Bulletin of Electrical Engineering and Informatics
Vol. 11, No. 2, April 2022, pp. 636~645
ISSN: 2302-9285, DOI: 10.11591/eei.v11i2.3569

Implementation of FOC algorithm using FPGA for GaN-based three phase induction motor drive

Tung Duong Do[1,3], Nam Duong Le[1,2], Vu Hoang Phuong[1], Nguyen Tung Lam[1]
[1]School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam
[2]Quy Nhon University, Binh Dinh, Vietnam
[3]Viettel High Technology Industries Corporation, Viettel Group, Vietnam

Article Info

Article history:

Received Aug 2, 2021
Revised Dec 10, 2021
Accepted Feb 26, 2022

Keywords:

Field programmable gate array
GaN HEMT
High power density
Induction motor drives
Three phase inverter

ABSTRACT

Induction motor is widely used in industrial applications due to its low cost, simple design, and reliability. In this paper, the induction motor control structure FOC will be implemented on the FPGA platform for the drive system using GaN devices. By using GaN technology, the switching frequency can be up to 100 kHz instead of 2 to 20 kHz when using IGBT transistors. It leads to a significant reduction in switching loss as well as increasing the power density of the power electronic converter. The control structure will be programmed in VHDL language on the system-on-chip environment of Xilinx Zybo z7010 FPGA development board. The validity of the research is verified by some results when operating with HIL device.

Corresponding Author:

Vu Hoang Phuong
School of Electrical and Electronic Engineering, Hanoi University of Science and Technology
No. 1, Dai Co Viet Road, Hai Ba Trung, Hanoi, Vietnam
Email: phuong.vuhoang@hust.edu.vn

1. INTRODUCTION

The development of induction motor drives has a long history for decades. Many control algorithms have been developed to meet the requirements of the industry [1]. Besides, high power density has become a research trend in recent years in power electronics field. Accordingly, the new converters are designed not only to meet the power requirement but also archive a low volume and lightweight in many applications [2]. The development of semiconductor technology using GaN material allows them to operate at switching frequency from 50 to 200 kHz instead of only 2 to 20 kHz on IGBT transistors [3]. In addition, GaN-based devices also have advantages such as reducing switching losses and high temperatures operation. Many studies have been published to prove its effectiveness in practice with power electronic converters [4], [5]. However, to operating at high frequencies, it requires computational hardware must have equivalent processing capability. Previously, some studies have successfully implemented the field-oriented control algorithm using a DSP processor as shown in [6], [7]. This research focused on programming the pulse width modulation (PWM) pulse generator and current loop control. However, the higher the sampling frequency, the more computational resources are required, which leads to the time left to implement other functions, and control loops are limited [8]. FPGA technology [9] with the ability to handle multi-threading, which allows high frequency sampling and flexibility [10] is the solution for this problem. Furthermore, FPGA manufacturers also provide a system-on-chip (SoC) environment that making it increasingly used in power electronics and electrical drives systems [11]-[18]. In [19]-[21], the V/f control for induction motor using FPGA platform has been proposed. The advantage of this method is simple design, but V/f scalar control is

only suitable for low power and low control accuracy applications. The stator voltage regulator in [22] uses the sinusoidal pulse width modulation technique (SPWM) to generate gates pulse, which does not maximize the utilization of DC side voltage of three-phase voltage source inverter (VSI). This paper focuses on implementing the FOC algorithm using Kit FPGA Xilinx Zybo Zynq 7010 in VHDL language programmed by manual coding method for GaN-based induction motor drive. Figure 1 depicts the structure of the FOC algorithm that will be implemented in this work.

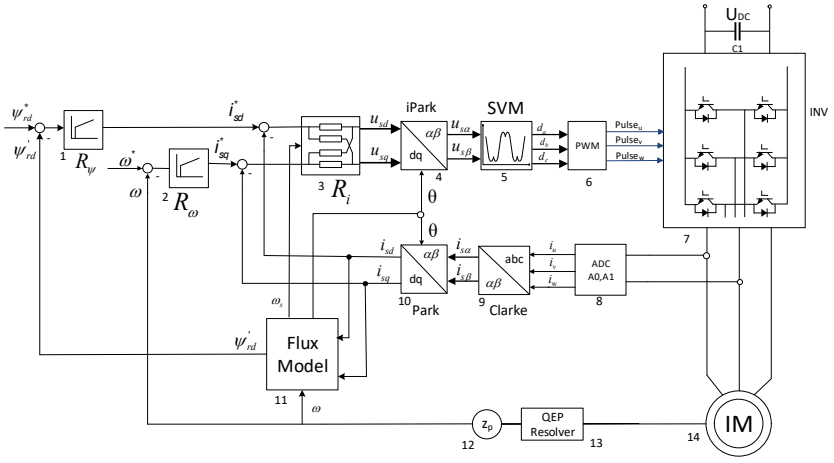


Figure 1. Block diagram of field-oriented control algorithm for induction motor drive

2. IMPLEMENTING FOC ALGORITHM ON FPGA PLATFORM

This section will present in detail the method of implementing the FOC algorithm on FPGA platform. All blocks in Figure 1 will be programmed and packaged as an integrated circuit (IC) in the development board.

2.1. FOC algorithm and modeling the induction motor

The FOC algorithm was developed decades ago by Siemens's engineers [23]. The vector control algorithm is based on the idea that stator current can be decoupled into two components by using transformation from stationary abc frame (I_{sa}, I_{sb}, I_{sc}) to rotating dq frame (I_{sd}, I_{sq}), the first one (I_{sd}) produces flux and the other (I_{sq}) produces torque. Thus, the induction motor now can be modeled as a separately excited DC motor and make DC motor control theories can be used for AC motor. Assume that the parameters of induction motor are stator resistance R_s, stator lost inductance $L_{\sigma s}$, mutual inductance L_m, rotor inductance R_r, rotor lost inductance $L_{\sigma s}$, number of pole pairs z_p, synchronous speed ω_s, rotor speed ω, rotor flux ψ_r, and T is sampling time. With L_s=$L_{\sigma s}$+L_m, L_r=$L_{\sigma r}$+L_m, ω_{sl}=ω_s-ω, T_r=L_r/T_r, T_s=L_s/T_s, $\sigma = 1 - L_m^2/(L_s L_r)$, $\psi'_{rd} = \psi_{rd}/L_m$. 4 difference equations of induction motor in dq frame can be given as [24]:

$$\begin{cases} i_{sd}(k) = \left(1 - \frac{T}{\sigma T_s} - \frac{T(1-\sigma)}{\sigma T_r}\right) i_{sd}(k-1) + \omega_s T \cdot i_{sq}(k-1) + \frac{1-\sigma}{\sigma T_r} T \cdot \psi'_{rd}(k-1) + \frac{T}{\sigma L_s} u_{sd}(k-1) & (1) \\ i_{sq}(k) = -\omega_s T \cdot i_{sd}(k-1) + \left(1 - \frac{T}{\sigma T_s} - \frac{T(1-\sigma)}{\sigma T_r}\right) i_{sq}(k-1) - T \frac{1-\sigma}{\sigma} \omega(k-1) \psi'_{rd}(k-1) + \frac{T}{\sigma L_s} u_{sq}(k-1) & (2) \\ \psi'_{rd}(k) = \frac{T}{T_r} i_{sd}(k-1) + \left(1 - \frac{T}{T_r}\right) \psi'_{rd}(k-1) & (3) \\ 0 = \frac{T}{T_r} i_{sq}(k) - [\omega_s(k) - \omega(k)] T \cdot \psi'_{rd}(k) & (4) \end{cases}$$

In (1) and (2) are used to calculate the parameters of the current controller, where (3) and (4) are used to design the flux model of induction motor.

2.2. Measurement of feedback current by XADC

XADC is an ADC module integrated into some FPGA series of Xilinx, including a dual 12-bit, 1 Mega sample per second (MSPS) ADC, and on-chip sensors [25]. Because induction motor is balance load so:

$$I_{sa}+I_{sb}+I_{sc}=0 \quad (5)$$

Therefore, only two input pairs are used to measure feedback current I_{sa} and I_{sb}, then I_{sc} will be calculated by using (5). In order to measure them, current signals must be transformed to voltage signals by:

$$U(V) = 0.01 \cdot I(A) + 0.5 \quad (6)$$

where I(A) is the instantaneous value of feedback current while U(V) is the value of voltage signal transformed. Since the ADC signal's value is in the range of 0 to 4095, permissible current measurement range is from -50 to 50 A.

2.3. Measuring the feedback rotor speed

In this work, the rotor speed is measured by using the incremental encoder and will be decoded by the QEP resolver programmed in FPGA. Specifically, let consider the states of 2 channels A and B at t_1, t_2, t_3, t_4 in Figure 2. When the motor is in the forward or backward rotate condition, the states of t_1, t_2, t_3, t_4 are also listed respectively. When all 4 states are read, the encoder has rotated 1 pulse cycle, then combine 2 adjacent states to obtain the values as shown in Table 1. By reading these combined values, both the direction of rotation and rotor speed of the induction motor will be determined.

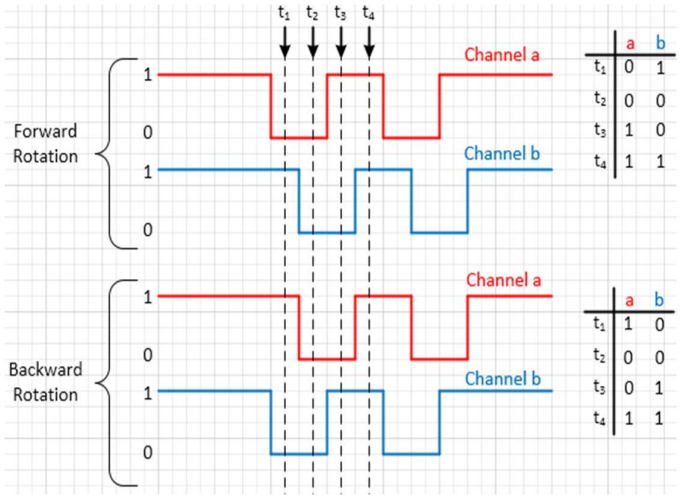


Figure 2. The logic state of 2 channels A and B

Table 1. Combination of 2 adjacent states

Combination state	Forward rotation	Backward rotation
$t_1 - t_2$	0100	1000
$t_2 - t_3$	0010	0001
$t_3 - t_4$	1011	0111
$t_4 - t_1$	1101	1110

Now let use a variable to count encoder pulses named encodercount, another variable read the state of encoder named EncoderVal. When the EncoderVal's value equal to any value in Table 1, the encodercount will be added or subtracted by 1 respectively. Otherwise, no changes are made to the variable encodercount. In this application, the clock frequency of this module is 1MHz and rotor speed will be sampled at 2000 Hz. Assume that when the counter is overflow, encodercount's value is x, number of pole pairs is z_p, the encoder has n PPR, the electrical rotor speed is calculated as:

$$\omega_e = x \cdot 2000 z_p \cdot 2\pi/(4n)(rad/s) \quad (7)$$

2.4. Frame transformation

As depicted in Figure 1, both the flux controller and the speed controller generate the reference for the current controllers in dq frame. They will be compared to the feedback current converted to the dq frame through park and clark transformation. The output of the current controller is V_{sd} and V_{sq} in dq frame will be

transformed to αβ frame to go to the SVPWM module. The transformation between them is presented as the Table 2. The frame transformation modules are programmed in the FPGA platform by using (8) to (13). The trigonometric functions sin(θ) and cos(θ) will be determined by using the look-up-table method.

Table 2. Fomula for frame transformation

Clarke transform	Park transform	Invert park transform
$\begin{cases} I_{s\alpha} = I_{sa} \cdot \frac{2}{3} - I_{sb} \cdot \frac{1}{3} - I_{sc} \frac{1}{3} \quad (8) \\ I_{s\beta} = I_{sb}/\sqrt{3} - I_{sc}/\sqrt{3} \quad (9) \end{cases}$	$\begin{cases} I_{sd} = I_{s\alpha} \cdot Cos(\theta) + I_{s\beta} \cdot Sin(\theta) \ (10) \\ I_{sq} = I_{s\beta} \cdot Cos(\theta) - I_{s\alpha} \cdot Sin(\theta) \ (11) \end{cases}$	$\begin{cases} U_{s\alpha} = U_{sd} \cdot Cos(\theta) - U_{sq} \cdot Sin(\theta) \ (12) \\ U_{s\alpha} = U_{sd} \cdot Sin(\theta) + U_{sq} \cdot Cos(\theta) \ (13) \end{cases}$

2.5. Programming flux model

Flux model the FOC algorithm is used to estimate the rotor flux, calculate the synchronous speed and the rotor angle. Firstly, rotor flux can be estimated directly from (3). The synchronous speed is calculated from (4):

$$\omega_s(k) = \omega(k) + \frac{i_{sq}(k)}{T_r \cdot \psi'_{rd}} \tag{14}$$

Then the rotor angle is the result of integrating the synchronous speed ω_s by (15):

$$\theta_{(k)} = T \cdot \omega_{s(k-1)} + \theta_{(k-1)} \tag{15}$$

The difference (3), (14), and (15) can be implemented by a digital system.

2.6. PI control algorithm

In this project, the PI controller is applied for both inner and outer control loops, this is the most control algorithm and has been demonstrated to be effective for motor control. The PI controller is described in a differential as in (16):

$$u(t) = f_P(t) + f_I(t) = K_p \cdot e(t) + K_i \int e(t)dt \tag{16}$$

Where K_p is the proportional gain, K_i is the integral time constant, e(t) is the steady-state error, and u(t) is the control signal of the controller. With the sample interval T, (16) also can become a difference equation by discretization as follows these steps:

$$\text{With the proportional component: } f_P(t) = K_p \cdot e(t) \xrightarrow{discretization} f_P(k) = K_p \cdot e(k) \tag{17}$$

The discretization of integral component is implemented by following the Riemann Sums with the sample interval T.

$$f_I(k) = \frac{K_i}{T} \cdot \sum_{n=0}^{k} e(n) = \frac{K_i}{T} \cdot e(k) + \frac{K_i}{T} \cdot \sum_{n=0}^{k-1} e(n) = \frac{K_i}{T} \cdot e(k) + f_I(k-1) \tag{18}$$

Finally, the difference equation of PI controller is calculated as:

$$u(k) = f_I(k-1) + K_p e(k) + \frac{K_i}{T} \cdot e(k) \tag{19}$$

From (19) can be applied for digital PI controller on any digital system such as FPGA, because PI controller is designed in the time domain, so the discretization can reduce the control quality of the controller. To avoid this phenomenon, the higher the sampling frequency, the more similar the characteristics of the controller designed in the digital domain to its characteristics in the time domain. FPGA platform is a powerful tool to achieve high control quality.

2.7. SVPWM technique

Space vector modulation (SVM) is a technique used for PWM, it is commonly applied for VSI. The biggest advantage of this technique is that it can utilize DC side voltage (≈0.577 U_{dc}) better than the SinPWM technique. In this research, it will be programmed to generate the gate pulse at frequency 100kHz for 2-level VSI using GaN devices. The SVPWM technique has been proposed in detail in [8]. Figure 3 depicts the SVPWM subsystem programmed in FPGA. Figure 4 depicts the flowcharts of the important blocks in this work to clarify the programming method, which presented in the previous sections.

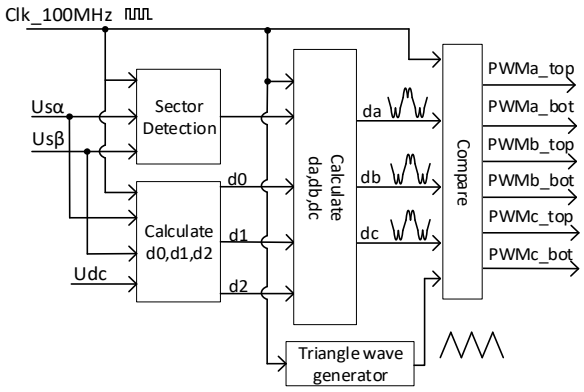


Figure 3. Block diagram of SVPWM technique in FPGA

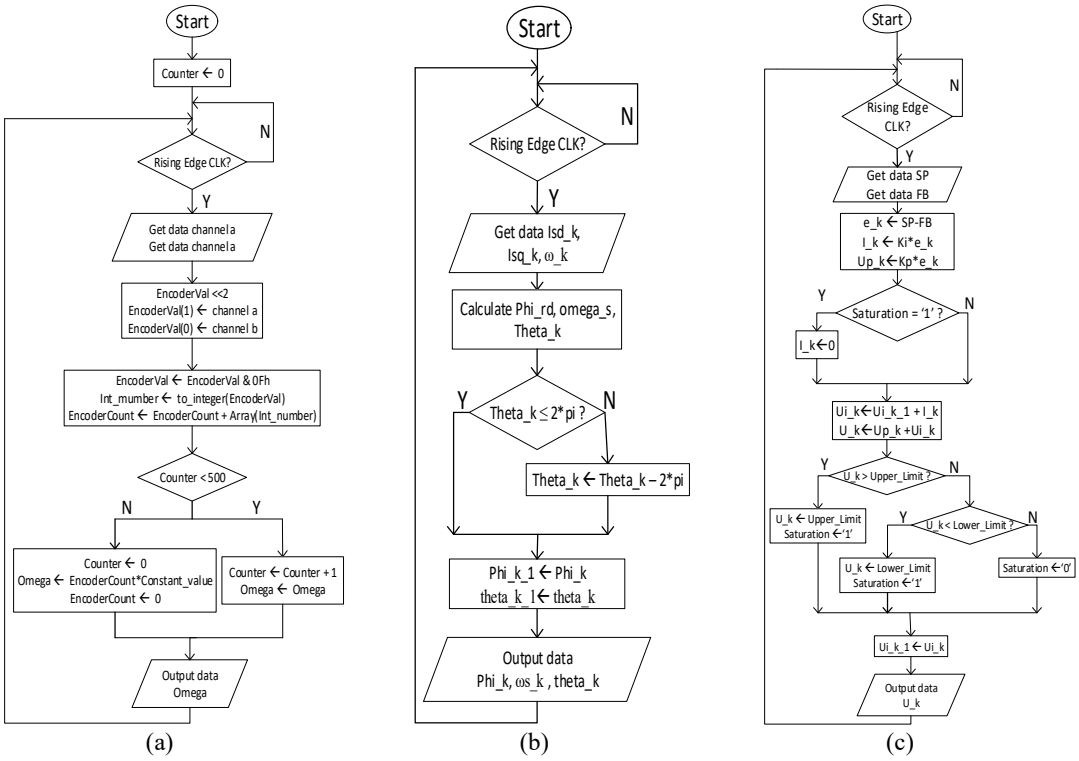


Figure 4. Flow charts of, (a) QEP resolver, (b) flux model, and (c) PI controller

3. RESULTS AND DISCUSSION

In this section, several of the results when launching the FOC algorithm implemented in the FPGA platform with Typhoon HIL 402 devices to verify the validity of this research and make some discussion. System clock of the design is 100MHz. The connecting diagram in detail of FPGA and Typhoon HIL is shown in Figure 5. The parameter is presented in Table 3 and Table 4.

Table 3. Parameters of induction motor

R_s	$L_{\sigma s}$	L_m	R_r	$L_{\sigma r}$	z_p	J	Pole pairs	Ψ_{rd}^*
2.52 Ω	0.0062 H	0.1763 H	0.97 Ω	0.0095 H	2	0.117 Kg.m/s^2	2	1.2341Wb

Table 4. Parameters of (a) VSI circuit and (b) controllers

(a)

Parameters	Value
U_{dc}:	700 V
C_1:	1000 uF
f_{sw}:	100 kHz

(b)

Controller	K_p	K_i	Limit
Speed controller	5	178	± 22
Current controller	30	6690	± 310
Flux controller	38	1976	± 18

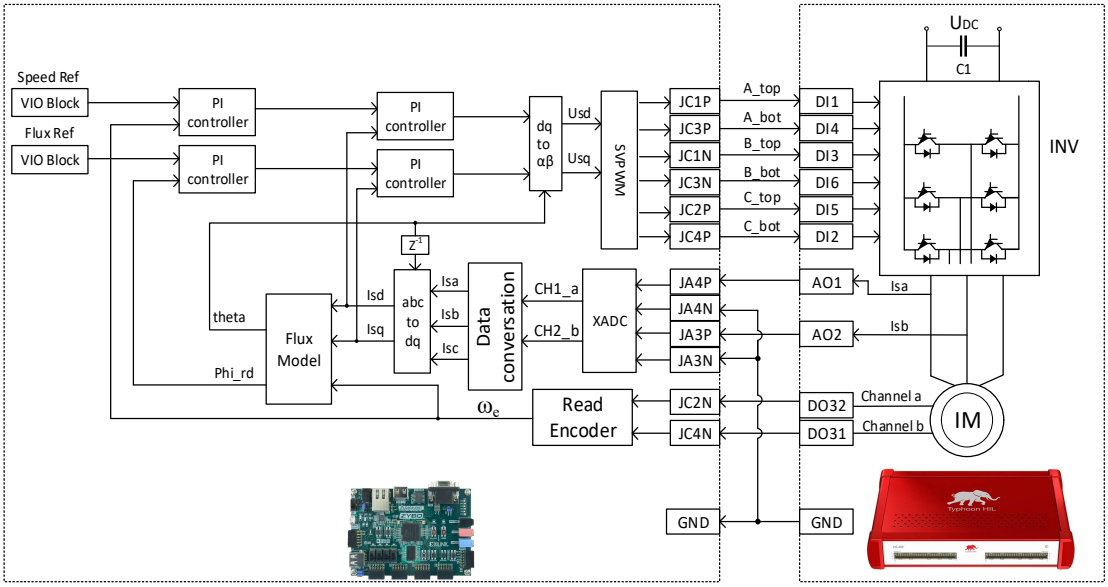


Figure 5. Connecting FPGA and typhoon HIL

3.1. Device utilization and processing time

Here, the device utilization of the FPGA board is counted in Table 5 giving an overview of the optimization of design. The processing time also calculated by setting a flag at the beginning and the end of the computation cycle, Figure 6 shown the processing time of FPGA in one sample. The processing time is 0.22µs, it significantly smaller than the sample time and allows the duty cycle value just calculated can be employed immediately after finishing the calculation instead of delay to the next cycle.

Table 5. Device utilization in FPGA board

Parameter	Used	Available	Utilization
Look up table (LUT)	9218	17600	53.8%
LUTRAM	24	6000	0.4%
Flip flop (FF)	2542	35200	7.2%
BRAM	1	60	47.6%
DSP	26	80	32.5%
IO	17	100	17%
BUFB	4	32	12.5%
MMCM	1	2	50%

3.2. Result when launching with HIL devices

In this section, the magnetizing process of induction motor, the waveform of stator current, speed response, and motor mechanical torque when operating in 4 difirrence scenarios, magnetizing process, no-load condition, and constant load ± 30Nm will be presented in Figures 7 to 11 to demonstrate the efficiency of FOC algorithm implemented on FPGA platform.

Scenario 1: Magnetizing process

Figure 7 shown the current response in magnetizing process, the magnetizing current is 7A as expected because it always equals to I_{sa} and double times I_{sb} and I_{sc} at the final value. The settling time is approximately 3 ms and there is no overshoot and ringing. Figure 7 to 11 shown that the stator current when motor operating with different load has a sinusoidal waveform, rotor speed responses equivalent to the speed command as in theory in every scenario. The design has been confirmed that it meets the requirement of system.

Scenario 2: No-load condition
Scenario 3: Operating with constant load 30Nm
Scenario 4: Operating with constant load -30Nm

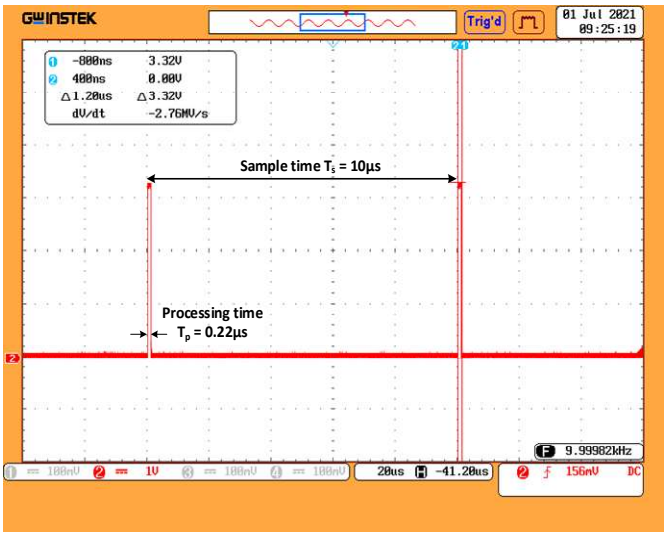


Figure 6. Procesing time in FPGA

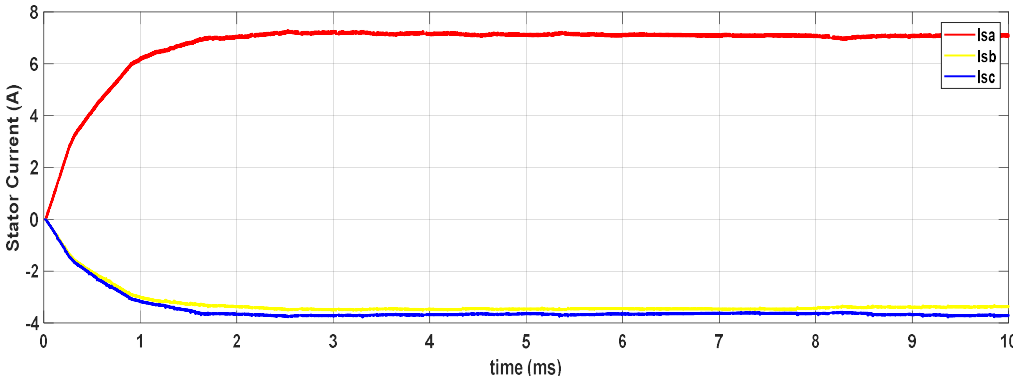
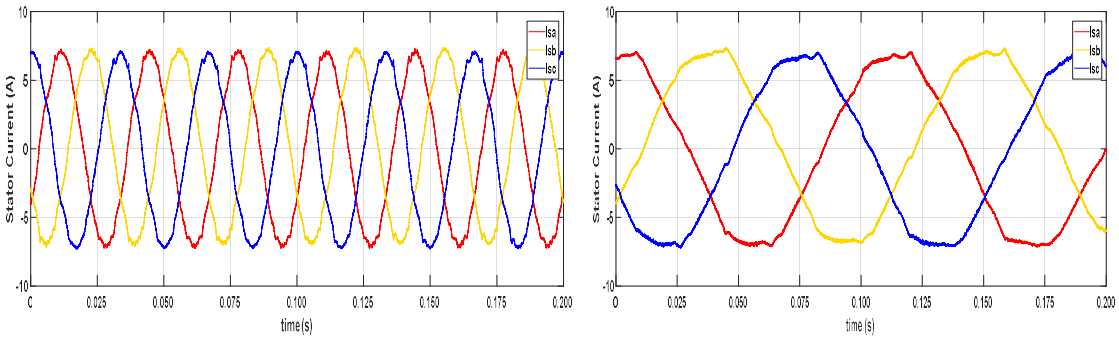
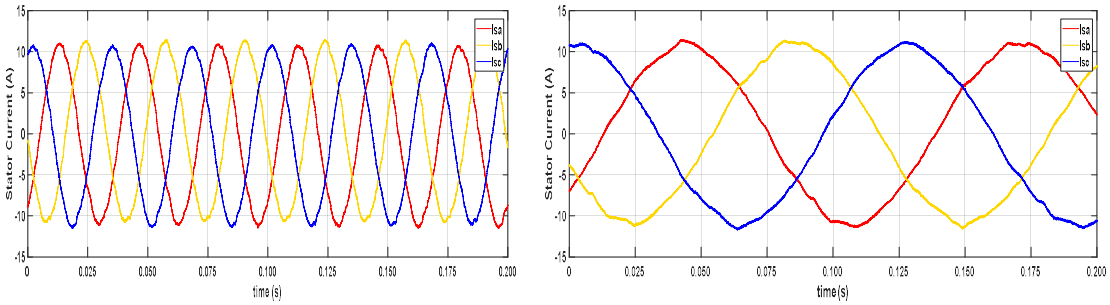


Figure 7. Magnetizing process of induction motor



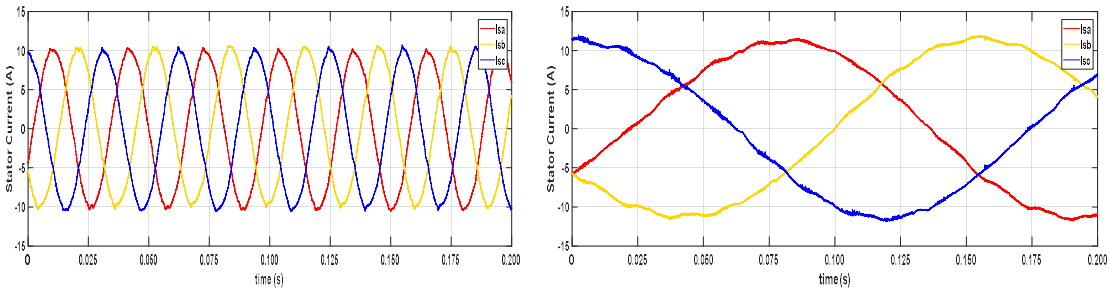
(a) (b)

Figure 8. Stator current of 2nd scenario when speed command is (a) 190rad/s and (b) 20 rad/s



(a) (b)

Figure 9. Stator current of 3rd scenario when speed command is (a) 190rad/s and (b) 20 rad/s



(a) (b)

Figure 10. Stator current of 4th scenario when speed command is (a) 190rad/s and (b) 20 rad/s

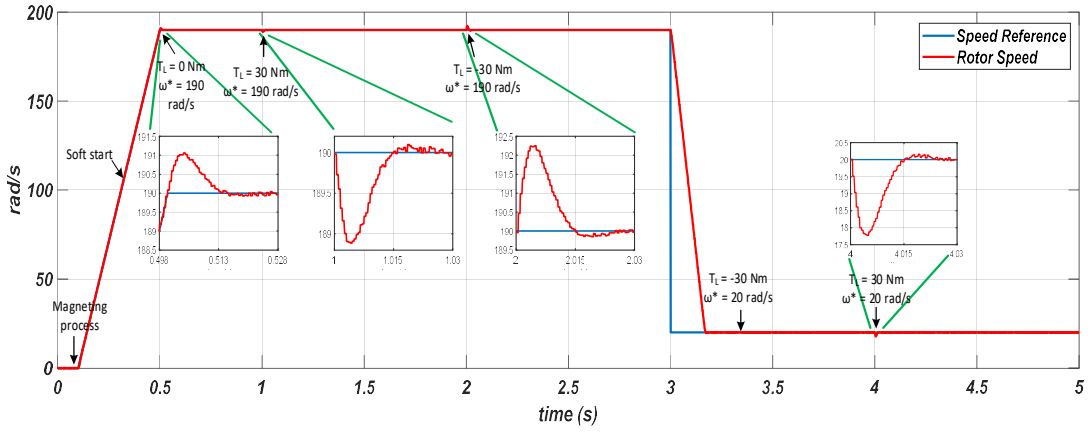


Figure 11. Speed response

4. CONCLUSION

The implementation of a vector control algorithm using FPGA for GaN-based induction motor drive was successfully developed by manual coding method in this contribution. From the achieved result, the processing time in one sample is 0.22µs, which meets the requirement of the power electronics system with switching-frequency up to 100kHz. The magnetizing process is fast and has no overshoot and ringing, the induction motor can operate under different load conditions and different speed commands with only one setting for parameters of PI controllers. These results verify the efficiency of the algorithm programmed on the FPGA platform and become a worthy replacement for DSP or microcontroller platform.

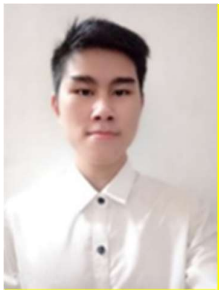
ACKNOWLEDGEMENTS

This research is funded by the Hanoi University of Science and Technology (HUST) under project number T2021-PC-001.

REFERENCES

[1] J. Bocker and S. Mathapati, "State of the Art of Induction Motor Control," *IEEE International Electric Machines & Drives Conference*, 2007, pp. 1459-1464, doi: 10.1109/IEMDC.2007.383643.
[2] R. Lai *et al*., "A High-Power-Density Converter," *IEEE Industrial Electronics Magazine*, vol. 4, no. 4, pp. 4-12, 2010, doi: 10.1109/MIE.2010.938722.
[3] K. Shirabe *et al*., "Efficiency Comparison Between Si-IGBT-Based Drive and GaN-Based Drive," *IEEE Transactions on Industry Applications*, vol. 50, no. 1, pp. 566-572, 2014, doi: 10.1109/TIA.2013.2290812.
[4] F. Canales, P. Barbosa, C. Aguilar and F. C. Lee, "A high-power-density DC/DC converter for high-power distributed power systems," *IEEE 34th Annual Conference on Power Electronics Specialist, PESC '03.,* pp. 11-18 vol.1, doi: 10.1109/PESC.2003.1218267.
[5] H. Li *et al*., "Design of a 10 kW GaN-based high power density three-phase inverter," *IEEE Energy Conversion Congress and Exposition (ECCE)*, 2016, pp. 1-8, doi: 10.1109/ECCE.2016.7855019.
[6] V. Vlatkovic and D. Borojevic, "Digital-signal-processor-based control of three-phase space vector modulated converters," *IEEE Transactions on Industrial Electronics*, vol. 41, no. 3, pp. 326-332, 1994, doi: 10.1109/41.293903.
[7] S. Meshkat and I. Ahmed, "Using DSP's in AC induction motor drives," *Control Engineering*, vol. 35, no. 2, pp. 54-56, 1988.
[8] A. O. Rait and P. Bhosale, "FPGA implementation of space vector PWM for speed control of 3-phase induction motor," *International Conference On Recent Advancements In Electrical, Electronics And Control Engineering*, 2011, pp. 221-225, doi: 10.1109/ICONRAEeCE.2011.6129782.
[9] J. J. R. Andina, M. J. Moure and M. D. Valdes, "Features, Design Tools, and Application Domains of FPGAs," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 4, pp. 1810-1823, 2007, doi: 10.1109/TIE.2007.898279.
[10] R. Joost and R. Salomon, "Advantages of FPGA-based multiprocessor systems in industrial applications," *31st Annual Conference of IEEE Industrial Electronics Society*, IECON, 2005, pp. 1-6 doi: 10.1109/IECON.2005.1568946.
[11] J. L. Bastos, H. P. Figueroa and A. Monti, "FPGA implementation of neural network-based controllers for power electronics applications," *Twenty-First Annual IEEE Applied Power Electronics Conference and Exposition*, *APEC* pp. 1-6, doi: 10.1109/APEC.2006.1620729.
[12] E. T. Mekonnen, J. Katcha and M. Parker, "An FPGA-based digital control development method for power electronics," *IECON 38th Annual Conference on IEEE Industrial Electronics Society*, 2012, pp. 222-226, doi: 10.1109/IECON.2012.6388804.
[13] P. Le-Huy, S. Guerette, L. A. Dessaint and H. Le-Huy, "Real-Time Simulation of Power Electronics in Power Systems using an FPGA," *Canadian Conference on Electrical and Computer Engineering*, pp. 873-877, doi: 10.1109/CCECE.2006.277356.
[14] E. Monmasson, L. Idkhajine, I. Bahri, M. Naouar and L. Charaabi, "Design methodology and FPGA-based controllers for Power Electronics and drive applications," *5th IEEE Conference on Industrial Electronics and Applications*, 2010, pp. 2328-2338, doi: 10.1109/ICIEA.2010.5515585.
[15] M. V. Chung, D. T. Anh and P. Vu, "A finite set-model predictive control based on FPGA flatform for eleven-level cascaded H-Bridge inverter fed induction motor drive*," International Journal of Power Electronics and Drive System (IJPEDS)*, vol. 12, no. 2, pp. 845-856, 2021.
[16] W. Zhao, B. H. Kim, A. C. Larson and R. M. Voyles, "FPGA implementation of closed-loop control system for small-scale robot," *ICAR '05. Proceedings, 12th International Conference on Advanced Robotics*, 2005, pp. 70-77, doi: 10.1109/ICAR.2005.1507393.
[17] Y. P. Siwakoti and G. E. Town, "Design of FPGA-controlled power electronics and drives using MATLAB Simulink," *IEEE ECCE Asia Downunder*, 2013, pp. 571-577, doi: 10.1109/ECCE-Asia.2013.6579155.
[18] K. Sugahara, S. Oida and T. Yokoyama, "High performance FPGA controller for digital control of power electronics applications, *IEEE 6th International Power Electronics and Motion Control Conference*, 2009, pp. 1425-1429, doi: 10.1109/IPEMC.2009.5157608.
[19] R. Kanti and M. G. Rathi, "Speed Control of Three-Phase Induction Motor using FPGA," *IJSRD-International Journal for Scientific Research & Development*, vol. 2, no. 12, pp. 207-209, 2015.
[20] K. C. Vinay, H. N. Shyam, S. Rishi and S. Moorthi, "FPGA Based Implementation of Variable-Voltage Variable-Frequency Controller for a Three Phase Induction Motor," *International Conference on Process Automation, Control and Computing*, 2011, pp. 1-6, doi: 10.1109/PACC.2011.5978884.
[21] G. El. Saady, E. N. A. Ibrahim and M. El-Besealy, "V/F control of Three Phase Induction Motor Drive with Different PWM Techniques," *Innovative Systems Design and Engineering*, vol. 4, no. 14, pp. 131-144, 2013.
[22] A. M. Eltamaly, "FPGA based speed control of three-phase induction motor using stator voltage regulator," *Mansoura Engineering Journal, Electrical Engineering Deptement Elmansourer University*, Egypt, vol. 32, no. 2, pp. 93-100, 2020, doi: 10.21608/bfemu.2020.128526.
[23] M. Sulaiman, F. A. Patakor, and Z. Ibrahim "DSP Based Implementation Of Field Oriented Control Of Three-Phase Induction Motor Drives," *International Journal of Research in Engineering and Technology*, vol. 2, no. 9, pp. 179-186, 2013.
[24] Q. N. Phung, "Vector Control of Three-Phase AC Machines – System Development in the Practice," Springrer, 2008.
[25] Xilinx, "7 Series FPGAs and Zynq-7000 SoC XADC Dual 12-Bit 1 MSPS Analog-to-Digital Converter," User Guide, Xilinx, 2018, [Online] Available: Retrieved from https://www. xilinx. com/support/documentation/user_guides/ug480_7Series_XADC.

BIOGRAPHIES OF AUTHORS



Tung Duong Do received his B.S. degress from Hanoi University of Science and Technology, Vietnam, in 2021 in Control Engineering and Automation. Since 2019, he has started to study his M.S degree at Hanoi University of Science and Technology. His research interests include power electronics, electrical machine drive, FPGA. He can be contacted at email: duong.dt212434m@sis.hust.edu.vn.

Nam Duong Le received a master's degree in Electronic Engineering from the University of Danang in 2011. Since 2004 he has been a lecturer at the Faculty of Engineering and Technology, Quy Nhon University. Currently, he is a PhD student at the Institute of Engineering and Automation-Hanoi University of Science and Technology. His research interests include microcontrollers, electric drive control, electronics engineering, power electronics, FPGA. He can be contacted at email: lenamduong@qnu.edu.vn.

Vu Hoang Phuong received his B.S., M.S., and Ph.D. degrees from Hanoi University of Science and Technology, Vietnam, in 2006, 2008, and 2014, respectively, all in Control Engineering and Automation. Since 2006 he has been employed at Hanoi University of Science and Technology, where he is a lecturer and researcher at school of electrical engineering. His research interests include modeling and controlling of power electronics converters for applications such as photovoltaic, wind system, electrical machine drive. He can be contacted at email: phuong.vuhoang@hust.edu.vn.

Nguyen Tung Lam received the B.S degree in Control and Automation Engineering from Hanoi University of Science and Technology, Hanoi, Vietnam, 2005, the M.S degree from Asian Institute of Technology, 2007, and the Ph.D from The University of Western Australia, 2014. He is current working as a lecturer at Department of Industrial Automation, School of Electrical Engineering, Hanoi University of Science and Technology. He is currently appointed as an Associate Professor in Control Engineering and Automation at Hanoi University of Science and Technology. His research interests include motion control, control system and its applications. He can be contacted at email: lam.nguyentung@hust.edu.vn.