Automated control system of electric drive with doubly fed induction motor

P V Avlasko, S A Bronov, D A Antonenko, Z A Avlasko

Siberian Federal University, 79, Svobodny ave., Krasnoyarsk, 660041, Russia

E-mail: pavlasko@sfu-kras.ru

Abstract. The article describes the development of an automated control system for electric drives, which includes a doubly fed induction motor, a digital position sensor and National Instruments equipment, represented by the PXI modular platform and LabVIEW software. The features of the interaction of software and hardware of the electric drive control system are considered. During operation, the system uses various levels of interaction of virtual instruments within one project. Thus, the use of virtual instruments of the project occurs at the level of the personal computer of the workplace, controller and the module with a programmable logic integrated circuit. The method of using a dynamically connected library for the calculation of parameters and state variables of the electric drive in real time is presented.

1. Introduction
A development of industrial production has led to a significant increase in the amount of used electric drives. Nowadays the features of the modern electric drive are the widespread use of semiconductor energy converters to control the speed of electric drives and microcontrollers usage to perform electric drive control tasks. Electric drive control systems create an opportunity for rational construction of technological processes increasing productivity and labor efficiency, improving the quality and reducing the cost of products [1-3].

In order to ensure the compliance of electric drives with modern requirements there is a need to use software development tools for electric drive control systems and apply new control methods with using new element base [4-6]. One of the platforms for researching the capabilities of electric drives is the PXI bus and real-time (RT) controller, in conjunction with the LabVIEW graphical application development environment presented by National Instruments (NI). The modular PXI platform includes microprocessor control tools and modules with programmable logic integrated circuit (FPGA).

2. The features of the interaction of software and hardware of the electric drive control system
The concept of modularity platform is the ability to build the system in accordance with the requirements of the user. Installed modules with reconfigurable FPGA-based I/O are used for control operations and data processing. Using the LabVIEW FPGA software, it is possible to individually configure digital lines to operate as I/O, timers and counters or for organizing specialized communication protocols.

The PXI RT controller uses the PXI bus, which is an extended version of the PCI computer bus and communicates with the modules installed in the system in particular the NI PXI-7833R multifunctional I/O module. These module contain FPGA and have 8 channels of analog I/O lines and 96 digital I/O lines.
The electric drive with the doubly fed induction motor with phase-dependent control includes the motor with the first and second stator windings and the systems that generate control signals of the required shape (Fig. 1). The master control unit forms the laws of variation of the control signals for the amplitudes, frequencies and phase shifts of the supply voltages depending on the desired angle of rotation and the angular velocity of the rotor. The distribution control unit generates separate signals for each of the two inverters based on the values of the angle position sensor and the mechanical load.

![Figure 1. Block diagram of the electric drive with doubly fed induction motor.](image)

The LabVIEW FPGA software module in the NI LabVIEW graphical programming environment was used to develop the control and data acquisition unit for the NI PXI-7833R module.

Software modules running on the controller and I/O modules provide real-time system operation generate with pulse-width modulation (PWM) sinusoidal currents in two of three-phase windings of a doubly fed induction motor.

The software developed in the LabVIEW environment consists of a project represented by virtual instruments located on the operator’s workplace, the PXI controller with a real-time operating system and a multifunctional I/O module with FPGA (Fig. 2).

![Figure 2. Project deployment scheme.](image)
Each of the used virtual instruments is unique in function. The project is represented by a set of virtual instruments (Fig. 3):

1) CompMain.vi is a virtual instrument located at the operator’s workplace. It is used to upload experimental data from the controller and further process the received data;
2) HostMain.vi is a virtual instrument located on the NI PXI – 8106 RT controller. It is used for generating real-time PWM signals and obtaining information from sensors;
3) MathModel.vi is a virtual instrument in the form of dynamical linked library (DLL) used to process a mathematical model in real time;
4) FPGA_RIO1_Main.vi is a virtual instrument located in the NI PXI – 7833R I/O module. The FPGA of this module is used to generate control signals with transistor keys of inverters, digitize analog signals from current sensors and receive data from a digital position sensor;
5) FPGA_RIO1_SubVI_GetIntervalNumber.vi is a virtual instrument used with the FPGA_RIO1_Main.vi instrument as a pulse selector in the form of a user library;
6) FPGA_RIO1_SubVI_GetShim.vi is a virtual instrument used as a user library in the FPGA_RIO1_Main.vi instrument as a PWM pulse shaper;
7) FIFO_DATA.vi is a virtual instrument used to transfer data between FPGA_RIO1_Main.vi and Host_Main.vi during program execution.

Figure 3. Scheme of project components interaction.

3. The method of using a dynamically link library for the calculation of parameters and state variables of the electric drive

The dynamic link library presented in the MathModel.dll is written in the C language and is used to calculate parameters and state variables of the electric drive in real time. The dynamic link library is located in the workplace with installed NI LabVIEW, however, when the executable program is launched, it is loaded and executed at the controller level in real time.

The following method is proposed for use for carrying out calculations in real time. The main essence of the method is to create a dynamically connected library of a specific configuration in the C/C++ language and embed this library into the virtual instrument used in the LabVIEW project.

This method reduces the time required for create and debug code implemented in LabVIEW using the built-in Call Library Function Node. This node is one of the many basic functions in the integrated development environment, which makes it possible to use the method without additional software packages supplied by the manufacturer.
A characteristic feature of the dynamic link library is the ability to use one library loaded into the system during the execution of many programs. This has a positive effect on the use of random access memory (RAM) resources, especially in conditions of limited system resources. This library is a plug-in that extends the capabilities of LabVIEW software.

To create the library, a text-based integrated development environment is required. This project uses Microsoft Visual Studio development environment.

The program code presented when creating the DLL differs from the code implemented in classic C/C++ applications using keywords and the application build method. So, the prototype of the flux linkages in electric drive is as follows:

```c
double __declspec(dllexport)
    psiY(double alpha1, double alpha2, double k1, double k2, double omega1, double omegadv, double U2x0, double U2y0, double u1y, double psi1x, double psi1y)
```

The `__declspec(dllexport)` keyword is a specific extension of C and C++ languages to Microsoft and is used to export functions, data and objects to a dynamic link library.

To create a DLL correctly, you need to set special settings in the configuration properties. Additional directories for include files and libraries are listed here. In this case, the directory should point to labview.lib to bind the generated program code to the LabVIEW core. You also need to add the library file labview.lib to the configuration of the project being created. The program code should include the "windows.h" header file, which defines a large number of special Windows functions that can be used in C, as well as "extcode.h" used in LabVIEW for working with imported code.

After performing the above operations, the project in LabVIEW using the DLL is ready for assembly.

In the case of using a DLL with a real-time controller, to simplify the interaction between the real-time system and the system on which the library was developed, the NI software vendor presents the LabVIEW RT DLL Checker utility (Fig. 4). This utility is designed to check the compatibility of individual elements and provides information about functions or DLLs in general, imported into a specific version of the software.

![Figure 4. Dynamic link library compatibility check.](image)
After successful compatibility testing, you must configure the Call Library Function Node block to use. When opening block settings in the Function block, the path or name of the connected library is...
assigned. Also, when transferring external code, there is a choice of calling convention Cdecl and stdcall(WinAPI). Cdecl is the calling convention used by compilers for the C language. Stdcall or WinAPI is the calling convention used in Windows for calling WinAPI functions. In the "Parameters" block, the variables passed to the function, as well as their name, variable type, data type used and pointer mode are indicated.

It is necessary to strictly follow the sequence of variables passed to the function in accordance with the program code (Fig. 5). Thus, the export of the dynamic link library is completed and the virtual instrument is ready to run.

Figure 5. Method of using dynamic link library.

4. Conclusion
The project of the control system for doubly fed induction electric drive in the graphical integrated development environment LabVIEW is implemented. The project is represented by a set of virtual instruments. Each of them performs a specified set of functions and is physically located at different hardware levels. To calculate the parameters and variables of the electric drive state, a virtual device with a dynamically link library is used. The calculations are performed on the PXI controller in real time.

Acknowledgements
This work was supported by the Ministry of Education and Science of the Russian Federation in the framework of the Federal target program «Research and development on priority directions of development of the scientific-technological complex of Russia for 2014-2020» (agreement № 14.575.21.0142, unique ID project RFMEFI57517X0142).

References
[1] Finch J W and Giaouris D 2008 Controlled AC Electrical Drives IEEE Transactions on Industrial Electronics 55(2) 481-491
[2] Gottlieb I M 1994 Electric Motors and Drives: Fundamentals, Types and Applications (London: Elseveir Inc.) p 410
[3] Harnefors L, Saarakkala S E and Hinkkanen M 2013 Speed Control of Electrical Drives Using Classical Control Methods IEEE Transactions on Industry Applications 49(2) 889-898
[4] Bose B K 2006 Power Electronics and Motor Drives (London: Elsevier Inc.) p 917
[5] Holtz J 2002 Sensorless control of induction motor drives Proceedings of the IEEE 90(8) 1359-1394
[6] Rodriguez J, Kennel R M, Espinoza J R, Trincado M, Silva C A and Rojas C A 2012 High-Performance Control Strategies for Electrical Drives: An Experimental Assessment IEEE Transactions on Industrial Electronics 59(2) 812-820