

# Tuning PID Controllers using the ITAE Criterion\*

FERNANDO G. MARTINS

Chemical Engineering Department, LEPAE, Faculty of Engineering, University of Porto, Rua Dr Roberto Frias sln, 4200–465 Porto, Portugal. E-mail: fgm@fe.up.pt

*The minimization of the integral of time-weighted absolute error (ITAE) is usually referred to in literature as a good tuning criterion to obtain controller PID parameters. However this criterion is not often used because its computer implementation is not a very easy task. This paper describes how MATLAB/Simulink can be useful to apply the ITAE criterion to calculate controller parameters.*

## INTRODUCTION

DESPITE ALL ADVANCES in process control over the past 60 years, the PID controller is still the most common controller [1]. The main reasons are the simplicity, robustness and successful applications provided by PID-based control structures [4].

Minimizing integral of time-weighted absolute error (ITAE) is commonly referred to as a good performance index in designing PID controllers [5]. As an advantage, the search of controller parameters can be obtained for particular types of load and/or set point changes and as this criterion is based on error calculation it can be applied easily for different processes modelled by different process models.

According to MATLAB website (<http://www.mathworks.com>), Simulink is an interactive tool for modeling, simulating and analyzing dynamic, multi-domain systems. It allows accurate describing, simulating, evaluating and refining of a system's behavior through standard and customized block libraries. Simulink integrates seamlessly with MATLAB, providing an immediate access to an extensive range of analysis and design tools.

One of these tools is the MATLAB Optimization Toolbox which provides tools for general and large-scale optimization. The toolbox includes algorithms for solving many types of optimization problems, including unconstrained nonlinear minimization.

This paper presents software modules developed in Simulink and MATLAB for tuning PID controllers using the ITAE criterion.

The paper is organized as follows. Section 2 describes briefly the steps to implement the ITAE criterion in Simulink and MATLAB, Section 4 is devoted to the cases studies: tuning PID controller's parameters for processes translated by first

and second order plus dead time transfer functions and the problem of control of a continuous stirred tank reactor process. For the two first cases, the results are compared with the results obtained by Ziegler-Nichols tuning relations [5].

## THE PROCEDURE

The steps taken to design PID controllers using the ITAE performance index are:

1. Develop the process model including the controller algorithms in Simulink.
2. Create a MATLAB m-file with an objective function that calculates the ITAE index.
3. Use a function of MATLAB Optimization Toolbox to minimize the ITAE index.

Step 1 is realized opening a new Simulink window and drag-and-drop all necessary blocks to simulate the process in Simulink. Some global variables must be defined (in this case, the controller parameters).

In step 2, a MATLAB m-file is defined to calculate the ITAE index (the objective function). The ITAE performance index is mathematically given by:

$$ITAE = \int_0^{\infty} t|e(t)|dt \quad (1)$$

where  $t$  is the time and  $e(t)$  is the difference between set point and controlled variable.

In step 3, a function of MATLAB Optimization Toolbox is called to calculate the minimum of the objective function defined in step 2.

On each evaluation of the objective function, the model developed in Simulink is executed and the ITAE performance index is calculated using the multiple-application Simpson's 1/3 rule [2].

\* Accepted 30 May 2005.

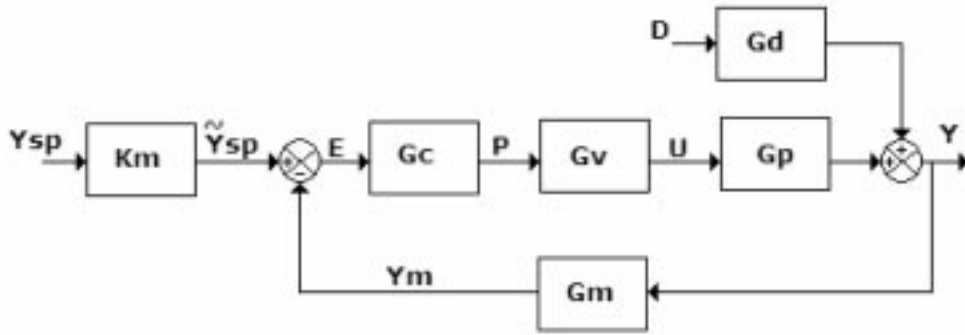


Fig. 1. Block diagram of a feedback control system.

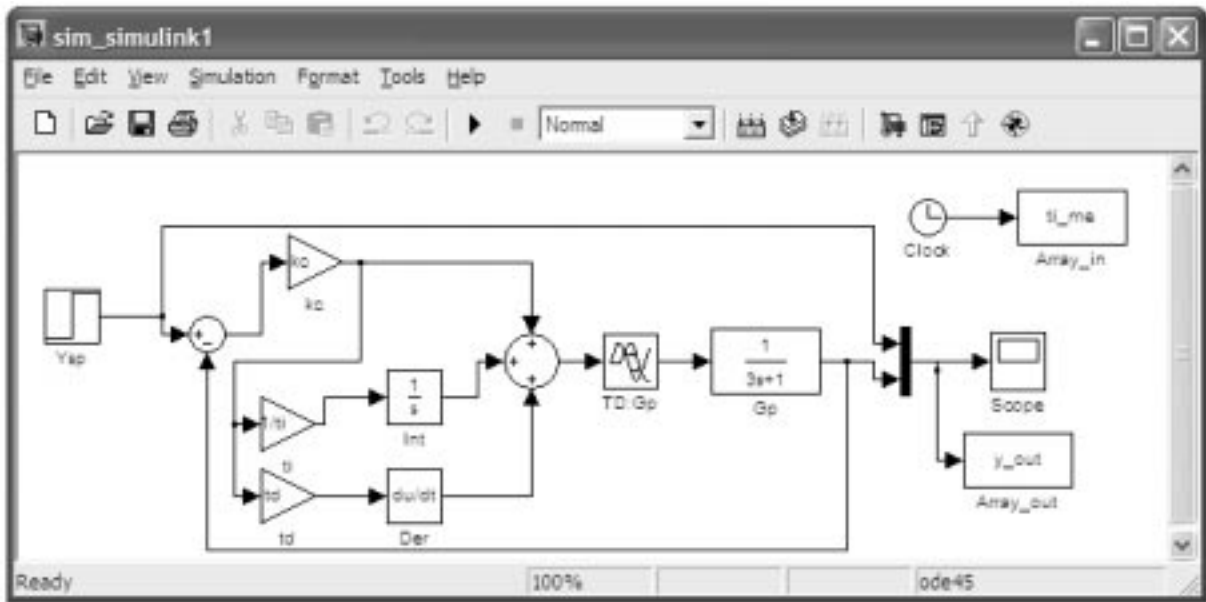


Fig. 2. Simulink model of feedback controller system.

**CASE STUDIES**

The first two case studies analysed in this work are control problems represented by the standard block diagram depicted in Fig. 1 [5].

In this figure, Y is the controlled variable, U is the manipulated variable, D is the disturbance variable, P is the controller output, E is the error signal, Ym is the measured value of Y, Ysp is the set point, Y-tilde-sp is the internal set point (used by the controller), Gc is the controller transfer function, Gv is the transfer function for final control element, Gp is the process transfer function, Gd is the disturbance transfer function, Gm is the transfer function for measuring element and transmitter and Km is the steady-state gain for Gm.

The third case corresponds to control of a non-isothermal continuous stirred tank reactor.

*Case study 1*

Consider a feedback controller system presented in Fig. 1 and the following transfer functions:

$$G_p = \frac{e^{-s}}{3s + 1} \quad G_d = G_p \quad G_v = 1 \quad G_m = 1$$

Calculate the PI controller settings using the ITAE performance index.

Figure 2 depicts the model develop in Simulink to simulate the feedback controller system. Appendix A presents the MATLAB files for this case study.

Table 1 presents the controller parameters obtained by ITAE and by Ziegler-Nichols tuning relations.

The system responses for a step change at time

Table 1. Controller settings for a first-order system plus dead time

	ITAE	Ziegler-Nichols
kc	1.76	2.41
τ <sub>i</sub>	3.14	2.97

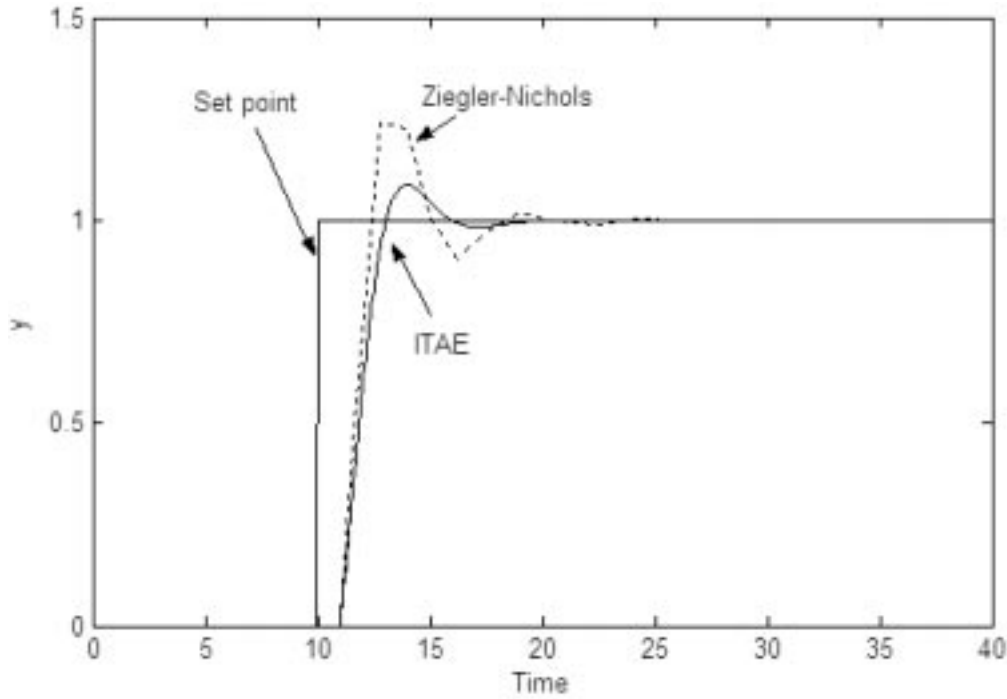


Fig. 3. Comparison of set point responses for case study 1.

10 in the set point are shown in Fig. 3. The simulation results indicate that set point response presents better performance using ITAE tuning methodology.

*Case study 2*

Consider a feedback controller system presented in Fig. 1 and the following transfer functions:

$$G_p = \frac{e^{-2s}}{s^2 + 0.7s + 1} \quad G_d = G_p \quad G_v = 1 \quad G_m = 1$$

Calculate the PID controller settings using the ITAE performance index.

The Simulink model of this system is very similar to that presented in Fig. 2. The difference is only in  $G_p$ , now a second-order transfer function.

The controller parameters for ITAE and for Ziegler-Nichols method are listed in Table 2.

Comparison results obtained by the two tuning methods for PID controller for the system responses for a step change in set point at time 10 is shown in Fig. 4. The closed-loop response is more sluggish and less oscillatory if the controller parameters are obtained by ITAE index.

Table 2. Controller settings for a second order system plus dead time

	ITAE	Ziegler-Nichols
kc	0.28	0.39
$\tau_i$	0.96	3.44
$\tau_d$	1.18	0.86

*Case study 3*

Controller design based on the ITAE performance index for a non-isothermal continuous stirred tank reactor. A non-isothermal continuous stirred tank reactor is sketched in Fig. 5 [3]. Constants for holdups and flow rates will be assumed, plus first-order kinetics and constant cooling-water temperature.

The equations describing the system are:

$$\frac{dC_A}{dt} = \left(\frac{F}{V}\right) C_{A0} - \left(\frac{F}{V}\right) C_A - C_A \alpha e^{-E/RT} \quad (2)$$

$$\begin{aligned} \frac{dT}{dt} = & \left(\frac{F}{V}\right) T_0 - \left(\frac{F}{V}\right) T - \left(\frac{\lambda}{C_p \rho}\right) C_A \alpha e^{-E/RT} \\ & - \left(\frac{UA}{C_p V \rho}\right) (T - T_j) \end{aligned} \quad (3)$$

$$\frac{dT_J}{dt} = \frac{F_J(T_{J0} - T_J)}{V_J} + \left(\frac{UA}{C_J V_J \rho_J}\right) (T - T_J) \quad (4)$$

$$F_J = \bar{F}_J - kc \left[ (T^{set} - T) + \frac{1}{\tau_i} \int_0^\infty (T^{set} - T) dt \right] \quad (5)$$

Equation (5) represents a proportional-integral controller that changes the jacket flow rate as the reactor temperature varies. Possible load variables are feed concentration ( $C_A$ ) and feed temperature ( $T_0$ ). Table 3 lists parameters and steady-state values. Barred quantities are steady-state values. Figure 6 shows this model developed in Simulink.

The controller parameters obtained by MATLAB/Simulink through the implementation of the ITAE performance index are  $kc = 3.13$  ( $\text{ft}^3/\text{hr})/\text{R}$  and  $\tau_i = 2.43$  hr. The responses of  $T$  and  $C_A$  after a step change of 10 R in  $T^{set}$  at time of 2 hr are presented in Fig. 7. The simulation results show that ITAE tuning method is satisfactory for set point response.

*Acknowledgements*—This work was supported by FCT (Fundação para a Ciência e Tecnologia). MATLAB version 6.5 (R13) and

Simulink version 5.0 (R13) were used in this work. All MATLAB and Simulink files that support this paper can be found in the following URL: [http://paginas.fe.up.pt/~fgm/IJEE\\_ITAE](http://paginas.fe.up.pt/~fgm/IJEE_ITAE)

### CONCLUSIONS

A procedure for tuning PID controllers with Simulink and MATLAB is proposed. It is shown how MATLAB Optimization Toolbox can interact easily with Simulink to implement the ITAE

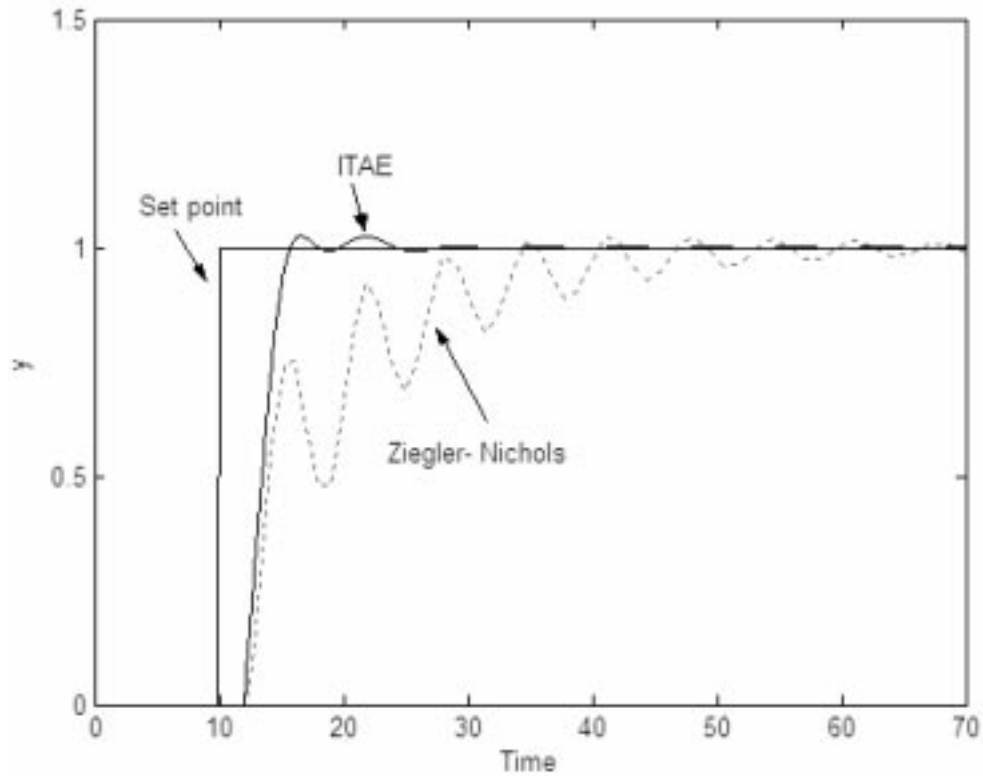


Fig. 4. Comparison of set point responses for case study 2.

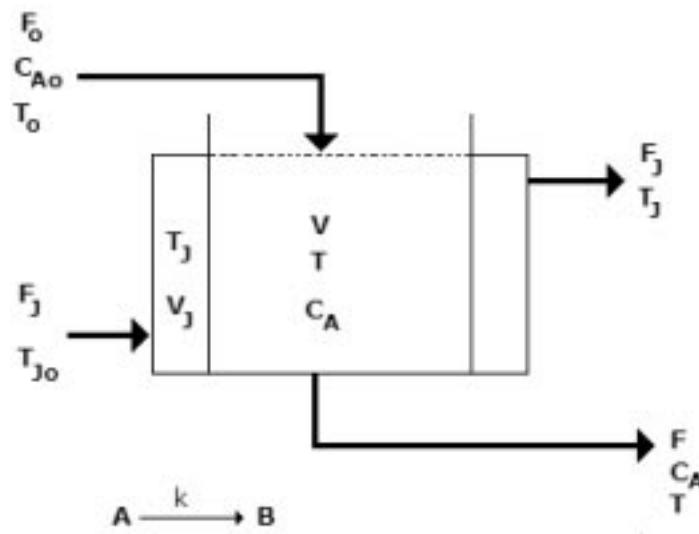


Fig. 5. Non-isothermal CSTR.

Table 3. Non-isothermal CSTR parameter values

$\bar{F} = 40 \text{ ft}^3/\text{hr}$	$U = 150 \text{ Btu}/(\text{hr ft}^2 \text{ R})$
$\bar{V} = 48 \text{ ft}^3$	$A = 250 \text{ ft}^2$
$\bar{C}_{A0} = 0.50 \text{ mole}/\text{ft}^3$	$T_{J0} = 530 \text{ R}$
$\bar{C}_A = 0.245 \text{ mole}/\text{ft}^3$	$\bar{T}_0 = 530 \text{ R}$
$\bar{T} = 600 \text{ R}$	$\lambda = -30000 \text{ Btu}/\text{mole}$
$\bar{T}_J = 594.6 \text{ R}$	$C_p = 0.75 \text{ Btu}/(\text{lbm R})$
$\bar{F}_J = 49.9 \text{ ft}^3/\text{hr}$	$C_J = 1.0 \text{ Btu}/(\text{lbm R})$
$V_J = 3.85 \text{ ft}^3$	$\rho = 50 \text{ lbm}/\text{ft}^3$
$\alpha = 7.08\text{E}10 \text{ hr}^{-1}$	$\rho_J = 62.3 \text{ lbm}/\text{ft}^3$
$E = 30000 \text{ Btu}/\text{hr}$	
$R = 1.99 \text{ Btu}/(\text{mole R})$	

critierion. The three case studies demonstrated that MATLAB and Simulink is a powerful and easy-to-use tool to tune PID controllers using the ITAE index.

The approach presented in this work can enhance considerably the learning progress of process control. An enormous set of process models can be implemented in Simulink and the behaviour of the control systems can be analysed very easily. Another relevant aspect of this procedure is the connection between two knowledge fields that usually are presented in engineering curriculum: process control and optimization.

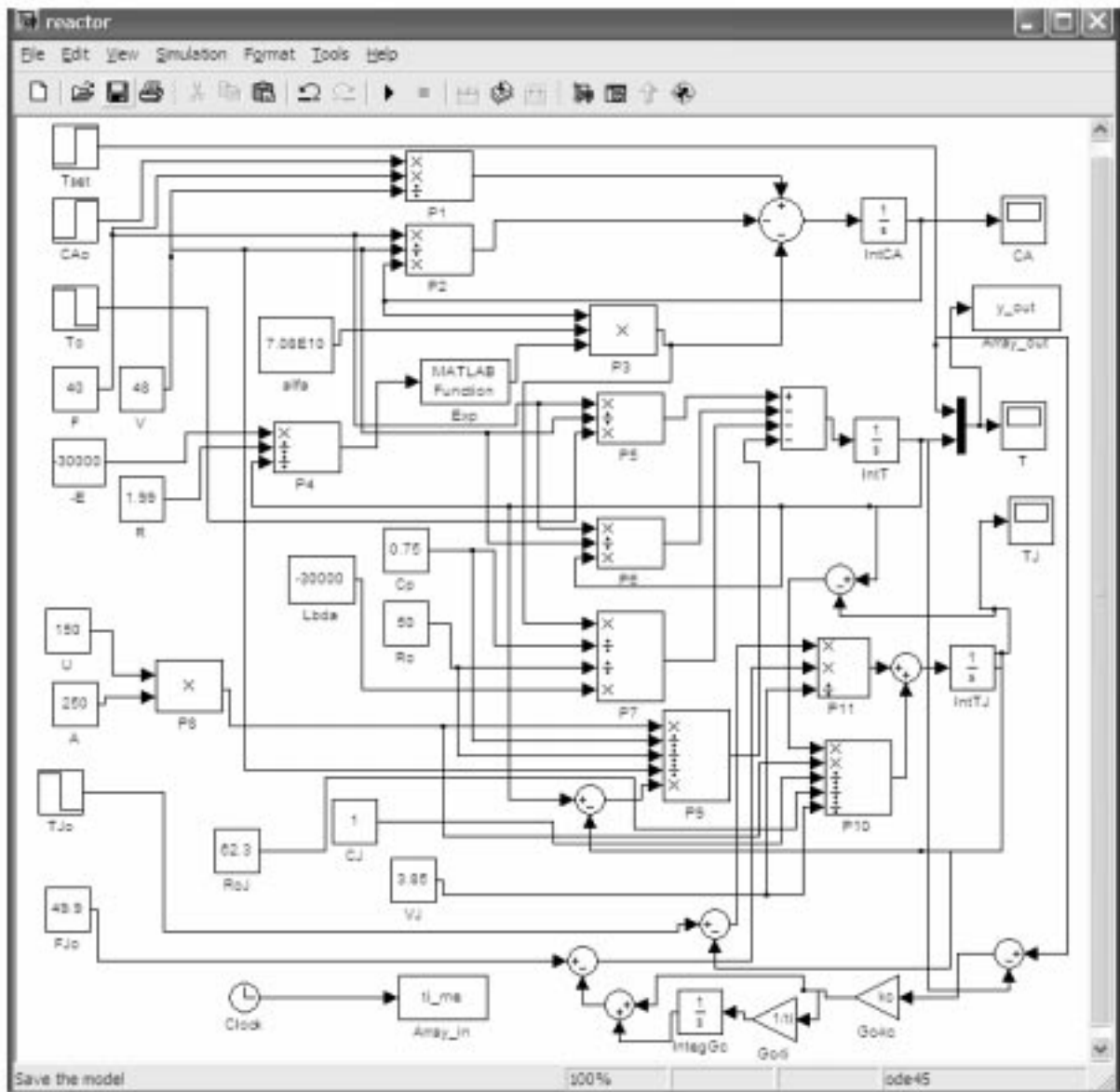


Fig. 6. Simulink model of non-isothermal CSTR.

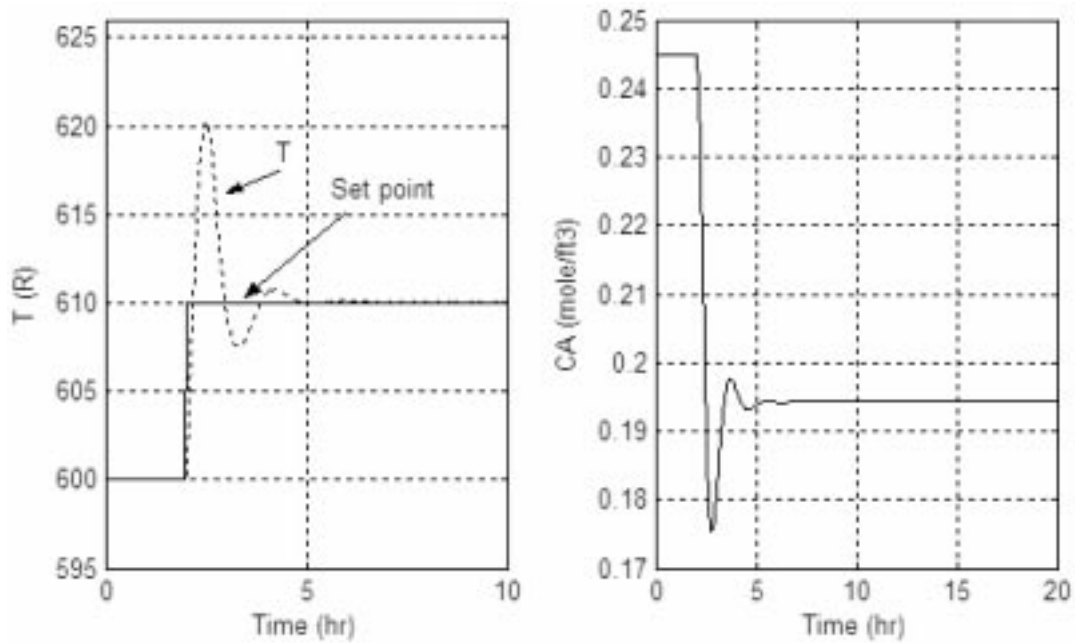


Fig. 7. Simulation results for temperature and concentration for a step change in temperature set point.

## REFERENCES

1. K. J. Åström, and T. Hägglund, Revisiting the Ziegler-Nichols step response method for PID control, *J. Process Control*, **14**, 2004, pp. 635–650.
2. S. C. Chapra, and R. P. Canale, *Numerical Methods for Engineers*, McGraw-Hill International Editions, New York (1998).
3. W. L. Luyben, *Process Modeling, Simulation and Control for Chemical Engineers*, McGraw-Hill, Tokyo (1973).
4. F. G. Martins and M. A. N. Coelho, Application of feedforward artificial neural networks to improve process control of PID-based control algorithms, *Computers and Chemical Engineering*, **24**, 2000, pp. 853–858.
5. D. E. Seborg, T. F. Edgar and D. A. Mellichamp, *Process Dynamics and Control*, Wiley, New York (2004).

## APPENDIX A

### *MATLAB files for case study 1*

The task in MATLAB/Simulink is to create a simulation model in Simulink. For this case, this Simulink model is presented in Fig. 2.

The first MATLAB file is the `main.m` and invokes a function `fminsearch` from MATLAB Optimization Toolbox. This file contains time to stop the simulation, sample time and the initial estimations for controller parameters;

```
%Main program
global kc
global ti
global td
global y_out
global ti_me
t_end=100;
h=.05;
x0=[5 10];
x=fminsearch(@fobs,x0,[],t_end,h)
```

The file `fobs.m` has the MATLAB source code with a call to execute the simulation in Simulink and the implementation of the multiple-application Simpson's 1/3 rule:

```
% Objectivefunction
function f=fobs(x,t_end,h)
global kc
global ti
global td
global y_out
global ti_me
kc=x(1);
ti=x(2);
td=0;
tt=(0:h:t_end);
[t,y]=sim('sim_simulink1',tt);
f=0;
l=length(ti_me);
f=ti_me(1)*abs(y_out(1,1)-y_out(1,2));
for i=2:2:l-1
f=f+4*ti_me(i)*abs(y_out(i,1)-y_out(i,2));
end
for i=3:2:l-2
f=f+2*ti_me(i)*abs(y_out(i,1)-y_out(i,2));
end
f=h/3*f
```

It is also relevant to note that `kc` and `ti` must be declared in Simulink as global variables. This is realized through the option *Simulation parameters* in the menu *Simulation* and choosing *Advanced, inline parameters* and click in *Configure*. Then, specify `kc` and `ti` as *Global parameters*.

**Fernando G. Martins** holds a Ph.D. degree in Chemical Engineering from the Faculty of Engineering (University of Porto). he is now Assistant Professor of Chemical Engineering Department at the Chemical Engineering at Faculty of Engineering, University of Porto. He is a member of LEPÆ, a research laboratory dedicated to the fields of process system engineering and complex systems understanding and development in the areas of chemical processes, environment and energy engineering.