

## SPEED CONTROL OF DC MOTOR USING FPGA

N. Genc    H. Hatas

*Electrical and Electronics Engineering Department, Van Yuzuncu Yil University, Van, Turkey  
nacigenc@yyu.edu.tr, hasanhatas@yyu.edu.tr*

**Abstract-** Direct Current (DC) motors have broad range of use. Therefore, studies on various semiconductor devices have been performed for best speed control. Field Programmable Gate Array (FPGA) is one of the most talented platforms for robust control. For generating FPGA code Xilinx System Generator (XSG) usage makes writing and saves time. In this study Permanent Magnet Direct Current (PMDC) motor was modelled and simulated by using MATLAB/Simulink environment and XSG block sets. Then XSG blocks were implemented to FPGA board by using Hardware Co-simulation tool. Proportional-Integral (PI) controller was used to control speed. The PI parameters were found by PID Tuner in MATLAB. Simulations was performed to verify the theoretical results and implementation. The results of the application were consistent with the results of the simulation study.

**Keywords:** DC Motor, FPGA, MATLAB/Simulink, XSG, DC Chopper.

### I. INTRODUCTION

PMDC motor is a machine that converts the current generated from the wire coils in the motor to mechanical energy. It supplies high torque initially and makes wide range speed control possible [1]. It has a broad range of applications including DC motors, electric traction, lifting equipment and rolling mills [2]. The crucial point for usage of DC motor is speed control under the variable load. For this purpose, different controllers are developed used. One of boards which make possible to perform these controllers is FPGA. It is capable of massively parallel operations. By using FPGA, numerous computational operations on real-time control systems can be made within a short period of time [3]. The system needs some important blocks such as controller for speed, incremental counter signal converting speed controller output to Pulse Width Modulation (PWM) signals. There are XSG toolbox in MATLAB/Simulink for both simulation and implementation. With these blocks in the toolbox many new models can be created. For simulation some MATLAB/Simulink blocks can be used. However, for implementation all blocks must be XSG blocks. Mainly used as a speed controller are proportional (P), proportional integral (PI), proportional derivative integral (PID), adaptive, fuzzy logic controller (FLCs) and Artificial Neural Networks (ANN) [4].

### II. MODELLING OF DC MOTOR AND PWM GENERATOR

DC motors have many features that are expected to be in a machine. Some of them are safe, long-term, cheap, and in addition to being able to operate at low voltage. Because of these, DC motors often find space in applications. The equivalent circuit of the PMDC motor is demonstrated in Figure 1.

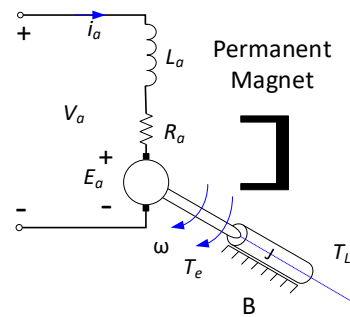


Figure 1. Equivalent circuit of PMDC motor

The equations of the DC motor which is used in the simulation are given below [5]. From Kirchoff's circuit laws:

$$V_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + E_a(t) \quad (1)$$

The motor back emf is expressed as:

$$E_a(t) = K_b \omega(t) \quad (2)$$

In order to maintain energy equilibrium in the system, the sum of the motor moments must be zero. In this case; torque developed by the motor is:

$$T_e - T_j - T_\omega - T_L = 0 \quad (3)$$

When rearranged Equation (3):

$$K_t i_a - J \frac{d\omega}{dt} - B\omega - T_L = 0 \quad (4)$$

When converted Equations (1) and (3) into Laplace transform;

$$s i_a(s) - i_a(0) = -\frac{R_a}{L_a} i_a(s) - \frac{K_v}{L_a} \omega(s) + \frac{1}{L_a} V_a(s) \quad (5)$$

$$s \omega(s) - \omega(0) = \frac{K_t}{J} i_a(s) - \frac{B}{J} \omega(s) - \frac{1}{J} T_L(s) \quad (6)$$

Assume that initial conditions are accepted as zero, the equations will be below:

$$i_a(s) = -\frac{K_v \omega(s) + V_a(s)}{L_a s + R_a} \quad (7)$$

$$\omega(s) = \frac{K_v i_a(s) - T_L(s)}{J s + B} \quad (8)$$

The block diagram of the PMDC motor with the help of the equations is shown in Figure 2.

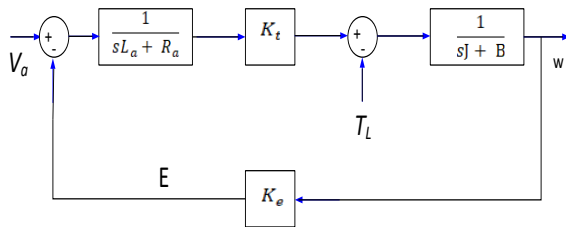


Figure 2. General transfer function of PMDC motor

Table 3. DC motor parameters

Parameters	Values	Units
Armature Resistance, $R_a$	4.3	$\Omega$
Armature Inductance, $L_a$	0.0263	H
Moment of Inertia, $J$	0.022	Nm <sup>2</sup>
Friction constant, $B$	0.00034	Nms/rad
Torque constant, $K_t$	0.662	Nm/A
EMF constant, $K_b$	0.662	Vs/rad
Friction moment	0.17437	N.m
Armature Voltage	180	V
Power	0.4	HP
Nominal Speed	1750	RPM

When DC motor parameters in Table 3 are replaced in Figure 2, transfer function of DC motor we used is shown as Figure 4.

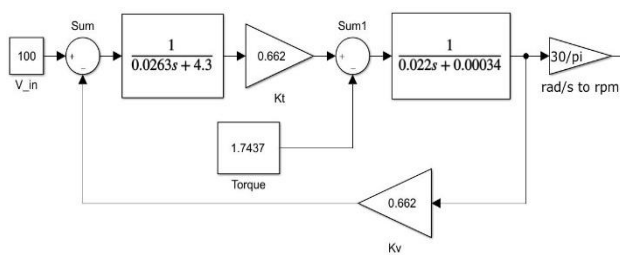


Figure 4. Transfer function of PMDC motor used

The technique of Pulse Width Modulation (PWM) provides a logic "1" and logic "0" for a specific period of time. PWM is used in lots of applications like speed control of DC motor [6]. Assuming a modulated signal as a square waveform  $f(t)$  with a high value ( $y_{max}$ ), a low value ( $y_{min}$ ) and a duty cycle, the mean waveform value is written as below:

$$y = \frac{1}{T} \int_0^T f(t) dt \quad (9)$$

When  $f(t)$  is a square wave, maximum value is for the highest duty cycle and minimum value is for low value of duty cycle [7]. Then equation (9) will be as the following:

$$f(x) = D.y_{max} + (1-D).y_{min} \quad (10)$$

In this study we use right-aligned PWM shown in Figure 5. For the variable duty cycle, the left edge of the signal is fixed at the edge level and the rising edge is modulated [8].

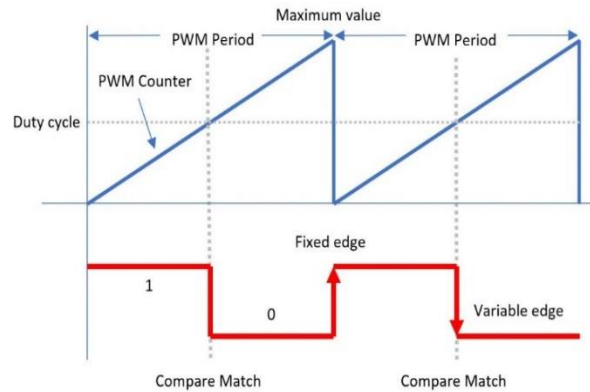


Figure 5. Right-aligned PWM generator

A Counter and a Relational block to create PWM Generator shown in Figure 6. When input signal is greater less than sawtooth signal, the relational block yields a logic "1".

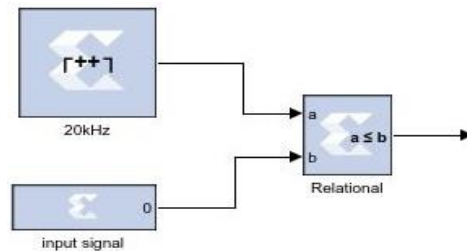


Figure 6. PWM generator with XSG block set

In this study Genesys II was used as a FPGA development board. When using Hardware Co-simulation, Genesys II is able to support 50 MHz as a clock speed. Thus, the period of the counter is set to 1/50000000. That means this counter is 50 million counts per second. Because we want to run the simulation in 20 kHz, counter's type is chosen as count limited and count to value option is adjusted to 2499 for the counter begins from 0. In doing so, we set frequency of the simulation to 20kHz. When set input of blocks shown in Figure 6 to value of 249, the outputs of PWM generator illustrated in Figure 7.

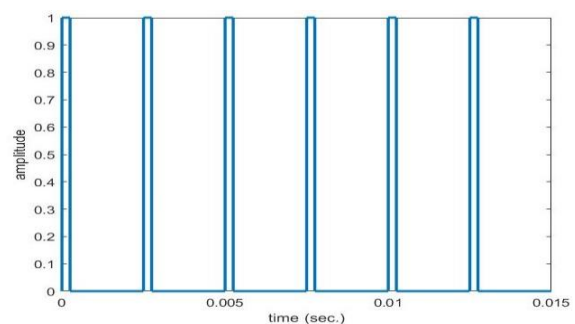


Figure 7. PWM output, setting input to 249

When input of blocks shown in Figure 6 is adjusted to value of 2249, the PWM generator yields output shown in Figure 8.

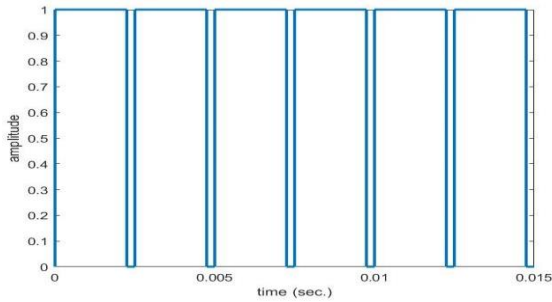


Figure 8. PWM output, setting input to 2249

### III. CHOPPER TOPOLOGY

The use of chopper drivers in mobile applications, which are connected to change the armature voltage between a constant voltage source and the motor, is quite common. The DC-DC converter drives are used both to control the armature voltage and to restore the source of energy to the DC-DC converter motors by regenerative braking [9]. Chopper circuit is shown in Figure 9.

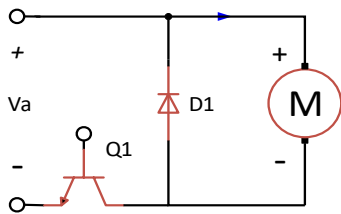


Figure 9. Chopper circuit

A transistor chopper controlled PMDC motor driver has transistor Q1 which is driven periodically with period 'T' and stays on for a time ton. In this interval, armature current increases from ia2 to ia1 shown in Figure 10. Because motor voltage is equal to the source during this interval, it is known as duty interval. At t=ton transistor Q1 is turned off [10].

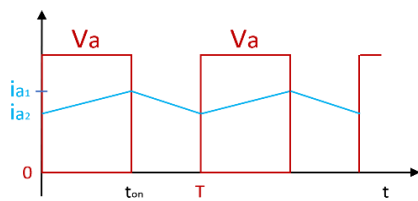


Figure 10. Switching pulses

For forward power control, transistor Q1 must operate. When Q1 is opened, the voltage source appears at the motor terminals and motor runs shown in Figure 11 [11].

When the transistor is opened,  $0 \leq t \leq t_{on}$ , the motor terminal voltage equals source voltage  $V_a$ . The operation is characterized by

$$R_a i_a(t) + L_a \frac{di_a(t)}{dt} + E_a(t) = V_a(t) \quad (11)$$

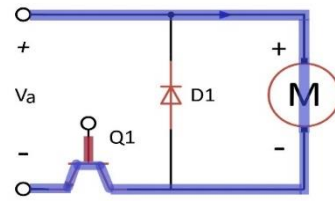


Figure 11. T Forward power control

Motor current flows through freewheels diode D1 shown in Figure 12 and motor terminals voltage is zero in the course of interval  $t_{on} \leq t \leq T$  [10].

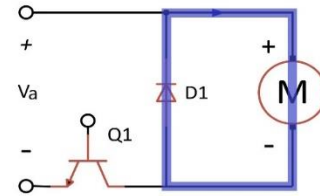


Figure 12. Freewheeling interval

During this interval motor operation, known as freewheeling interval [10], is described by,

$$R_a i_a(t) + L_a \frac{di_a(t)}{dt} + E_a(t) = 0 \quad (12)$$

In this study IR2113 was used to drive MOSFET. IR2113 applies the control signal to the MOSFETs by increasing the high side and the low side MOSFET to the level of the supply voltage isolated from each other. In this study only low-side driver was used. The basic components of a low-side driver circuit using an IR2113 [12] were given in Figure 13 with the Proteus/ISIS drawing.

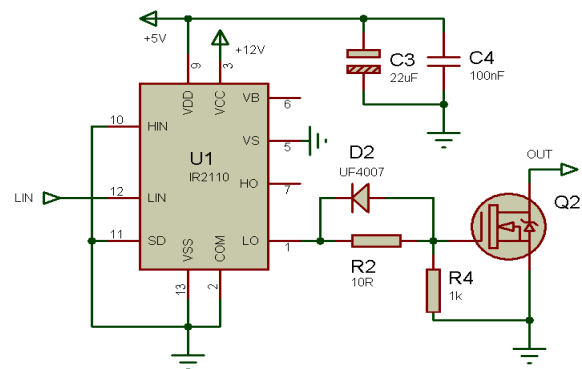


Figure 13. Chopper driver circuit

### IV. CONTROLLER

PI control is a kind of closed loop control method. In the closed loop control systems, the information related to the current state of the system is provided with the help of sensors. At the same time, the controller detects and corrects the error that occurs when compared to the reference input at the output. The data from the feedback is compared with the desired reference value to determine the error signal. The detected error signal is sent to the controller [13].

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt \quad (13)$$

The transfer function of Equation (13) can be formulated as

$$K(s) = K_p + \frac{K_i}{s} \quad (14)$$

The PI controller system block diagram of this paper is shown in Figure 14 and its XSG block design can be seen in Figure 15.

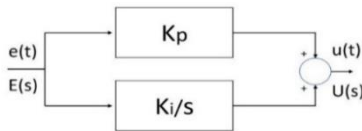


Figure 14. PI controller block diagram

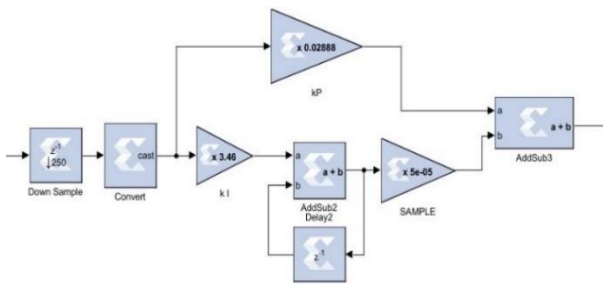


Figure 15. PI Controller Block Diagram with XSG blockset

The PID controller setting is easy to find, but it is a complex task to find the gain set that provides the best performance for your control system. Traditionally, PID controllers are set using manual or rule-based methods. Manual setting methods are repetitive and time-consuming and can cause damage if used on hardware. Rule-based methods also have serious restrictions: they do not support unstable plants, higher order plants or plants have little or no delay [14].

MATLAB has PID Tuner application to find P, PI, PID parameters shown in Figure 16. The MATLAB/Simulink PID controller block offers some key advantages for controller design. The user has the flexibility to experiment with writing on the PID parameters to achieve better system performance. As a result, the PID controller design process has been greatly accelerated [15].

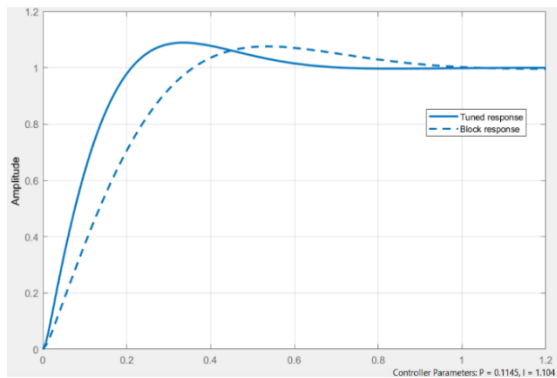


Figure 16. PID Tuner

By adjusting Response Time and Transient Behavior shown in Figure 16, some acceptable PI parameters were found. One of these parameter couple is  $K_p = 0.1145$  and  $K_i = 1.104$ .

### V. STUDY RESULTS

Simulation blocks and code generated XSG blocks are set to run PWM out at 20 kHz. The parameters of PI controller were set is  $K_p = 0.1145$  and  $K_i = 1.104$  by using MATLAB/Simulink PID Tuner.

#### A. Simulation Result

For simulation result, MATLAB/Simulink blocks are shown in Figure 17. In first section chopper circuit was designed and measured the chopped voltage. In second section transfer function of PMDC motor created and speed converted into RPM. Lastly speed of motor was subtracted from reference speed. The output of this operation went to PI block. PI controller output was limited between '0' and '1' because the sawtooth signal increases from '0' to '1'. Then it compared to 20 kHz sawtooth signal. The output of this comparison was given to MOSFET in chopper circuit shown in Figure 17.

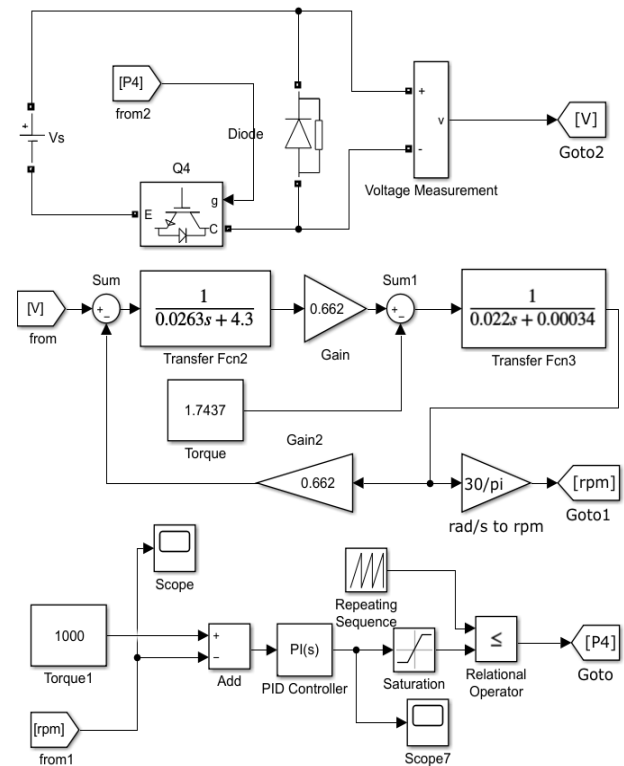


Figure 17. Simulation blocks

PMDC Motor simulation was created by MATLAB/Simulink blocks shown in Figure 17. Result of this simulation, speed of PMDC motor is illustrated in Figure 18. When the speed of the motor increased to 1000 RPM, it traveled at 1000 RPM after making an approximate 6% overshoot. It sat at a reference speed after about 1 seconds.

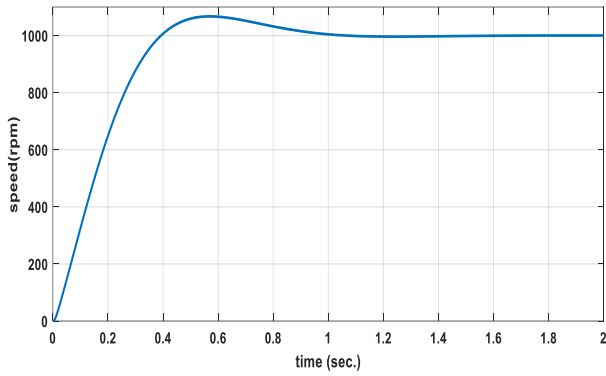


Figure 18. Motor speed on simulation

**B. Experimental Result**

For experimental result, blocks generated code are shown in Figure 19. The duration of two consecutive signals was calculated using the increasing edge of the logic signal taken from the shaft of the PMDC motor. Then this duration was converted into RPM. After subtracted from reference speed it went to PI controller created by XSG blocks shown in Figure 19. PI controller output was compared to 20 kHz sawtooth signal. The output of this comparison was given to physical port represented Gateway Out shown in Figure 19. Gateway Out port was sent signal to Chopper Driver and motor's shaft turned. And whole process was repeated during operation.

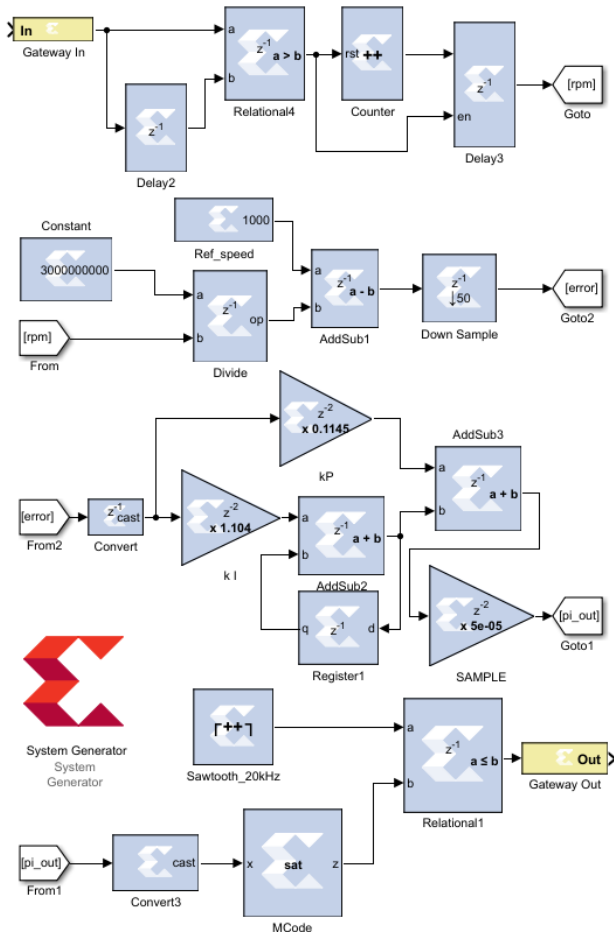


Figure 19. XSG blockset generated code

We used Arduino Mega board to calculate and show on MATLAB/Simulink scope by using m- function. Because both Arduino and FPGA boards run by MATLAB/Simulink on one PC, Arduino board was run before FPGA board. Therefore, there are about 11 seconds delay in Figure 20. When the speed of the motor increased to 1000 RPM, it traveled around 1000 RPM after making an approximate 7% overshoot. It sat at a reference speed after about 2 seconds.

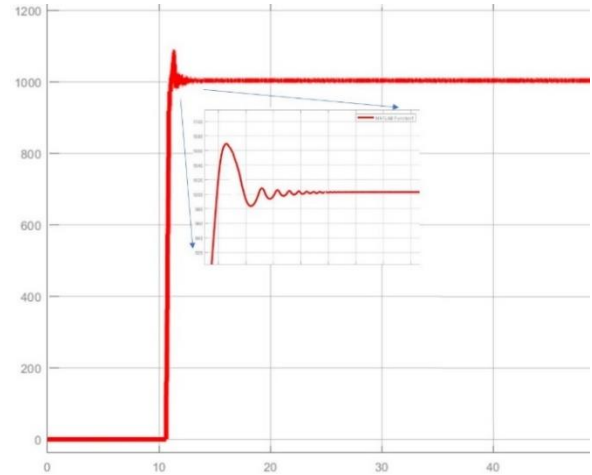


Figure 20. Motor speed on experiment

**VI. CONCLUSIONS**

The results obtained from the simulation and experimental studies in the scope of the study correspond to 90% because of the fact that each element works ideal in the simulation program, the driver circuit used in the experimental operation transmits the signals with a certain delay. The implementation of FPGA board which has higher speed operation compared to other microcontrollers for speed control of chopper based PMDC is the main contribution of this study.

**NOMENCLATURES**

- $L_a$  : Armature inductance
- $R_a$  : Armature resistance
- $i_a$  : Armature current
- $i_f$  : Field current
- $e_a$  : Input voltage
- $e_b$  : Back electromotive force (EMF)
- $T_m$  : Motor torque
- $\omega$  : An angular velocity of rotor
- $J$  : Rotating inertial measurement of motor bearing
- $K_b$  : EMF constant
- $K_t$  : Torque constant
- $B$  : Friction constant

**ACKNOWLEDGMENTS**

The authors would like to acknowledge Scientific Research Fund of the Van Yuzuncu Yil University (YYU-BAP) for the financial support of this research with project number of FBA-2017-5891.

### REFERENCES

- [1] M.A.H. Azman, J.M. Aris, Z. Hussain, A.A.A. Samat, A.M. Nazelan, "A Comparative Study of Fuzzy Logic Controller and Artificial Neural Network in Speed Control of Separately Excited DC Motor", 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2017), pp. 336-341, Penang, Malaysia, November 2017.
- [2] P. Kumar, R.K. Behera, D.V. Bhaskar, "Novel Closed Loop Speed Control of Permanent Magnet Brushless DC Motor Drive", Technologies for Smart-City Energy Security and Power (ICSESP), IEEE, pp. 1-6, Bhubaneswar, India, March 2018.
- [3] A. Molina, G. Tello, L. Ibarra, B. Mac Cleery, M. Ramirez, "Experimental Study for FPGA PID Position Controller in CNC Micro-Machines", 5th IFAC Symposium on Information Control Problems in Manufacturing: INCOM 2015, Vol. 48, Issue 3, pp. 2203-2207, 31 August 2015.
- [4] A. Bernard, "Speed Control of Separately Excited Dc Motor Using Artificial Intelligent Approach", Faculty of Electrical and Electronic Engineering, University of Tun Hussein Onn, Malaysia, January 2013.
- [5] Y. Oner, "Control of Output Voltage of Permanent Magnet DC Motor Driven Permanent Magnet Synchronous Generator", M.Sc. Thesis, Yildiz Technical University, Istanbul, Turkey, 2009.
- [6] C. Agarwal, A. Gupta, H. Rana, "Performance Analysis and FPGA Implementation of Digital PID Controller for Speed Control of DC Motor", International Journal of Computers & Technology, Issue 3, Vol. 7, pp. 638-645, June 2013.
- [7] Kh.S. Gaeid, J.A. Hameed, M. Hameed Ali, M.K. Habeeb, "Static DC Motor Speed Controlled Parameters Correction", British Journal of Applied Science & Technology, pp. 586-597, July-September 2013.
- [8] F. Mendoza Mondragon, A. Espinosa Calderon, A. Martinez Hernandez, J. Rodriguez Resendiz, "Implementation of High Resolution Unipolar PWM Inverter Using Xilinx System Generator", Mechatronics, Adaptive and Intelligent Systems (MAIS), IEEE Conference, pp. 1-6, Hermosillo, Mexico, 20-22 October 2016.
- [9] M.H. Rashid, "Power Electronics Devices, Circuits, and Applications", Pearson Prentice Hal, 2004.
- [10] P. Chittal, P. Reddy, J. Kittur, P. Patil, P. Pattar, "Open Loop and Closed Loop Speed Control of Separately Excited DC Motor", MJRET, Hubli, India, January 2015.
- [11] R. Krishnan, "Electric Motor Drivers Modeling, Analysis, and Control", Prentice Hall, 2001.
- [12] S. Tahmid, "Using the High-Low Side Driver IR2110 - Explanation and Plenty of Example Circuits", 2013, <https://tahmidmc.blogspot.com/2013/01/using-high-low-side-driver-ir2110-with.html>.
- [13] H. Yalduz, "Design and PI Control of DC-DC Buck Converter", M.Sc. Thesis, Van Yuzuncu Yil University, Van, Turkey, January 2015.
- [14] MathWorks PID TUNER, <https://www.mathworks.com/discovery/pid-tuning.html>.
- [15] G. Wong, J. Wang, "Real-Time Rendering: Computer Graphics with Control Engineering", CRC Press, 2013.

### BIOGRAPHIES



**Naci Genc** received the B.Sc., M.Sc., and Ph.D. degrees from Gazi University (Ankara, Turkey), Van Yuzuncu Yil University (Van, Turkey) and Gazi University in 1999, 2002, and 2010, respectively. He is a Professor in the Electrical and Electronics Engineering Department, Van Yuzuncu Yil University. His interests include energy conversion systems, power electronics and electrical machines.



**Hasan Hatas** was born in Kirikkale, Turkey, 1989. He received the B.Sc. and the M.Sc. degrees from University of Erciyes (Kayseri, Turkey) in 2014. He is continuing the M.Sc. degree at Van Yuzuncu Yil University (Van, Turkey). Currently, he is a research Assistant in Department of Electrical and Electronics Engineering at Van Yuzuncu Yil University (Van, Turkey). His research interests are renewable energy, power electronics and electric machines.