

DESENVOLVIMENTO DE UM DRIVER E ESTUDO DE TÉCNICAS DE CONTROLE DE MOTORES BRUSHLESS DE PEQUENO PORTO

Miguel L. Rodrigues * Cláudio H. G. Santos *
Micaellen C. C. Carvalho *

* Faculdade de Engenharia Mecatrônica, CEFET, MG, (e-mails:
miguellukas52@gmail.com, claudiosantos@cefetmg.br,
mica.christie@hotmail.com).

Resumo: Este trabalho se concentra no estudo e na aplicação de um driver destinado ao controle de motores brushless de pequeno porte, com ênfase na utilização de materiais acessíveis e de baixo custo, através da técnica *Field-Oriented-Control*. Nesse sentido, foram desenvolvidas simulações computacionais utilizando o *Simulink*, juntamente da criação de uma pequena planta para realização de experimentos práticos.

1. INTRODUÇÃO

No contexto histórico, os inversores de frequência representaram uma significativa evolução na forma como os motores elétricos *BLDC* e *PMSM* eram acionados e controlados. Somente no século XX, com os avanços na eletrônica de potência e na tecnologia de controle, os inversores de frequência tornaram-se viáveis. Esses dispositivos permitiram a conversão da energia elétrica de corrente contínua para corrente alternada, proporcionando um controle muito mais preciso sobre a velocidade e o torque dos motores elétricos *BLDC* e *PMSM*.

Dentre as diversas técnicas desenvolvidas para realizar o acionamento de motores *BLDC*, em particular, o *field oriented control* (controle orientado a campo), foi uma técnica de controle avançada que se tornou possível graças aos inversores de frequência. Essa abordagem permitiu que os motores elétricos *BLDC* fossem controlados de maneira extremamente precisa, imitando o comportamento dos motores de corrente contínua (CC) em termos de resposta de torque e velocidade. O controle orientado a campo envolve o mapeamento e o controle cuidadoso do fluxo de corrente e do campo magnético dentro do motor elétrico, permitindo um controle preciso do motor. Isso trouxe melhorias significativas em eficiência energética, precisão de controle, capacidade de resposta e durabilidade em uma ampla gama de aplicações industriais, desde sistemas de transporte até robótica avançada, drones, etc.

Nesse sentido, este trabalho busca estudar um sistema de controle de um motor *BLDC*, através da técnica *Field-Oriented-Control*. Para isso, serão desenvolvidas simulações computacionais, além de uma planta física para realização de experimentos práticos. Na figura 1 é possível ver um diagrama de blocos de um motor *BLDC* controlado por *Field-Oriented-Control*.

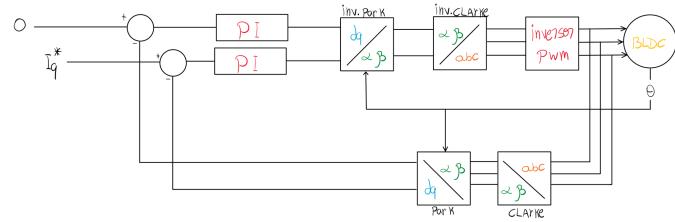


Figura 1. Diagrama Controle Orientado a Campo.

2. REFERENCIAL TEÓRICO

2.1 Modelo do Motor *BLDC*

As equações dinâmicas do motor *BLDC* podem ser obtidas de diversas formas, como por exemplo através do uso das equações diferenciais eletromecânicas, que descrevem a relação entre as variáveis elétricas e mecânicas ao longo do tempo. Além disso, de acordo com (Louis, 2011), é usual que essas equações diferenciais sejam obtidas no referencial *dq*, o que resulta em:

$$\frac{d}{dt}i_{d'} = \frac{1}{L_d}v_{d'} - \frac{R}{L_d}i_{d'} + \frac{L_q}{L_d}P\omega_m i_{q'} \quad (1)$$

$$\frac{d}{dt}i_{q'} = \frac{1}{L_q}v_{q'} - \frac{R}{L_q}i_{q'} - \frac{L_d}{L_q}P\omega_m i_{d'} - \frac{\lambda P\omega_m}{L_q} \quad (2)$$

$$T_e = 1.5P[\lambda + (L_d - L_q)i_{d'}i_{q'}] \quad (3)$$

em que L_d e L_q são as indutâncias *d* e *q*, ω_m é a velocidade angular do motor, P é o número de par de polos, $i_{d'}$ $i_{q'}$ são as correntes no referencial *dq*, $v_{d'}$ $v_{q'}$ são as tensões também no referencial *dq*, e λ é a amplitude do fluxo induzido pelos Imãs permanentes nas bobinas do estator do motor.

As equações mecânicas do sistema são dadas por:

$$\frac{d}{dt}\omega_m = \frac{1}{J}(T_e - T_f - \mu\omega_m - T_m) \quad (4)$$

$$\frac{d\theta_m}{dt} = \omega_m \quad (5)$$

sendo J o momento de inércia combinado do motor e da carga, μ o coeficiente de atrito viscoso combinado do motor e da carga, θ_m a posição angular do motor, T_m o torque no eixo do motor, T_e o torque eletromagnético, e T_f o contra-torque causado pela fricção estática.

De acordo com (Vaez-Zadech, 2018), se considerarmos que $L_q = L_d = L$, as equações (1) e (2) podem ser simplificadas para:

$$\frac{d}{dt}i_{d'} = \frac{1}{L}v_{d'} - \frac{R}{L}i_{d'} \quad (6)$$

$$\frac{d}{dt}i_{q'} = \frac{1}{L}v_{q'} - \frac{R}{L}i_{q'} - \frac{\lambda P \omega_m}{L} \quad (7)$$

Ao realizar essas transformações, é possível obter uma representação do sistema trifásico de uma forma simplificada, o que torna possível a aplicação de técnicas de controle mais avançadas. Um aspecto interessante e que deve ser ressaltado é que, com o sistema em representação dq , o circuito equivalente fica muito semelhante ao de uma máquina de corrente contínua, como pode ser visto na figura 2.

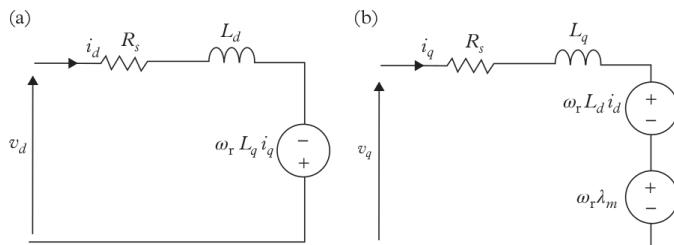


Figura 2. Circuitos equivalentes do motor bldc em malha dq

2.2 Transformações de Clarke e Park

As transformações de *Clarke* e *Park* são uma ferramenta fundamental na engenharia elétrica e eletrônica de potência, desempenhando um papel crucial na análise e no controle de sistemas elétricos trifásicos. Essa técnica matemática permite a conversão de grandezas elétricas trifásicas complexas em formatos mais acessíveis e manipuláveis, simplificando significativamente o entendimento e a implementação de estratégias de controle em uma variedade de aplicações elétricas.

A transformação de *Clarke* é utilizada para obter o valor de α e β , a partir dos valores das correntes ou tensões nos enrolamentos a , b , e c , como mostra a figura 3.

Os valores de α e β são obtidos através das projeções de a , b , e c nos eixos horizontal e vertical. Isso é feito através de relações trigonométricas, que para α resulta em:

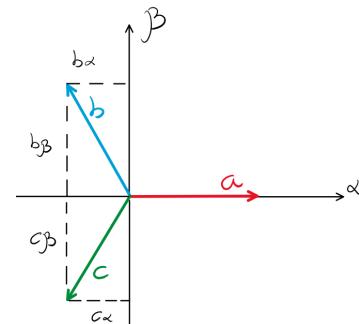


Figura 3. Sistema Ortonormal Utilizado Para Derivar a Transformação $\alpha \beta$.

$$\begin{aligned} \alpha &= a_\alpha + b_\alpha + c_\alpha \\ &= a - b \cos 60^\circ - c \cos 60^\circ \\ &= a - \frac{1}{2}b - \frac{1}{2}c \end{aligned} \quad (8)$$

De forma semelhante, β é obtido através de:

$$\begin{aligned} \beta &= a_\beta + b_\beta + c_\beta \\ &= b \cos 30^\circ - c \cos 30^\circ \\ &= \frac{\sqrt{3}}{2}b - \frac{\sqrt{3}}{2}c \end{aligned} \quad (9)$$

A transformação de *Clark* também pode ser escrita na forma matricial, resultando em:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (10)$$

Como a transformação foi obtida em um sistema ortogonal, a transformada inversa será dada pela transposta da matriz de mudança de base presente em (10), resultando em:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (11)$$

Já a transformação de *Park*, tem como propósito passar da representação $\alpha\beta$ para representação dq , e para isso é necessário conhecer a posição angular θ do rotor, como pode ser visto na figura 4.

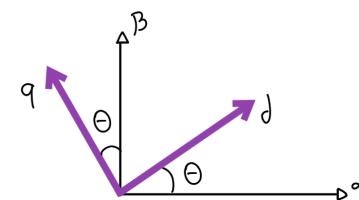


Figura 4. Eixo de Coordenadas Fixo no Rotor Utilizado para Derivar a Transformação dq .

Assim, transformação dq é obtida através da decomposição dos eixos q e d nos eixos α e β , resultando em:

$$\begin{bmatrix} q \\ d \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (12)$$

De modo análogo à transformação inversa de *Clarke*, a transformada inversa de *Park* também é dada pela transposta da matriz de mudança de base, sendo esta:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} q \\ d \end{bmatrix} \quad (13)$$

2.3 Controle em Comutação de Seis Etapas

O controle por comutação em seis etapas, ou *six-step-commutation* é o método de controle de motores BLDC mais encontrado no mercado, presente principalmente em *ESC* (Electronic-Speed-Controller) comerciais. Este método utiliza a *Back-EMF* do motor para produzir seis passos de comutação, em que cada um deles tem-se a corrente fluindo de uma bobina para outra, em que passo atual é determinado pela direção da corrente.

Para realizar a medição da *Back-EMF*, é utilizado um arranjo com resistores e comparadores, como mostra o diagrama presente na figura 5. Os resistores são usados para criar um ponto de zero virtual, e os comparadores verificam se a tensão ultrapassou ou não este ponto. Com isso, são geradas três formas de onda que entram em um processo de comparação, e por fim produzem um sinal que varia de 1 à 6, representando o passo atual de comutação. Por fim, o passo é informado ao bloco que gera os sinais de comutação do inversor de frequência, que por sua vez é conectado aos terminais do motor.

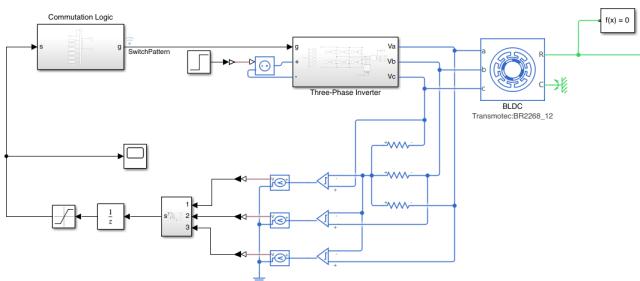


Figura 5. Circuito de Acionamento - *Six Step Commutation*.

Esse acionamento é benéfico em situações onde não se necessita de precisão e controle de posição, pois não tem-se nenhum tipo de realimentação de controle, ou seja, o motor está totalmente em malha-aberta, o que se controla é apenas a tensão que alimenta o inversor. Para casos onde se necessita maiores precisões, tanto para controle de velocidade quanto de posição, é necessário utilizar o *Field Oriented Control*.

2.4 Field Oriented Control

O controle orientado pelo campo magnético é um método altamente eficaz para o controle de motores *BLDC* e *pmsm*. Ele é amplamente utilizado em aplicações que exigem alto desempenho e precisão no controle de posição

e velocidade, como em sistemas de acionamento de máquinas, veículos elétricos, e até mesmo em aplicações de robótica.

O princípio fundamental do FOC (*Field Oriented Control*) é transformar as grandezas de controle do motor de forma a simplificar o processo de controle. Isso é feito através da transformação de *Clarke* e *Park*, que converte as grandezas trifásicas do motor (correntes e tensões) em um referencial de coordenadas fixo no rotor do motor, tornando o controle mais eficiente. Em outras palavras, o FOC enxerga o motor como se ele fosse um motor de corrente contínua (DC) virtual, o que simplifica muito o processo de controle.

Ao contrário de métodos de comutação mais simples, como o *Six Step Commutation*, o FOC tem como objetivo maximizar o torque gerado pelo motor. Para atingir esse objetivo, é essencial considerar a relação entre o campo magnético do rotor e o do estator. Quando esses campos estão perfeitamente alinhados, não há torque resultante. À medida que a projeção do campo magnético do rotor no estator diminui, o torque resultante aumenta, atingindo seu valor máximo quando não há sobreposição entre os campos magnéticos, como ilustrado na figura 6.

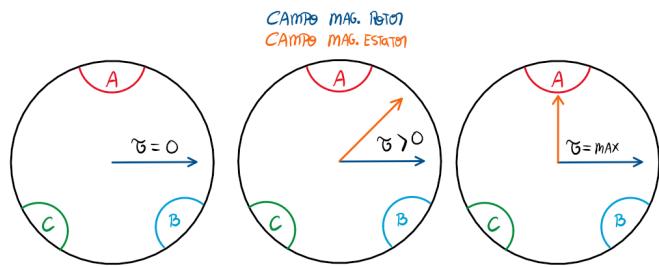


Figura 6. Torques Obtidos em Diferentes projeções.

Dado que o campo magnético do rotor em motores BLDC é estático, o objetivo principal é projetar o campo magnético do estator de maneira a controlar a interação entre ambos. Para isso, é necessário a utilização das transformações de *Clarke-Park*, juntamente com a medição da posição angular do rotor. Essas transformações permitem a criação de um campo magnético personalizado no estator, proporcionando assim o controle exato do torque.

Utilizando as transformações de *Clarke-Park*, é possível decompor o campo magnético desejado em suas componentes *dq*, conhecidas como Eixo-Direto e Eixo de Quadratura, as quais são mutuamente ortogonais. O Eixo-Direto é direcionado na mesma orientação que o campo magnético do rotor, como pode ser visto na figura 7. Para alcançar o torque máximo, a componente do campo magnético desejado no Eixo-Direto deve ser zero, garantindo que o campo magnético do estator seja perpendicular ao do rotor. Da mesma forma, se desejarmos outros valores de torque diferentes do máximo, podemos simplesmente ajustar os valores de referência *dq* correspondentes.

Os resultados de simulações e discussões à respeito do FOC serão apresentados na seção de **Simulações Computacionais**.

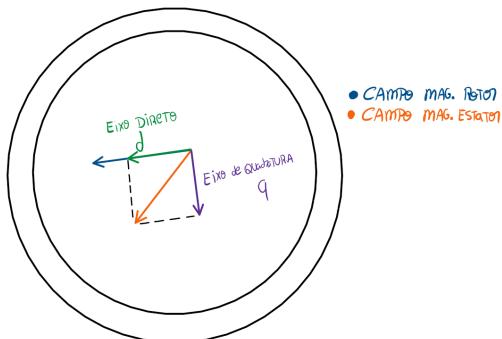


Figura 7. Decomposição dq do campo magnético do estator.

3. MATERIAIS E MÉTODOS

3.1 Plataforma STM

A (STMicroelectronics, 2023) é uma empresa franco-italiana dedicada a fabricação de componentes eletrônicos e semicondutores, sediada em Genebra, na Suíça. Entre os produtos que a ST produz, tem-se o microcontrolador SMT32 Blue Pill, ou STM32F103C8T6 (figura 8). Este microcontrolador é extremamente versátil, e atende as especificações necessárias para realizar um projeto de inversor de frequência de pequeno porte, pois é capaz de gerar sinais PWM trifásicos com a mesma onda portadora, sem a necessidade de realizar sincronizações entre os *timers*.

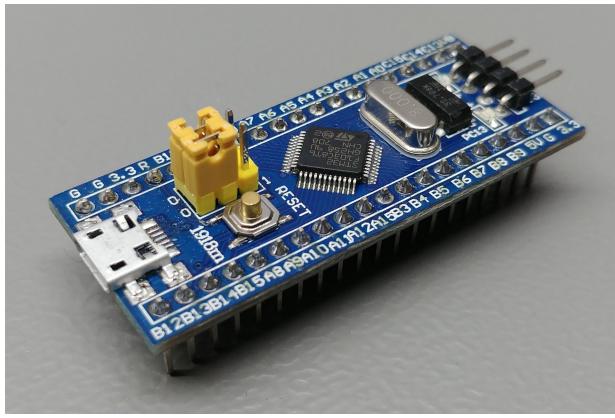


Figura 8. Microcontrolador STM32 Blue Pill.

Em geral, os microcontroladores da marca *STM* possuem uma interface própria de programação, que é chamada de *stm32cubeide*, que pode ser vista na figura 9. Nela, é possível realizar a configurações de *GPIO's*, *timers*, *clock*, acesso direto de memória, entre outras funções pré-definidas.

3.2 Inversor de Frequêcia

O inversor de frequência utilizado foi construído utilizando três ponte h do tipo L298N, controladas por um microcontrolador STM32F103C8T6. Como é necessário realizar manipulação tanto da frequência quanto da amplitude, os códigos desenvolvidos buscam simplificar a maneira com a

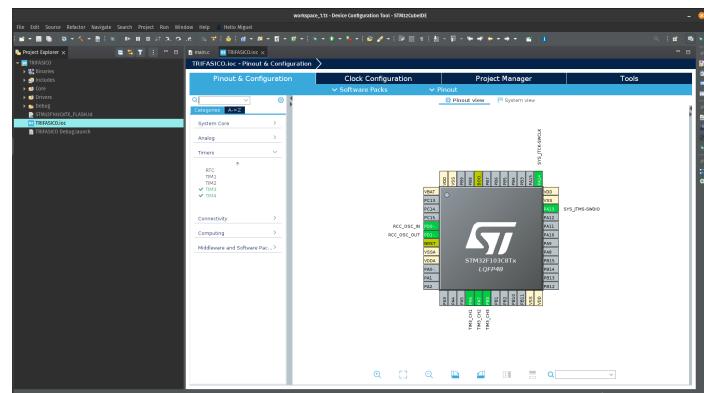


Figura 9. Interface STM32 Cube Ide.

qual essas modificações são feitas, utilizando para isso os *timers* presentes no microcontrolador.

Para gerar efetivamente os sinais PWM trifásicos que controlam a ponte H, foi utilizado um *timer* com quatro canais, dos quais foram utilizados três. Este *timer* permite a manipulação dos registradores, possibilitando a alteração do período em que cada canal permanece em nível lógico alto. Dessa maneira, é possível realizar a modulação de sinais senoidais com base em valores pré-definidos no código.

Assim, para gerar o sinal desejado, foi utilizada a estratégia de pré-calcular os vetores que contêm os pontos correspondentes às senoides para os períodos em que os canais estão em nível lógico alto. Isso é feito para economizar processamento, visto que realizar esse cálculo em tempo de execução seria desnecessário e poderia influenciar no tempo de contagem do *timer* como um todo. Dessa forma, o pré cálculo dos valores das senoides é realizado usando a seguinte expressão:

$$\begin{aligned} V_{A,i} &= W \left(1 + \sin \left(\frac{2\pi i}{N} \right) \right) \\ V_{B,i} &= W \left(1 + \sin \left(\frac{2\pi i}{N} + \frac{2\pi}{3} \right) \right) \\ V_{C,i} &= W \left(1 + \sin \left(\frac{2\pi i}{N} - \frac{2\pi}{3} \right) \right) \end{aligned} \quad (14)$$

em que i é o índice do ponto em que se deseja calcular o valor da tensão, N é o número de pontos total da senoide, e W é o valor sobre o qual a senoide excursiona, usualmente escolhido como metade do valor máximo de contagem do *timer*, que para este caso é 511.

Quanto a modulação de frequência do sinal, é necessário alterar a frequência com que os valores de amplitude são modificados. Isso pode ser feito utilizando um outro *timer*, o qual tem o tempo de estouro definido em código, através da manipulação de registradores. Ao estourar, este segundo *timer* gera uma interrupção, e é nela onde é feita a alteração dos valores de amplitude do sinal desejado, ou seja, ao alterar o tempo de estouro do segundo *timer*, estamos efetivamente alterando também a frequência da senoide que está sendo modulada. O resultado do sinal PWM obtido pode ser visto na figura 10. Nesta, também é possível ver o sinal filtrado, que resulta em uma senoide

como esperado. Existe um pequeno atraso, mas que foi introduzido apenas devido ao uso do filtro passa-baixa para que se tornasse possível a visualização do sinal senoidal. Já na figura 11 tem-se o sinal trifásico, resultante da utilização dos três canais do *timer*. É notório que o sinal gerado atende com às expectativas de projeto, visto que as senoides geradas estão de fato defasadas em 120°.

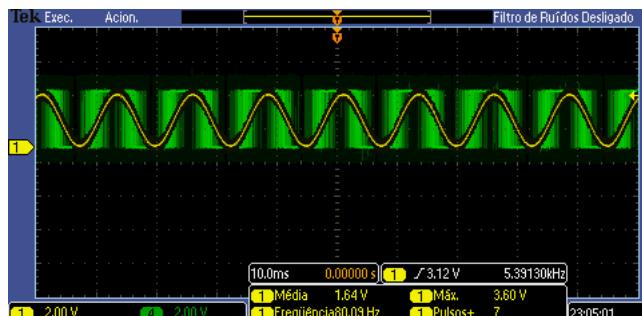


Figura 10. PWM Senoidal Com e Sem o Uso do Filtro Passa-Baixa

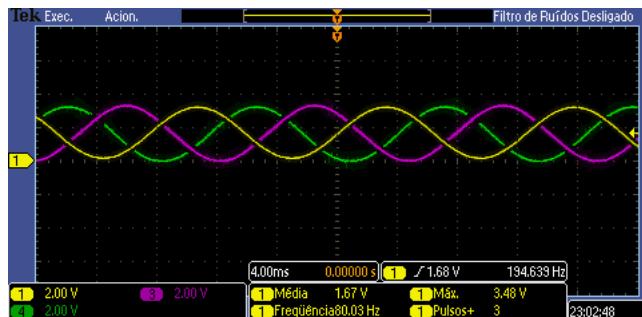


Figura 11. Sinal Senoidal Trifásico

Para realizar a configuração dos *timers*, é necessário levar em conta a frequência desejada de chaveamento, que para este caso foi escolhida como 20kHz. Assim, através da interface de programação da *STM*, foi configurado um clock de 36MHz para os timers. O primeiro *timer*, responsável por de fato realizar o chaveamento, tem 1024 contagens, e funciona no modo *Phase-Correct*, o que resulta em uma frequência de chaveamento de:

$$F_s = \frac{36000000}{2 \cdot 1024} = 17.575\text{kHz} \quad (15)$$

Já o segundo *timer* funciona no modo de contagem *Fast*, e tem no máximo 65535 contagens. Assim, é possível obter valores de frequência na faixa dos 10 Hz até 1000 Hz.

A fórmula para encontrar o número de contagens do segundo *timer* para uma determinada frequência é dada por:

$$N_\omega = \frac{36000000}{N \cdot \omega} \quad (16)$$

em que N o número de pontos da senoide, ω é a frequência em Hz, e N_ω é o número de contagens que se deseja encontrar.

3.3 Ensaio à Vazio

O ensaio em vazio desempenha um papel crucial na obtenção dos parâmetros essenciais de um motor BLDC, em conjunto com o ensaio de curto-círcuito. Para realizar esse procedimento, é necessário girar o rotor do motor a uma velocidade pré-determinada, e, simultaneamente, medir uma das tensões fase-fase. Durante esse ensaio, são obtidos parâmetros importantes, como a Tensão de Circuito Aberto e a Corrente de Magnetização.

Na figura 12, é possível observar a tensão fase-fase obtida ao girar o rotor do motor, resultando em um sinal senoidal de aproximadamente 100 [Hz].

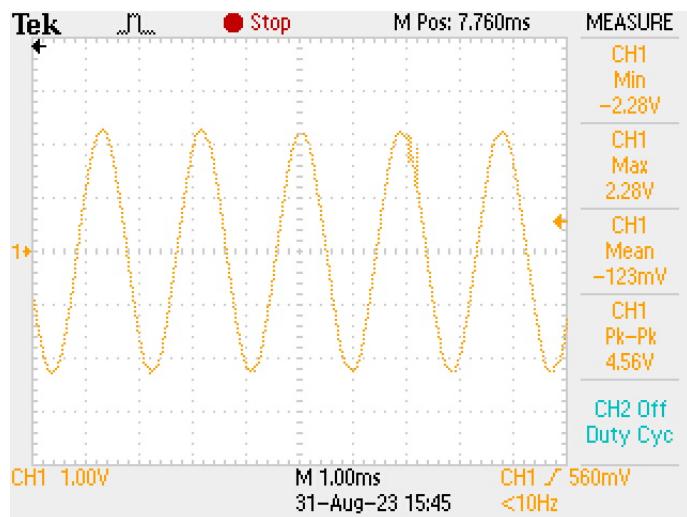


Figura 12. Resultado Ensaio à Vazio

3.4 Ensaio de Curto

O ensaio de curto-círcuito tem como objetivo a determinação dos parâmetros essenciais do motor BLDC, que incluem a resistência das fases (R_a) e a indutância das fases (L_a). Este procedimento envolve o curto-círcito de duas das fases do motor e a necessidade de sincronizar a frequência da tensão fase-fase com a obtida no ensaio à vazio. A Figura 13 ilustra a tensão fase-fase obtida ao curto-circuitar duas das fases do motor enquanto o rotor é acionado em velocidade.

Assim, obtém-se que o valor da indutância das fases L_a é de aproximadamente 845 μ H, e o valor da resistência das fases é de 20m Ω . O valor de R_a é obtido ao medir a resistência entre duas das fases do motor e dividindo o valor por dois, devido ao enrolamento estrela. Já o valor de L_a é obtido através de:

$$Z = \frac{V_{linha}}{I_{rms}} = 0.5318\Omega \quad (17)$$

$$X = \sqrt{Z^2 - R^2} = 0.5315\Omega \quad (18)$$

$$L = \frac{X\Omega}{2 \cdot \pi \cdot 100\text{Hz}} = 845.92\mu\text{H} \quad (19)$$

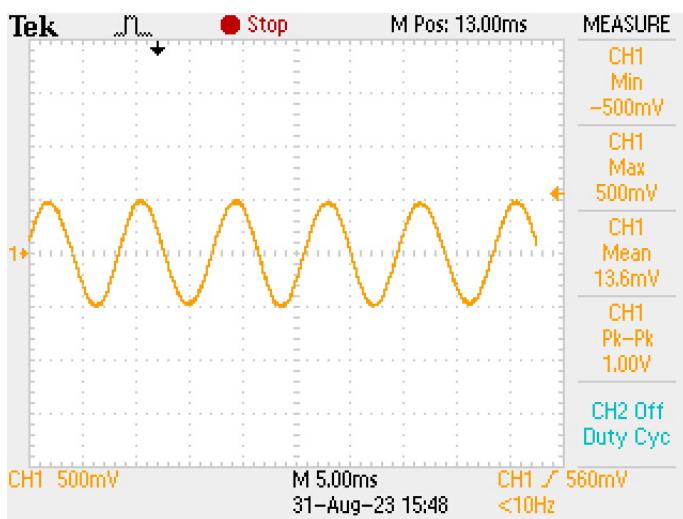


Figura 13. Resultado Ensaio de Curto

4. CONSTRUÇÃO E PROJETO DA PLANTA

Para fins práticos, foi construída nas instalações do Cefet-MG Campus Divinópolis uma planta simplificada para realizar o controle de um motor BLDC usando um inversor de frequência. Em um primeiro momento, foi feito um projeto em 3d com os principais elementos da planta, que pode ser visto na figura 14. Já a planta construída pode ser vista na figura 15.

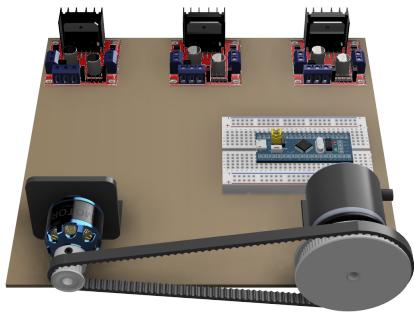


Figura 14. Projeto Inicial Planta

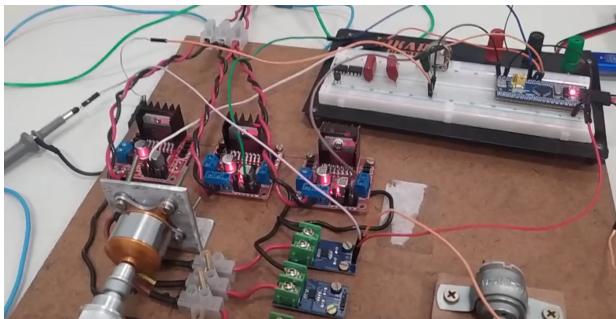


Figura 15. Planta construída

Foram utilizados 3 módulos L298N, com o intuito de dividir os esforços para reduzir a possibilidade da queima

desses componentes, visto que estes tem corrente máxima de operação próxima de 1 amper. Dessa forma, estes foram conectados de forma que cada um funciona como um braço do inversor, maximizando assim a potência que cada um pode fornecer.

O microcontrolador utilizado foi o STM32F103C8T6, que é capaz de gerar o sinal PWM trifásico necessário para realizar o acionamento do motor. Para realizar a medição de posição (fase), necessária para as transformações $dq - \alpha\beta$, foi utilizado um *encoder* incremental de 600 pulsos-por-revolução, que pode ser visto na figura 16.



Figura 16. Encoder Incremental

Para estimar a velocidade do motor, será utilizado um filtro de primeira ordem do tipo:

$$\frac{\omega(s)}{\theta(s)} = \frac{\beta s}{s + \beta} \quad (20)$$

em que β determina o quanto boa será a aproximação da velocidade.

No entanto, é importante notar que, este tipo de filtro irá amplificar ruídos existentes no sinal de entrada, por isso é necessário realizar um ajuste fino no valor de β . Para fins de implementação, (20) pode ser discretizado com um período de amostragem desejado, através da aproximação bilinear de *tustin*. Dessa forma é possível transformar o filtro em uma equação à diferenças, e então finalmente implementá-lo em código.

Assim, considerando que o período de amostragem será de 50 milissegundos, e $\beta = 8$, a discretização de (20) resulta em:

$$\frac{\omega(z)}{\theta(z)} = \frac{6.667(z - 1)}{z - 0.667} \quad (21)$$

que por sua vez pode ser escrito como a seguinte equação à diferenças:

$$\omega[k] = \frac{1}{0.667} (6.667(\theta[k] - \theta[k - 1]) + \omega[k - 1]) \quad (22)$$

Para realizar a medição das correntes do motor, foram utilizados 3 módulos sensor de corrente **ACS712**, que pode ser visto na figura 17. Esse sensor de corrente tem sensibilidade de aproximadamente $60 \frac{mv}{A}$, o que é

relativamente baixo para fins onde às correntes que se deseja medir são da ordem de 1 a 10 amperes.



Figura 17. Módulo Sensor de Corrente ACS712

Assim, como consta em (Application-Note, 2023), é necessário utilizar um circuito com amplificador operacional para aumentar a sensibilidade do sensor. O circuito em questão pode ser visto na figura 18.

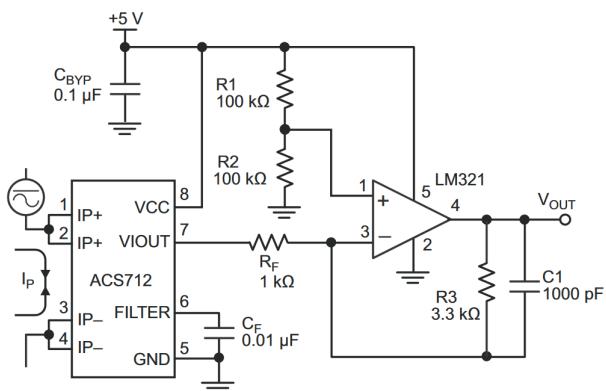


Figura 18. Circuito Amplificador para o ACS712

5. SIMULAÇÕES COMPUTACIONAIS - FOC

Para realizar as simulações computacionais, foi utilizado o software *Matlab Simulink*, que contém uma vasta biblioteca e recursos disponíveis para realizar simulações de máquinas elétricas, inversores de frequência, e circuitos de potência no geral.

Assim, foi construído o esquema de controle como na figura 1, no *Matlab Simulink*, utilizando como parâmetros do motor BLDC aqueles obtidos através dos ensaios de curto e à vazio. O projeto completo pode ser visto no final do trabalho.

5.1 Inversor de Frequência

No *Simulink* existem diversas formas de se implementar um inversor de frequência. Para este trabalho, obtou-se por utilizar o bloco *Converter (Three Phase)*, em que é possível configurar diversos aspectos, como tipo de dispositivo de chaveamento será utilizado, uso de diodos ou *snubbers*, dentre outras opções. A configuração utilizada por ser vista na figura 19.

5.2 PWM Senoidal

Para gerar os sinais PWM que realizam o acionamento do inversor de frequência, foi utilizado o bloco *PWM Generator (Three-Phase, Two-level)*, que é capaz de gerar os

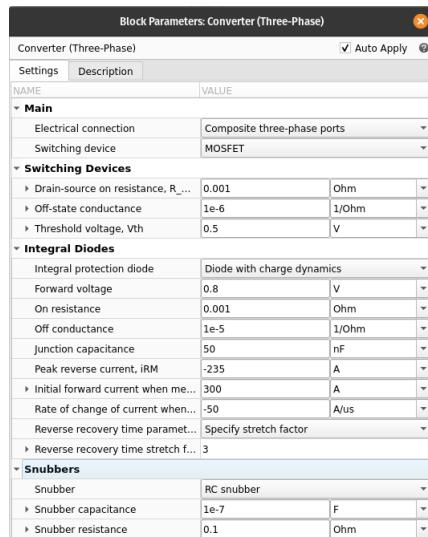


Figura 19. Configuração do Bloco do Inversor de Frequência

sinais *HIGH* e *LOW* de cada sinal PWM. Além disso, é necessário multiplexar os sinais gerados para que possam ser lidos pelo gerador de frequência. Isto é feito utilizando outro bloco, chamado *Six-Pulse Gate Multiplexer*. O esquema utilizado de geração de PWM pode ser visto na figura , sendo que V_{abc} é a tensão de referência, V_{dc} a de alimentação do inversor de frequência, e g o sinal dos pulsos.

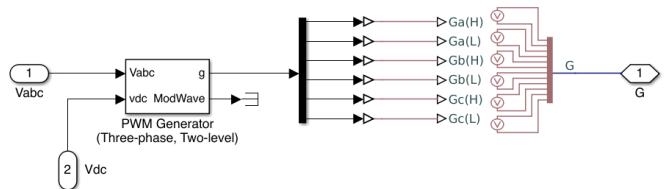


Figura 20. Circuito Gerador de PWM

5.3 Malha de Controle

A malha de controle utiliza das transformações de *Clarke-Park* para obter a corrente trifásica no referencial *dq*. Isso é feito usando blocos já existentes no *Simulink*. Após transformar as correntes, é necessário compará-las com às de referência. Para a corrente I_q , que estamos associando diretamente ao torque do motor, a referência é gerada utilizando um controlador PI, que compara a velocidade atual do motor com a velocidade desejada, e então gera uma corrente I_q de referência. Já para a corrente I_d , optamos por deixar a referência em zero, pelos motivos comentados em seções anteriores. Para realizar o controle de velocidade e de corrente, foram utilizados controladores do tipo PI, que foram sintonizados de forma a serem rápidos o suficiente para não influenciarem na dinâmica do controle de velocidade.

Como mencionado anteriormente, é ideal que se deseje que a corrente i_d tenha referência em zero, pois assim pode-se obter o máximo torque para um determinado ponto de operação. Existe também a possibilidade de que se queira realizar **enfraquecimento de campo**, que pode ser

feito ao dar *set-points* menores que zero para o valor de i_d . Usualmente, o enfraquecimento de campo se dá quando existe a necessidade de alcançar velocidades maiores que a nominal.

5.4 Resultados das Simulações

Na Figura 21, podemos observar a velocidade do motor em contraste com a velocidade de referência. Nota-se claramente que, quando a referência não é constante, há um erro persistente em relação à velocidade do motor. Isso se deve à natureza do sistema, que é do tipo 1, ou seja, possui um integrador no ramo direto, uma característica comum em motores em geral.

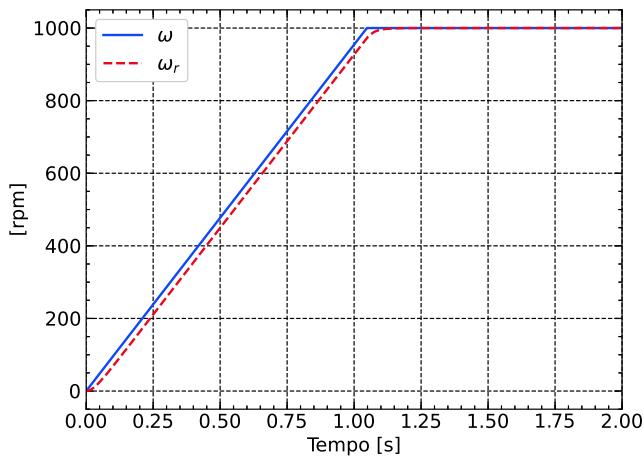


Figura 21. Evolução temporal da velocidade do motor.

Ao analisarmos a Figura 22, notamos que, quando o motor atinge a velocidade de referência constante, ocorre uma diminuição nos valores da tensão e da corrente trifásica. Isso ocorre porque, ao atingir a referência de velocidade, a aceleração não é mais necessária, apenas a manutenção da velocidade, resultando em uma redução da potência fornecida.

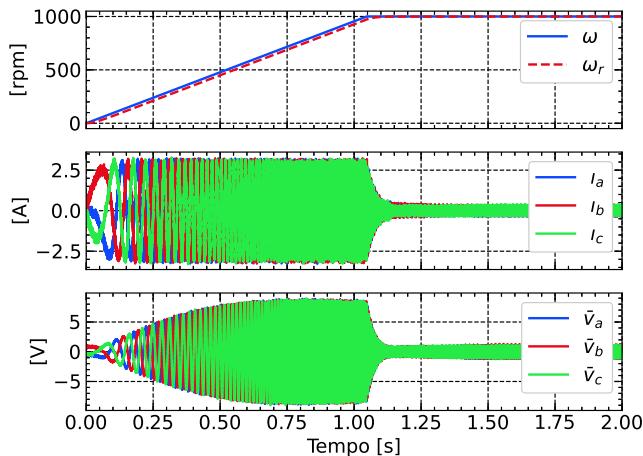


Figura 22. Tensões e correntes trifásicas em relação à velocidade do motor.

Vale destacar que a corrente I_q tende a zero quando a velocidade do motor atinge a referência, como evidenciado na Figura 23. Isso acontece porque o controlador de velocidade gera uma corrente de referência I_q quando é

necessário acelerar o motor, o que requer uma diferença de potencial dq . Consequentemente, a observação da Figura 23 mostra que, quando há a necessidade de aumentar a corrente I_q para aceleração, a tensão V_q também aumenta. No entanto, quando o motor atinge a velocidade de referência e não há mais necessidade de aceleração, a referência de I_q converge para zero, mantendo assim as tensões V_{dq} constantes.

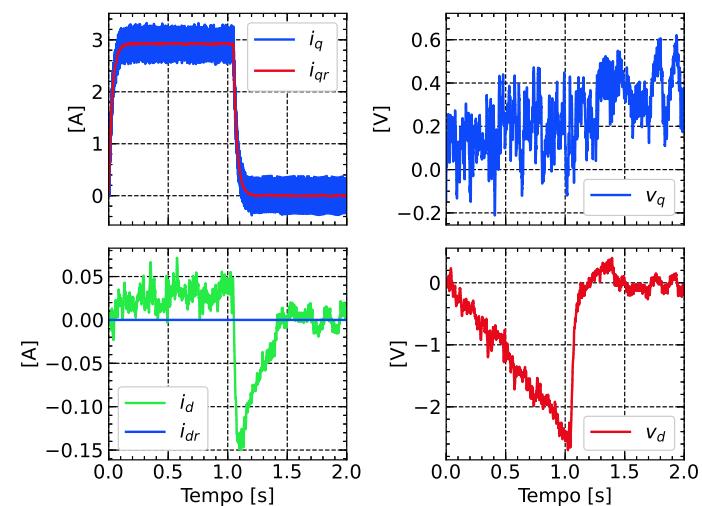


Figura 23. Correntes e Tensões dq .

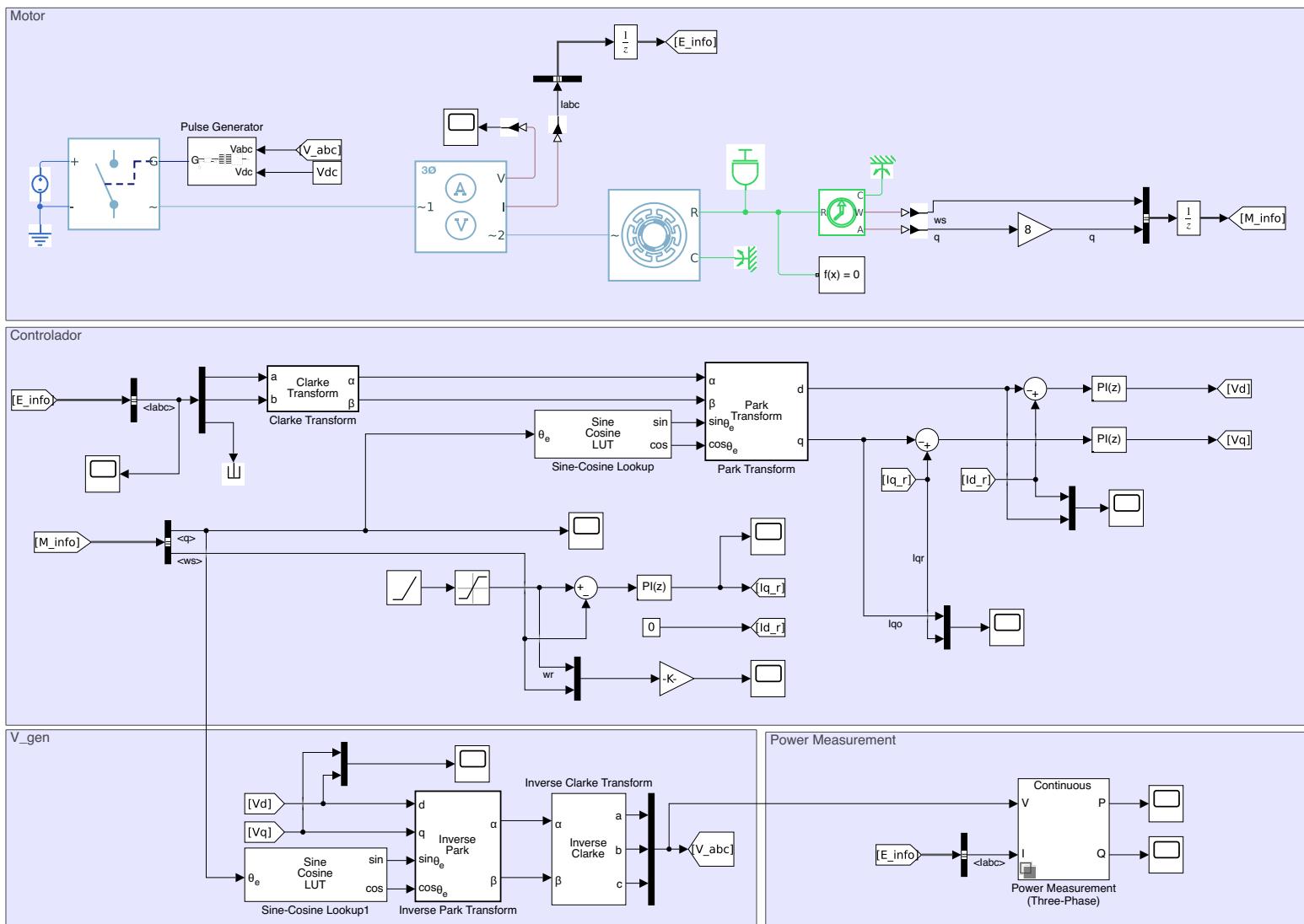
6. CONCLUSÃO

O objetivo primordial deste estudo foi aprofundar nossa compreensão dos princípios fundamentais de modelagem e controle de motores do tipo BLDC. Para alcançar esse objetivo, desenvolvemos uma planta simplificada que nos permitiu realizar experimentos práticos com um motor BLDC e um inversor de frequência, ambos de pequeno porte. Além disso, foram conduzidas simulações utilizando o software *Matlab Simulink*. Nos experimentos práticos foi possível acionar o motor em malha aberta, para velocidades em torno de 500 [RPM].

Uma das perspectivas para projetos futuros é o desenvolvimento de um sistema de controle mais robusto, utilizando por exemplo, um observador para estimar os estados do sistema para um determinado ponto de operação.

REFERÊNCIAS

- Application-Note (2023). <https://www.sparkfun.com/datasheets/b/>
- Louis, J.P. (2011). *Control of Synchronous Motors*. Wiley, first edition.
- STMicroelectronics (2023). <https://www.st.com/>
- Vaez-Zadech, S. (2018). *Control of Permanent Magnet Synchronous Motors*. Oxford, first edition.



```
#include "main.h"

TIM_HandleTypeDef htim3;
TIM_HandleTypeDef htim4;

void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_TIM3_Init(void);
static void MX_TIM4_Init(void);

#include <math.h>

#define TIM4_CLK 36000000
#define WAVE_RESOLUTION 64

uint32_t compute_tim4_period(uint16_t omega) {
    return TIM4_CLK/(WAVE_RESOLUTION*omega);
}

float compute_wave_amplitude(uint16_t omega) {
    return 0.0008860759493670887*omega + 0.0911392405063287;
}

uint16_t phaseA [WAVE_RESOLUTION];
uint16_t phaseB [WAVE_RESOLUTION];
uint16_t phaseC [WAVE_RESOLUTION];

uint8_t wave_increment = 1;
uint16_t i = 0;

float amplitude = 1.0;

int main(void)
{
    HAL_Init();

    SystemClock_Config();

    MX_GPIO_Init();
    MX_TIM3_Init();
    MX_TIM4_Init();

    for (uint16_t i = 0; i < WAVE_RESOLUTION; i++) {
        phaseA[i] = 511 * (1 + sin(2 * M_PI * i / WAVE_RESOLUTION));
        phaseB[i] = 511 * (1 + sin(2 * M_PI * i / WAVE_RESOLUTION - 2 * M_PI / 3));
        phaseC[i] = 511 * (1 + sin(2 * M_PI * i / WAVE_RESOLUTION + 2 * M_PI / 3));
    }

    HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_1);
    HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_2);
    HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_3);

    HAL_TIM_Base_Start_IT(&htim4);

    uint8_t has_start = 0;
    while (1)
    {
        if (has_start == 0) {
            for (uint16_t i = 10; i < 150; i++) {
                TIM4->ARR = compute_tim4_period(i);
                amplitude = compute_wave_amplitude(i);
                HAL_Delay(100);
            }
            has_start = 1;
        }
    }
}

void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) {
    UNUSED(htim);

    if (htim->Instance != htim4.Instance) return;

    TIM3->CCR1 = amplitude*phaseA[i];
    TIM3->CCR2 = amplitude*phaseB[i];
    TIM3->CCR3 = amplitude*phaseC[i];

    i = (i + wave_increment) % WAVE_RESOLUTION;
}
```