Modeling and control of processes using the EMSO simulator and SIMATIC PLC as OPC inter-face

Modelado y control de procesos usando el simulador EMSO y PLC SIMATIC como interfaz OPC

Jonathan Ospino-Pinedo[1], Mauricio Esteban Sánchez[1*], Luis Gerónimo Matallana-Pérez[1]

[1]Grupo de investigación en Simulación, Diseño, Control y Optimización de Procesos (SIDCOP), Facultad de Ingeniería, Universidad de Antioquia. Calle 67 # 53-108. A. A. 1226. Medellín, Colombia

ARTICLE INFO:
Received October 31, 2016
Accepted May 08, 2018

KEYWORDS:
EMSO simulator, process control, educational tools, PID and PBSM controllers

Simulador EMSO,control de procesos, herramientas educacionales, controladores PID y PBSM

ABSTRACT: This work shows a procedure for modeling and control of level and flow processes. It uses the EMSO simulator to connect to a process by means of a SIMATIC S7-200 PLC as interface for the communication between the process instrumentation and the controller computing system. In order to supervise the variables, a SIMULINK model was connected to a laboratory process module through OPC communication. The model obtained was used in the design of controllers PID and PBSM, which were tested for step disturbances of the set point and an imposed set point trajectory.

RESUMEN: Este trabajo ilustra un procedimiento de modelado y control de procesos de nivel y flujo usando el simulador EMSO acoplado al proceso con un PLC SIMATIC S7-200 como interfaz de comunicación OPC entre la instrumentación del proceso y el sistema de cómputo del controlador. Para la supervisión de las variables del proceso se utilizó un modelo en SIMULINK conectado al proceso a través de comunicación OPC. El modelo obtenido fue utilizado en el diseño de controladores PID y PBSM, los cuales fueros probados ante saltos de set point y una trayectoria de set point impuesta.

1. Introduction

The development of reliable and powerful process simulators, with good databases and ease of communication is mainly carried out by private and foreign enterprises. Such companies put simulators on the market at higher prices, and most of the times, inaccessible for mid-range and small-range domestic enterprises, public higher education institutions, and new research groups. The EMSO (*Environment for Modeling, Simulation, and Optimization*) can be considered as a good option among those of free access (e.g. COCO simulator, Ascend, etc.) and it has been developed as *free software* by the ALSOC Project, which is financed by the Brazilian Government and certain private and public institutions. Several publications validate the growing importance of the EMSO software [1–6]. This simulator is a graphical environment where the user can easily carry out the tasks of modeling, simulation, optimization, parameter estimation, and data reconciliation, among others. The software allows both the development of code-based models and the development of models composed graphically by connecting several built-in models from the EMSO Model Library (EML) or those created previously by the users themselves. A previous work shows the application and the advantages of using EMSO as a computational tool for assisting a course of chemical engineering reactions [7]. In addition, EMSO allows data communication with other software and/or devices, such as MATLAB® and Scilab® (free use), by means of external interfaces like the EMSO-MATLAB-SCILAB [8]. Another advantage of this software is the possibility of communicating with other devices such as the PLC (Programmable Logic Controller) via OPC communication and an interface called EMSO-OPC. OPC (stands for *Object Linking and Embedding for Process Control*) makes reference to a communication standard in the field of control and supervision of industrial processes. It is based on a Microsoft® technology that has a common interface for communication to allow standalone software components to interact and share data. OPC communication is done through client-server architecture. The OPC server is the data source (like a hardware device located in the plant) and any application compatible with OPC communication can have access to the server in order to read/write any variable available within it. This approach is an open and flexible solution to the classic problem of proprietary drivers. Practically all major developers of control systems, instrumentation

* Corresponding author: Mauricio Esteban Sánchez
E-mail: mauricio.sanchez@udea.edu.co
ISSN 0120-6230
e-ISSN 2422-2844

DOI: 10.17533/udea.redin.n87a08

and processes have included OPC in their products. The present work illustrates the procedure for the modeling and control of a process by using the EMSO simulator and a SIMATIC S7-200 PLC as an OPC communication interface. This document is organized as follows:

Section 2 describes the methodology used for obtaining the phenomenological-based models and the implementation of control systems via OPC. Section 3 describes the statement of a Phenomenological-Based Semiphysical Model (PBSM), estimation of unknown parameters, the validation of the model obtained and the implementation of control systems using EMSO. Finally, section 4 gives the conclusions of the paper.

2. Methodology

2.1 Equipment and Instrumentation

The following resources were used in the development of this project:

(1) a control laboratory module for level and flow control,

(2) a SIMATIC S7-200 Programmable Logic Controller (PLC),

(3) a TCP/IP connection cable,

(4) a laptop computer,

(5) the EMSO simulator and its OPC interface, EMSO-OPC, and

(6) Siemens S7-200 PC-Access software (needed for setting up an OPC server using the PLC controller).

The methodology of the current work can be divided into two fundamental stages: (1)the process model statement and (2) the process control via OPC communication.

2.2 Process model statement

In this paper, the methodology proposed by Álvarez *et al.* [9] is used to obtain Phenomenological-Based Semiphysical Models (PBSM) by using empirical correlations and basic process principles of mass, energy and momentum balances. The main steps proposed by the authors are listed below, which were followed for process modeling in this work:

1. Write a verbal description and draw a process flow diagram for the process to be studied.
2. Fix a level of detail to be used in the model.
3. Define many process systems (PS) over the whole process to be modeled as required by the level of detail specified in the previous step and sketch a block diagram showing the different process systems and its interactions.
4. Apply the conservation principles of mass, energy and momentum over each process system.
5. From the conservation principles, select those balance dynamic Equations (BDE) with relevance to accomplish the objective of the model.
6. Define the known parameters, variables and constants for the relevant BDEs in each process system.
7. Find constitutive Equations for computing many parameters as possible in each process system.
8. Check the degrees of freedom of the model.
9. Obtain the computational model or model solution.
10. Validate the model under different conditions to assess its performance.

The modeling method described by the aforementioned authors was carried out to develop a mathematical model for a laboratory process module to control level and flow. The system modeling, the parameter estimation and model validation were done using the EMSO simulator as described in sections 3.1 and 3.2.

2.3 Process Control via OPC communication

With the aim of designing and analyzing the behavior of different control systems implemented in a process via OPC, mathematical models were developed for PID and PBSM-Based controllers. This was done to establish a clear and concise methodology to implement control and supervision systems in process modules by using the EMSO-OPC application. Firstly, it is worth mentioning that although the EMSO-OPC application allows the implementation of control systems via OPC using models written in the EMSO modeling language, it does not (currently) offer any capabilities to plot the response of process variables (supervision). Therefore, an additional OPC-based application is required to plot the results. In the present work, the software SIMULINK (from MATLAB) was used as an OPC client for the reading and visualization of the process variables. In addition, a SIMATIC S7-200 PLC was employed as an interface for OPC communication between the process instrumentation and the mathematical computing model of the control system. Figure 1 shows a diagram of the connections used in the laboratory. The different steps for obtaining and implementing a control system using EMSO and

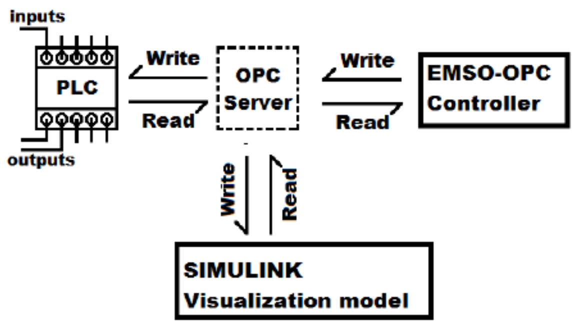


Figure 1 Connections diagram

EMSO-OPC are described below. In order to illustrate the procedure used, a case study that implements a model-based control over a flow module was considered, including the following stages:

1. Create a mathematical model for the process controller.
2. Configure the OPC server.
3. Set the connection between the variables of the EMSO model and variables of the OPC server.
4. Create an OPC client application to visualize the process variables via OPC connection.
5. Implement the control system start up the process.

The methodology described was carried out to implement two different controllers for the studied process module. Section 3.3 shows the detailed implementation of the process control system.

3. Results

3.1 Modeling of a laboratory module for controlling flow and level

The Flow Control Module (see Figure 2) basically involves a piping system connected to two interacting tanks that are filled with water and then emptied in order to analyze and/or control the tank filling process. The module contains the following process elements: two gas cylinders (TK-01 y TK-02), an orifice plate (FE-01), an U-tube manometer (PDI-01), a flow switch (FS-01), a flow transmitter (FT-01), three differential pressure cells (LT-01 to LT-03), two level gauge indicators (LGI-01 y LGI-02), two pressure-current transducers (LY-01 and FY-01), two current-pressure transducers (LY-02 and FY-02), four pressure gauge indicators (PI-01 to PI-04), a pneumatic control valve (LCV-01), an electric-actuated control valve (LCV-02), two centrifugal pumps (P-01 and P-02), a recycling tank (TK-03), two PID controllers (LIC-01 and FIC-02), and several manual valves (V-01 to V-23) to interact with the flow directions in the whole system.

The process module described can be set to different operational modes for filling and draining of the tanks, depending on the configuration adopted for the valves V-19 to V-22 (valves V-20, V-21, and V22 are also referred as VF3, VF4, and VF5, respectively, in Equations (8) - (10) of the process model). In total, the module can be set in 16 different ways, but if the systems with the pressurization valve (V-19) are omitted, the total number of operational modes can be reduced to the following 9:

1) Use of tank TK-01 only and considering gravity discharge.

2) Use of tanks TK-01 and TK-02 with gravity discharge through the TK-01 discharge.

3) Use of tanks TK-01 and TK-02 with gravity discharge through the TK-02 discharge.

4) Use of tanks TK-01 and TK-02 with gravity discharge through both discharges.

5) Use of tank TK-01 with pumping discharge.

6) Use of tanks TK-01 and TK-02 with pumping discharge through the TK-01 discharge.

7) Use of tanks TK-01 and TK-02 with pumping discharge through the TK-02 discharge.

8) Use of tanks TK-01 and TK-02 with pumping discharge through the both discharges.

Several control systems can be implemented in the system studied in order to control level and/or flow. The following control schemes can be mentioned as example: (1) a single loop for controlling the tank level, (2) a single loop for controlling the input flow, and (3) a cascade control scheme for controlling the tank level by using the input flow as a secondary variable, as shown in Figure 2. However, if the possibility of connecting the process module to a data acquisition system and a computer via OPC communication is considered, additional advanced control strategies can be tested.

The proposed model aims to answer the question: how does the liquid level inside the tanks vary when subjected to disturbances such as changes in the feed flow or in the discharge flow? In order to answer this question, a macroscopic scale model (lumped parameters) is used and described by ordinary differential Equations. Figure 3 shows a block flow diagram for the different process systems (PS) considered for the model deduction and the interactions between them.

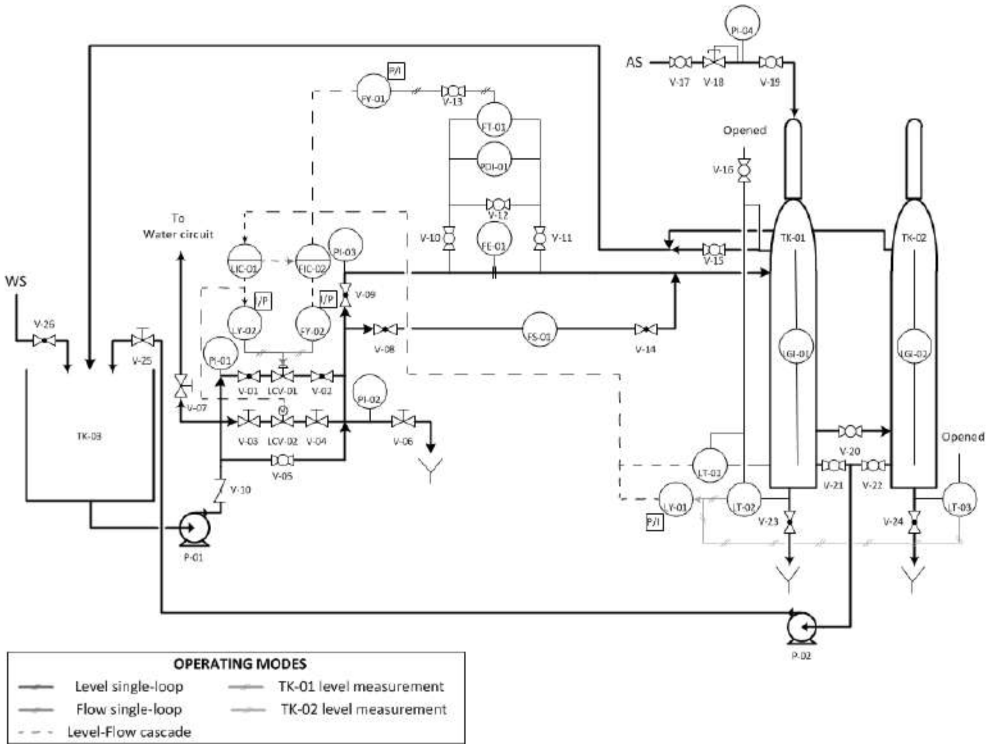


Figure 2 Simple diagram of the flow and level control module instrumentation

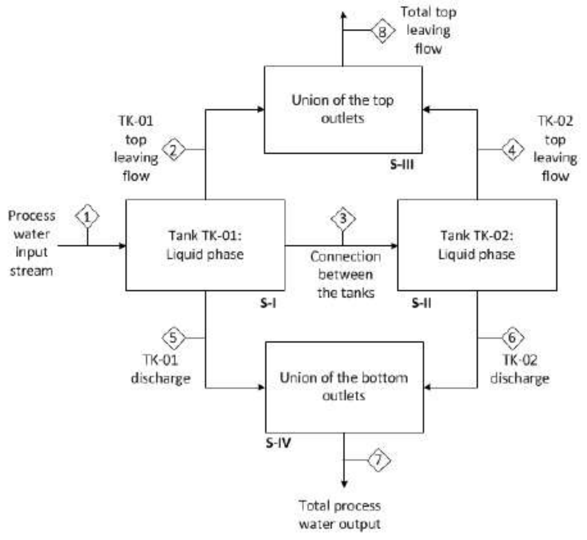


Figure 3 Block flow diagram of the process systems in the flow module

The model to be developed is of macroscopic scale. Therefore, the only process systems considered are the different compartments between the tanks: Systems I and II consist of the liquid phases in the tanks TK-01 and TK-02, respectively. System III represents the connection among the discharges of the two tanks.

In order to accomplish the model objectives, the balances of mass and energy for each process system must be stated. The following assumptions were considered in the deduction of the model:

- Properties of liquids are considered constant in each of the process systems considered (lumped parameters).
- Frictional and accessory losses are considered negligible.
- All the processes are considered isothermal.
- The operating volumes of the liquid phases are always inside the cylindrical section of the tanks.

By applying the conservation principle to the process systems shown in Figure 3, the PBSM model is obtained using Equations (1) - (5)

Liquid level in tank TK-01 (System S-I):

$$\rho\frac{dV_{L1}}{dt} = \rho F_1 - \rho F_3 - \rho F_5 - \rho F_2 + \rho F_{4,t} \tag{1}$$

Liquid level in tank TK-02 (System S-II):

$$\rho\frac{dV_{L2}}{dt} = \rho F_3 - \rho F_6 + \rho F_{2,t} - \rho F_4 \tag{2}$$

Total top stream flow leaving the system (System S-III):

$$\rho F_{2,t} + \rho F_{4,t} - \rho F_8 = 0 \tag{3}$$

Total bottom stream flow leaving the system (System S-IV):

$$\rho F_5 + \rho F_6 - \rho F_7 = 0 \tag{4}$$

Balance of mechanical energy in the system (System S-IV):

$$P_{o56} + \rho g z_{o56} + \rho \frac{v_{o56}^2}{2} = P_{o7} + \rho g z_{o7} + \rho \frac{v_{o7}^2}{2} \tag{5}$$

Where, the subscript *o56* refers to the joining points in the tank discharges and the subscript *o7* refers to the final discharge to TK-03. The rest of the subscripts are used to refer to the numbers of the different streams shown in Figure 3.

Equations (1) and (2) correspond to the dynamics of the liquid volumes inside the Systems I and II, respectively; and solving them, the values of the levels (L_1 and L_2) we are interested in can be found. Although the remaining Equations are stated in steady-state, they are needed to compute the aforementioned variables.

Equations (1) - (5) establish the phenomenological basis of the model for tanks TK-01 and TK-02. In order to solve them, the following constitutive Equations need to be stated: Equations (6), (7) for calculating the operating liquid volumes inside the tanks (V_{L1}, V_{L2}), Equations (8) - (10) for calculating the discharge flows (F_3, F_5, F_6) and Equations (13) - (16) for calculating the top stream flows transferred between the tanks ($F_{2,t}$ and $F_{4,t}$).

The model stated here is used for predicting the levels inside the tanks. Apart from the physical properties of the liquids and the geometrical parameters of the tanks, the other variables and parameters that must be specified, estimated, or computed using constitutive Equations, are:

Operating liquid volumes

$$V_{L1} = \frac{\pi D_{TK1}^2}{4}.L_1 \tag{6}$$

$$V_{L2} = \frac{\pi D_{TK2}^2}{4}.L_2 \tag{7}$$

Discharge flows of the tanks:

$$F_3 = C_{v,F3}\sqrt{\frac{P_{fTK01} - P_{fTK02}}{G}} \tag{8}$$

$$F_5 = C_{v,F5}\sqrt{\frac{P_{fTK01} - P_{o56}}{G}} \tag{9}$$

$$F_6 = C_{v,F6}\sqrt{\frac{P_{fTK02} - P_{o56}}{G}} \tag{10}$$

Where, the pressures at the bottom of the tanks are given by Equations ((11), (12)):

$$P_{fTK01} = P_{atm} + \rho g L_1 \tag{11}$$

$$P_{fTK02} = P_{atm} + \rho g L_2 \tag{12}$$

Top stream flows leaving the tanks

$$F_2 = \begin{cases} 0, & L_1 < h_{\max,TK01} \\ F_1 - F_3 - F_5, & L_1 \geq h_{\max,TK01} \end{cases} \tag{13}$$

$$F_{2,t} = \frac{F_2}{2} \tag{14}$$

$$F_4 = \begin{cases} 0, & L_2 < h_{\max,TK02} \\ F_3 - F_6, & L_2 \geq h_{\max,TK02} \end{cases} \tag{15}$$

$$F_{4,t} = \frac{F_4}{2} \tag{16}$$

Where $F_{4,t}$ and $F_{2,t}$ denote the flows entering to one tank from the other in case of tank overfilling, which can be approximated by Equations (14) and (16) when having a flow through equal diameter pipes and neglecting the flow resistance.

The flow velocity for the final discharge can be approximated as in Equation ((17))

$$v_{o7} = v_{o56} \tag{17}$$

The Equations system that needs to be solved is given by Equations (1) - (17). An analysis of the degrees of freedom for this model shows that if the input flow (F_1) is specified, the values of the coefficients $C_{V,F3}$, $C_{V,F5}$ and $C_{V,F6}$ for simulating the model have to be known. In order to estimate the values of such coefficients, several parameter estimation procedures must be carried out, as shown in section 3.2.

3.2 Parameter estimation and model validation

The following experiments were carried out in order estimate the unknown parameters:

1) Draining of tank TK-01 using the discharge pump.

2) Draining of tank TK-01 by gravity discharge.

3) "Opening vs. Volumetric Flow" curve of the pneumatic valve.

4) Calibration curve of the differential pressure cell used for measuring the level in tank TK-02.

5) Draining of tank TK-02 by gravity discharge.

6) Draining of tank TK-01 towards tank TK-02.

7) Filling of tanks TK-01 and TK-02 with gravity discharge through the outlet of the first tank.

8) Filling of tanks TK-01 and TK-02 with gravity discharge through the outlet of the second tank.

9) Filling of tanks TK-01 and TK-02 with gravity discharge through the outlets of both tanks.

Initially, experiments 1, 2, 5, and 6 were performed in order to estimate the parameters of each subsystem separately. Experiment 1 was used to estimate the total discharge flow when using the discharge pump. Experiments 2 and 5 were used to estimate the apparent flow coefficients of the valves located at the discharges of tanks TK-01 and TK-02, respectively. Experiment 6 was used to estimate the apparent flow coefficient of the discharge valve located at the point of interconnection between tanks TK-01 and TK-02. The discharge flow when using the discharge pump was determined experimentally to be about 44.94 l/min. This result was obtained by measuring the time taken for the level to go down between two specific points of the level indicator tape.

Experiment 7 was performed in order to estimate the apparent flow coefficient of the discharge valve located at tank TK-01 while the tank is both filled and drained by gravity.

Performing experiments 8 to 10 allowed estimating the different valve coefficients from the values initially estimated in the separate analyses of each subsystem. Figures 4 and 3.2 show the level dynamics observed experimentally (see white markers), the level dynamics given by the model (see continuous lines), and the experimental data used in the model validation (see black markers) for the variables L_1 and L_2, in the different operating modes considered. The values of the estimated parameters of all the operating modes are summarized in Table 1.

Table 1 Parameters estimated for the different experiments

Exp.	$C_{V,F3}$ l/(min.psi$^{0.5}$)	$C_{V,F5}$ l/(min.psi$^{0.5}$)	$C_{V,F6}$ l/(min.psi$^{0.5}$)
2	–	19.3218	–
5	–	–	15.0002
6	57.4525	–	–
7	–	13.8208	–
8	45.1166	13.8804	–
9	48.6597	–	13.9624
10	46.4146	8.89365	6.35297

As can be seen in Figure 4 and 3.2, the model offers very good predictions of the actual system. Although the model predictions (see continuous line) show a certain degree of deviation from the experimental data (see markers), it is clear that the major deviations were obtained for the separate analyses of the subsystems and not in the dynamical analyses. In general, the model predicts well the qualitative and quantitative behaviors observed in the actual system.

From Table 1, it is evident that the results obtained separately showed different values for the estimated parameters. This indicates that the parameters vary when the operating mode of the process system changes from one operating mode to another. Hence, there is not a unique set of universal parameters for all the operating modes as expected in a steady-state flow through a valve. However, it can be seen that when the valves interact, as in experiments 8 to 10, the process dynamics set the valve coefficients values in order to assure the mass and energy balances for each specific dynamic.

3.3 Implementation of control systems using EMSO

Creating the controller mathematical model

A simulable mathematical model or *FlowSheet* (with file extension **.mso*) must be created in EMSO. This *FlowSheet* must contain among its specifications (the SPECIFY section) the input variables referring to the read signals from the OPC server [10, 11].

OPC Server configuration

In order to read and/or write data from/to an OPC server using a PLC as an interface, a configuration file is required to allow the access to the desired variables in the server. In the case of the SIMATIC S7-200 PLC controller, this file has the extension **.pca* and contains the names, the data types, and the memory addresses of all the PLC variables required to create the OPC server. The basic procedure for the setting up an OPC server with the SIMATIC S7-200 PLC is described in [12].

Setting the connection between the EMSO model and the OPC server

A file (with extension **.eof*) must be created using the EMSO-OPC application to set up the OPC connection between the EMSO model file (extension **.mso*) and the OPC server. All the connections between the EMSO model variables and the OPC server tags must be listed in that file. Figure 6 shows the graphical user interface of the EMSO-OPC application and some connection examples.

The basic procedure for configuring the OPC connections among the EMSO model variables and the OPC server tag is as follows:

1) Create a new EMSO-OPC Project.

2) Select the OPC server you want to work with.

3) Select the simulable mathematical model (*FlowSheet* entity) of the controller.

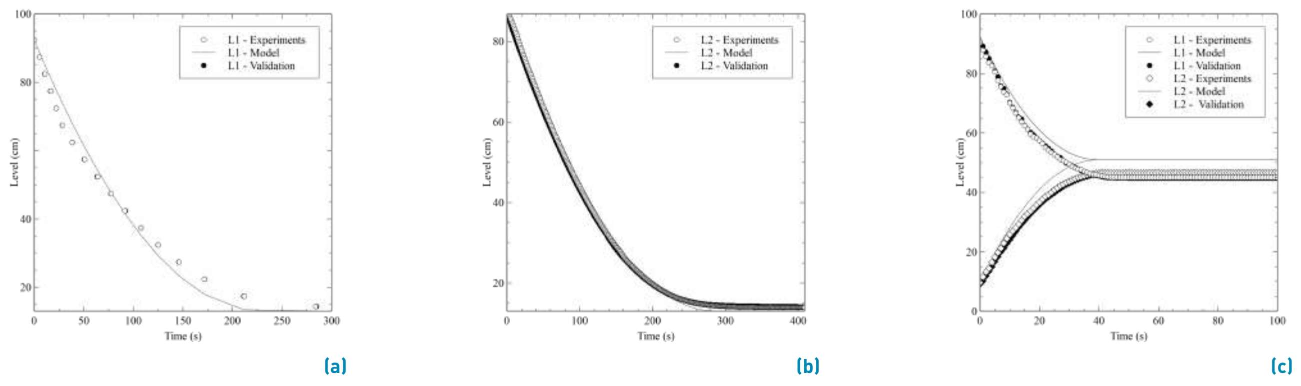


Figure 4 *a*) Level dynamics when draining tank TK-01 by gravity, *b*) level dynamics when draining tank TK-02 by gravity, and *c*) level dynamics when draining tank TK-01 by gravity towards the tank TK-02

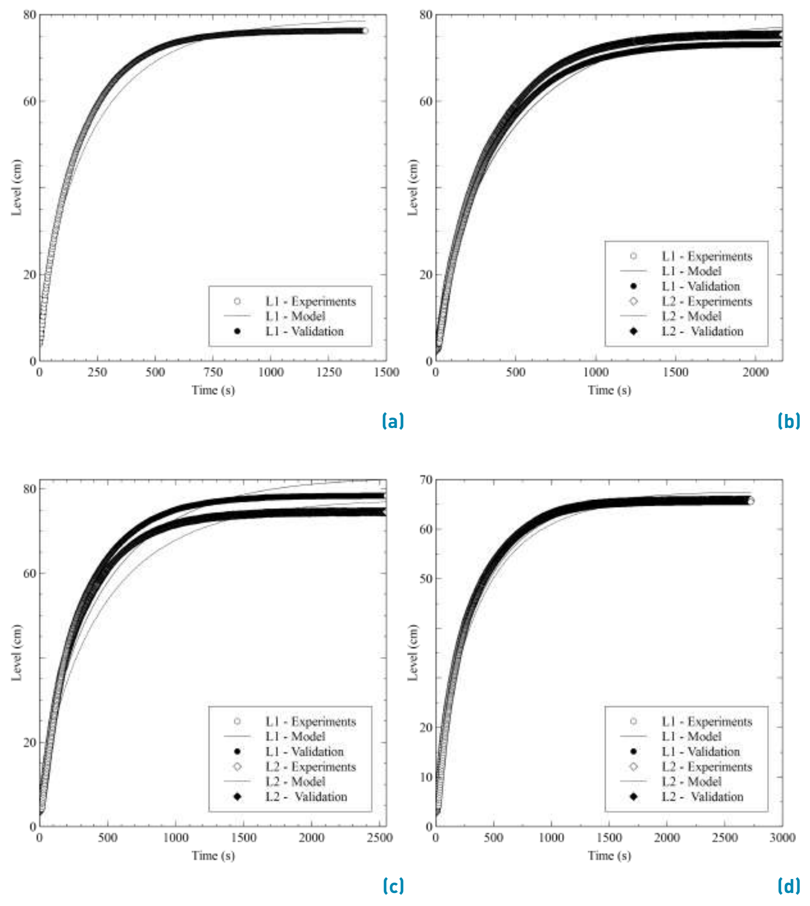
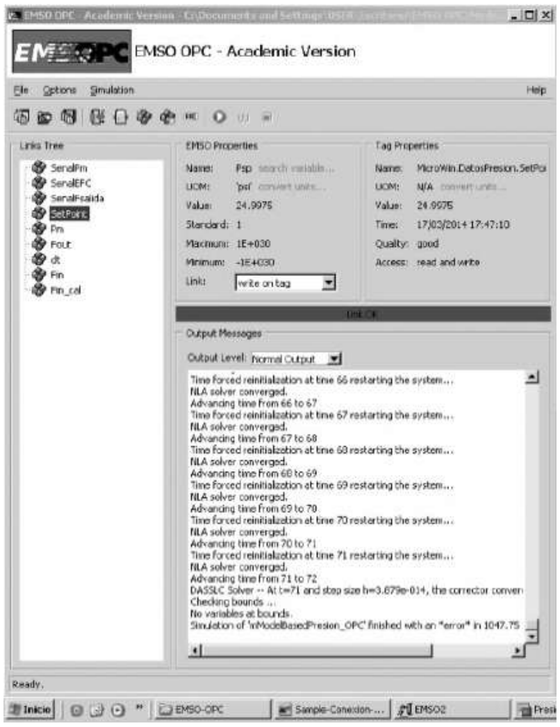


Figure 5 *a*) Level dynamics when filling tank TK-01 with gravity discharge, *b*) level dynamics when filling the tanks TK-01 and TK-02 with discharge through the outlet of TK-01, *c*) level dynamics when filling tanks TK-01 and TK-02 with discharge through the outlet of TK-02, and *d*) level dynamics when filling tanks TK-01 and TK-02 with discharge through the outlet of both tanks

4) Create the links or connections by searching the respective EMSO model variables and OPC server tags you want to connect. It is important to specify whether the OPC server tag in each link is read-only (*Read from tag*) or write-only (*Write on tag*).

5) It is important to check that the OPC server tags can be read correctly before proceeding. Otherwise, check that the data types on the OPC server correspond to DINT or REAL data types.



EMSO Properties

Name: Psp
UOM: 'psi'
Value: 24.9975
Standard: 1
Maximum: 1E+030
Minimum: -1E+030
Link: write on tag

Tag Properties

Name: MicroWin.DatosPresion.SetPoi
UOM: N/A
Value: 24.9975
Time: 17/03/2014 17:47:10
Quality: good
Access: read and write

Output Messages

Output Level: Normal Output

```
Time forced reinitialization at time 66 restarting the system...
NLA solver converged.
Advancing time from 66 to 67
Time forced reinitialization at time 67 restarting the system...
NLA solver converged.
Advancing time from 67 to 68
Time forced reinitialization at time 68 restarting the system...
NLA solver converged.
Advancing time from 68 to 69
Time forced reinitialization at time 69 restarting the system...
NLA solver converged.
Advancing time from 69 to 70
Time forced reinitialization at time 70 restarting the system...
NLA solver converged.
Advancing time from 70 to 71
Time forced reinitialization at time 71 restarting the system...
NLA solver converged.
Advancing time from 71 to 72
DASSLC Solver -- At t=71 and step size h=3.679e-014, the corrector conver
Checking bounds ...
No variables at bounds.
Simulation of 'mModelBasedPresion_OPC' finished with an *error* in 1047.75
```

Figure 6 Configuring the OPC connection

Creating an application for visualizing the process variables via OPC

In order to visualize the process variables and the controller performance, it is recommended that an additional OPC-compatible application be created, which can be connected to the OPC server to acquire and plot the data of the desired process variables. In this work, a model was implemented in SIMULINK and was used as a client application for reading and visualizing the process variables from the OPC server. In order to work with this SIMULINK model, a license for the *MATLAB OPC Toolbox* is required. Figure 7 shows a simple SIMULINK model which allows us to connect with the OPC server and build the dynamic plots of the process variables we are interested in.

Implementation of the control system and start-up

Once the required files have been created, the control system can be implemented and started up by connecting the SIMATIC S7-200 PLC and the computer through a TCP/IP connection cable. The control system is started as follows:

1) Run the simulation of the SIMULINK supervision model.

2) Run the OPC connection with the EMSO model in the respective EMSO-OPC project.

By using the methodology described in section 2, two control systems were stated: (1) a single-loop feedback control system with a PID controller and (2) a control system based on the process model obtained in section 4. The screenshot in Figure 8 shows how the control system based on EMSO and implemented via OPC (left-hand side) can be coupled with the supervision system in SIMULINK (right-hand side) in order to provide a simple control system connected in real-time to a process module.

Single-loop control system using a PID controller

In order to state this control system, a mathematical model had to be built to describe the PID controller. In the literature, there are a lot of models for describing PID controllers [13, 14]. However, in this work we only used the ideal form of a PID controller described in [14], as shown in Equation (18).

$$m(t) = K_c e(t) + \frac{K_c}{\tau_I}\int_0^t e(t) + K_c \tau_D \frac{d}{dt}e(t) \tag{18}$$

Where, $m(t)$ is the output signal leaving the controller expressed as a percentage (%). Figure 9 shows the statement of the PID controller model in EMSO.

The performance of this controller for different disturbances in the Set-Point is discussed in subsection 3.4.

Single-loop control system based on the PBSM model

In order to state the control system based on the Phenomenological-Based Semiphysical Model (PBSM), the operating mode of the process module which we were interested to work with (described in section 3) had to be selected manually. In this subsection, a simple controller based on the PBSM is stated. This controller was used to govern the liquid level in tank TK-01 by regulating the liquid flow coming into the tank. The controller stated here is a short version of the complete PBSM stated originally and was designed to work with only one of the tanks, TK-01 in this case. This is because is one of the most widely used controllers in the process control laboratory. However, working in a similar way, it is possible to obtain specific controllers based on the PBSM for each of the rest of the operating modes.

If only TK-01 is considered, it is assumed that the liquid level never goes beyond the cylindrical section of the tank. When using the model Equations (1), (5), (6), (9), (11) described in section 3, a reduced form of the mathematical model can be obtained, as shown in Equation (19), which allows the manipulated variable to be computed directly.

$$F_1 = \frac{\pi D_{TK1}^2}{4}\left(\frac{\Delta L_1}{\Delta t}\right) - C_{v5}\sqrt{\frac{\rho g (L_1 - z_{o7} + z_{o56})}{G}} \tag{19}$$

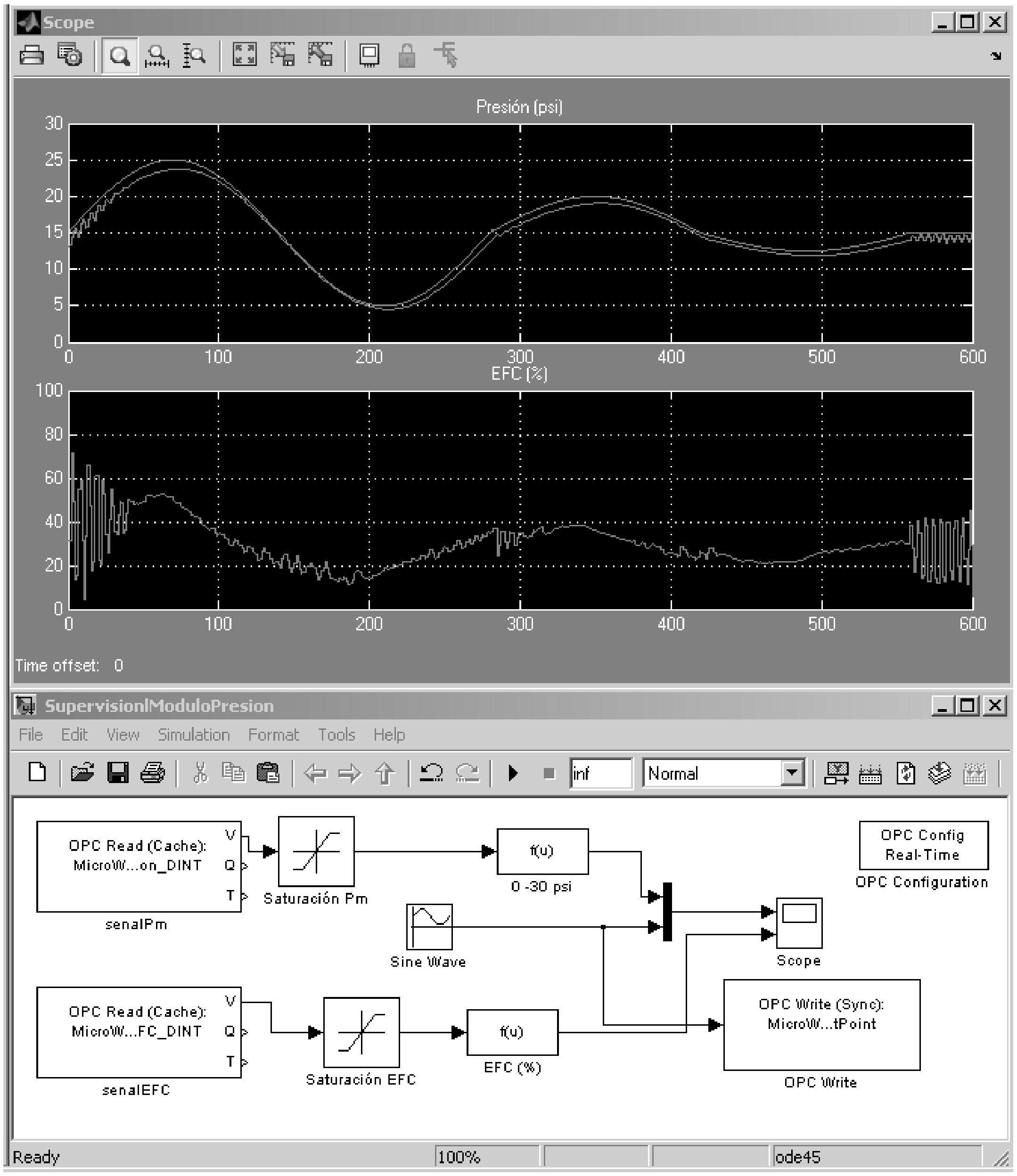


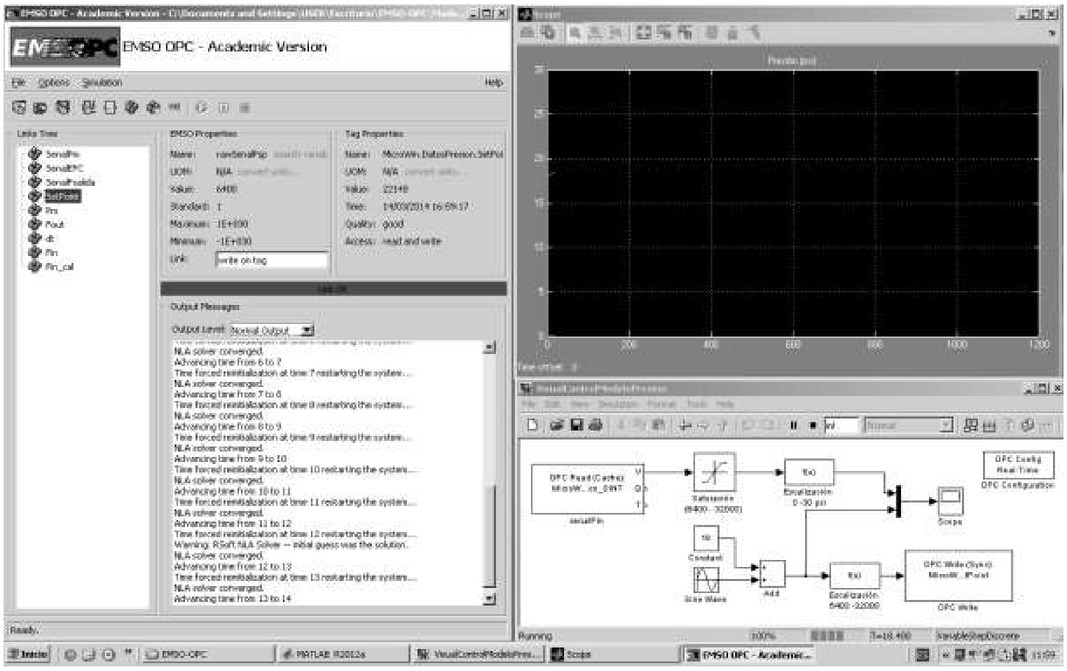
Figure 7 SIMULINK supervision model

Where, $\Delta L1 = L_{1sp}(t) - L_1(t)$, represents the deviation error in the controller, Δt corresponds to the sampling time used for the controller computations and the other terms corresponds to parameters of the model. Figure 10 shows the PBSM-based controller model as stated in EMSO.

In general, these controllers were implemented via OPC in the laboratory by scaling the input and output variables of the controller models from and to the range [6400, 32000] respectively because this represents the range of analog signals in the SIMATIC S7-200 PLC.

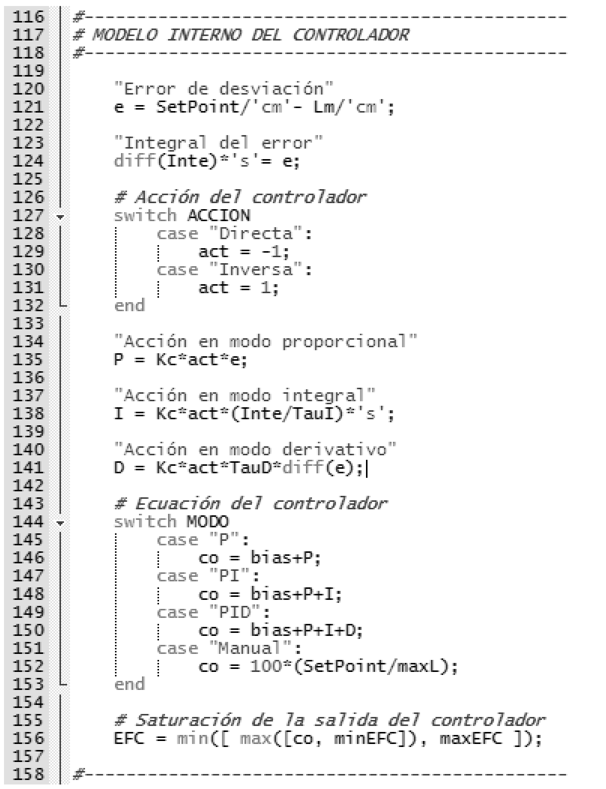
3.4 Analysis of controller performance

Figures 11 and 12 show the performance of the PID controller and PBSM-based controller for different disturbances in the Set-Point. In the case of the PID controller, the parameters used were $K_C = 10.9307$, $\tau_I = 61.2497s$, and $\tau_D = 4.4192s$. These parameters were obtained by process identification assuming a first-order-plus-dead-time dynamics (FOPDT) followed by the implementation of different tuning rules (Ziegler-Nichols in closed-loop [15], Tyreus-Luyben in closed-loop [15], Ziegler-Nichols in open-loop [14], Cohen-Coon in open-loop [16], Minimum integral error criteria of López *et al.* and Rovira *et al.* [14], and the



```
EMSO Properties
Name:     varnSenalPsp
UOM:      N/A
Value:    6488
Standard: 1
Maximum:  1E+030
Minimum:  -1E+030
Link:     write on tag

Tag Properties
Name:     MicroWin.DatosPresion.SetPoi
UOM:      N/A
Value:    22148
Time:     14/03/2014 16:59:17
Quality:  good
Access:   read and write

Output Messages
Output Level: Normal Output

NLA solver converged.
Advancing time from 6 to 7
Time forced reinitialization at time 7 restarting the system...
NLA solver converged.
Advancing time from 7 to 8
Time forced reinitialization at time 8 restarting the system...
NLA solver converged.
Advancing time from 8 to 9
Time forced reinitialization at time 9 restarting the system...
NLA solver converged.
Advancing time from 9 to 10
Time forced reinitialization at time 10 restarting the system...
NLA solver converged.
Advancing time from 10 to 11
Time forced reinitialization at time 11 restarting the system...
NLA solver converged.
Advancing time from 11 to 12
Time forced reinitialization at time 12 restarting the system...
Warning: RSoft NLA Solver -- initial guess was the solution.
NLA solver converged.
Advancing time from 12 to 13
Time forced reinitialization at time 13 restarting the system...
NLA solver converged.
Advancing time from 13 to 14
```

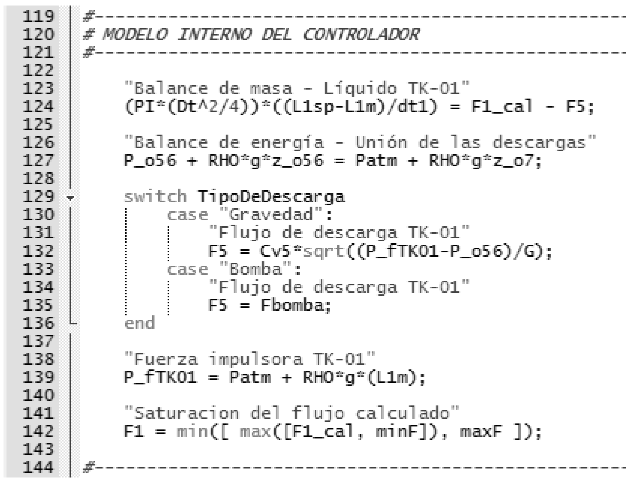
Figure 8 Simple control system based on EMSO-OPC with supervision in SIMULINK



```
#-----------------------------------------------
# MODELO INTERNO DEL CONTROLADOR
#-----------------------------------------------

    "Error de desviación"
    e = SetPoint/'cm'- Lm/'cm';

    "Integral del error"
    diff(Inte)*'s'= e;

    # Acción del controlador
    switch ACCION
        case "Directa":
            act = -1;
        case "Inversa":
            act = 1;
    end

    "Acción en modo proporcional"
    P = Kc*act*e;

    "Acción en modo integral"
    I = Kc*act*(Inte/TauI)*'s';

    "Acción en modo derivativo"
    D = Kc*act*TauD*diff(e);

    # Ecuación del controlador
    switch MODO
        case "P":
            co = bias+P;
        case "PI":
            co = bias+P+I;
        case "PID":
            co = bias+P+I+D;
        case "Manual":
            co = 100*(SetPoint/maxL);
    end

    # Saturación de la salida del controlador
    EFC = min([ max([co, minEFC]), maxEFC ]);

#-----------------------------------------------
```

Figure 9 PID controller model

Ciancone correlations [17]). Each controller obtained was tested in the real system. As result, the Tyreus-Luyben controller was selected due to the best performance.



```
#-----------------------------------------------
# MODELO INTERNO DEL CONTROLADOR
#-----------------------------------------------

    "Balance de masa - Líquido TK-01"
    (PI*(Dt^2/4))*((L1sp-L1m)/dt1) = F1_cal - F5;

    "Balance de energía - Unión de las descargas"
    P_o56 + RHO*g*z_o56 = Patm + RHO*g*z_o7;

    switch TipoDeDescarga
        case "Gravedad":
            "Flujo de descarga TK-01"
            F5 = Cv5*sqrt((P_fTK01-P_o56)/G);
        case "Bomba":
            "Flujo de descarga TK-01"
            F5 = Fbomba;
    end

    "Fuerza impulsora TK-01"
    P_fTK01 = Patm + RHO*g*(L1m);

    "Saturacion del flujo calculado"
    F1 = min([ max([F1_cal, minF]), maxF ]);

#-----------------------------------------------
```

Figure 10 Model of the PBSM-based controller in EMSO

Although both control systems worked well for step disturbances of the Set-Point, it is clear that the PBSM-based controller was faster in reaching its final condition and it had a smaller overshoot than the PID controller. The PID reached the final value, but the PBSM-based model shown small continuous oscillations. With respect to the trajectory imposed for the Set-Point, it is clear that both controllers tried to reach the path. However, the PID controller failed severely compared to the PBSM-based controller. In general, the performance of the PBSM-based controller was good despite the small continuous oscillations observed in its responses, oscillations that could appear due to the time delays

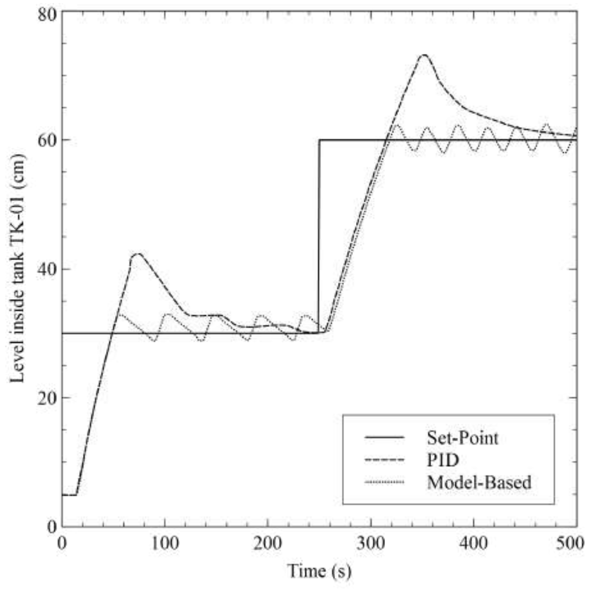


Figure 11 Performance of the controllers under step disturbances on the Set-Point

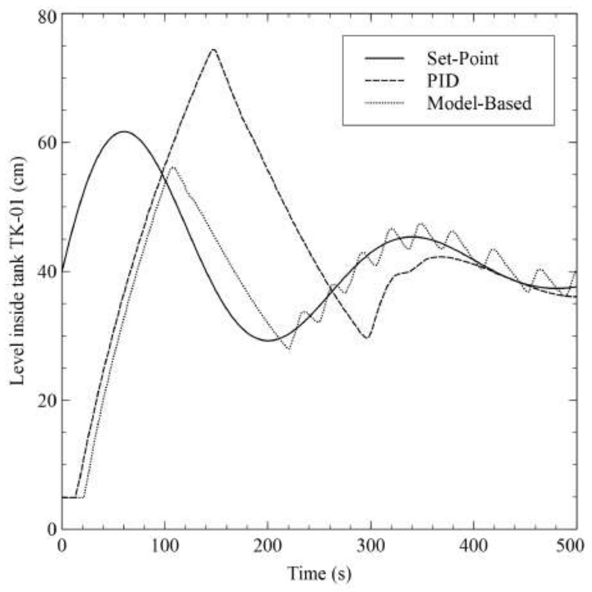


Figure 12 Performance of the controllers under an imposed trajectory for the Set-Point

generated with the sampling time of the EMSO-OPC interface and the computation time used by the simulator.

4. Conclusions

In the present work, a basic methodology was proposed for controlling and supervising a process via OPC using the EMSO simulator and a SIMATIC S7-200 as an OPC communication interface. This methodology was illustrated by identifying and modeling a laboratory module process using the EMSO simulator. The model obtained was used directly in the design and implementation of a model-based control system via OPC. The PBSM stated for the process module was in very good agreement both qualitatively and quantitatively with the experimental data obtained, and it produced low deviation errors. Regarding the control systems implemented, in general the performance observed for the controllers was good. The PID controller worked well for step disturbances of the Set-Point, but it failed for an imposed Set-Point trajectory. On the other hand, the PBSM-based controller did a great job in both cases, despite some small and continuous oscillations around the Set-Point. In addition, time delays were observed when coupling the EMSO simulator, the EMSO-OPC interface and the real process module, due to problems in the synchronization due to the time delays generated by the sampling time of the EMSO-OPC interface and the computation time used by the simulator. As final note for future works, it was observed that varying the sampling time parameter can improve the performance of the control system.

5. Acknowledgements

This project was developed thanks to the intern-scholarship COLCIENCIAS for Young Researchers (p-2012-1842) offered by COLCIENCIAS and the Universidad de Antioquia within the project entitled "Development of process control systems based on simulator and free communication systems" (in Spanish). In addition, the authors wish to express their most sincere acknowledgement to the following professors: Silvia Ochoa C., Milton Mejía, and José Tamara from the Universidad de Antioquia; and Professor Argimiro Resende Secchi from the Universidade Federal do Rio de Janeiro for his orientation and tips during the development of this project.

6. Notation

$C_{v,Fj}$	Parameter of the valve at stream j. [(l/min)(psi$^0.5$)]
D_{TKi}	Diameter of tank TK-0i. [cm]
$e(t)$	Controller deviation error.
F_j	Volumetric flowrate at stream number j.[l/min]
$F_{j,t}$	Partial volumetric flowrate from stream j after flow division. [l/min]
g	Acceleration due to the force of gravity. $[m^2/s]$
G	Liquid specific gravity, ρ/ρ$_H 2O$. [adimensionless]

$h_{max,TK0i}$	Maximum height that can be reached in the tank TK-0i.
K_C	PID controller proportional gain. [adimensionless]
L_i	Liquid level inside the tank TK-0x. [cm]
$m(t)$	PID controller output variable.
P_{atm}	Atmospheric pressure [mmHg]
P_{fTK0i}	Pressure at the bottom of tank TK-0i. [psi]
P_{oj}	Pressure at a point of stream j. [psi]
t	time. [s]
V_{Li}	Volume occupied by the liquid in tank TK-0i. [m^3]
v_{oj}	Velocity at a point of stream j. [m/s]

Greek Symbols

π	Constant PI = 3.14159265. [adimensionless]
ρ	Water density [kg/m^3]
τ_I	PID controller integral time. [s]
τ_D	PID controller derivative time. [s]

References

[1] R. de Pelegrini, "Desenvolvimento de um simulador genérico de processos dinâmicos," M.S. thesis, Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil, 2003.

[2] R. de Pelegrini and A. R. Secchi, "Emso: A new environment for modelling, simulation and optimisation," *Comput. Chem. Eng.*, vol. 14, pp. 947–952, 2003.

[3] R. de Pelegrini and A. R. Secchi, "Emso: An integrated tool for process modeling, dynamic simulation and optimization," in *Proceedings of the AIChE Annual Meeting*, 2003, pp. 451–464.

[4] T. corrêa, A. R. Secchi, and E. C. Biscaia, "A continuous implementation of the ideal time delay in EMSO simulator," *Comput. Aided Chem. Eng.*, vol. 27, pp. 273–278, 2009.

[5] J. L. Díaz and H. D. Álvarez, "Una plataforma en EMSO® para modelar pérdidas por fricción en plantas de procesos," *Lámpsakos*, pp. 79–91, Jul. 2014.

[6] J. ospino, M. E. sánchez, and A. R. Secchi, "Implementation of a block-oriented model library for undergraduate process control courses in EMSO simulator," *Educ. Chem. Eng.*, vol. 18, pp. 45–57, Jan. 2017.

[7] R. Rodrigues, R. de Pelegrini, and A. R. Secchi, "Teaching chemical reaction engineering using EMSO simulator," *Comput. Appl. Eng. Educ.*, vol. 18, no. 4, pp. 607–618, Dec. 2010.

[8] "Dynamic simulation and optimization using EMSO," Lecture Notes, A. R. Secchi, Brazil, Feb. 2012.

[9] H. Alvarez, R. Lamanna, P. Vega, and S. Revollar, "Metodología para la obtención de modelos semifísicos de base fenomenológica aplicada a una sulfitadora de jugo de caña de azúcar," *Rev. Iberoam. Automática e Informática Ind.*, vol. 6, no. 3, pp. 10–20, Jul. 2009.

[10] *EMSO Manual*, R. de Pelegrini, Brazil, 2007.

[11] *EMSO-OPC Link Manual*, T. F. Finkler and R. de Pelegrini, Brazil, 2007.

[12] Siemens Product Support. (2005, Jun. 14) ¿cómo se debe configurar el opc-server pc access para que el WinCC flexible runtime funcione como cliente opc? [Online]. Available: https://sie.ag/2wvvS9d

[13] A. O'Dwyer, *Handbook of PI and PID Controller Tuning Rules*, 3rd ed. London, England: Imperial College Press, 2009.

[14] C. A. Smith and A. R. Corripio, *Principles and practice of automatic process control*, 3rd ed. New York, USA: John Wiley & Sons, 2006.

[15] M. L. Luyben and W. L. Luyben, *Essentials for Process Control*. New York, USA: McGraw-Hill, 1997.

[16] J. A. Romagnoli and A. Palazoglu, *Introduction to Process Control*. Florida, USA: CRC Press, 2006.

[17] T. E. Marlin, *Process Control: Designing processes and control systems for dynamic performance*. New York, USA: McGraw-Hill, 2000.