

CONFIGURATION OF DELTAV SYSTEM AND OPC PROTOCOL IN ORDER TO IMPLEMENT A LEVEL CONTROL SYSTEM

Lect. Dr. Marian Popescu¹

¹ Automatic Control, Computers and Electronics Department/Petroleum-Gas University of Ploiesti, **Romania**
e-mail: mpopescu@upg-ploiesti.ro

ABSTRACT

DeltaV distributed system is used in many industries and offers the possibility of process control, with the goal of accelerating the planning, installation, data acquisition, operation, management, maintenance of an automated system, and optimization of the process. The present paper has as main goal the configuration of the DeltaV system and Simulink environment in order to transfer data between them in real time using the OPC protocol. Thus, in DeltaV system there are implemented modules associated with the transducer, the control valve and the water pump from a level control system. Also, in Simulink it is implemented the controller of the level control system, and OPC protocol functions are configured in order to allow the acquisition of level value from the level transducer (configured in DeltaV with an analog input block), and to transmit the commands for the control valve and the pump (configured in DeltaV using analog output and discrete output blocks).

Keywords: DeltaV, Simulink, OPC protocol, level control system

INTRODUCTION

The OPC (Object Linking and Embedding (OLE) for Process Control) protocol is a software interface technology used to facilitate data transfer between industrial control systems, Human-Machine Interfaces (HMI), and surveillance systems. This protocol was developed in response to the need for a standardized method to allow the interconnection of different control systems, being a protocol whose specifications are made by the OPC Foundation [1].

Today it has become the most important technology for the integration of various control devices. The protocol allows programs installed on Windows to communicate with industrial hardware devices.

OPC is implemented as client / server pairs. The OPC server is a software program that converts the hardware communication protocol used by a control equipment into the OPC protocol. It usually gathers information from devices (such as PLC, DCS or

SCADA). The server then provides access to this data, allowing multiple OPC clients to read and write indirectly to the device in the field through the OPC server. The OPC software client is an application that accesses data held by OPC servers. The OPC client uses the OPC server to obtain data or send commands to the hardware [2].

The OPC protocol is based on Microsoft Distributed Component Object Model (DCOM) technology, which is the culmination of several other technologies, including Component Object Model (COM) and Object Linking and Embedding (OLE).

OPC is designed to develop high interoperability between automation applications, field devices, and process control applications.

The OPC protocol is an open standard, which means lower costs for manufacturers and more options for users, and has the following specifications [3]:

- OPC Data Access (OPC DA) – used for real-time data transfer from PLC, DCS and other control devices to human-machine interfaces (HMI);
- OPC Alarms & Events (OPC A&E) – provides notifications of alarms and events upon request;
- OPC Batch – is a specification for batch processes;
- OPC Data Exchange – provides client/server-to-server communications over Ethernet networks;
- OPC Historical Data Access (OPC HDA) – allows access to stored data;
- OPC Security – specifies how to control client access to servers for information protection purposes;
- OPC Complex Data – gives servers the ability to display and describe complex data types such as binary structures or XML documents;
- OPC Commands – is a set of interfaces that allow OPC clients and servers to identify, send, and monitor commands executed by a device.

The OPC Unified Architecture contains specifications that provide standards that can be used on multiple hardware and software platforms and integrates all the OPC classic specifications functionalities into one framework [4].

The OPC protocol is part of many industrial DCSs, one of these being DeltaV system from Emerson Process Management.

From a hardware point of view, the DeltaV system consists of the following elements:

- input-output subsystems that process information from field devices;
- DeltaV Controller that performs local control and manages information and communications between the Industrial Control Network and the input-output subsystem [5];
- power supplies;
- workstations with the role of providing the graphical user interface for the process;

- industrial network Control Network that provides communication between system nodes.

Within the system, the DeltaV Controller coordinates the digital communication protocols FOUNDATION Fieldbus, OPC, HART, DeviceNet, AS-Interface or Profibus DP [6].

DeltaV offers a series of applications which allow the user to configure software modules associated with the devices (transducers, control valves, pumps etc.) from a plant.

SYSTEM STRUCTURE

In figure 1 it is presented the structure of the studied system, which contains: water tank with two compartments, centrifugal pump, manual valves (R_1 , R_2 , R_3), level transducer (LT), control valve (CV), Controller and Input/Output Subsystem of the DeltaV system, DeltaV Server, DeltaV OPC Server, computer with the control algorithm implemented in Simulink.

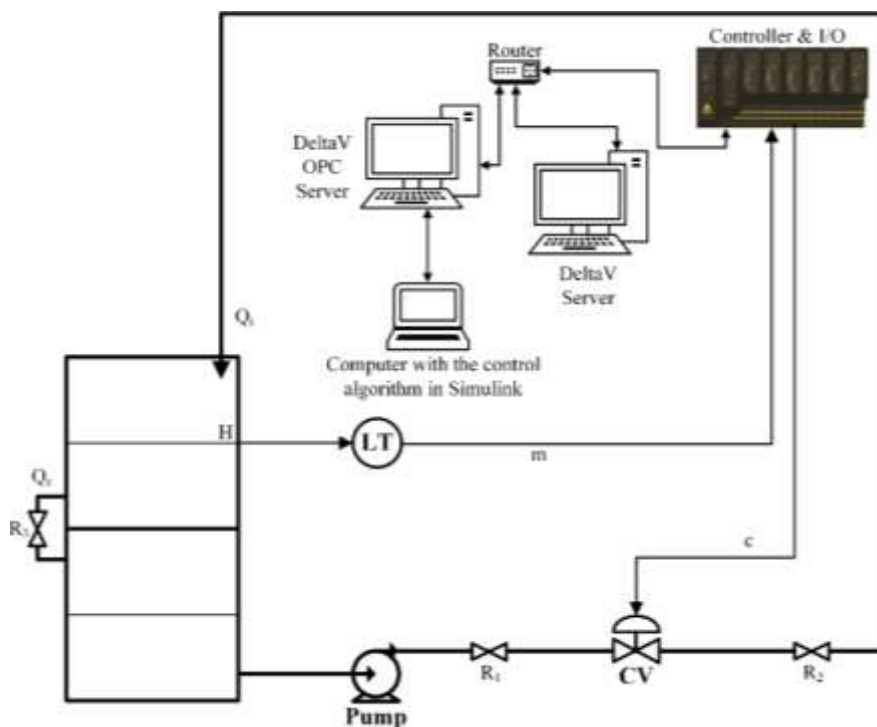


Figure 1. Structure of the studied system

The DeltaV Server is used to configure the level control system modules necessary to acquire the current value of the level from the level transducer (m) and to transmit the control signal (c) to the control valve. Also, the module for the centrifugal pump command is configured here. The level controller is implemented on the computer with

the Simulink environment. The connection between these two entities is facilitated by the DeltaV OPC Server through OPC protocol functions.

The configurations involved in the development of the level control system are presented in the following sections.

CONFIGURATION OF DELTAV SYSTEM IN ORDER TO IMPLEMENT A LEVEL CONTROL SYSTEM

As mentioned before, on the DeltaV Server three software modules must be created and configured, namely: LEVEL_TR (module associated with the level transducer), CTRL_VALVE (module associated with the control valve) and PUMP (module associated with the centrifugal pump). The creation of the modules is done using DeltaV Explorer application and the actual configuration is done using Control Studio application [5].

The **LEVEL_TR** module configuration involves the use of an analog input (AI) function block from the Control Studio library, as seen in figure 2.

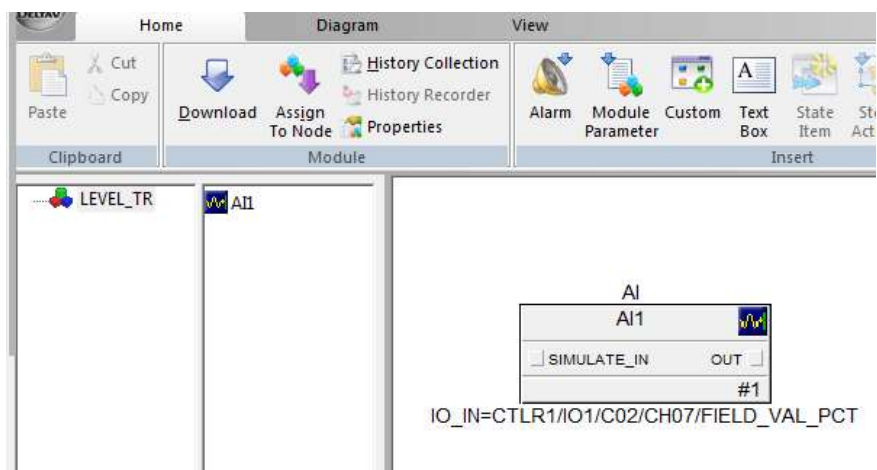


Figure 2. LEVEL_TR module.

The AI function block is used to read the current value of the allocated analog channel. In order to configure this function block, the IO_IN parameter must be modified by setting the *Device Tag* associated with the analog input channel to which the level transducer is connected.

In order to display the output of the AI function block in level units (in this case percent is used), according to level transducer measurement range, OUT_SCALE parameter is set. In figure 3 is presented the configuration of this parameter, by setting the properties: 0% of scale – 0; 100% of scale – 100; Engineering unit: %.

For the *CTRL_VALVE* module configuration, an analog output (AO) function block is used (figure 4).

The AO function block assigns an output value to a field equipment through a specified output channel. To make the connection between the function block and the analog output channel to which the control valve is connected, the *IO_OUT* parameter must have the *Device Tag* property set.



Figure 3. *OUT_SCALE* parameter of the AI block.

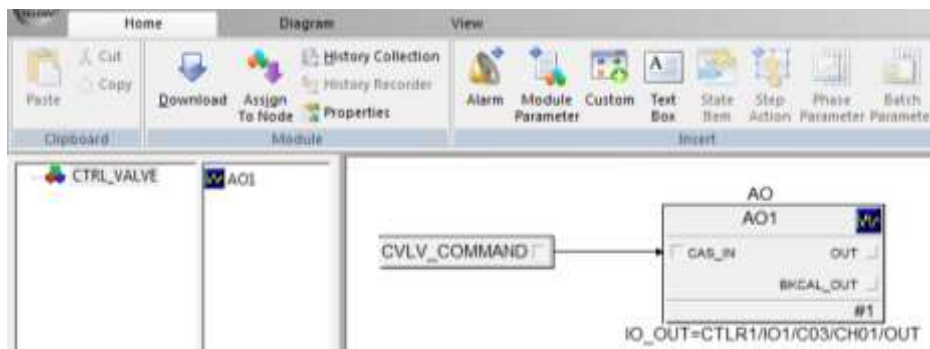


Figure 4. *CTRL_VALVE* module.

The CV_COMMAND parameter is used to change the command for the control valve more directly and easily.

In figure 5 is presented the **PUMP** module which contains a discrete output (DO) function block to transmit the start/stop command to the pump. In order to do this, the connection between the function block and the discrete output channel to which the pump is connected must be done by setting the IO_OUT parameter of the DO block with the corresponding *Device Tag*.

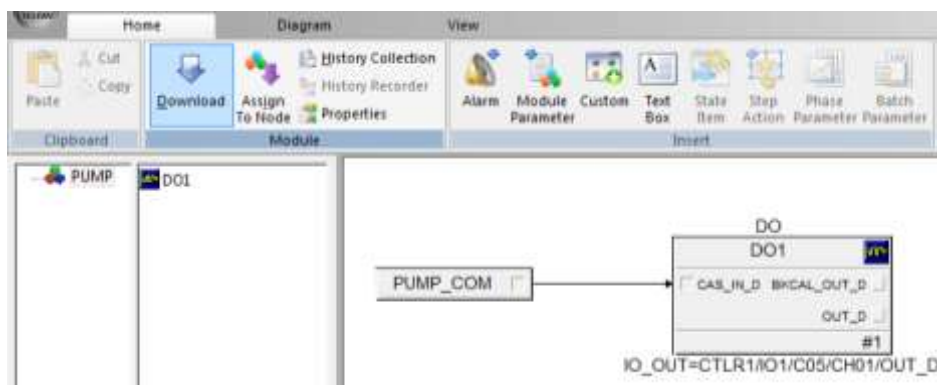


Figure 5. PUMP module.

For a more direct and easy change of the pump command, the PUMP_COM parameter is used.

CONFIGURATION OF SIMULINK FUNCTIONS IN ORDER TO IMPLEMENT THE LEVEL CONTROLLER AND THE CONNECTION WITH DELTA V

In figure 6 it is presented the Simulink model which implements the level controller and the connection with DeltaV parameters.

The *OPC Configuration* block defines the OPC clients which will be used within the model, configures the pseudo-real behavior of the model and defines the behavior for OPC errors and events. This block does not have input ports. It has an optional output port to display the latency of the model.

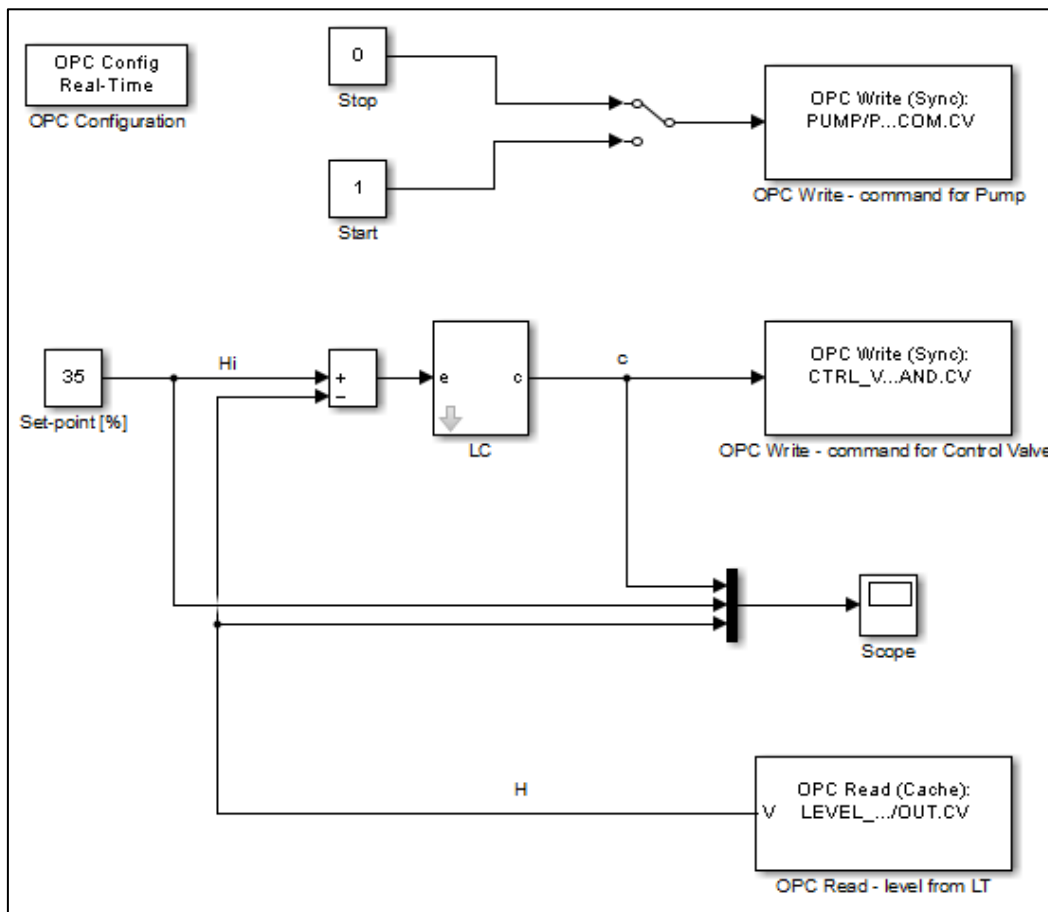


Figure 6. Simulink model.

To add the OPC client, the name of the host where the server is residing and the name of the OPC Server must be completed, the result being presented in figure 7.

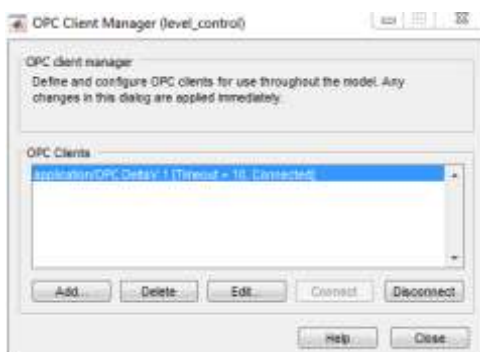


Figure 7. OPC Server added to OPC Client Manager.

The OPC Read block reads data from one or more items on an OPC server. The read operation can be done synchronously (from cache memory or from device) or asynchronously (from device). The OPC Read block (“OPC Read – level from LT”) from figure 6 is used to read the output signal of the level transducer, configured in DeltaV through the LEVEL_TR module. In order to assign a parameter to the “OPC Read – level from LT” block the selections presented in figure 8 are performed. Thus, from the DeltaV system, the LEVEL_CONTROL area is selected, and then LEVEL_TR module → All block → OUT parameter → CV (current value) → “>>” button. Practically, the output of the analog input function block from DeltaV is brought in Simulink and can be sent to the level controller as feedback variable.

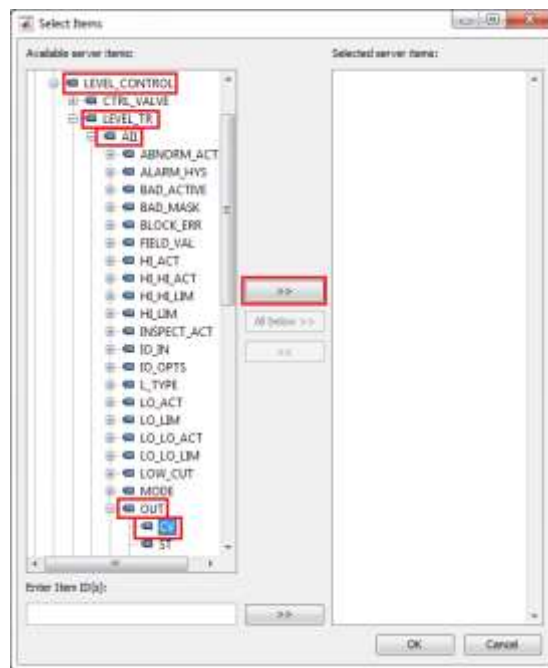


Figure 8. Parameter selection for “OPC Read – level from LT” block.

The OPC Write block writes data to one or more items from an OPC server. This operation takes place synchronously or asynchronously. Each element of the input vector is written in the corresponding element from the items defined for the OPC Write block. For this model, two OPC Write blocks were used: one to transmit the start/stop command for the pump, and the other to transmit the control signal from the level controller to the control valve.

The “OPC Write – command for Pump” block has the role of transmitting to the water pump, the on/off signal, provided by two constant blocks with the values 0 (off) and 1 (on), selected by means of a manual switch. The parameter set within this OPC Write block is assigned through the selections presented in figure 9: LEVEL_CONTROL area → PUMP module → PUMP_COM parameter → CV (current value) → “>>” button.

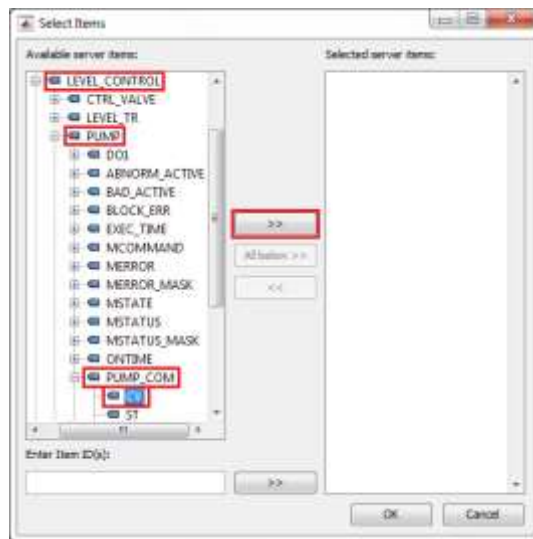


Figure 9. Parameter selection for “OPC Write – command for Pump” block.

The “OPC Write – command for Control Valve” block is used to transmit the control signal to the analog output function block from DeltaV and implicitly to the control valve. The parameter set within this OPC Write block is assigned through the selections presented in figure 10: *LEVEL_CONTROL* area → *CTRL_VALVE* module → *CVLV_COMMAND* parameter → *CV* (current value) → “>>” button.

With the configurations and connections described above the level control system development is complete: the level controller implemented in Simulink receives the signal from the level transducer through the presented connections (AI block in DeltaV and OPC Read function in Simulink), compares it with the set-point and computes the control signal sent to the control valve (OPC Write block in Simulink and AO block in DeltaV).

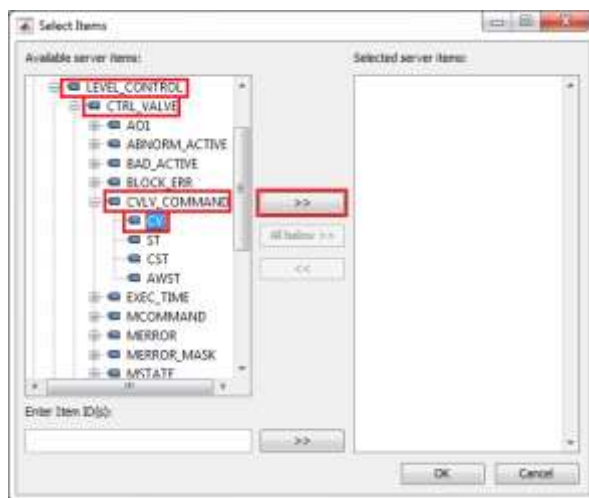


Figure 10. Parameter selection for “OPC Write – command for Control Valve” block.

TESTING THE LEVEL CONTROL SYSTEM

Using a tuning method [7], the tuning parameters of the level controller (K_p , T_i) are determined. Some system responses at step set-point changes, for different tuning parameters, are presented in figure 11.

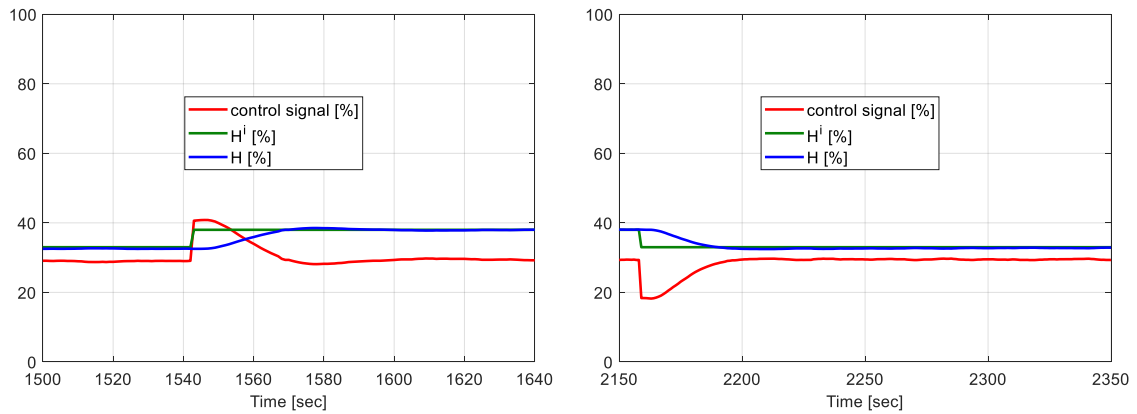


Figure 11. System response at set-point changes.

The system response to disturbance modification is presented in figure 12.

From figures 11 and 12 can be seen that the developed level control system performs very well both to set-point and disturbance changes.

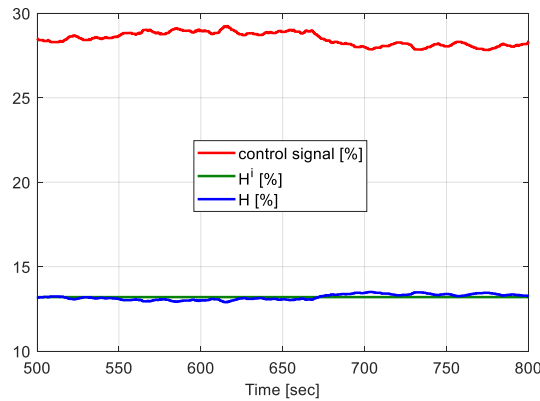


Figure 12. System response to disturbance change.

CONCLUSIONS

The paper demonstrates the possibility of transfer data in real-time between DeltaV system and Simulink environment by using OPC protocol functions. This involved the configuration of DeltaV modules for: the acquisition of the water level from level

transducer, the transmission of the control signal to the control valve, and the pump command. Also, in Simulink OPC protocol functions were configured in order to read the signal from the transducer and write (transmit) the commands for the control valve and pump by making the connection with DeltaV modules. The developed level control system was tested, and the results proved that the system responds very well to set-point and disturbance modifications, thus confirming that this is a viable solution of control system development.

REFERENCES

- [1] Sharma K.L.S., Chapter 16 - Data Communication and Networking, in Overview of Industrial Process Automation (Second Edition), Elsevier, pp. 237-261, 2017.
- [2] Knapp E.D., Chapter 4 - Industrial Network Protocols, in Industrial Network Security, Syngress, Boston, USA, 2011, pp. 55-87.
- [3] Hong X., Jianhua W., Using standard components in automation industry: A study on OPC Specification, Computer Standards & Interfaces, vol. 28 (4), pp. 386-395, 2006.
- [4] OPC Foundation, <https://opcfoundation.org/about/opc-technologies/opc-ua/>
- [5] Popescu M., Sisteme distribuite de supervizare și control. Îndrumar de laborator, Editura Universității Petrol-Gaze din Ploiești, Romania, 2018.
- [6] Paraschiv N., Popescu M., Sisteme distribuite de supervizare și control, Editura Universității Petrol-Gaze din Ploiești, Romania, 2014.
- [7] Skogestad S., Simple analytic rules for model reduction and PID controller tuning, Journal of Process Control, 13, pp. 291–309, 2003.