2015 20th International Conference on Control Systems and Science

# *Environment communication and control systems integrated on teaching platforms*

Case study: double water tank system

Alexandru Dumitraşcu, Cezar-Ştefan Istrătescu, Dan Ştefănoiu, Janetta Culiță
Department of Automatic Control and System Engineering
University Politehnica of Bucharest
Bucharest, Romania
E-mails: {alexandru.dumitrascu,dan.stefanoiu,janetta.culita}@acse.pub.ro

***Abstract*—This paper introduces a solution for integrating environment systems on teaching platforms using two components: communication and primary control. The communication relies on a bus and specific protocol. The primary control algorithms were implemented in MATLAB-SIMULINK environment, while using Siemens S7-1200 PLC as acquisition server and some regular PC as a client, communicating to each-other through the Modbus-TCP protocol. The teaching platform is represented by the double water tank system ASTANK2, after being equipped with sensors and actuators.**

***Keywords-PLC control system; Modbus-TCP; communication bus; double water tank system.***

## I. INTRODUCTION

Many process control applications commonly integrate liquid level and flow control systems. Therefore, in the last two decades, control engineering community has shown an increasing interest in the multi-tank systems, for research and control education purpose. Due to their multi-variable and nonlinear characteristics, various level control strategies are approached in the literature. Thus, they are ranging from conventional PI (as in [9] and [14]) to adaptive and fuzzy logic PID (as in [11]) or, more complex, multi-variable robust control and non-linear or Distributed Model Predictive Control (DMPC) (as in [10] and [5]).

Few multi-tank laboratory processes have been addressed in the literature so far. They usually consist of two, three or four tanks, with cylindrical or rectangular shapes, which are interconnected in series or parallel setup. Perhaps the most reported installation is the quadruple rectangular tank process, early introduced in [9] and furthermore employed in [12], [10] and [5], in order to illustrate concepts in multi-variable control. In [13], a hybrid control scheme is designed and implemented for a two-tank system. Besides the regular shape of tanks, there have been experienced conical tanks (as in [6]), whose variable cross-section introduces a nonlinearity in the plant model. In [11], a comparison between the adaptive PID control and the model predictive control algorithms is outlined on a three conical tank process. Actually, many papers are involved within the evaluation, testing or comparing the performances of different control solutions. However, they rather are designed and implemented on simulated plants than on real ones.

A less addressed problem related to the real laboratory tank system is the communication protocol designed to bridge the process control infrastructure (usually based on PLCs) and the Windows based applications, in view of advanced control strategies implementation. Thus, control applications developed in MATLAB-SIMULINK environment usually require the use of an OPC server, to which one communicates through Ethernet. To the best of our knowledge, only few works on control implicitly address the communication problem in such plants. For example, in [9], the physical process was extended with a PC interface, which allows loading and running a MATLAB control application (a PI controller, in fact) on the real multi-tank system. In [13], a hybrid control scheme for a real two-tank system was designed and implemented directly on a PLC (of 1768-L43 CompactLogix type), by using ladder diagram. In [14], a real three-tank system is PI controlled in real time, by using a standard industrial SIMATIC PLC connected to MATLAB-SIMULINK. Here, the communication between MATLAB and PLC is based on OPC protocol and the MATLAB application runs on the OPC server. Also an OPC server is used for implementing advanced control algorithms on a modified quadruple tank laboratory installation presented in [5], whilst the low level controller PID are implemented on PLC. Another didactical installation is Multitank System (belonging to INTECO Ltd. [2]), which consists of three-tanks system interconnected in cascade. The communication protocol between PC and I/O board is supported by their own product, namely RT-CON.

This paper mainly focuses on a solution for integrating a practical multi-tank system (namely ASTANK2) in MATLAB-SIMULINK environment. As its name suggests, ASTANK2 (manufactured by ASTI Automation [1]) represents a hydraulic system designed for communication and control education, which consists of two serially coupled tanks: a rectangular one and a sloped wall one. The communication protocol between the PLC (of SIEMENS S7-1200 type) and a portable PC integrating MATLAB-SIMULINK application is based on Modbus-TCP protocol, which is tested by using two conventional PI control algorithms.

One has to outline that this article rather approaches the communication problem than the plant control, which is only left for demonstrative purposes (as the designed controllers are non optimal). The real complexity of the proposed solution lies

in the implementation of the communication protocol Modbus-TCP in MATLAB and its synchronization with SIMULINK environment. Due to the implementation of this specific communication protocol, the proposed solution proves to be more efficient than when using the standard OPC protocol.

The method described within the article is mainly targeting the data acquisition and not necessarily the controller design. One assumes that, for the auto-stabilizing systems (like ASTANK2), the control strategy can be implemented on the PC client with good performance.

The paper is structured as follows. The next section introduces ASTANK2 installation and describes the acquisition plus the control systems. Section 3 presents the communication network topology. The basic functions of the communication protocol between the PLC and MATLAB-SIMULINK environment are outlined in section 4. Section 5 illustrates the performance of the implemented protocol and the validation results. Some concluding remarks are given in section 6.

## II. THE ASTANK2 DIDACTICAL PLANT

### A. *Architecture of ASTANK2*

The picture in Figure 1 illustrates the ASTANK2 installation.

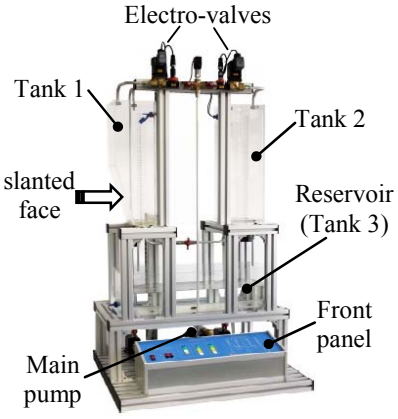


Figure 1. ASTANK2 – a double-tank water installation.

As one can notice, one tank is a parallelepiped, while the other one has a slanted face, in order to introduce nonlinearities in water level variation. Both tanks are supplied with water from a third tank (a reservoir, in fact), which is located beneath them. The upper electro-valves yield to control the water level into the tanks, while the main pump behind the front panel pushes the water from the reservoir to the tanks. The two main tanks can communicate to each-other through a median pipe endowed with a manual switch. Overall, this plant allows performing a broad range of hydraulic pressure experiments: tank filling and drainage, coupled vessels or pipe pressure variation processes.

Signals from sensors and actuators are available on the front panel, through 4 mm insulated connectors, as shown in Figure 2. Analog signals have been converted to 0-10 V range for convenient use [4].

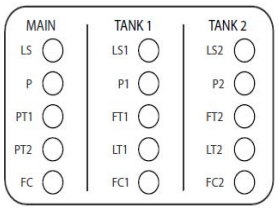


Figure 2. The front panel scheme of ASTANK2.

The acronyms of the platform front panel are listed in Table I, together with their meanings and some technical details.

TABLE I. SIGNALS LIST AND VARIATION RANGES IN ASTANK2

| Label | Name | Direction | Signal | Description |
|---|---|---|---|---|
| **Tank {1,2}** | | | | |
| LS{1,2} | Level switch | Out | 0/24V | Active high |
| P{1,2} | Auxiliary pump | In | 0/24V | on/off |
| FT{1,2} | Flow transducer | Out | 2-10V | 0.4-12L/min |
| LT{1,2} | Level transducer | Out | 2-10V | 0-0.1bar |
| FC{1,2} | Proportional valve | In | 0-10V | 0%-100% |
| **Reservoir** | | | | |
| LS | Level switch | Out | 0/24V | Active low |
| **Main pump** | | | | |
| P | Inverter | In | 0/24V | on/off |
| FC | Inverter | In | 0-10V | 0%-100% |
| PT1 | Pressure transducer | Out | 2-10V | 0-1.6bar |
| PT2 | Pressure transducer | Out | 2-10V | 0-1.6bar |

Thus, the Tank{1,2} labels are as follows: LS{1,2} – normally open level switches; P{1,2} – auxiliary centrifugal pumps of EHEIM 1046 type; FT{1,2} – flow transducers of Kobold DPL-1P20 type (which is able to accurately measure low flow rates); LT{1,2} – level transducers of Sitrans type (which are very accurate and reliable); FC{1,2} – control electro-valves of Burkert type (which are actuated solenoids proportionally working in 0÷0.7 bar range). The other labels refer to the *reservoir* and *main pump*. More specifically: LS – a normally closed level switch; P – the main pump, whose engine is driven with a Siemens Sinamics G110 inverter; PT{1,2} – two pressure transducers with Sitrans P210 series static pressure sensor; FC – an inverter.

The level transducers LT1 and LT2 are represented by pressure sensors. In case of tank 2, the pressure can easily be translated into water column height, by means of the well known hydrostatic pressure formula:

$$p = \rho g h + p_0 \quad \Leftrightarrow \quad h = \frac{p}{\rho g} - h_0 \,. \tag{1}$$

In equations (1), $p_0$ is the atmospheric pressure, while $h_0$ is a correction factor that depends on $p_0$. Nevertheless, $h_0$ will be determined experimentally. Thus the level linearly depends on

the pressure. In case of tank 1, the pressure-level equation is more complicated and nonlinear, because of the slanted wall. (Analytical modeling of ASTANK2 is the goal of another article, see [7].)

Figure 3 displays the schemata of ASTANK2 architecture.

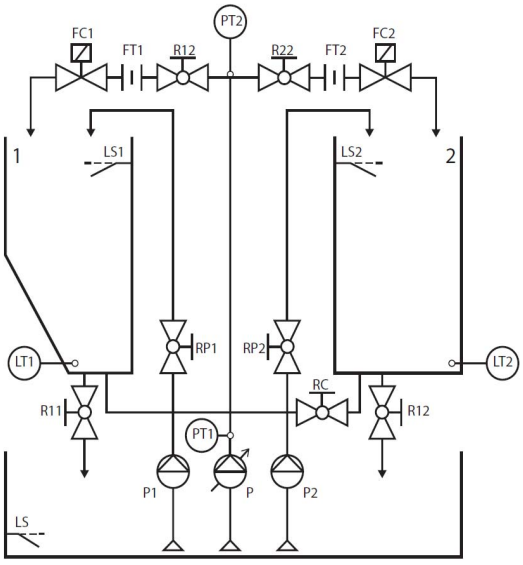


Figure 3. Schematic architecture of ASTANK2.

In the figure, the manual valves Rnn (e.g. R12) are employed to modify the plant configuration, in order to allow working with different physical models. In the example of section V, the manual valves RC (between the two tanks) and R12 (yielding to supply the tank 1 with water) are closed, in order operate with tank 2 only.

### *B. The Acquisition and Control System*

In order to facilitate signal handling, the Siemens PLC S7-1200 controller of Figure 4 has been integrated. Its characteristics are the following: CPU 1214C; 14/10 digital I/O signals; 4/2 analog I/O signals [8].

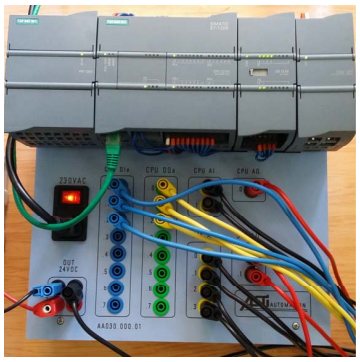

Figure 4. Photo of Siemens PLC S7-1200 wired to ASTANK2.

The PLC works as a server in the network with one client. The client can be any PC endowed with a proper hardware configuration to run MATLAB scripts and some network board. Such requirements are satisfied by almost all the commercial regular computers.

Although the implementation of algorithms was made in MATLAB 2013b environment, the system can work on previous versions as well.

A number of signals are received at the board level of S7-1200 PLC. Therefore, the signals from ON/OFF level switches (LS, LS1, and LS2) are connected to the first three digital inputs of the PLC (see the blue wires on the left side jacks column). The signals from pumps (P, P1, and P2) are connected to the first three digital outputs of the PLC (see the yellow wires on the middle jacks column). The signals from pressure, flow, and level transducers (PT1, PT2, FT1, FT2, LT1, and LT2) are connected at the six analog inputs (see the black wires on the right side jacks column). The signals from inverter (FC) and proportional valves (FC1 and FC2) are connected at the three analog outputs of PLC (see the red wires to the right of the jacks matrix).

Nearby the S7-1200 PLC, a specific switch unit for Ethernet connections is installed. It will be used to expand the application by connecting a HMI device type – a color touch screen operator panel KTP 600 series.

## III. NETWORK TOPOLOGY

The network is composed of the ASTANK2 plant, the S7-1200 PLC and a regular computer.

The PLC is connected to the I/O ports of the plant with regular 4 mm connectors. Its tasks are to acquire the outputs and to set the inputs in the system. An Ethernet twisted pair cable (see the green cable just above the left side jacks matrix in Figure 4) is connecting the PLC to the PC. The Ethernet communication protocol is employed to configure, transfer and monitoring the control program into the PLC [15].

The client-server architecture has been employed to configure the PC-PLC network. The PC plays the client role, while the PLC is the server. Therefore, hereafter, the word *Server* will be used to refer to the Siemens PLC, whilst the word *Client* stands for the PC (i.e. the client MATLAB application running on it).

Since the Server supports TCP-IP protocol over Profinet, a TCP related protocol for communication has been adopted. More specifically, the Modbus-TCP is the selected protocol, due to the fact that it can operate with acknowledgments and, moreover, it can easily manipulate sending and receiving data functions [8].

A basic acquisition and control cycle consists of the following steps:

1. The Server reads and saves in memory all digital and analog output values of the plant.
2. A reading request is received from the Client.
3. The Server responds with the saved values at step 1.
4. A writing request is received from the Client along with the data to be written.
5. The Server saves the received data from the Client and sends a completion acknowledgment.
6. The Server sets the plant input ports to the values received from the Client.

## IV. IMPLEMENTATION

### A. The S7 program

A 100 ms cyclic interrupt routine has been designed in order to update the I/O values and to synchronize them with the Modbus Server register employed in communication.

Normalizing and scaling of analog values are applied in the PLC. Thus, the output sensor values are converted to their corresponding units and the input control signals are expected to vary in range of 0-100%. Two function blocks, namely the *input_interface* and the *output_interface* were implemented in order to separate the raw analog values from their counterparts.

The default Modbus Server block provided for the S7-1200 controller is in charge with the main program loop.

### B. The MATLAB program

A MATLAB functions library has been designed, in order to perform communication [3]. The library includes the following routines:

1. *ModbusTCP_open()* – performs initialization of a TCP socket for the default Modbus port 502, by setting the byte order to big Endian for compatibility with the Server.
2. *ModbusTCP_close()* – closes the Client side of the connection, by removing the TCP socket.
3. *ModbusTCP_put()* – implements the function with code '6' from the Modbus protocol; its task is to save a 16-bit word into a Modbus register.
4. *ModbusTCP_write()* – implements the function with code '16' from the Modbus protocol; its task is to save several 16-bit words in an array of registers.
5. *ModbusTCP_read()* – implements the reading data from the Server, i.e. the function with code '3' from the Modbus protocol; its task is to read an array or registers from the Modbus Server (after being written by the previous function).

This library is integrated into a graphic acquisition interface, as well as into a SIMULINK function block. The reading and writing functions are implemented such that conversions to and from the data types supported by the Server are included. Also, the implementation allows reading/writing of integers, floating point values and date formats.

### C. The Matlab-Simulink project

The MATLAB acquisition interface shown in Figure 5 allows quick modifications of the control elements (valves and pump inverter frequency) and live display of sensor values (where each sensor can be triggered by ticking the corresponding check box). The time used for the abscissa in the plot is read in online manner, together with the data. All information is quite accurate, since the Server is a real time working system.

A SIMULINK block that mirrors the behavior of the plant has been designed on the basis of a level 2 S-function block. This block aims to help the user to create a simulation setup.

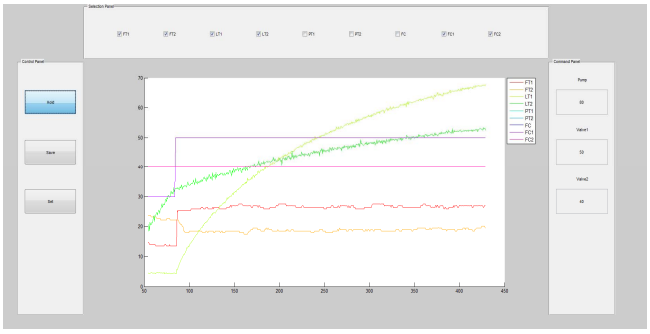

Figure 5. MATLAB acquisition interface.

In this case, a discrete block is necessary. The block has to send the input ports to the Server (for the plant control purpose) and to get the output port values from the acquisition interface.

The S-function package includes the following routines [4]:

1. *Setup()* – integrates the other functions.
2. *DoPostPropSetup()* – initializes the memory (*Dwork* vectors) and starts the Client socket.
3. *InitializeConditions()* – initializes the block memory and performs the first reading of sensor values.
4. *Outputs()* – updates the discrete outputs of the system with the values saved in the block memory (*Dwork*).
5. *Update()* – manages the reading and writing commands (being the most important routine of the package); the input values of the block are sent to the plant; a temporary break is taken (through the *pause* function), if necessary, in order to get a sampling period close to 1 s.
6. *Terminate()* –cleans up the socket and puts on hold the plant actuators.

Functions 4 and 5 are called by the SIMULINK engine at every step of the simulation (in the main loop). The other functions are only called once.

## V. CASE STUDY

Through the case study to be presented next, one aims to demonstrate how the proposed method works.

A simple configuration of the plant has been selected, in order to better understand the overall algorithm. The communication and control problem has been focused on tank 2 only (due to its regular shape, which involves linear pressure-level variation). Thus, the manual valves RC and R12 are closed. The tank drainage is gravitational.

The Figure 6 illustrates the external view of the Simulink block associated to ASTANK2 model. According to the case study setting, the only used ports will be InPump, FT2 and LT2. The FC2 valve is set to 100%, so that the only input in the system is the pump inverter frequency (normalized in 0-100% range). The flow is acquired through FT2, whilst the level is acquired through LT2. A cascade controller has then been set.

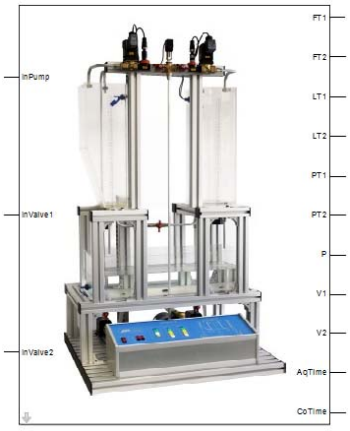


Figure 6. S-function implementing ASTANK2.

A step input excites the pump, whereas FT2 is displayed by means of a scope, in order to identify the flow model. This can subsequently be employed to design the PI flow controller by the pole-zero cancellation method. The same technique has been employed for the level model. The two PI controllers are cascaded as shown in Figure 7.

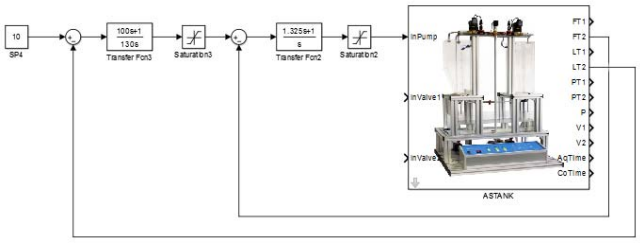


Figure 7. The cascade control algorithm.

After the first simulation, the controllers transfer functions have been set to (with natural notations):

$$H_f(s) = \frac{1.325s+1}{s} \quad \& \quad H_l(s) = \frac{100s+1}{130s}. \qquad (2)$$

Although the PI controllers are not necessarily optimal, they are known to be very robust and reliable. (Recall that the goal here is not to design optimal controllers, but to demonstrate the communication method effectiveness.)

As it can easily be noticed, the PI controllers are implemented through continuous transfer function blocks. Better performance could be obtained if they would have been implemented as discrete time systems, by using some discretizing method (e.g. the Tustin's one). The main problem with the discrete time controllers is that no constant sampling time can be set for the entire ASTANK2 plant. Fast and slow component blocks are included into the plant. So, in case a discrete time solution is required, the SIMULINK scheme has to work with variable sampling time, which might introduce parametric uncertainties in the model.

Two saturation blocks are necessary, in order to match the physical limitations of the pump inverter. They introduce some non linear behavior, in case the plant is working at the limit of its capabilities.

The ModbusTCP communication protocol implemented on a client-server structure needed higher implementation effort than the control algorithm.

Figure 8 displays the response of the controlled system to a series of step references.

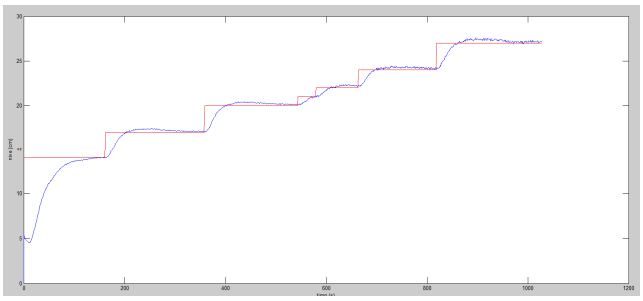

Figure 8. The PI controlled response signal of the level (smooth, in blue) and the reference signal (successive steps, in red).

The performance is fair, but the communications system is successfully accomplishing its task. Perhaps, better performance would have been obtained with discrete time blocks, or just by directly implementing the PI algorithms on the Siemens PLC. Recall however other (more complex) control algorithms than the PI one are difficult (if not impossible) to implement on a PLC, which is not compiling high level programming routines. Therefore the PC is a necessary and important component of the entire installation.

This method offers the obvious advantage of easiness in the both in the design and the implementation phases. It is much more suitable to use a high level programming language such as MATLAB, in order to acquire data and a simulation platform such as SIMULINK, in order to implement controllers. Their powerful features and rich routines libraries allow the user to sensibly reduce the design effort and duration than in case one may want to use data logs and to directly tune the Siemens controllers. For didactical and research purposes, the client-server solution is highly recommended.

To respond if the communication delay can affect the closed loop stability, the answer is: not really. This is due to the fact that the process time constants are at least 10 times greater than any possible communication delay (which, nowadays, does not exceed 100 ms). The models for both the flow and the level of tank 2 can easily be approximated by first order systems with corresponding gains ($K_f$, $K_l$) and time constants ($T_f$, $T_l$). In case of ASTANK2, such parameters take reasonable values (e.g. $K_f \in [0.5,1.5]$ and $T_f \in [1,5]$, for the flow modeling system). It can be proven (see the next rationale) that, if $K \le 1$, the delay margin is infinite, whereas, if $K > 1$, the delay margin is measurable in seconds (for ASTANK2). This proves that perturbations around 100 ms cannot really affect the system stability. The communication delay can become comparable with the system delay margin only if the controller is located in another network than the one directly connecting to the plant. In this case, the routing delay can become relevant. To avoid transmitting delayed commands, the Smith predictor can easily be employed.

To estimate the delay margin, the reduced Nyquist criterion can be used, in case of first order systems. Thus, it suffices to find the delay margin $T_d$ and the pulsation $\Omega_d$ for which:

$$\left|G\left(j\Omega_d\right)\right| = 1 \quad \& \quad \arg\left(G\left(j\Omega_d\right)\right) = -\pi, \tag{3}$$

where $G$ is the transfer function of the delayed first order system $H$ with parameters $K > 0$ (gain) and $T > 0$ (time delay):

$$G\left(j\Omega\right) = H\left(j\Omega\right)\mathrm{e}^{-j\Omega T_d}, \quad \forall \Omega \in \mathbb{R}. \tag{4}$$

After some elementary manipulations applied on equations (3), it follows that:

a. if $K \le 1$, then $T_d = \infty$;

b. otherwise (if $K > 1$), then $T_d = \dfrac{\pi - \arctan\left(\Omega_d T\right)}{\Omega_d}$, where $\Omega_d = \dfrac{\sqrt{K^2 - 1}}{T}$.

In case of ASTANK2, the results above lead to delay margins measurable seconds, while the communication delays are 10 times smaller. (The delay margin can also be computed with the MATLAB function *allmargin*.)

## VI. CONCLUSIONS

In this paper, an industrial communication method aiming to link the MATLAB-SIMULINK environment with a didactical plant (ASTANK2) is introduced. The method employs a PLC as acquisition *Server* and a PC as *Client*. They communicate to each-other through the MODBUS-TCP protocol. A MATLAB library has been designed, in order to allow the client to use Modbus, while a SIMULINK functional block has been created to manage the system states. The method is fast and provides good performance if correctly implemented.

Comparing to solutions that require the installation of an UTP server on the PC for communication, this approach reduces the overhead caused by the server and generally is faster, thanks to MATLAB-SIMULINK features. Improvements may be obtained if pre-compiled C functions are employed and a real-time system is set up for the client.

## ACKNOWLEDGMENTS

Although the idea of constructing ASTANK2 belongs to the third author, the physical realization of the plant is due to ASTI Automation S.R.L. Therefore, all four authors are very grateful to the team of ASTI professionals who successfully constructed this didactical installation (following the initial specifications). Special thanks are addressed to "Politehnica" University of Bucharest for supporting purchasing ASTANK2, which thus became an asset enriching the Faculty of Automatic Control and Computers research basis.

## REFERENCES

[1] *** – ASTI Automation SRL: http://www.astiautomation.ro.

[2] *** – INTECO Ltd.: http://www.inteco.com.pl

[3] *** – modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf.

[4] ASTI Automation, "Technical handbook of double water tank system.

[5] Alvaradoa I., Limona D., Muñoz de la Peña D. et al., "A Comparative Analysis of Distributed MPC Techniques Applied to the HD-MPC Four-Tank Benchmark", in Journal of Process Control, Vol. 21, Issue 5, pp. 800–815, June 2011.

[6] Christy Y., Dinesh Kumar D., "Modeling and Design of Controllers for Interacting Two Tank Hybrid System (ITTHS)", International Journal of Engineering and Innovative Technology (IJEIT), Vol. 3, Issue 7, pp. 88-91, January 2014.

[7] Culita J., Stefanoiu D., Dumitrascu A., "ASTANK2: Analytical Modeling and Simulation", Submitted to the 20th International Conference on Control Systems and Computer Science, Bucharest, Romania, May 27-29, 2014.

[8] Dumitrascu A., Stefanoiu D., Tomita I., "A new Challenge in Ecological Process Control based on PLCs with Profinet Communication Protocol", The 19th International Conference on Control Systems and Computer Science, Bucharest, Romania, IEEE Computer Society Publisher, pp. 285-288, May 29-31, 2013.

[9] Johansson K.H., "The Quadruple-Tank Process: A Multivariable Laboratory Process with an Adjustable Zero", IEEE Transaction on Control System Technology, vol. 8, no. 3, May 2000.

[10] Ma M., Chen H., Findeisen R., Allgower F., "Nonlinear Receding Horizon Control of Quadruple-Tank System and Real-Time Implementation", International Journal of Innovative Computing, Information and Control , Volume 8, Number 10(B), pp. 7083-7093, October 2012.

[11] Pushpaveni, T., Srinivasulu Raju S., Archana N., Chandana M., "Modeling and Controlling of Conical Tank System Using Adaptive Controllers and Performance Comparison with Conventional PID", International Journal of Scientific & Engineering Research, Volume 4, Issue 5, pp. 629-635, May 2013.

[12] Qamar S., Vali U., Reza K., "Multivariable Predictive PID Control for Quadruple Tank", World Academy of Science, Engineering and Technology (0043), pp. 861-866, 2010.

[13] Stanga F., Soimu A., "Hybrid Modeling and Control for a Two-Tank Systems", Scientific Bulletin of „Gheorghe Asachi" Technical Institute, Iasi, Romania, vol. 56, no. 4, 2010.

[14] Voglauer B., Garcia R., Jorgl P., "Improvements of a Three-Tank System Operated in Real Time With MATLAB in a PLC-PROFIBUS-Network", Proceedings of the 16th IFAC World Congress, Prague, Czech Republic, pp. 2279-2285, 2005.

[15] Zurawski R. (editor), "Industrial Communication Technology Handbook", Second edition, ISA Group, CRC Press, San Francisco, California, USA, 2015.