ISSN 1454 - 8003
Proceedings of 2017 International Conference on Hydraulics and Pneumatics - HERVEX
November 8-10, Băile Govora, Romania

SCADA SYSTEMS ARCHITECTURE BASED ON OPC SERVERS AND APPLICATIONS FOR INDUSTRIAL PROCESS CONTROL

Marcel NICOLA[1], Claudiu-Ionel NICOLA[1], Dumitru SACERDOȚIANU[1], Marian DUȚĂ[1]

[1]National Institute for Research, Development and Testing in Electrical Engineering – ICMET Craiova

marcel_nicola@yahoo.com, nicolaclaudiu@icmet.ro, dumitru_sacerdotianu@yahoo.com, marianduta@icmet.ro

Abstract: *SCADA (Supervisory Control and Data Acquisition) is the most modern tool used for the control and monitoring of technological processes. A SCADA system consists of two main components: the Server (one or more) and the Clients (the Viewers). The aim of OPC (OLE-Object linking and embedding for Process Control) is to define a common interface that once conceived can be reused for any other project, SCADA, HMI (Human Machine Interface) or other software packages. An OPC server is an application which acts as application programming interface (API-Application Programming Interface) or protocol converter. The paper presents an example of OPC server based application software which can be integral to a SCADA system. The application focuses on monitoring a quasi-general industrial process defined by a 2nd order transfer function, on identifying the transfer function and managing the client-server communication of the quantities of interest by online viewing and creating a record using TDMS (Technical Data Management Streaming) files and a MySQL database server.*

Keywords*: OPC-UA, database, SCADA, process control, transfer function*

1. Introduction

SCADA is a computer based system designed for the control and monitoring of technological processes.
SCADA is the most modern concept and the tool used for the control and monitoring of technological processes. SCADA systems include both software and hardware components. The hardware collects and transmits data to a PC with installed SCADA software. The PC then processes these data in an acceptable time frame [1, 2].
Substantial progress in this field have allowed the SCADA systems to be used in the most various fields, from the manufacturing of consumer goods, to metallurgy, industrial hydraulics and pneumatics, chemistry and power engineering, and the nuclear field.
A SCADA system consists of two main hardware components:

- The server (one or more) - It is connected to the (process) field elements through various data acquisition systems. Data acquisition systems are generally created based on microcontrollers designed to acquire data from the process and to monitor and control the operation of the process. Data acquisition is also carried out by using intelligent sensors which can connect directly to the computer or through some intermediate devices called stations or communication masters which collect data from multiple intelligent sensors. The data acquisition and process control devices in the industry field are represented mostly by programmable automata - PLC's (Programmable Logic Controller).
- The server manages all the data collected from the process (it achieves the database, provides the communication with the PLC's in the process);
- The Client (the Viewer) - is connected to the network with the server, it uses the data from it and provides communication with the human operator.
- The servers are connected to the controllers through a wide range of communication drivers (hundreds of drivers providing connections with all PLC's from known companies).

A single server can communicate simultaneously across multiple protocols, but new communication drivers can also be developed.

The servers and viewers are connected to the Ethernet network. The web technology used now also allows visualization of a process done via the Internet.
One of the most important functions of the SCADA systems is the monitoring and control function.
The monitoring and control of technological processes is carried out by means of graphical pages which mimic the technological process and are displayed on one or more computer monitors. These graphical pages are also called HMI's (Human Machine Interfaces). The control operation is also called monitoring. We can thus state that the monitoring and control of technological processes is carried out via the HMI's.
Among the main functions of the SCADA system we can mention the following:

- The automatic control of the technological process in order to optimize the output parameters and in order to improve the efficiency;
- The real-time display of the technological process state;
- The graphical display of the process data with a view to creating efficient operating strategies;
- The efficient management of process quantities record, state of equipment, and condition of alarms;
- The periodic generation of operation reports;
- The possibility of direct user intervention in the process depending on his rights of access;
- The possibility of remote process control by using SCADA client stations.

The OPC (OLE for Process Control) standard or the more recent Openness Productivity Connectivity is a series of specifications defined by the OPC Foundation to facilitate connectivity in industrial automation. The OPC uses Microsoft DCOM (Distributed Component Object Model) technology to provide a communication line between the OPC servers and OPC clients. The OPC was designed to enable the safe communication in industrial processes, such as in electric power generation and distribution, industrial hydraulics and pneumatics, petrochemical refining, assembly lines for motor vehicles, etc.
The state-of-the-art control of industrial processes integrated into SCADA systems entails both the use of up-to-date data transmission and management facilities, and also the use of drive and control software facilities starting with the estimation of parameters and identification of the linearized transfer function, proceeding with the control design of the controller and finishing with the validation and testing of the entire chain of the industrial process control [3-5].
LabVIEW is the development software which includes the concepts outlined above and which will be used for the application described in this article.
The rest of the article is organized as follows: section 2 presents the OPC client/server concept, and section 3 presents an application for monitoring a quasi-general industrial process described by a 2nd order transfer function, the identification of the transfer function and management of client-server communication for the quantities of interest by online viewing and creating a record using TDMS files and a MySQL database server. Section 4 presents conclusions and course of action concerning future research.

2. OPC server

The OPC (OLE for Process Control) is an industry standard created by the collaboration of a group of important world suppliers of automation software and hardware with Microsoft company. The standard defines the methods for data communication between real time automation systems and client applications run on computers with Microsoft operating systems [6].
OPC standard was created to enable the access of client applications to automation data in a uniform manner. Widespread conformity in the industry can bring many benefits such as:

- The hardware manufacturers will have to write a single set of software components for their products, which customers will use for their applications;
- The software manufacturers will no longer have to rewrite the drivers as a result of the changes or additions from new hardware versions;

- The customers will have a wider range of options to achieve high quality integrated production systems, in a heterogeneous computer environment.

The OPC standard provides access of production and economic applications to field information in real time and in a consistent manner, facilitating the interoperability between different equipment and the "plug and play" connectivity, but also a greater flexibility, lower integration, development and assembly costs for process automation or control systems.

An OPC server (see Figure 1) is an application which behaves as an API-Application Programming Interface or protocol converter.

OPC servers arose from the imperative necessity of making the production and economic computer systems communicate among themselves. There were often barriers due to incompatibilities between custom communication interfaces and the automation hardware and software from different vendors.

The specifications of the OPC define a standard COM (Component Object Model) interface for use in industrial applications for data acquisition and control.

The specs include a protocol for defining objects, for determining their properties and for standardization of function calls and events. For this purpose, OPC comprises a wide variety of data sources.

The input-output devices include data acquisition devices, actuators, communication bus systems and programmable logic controllers (PLC). The specifications also include protocols for working with data control systems (DCS) and database applications as well as for access to on-line data, alarm and event control and access to data records for all these data sources [7].

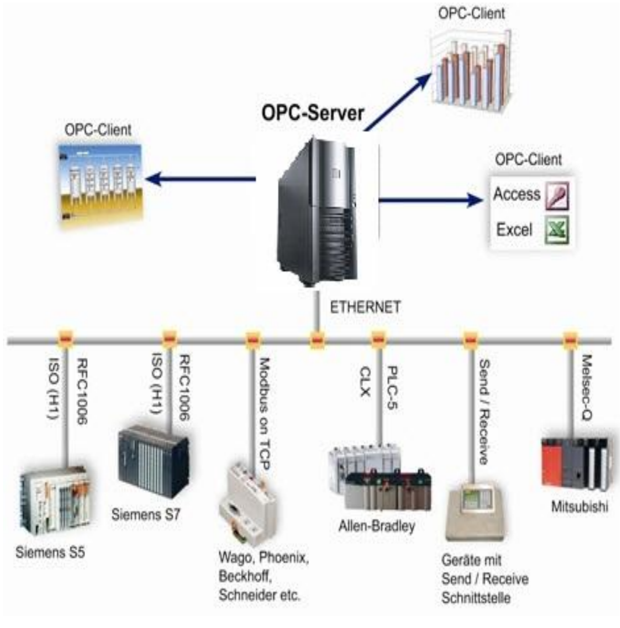


Fig. 1. SCADA system based on OPC server architecture

The OPC architecture benefits by the advantages of COM interfaces, which provide a convenient mechanism for extending OPC functionality.

Their architecture and design facilitates the creation of OPC servers which allow the client applications to access data from multiple OPC servers from different vendors running on different nodes on a single object.

The main advantage of the OPC interface is independence from a particular manufacturer, product or hardware. All commercially available SCADA systems, process control systems and PC-based controllers now provide a free OPC client interface for access to any OPC server.

The technology provides clear distinction between server and client applications, it encapsulates product-specific features and allows easy upgrading to a new version or switching to a different product. The independence to the platform also enables OPC communication between

components running different operating systems and via the Internet.
Considering that the OPC standard is based on the DCOM (Distributed Component Object Model) technology, its use is restricted to Windows operating system, at the same time, the DCOM contributed to the success of OPC.
The rapid acceptance of Windows computers as automation components has enabled the OPC technology to gain worldwide relevance in just a few years, while on the other hand, by the more intense use of the OPC, the emerging new application fields which determine the general trend towards Web technologies entail new requirements for the OPC standard.
OPC UA (Unified Architecture) is the next generation of OPC technology and is the OPC technology used by the application described in section 3. OPC UA is a more secure, open, reliable mechanism for transferring information between servers and clients. It provides more open transports, better security and a more complete information model than the original OPC, “OPC Classic.” OPC UA provides a very flexible and adaptable mechanism for moving data between enterprise-type systems and the kinds of controls, monitoring devices and sensors that interact with real world data [8].
OPC UA is much more complex than previous OPC specifications and is designed to:

- use cross-platform capable communication instead of Windows DCOM;
- combine the OPC DA (Data Access), A&E (Alarm &Events), HDA (Historical Data Access) functionality into a single set of services;
- model complex data structures for collaboration with other standards organizations;
- be implementable on different platforms, from embedded systems to enterprise systems.

An OPC UA server endpoint is the side of an OPC UA communication that initiates a communication session and provides data to an OPC UA client. There is no standard OPC UA server either in functionality, performance or device type. Devices from small sensors to massive chillers may be OPC UA servers. Some servers may host just a couple of data points. Others might have thousands. Some OPC UA servers may use mappings with high security and lower performance XML, while others may communicate without security using high performance OPC UA Binary Encoding. Some servers may be completely configurable and offer the client the option to configure data model views, alarms and events. Others may be completely fixed.
Clients in OPC UA are much more flexible than other network clients. OPC UA Clients have the capability to search out and discover OPC UA servers, discover how to communicate with the OPC UA server, discover what capabilities the OPC UA servers have, and configure the OPC UA server to deliver specific pieces of data when and how they want it. OPC UA clients will generally support many different protocol mappings so that they can communicate with all different types of servers.

3. Example of OPC-UA server based SCADA application for the control of an industrial process

Most industrial processes can be defined around the quiescent operating point by a 2^{nd} order transfer function. The application presented is achieved in the LabVIEW development environment using Mathscript type Matlab scripts, it uses an OPC-UA type server, TDMS type files and a MySQL type database server [9]. The general architecture of the application is shown in Figure 2.
For the identification of the transfer function of the technological process, by using an acquisition board, the step signal applied is acquired and respectively the response (represented in simulated form in Figure 3).
The types of software modules of the experimental model, on the server machine are the following:

- The OPC read-write module; this module will provide intrinsic connection between the data acquired from input modules and the OPC server.
- The MySQL read-write module; this module will achieve MySQL type database query, where obviously data writing and reading are the most used functions.
- The TDMS read-write module; this module will manage data writing and reading in TDMS files, in order to achieve additional cache. These files can be opened in EXCEL, the facilities being thus obvious.

- Interface configuration module for data acquisition channels; this module will achieve the configuration of acquisition channels from various acquisition modules, but in the application presented the process quantities are simulated but maintain a natural similarity to real applications which process data from transducers.

The clients will be programs located on the server computer or on separate computers connected through a router in the same Intranet network.

These programs will be Viewer type programs which will access data from the OPC client, from MySQL database or from TDMS files. Data processing and viewing will be primary, focusing on the data communication itself, in that the data will be acquired by the client uniformly from databases or EXCEL-compatible files, although initially the data sources are heterogeneous (different data acquisition modules, for different protocols) [10-15].

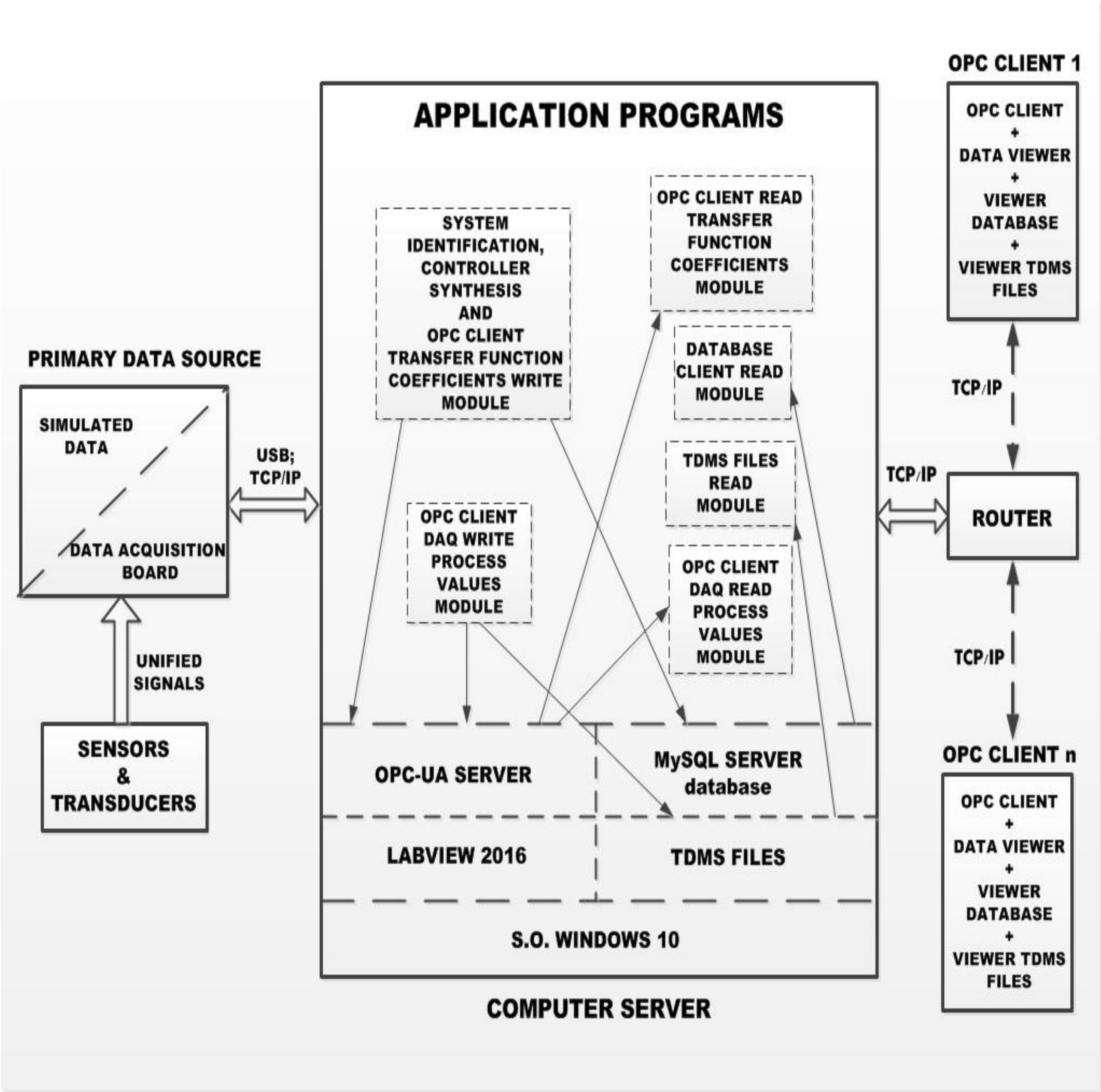


Fig. 2. SCADA application architecture

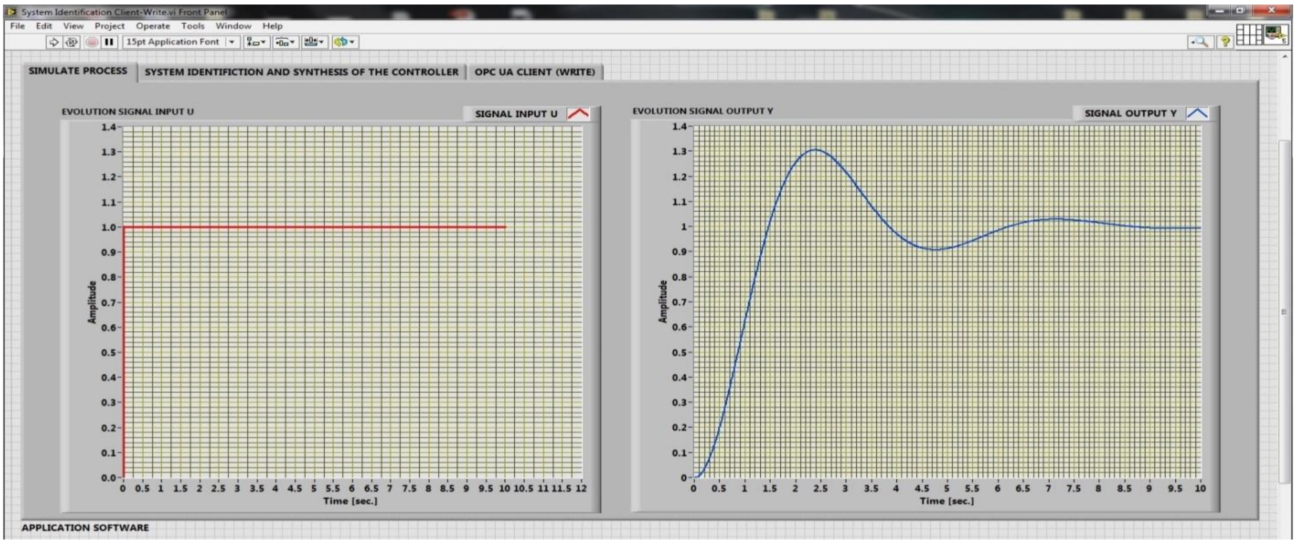


Fig. 3. The step signal applied and the response of the second order system

The use of LabVIEW further allows the transfer function coefficients to be determined, a LQR (Linear Quadratic Regulator) type regulator to be synthesized, a possible state description equations of open circuit, and respectively closed circuit (with regulator) system, as shown in Figure 4. It is obvious that this procedure is asynchronous and is run on operator request when the operating conditions allow it. The validation of the system and regulator identification is also presented in Figure 4. where it’s see that the identified system response is overlaid on the initial response.

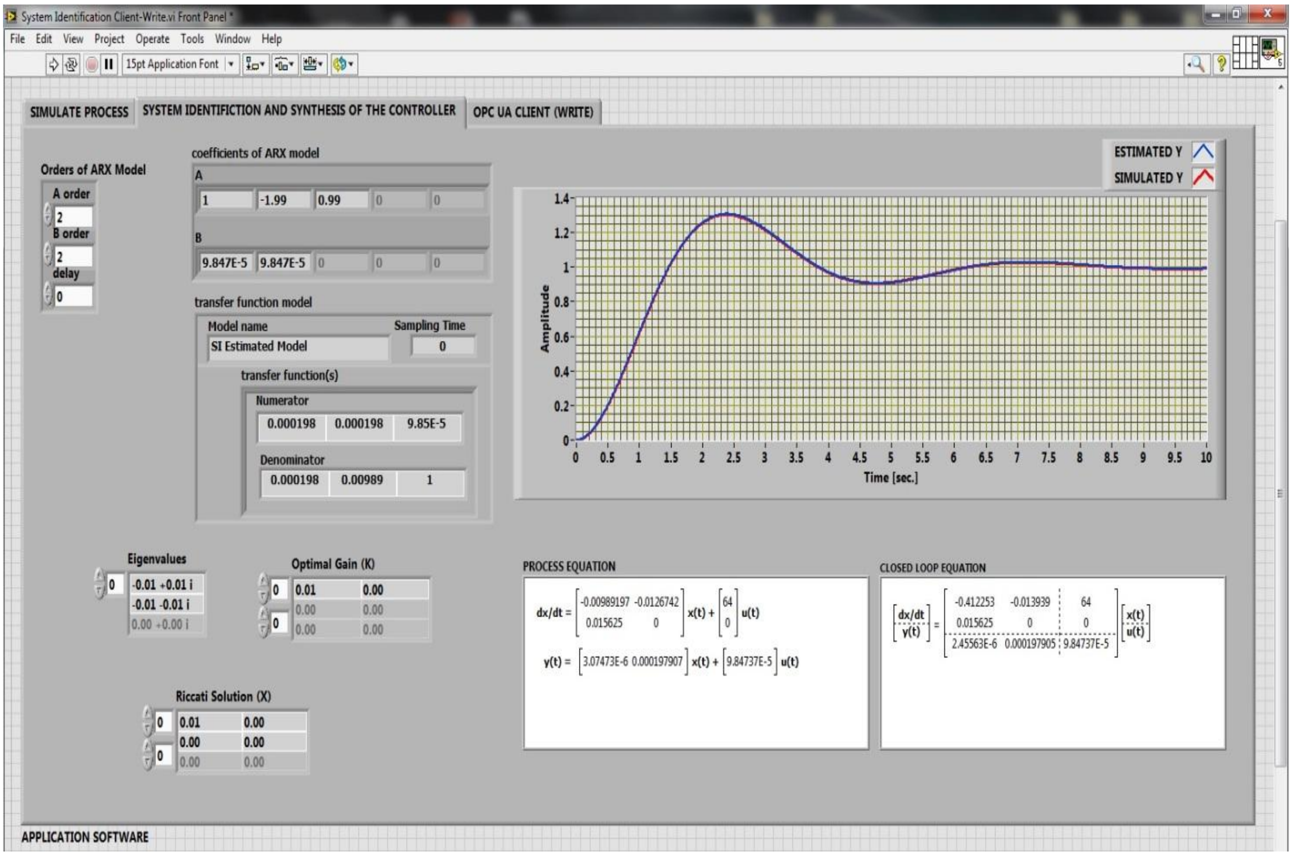


Fig. 4. System identification, LQR controller synthesis, state equations and system validation

The OPC-UA Server which supports communication is defined by the block diagram in Figure 5, and its interface is presented in Figure 6.

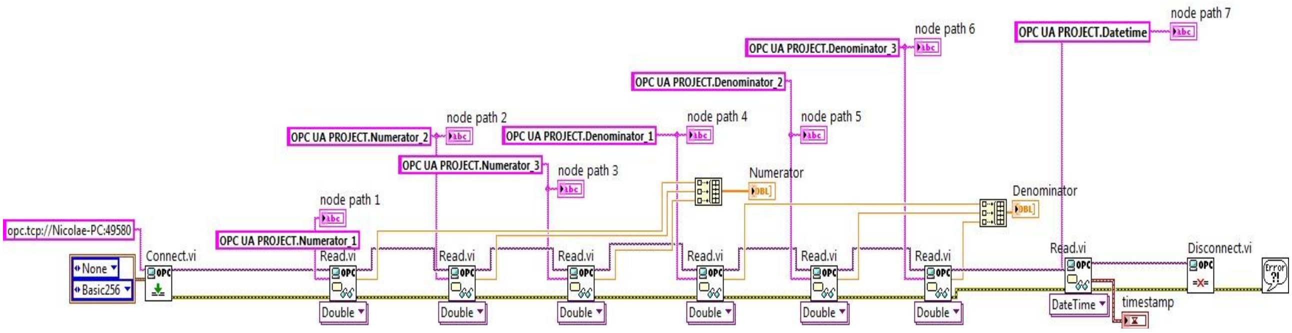


Fig. 5. OPC-UA server block diagram

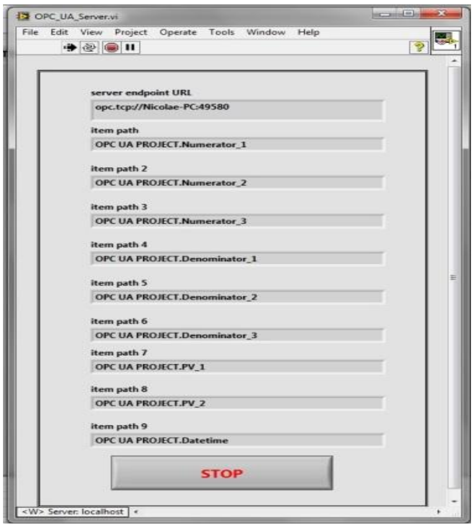


Fig. 6. OPC-UA server interface

There will be 8 quantities which will be transferred via the OPC client-server technology, in the presented application. The first 6 quantities are the transfer function numerator, respectively denominator coefficients (see Figure 7). These quantities will be transferred asynchronously at request and then they will be stored on the MySQL Server (see Figure 8).

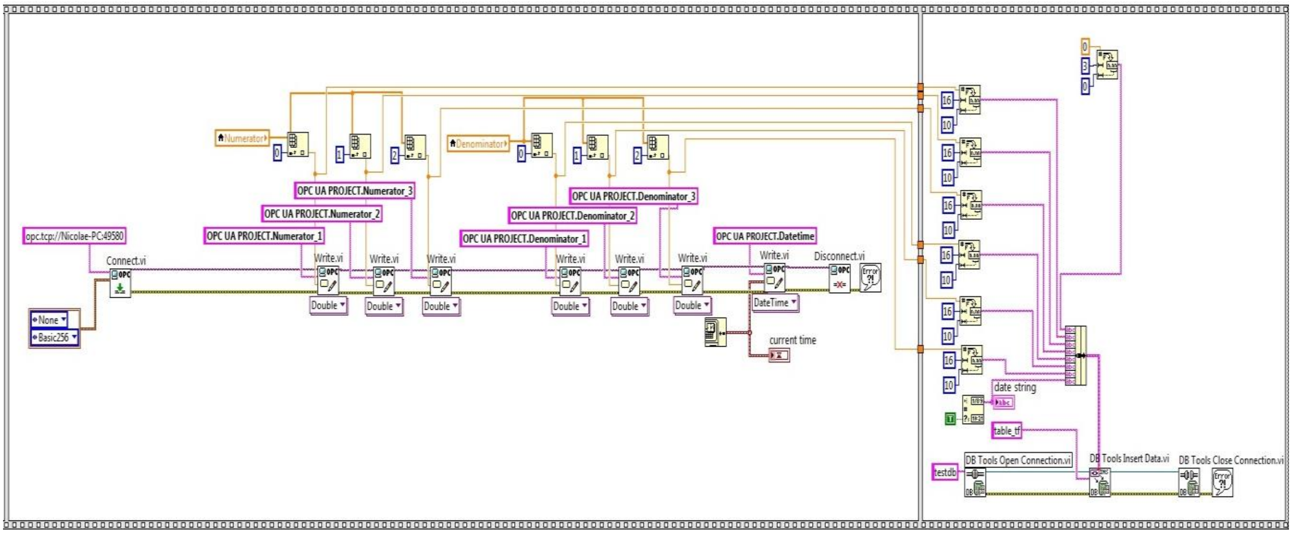


Fig. 7. OPC-UA client block diagram for transfer function coefficients

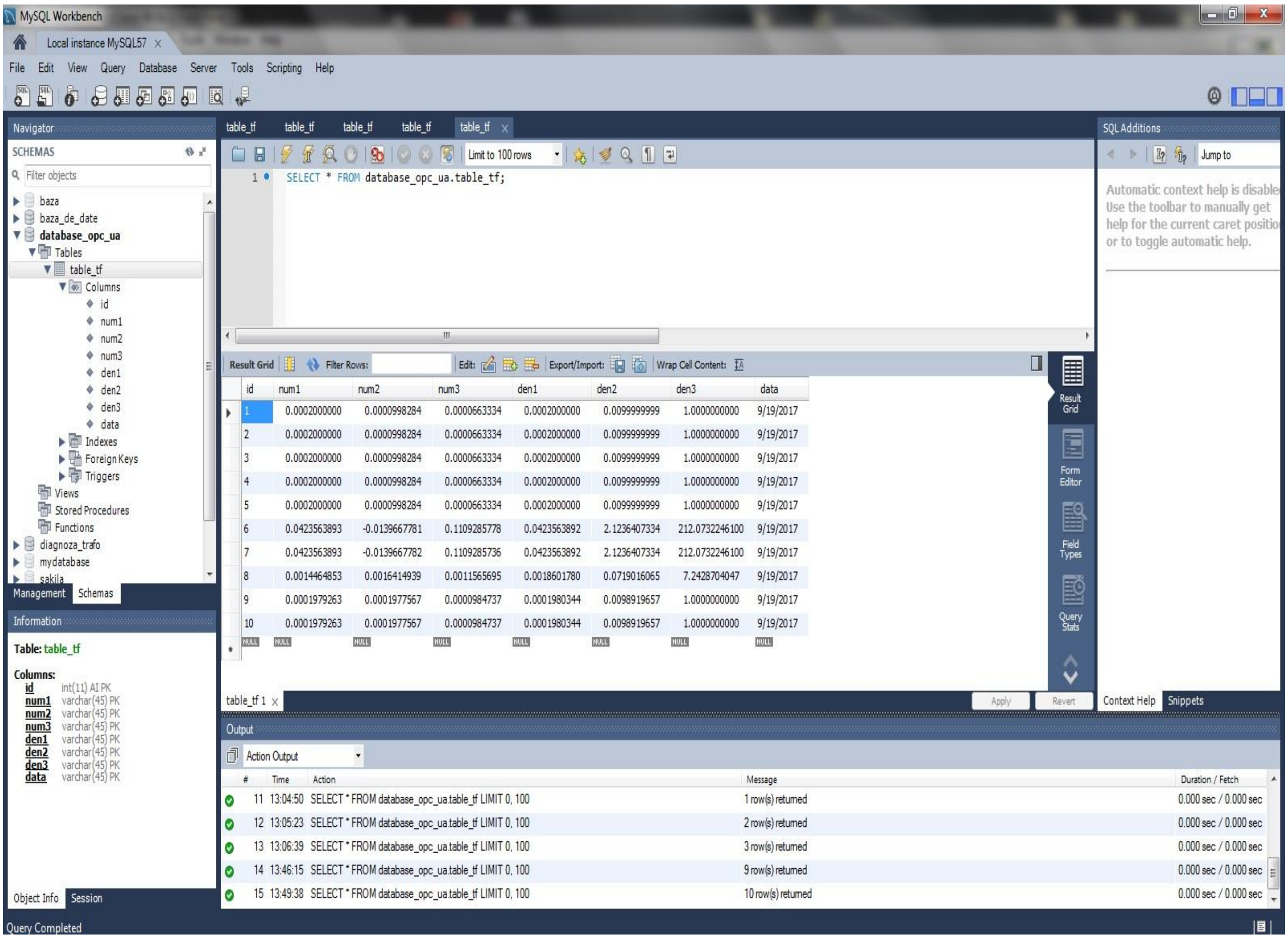


Fig. 8. MySQL server database for transfer function coefficients

In addition, another asynchronous software module can read the history of the transfer functions stored in the MySQL Server (defined as the six coefficients) see figs. 9 and 10.

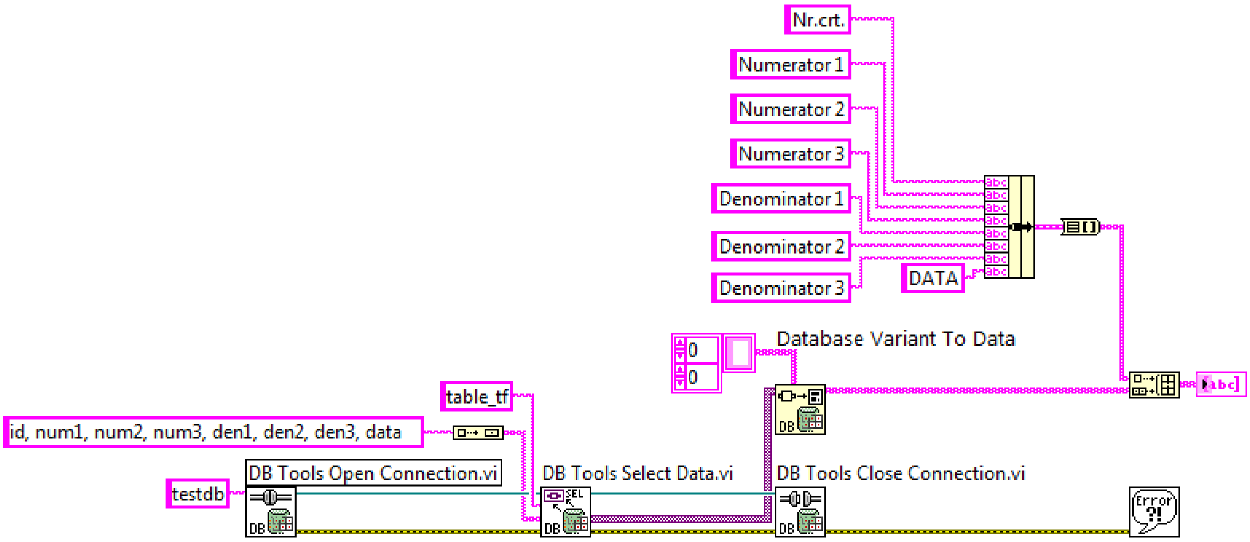
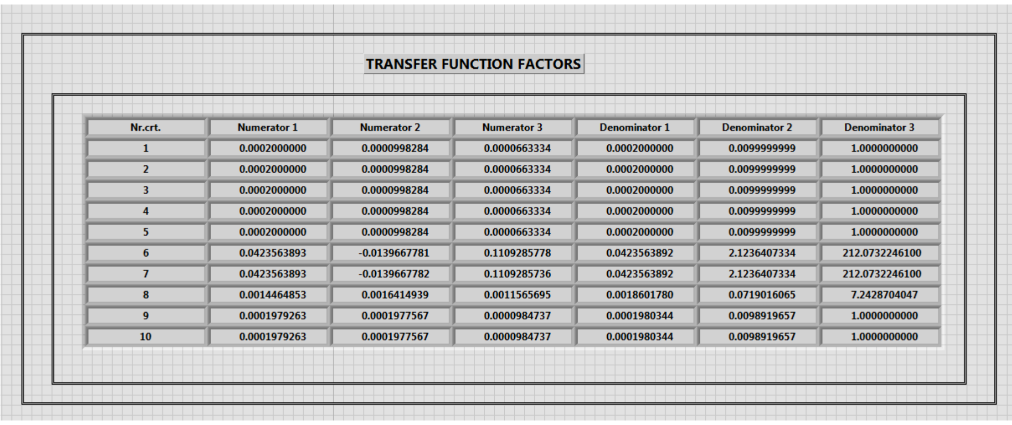


Fig. 9. Client read from MySQL server database transfer function coefficients block diagram



TRANSFER FUNCTION FACTORS

Nr.crt.	Numerator 1	Numerator 2	Numerator 3	Denominator 1	Denominator 2	Denominator 3
1	0.0002000000	0.0000998284	0.0000663334	0.0002000000	0.0099999999	1.0000000000
2	0.0002000000	0.0000998284	0.0000663334	0.0002000000	0.0099999999	1.0000000000
3	0.0002000000	0.0000998284	0.0000663334	0.0002000000	0.0099999999	1.0000000000
4	0.0002000000	0.0000998284	0.0000663334	0.0002000000	0.0099999999	1.0000000000
5	0.0002000000	0.0000998284	0.0000663334	0.0002000000	0.0099999999	1.0000000000
6	0.0423563893	-0.0139667781	0.1109285778	0.0423563892	2.1236407334	212.0732246100
7	0.0423563893	-0.0139667782	0.1109285736	0.0423563892	2.1236407334	212.0732246100
8	0.0014464853	0.0016414939	0.0011565695	0.0018601780	0.0719016065	7.2428704047
9	0.0001979263	0.0001977567	0.0000984737	0.0001980344	0.0098919657	1.0000000000
10	0.0001979263	0.0001977567	0.0000984737	0.0001980344	0.0098919657	1.0000000000

Fig. 10. Client read from MySQL server database transfer function coefficients interface

The process quantities defined in the OPC-UA Server as process value PV_1 and PV_2 are transferred synchronously within 1 second and stored in TDMS files with preset time periods (days, weeks, etc.), see Figures 11 and 12.

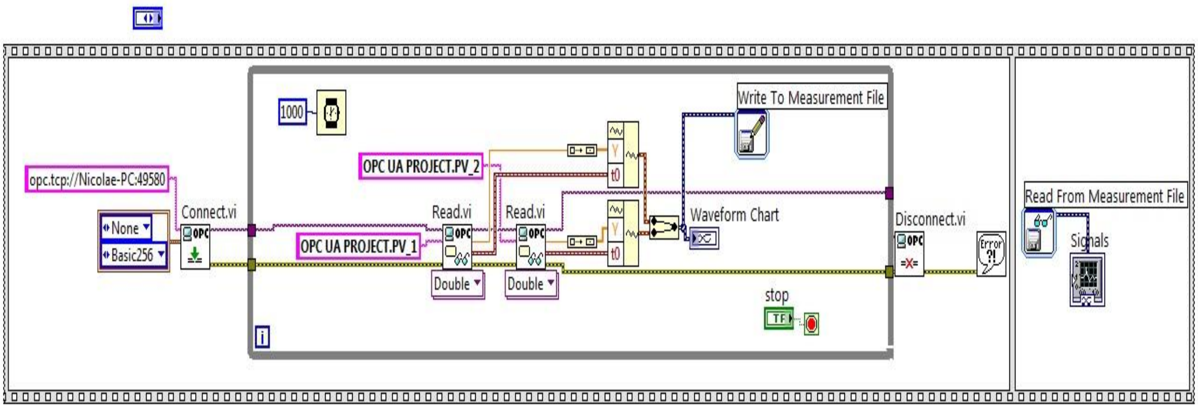


Fig. 11. OPC-UA client write and read TDMS block diagram

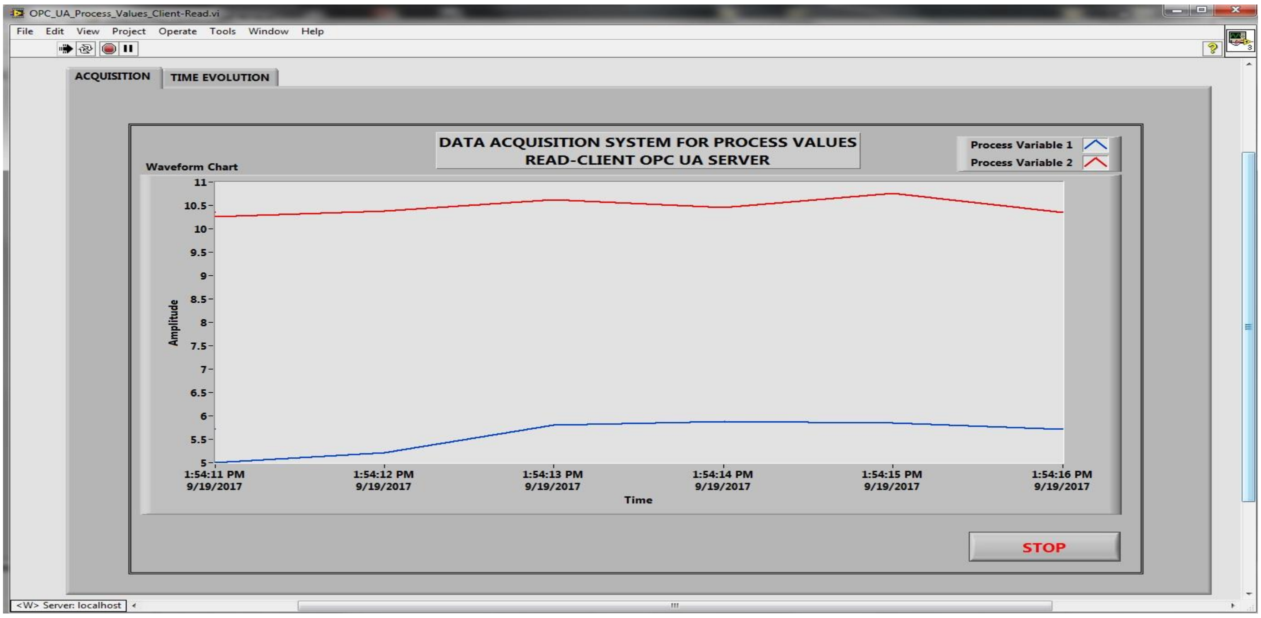


Fig. 12. OPC-UA client read TDMS interface

The presented application is modular and can be extended to multiple quantities for an actual industrial process by following the steps implemented in the presentation above. Practically the quantities implemented into SCADA are defined in each software module by groups of interest (quantities accessed synchronously or asynchronously, quantities acquired from the process through acquisition boards from transducers or system description quantities resulting from the process of identification).
The inclusion of several quantities of interest is carried out naturally by following the steps described above.

4. Conclusions

This article summarizes the SCADA systems and the principles of OPC client-server communication. Hence an application was achieved and presented for monitoring a quasi-general industrial process defined by a 2nd order transfer function, for identifying the transfer function and managing the client-server communication of the quantities of interest by online viewing and creating a record using TDMS files and a MySQL database server.
The issue of data communications in industrial environments is not abstract, but arises practically in industrial and monitoring applications, and can become a separate issue, but it can only be studied on condition that minimal software modules are provided for data communication and management from an actual application.
We consider that by achieving such an experimental model like the one described above, but especially by means of the experiments conducted on it, the main benefit will consist in the increase of the amount of knowledge on this topic and a significant facilitation in achieving future complex monitoring applications for systems with a relatively large volume of data and for heterogeneous data acquisition systems.

Acknowledgments

The paper was developed with funds from the Ministry of Education and Scientific Research as part of the NUCLEU Program: PN 16 15 02 06.

References

[1] M. Nicola, D. Sacerdoțianu, M. Duță, D. Popa, „Sisteme SCADA pentru monitorizarea echipamentelor electrice”, Ed. Electra (ICPE), 2011;
[2] D. Bailey, E. Wright, „Practical SCADA for Industry”, IDC Technologies, 2003;
[3] S. A. Boyer, „SCADA: Supervisory. Control and Data. Acquisition. 3rd Edition”, ISA The Instrumentation, Systems, and Automation Society, 2004;
[4] http://www.ni.com/getting-started/labview-basics/environment;
[5] http://www.mathworks.com/products/simpower/model-examples.html.
[6] http://www.opcconnect.com/;
[7] T. V. Bhaskarwar, S. Giri, R. G. Jamakar, „Automation of shell and tube type heat exchanger with PLC and LabVIEW”, Industrial Instrumentation and Control (ICIC), pp. 841 - 845, 28-30 May 2015;
[8] https://opcfoundation.org/about/opc-technologies/opc-ua/
[9] https://www.mysql.com/
[10] http://home.hit.no/~hansha/documents/labview/training/Database%20Communication%20in%20LabVIEW/Database%20Communication%20in%20LabVIEW.pdf
[11] J. Jovitha, “Virtual Instrumentation Using Labview“, PHI Learning, New Delhi, 2011;
[12] R. Bitter, T. Mohiuddin, M. Nawrocki, “LabVIEW: advanced programming techniques, Second Edition“, CRC Press, 2006;
[13] N. Kehtarnavaz, P. Loizou, M. Rahman, “An Interactive Approach to Signals and Systems Laboratory“, Connexions, Huston, 2011;
[14] http://www.ni.com/pdf/manuals/371303n.pdf
[15] D. Selisteanu, C. Ionete, E. Petre, “Virtual Instrumentation. Digital Signal Processing Applications,” Ed. Universitaria, Craiova, 2012;