Processor in the Loop for Testing Series Motor Four Quadrants Drive Direct Current Chopper for Series Motor Driven Electric Car

Part1: Chopper Operation Modes Testing

S. Arof, N. H. Diyanah, N. M. Yaakop, P. A. Mawby and H. Arof

Abstract This paper describes the process of testing a four quadrant drive direct current (DC) chopper (FQDC) that controls a DC series motor using the processor in the loop (PIL) technique. The DC motor serves as a propeller to an electric vehicle (EV). The main function of the four quadrant drive chopper is to provide a smooth operation for the electric vehicle(EV) while optimizing battery power consumption and maximizing distance traversed. In the processor in the loop (PIL) test, MATLAB/Simulink environment was used as the platform for the four quadrants drive chopper and electric vehicle, whereas in the hardware part, the FQDC was controlled by three PIC microcontrollers. Serial communication was used as the channel of data transfer between the hardware and software. The simulation results of the ATLAB/Simulink indicate that the FQDC controller was able to control the DC motor that drove the EV. Overall, the PIL technique was suitable for testing and validating the operation of the FQDC, its controllers and the control algorithm.

Keywords Direct current drive · Electric vehicle and hybrid electric vehicle Series motor · Four quadrant drive chopper

S. Arof (✉) · N. H. Diyanah · N. M. Yaakop
Malaysian Spanish Institute, Universiti Kuala Lumpur, 09000 Kulim, Kedah, Malaysia
e-mail: saharul@unikl.edu.my

N. H. Diyanah
e-mail: diyanahisham94@gmail.com

S. Arof · P. A. Mawby
University of Warwick School of Engineering, Coventry CV47AL, UK
e-mail: P.A.Mawby@warwick.ac.uk

H. Arof
Engineering Department, Universiti Malaya, Jalan Universiti, 50603 Kuala Lumpur, Malaysia
e-mail: ahamzah@um.edu.my

A. Ismail et al. (eds.), *Advanced Engineering for Processes and Technologies*,
Advanced Structured Materials 102, https://doi.org/10.1007/978-3-030-05621-6_5

1 Introduction and Review

Using electric vehicles (EV) and hybrid electric vehicles (HEV) is one of the solutions to reduce the use of fossil fuel [1]. The present EV/HEV mainly use alternating current(AC) drive but interest in direct current (DC) drive has increased since the introduction of highly efficient brushed direct current motor that operates at low voltage [2]. One of the applications suitable for these direct current(DC) motors is for propelling electric vehicles(EVs) and hybrid electric vehicles(HEVs).

1.1 Proposed Four Quadrant DC Drive Chopper Design

The main purpose of developing a direct current (DC) drive system for EV/HEV using direct current brushed motor is to produce an efficient but cheap hybrid or electric car that is affordable. Brushed direct current motor requires four quadrants chopper to operate in various modes [3–5]. The proposed four quadrants chopper, shown in Fig. 1, has seven modes of operation namely driving, reverse, field weakening, parallel mode, regenerative braking, resistive braking, and generator [6]. This is much better than what the conventional H bridge chopper can offer.

The mathematical model of each mode of the four quadrants drive chopper (FQDC) is given a set of equations below [6].

Driving and reverse mode equations

$$e_g = K_{bemf} i_f \omega \tag{1}$$

$$T_d = K_t i_a i_f \tag{2}$$

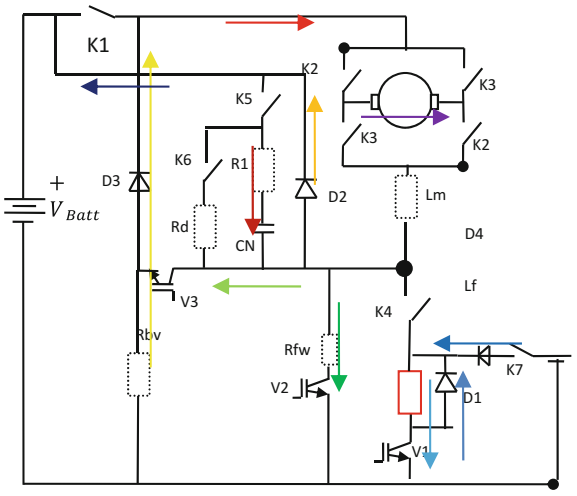


Fig. 1 New four quadrants DC Chopper

$$T_d = \mathrm{J}\frac{d\omega}{dt} + B\omega + T_L \tag{3}$$

$$\frac{d}{dt}\begin{bmatrix} I_a \\ I_f \\ V_c \end{bmatrix} = \begin{bmatrix} \frac{-(R_a+R_c)}{(L_a)} & \frac{R_c}{(L_a)} & \frac{1}{(L_a)} \\ \frac{R_c}{(L_f)} & \frac{-(R_f+R_c)}{(L_f)} & \frac{1}{(L_f)} \\ \frac{-1}{(C)} & \frac{1}{(C)} & 0 \end{bmatrix}\begin{bmatrix} I_a \\ I_f \\ V_c \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{L_f} & \frac{-1}{L_f} & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} V_{dc}K_{drv} \\ E_g \\ 0 \end{bmatrix} \tag{4}$$

Field weakening

$$\frac{d}{dt}\begin{bmatrix} I_a \\ I_f \\ V_c \end{bmatrix} = \begin{bmatrix} \frac{-(R_a+R_c)}{(L_a)} & \frac{R_c}{(L_a)} & \frac{1}{(L_a)} \\ \frac{R_c}{(L_f)} & \frac{-(R_f+R_c)}{(L_f)} & \frac{1}{(L_f)} \\ \frac{-1}{(C)} & \frac{1}{(C)} & 0 \end{bmatrix}\begin{bmatrix} I_a \\ I_f \\ V_c \end{bmatrix} \begin{bmatrix} 0 & 0 & \frac{R_c}{R_{fw}*L_a} \\ \frac{1}{L_f} & \frac{-1}{L_f} & \frac{-R_c}{R_{fw}*L_f} \\ 0 & 0 & \frac{1}{(R_{fw}*C)} \end{bmatrix}\begin{bmatrix} V_{dc}K_{drv} \\ E_g \\ V_f K_{fw} \end{bmatrix} \tag{5}$$

Generator

$$\frac{d}{dt}[I_f] = \left[-\left(\frac{R_f}{L_f}\right)\right][I_f]\left[\frac{1}{(L_f)}\right][V_{ext}K_{drv}] \tag{6}$$

$$\frac{d}{dt}[I_a] = \left[\frac{-(R_a+R_{bh})}{L_a}\right][I_a]\begin{bmatrix} \frac{1}{L_a} & -\frac{1}{L_a} \end{bmatrix}\begin{bmatrix} E_g \\ V_{dc} \end{bmatrix} \tag{7}$$

Parallel

$$\frac{d}{dt}[I_f] = \left[-\left(\frac{R_f}{L_f}\right)\right][I_f]\left[\frac{1}{(L_f)}\right][V_{ext}K_{drv}] \tag{8}$$

$$\frac{d}{dt}[I_a] = \left[\frac{-(R_a)}{L_a}\right][I_a]\begin{bmatrix} \frac{1}{L_a} & -\frac{1}{L_a} \end{bmatrix}\begin{bmatrix} V_{dc} \\ E_g \end{bmatrix} \tag{9}$$

Regenerative brake pre excitation

$$\frac{d}{dt}[I_f] = \left[-\left(\frac{R_f}{L_f}\right)\right][I_f]\left[\frac{1}{(L_f)}\right][V_{ext}K_{drv}] \tag{10}$$

Regenerative brake resonance

$$\frac{d}{dt}\begin{bmatrix} I_f \\ V_c \end{bmatrix} = \begin{bmatrix} \frac{-(R_c+R_f)}{L_f} & \frac{-1}{L_f} \\ \frac{I_f}{C} & 0 \end{bmatrix}\begin{bmatrix} I_f \\ V_c \end{bmatrix}\begin{bmatrix} \frac{1}{L_f} & 0 \end{bmatrix}\begin{bmatrix} V_{dc}K_{drv}K_{rgb} \\ 0 \end{bmatrix} \tag{11}$$

Regenerative and resistive brake general equations

$$\frac{d}{dt}[I_f] = \left[-\left(\frac{(R_a+R_{bh}+R_f)}{L_a+L_f}\right)\right][I_f]\left[\frac{1}{(L_f)}\right][E_gK_{drv}K_{rgb}] \tag{12}$$

$$\frac{d}{dt}[I_{batt}] = \left[-\left(\frac{R_a+R_{batt}}{L_a}\right)\right][I_{batt}]\left[\frac{1}{(L_a)}\right][E_g - V_{batt}] \tag{13}$$

$$\frac{d}{dt}[I_{short}] = \left[-\left(\frac{R_a+R_{short}}{L_a}\right)\right][I_{short}]\left[\frac{1}{(L_a)}\right][E_gK_{rgb}] \tag{14}$$

$$\frac{d}{dt}[I_{fw}] = \left[-\left(\frac{R_{fw}+R_a}{L_a}\right)\right][I_{fw}]\left[\frac{1}{(L_a)}\right][E_gK_{fw}] \tag{15}$$

2 Simulation Model Established from Mathematics Equation

Each set of mathematical equations can be transformed into a simulation model as shown in Fig. 2. The simulation model generated from mathematical model is fast and as a result, a longer simulation is feasible. A similar simulation model can also be created using MATLAB/Simulink toolbox by selecting appropriate components from its library as shown in Fig. 3. This model is easier to establish but is slower in execution. Thus for the same execution time, this model produces a shorter simulation [7].

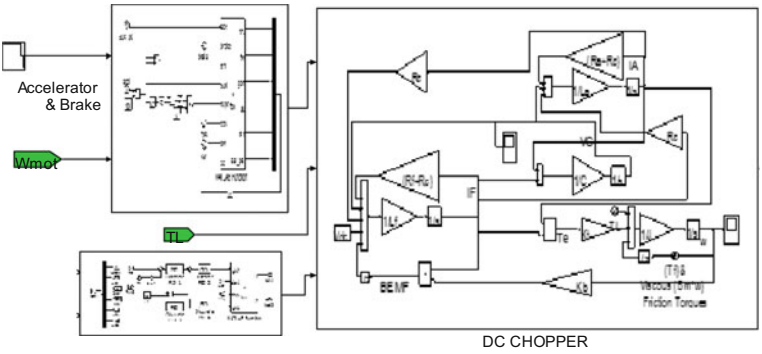


Fig. 2 Complete integrated transfer function of FQDC chopper

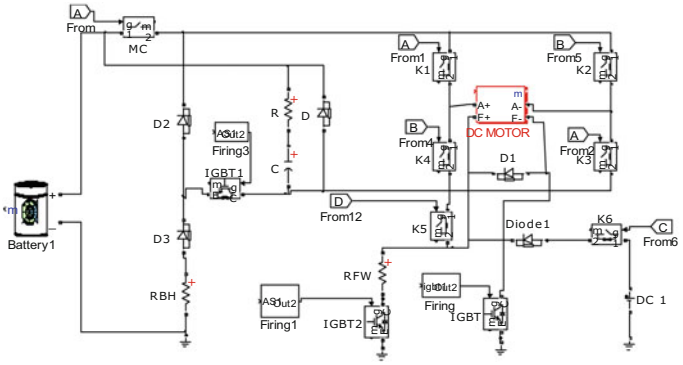


Fig. 3 Overview of proposed four quadrants DC chopper simulation

2.1 Methodology

2.1.1 Processor in the Loop (PIL)

The performance of the proposed FQDC and controllers can be evaluated by simulation or by implementing a real hardware prototype. A real hardware implementation is very desirable as it reveals the true potential and limitations of the system. However, it is very time consuming and expensive. In contrast, simulation is fast and cheap, but the value of its results depends on its ability to emulate the system accurately [8–10]. Hence, simulation also known as virtual experiment, is chosen as a platform to test the four quadrants drive chopper(FQDC) and the algorithm developed for the Electric vehicle or hybrid electric vehicle(EV/HEV) application. Processor in the loop (PIL) is a part of the hardware in the loop (HIL) technique used for verifying hardware in conjunction with simulation software [11–14]. In this study the main objective of applying the PIL is to test the

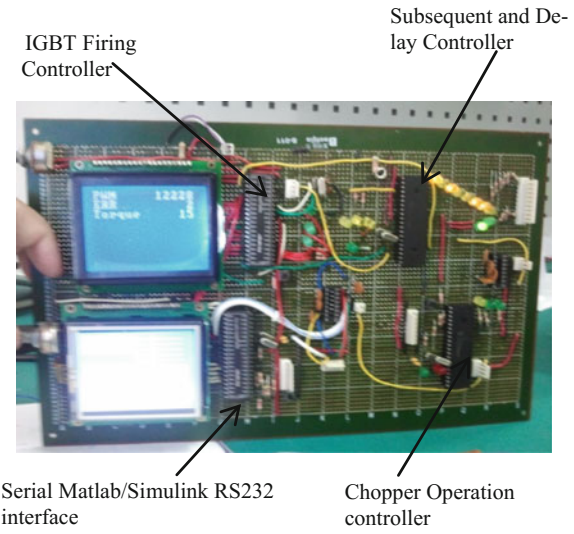


Fig. 4 Four quadrants chopper controller

FQDC hardware controllers and its control algorithm. Since a four quadrants chopper has many operations, it requires three sub controllers (PIC microcontrollers), which are assigned for chopper operation (COC), subsequent and delay (SAD), and insulated gate bipolar transistor(IGBT) firing controller (IFC) (Fig. 4).

2.1.2 Chopper Operation Controller

MATLAB/Simulink receives data from sensors at the direct current(DC) motor and vehicle. Some important data are field current, armature current, armature voltage, battery voltage, motor speed, accelerator pedal position, brake pedal position, etc. In turn, it sends the data to the chopper operation controller and IGBT firing controller via serial interface. The chopper operation controller selects the suitable mode of operation for the DC motor to run based on the data received [15, 16]. Specifically, the controller runs an expert system(ES) algorithm which processes the input data and determines the best operation mode and passes the decision and other information to the subsequent and delay (SAD) controller via serial communication (UART). Details of the expert system(ES) algorithm is not discussed in this paper.

2.1.3 Subsequent and Delay Controller (SAD)

The subsequent and delay(SAD) controller has four important tasks. The first task is to control the contactors' switching according to the chopper operation. The second is to introduce a delay in the transition from changing one chopper operation to another operation for soft switching [17] transition. The third is to send or remove

ready signal via input or Output(I/O) pin to the igbt firing controller(IFC) in order to start or stop IGBTs firing sequence. The last task is to relay the selected chopper operation mode to the IFC via SPI communication. The delay subroutine algorithm is implemented in this controller but its details are not covered in this paper.

2.1.4 IGBT Firing Controller (IFC)

Besides data from the SAD controller, the IFC also receives several data signals from MATLAB/Simulink such as, accelerator pedal position, brake pedal position, speed, torque, current, voltage, etc. Then the IFC produces the PWM signals for firing the IGBTs and sends information on the chopper operation mode back to MATLAB/Simulink via serial interface. The PWM signals for firing the IGBTs are the output of digital or discrete proportional Integrator derivate (PID) algorithm used to control the direct current(DC) series motor with direct torque control (DTC) implemented in MATLAB Simulink model. Details of the digital PID algorithm and DTC implemented in this controller are also not discussed in this paper [18].

The flow of the data and signals in the Processor in the loop (PIL) from MATLAB/Simulink to FQDC controllers and vice versa is simplified in the block diagram shown in Fig. 5. The signals flow from FQDC controllers to the Matlab/Simulink for communication and data distribution is shown in block diagram as in Fig. 6.

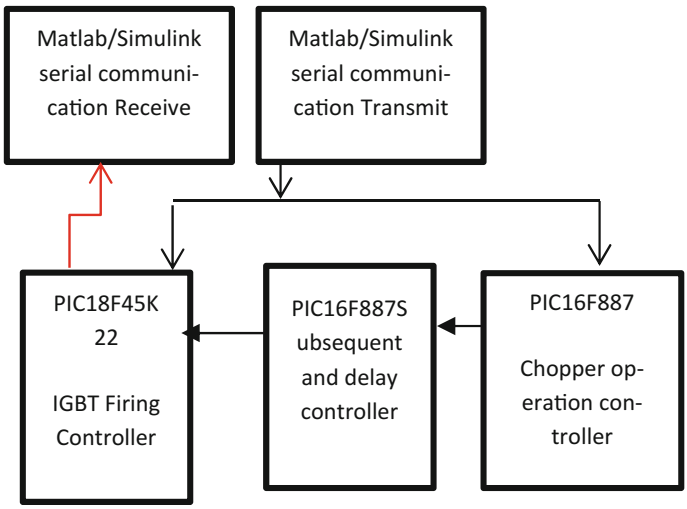


Fig. 5 communication and data distribution between controllers

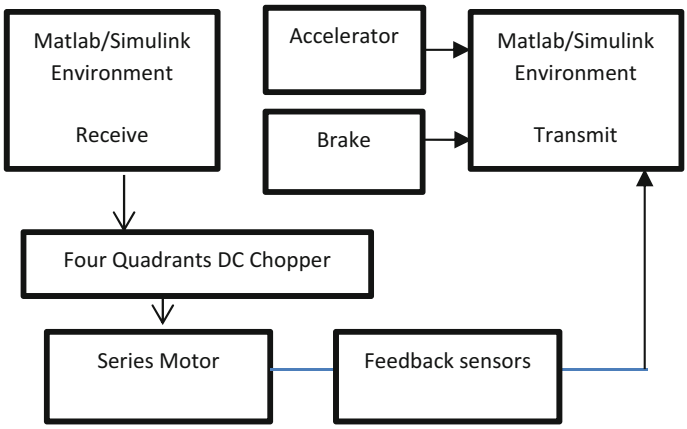


Fig. 6 Block diagram of controller data path

2.1.5 Signal Conversion

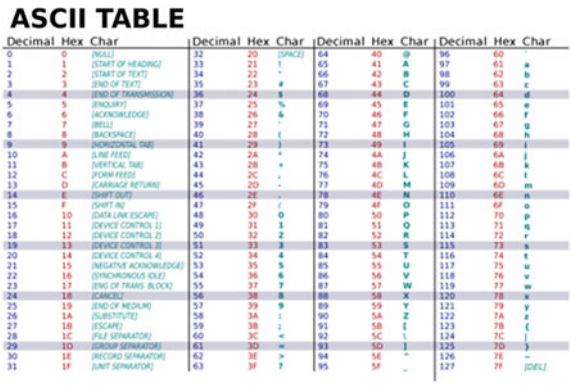
MATLAB/Simulink works with data and signals from the FQDC and EV in the form of real, integer, signed integer, double etc. The hardware controllers (PIC microcontrollers) on the other hand, are all running in digital/binary forms. Therefore signal conversion is required. First, all input signals sent to the controllers, such as speed, current, torque, voltage and others, are transformed into voltages. As the microcontrollers run on 5 V direct current (DC) power supply, the voltages are then adjusted using voltage dividers or similar circuits to step down the voltages levels so that their final values are in the range of zero(0)V to five(5)V. Then analog to digital conversion is carried out using analog to digital converter (A/D) with a 10-bit resolution. After the conversion, 5 V is represented by 1024 in decimal and 0 V is represented by zero in decimal. Given a normalized voltage level from zero to five voltage (0 V-5 V), its decimal value is determined by the following equation.

The digital representation = (input voltage/5v) * $2^{n} - 1$

where n = 10, is the A/D resolution in bits.

After A/D conversion, the decimal signals are split into two bytes because at one time only 8 bits of data can be transferred via RS232 or serial port for communication. Since the signals are 10 bits data, they must be divided into two bytes where the two most significant bits (MSB) constitute the first byte and the 8 least significant bits (LSB) comprise the second byte. The action is done using MATLAB/Simulink byte conversion function. In byte utility, the data are converted from decimal numbers into 8 byte ASCII characters. An example of ASCII character table is shown in Fig. 7. This is done automatically before the data signals are sent via UART serial communication.

Fig. 7 ASCII table



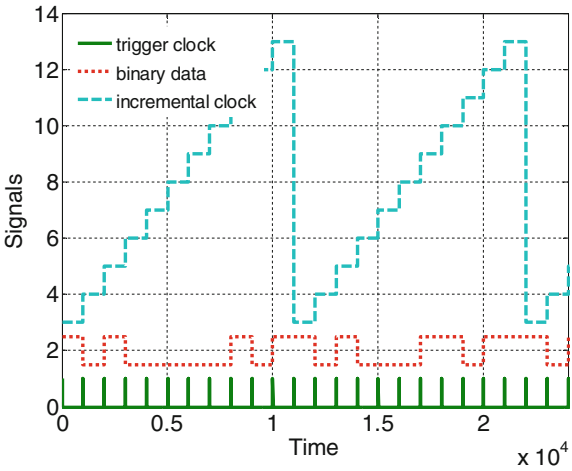
2.1.6 Signal Serial Transmission and Baud Rate

The baud rate of the RS 232 or serial communication determines the speed of data transmission. It can be set within the range of 9600 bits per seconds (bps) to 115,200 bps. A high baud rate improves communication speed, but a lower baud rate guarantees data integrity. An example of data being transmitted using 11 bits serial data with ASCII is shown in Fig. 8.

2.1.7 Data String Protocol

A data string protocol is implemented to avoid data corruption due to noise. The data string consists of a header, a sequence of data and a terminator. The header is a

Fig. 8 Serial 11 bits binary data transfer



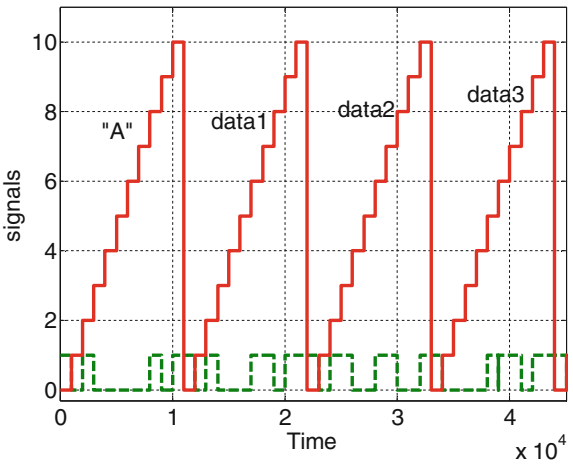


Fig. 9 Data protocol

character "A" and the terminator is a character ('/n'). The overall data sequence will be as follow

A, data1, data2, data3, data4……, /n (Fig. 9).

2.1.8 Hardware in the Loop Processing Time

The processing time is defined as the total time taken to process a batch of data. It starts with the time taken by MATLAB/Simulink to read signals from FQDC and the sensors and then send these signals to COC and IFC to be processed. It also includes the time taken by IFC to transmit output data (IGBT PWM firing and chopper operation mode signals) back to MATLAB/Simulink plus the implementation of the received data on the FQDC and EV. Finally, it adds the delay time before new data are gathered and transmitted again. The sum is called the processing time. In Fig. 10, an example of one cycle of processing time for MATLAB/

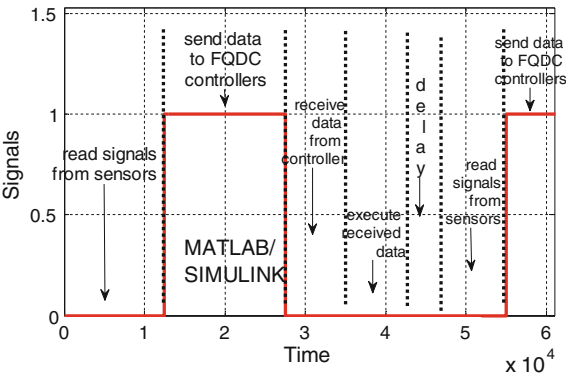


Fig. 10 Processing time of MATLAB/Simulink

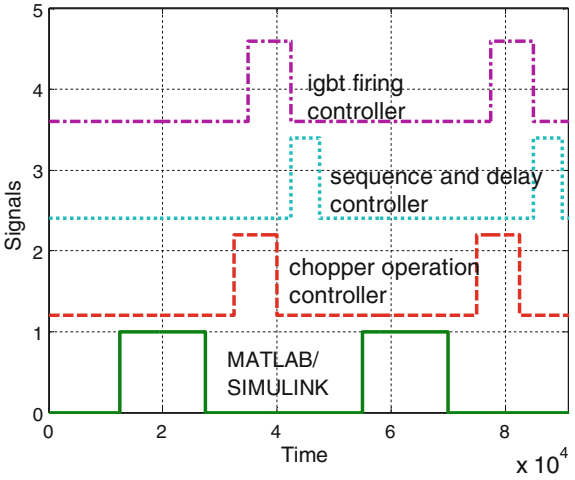


Fig. 11 Data pattern and processing time of each controller

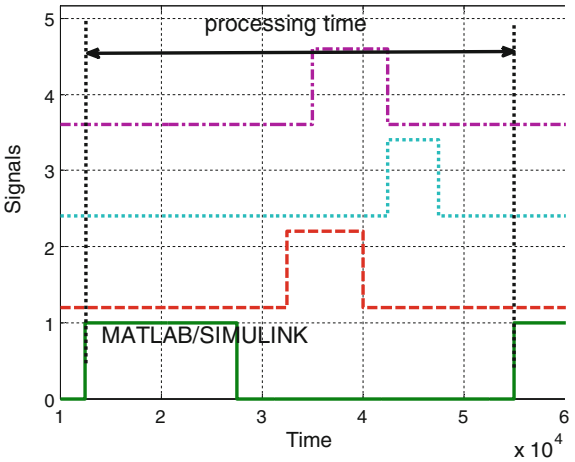


Fig. 12 Processing time in zoom

Simulink with its tasks is shown. In Fig. 11, the processing time of the whole system including Matlab/Simulink and the FQDC controllers is shown.

Figure 12 shows one cycle processing time for MATLAB/Simulink under magnification.

PIC 16F microcontrollers can operate at 20 MHz while PIC 18 microcontrollers can operate at speed of sixty(60) Mhz. The program execution time is calculated by dividing the cycle counts with the clock rate. The results show that the program execution times for the two microcontrollers are 9 and 14 microseconds(μs) respectively. The processing time also includes the controller's execution time (run by PIC microcontrollers at 20 MHz) and data transfer time by RS-232 serial communication at 9600 b/s (chosen for low error rate). For data transmission,

all signals require $11 \times 10 \times 3 = 330$ bits, plus a header and terminator of $2 \times 2 \times 11 = 44$ bits. So in total 374 bits are required. If we divide the total bits by the baud rate, 374/9600, it is equal to 0.038 s (s). This is the time it takes for the serial data to be transmitted and received. If reading an analog channel requires one (1) milliseconds (ms), reading eight (8) A/D channels require eight milliseconds (8 ms). Adding a delay of ten milliseconds (10 ms), the total processing time for MATLAB/Simulink and FQDC controllers which include processing, execution and data transfer can be up to 0.075 s.

2.1.9 Processor in the Loop Test Experiment

The main objective of processor in the loop (PIL) test is to test whether the controllers can handle the data and work as the FQDC. The COC must be able to choose the expected operation mode, the SAD controller can handle the transition from one chopper operation to another without causing torque and current spikes during transition and the IFC can work as DTC and PID controller. For this purpose the system has to pass the preset earth profile test. There are a few different earth profiles that the EV can be tested on such as flat driving, going downhill and uphill as shown in Fig. 13 [19]. The number 1 driving profile is chosen because it can test all FQDC modes of chopper operation.

2.1.10 Accelerator and Brakes Signals

The accelerator pedal and brake pedal signals must be provided so that a full data set is available. In this test, all FQDC chopper modes are tested. The expected accelerator and brake signals are shown in Figs. 14 and 15 respectively.

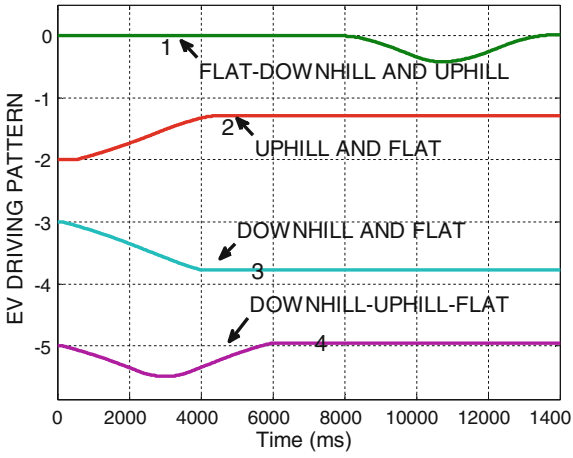


Fig. 13 Driving profiles

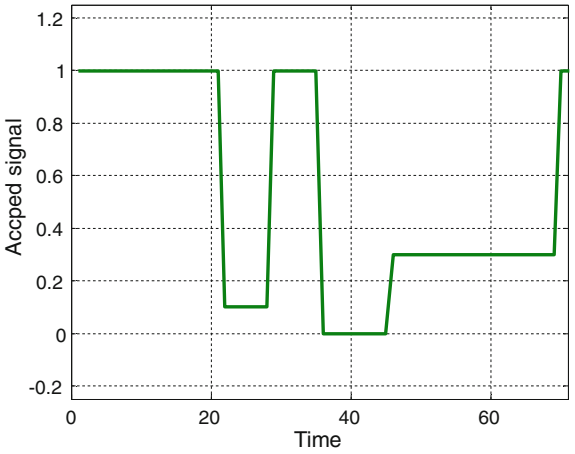


Fig. 14 Accelerator pedal signal

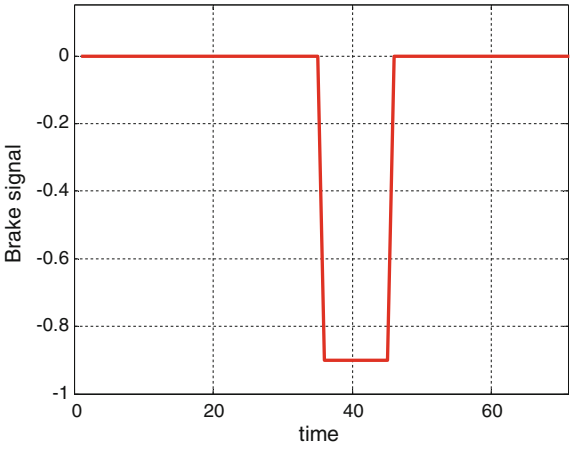


Fig. 15 Brake signal

With the chosen test profile and the accelerator and brake pedal signals, the controller is expected to engage all modes of the FQDC which are driving, field weakening, generator regenerative braking, resistive braking and parallel modes.

Figure 17 shows the block diagram of the processor in the loop(PIL) in MATLAB/Simulink model, and Fig. 16a its hardware setup. In the model, the EV uses a thirty five kilowatt(35 kW) motor operating at the maximum power of twenty kilowatt(22 kW) to drive its total weight of a thousand three hundred and thirty five kilogram (1325 kg). The time interval to update, process, send and receive data signals from MATLAB/Simulink to FQDC controllers and vice versa is in every seventy five milliseconds (75 ms). To speed up MATLAB/Simulink simulation, the model is run in accelerator or rapid accelerator mode.

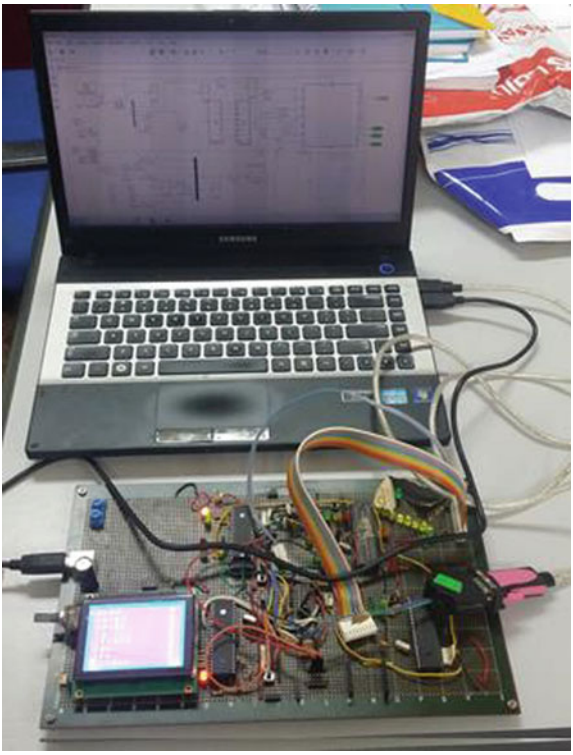


Fig. 16 Processor in the loop hardware setup

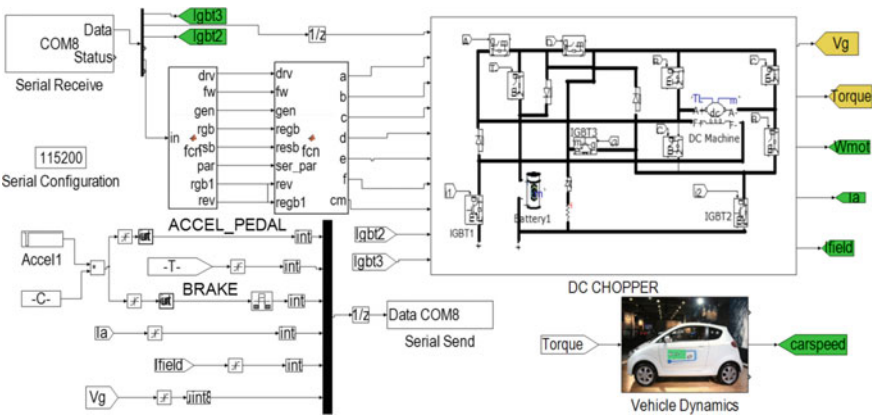


Fig. 17 Matlab/Simulink simulation model for processor in the loop

3 Result

The system is tested according to the signals shown in Fig. 18. At start up, the chopper operation controller(COC) selects driving mode. After building enough speed, the chopper operation controller(COC) selects field weakening mode since the accelerator signal is still high. However, when the accelerator signal decreases, generator mode is selected to save excess energy in the battery. In contrast, when the accelerator signal and the speed are high, field weakening mode is engaged. When brake command is triggered but the vehicle speed is high, regenerative mode is activated. As the vehicle speed drops, resistive braking mode is deployed. When brake command is replaced with low driving command, while the vehicle speed is low, the chopper operation controller(COC) selects driving mode again. As the vehicle moves downhill with a low driving command, the COC selects generator mode to save excess kinetic energy. However, if the driving command is high, the controller selects field weakening mode. As the vehicle climbs a steep hill, the speed is expected to drop. Hence, parallel mode is selected. Finally, as the vehicle regains its speed, field weakening mode is selected.

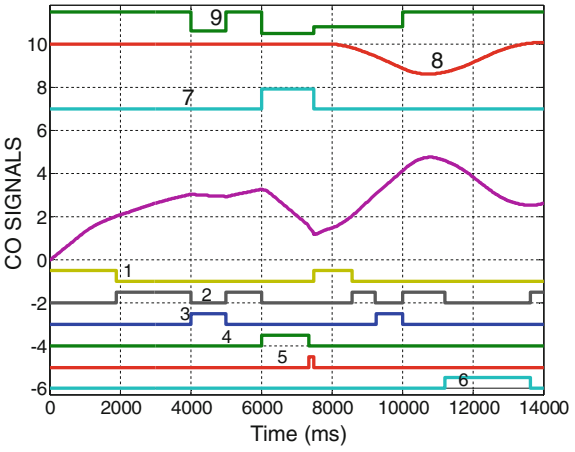


Fig. 18 Result of FQDC operation modes

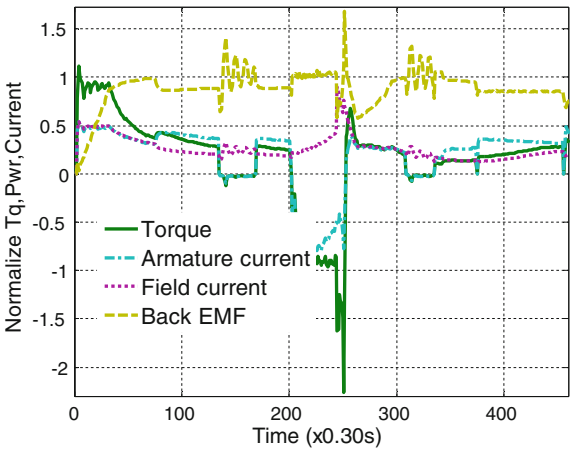


Fig. 19 FQDC performance

Figure 19 shows the output signals of the direct current (DC) series motor, the four quadrants drive chopper (FQDC), and the electric car (EC) in the forms of torque, armature current, field current and back electromagnetic force(EMF). In the figure, two important parameters to be observed are current and torque. They indicate the performance of the subsequent and delay (SAD) and igbt firing (IFC) controllers. The constant currents and torque at the beginning indicates that the IGBT firing controller works well with the PID and DTC control algorithms. No truncated torque and current are detected during chopper mode transition. The smooth transition indicates that the SAD controller and its algorithm functions well. At the beginning, under driving mode, the armature current, field current and torque are almost constant. Then these start to decrease as the back electro magnetive force (emf) increases along with speed. The back electro magnetize force (EMF) increases from zero to its maximum value as the motor reaches its maximum speed. In the field weakening mode there is a slight increase in torque due to the increase in armature current, but the field current decreases and consequently cuts the back electro magnetive force(emf). In generator mode the torque reduces slightly which results in an increase in back electro magnetive force (EMF). In regenerative and resistive braking the torque is negative due to the negative armature current. The back EMF is constant in regenerative braking but reaches its maximum value at the end of resistive braking. This is due to the rise of armature current and demand on braking torque. In parallel mode the torque increases slightly to overcome the steep climbing effect but the back EMF drops to produce a higher armature current.

4 Conclusion

PIL provides a simple method to test the FQDC that controls a series DC motor that propels an EV. It is very suitable when the input and output signals are in digital (binary) form. The capability of the controllers of the FQDC and its control algorithms were demonstrated when the COC managed to select the most appropriate modes of operation smoothly. During mode transition, the SAD was able to provide sufficient delay so the operation mode could change without producing spikes of current or torque. The IFC was able to execute all commands well to achieve the desired speed and torque using DTC and digital PID algorithms. With an appropriate tuning of the controllers, the EV performance can be further optimized. All six modes of FQDC were successfully tested according to the earth profile, and accelerator and brake pedal signals.

References

1. Gao, Y., Ehsani, M.: Design and control methodology of plug-in hybrid electric vehicles. IEEE Trans. Ind. Electron **57**(2), 633–640 (2010)
2. Oak Ridge National Laboratory.: Advanced Brush Technology for DC Motors (2009). Available: http://peemrc.ornl.gov/projects/emdc3.jpg
3. Heinrich, Walter Rentsch, Herbert Dr.-Ing, ABB Industry: "Guide to Variable Speed Drives," Technical Guide No. 41180 D-68619 LAMPERTHEIM, Germany, 3ADW 000 059 R0201 REV B (02.01), DCS 400/DCS 500/DCS 600: ABB 2003
4. Bansal, R.C.: Birla Institute of Technology and Science, Pilani, India. Electric Vehicle, Handbook of Automotive Power Electronics and Motor Drives, Taylor & Francise Group CRC Press (2005)
5. Rashid, M.H.: Power Electronics, Circuits, Devices and Applications, 3rd edn. Prentice Hall (2004)
6. Arof, S., Yaakob, N.M., Jalil, J.A., Mawby, P.A., Arof, H.: Series motor four quadrants drive DC chopper, Part 1: Overall. In: International Conference on Power Electronics (2014)
7. Arof, S., Hassan, H., Rosyidi, M., Mawby, P.A., Arof, H.: Implementation of series motor four quadrants drive DC chopper for DC drive electric car and LRT via simulation model. J. Appl. Environ. Biol. Sci. **7**(3S), 73–82 (2017)
8. Tipsumanporn, V., Thepsathorn, P., Piyarat, W., Numsomran, A., Bun jungjit, S.: 4—Quadrant DC Motor drive Control By BRM Technique. Warwick University—IEEE. Proceedings IPEMC 2000. Third International Power Electronics and Motion Control Conference (IEEE Cat. No.00EX435) (2002)
9. Amjadi, Z., Williamson, S.S.: Power-electronics-based solutions for plug-in hybrid electric vehicle energy storage and management systems. IEEE Trans. Ind. Electron. **57**(2), 608–616 (2010)
10. Ghao, D.W., Mi, C., Emadi, A.: Modelling and simulation of electric and hybrid vehicles. Proc. IEEE **95**(4), 729–745 (2007)
11. Higashikawa, K., Tajima, M., Urasaki, S., Inoue, M., Fukumoto, Y., Tomita, M., Kiss, T.: Hardware-in-the-loop simulation on fault current limiting operation of RE-123 coated conductors under the influence of spatial inhomogeneity. IEEE Trans. Appl. Supercond. **28**(4) (2018)

12. Mina, J., Flores, Z., López, E., Pérez, A., Calleja, J.-H.: Processor-in-the-loop and hardware-in-the-loop simulation of electric systems based in FPGA. In: 2016 13th International Conference on Power Electronics (CIEP) (2016). https://doi.org/10.1109/CIEP.2016.7530751
13. Ruba, M., Hunor, N., Hedesiu, H., Martis, C.: FPGA based processor in the loop analysis of variable reluctance machine with speed control. In: 2016 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR) (2016). https://doi.org/10.1109/AQTR.2016.7501375
14. Lee, S., Bang, H., Lee, D.: Predictive ground collision avoidance system for UAV applications: PGCAS design for fixed-wing UAVs and processor in the loop simulation. In: International Conference on Unmanned Aircraft Systems (ICUAS), IEEE (2016). https://doi.org/10.1109/ICUAS.2016.7502561
15. Arof, S., Muhd Khairulzaman, A.K., Jalil, J.A., Arof, H., Mawby, P.A.: Self tuning fuzzy logic controlling chopper operation of four quadrants drive DC chopper for low cost electric vehicle. In: 6th International Conference on Intelligent Systems, Modeling and Simulation, IEEE Computer Society, pp. 40–24 (2015). https://doi.org/10.1109/isms.2015.34
16. Arof, S., Muhd Khairulzaman, A.K., Jalil, J.A., Arof, H., Mawby, P.A.: Artificial intelligence controlling chopper operation of four quadrants drive DC Chopper for low cost electric vehicle. Int. J. Simul. Sci. Technol. **16**(4), 3.1–3.10 (2015). https://doi.org/10.5013/ijssst.a.16.04.03
17. Ching, T.W.: Soft-switching converters for EV propulsion. J. Asian Electric Veh. **5**(2) (2007)
18. Arof, S. Jalil, J.A., Kamaruddin, N.H., Yaakop, N.M., Mawby, P.A., Arof, H.: Series motor four quadrants drive DC chopper Part 2: driving and reverse with direct current control, pp. 775–780. In: International Conference on Power Electronics. https://doi.org/10.1109/pecon.2016.7951663 (978-1-5090-2547-3/16)
19. Arof, S., Noor, N.M., Elias, F., Mawby, P.A., Arof, H.: Investigation of chopper operation of series motor four quadrants DC Chopper. J. Appl. Environ. Biol. Sci. **7**(3), 49–56 (2017)