Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Field Oriented Control Implementation on BLDC Motor Controller with PI and SVPWM using STM32F103C8T6

To cite this article: Rofiq Cahyo Prayogo *et al* 2023 *J. Phys.: Conf. Ser.* **2622** 012025

Field Oriented Control Implementation on BLDC Motor Controller with PI and SVPWM using STM32F103C8T6

Rofiq Cahyo Prayogo[1], Aris Triwiyatno[2], Munawar Agus Riyadi[3]
[1]Dept. of Electrical Engineering, Diponegoro University, Semarang, Indonesia

Email: rofiqcp@gmail.com, aristriwiyatno@undip.ac.id, munawar@elektro.undip.ac.id

Abstract. Brushless direct current (BLDC) motor are high power density motors with a wide scope for use in electric vehicles, home automation, and industry. But the motor performance degrades due to improper control actions. This is caused by uncontrolled BLDC motor torque currents. Field Oriented Control (FOC) method is a solution for controlling BLDC motor, because it has the advantaged of overcoming torque ripples at low speeds. This paper presents the implementation of FOC on BLDC motor using STM32F103C8T6, which is high speed and low cost microcontroller. STM32F103C8T6 has been applied to handle FOC algorithm calculation and data processing. In order to validate the FOC algorithm system and assess the performance of the STM32F103C8T6 microcontroller, a 350-watt BLDC motor on an electric bicycle was tested and compared with a Universal controller. The tests that have been conducted proven that the FOC controller makes power more efficient, rotates BLDC motor at low speeds, and reduces torque ripple.

1. Introduction

Electric drives using Brushless Direct Current (BLDC) motor have become increasingly significant in recent years. The use of electric drives BLDC motor is often used in electric vehicles, industry, and home automation. BLDC motor have good durability, high efficiency, large torque, and good performance in speed and position precision [1][2]. BLDC motor is a type of motor that has the highest efficiency compared to other types of motors [3]. BLDC motor consists of two main components, namely the permanent magnet as the rotor and the coil on the stator. In principle, BLDC motor are the same as ordinary DC motors, but the main difference is that BLDC motor are operated without a brush, so electrical commutation is required to change the motor phase current in order to produce the desired torque and speed [2]. The commutation process on a BLDC motor requires detecting the position of the rotor in order to determine which coil must be supplied with energy correctly. The process of detecting the position of the rotor can use a hall effect sensor [4].

In the paper [5], the BLDC motor commutation used scalar control. It directly regulates voltage output with six-step commutation. This method is quite simple, so it is easy to implement, but it has a unique problem, namely that the torque ripple produced is quite large, especially at low speeds and uncontrolled torque. This makes the motor speed unstable, and the motor has a large enough vibration. In the paper [6], the vector control method manages the vectors in the BLDC motor, namely torque and flux. Vector control consists of Direct Torque Control (DTC) and Field Oriented Control (FOC). DTC has a very fast response to adjust the torque, but still has the disadvantage of reducing torque ripple. To overcome this problem, FOC can reduce torque ripple. This FOC control method uses torque (i_q) and flux (i_d) variable references. The i_q variable has an effect on the torque and the i_d variable has an effect on the flux. Properly torque generated provides a stable speed on the BLDC motor [7]. To get these variables properly, a Proportional Integral (PI) control system is used to control speed, torque, and flux [2]. The Space Vector Pulse Width Modulation (SVPWM) technique calculates the optimum switching pattern for the three phase inverter to ensure that the desired space vector voltage is obtained [8].

The electric drive, now powered by a numerical processor, is much cheaper than pure analog chip. In addition, the market actually provides a wide range of fast digital controllers that can be used to implement various electric drives, including BLDC motor. For all of these types of applications, in particular, Digital Signal Processors (DSP) [9], Field Programmable Gate Arrays (FPGA) [10], and microcontrollers can be used, with the choice of each influenced mainly by technical and financial factors such as cost, computational performance, flexibility, immunity to electromagnetic interference, reliability, programming languages, energy consumption, etc. In particular, microcontrollers have taken on an important role as control units in recent years due to the fact that they can be thought of as "mini computers" and are equipped with various hardware peripherals useful for automated and electrical drive applications, such as: analog to digital converter (ADC).), digital to analog converters (DAC), pulse width modulation (PWM), encoder units (for speed and position measurements), speed/high resolution counters, flash memory, communication ports, etc [11].

In this paper, the authors have used the STM32F103C8T6 microcontroller to implement the FOC algorithm to control the speed, torque, and flux of the BLDC motor. The STM32F103C8T6 is an Arm Cortex M3 microcontroller that features a high-speed 72 MHz 32-bit processor, 128 Kbytes of flash memory, a motor control library, USB, and CANBUS [14]. These microcontrollers are low-cost, in the \$5–\$8 range. The STM32 Cube Ide platform was chosen mainly because it can be easily programmed in ANSIC, a programming language widely used in the industry, and because it has a freely accessible open source integrated development environment (IDE) [11], [14-16]. The user interface on the computer uses open source software made by Vedder B, namely the VESC Tool, which is used to display the results of the BLDC motor FOC algorithm data [18].

2. Research Methods

The FOC algorithm is based on the Park-Clarke Transformation. Figure 1 shows that to control a three phase BLDC motor, we have to provide proper voltage to the motor by reading the phase currents (i_a, i_b, i_c). The FOC algorithm simplifies the control of three-phase sinusoidal current reference frames by decomposing them into torque and flux reference frames. These two components can be controlled separately [14].

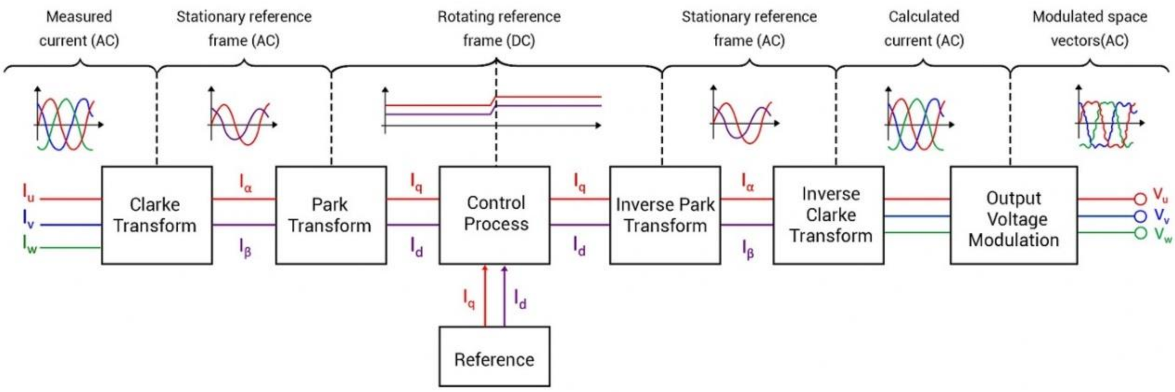

Figure 1. Signal transform of Field Oriented Control.

2.1. Field Oriented Control Algortihm

The three-phase BLDC motor current is converted by Clarke transformation into two orthogonal currents (i_α, i_β) Clarke transform can be seen from the equation (1).

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \tag{1}$$

Where i_α, i_β is orthogonal stationary frame currents and i_a, i_b, i_c is three-phase system currents. The newly converted currents are now indicated as torque generating and flux producing currents respectively. Even though the phase current has been converted into components of flux and torque, these components are still sinusoidal, making it difficult to control because they are constantly changing. With the Park transformation it will change two AC currents (i_α, i_β) into DC currents (i_q, i_d).

The Park transform changes the stationary frame of reference from the stator point of view to a rotating frame of reference from the rotor point of view, by the equation (2).

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \tag{2}$$

Where θ is angular position rotor and i_q, i_d is orthogonal rotary frame currents. Figure 2 shows that i_q is a variable for generating torque because it is perpendicular or quadrature to the stator axis, which produces the largest magnetic field to drive the BLDC motor. i_d is a variable to produce flux because it is the same direction to the stator axis.

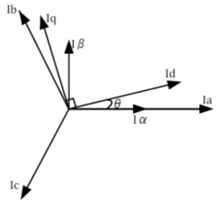

Figure 2. Combined vector representation [14].

BLDC motor get a speed reference input to controlled by a PI controller. The output of the speed control generates a value that will be entered into the torque (i_q) current reference value. To get the maximum torque, the flux (i_d) current reference value must be given a zero value. After that the i_q and i_d values will go into each PI controller. The outputs given by the two PI blocks are voltage v_q and v_d. The v_q and v_d outputs transformed into v_α and v_β values, using the inverse Park transformation. The inverse Park transformation block converts a rotating frame of reference into a stationary reference frame. Inverse Park Transformations equation can be seen in equation (3).

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} v_d \\ v_q \end{bmatrix} \tag{3}$$

Where v_q, v_d is orthogonal rotary frame voltages, and v_α, v_β is orthogonal rotary frame voltages. The v_α and v_β outputs transformed into v_a, v_b, v_c values, using the inverse Clarke transformation. The inverse Clarke transformation block converts a orthogonal stationary frame voltages into a three-phase system voltages. Inverse Clarke Transformations equation can be seen in equation (4). Then v_a, v_b, v_c is processed by SVPWM to be modulated into a 3 phase inverter.

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} \tag{4}$$

Where v_a, v_b, v_c is three-phase system voltages. Figure 3 shows a block diagram of Field Oriented Control (FOC) in which there is Park-Clark transformation, Proportional Integral (PI) controller, Space Vector Pulse Width Modulation (SVPWM).

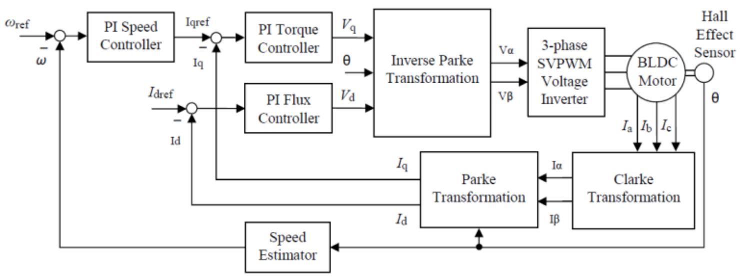

Figure 3. Block diagram of Field Oriented Control [12].

2.2. *Proportional Integral (PI) Controllers*

PI controllers with anti-wind-up units are employed in the FOC algorithm. The discrete time equation used in the implementation is given as follows equation (5).

$$x_{out(k+1)} = K_p.x_{in(k)} + K_i.T_s.x_{in(k)} + x_{out(k)} \tag{5}$$

Where $x_{out(k+1)}$ is data output PI controller, $x_{in(k)}$ is data input PI controller, K_p is proportional gain, K_i is integral gain, T_s is time sampling. The PI controller is added anti windup which is used in the FOC algorithm to control three parameter (Speed, torque, and flux). Anti windup is used to limit integral component so that they do not exceed the limit when carrying out the integration process. The discrete-time equations used in the implementation are given in equations (6).

$$x_{out(k+1)} = K_p.x_{in(k)} + w_{up}.K_i.T_s.x_{in(k)} + x_{out(k)}$$
$$w_{up} = \begin{cases} 1 & if -x_{lim} < x_{in(k)} < x_{lim} \\ 0 & otherwise \end{cases} \tag{6}$$

Where w_{up} and x_{lim} is limit of integral component. To prevent excessive output signal overshoots, symmetrical limiting saturators with limits equal to x_{lim} are employed. In order to "freeze" the PI controller output under the limitation situation, an anti wind-up operation is required. Block schematic of the PI controller shows in Figures 4.

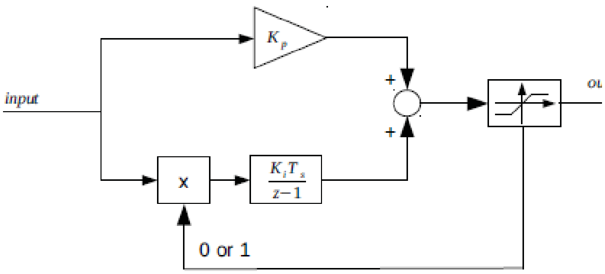

Figure 4. Block diagram of the PI controller [11].

2.3. *Space Vector Pulse Width Modulation (SVPWM)*

A simplified equation (7,8) that directly calculates the duty cycle to be applied for a 3-phase inverter has been utilized for the SVPWM [11].

$$d_i = 0.5V_{dc} + v_{ref\ i} + U^* \tag{7}$$

Where

$$U^* = -0.5\left[max\left(\textstyle\sum_i v_{ref\ i}\right) + min\left(\textstyle\sum_i v_{ref\ i}\right)\right] \tag{8}$$

Figure 3 shows SVPWM signals, phase voltages ($v_{ref\ i}$) are obtained using equation (9). Where $i = a, b,$ and c. Because the PWM counters used to create the PWM signals have a resolution of 12 bits, it is assumed that V_{dc} in this specific application equals 4096. Figure 5 shows the SWPVM Signal.

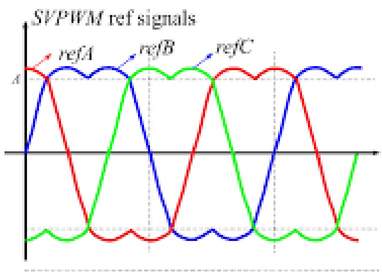

Figure 5. Space Vector Modulation signal [8].

3. Results and discussion

The FOC algorithm requires reading the phase current values on the BLDC motor. Current can be read in series by adding a small resistor, such as 0.001 ohm. Then the difference between the two resistor points is connected to the operational amplifier (OpAmp) circuit to amplify the current value. An inverter circuit is used to generate voltage on the BLDC motor and can use Metal Oxide Semiconductor Field Effect Transistor (MOSFET). To connect the microcontroller to the MOSFET, a high-speed MOSFET driver is needed, which can be used with the MT8006A or FAN 7842. The MOSFET driver will drive the high-side MOSFET and the low-side MOSFET. The MOSFET is N-channel, so on the high side of the MOSFET, a bootstrap circuit must be provided by adding a diode and capacitor. Figure 6 shows the inverter circuit and the current circuit.

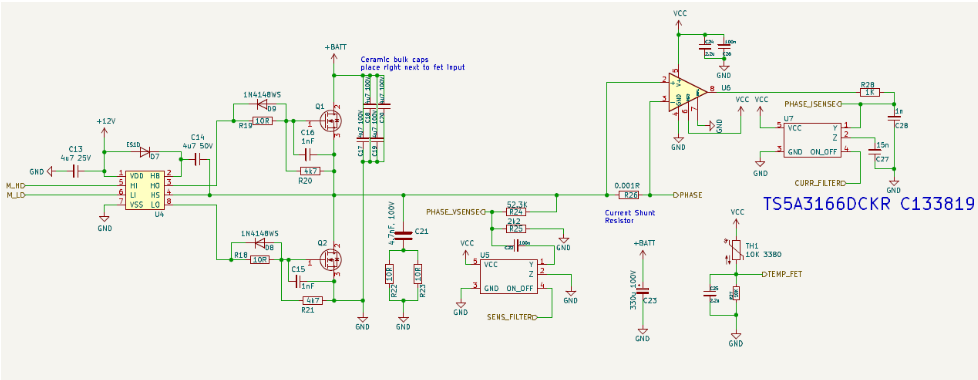

Figure 6. Scheme of the current signals and phase inverter.

The FOC control algorithm [17] was built using open-source software using the STM32 Cube IDE. All control system parameters can be changed online using the open-source user interface. Figure 7 shows the open-source user interface VESC tool [18].

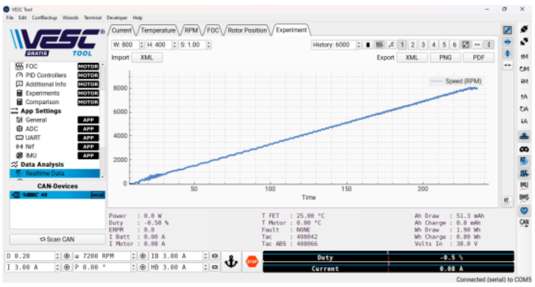

Figure 7. VESC Tool Open Source User Interface [18].

In this test, we compare the FOC controller and Universal controller tested on an electric bicycle with a 350-watt BLDC motor. The parameters that were measured and compared are speed, current, voltage, and power. Figure 8 shows the testing of two controllers on an electric bicycle.

Figure 8. Comparative test of two controllers on electric bicycle with 350 watt BLDC motor.

Tests that have been conducted by increasing the speed of the BLDC motor in the no-load state Speed is given from throttle 0 to 100% in a gradual way. Figure 9 on the left shows a graph of the FOC controller speed when the speed increases gradually. At 1.8 seconds, the BLDC motor starts rotating and reaches a maximum speed of 11200 RPM at 9.5 seconds. Figure 9 on the right shows a graph of the speed of the Universal controller when the speed increases gradually. At 3.2 seconds, the BLDC motor starts rotating at 2000 RPM and reaches a maximum speed of 11700 RPM at 11 seconds. These proved that the FOC controller can run the BLDC motor slowly and reduce torque ripple [7].

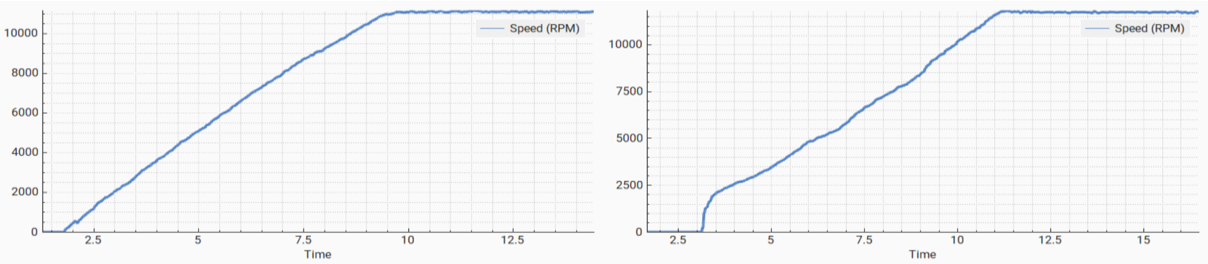

Figure 9. The speed signals of FOC controller and Universal controller.

Figure 10 on the left shows the current graph of the FOC controller when the speed increases gradually. At 1.8 to 8.7 seconds, the BLDC motor rotates with 1 amperes peak current. At 8.7 to 10 seconds, the BLDC motor has a decrease in current because it has reached the given speed. From 10 seconds onwards, the BLDC motor produces a steady current of 0.5 amperes. Figure 10 on the right shows the current graph of the Universal controller. The BLDC motor has a peak current of 3.1 amperes at 10.8 seconds. At 11.3 seconds onwards, the BLDC motor produces a steady current of 0.8 amperes. These proved that the FOC controller only requires less current than the universal controller [11].

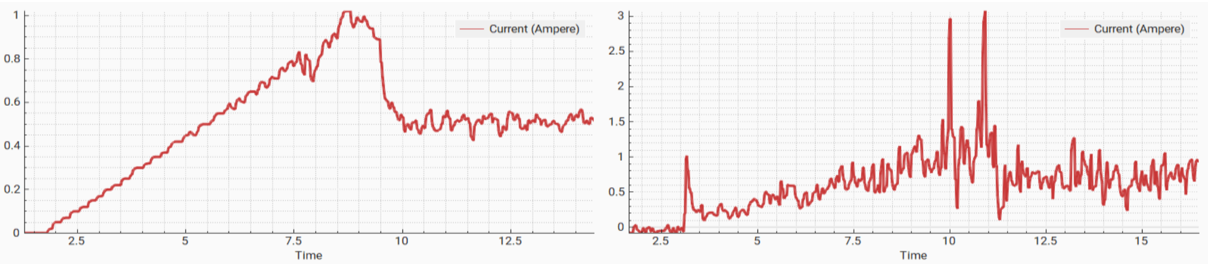

Figure 10. The current signals of FOC controller and Universal controller.

Figure 11 on the left shows the voltage graph of the FOC controller when the speed increases gradually. At 1.8 to 9.2 seconds, the BLDC motor rotates and has a peak voltage drop of 0.27 volts. At 9.5 seconds onwards, the BLDC motor rotates stably and has a steady voltage drop of 0.15 volts. Figure 11 on the right shows the voltage graph of the Universal controller. At 3.1 to 3.2 seconds, the BLDC motor rotates and has a peak voltage drop of 0.55 volts. At 4 seconds onwards, the BLDC motor rotates stably, and the steady voltage drop of 0.2 volts. Voltage drops can affect battery health, so the FOC controller is better used for longer battery life [13].

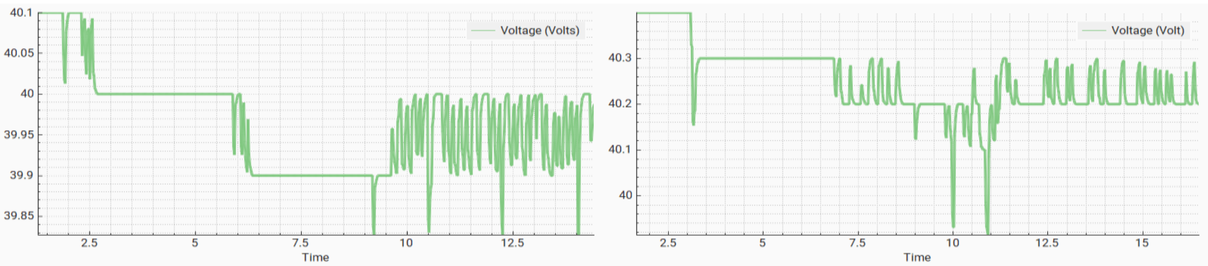

Figure 11. The voltage signals of FOC controller and Universal controller.

Figure 12 on the left shows the power graph of the FOC controller. At 1.8 to 8.7 seconds, the BLDC motor rotates with a power of 0 to 40 watts of peak power. At 10 seconds onwards, the BLDC motor has a steady power of 20 watts. Figure 12 on the right shows the power graph of the Universal controller. The BLDC motor has a peak power of 125 watts at 10.8 seconds. At 11.3 seconds onwards, the BLDC motor has a steady power of 30 watts. The result proved that the FOC controller requires less power than the Universal controller [11].

Figure 12. Power signals of FOC controller and Universal controller.

Figure 13 on the left shows that an increase in speed will affect the increase in i_q (Torque current). Figure 13 on the right shows that i_d (Flux current) is given a zero setpoint to produce maximum torque. Greater torque and speed generated will affect the i_d generated [7].

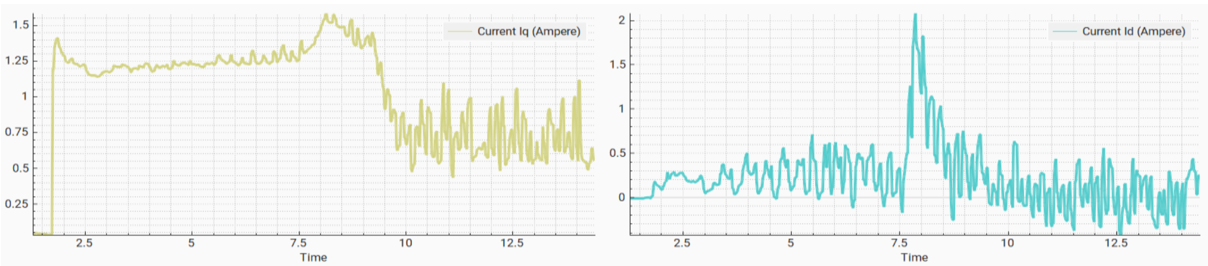

Figure 13. The i_q current and i_d current of FOC controller.

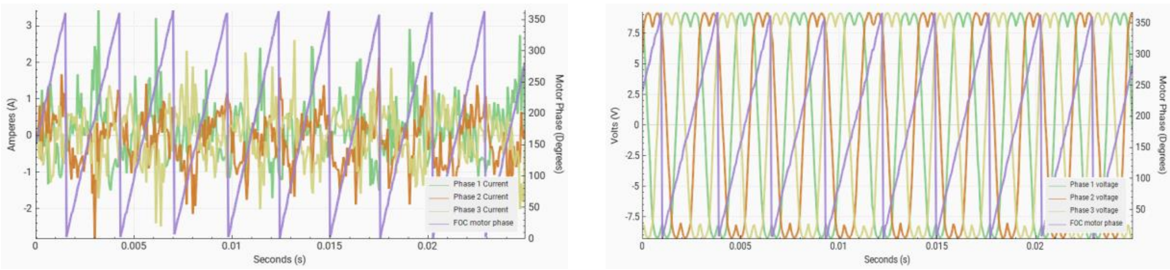

Figure 14. The 3-phase current and 3-phase voltage SVPWM of FOC controller.

Figure 14 on the left shows the 3-phase current of FOC controller at 11200 RPM that has a current of 1 amperes. Figure 14 on the right shows the 3-phase voltage SVPWM of FOC controller that has a voltage of 9.2 volts. The purple signal is the position of the BLDC motor from the Hall effect sensor.

Table 1. Comparison of two controllers at the reference speed up

No	Parameter	FOC controller	Universal controller	unit
1	Speed	11200	11700	RPM
2	Peak current	1	3.1	amperes
3	Steady current	0.5	0.8	amperes
4	Peak voltage drop	0.27	0.55	volts
5	Steady voltage drop	0.15	0.2	volts
6	Peak power	40	125	watts
7	Steady power	20	30	watts

Table 1 shows that the FOC controller is superior to the Universal controller based on the data that has been summarized into a comparison.

4. Conclusions

The FOC algorithm has been implemented on a BLDC motor using the low-cost STM32F103C8T6 microcontroller. The STM32F103C8T6 microcontroller is able to calculate all algorithms and process data. In order to validate the FOC algorithm system and assess the performance of the STM32F103C8T6 microcontroller, a 350-watt BLDC motor on an electric bicycle was tested and compared with a Universal controller. The test has been conducted that the FOC controller was superior to the Universal controller in saving power. The required current is less in a steady state or in a peak state. So that the battery used will last longer. This is due to the use of the Space Vector Pulse Width Modulation (SVPWM) method, which increases the efficiency of 3-phase voltage generation on BLDC motor. In testing, it was also proven that FOC can rotate BLDC motor at low speeds and reduce torque ripple.

References

[1] Kommula B N and Kota V R 2015 *Performance Evaluation of Hybrid Fuzzy PI Speed Controller for Brushless DC Motor for Electric Vehicle Application* Conf. PCCCTSG
[2] Madhusudan Singh and Archna Garg 2012 *Performance Evaluation of BLDC motor with Conventional PI and Fuzzy Speed Controller* Conf. IICPE
[3] Merve Yildirim, Mehmet Polat, and Hasan Kürüm 2014 *A Survey On Comparison Of Electric Motor Types And Drives Used For Electric Vehicles* Conf. IPEMCCE
[4] R Somanatham, P V N Prasad, and A D Rajkumar *2006 Modeling and Simulation of Sensorless Control of PMBLDC motor Using Zero-Crossing Back EMF Detection* Symp. SPEEDAM
[5] Protik C B, Bashudeb C G and Md Ashraful Islam 2016 *Field Oriented Control of a Current Fed PMBLDC motor and Its Comparison to Scalar Control Drive* J. AJSE **Vol 15** No 1
[6] Carey, K D, Zimmerman, N, and Ababei C 2019 *Hybrid field oriented and direct torque control for sensorless BLDC motor used in aerial drones* IET Power Electronics 12(3) 438–449
[7] Lokhande, Rohita, and Prof Priyanka Dukre 2020 *Design of FOC Brushless DC (BLDC) Motor Controller* J. IJSDR **Volume 5** Issue 9
[8] Irimia N D, Lazar F I and Luchian M 2017 *Comparative analysis of field oriented control of BLDC motor using SPWM and SVPWM techniques* Conf. RTEICT 640-645
[9] C Cavallaro, A O Di Tommaso, R Miceli, A Raciti, G Ricco Galluzzo, and M Trapanese 2003 *Analysis a dsp implementation and experimental validation of a loss minimization algorithm applied to permanent magnet synchronous motor drives* Conf. IECON **vol 1** 312–317
[10] Reddy, B P, and Murali, A 2016 *SoC FPGA-based field oriented control of BLDC motor using low resolution hall sensor* Conf. IECON 2941–2945
[11] Castiglia V, Ciotta P, Di Tommaso A, Miceli R, and Nevoloso C 2018 *High performance foc for induction motors with low cost ATSAM3X8E microcontroller* Conf. ICRERA 1495–1500
[12] Bida Vladislav M, V Dmitry. Samokhvalov, and Al-Mahturi Fuad Sh 2018 *PMSM Vector Control Techniques – a Survey* Conf. EIConRus 577-581
[13] Kumar Arjun, Hoque Mohammad A, Nurmi Petteri, Pecht Michael G., Tarkoma Sasu, and Song Junehwa 2020 *Battery Health Estimation for IoT Devices using V-edge Dynamics* HotMobile '20 Austin, TX, USA 3–4,
[14] Microsemi *Park, Inverse Park and Clarke, Inverse Clarke Transformations MSS Software Implementation*, Application Note
[15] STMicroelectronics *Medium-density performance line Arm®-based 32-bit MCU with 64 or 128 KB Flash, USB, CAN, 7 timers, 2 ADCs, 9 com. interfaces*
[16] STMicroelectronics, *Getting started with STM32 motor control SDK v54*, User manual UM2374, wwwstcom (2019)
[17] Kerrinnes N J, Koxx3, Chrisgotboost, and Foujiwara *github.com/Koxx3/ SmartESC_STM32_v2*
[18] Benjamin Vedder *github.com/vedderb/vesc_tool*