Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-6 (2016) 156–161

Development of the supervision/control software for a multipurpose three-tank system

Santiago Rúa, Carlos A. Zuluaga, Norha L. Posada, Fabio Castrillón and Rafael E. Vásquez *

* *Universidad Pontificia Bolivariana, Circular 1 # 70-01, School of Engineering, 050031, Medellín, Colombia (e-mail: rafael.vasquez@upb.edu.co).*

Abstract:
This paper addresses the development of the supervision/control software for a multipurpose three-tank system to teach control engineering fundamentals. The laboratory equipment consists in a three series tanks with industrial instrumentation. The design requirements were determined by the users that perform academic activities in the area of process control. The requirements for the software include: data acquisition; clear visualization of the components involve in the process; communication with the different actuators; alarm generation; recorder; data logger; selection of different control strategies and technological systems (PLC, industrial controller or PC); and a friendly human machine interface. The equipment is being used to teach undergrad courses, grad courses, industrial training, and research projects.

© 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Process control, control education, control software.

1. INTRODUCTION

Control theory courses sometimes are strongly based in traditional books which contain some abstract concepts which are difficult to understand. To come up with this problems, modern books and several courses rely on simulation tools that allow students to speed the solution process and dedicate more time to analysis (Duro et al., 2008; Rasteiro et al., 2009; Zhang et al., 2013). However, simulations have restrictions and consider simplifications that in most cases differ from real phenomena present in physical systems. Hence, using real plants with direct and remote access becomes more effective for learning experiences, even considering small lab-scale processes with real interactions (Rasteiro et al., 2009; Ranade et al., 2012; Taylor et al., 2013; Ruiz et al., 2015).

Laboratory experiences in control engineering education have been focused in phenomena demonstration, comparison of theoretical and experimental behaviors, and verification of theoretical controller designs, among others. In general, lab experiences are limited to recipe-like procedures in which the student just follow instructions, limiting active learning opportunities. Modern education methodologies in engineering promote the active role of students in lab experiences, so they become protagonists in a real active learning environment. What is more, students can connect in a more direct way theory and practice (Haugen and Wolden, 2013; Iborra et al., 2014).

Applied control labs must have physical plants or processes, measurement and actuation instrumentation, controllers where control algorithms are implemented, and human machine interfaces (HMIs) that allow the student to interact with the equipment. Therefore, software interfaces are quite important in order to promote active learning since they represent the way for user interaction with the real process; such interfaces can be developed by experts and used as toolboxes for students, or can be developed by them, depending on the education level (technicians, undergrad or grad students) that is being pursued and the process needs (Murray et al., 2003; Baloi et al., 2015).

Vásquez et al. (2014) developed a lab equipment for process dynamic systems and process control education. They divided the design process into three stages: concept, basic, detailed design of the process in order to satisfy different user's requirements and educational outcomes (Vasquez et al., 2015). This platform was designed and built, so three different control technologies could be used: a programmable logic controller, an industrial controller, and a PC; however, previous works did not include the supervision/control software intended to run in the PC which is the scope of this paper.

Section 2 contains a brief description of the process, its components, and its possible configurations. Section 3 shows the software design methodology based on user's requirements. Section 4 describes the software implementation for data acquisition in the PLC and the human-machine interface for the PC. Section 5 contains an example of the utilization for the dynamic system model's identification in a lab session. Finally, some conclusions are addressed.

2. PROCESS DESCRIPTION

There are several options to select a system to teach control process fundamentals in undergrad and grad courses.

2405-8963 © 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.07.170

However, tank systems are reported in literature as good options due to the reconfiguration possibilities for the process, and their good visualization capabilities (Horacek, 2000; Hou et al., 2005).

The laboratory equipment is a series tanks process, with an upper tank which is shown as two interacting tanks (T-1,T-2) in Fig. 1, and a lower tank (T-3). Level measurements in each tank are indicated with level transmitters (LT). An electrical pump (P-1) brings water to the system through two different branches; flow transmitters are shown in each branch (FT). The main loop has a control valve to allow flow manipulation, and the secondary loop has a manual valve to introduce disturbances into the tank (T-1, T-2, or T-3). The flow can be also manipulated through pump speed variations using an industrial AC Driver. The pressure drop in the control valve can be measured with a differential pressure transmitter (DPT). The upper tank has a removable divider, allowing the user to choose between one big tank, one small tank, and two interacting tanks. Therefore, the system's dynamics (order) can be chosen, depending on the user's selection as shown in Table 1.

Linear or nonlinear behavior can be introduced with sharp crested open channel weirs used for the output of tanks (T-2) and (T-3); different shapes of weirs can be chosen: v-notch, rectangular, linear, etc. Dead time can be created by using a removable pipe for the discharge of (T-2); such pipe can be selected long enough to create a variable time delay (since it depends on the flow), which represents a challenging control situation if it approaches the dominant time constant of the system. Fig. 2 shows the real plant.

3. SOFTWARE DESIGN

To define the software interface, a group of potential users within the School of Engineering at the Universidad Pontificia Bolivariana (UPB), located in Medellín, Colombia, was surveyed. This group, comprised of professors with different backgrounds (mechanical, chemical, software, electrical and control engineers), defined the following list of requeriments:

- Data acquisition for instrumentation: communication with instruments, recording and data logging capabilities.
- Communication with actuators (AC driver and control valve).
- Visualization of components involved in the process.
- Alarm generation.
- Capability of different options for the controller: programmable logic controller (PLC), industrial controller or PC.
- Friendly human machine interface.

Selection	Order	Interaction
1 big tank (T-1 + T-2)	1	No
1 small tank (T-2)	1	No
2 series big tanks (T-1+T-2),(T-3)	2	No
2 small interacting tanks (T-1),(T-2)	2	Yes
1 small tank (T-2), 1 big tank (T-3)	2	No
2 small interacting tanks (T-1), (T-2) and 1 big tank	3	Yes

Table 1. Tank system configuration

4. IMPLEMENTATION

The development of the control software for the multipurpose tank system was divided into three stages:

- PLC program for instrumentation.
- PC interface.
- HMI for communication with the PLC.

For the design process, each control software component has to consider physical connections in the system (Fig. 3). The PLC gets signals from instruments and sends signals to actuators. The PLC, the PC and the industrial controller are connected using Ethernet through a switch. This work does not include the HMI that is being developed for the PLC using a touch screen, but the development of the software interface for the computer.

4.1 PLC program

Within the software architecture shown in Fig. 3 the PLC is the default control system, hence the process can operate without the connection of the PC or the industrial controller. The controller option can be selected in both the PLC HMI (in development) or the PC HMI developed in LabVIEW™ and discussed in this work. The controller selection is done through the information exchange using a data structure which includes the state of each device connected to network, data from measurements and actuators, and different parameters. The PLC has a dedicated memory space for this structure.

Physical communications with instruments are done by using analog modules with four inputs and two outputs. Such modules are configurable to allow 0-20 mA and ±10 V inputs with a 14-bits resolution; all measurements are then scaled to 0-100 % signals. The reverse process is used to write signals in the actuators; a 0-100 % signal is sent from the controller, then is converted to 4-20 mA which is equivalent to a value in the analog/digital converter within 5529 and 27648. The PLC program has two communication blocks: data transmission and data reception; this communication is done in a synchronized way at 10 Hz. When the supervision mode is selected (control is running on the PC), the PLC has communication failure alarms, so emergency actions can be carried out.

4.2 LabVIEW™ interface

LabVIEW™ was selected as the language for the software interface due to its flexibility and high level specifications (Morris and Langari, 2016). The interface has been developed to run as Supervisory Control And Data Acquisition (SCADA) system, which allows one to control the process from a remote computer. The LabVIEW™ interface was developed in order to meet the following high level specifications:

- Controller type selection.
- Perform control actions.
- Controller parametrization.
- Visualization of variables.
- Data storage.
- Run predefined plant tests.
- Window control.

Download English Version:

https://daneshyari.com/en/article/710937

Download Persian Version:

https://daneshyari.com/article/710937

Daneshyari.com