Fault Tolerant BLDC Motor Control for Hall Sensors Failure

Sova V., Chalupa J., Grepl R.
Mechatronics laboratory (www.mechlab.cz)
Brno University of Technology, Faculty of Mechanical Engineering
Brno, Czech Republic
sova@fme.vutbr.cz

***Abstract*—In most brushless direct current (BLDC) motor drives, there are three hall sensors as a position reference. This paper presents a method that allows the operation of a BLDC motor with one faulty hall sensor. The situations considered are when the output from a hall sensor stays continuously at low or high levels, or a short-time pulse appears on a hall sensor signal. For fault detection, identification of a faulty signal and generating a substitute signal, this method only needs the information from the hall sensors. So this method can be added as a standalone subsystem to existing drives. The paper provides many simulations on how the presented method reacts for various types of faults. An experimental evaluation is provided too. Due to the demand to operate at a high speed, the method is implemented into the FPGA.**

Keywords- fault detection; brushless motor; BLDC

I. Introduction

Brushless direct current (BLDC) motors are becoming more popular. Their advantages are: a simple control mechanism, power density and long service life. They are increasingly used in different industrial and commercial applications. They are also used in applications where the failure of a BLDC motor drive can lead to big economic loses or even safety threats, e.g. in the automotive [1][2] or aerospace [3][4] industries. An example could be a turboprop engine fuel pump powered by a BLDC motor. There is a demand for a high degree of reliability. In case of fault, the running fuel pump must remain functional, but the start-up of a pump is not necessary [5].

Many parts in a BLDC motor drive can malfunction. For example: the open-circuit or short-circuit fault of a switch, voltage or current sensor fault in a power inverter. Motor faults: the inter-turn short circuit in winding, overheating, bearing failure, phase open-circuit fault, etc. And position sensors are subjected to faults too. In addition to these faults, every part and signal of a BLDC motor drive is subjected to interferences.

On almost all BLDC motors, three hall sensors shifted by 120° el. (electrical degrees) are used as a position reference for control algorithm [6]. Our proposed method tries to preserve the functionality of a BLDC motor drive during a hall sensor fault. It should preserve the functionality of a BLDC motor drive during a long-term fault of a hall sensor or even during short-term pulses on hall sensor signals caused mostly by interferences.

Fault detection and controlling various systems during faults is the current subject of scientific research. In areas around BLDC motors new contributions still arise. Jeong et al. [7] deals with various faults in interior permanent magnet motor drives. In the case of a position sensor fault, they very briefly say, it can be detected when the estimated rotor position differs from the expected and in that case, the algorithm switches to a sensorless control scheme. The faults in a power inverter and remedial strategies for that case are in [8][9]. Especially for hall sensor faults [10][11] are dedicated.

Tashakori and Ektesabi [10] only address the situation when a hall sensor stays permanently at a low or high level. The proposed algorithm is divided into three parts: fault detection, identification of a faulty signal and generating a substitute signal. The time between a hall sensor going wrong and a substitute signal being used is quite long.

Firmansyah et al. [11] check the sequence of hall signals states, which are defined by hall sensor signals. The transition to a following state is allowed only if the change of hall sensor signal defines a following state and if the change happens within an expected time.

II. Method Proposal

There was a demand to use only hall sensors signals, so a fault tolerant BLDC motor control algorithm based only on checking hall sensor signals was created. The algorithm can be implemented in a device which can form a standalone unit in between a motor and an ECU (Fig. 1). Obviously the algorithm can be implemented into the existing ECU.

In Fig. 2 there is a block diagram illustrating the algorithm functionality. There are three main building blocks in the algorithm. The blocks shown in red are responsible for generating a substitute signal from two fine signals.

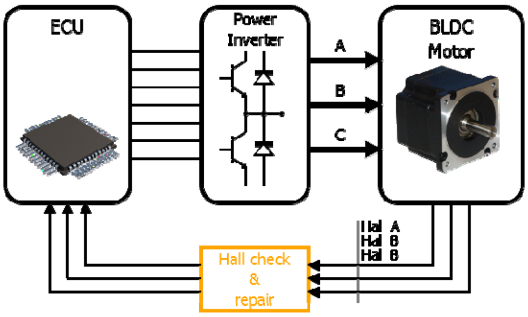

Figure 1. Location of proposed algorithm in a BLDC motor drive

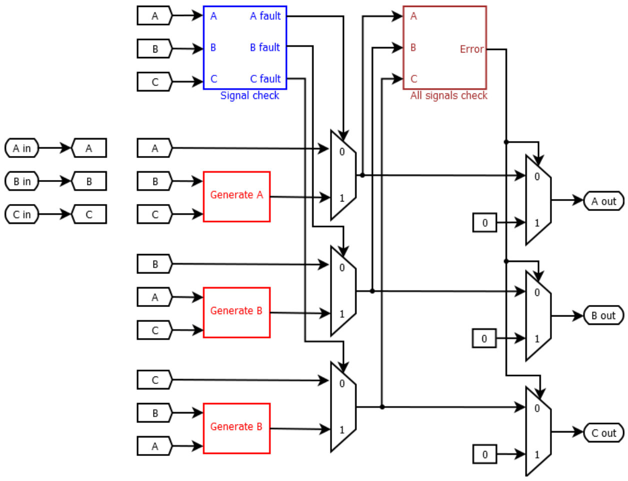

Figure 2. Overall block diagram of proposed algorithm

There are three independent generating blocks in this algorithm and they are generating the substitute signals continuously. Signal A is generated using signals B and C, signal B is generated using A and C, and C is generated using A and B.

The next important block in the algorithm is the block called the Signal check. The Signal check block is responsible for recognizing a faulty signal. When a faulty signal is detected, then this block sets the corresponding error output to the high level and the faulty signal is replaced by its substitute.

The last important block in the algorithm is the block increasing the reliability, but in some cases it can be omitted. This block sets all the signals to the low level in case of a fault, before the faulty signal is switched to the substitute signal. Because the recognition of a faulty signal by the Signal check block is not immediate, there might occur a sequence on signals, which may lead to switching the wrong transistors in a power inverter, thus causing overcurrent. We assume that setting all the signals to a low level is a forbidden combination and the ECU switches all the transistors in a power inverter off. The time of leaving out is very short (less than rotating by 180° el.) and the rotor overcomes it by its inertia.

Because a lot of the parts of the algorithm are based on measuring time intervals in signals and generating signals, the proposed algorithm requires good resolution on the time axis and multiple simultaneous measurements, so the algorithm was targeted into the FPGA.

In accordance with the Rapid Control Prototyping technique, dSPACE [12] hardware as a target device and Xilinx System Generator for DSP™[13] as a high-level tool for generating the FPGA code were used.

The three main building blocks are described in more detail:

A. Generating a substitute signal from two fine signals

Signal waveforms from hall sensors are periodic, rectangular, with a phase shift of 120° (Fig. 3). Edges in hall sensor signals indicate the time of commutation for a motor.

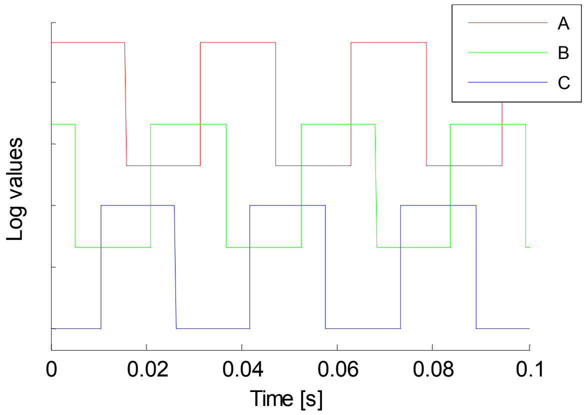

Figure 3. Waveforms of hall sensor signals

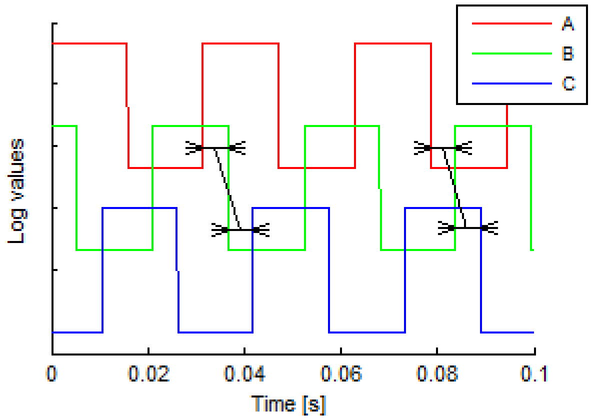

Figure 4. Generating a substitute signal according to two fine signals

Theoretically it should be able to generate two other signals from one functioning signal. If the signal period can be measured, divided by 3 (a phase shift of 120°) and this time measured out from the signal edges, it should be possible to generate two other signals from one functioning signal. In reality, it turns out that this approach is unacceptable. Thanks to manufacturing inaccuracies in hall sensor mount positions and inaccuracies in the sensing ring for hall sensors, the resulting signal edges come too far from the actual moment of commutation. This results in increased noise, bigger vibrations and worse efficiency. The substitute signals can be generated from an average time from a few last periods of the signal. This improves the behaviour in a steady state, but leads to worse transient states.

So the focus was kept on generating one signal from two others, and using the most recent time information, not averaging. The time interval between the falling edge of one signal and rising edge of a second signal was measured. This time we measured out from the rising edge of the second signal and then the falling edge of the generated signal was made. And similarly for the second edge (Fig. 4).

FPGA implementation of this building block is in Fig. 5. The upper part of the diagram is responsible for setting the generated signal to the low level and the bottom part for setting it to the high level. The upper part is described in more detail: The falling edge of signal A resets a counter which increments with every FPGA clock tick. The rising edge of signal B stores the value of the first counter and resets the second counter. When the value of the second counter equals the stored value from the first counter, the generated signal is set to the low level.

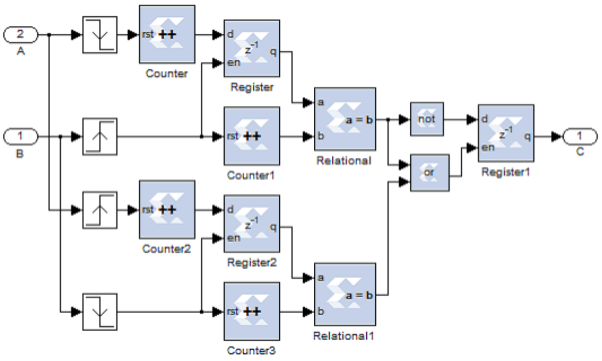

Figure 5. FPGA implementation of Generating block algorithm

B. *Signal check - recognizing the faulty signal*

There exist many options for how to recognize a faulty signal. The faulty signal can be recognized by the fact that the change in hall signals is expected at a certain time [11]. This method fails during fast transients because the expected time of a signal change differs from the actual one. In [10] they recognize the faulty signal by analyzing the waveforms of the terminal voltages. Even sensorless BLDC motor control methods can be used for recognizing the faulty signal [7].

In our proposed method, the sequence of signal changes is checked to recognize the faulty signal. This is primarily due to the fact that it only needs the signals from the hall sensors. Also good behaviour during transient states can be expected, because the time information of the signals is not needed.

The sequences of signal changes are checked for three pairs of hall signals (AB, BC, AC). In the case of an individual signal fault, two pairs report the fault and we know that the faulty signal is the one which is common for both pairs.

For each signal pair, during the positive direction of rotation, the repeating sequence 00, 10, 10, 11, 01, 01 (Table 1) can be seen. It is checked to see whether this sequence is kept and, if not, the algorithm reports a fault for the signal pair.

TABLE I. HALL SENSORS STATES DURING ONE ELECTRICAL REVOLUTION

State number	□ [°el.]	A	B	C
1.	0-60	0	1	0
2.	60-120	0	1	1
3.	120-180	0	0	1
4.	180-240	1	0	1
5.	240-300	1	0	0
6.	300-360	1	1	0

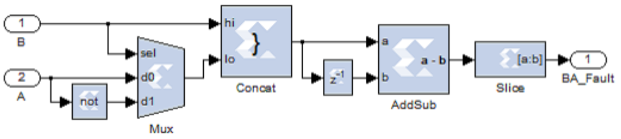

Figure 6. FPGA implementation of signal pair check

The FPGA implementation of this algorithm is in Fig. 6. The sequence 00, 10, 11, 01 (repeating states are omitted) is transformed to sequence 00, 01, 10, 11, which represents binary counting by inverting the second signal when the first one is at the high level. Then these two signals are concatenated into one two-bit signal and the delayed sample of this signal is subtracted from the current sample. If the result of subtracting is 0 or 1, the change of the signal state was correct, otherwise it was incorrect. Taking the first bit (counting from zero) from the results, the fault indicator signal is obtained.

The fault indicator signal indicates a problem in one or both signals. To identify the specific signal, every pair of the fault indicator signals is carried to AND gates. At the end of this part which recognizes the faulty signal, there are registers in which the values of individual signal fault indicators are stored until a new edge on the signals occurs.

C. *All signals check*

This block is based on checking the sequence of signal changes on all three signals. If the change of the state number is directly following the previous state, we know that it's all right, otherwise there is a fault. A look-up table for recognizing the state number was implemented into the FPGA. Individual hall signals were concatenated into one 3-bit signal and this signal is used to choose the state number from the block which behaves like a ROM memory. In the ROM memory block is information from table 1. The state number is then subtracted from the delayed one and the result is checked to see whether it is within the expected range. The error signal is stored in a register and can change only if there is a change in some signal from the hall sensors.

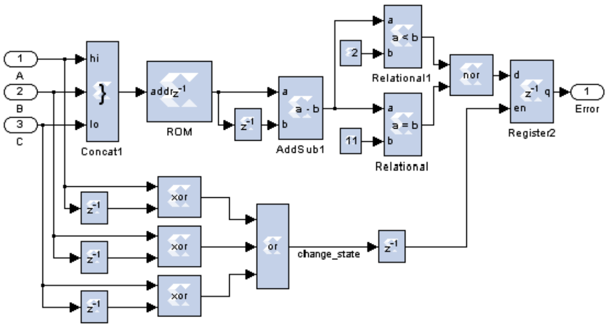

Figure 7. FPGA implementation of all signals check block

III. SIMULATION RESULTS

First of all, the behaviour of the proposed method under various conditions was simulated. The focus was kept on situations where the signal from the hall sensor stays permanently at the low or high level and when there is a short time pulse on the signal. These errors were simulated for the motor in steady state conditions and also during the changes of rotational speed.

A. Permanent low or high fault

When one of the signals stays permanently at the low or high level, a very short moment when all signals are at the low level occurs. This suspends transistors switching in a power inverter, thus preventing them from wrong switching, which may lead to overcurrent. After this short leave out moment, the substitute signal is used instead of the faulty one.

In Fig. 8 the leave out time (all signals at the low level) is very short. It is represented only by a separate peak at time 0.018s. Generally the leave out time can last between 0-180˚el.

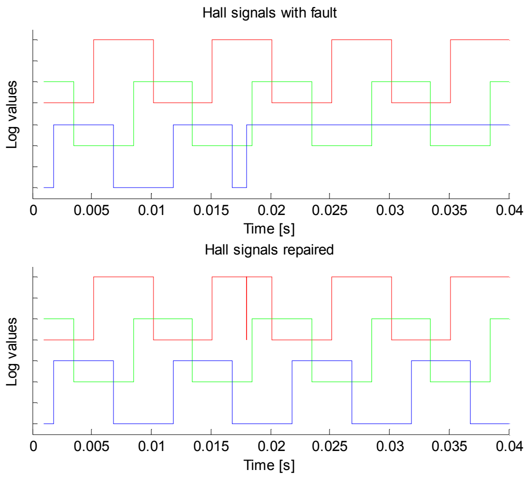

Figure 8. Hall signals with a permanent high fault

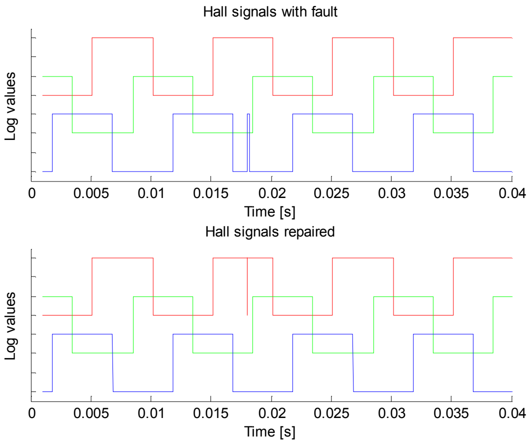

Figure 9. Hall signals with a short time pulse fault

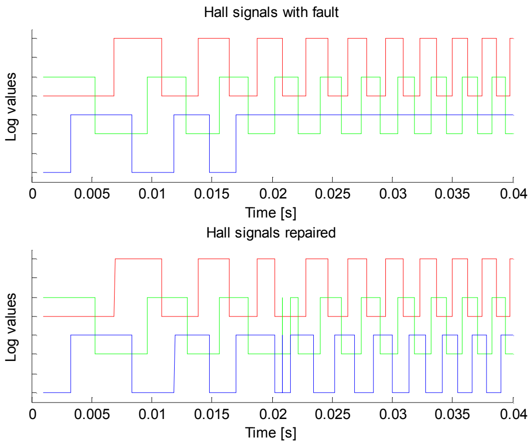

Figure 10. Hall signals with a permanent high fault during motor acceleration

B. Short time pulse fault

In the simulation for a short time pulse on otherwise fine signals (typically caused by interferences), a situation where all signals are at the low level for safety reasons appears for a brief moment, until the previous parts of the algorithm react correctly. For the situation in Fig. 9, the time where all the signals are at the low level is very short and almost immediately it is switched to the simulated signal. After the pulse is out and the original signals are fine, the simulated signal is replaced by the original one.

C. Permanent low fault during acceleration

There is no problem to repair a faulty signal even during acceleration of the motor. As usual a brief moment appears when all the signals are at the low level and after that the substitute signal is used (Fig. 10). The substitute signal is generated using the most recent edges in signals, without filtering, so the missing signal is substituted quite well.

IV. EXPERIMENTAL EVALUATION

Various simulations showed good results, so the experimental evaluation followed. The algorithm was tested on our laboratory BLDC motor test bench, which consists of a BLDC motor [15], torque meter and a DC motor which acts as a load. The power inverter is a Microchip MC1L [16]. The proposed algorithm is implemented on the modular dSPACE Hardware with a DS5203 FPGA board [17].

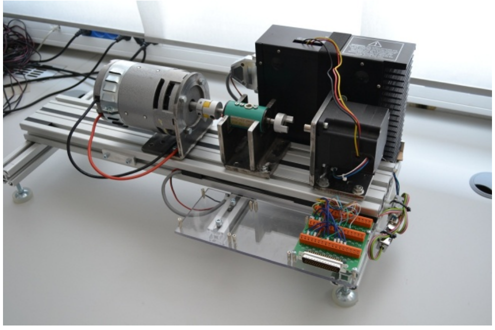

Figure 11. BLDC motor test bench available in the Mechatronics laboratory [14]

For the experimental evaluation, besides the proposed algorithm, a simple controller for the BLDC motor and an algorithm for simulating faults were implemented on the dSPACE FPGA board (Fig. 12). The controller of the BLDC motor was represented only by a simple commutation table, which assigns transistor switching commands for every combination of hall sensor signals.

To manage the experiment, the control GUI in ControlDesk® [18] that allowed us to switch between normal mode and safe mode and simulate long-term and short-term errors on hall sensor signals was created.

A. Permanent low or high fault

In a time of 55ms a high level fault on signal A was simulated. From the plots in Fig. 13, we can see that at the moment of the fault, the current starts to grow until a leave-out time occurs. A moment later a substitute signal was generated.

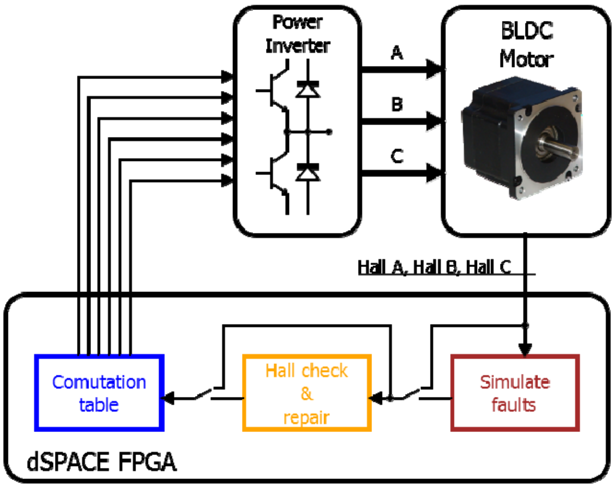

Figure 12. Setup for experimental evaluation

TABLE II. IMPORTANT PARAMETERS OF BLDC MOTOR B8672-48 [15]

Nominal voltage	48 V
Nominal power	117 W
No load speed	3700 rpm
Nominal speed	3102 rpm
Nominal torque	0.359 Nm
Nominal current	19.4 A
Number of pole pairs	4

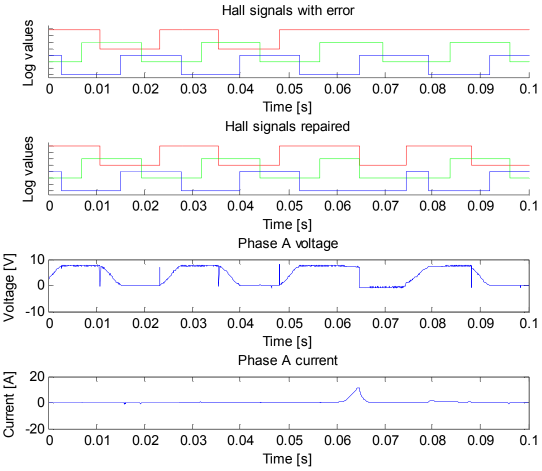

Figure 13. Hall signals, phase A terminal voltage and current under a permanent high fault

B. Short time pulse fault

Short-time pulses lasting 3 ms were generated with a period of 40 ms. In Fig. 14 we can see that the first pulse was immediately repaired, even without a leave-out time. The second pulse caused a short moment where all the signals are at the low level.

We can see that the presented waveforms are in accordance with the simulation results.

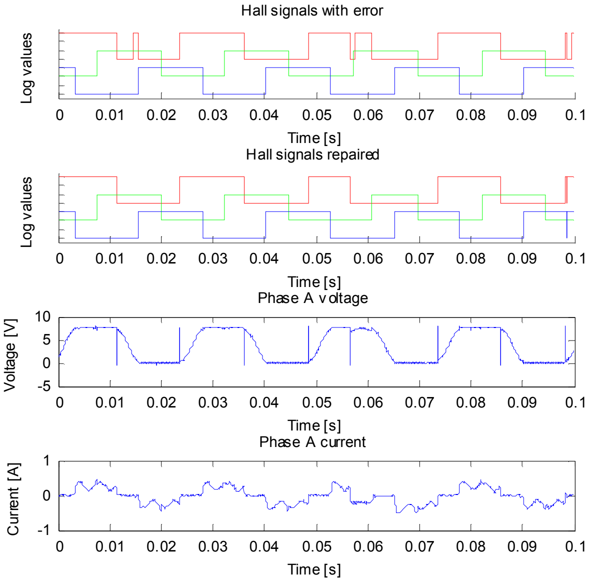

Figure 14. Hall signals, phase A terminal voltage and current waveforms under a pulse fault

C. Permanent low fault during acceleration

The behavior when a fault happens during acceleration was investigated too. The presented waveforms are in accordance with the simulation results.

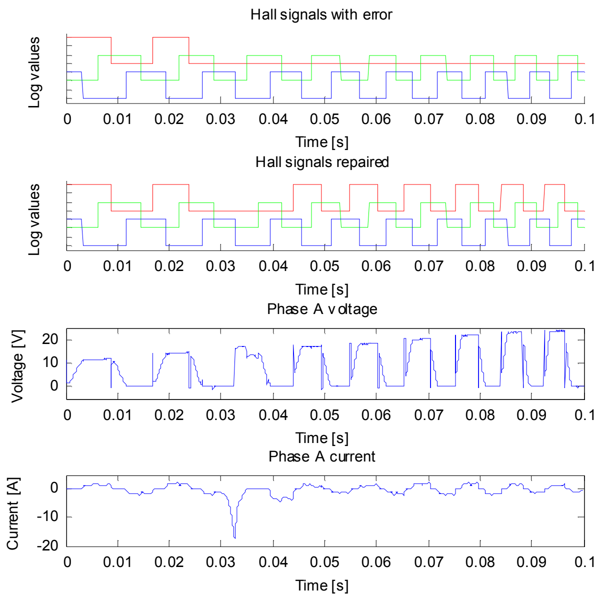

Figure 15. Hall signals, phase A terminal voltage and current under a permanent low fault during acceleration

V. Conclusion

The algorithm that allows the operation of a BLDC motor drive with one faulty hall position sensor was presented. Faults when the hall sensor stays permanently at the low or high level, or a short-time pulse appears on a hall sensor signal, are repaired. The algorithm was verified on numerous simulations and in a real experiment too. FPGA was used for implementation, so the reaction time is very short and the motor speed is almost unlimited. The presented algorithm does not allow start-up or near zero speed operation of a motor, because it is impossible to measure the pulse length on hall sensors signals when the motor in not rotating. But once the motor is rotating, the functionality during a fault is maintained. Plenty of methods exist on how to maintain an open-loop start-up of a BLDC motor. The proposed algorithm only needs the information from the hall sensors so it can be implemented into an existing solution as a standalone subsystem. One of the possible ways to improve the proposed method is to merge our algorithm for recognizing a faulty signal with an algorithm that checks the time when a signal change occurs [11].

Acknowledgment

This work was supported by the European Commission within the FP7 project Efficient Systems and Propulsion for Small Aircraft “ESPOSA”, contract No. ACP1-GA-2011-284859-ESPOSA, and by NETME CENTRE PLUS (LO1202) created with financial support from the Ministry of Education, Youth and Sports under the „National Sustainability Programme I“.

References

[1] Jianwen Shao; Nolan, D.; Teissier, M.; Swanson, D., "A novel sensorless brushless DC (BLDC) motor drive for automotive fuel pumps," Power Electronics in Transportation, 2002 , vol., no., pp.53,59, 24-25 Oct. 2002

[2] Jun-Hyuk Choi; Se-Hyun You; Jin Hur; Ha-Gyeong Sung, "The Design and Fabrication of BLDC Motor and Drive for 42V Automotive Applications," Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on , vol., no., pp.1086,1091, 4-7 June 2007

[3] Laxminarayana, Y.; Tarakalyani; Ravindranath, "“Design of a compact BLDC motor for high power, high bandwidth Rotary ElectroMechanical Actuator for aerospace application”," India Conference (INDICON), 2011 Annual IEEE , vol., no., pp.1,4, 16-18 Dec. 2011

[4] Banerjee, T.P.; Roychoudhury, J.; Das, S.; Abraham, A., "Hybrid Intelligent Predictive Control System for High Speed BLDC Motor in Aerospace Application," Emerging Trends in Engineering and Technology (ICETET), 2010 3rd International Conference on , vol., no., pp.258,262, 19-21 Nov. 2010

[5] Andrs, O.; Hadas, Z.; Kovar, J., "Introduction to design of speed controller for fuel pump," Mechatronics - Mechatronika (ME), 2014 16th International Conference on , vol., no., pp.672,676, 3-5 Dec. 2014

[6] Elvich L.N., 3-Phase BLDC Motor Control with Hall Sensors Using 56800/E Digital Signal Controllers, URL: http://cache.freescale.com/files/product/doc/AN1916.pdf, June 2015

[7] Yu-Seok Jeong; Seung-Ki Sul; Schulz, S.E.; Patel, N.R., "Fault detection and fault-tolerant control of interior permanent-magnet motor drive system for electric vehicle," Industry Applications, IEEE Transactions on , vol.41, no.1, pp.46,51, Jan.-Feb. 2005

[8] Byoung-Gun Park; Tae-Sung Kim; Ji-Su Ryu; Dong-seok Hyun, "Fault Tolerant Strategies for BLDC Motor Drives under Switch Faults," Industry Applications Conference, 2006. 41st IAS Annual Meeting. Conference Record of the 2006 IEEE , vol.4, no., pp.1637,1641, 8-12 Oct. 2006

[9] Tashakori, A.; Ektesabi, M., "Fault diagnosis of in-wheel BLDC motor drive for electric vehicle application," Intelligent Vehicles Symposium (IV), 2013 IEEE , vol., no., pp.925,930, 23-26 June 2013

[10] Tashakori, A.; Ektesabi, M., "A simple fault tolerant control system for Hall Effect sensors failure of BLDC motor," Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on , vol., no., pp.1011,1016, 19-21 June 2013

[11] Firmansyah, E.; Wijaya, F.D.; Aditya, W.P.R.; Wicaksono, R., "Six-step commutation with round robin state machine to alleviate error in hall-effect-sensor reading for BLDC motor control," Electrical Engineering and Computer Science (ICEECS), 2014 International Conference on , vol., no., pp.251,253, 24-25 Nov. 2014

[12] dSPACE GmbH, URL: https://www.dspace.com, April 2015

[13] System Generator for DSP, Xilinx Inc., URL: http://www.xilinx.com/products/design-tools/vivado/integration/sysgen.html, April 2015

[14] Mechatronics Laboratory, Faculty of Mechanical Engineering, Brno University of Technology, URL: http://mechlab.fme.vutbr.cz/, April 2015

[15] B86 series brushless dc motors, Transmotec Inc., URL: http://www.transmotec.com/brushless-dc-motors/no-gear/b86-series.aspx, April 2015

[16] MC1L 3-Phase Low Voltage Power Module Microchip Technology Inc, URL: http://ww1.microchip.com/downloads/en/DeviceDoc/70097A.pdf, April 2015

[17] DS5203 FPGA Board Overview, dSPACE GmbH, URL: https://www.dspace.com/en/pub/home/products/hw/modular_hardware_introduction/i_o_boards/ds5203_fpga_board.cfm, April 2015

[18] dSPACE ControlDesk Overview, dSPACE GmbH, URL: https://www.dspace.com/en/pub/home/products/sw/expsoft/controldesk.cfm, April 2015