

# BLDC motor control in HiL configuration with the use of Matlab/Simulink software and PLC

**Abstract.** The paper presents the solution uses the Matlab/Simulink environment and the PLC in Hardware in the Loop (HiL) configuration. It describes the advantages of rapid prototyping using simulation techniques, for example the HiL method. It also presents the construction of the system in HiL configuration with BLDC motor as the test object. Results presented in this paper relate to the three of proposed methods, using different ways to communicate between Matlab/Simulink and PLC, such as: OPC server; S-Function; M-file.

**Streszczenie.** W artykule przedstawiono zastosowanie oprogramowania Matlab/Simulink oraz sterownika PLC w konfiguracji pętli sprzętowej (HiL). Opisano zalety szybkiego prototypowania z wykorzystaniem technik symulacyjnych na przykładzie metody HiL. Przedstawiono również budowę systemu w konfiguracji HiL z silnikiem BLDC, jako obiektem badanym. Zamieszczone w artykule wyniki badań odnoszą się do trzech metod bazujących na różnym sposobie komunikacji między oprogramowaniem Matlab/Simulink, a sterownikiem PLC, takimi jak: serwer OPC; S-Function; M-plik. (Sterowanie silnikiem BLDC w pętli sprzętowej HiL z wykorzystaniem oprogramowania Matlab/Simulink i sterownika PLC).

**Keywords:** Hardware in the Loop, Matlab/Simulink, communication protocols, Programmable Logic Controller (PLC).

**Słowa kluczowe:** metoda symulacji HiL, oprogramowanie Matlab/Simulink, protokoły komunikacyjne, sterownik programowalny.

## Introduction

Construction of new machines or solutions has always been a great challenge for a project engineer. It sets a series of difficulties, starting from proposing a concept based on current knowledge, to creating a prototype of a particular project. It used to take countless amounts of hours and frequently turned out to be futile. The appearance of computers enabled quick development of designing techniques, as in many cases it significantly accelerated complicated calculations.

In the course of time computer assisted design evolved significantly, the result of which is what we know today as, for example, system modelling in CAD programmes and subjecting them to different types of tests [3].

The things that connect the times before and after the appearance of computers are two basic assumptions:

- The project and the prototype should be made in the shortest possible time.
- The project and the prototype should be made with the lowest possible cost.

Currently, both of these requirements may be fulfilled using quick prototyping by means of simulation techniques. One of such simulation techniques is Hardware in the loop (HiL). It enables connecting the modelled part of a project, e.g. a drive, with actual objects, such as measurement sensors. The application of such solution enables quick introduction of changes into the designed unit. It does not require from the constructor constant physical regulations, often on an expensive prototype.

The current solutions enable to test devices in the HiL configuration [4], but in most cases dedicated platforms are used for that purpose. The introduction of the PLC controller, as the element that intermediates communication between the control algorithm and the controlled object results in universality of such solution.

The study was made for three selected methods using different ways of communication between the software Matlab/Simulink and PLC controller.

## System concept in the HiL configuration

The concept of the system in HiL configuration was presented in Fig. 1. The BLDC motor was chosen as a research object [1, 2], and during tests a change of motor's speed was being made in the feedback for three selected methods of communication between the software Matlab/Simulink and PLC.

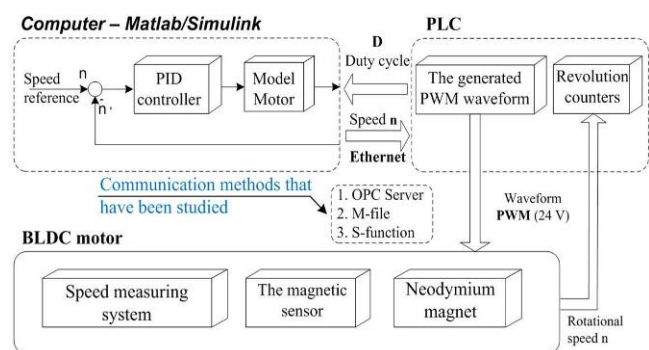


Fig. 1. Block diagram presenting the concept of the unit that enables to study possibilities of application of the software Matlab/Simulink and the PLC in HiL configuration

The system consists of three main parts:

- PC – it is used to do simulations of the PID controller in the Matlab/Simulink environment. The motor rotational speed is also given from the simulation level.
- PLC – it provides generation of the PWM signal and its transmission to the BLDC motor. The PLC also performs the function of counting pulses coming from the magnetic sensor, the result of which is estimation of motor's rotational speed.
- BLDC motor – it rotates with the speed given by the PLC. Its construction was enriched with the rotational speed measurement system.

## Construction of the system in HiL configuration

In the HiL system the product of the Saia Burgless Company was used as the PLC. Saia controller includes, common for all the studied methods, a program generating PWM signal, and counting of the BLDC motor rotational speed. The assembled measurement station is presented in Fig. 2.

The PLC as well as the magnetic sensor is powered by Saia Power module, which supplies the system with a direct voltage of the value of 24 V. The rotational speed of the motor is modulated by the PWM signal. The value of duty cycle is set by applying PID controller (on the basis of the readout of the actual rotational speed), which is implemented in Matlab/Simulink environment.

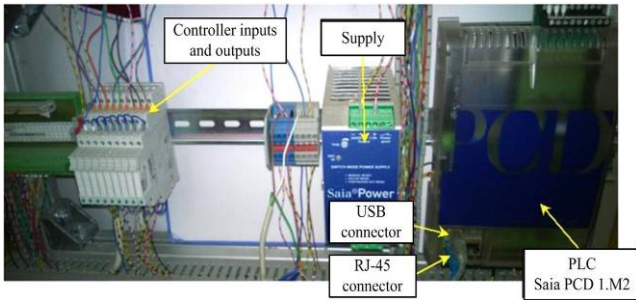


Fig.2. Measurement station equipped with the PLC

The measurement of rotational speed is done by the application of a magnetic sensor in the system, as well as mounting a miniature neodymium magnet on one of the shoulder, which is powered by the BLDC motor.

The PLC as an element of the control system intermediates in communication between the PID controller and the BLDC motor does the PWM modulation and, additionally, reads magnetic sensor pulses, thanks to which a quick assessment of the actual motor rotational speed value is possible.

The part connected with communication between the PLC and Matlab/Simulink environment undergoes modifications.

### Methods of communication

Data transmission between the PLC and the Matlab/Simulink environment was realized for three data transmission methods:

- OPC server – method using the OPC server as a medium that mediate in communication between Matlab/Simulink and the PLC.
- M-file – method involving on the use of Matlab functions to get data from PID and send them via the Modbus protocol.
- S-Function – method which uses applications developed in C++. This application is run from the Matlab/Simulink (as an element of implemented control model with PID controller) and provides communication with the PLC using one of the network protocols.

Each of these methods has been implemented on the test bench, and then for each method independently conducted a series of measurements.

### Method using OPC Server

OPC server is a software providing data in the standard OPC [5]. OPC Servers are widely use in the industry to collect information about the production process, and therefore it was decided to also carry out studies for this method in the configuration HiL. The OPC server is used as a medium mediating the communication between the PLC and the Matlab/Simulink. The use of such solutions requires changes in the loop the PID controller as well as installation of the appropriate software. PID controller with the function responsible for the speed reference is in Matlab/Simulink. Into the loop of PID controller have been added respective blocks which enable communication with the OPC server. These blocks are located in a dedicated library called OPC Toolbox. A properly configured OPC server receives data from the controller at the same time saving them to the appropriate registers PLC. It retrieves the data from the controller and gave them to the simulation process.

Communication between the PLC and the OPC server is via Ethernet S-Bus [6, 7]. For this purpose to communicate with the OPC server was used library OPC Toolbox, which

provides functions (in the form of blocks) enabling the exchange of data with the server:

- OPC Configuration – this block is necessary to realize the connection. When it is selected from the list should choose the appropriate server. In the project was selected server *CommServer*, which is dedicated to Ethernet S-Bus.
- OPC Read – this block is responsible for reading data from the OPC server. When it is selected, need to select which data should be retrieved.
- OPC Write – this block is responsible for writing data to the OPC server. When it is selected, need to select in which place on the server information should be recorded.

Communication with OPC server can be realized synchronously or asynchronously. In the project selected the synchronous communication what causes read and write at the same time.

Fig. 3 presents rotational speed waveform for different setting – a method using the OPC server. Visible strong overshoots are the result of errors in communication. Breaking the connection between the PLC and the server OPC resulted in longer runtime of the motor in the set speed.

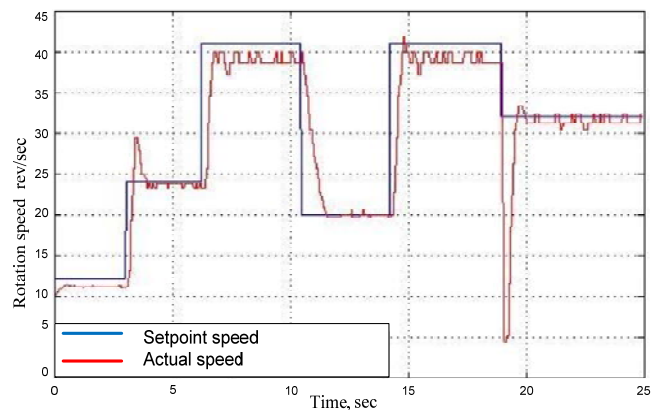


Fig. 3. Rotational speed waveform for different setting – the method using the OPC server

OPC server configuration significantly affects the quality of data transmission. Well-chosen time settings of the server they allow for continuity of communication mismatched cause a break in communications what is dangerous for the real system.

The value of the average time between successive packets S-Bus, registered with the help of Wireshark software is: 0,091922s (were averaged 20,000 samples).

Advantages of the method using the OPC server:

- Use a dedicated S-Bus protocol and a wide selection of other suppliers.
- Sharing by the Matlab/Simulink library of OPC Toolbox.
- Data collected by the OPC server can be used by other software in real time.

Disadvantages of the method using the OPC server:

- Critical to the operation of the system is the correct configuration of the OPC server.
- The relatively long time it takes for data exchange.

### Method using M-file

The Matlab software enables you to write scripts called M-files [8, 9], which may contain code implementing various types of logical operations as well as the code that implements the support of specialized software. M-files are

launched as separate processes. Therefore, it is possible to be run simultaneously the script code and simulation in Simulink software as well as it is possible to exchange data between the two operating processes. Method using M-file that was used in the project includes a developed script code enable communications with the PLC via Modbus TCP, and read and writes data to the PID controller in Simulink software and provides read and writes data to the PID controller in Simulink software. The PID and the block responsible for the speed reference are implemented in Matlab/Simulink environment.

The code of M-file, reads the output value of PID controller and then transmits via Modbus protocol. In the next step, gets the value of the speed of the PLC and transmits it to the PID controller.

Fig. 4 presents rotational speed waveform for different setting – a method using the M-file.

The value of the average time between successive packets Modbus, registered with the help of Wireshark software is: 0,027145219s (were averaged 26814 samples).

Sampling time simulation has been set to 0.02s. Averaged data include both question and answer for the Modbus protocol. It follows that the delay associated with the use of this protocol is approx. 2 samples, that is. 0.04s.

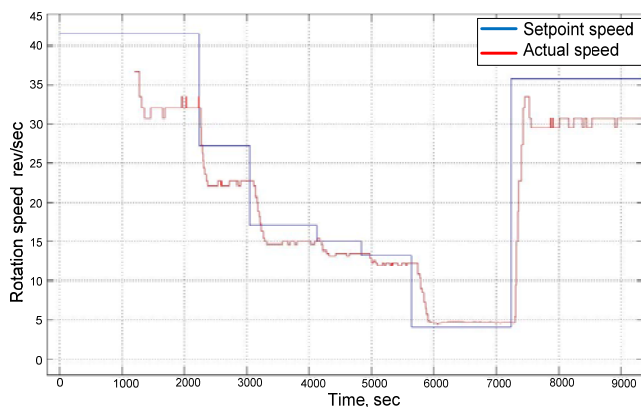


Fig. 4. Rotational speed waveform for different setting – the method using the M-file

Advantages of the method using the M-file:

- The possibility of implementing different communication protocols.
- The simple construction of the system.
- The possibility to download data for analysis during the simulation.

Disadvantages of the method using the M-file:

- There are no commands that synchronize simulation with M-file.
- Operation in Matlab software is possible only after end of the operation of the M-file.

Method using M-file allows the testing of the configuration hardware-in-the-loop with the use of PLC. Time data exchange between the PLC and the Matlab/Simulink environment is satisfactory for the test object.

The Modbus protocol was implemented on the PLC in a programmatic way. It should therefore be noted that the reading and writing of data by the PLC was not immediate. Updating is possible in the next cycle of the PLC.

#### Method using S-function

The Matlab/Simulink environment has been equipped with libraries for testing and simulation of physical phenomena. It also provides the ability to extend its

functionality through the use of so-called S-functions [10]. They can write in the Matlab software and in several others, such as C, C ++, Fortan. S-functions are compiled as files MEX (Matlab Executable).

Method using S-function requires writing an external application in C ++ that allows receiving and recording data to the PLC, and then use this application as one of the elements in the model as implemented in Matlab/Simulink.

S-function consists of two main subprograms (Fig. 5.):

- Program running in Matlab / Simulink environment, which has the task of reading the input and writing output values.
- Communications program, which is run as a separate thread running independently of Matlab / Simulink.

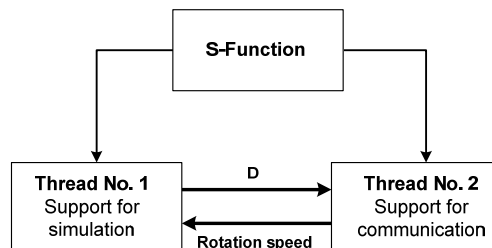


Fig. 5. Multithreading operated by S-function

The PID and the block responsible for determine the rotational speed  $n$  are located in Matlab/Simulink. The value of the duty cycle  $D$  is transmitted to the S-function. Operation of S-function is based on the launch an external application to enable communications with the PLC via HTTP. The PLC works as a web server, so the application connects to it as a client. Fig. 6 presents rotational speed waveform for different setting – a method using the S-function.

The program, which is responsible for communication with the PLC uses for its operation the library winsock2.h. These are ready-made functions for data exchange with external devices by using one of the Internet protocols. After starting the application establishes the connection to a PLC. If it goes error free it starts data exchange.

Reading and writing data to the PLC via the http protocol is possible after sending appropriate questions.

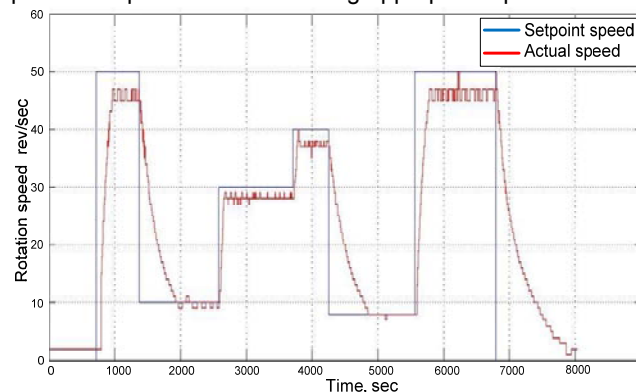


Fig. 6. Rotational speed waveform for different setting – the method using the S-function

In the control model, which has been implemented in Simulink software was also used *RTBlock* the block. This is a function responsible for the work Matlab/Simulink environment in real time. It works by allocating more resources simulation system so that the accuracy of the sampling time is preserved. This is the block was taken by Leonardo Daga.

The value of the average time between successive Http requests, registered with the help of Wireshark software is: 0,00134566s (were averaged 34501 samples).

Method using S-function enables the use of the PLC in HiL configuration to control the speed of rotation of BLDC motor. The use of an external application, which implements communication, enables fast, easy exchange of data.

Advantages of the method using the S-function:

- Fast data exchange with the help of external applications.
- Synchronization Matlab/Simulink to an external application communication.
- The possibility of implementing different communication protocols

Disadvantages of the method using the S-function:

- Required is suitable configuration of the Matlab software - the installation of additional compiler.

### Conclusions

The results acquired for a given object were also presented in Tab. 1. As implied by the conducted research, each method allows the use of the Matlab/Simulink environment and PLC. They are characterised by different data transmission times and different implementation difficulties. In the considered case – control of the BLDC motor rotational speed, all methods turned out to be appropriate, as this object has greater reaction time than the changeable force/mot.

Table 1. Juxtaposition of times between consecutive packages for selected communications methods with the PLC

Method	The protocol	The implementation of the protocol in the PLC	The time between packets [Approx]	Ack. of receipt of data
OPC Server	Ethernet S-Bus	Hardware	91 ms	None
M-file	Modbus	Software	27 ms	Yes
S-function	Http	Hardware and software	1 ms	Yes

The best results were acquired for the method that uses the S-function. The data exchange time between consecutive packets was about 1ms.

The system reaction time was also influenced by delays connected with:

- Data transmission – the protocol used.
- PLC operation cycle.
- The use of an intermediate Server – the OPC server.
- The measurement of BLDC motor rotational speed.
- Time constant of the BLDC motor.

The record of the value of duty cycle to the appropriate controller register was made in shorter time than rotational speed reading, which resulted from the specific construction of the measurement system.

The PID controller was designed on the basis of data gathered directly from the PLC. During studies it turned out, that not considering in settings selection the delay connected with data transmission had significant influence on the quality of regulation.

**Authors:** dr inż. Paweł Kielan, Politechnika Śląska, Wydział Elektryczny, Katedra Mechatroniki, ul. Akademicka 10a, 44-100 Gliwice, E-mail: [pawel.kielan@polsl.pl](mailto:pawel.kielan@polsl.pl); dr hab. inż. Damian Mazur, Politechnika Rzeszowska, Wydział Elektrotechniki i Informatyki, ul. W. Pola 2, B206, 35-959 Rzeszów, E-mail: [mazur@prz.edu.pl](mailto:mazur@prz.edu.pl); mgr inż. Adrian Szklarz.

### REFERENCES

- [1] Szafranski G., Czyba R., Fast prototyping of three-phase BLDC motor controller designed on the basis of dynamic contraction method, Variable Structure Systems VSS '08 (2008), 100-105
- [2] Dusmez S., Khaligh A., Krishnamurthy M., Ugr E., Uzunoglu M., Sensorless control of BLDCs for all speed ranges with minimal components, Electrical Machines and Power Electronics and Electromotion Joint Conference (ACEMP), (2011), 626-631
- [3] Szklarz A., Investigation on a PLC driver in a control system of DC motor with HiL configuration, Master's thesis, The promoter: Ph.D. Paweł Kielan, Gliwice 2015
- [4] Taksale A., Vaidya V., Shahane P., Dronamraju G., Deulkar V., Low cost hardware in loop for automotive application, Industrial Instrumentation and Control (ICIC), (2015), 1109-1114
- [5] Liu Qing, Qiu Yongsheng, Development of OPC server in a remote industrial control system, 12th IEEE International Conference on Electronic Measurement & Instruments (ICEMI), Vol. 01 (2015), 552-557
- [6] Li Dongjiang; Sun Ruiqi (2011), Implement of communication between configuration software and OPC server based on Modbus/TCP, Electronic Measurement & Instruments (ICEMI), Vol. 1 (2011), 218-221
- [7] Jasmin Velagic, Admir Kaknjo, Nedim Osmic, Tarik Džananovic, Networked based control and supervision of induction motor using OPC server and PLC, ELMAR (2011), 251-255
- [8] H. Afghoul, F. Krim, D. Chikouche, A. Beddar, Simulation of sliding mode controller to deliver maximum power from PV system using M-file, Industrial Electronics (ISIE), IEEE International Symposium on (2013), 1-6
- [9] Suzana Uran, Karel Jezernik, MATLAB Web Server and M-file Application, Power Electronics and Motion Control Conference. EPE-PEMC (2006), 2088-2092
- [10] L. K. Letting, J. L. Munda, Y. Hamam, Implementation of controller optimization using S-function based metaheuristic algorithms, Information Science and Control Engineering (ICISCE 2012), 1-5