Acta Scientiarum 22(5):1167-1171, 2000.
ISSN 1415-6814.

A simple real time process control experiment using serial communication

Ivo Neitzel[1]* and Marcelo Kaminski Lenzi[2]

*[1] Departamento de Engenharia Química, Universidade Estadual de Maringá, Av. Colombo, 5790, 87020-900, Maringá-Paraná, Brazil [2] Universidade Federal do Rio de Janeiro, Programa de Engenharia Química/Coppe, Cidade Universitária, C.P. 68502, 21945-970 Rio de Janeiro-Rio de Janeiro, Brazil. *Author for correspondence.*

ABSTRACT. The development and the low cost of microcomputers introduced process control into a new era. Currently many analog and on-off controllers are being replaced by simple softwares. In this paper a simple and didactic process control experiment will be described. Basically it consists of the temperature control of equipment (an oven) using two microcomputers connected by their serial ports. Experiment features introduction of the practical side of the process control, using a distributed control system (DCS). It also allows the student to interact with the process and build the controller himself.

Key words: process control, serial communication, software, education.

RESUMO. Usando comunicação serial em um experimento de controle de processos em tempo real. O desenvolvimento e baixo custo dos microcomputadores introduziram o controle de processos em uma nova era. Atualmente, muitos controladores *on-off* e analógicos estão sendo substituídos por programas computacionais (*softwares*). Neste artigo, descreveremos um experimento simples e didático visando ao controle de temperatura de um equipamento (uma mufla), usando dois microcomputadores conectados por suas portas seriais. As principais características deste experimento são os fatos de introduzir o aluno no lado prático do controle de processos, usando um sistema distribuído de controle digital (SDCD), e permitir tanto a construção do controlador quanto a interação (aplicação distúrbios) com processo.

Palavras-chave: controle de processos, comunicação serial, programa computacional, educação.

Due to the development and to the low cost of microcomputers, a new era came into being in process control education. Microcomputers are widely used in a "theoretical" way, namely for designing, tuning and simulating the controller behavior (Gerry, 1988); studying the influence of pole location and dynamic response of the system to some kind of perturbations (Marlin, 1996), among others. Besides the "theoretical" way, many didactic experiments have been reported, such as the control of liquid level (Adb-El-Bary, 1983) and the control and operation of batch distillation columns (Fileti and Pereira, 2000). In all these reported experiments the same microcomputer was used for data acquisition and control tasks.

In this paper, we will describe a simple and didactic experiment to introduce the student to the practical side of process control. Our aim is the temperature control of an oven using a simple computer network. By means of this experiment the student will be introduced to DCS (Distributed Control System) and to the basic theoretical aspects of process control, such as controller designing and tuning. After a brief review of a few fundamental concepts, the experiment design and the software design will be given. Finally, the results concerning the behavior of the system to the application of two-step perturbations will be discussed.

Theoretical framework

Distributed control systems. In the early 1960s microcomputers began to be used in chemical process control (Edgar *et al.*, 1989; White and George, 1997). The first applications employed only one computer for all tasks involved, such as data acquisition and the control itself. Depending on the size of the system to be controlled, i.e., the number of manipulated and controlled variables, the complexity of the mathematical model of the system, the graphical information to be displayed, this configuration may still be used, although currently it is restricted to didactic experiments.

An actual chemical process has more than 100 manipulated and controlled variables and thus a single microcomputer for its control is an impossibility. The level of development and low cost of microcomputers reached after the mid-70s made possible the decentralization of the control system, or Decentralized Control System DCS. This boils down to the fact that there is more than one microcomputer in the process, the tasks are shared and there exists a hierarchical network to monitor and control the process. This network generally has a pyramidal form in which the basic tasks lie at the bottom and use less sophisticated devices. As information is passed upward through the pyramid, the complexity of the operations increases similar to the power of the microcomputer (Mellichamp, 1983). Some advantages in the use of a DCS are presented by Ogunnaike and Ray (1994). The principal advantage lies in the facility of system expansion, i.e., the increase or decrease of the number of manipulated or controlled variables. This occurs because separate units are designed for data acquisition, single-loop control, operator display and so on (Edgar *et al.*, 1989).

The control system we are presenting in this paper has the same main features of a DCS. There are a simple computer network, a unit designed for data acquisition and another one for single-loop control. The single-loop control was done with PID (proportional-integral-derivative) control strategy, as will be seen later on. As the paper has didactic ends the DCS developed is not complex and employs only two microcomputers.

PID control strategy. The PID family of control strategy is quite old, but works smoothly for many kinds of actual applications, besides being very didactic. It is given by Equation (01) and was initially developed for continuous time domain (Stephanopoulos, 1984). See Nomenclature.

$$C(t)=C(t_0)+K_C\cdot\left[e(t)+\frac{1}{\tau_I}\cdot\int_0^t e(t)dt+\tau_D\cdot\frac{de(t)}{dt}\right] \qquad (01)$$

Our system used the velocity form, which reduces the reset windup (Edgar *et al.*, 1989). This form is obtained from Eq. (01). The computer (controller) output at the n-th and (n-1)-th sampling instant can be respectively written as:

$$C(n)=C(0)+K_C\cdot\left[e(n)+\frac{\Delta T}{\tau_I}\cdot\sum_{i=0}^{n}e(i)+\frac{\tau_D}{\Delta T}\cdot[e(n)-e(n-1)]\right] \qquad (02)$$

(03)

Subtracting Eq. (03) from Eq. (02), one obtains the PID velocity form given by:

$$C(n)=C(n-1)+K_C\cdot\left[[e(n)-e(n-1)]+\frac{\Delta T}{\tau_I}\cdot e(n)+\frac{\tau_D}{\Delta T}\cdot[e(n)-2\cdot e(n-1)+e(n-2)]\right] \qquad (04)$$

It may be seen that Eq. (04) represents the whole family of PID controllers. With the appropriate setting it may be either a P controller ($\tau_I \rightarrow \infty$; $\tau_D = 0.0$) or a PI ($\tau_D = 0.0$), or a PD ($\tau_I \rightarrow \infty$), or finally a PID. In the experiment, we decided to use a PI form for the single-loop control of the DCS system.

Controller tuning. Controller tuning is one of the most important steps in the design of a control system. It consists of a selection of the best parameters of the controller (K_c, τ_I, τ_D in the case of a PID family) (Gerry, 1988). There are many methods for its execution, each one considered a performance criteria, such as *one-quarter decay ratio*, *minimum offset*, *minimum integral square error (ISE)* and others. The simplest one is the Cohen-Coon method described by (Edgar *et al.*, 1989) and (Stephanopoulos, 1984) and based on the one-quarter decay ratio.

The method has the following steps:

1. Let's assume that the system has a first order plus time delay behavior, Eq. (05) (in Laplace domain);

$$G_{PROCESS}(s)=\frac{K\cdot e^{-t_d\cdot s}}{\tau\cdot s+1} \qquad (05)$$

2. Apply a step perturbation to the system;
3. Record the data and wait until steady-state is reached;
4. Fit the data obtained to Eq. (05) in order to obtain parameters K, τ, t_d of the system;
5. With the parameters of the system calculate those of the controller according to type chosen, whether P, PI, PD or PID.

As has been explained above, the controller type chosen was PI. The following equations are thus used to calculate its parameters:

$$K_C=\frac{1}{K}\cdot\frac{\tau}{t_d}\cdot\left[0.9+\frac{1}{12}\cdot\left[\frac{t_d}{\tau}\right]\right] \qquad (06)$$

$$\tau_I=t_d\cdot\left[\frac{30+3\cdot\left[\frac{t_d}{\tau}\right]}{9+20\cdot\left[\frac{t_d}{\tau}\right]}\right] \qquad (07)$$

Experiment design

The experiment is divided into two parts: the first one consists in applying the Cohen-Coon method to the system. The second one is the control experiment itself. Explanation below is based on the second part of the experiment.

The use of two microcomputers to control the temperature of an oven with an external ventilator, using feedback control and a PI controller routine characterizes the experiment. Since the microcomputers are connected by their serial ports, they intercommunicate between themselves and the control tasks are shared. The process is similar to a simple network.

One of the microcomputers, which contains the analog-to-digital and digital-to-analog conversion board, is called *slave microcomputer* because its only function is signal conditioning (a digital filter was implemented, as suggested by Edgar *et al*., 1989) and is connected directly to the equipment with the temperature controller. The other microcomputer is called *master microcomputer* because it contains the PI controller routine (Mellichamp, 1983), which is the core of the control software. It is connected to the slave microcomputer, as shown in Figure 1.

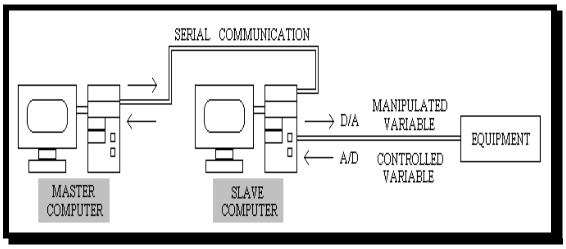

Figure 1. Schematic representation of the system in the experiment

The internal structure of the equipment (oven) used in the experiment and its connection to the slave computer is presented by a scheme representation in Figure 2.

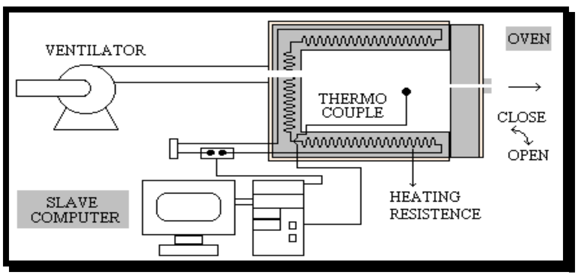

Figure 2. Schematic representation of the equipment used in the experiment

A thermocouple, Iron-Constantan - Type J, (Perry and Green, 1997) was used to measure the temperature of the outgoing airflow. This temperature is the *controlled variable*. The *manipulated variable* is the electric current that passes along the resistance inside the oven structure. The ventilator is always turned on during the experiment.

Software design

The software, the system's core, was carefully designed in order to share the control tasks between the microcomputers and to use serial communication in a correct manner (Mellichamp, 1983).

The first step in the master microcomputer is to set up the sample rate for data acquisition, which channels of the A/D - D/A conversion board are to be used, the *set point* of the controlled variable and an initial value for the manipulated variable (used to start-up the oven). All these initial values, except the initial value for the manipulated variable, may be changed by manipulating the keyboard during experiment. Figure 3 illustrates software stages in each microcomputer.

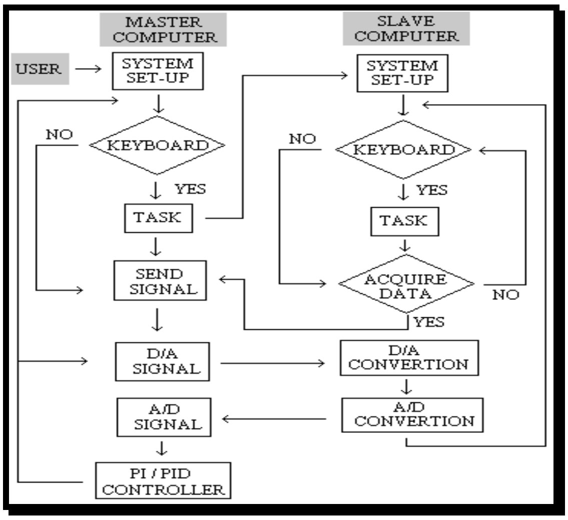

Figure 3. Illustration of the software stages

The figure shows that when it is necessary to acquire data, the slave microcomputer "advises" the master (Send Signal). Immediately the last value of the manipulated variable calculated by the controller routine (D/A Signal) is sent from the master to the slave microcomputer. This signal is converted (D/A Conversion) and output to the process. Next, the slave microcomputer reads the signal of the controlled variable from the equipment, makes the conversion (A/D Conversion) and sends the value to the master microcomputer. The received value (A/D Signal) is passed to the controller routine (PI/PID Controller), which will calculate the next value of the manipulated variable to be transmitted to the process through the slave microcomputer. The software then checks for any pending task requested by the user. If any, the task is executed. Finally the software stays in an infinite loop until another data acquisition is necessary.

Values of the controlled and manipulated variables and time are recorded in a file. These data are available for any mathematical package and may be used for more detailed analysis.

The serial communications routines were developed in C++, (see http://www.marshallsoft.com), the low level A/D and D/A conversions were developed in C and the high level routines in Fortran. The controller tasks and the rest of the software were developed in Fortran.

The modular characteristic of the software, seen in the block diagram (Figure 3), shows three important features:

i) controller may be any of the PID family; the setting of adequate parameters are needed for any change;
ii) any A/D - D/A signal conversion board may be used by properly rewriting the low level routines;
iii) control system software may be used to control any other process;
iv) software runs under either Windows NT system or Windows 95-98.

Experimental procedure

The following sequence should be used if the controller is already tuned:

1. Turn the ventilator on;
2. Close the oven and turn it on;
3. Turn slave and master microcomputer on;
4. In each microcomputer set the same values for baud rate, stop bits, parity bit, for serial communications;
5. In each microcomputer choose the serial port, i.e., COM2 or COM3 or COM4;
6. Set up all initial conditions in master microcomputer and start the program;
7. Wait until the oven temperature reaches steady-state at set point temperature;
8. Introduce a perturbation in the output airflow temperature by opening the oven. Wait until temperature reaches once more steady-state at set point temperature;
9. Introduce another perturbation in the output airflow temperature by closing the oven. Wait until temperature reaches steady-state again at set point temperature.

After step 9, the experiment is finished and the system should be shut:

10. Turn the oven off and open it;
11. Exit software;
12. After 20 minutes turn the ventilator off.

If the controller is not tuned, a reaction curve experiment should be performed to obtain the process parameters and consequently the controller parameters, using the Cohen-Coon method already explained. This is explained in the item *Controller Tuning*. It must be kept in mind that in this case the control routine (see Figure 3) is not available; so the software works as a temperature and time recorder.

Results and discussion

The oven was first set with the door closed. The first experiment performed was the reaction curve for controller tuning. The opening of the door of the oven was considered a step perturbation. With the recorded data and using the software Statistica® for the fitting, after some calculations, the following parameters of the controller were obtained:

$K_C = 82.95$

$\tau_I = 116.06$ s

The control experiment was performed and the recorded data plotted in Figure 4. The sample time for data acquisition was 1.0 s, the set point of the oven temperature was 350ºC (1.6 V), initial value for the manipulated variable was 1.0 V.

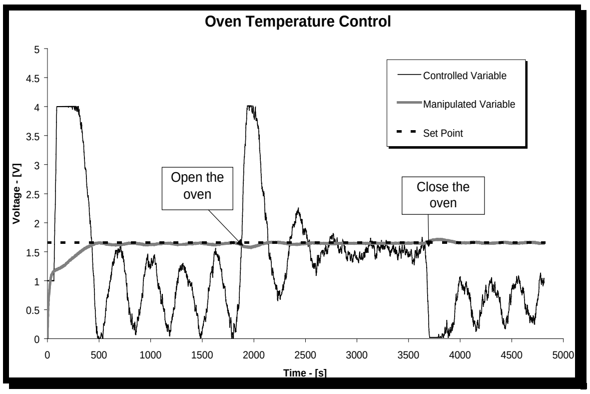

Figure 4. Experiment result graphic

Analyzing Figure 4, it may be seen that when the oven was first opened (t = 1900 s), the temperature became lower than the set point value, as expected; the PI controller worked fine, and brought the temperature back to the set point. When the oven was closed (t = 3700 s, step 9 of the experimental procedure), the temperature became higher than the set point value; once more the PI controller worked fine, bringing it back to the set point value. A good noise rejection was achieved with the digital filters implemented in the slave microcomputer. The perturbations (opening and closing of the oven) are taken as step perturbations.

The use of a two-microcomputer system is very important if a large number of variables are to be controlled and analyzed; otherwise it is necessary to have real time complex graphics of the data, as explained above.

Acknowledgments

We would like to thank Capes (Brazilian Agency) for its financial support.

Nomenclature

- A/D analog-to-digital
- D/A digital-to-analog
- DCS Distributed Control System
- C(t) Controller output signal (manipulated variable) at time t
- $C(t_0)$ Controller output signal (manipulated variable) at time t=0
- e(t) Error signal at time t (or sampling instant n), i.e., the diference between the set-point (desirable) value of the controlled variable and the value of the controlled variable read from the experimental system at time t (or sampling instant n)
- K_C Proportional gain of the controller
- τ_I Integral Time constant
- τ_D Derivative time constant
- ΔT Sampling interval
- t_d Process time delay constant
- Process time constant
- K Process gain
- P Proportional controller
- PI Proportional - integral controller
- PD Proportional - derivative controller
- PID Proportional - integral - derivative controller

References

Adb-El-Bary, M.F. Direct dgital control liquid level experiment. *Chem. Engineer. Educ.*, *17*(1):28-31, 1983.

Edgar, T.F.; Mellichamp, D.A.; Seborg, D.E. *Process Dynamics and Control*. New York: John Wiley, 1989.

Fileti, A.M.F.; Pereira, J.A.F.R. A direct digital control experiment for graduate and undergraduate chemical engineering students. *Comp. Chem. Engineer.*, *20*:1491-1495, 1996.

Gerry, J.P. Find out how good that PID tuning really is. *Control Engineer.*, *35*(08):69-71, 1988.

Marlin, T.E. The software laboratory for undergraduate process control education. *Comp. Chem. Engineer.*, *20*:S1371-S1376, 1996.

Mellichamp, D.A. Real-Time Computing - With applications to Data Acquisition and Control. New York: Van Nostrand-Reinhold, 1983.

Ogunnaike, B.A.; Ray, W.R. *Process Dynamics, Modeling and Control.* New York: Oxford University Press, 1994.

Perry, R.; Green, D.W. *Perry's Chemical Engineers' Handbook*. New York: McGraw-Hill, 1997.

Stephanopoulos, G. *Chemical Process Control.* An Introduction to Theory and Practice. New Jersey: Prentice-Hall, 1984.

White, P.; George, G. Controlling the Future. *Chem. Engineer., 12*(104):74-78, 1997.

Received on August 20, 2000.

Accepted on November 22, 2000.