3], ...
66 [B2_val, tsw_val, Ciss_val, Rc_val, Vplt_val, Vbe2_val, Vdrv_val Qgd_val
67 , Vb_val,Vbe3_val]);
68
69 % ---- Valor calculado para la resistencia ----
70 Rglon_val = double(sol_num2);
71
72 % ---- Determinar el tiempo total ----
73 Rglon_val = 47;
74 sol_num1 = subs(ton, ...

```

```

82 [B2,      Ciss,      Rc,      Vplt,      Vbe2,      Vdrv,      Qgd,      Rglon,
83      Vb,      Vbe3], ...
83 [B2_val,  Ciss_val,  Rc_val,  Vplt_val,  Vbe2_val,  Vdrv_val,  Qgd_val,
83      Rglon_val, Vb_val, Vbe3_val]);
84
85 ton_val=double (sol_num1);
86
87 sol_num1 = subs(tpo, ...
88 [B2,      Ciss,      Rc,      Vplt,      Vbe2,      Vdrv,      Rglon,      Vb,
88      Vbe3], ...
89 [B2_val,  Ciss_val,  Rc_val,  Vplt_val,  Vbe2_val,  Vdrv_val,  Rglon_val,
89      Vb_val, Vbe3_val]);
90
91 tpo_val=double (sol_num1);
92
93 t_total=ton_val+tpo_val;

```

D. Código para determinar la resistencia de apagado ($R_{g(hoff)}$) del lado alto.

Listing 4: DescargaAlto.m

```

1 clear; clc; close all;
2 syms t tau ie(t) Rgloff Rglon Vplt Ciss Vf Vdrv Vbe2 B1 B2 Rc Rb ie2 Qgd t_on
2      tsw Vbe3 Vb
3
4
5 % ----- Definir ecuaci n de carga del capacitor
6
7 eqn = -(Vbe3+Vb)+(1/Ciss)*int(ie(tau), tau, 0, t)== - ie(t)*(Rc/(B2+1)+Rglon) +
7      Vdrv-Vbe2;
8
9 % ----- Pasar a ecuaci n diferencial: derivar LHS y RHS por separado -----
10
11 odeEq = diff(lhs(eqn), t) == diff(rhs(eqn), t);
12 odeEq = simplify(odeEq);
13
14 % ----- Condici n inicial i_e(0) -----
15 eqn0 = subs(eqn, t, 0);
16
17 % Crear un s mbolo para ie(0) y sustituirlo
18 syms ie0;
19 eqn0_ie0 = subs(eqn0, ie(0), ie0);
20
21 % Resolver simb licamente para ie0
22 ie0_sol = solve(eqn0_ie0, ie0);
23 ie0_sol = simplify(ie0_sol);
24
25 % ----- Condici n inicial para dsolve -----
26 icEq = ie(0) == ie0_sol;
27
28 % ----- Resolver la ODE -----
29 ie_sol = dsolve(odeEq, icEq);
30 ie_sol = simplify(ie_sol);
31

```

```

32 % ---- Definir el voltaje en el capacitor ----
33 ie_tau = subs(ie_sol, t, tau);
34 Vcharge = 1/Ciss * int(ie_tau, tau, 0, t);
35
36 % ---- Despejar tpo ----
37
38 tpo = solve(Vcharge == Vplt, t);
39
40 % ---- Definir ecuaci n de carga de plateau
41 eqn = (Vdrv-Vplt-Vbe2)*t_on== Qgd*(Rc/(B2+1)+Rglon);
42
43 % ---- Despejar ton ----
44
45 ton = solve(eqn, t_on);
46
47 % ---- Determinar Rglon ----
48
49 eqn = tsw==ton+tpo;
50
51 valorRglon = solve(eqn, Rglon);
52
53
54 % ---- Evaluar las ecuaciones ----
55
56 % Valores utilizados
57 B2_val = 230;
58 tsw_val = 200e-9;
59 Ciss_val = 1625e-12;
60 Rc_val = 470;
61 Vplt_val = 3.6;
62 Vbe2_val = 0.7;
63 Vbe3_val = 0.7;
64 Vdrv_val = 12;
65 Qgd_val = 23e-9;
66 Vb_val = 1.2;
67
68
69 sol_num2 = subs(valorRglon, ...
70 [B2,          tsw,      Ciss,      Rc,       Vplt,      Vbe2,      Vdrv,      Qgd
71 , Vb, Vbe3], ...);
72 [B2_val, tsw_val, Ciss_val, Rc_val, Vplt_val, Vbe2_val, Vdrv_val Qgd_val
73 , Vb_val,Vbe3_val]);
74
75 % ---- Valor calculado para la resistencia ----
76 Rglon_val = double(sol_num2);
77
78 % ---- Determinar el tiempo total ----
79
80 Rglon_val = 47;
81 sol_num1 = subs(ton, ...
82 [B2,          Ciss,      Rc,       Vplt,      Vbe2,      Vdrv,      Qgd,
83 Vb,      Vbe3], ...)
```

```

83     [B2_val,      Ciss_val, Rc_val, Vplt_val, Vbe2_val, Vdrv_val  Qgd_val,
84      Rglon_val, Vb_val, Vbe3_val]);
85 ton_val=double (sol_num1);
86
87 sol_num1 = subs(tpo, ...
88     [B2,          Ciss,      Rc,        Vplt,       Vbe2,       Vdrv,
89      Vbe3], ...
90     [B2_val,      Ciss_val, Rc_val, Vplt_val, Vbe2_val, Vdrv_val, Rglon_val,
91      Vb_val, Vbe3_val]);
92 tpo_val=double (sol_num1);
93 t_total=ton_val+tpo_val;

```

E. Código de arduino para el control en lazo abierto del BLDC.

Listing 5: InversorTrifasico.ino

```

1 #include <Arduino.h>
2 #include "driver/mcpwm.h"
3
4 // ===== CONFIGURACION =====
5 static const int PIN_AH = 33; // MCPWM0A
6 static const int PIN_AL = 13; // GPIO
7 static const int PIN_BH = 27; // MCPWM1A
8 static const int PIN_BL = 14; // GPIO
9 static const int PIN_CH = 25; // MCPWM2A
10 static const int PIN_CL = 26; // GPIO
11
12 static const int PIN_POT = 32; // ADC
13
14 // PWM / tiempos
15 static const uint32_t PWM_FREQ_HZ = 25000; // 10 16 kHz (IGBT) / 20 30
16   kHz (MOSFET HV)
17 static const uint16_t DEADTIME_US = 2; // 1 3 s
18 static const uint32_t ALIGN_MS = 300; // alineacion inicial
19 static const uint32_t RAMP_MS = 600; // rampa abierta
20 static const float F_START_HZ = 5.0f; // inicio
21 static const float F_RAMP_HZ = 20.0f; // final rampa
22
23 // Lmites de duty en %
24 static const float DUTY_MIN = 6.0f; // mantener bootstrap
25 static const float DUTY_MAX = 94.0f; // dejar margen superior
26
27 // ===== PROTOTIPOS =====
28 static inline void setDutyAH(float duty_percent);
29 static inline void setDutyBH(float duty_percent);
30 static inline void setDutyCH(float duty_percent);
31 static inline void offAH();
32 static inline void offBH();
33 static inline void offCH();
34 static inline void alloff();

```

```

35 static inline void applyStep(uint8_t s, float duty);
36 static inline void commutate_to(uint8_t nextStep, float duty);
37 static inline void stepNext(float duty);
38 static void mcpwm_init_three_channels();
39 static void alignAndRampOpenLoop();
40
41 // ===== ESTADO =====
42 enum { S0=0, S1, S2, S3, S4, S5 };
43 volatile uint8_t curStep = S0;
44 volatile float duty_cmd = 10.0f;
45
46 // ===== UTILIDADES GPIO =====
47 inline void setHigh(int pin){ digitalWrite(pin, HIGH); }
48 inline void setLow (int pin){ digitalWrite(pin, LOW ); }
49
50 // ===== MCPWM (AH/BH/CH) =====
51 static void mcpwm_init_three_channels() {
52     // AH en MCPWM0A
53     mcpwm_gpio_init(MCPWM_UNIT_0, MCPWM0A, PIN_AH);
54     // BH en MCPWM1A
55     mcpwm_gpio_init(MCPWM_UNIT_0, MCPWM1A, PIN_BH);
56     // CH en MCPWM2A
57     mcpwm_gpio_init(MCPWM_UNIT_0, MCPWM2A, PIN_CH);
58
59     mcpwm_config_t cfg;
60     cfg.frequency = PWM_FREQ_HZ;
61     cfg.cmpr_a = 0.0f;
62     cfg.cmpr_b = 0.0f;
63     cfg.counter_mode = MCPWM_UP_COUNTER;
64     cfg.duty_mode = MCPWM_DUTY_MODE_0;
65
66     mcpwm_init(MCPWM_UNIT_0, MCPWM_TIMER_0, &cfg); // AH
67     mcpwm_init(MCPWM_UNIT_0, MCPWM_TIMER_1, &cfg); // BH
68     mcpwm_init(MCPWM_UNIT_0, MCPWM_TIMER_2, &cfg); // CH
69 }
70
71 inline void setDutyAH(float d){
72     mcpwm_set_duty(MCPWM_UNIT_0, MCPWM_TIMER_0, MCPWM_OPR_A, d);
73     mcpwm_set_duty_type(MCPWM_UNIT_0, MCPWM_TIMER_0, MCPWM_OPR_A,
74         MCPWM_DUTY_MODE_0);
75 }
76 inline void setDutyBH(float d){
77     mcpwm_set_duty(MCPWM_UNIT_0, MCPWM_TIMER_1, MCPWM_OPR_A, d);
78     mcpwm_set_duty_type(MCPWM_UNIT_0, MCPWM_TIMER_1, MCPWM_OPR_A,
79         MCPWM_DUTY_MODE_0);
80 }
81 inline void setDutyCH(float d){
82     mcpwm_set_duty(MCPWM_UNIT_0, MCPWM_TIMER_2, MCPWM_OPR_A, d);
83     mcpwm_set_duty_type(MCPWM_UNIT_0, MCPWM_TIMER_2, MCPWM_OPR_A,
84         MCPWM_DUTY_MODE_0);
85 }
86 inline void offAH(){ setDutyAH(0.0f); }
87 inline void offBH(){ setDutyBH(0.0f); }
88 inline void offCH(){ setDutyCH(0.0f); }

```

```

87 // ===== SEIS PASOS =====
88 inline void allOff() {
89     offAH(); offBH(); offCH();
90     setLow(PIN_AL); setLow(PIN_BL); setLow(PIN_CL);
91 }
92
93 // En cada estado: XH = PWM; YL = ON; Z = flotante
94 static inline void applyStep(uint8_t s, float duty) {
95     switch (s % 6) {
96         case S0: // A+ PWM, B- ON, C flotante
97             offBH(); offCH();
98             setLow(PIN_AL); setLow(PIN_CL); // A low OFF, C low OFF (float)
99             setHigh(PIN_BL); // B low ON
100            setDutyAH(duty); // A high PWM
101            break;
102
103        case S1: // A+ PWM, C- ON, B flotante
104            offBH(); offCH();
105            setLow(PIN_AL); setLow(PIN_BL); // A low OFF, B low OFF (float)
106            setHigh(PIN_CL); // C low ON
107            setDutyAH(duty); // A high PWM
108            break;
109
110        case S2: // B+ PWM, C- ON, A flotante
111            offAH(); offCH();
112            setLow(PIN_AL); setLow(PIN_BL); // A low OFF (float), B low OFF
113            setHigh(PIN_CL); // C low ON
114            setDutyBH(duty); // B high PWM
115            break;
116
117        case S3: // B+ PWM, A- ON, C flotante
118            offAH(); offCH();
119            setLow(PIN_BL); setLow(PIN_CL); // B low OFF, C low OFF (float)
120            setHigh(PIN_AL); // A low ON
121            setDutyBH(duty); // B high PWM
122            break;
123
124        case S4: // C+ PWM, A- ON, B flotante
125            offAH(); offBH();
126            setLow(PIN_BL); setLow(PIN_CL); // B low OFF (float), C low OFF
127            setHigh(PIN_AL); // A low ON
128            setDutyCH(duty); // C high PWM
129            break;
130
131        case S5: // C+ PWM, B- ON, A flotante
132            offAH(); offBH();
133            setLow(PIN_AL); setLow(PIN_CL); // A low OFF (float), C low OFF
134            setHigh(PIN_BL); // B low ON
135            setDutyCH(duty); // C high PWM
136            break;
137    }
138 }
139
140
141 static inline void commutate_to(uint8_t nextStep, float duty) {

```

```
142     allOff();
143     delayMicroseconds(DEADTIME_US);      // DEAD-TIME duro entre ramas
144     curStep = nextStep % 6;
145     applyStep(curStep, duty);
146 }
147
148 static inline void stepNext(float duty) {
149     commutate_to(curStep + 1, duty);
150 }
151
152
153 void setup() {
154     // Low-sides
155     pinMode(PIN_AL, OUTPUT);
156     pinMode(PIN_BL, OUTPUT);
157     pinMode(PIN_CL, OUTPUT);
158     allOff();
159
160     // High-sides MCPWM
161     mcpwm_init_three_channels();
162
163     Serial.begin(115200);
164     Serial.println("BLDC_MCPWM_listo_(six-step,_dead-time_SW .");
165
166     while (!Serial) { delay(10); }
167 }
168 void loop() {
169     // Duty fijo de prueba
170     duty_cmd = 20.0f;
171
172     // Frecuencia eléctrica fija (por ejemplo 12 Hz)
173     const float f_elec = 50.0f;
174     const uint32_t t_sector_us = (uint32_t)(1e6f / (6.0f * f_elec));
175
176     // Avanza un paso (S0..S5) con dead-time entre sectores
177     stepNext(duty_cmd);
178     delayMicroseconds(t_sector_us);
179 }
180 }
```