24

Modelling and Simulation of pH Neutralization Plant Including the Process Instrumentation

Claudio Garcia and Rodrigo Juliani Correa De Godoy
Escola Politécnica da Universidade de São Paulo
Brazil

1. Introduction

In this chapter, we aim to show the facilities available in Matlab/Simulink to model control loops. For this, it is implemented a simulator in Matlab/Simulink, which shows details about the modelling of each component in a pH neutralization plant, where pH and tank level are simulated and controlled in a CSTR (Continuous Stirred Tank Reactor). Both loops are modelled considering the plant itself, the measuring and actuating instruments and the control algorithms. The pH neutralization is normally a difficult process to control, due to the non-linearity caused by the titration curve (Asuero & Michalowski, 2011), mainly when strong acids and bases are involved.

It is presented the equations (linear or non-linear) corresponding to the loop elements and how they are translated into a Simulink model and how blocks are created in Simulink to ensemble the components of the loop. Another objective is to show a case in which a P&ID diagram of the control system is presented and how it is used to reach an equivalent Simulink model.

All the model parameters, initial conditions and data related to the simulations are inserted in a Matlab file and it is stressed how it can generate a well-documented project. It is also addressed the option to create a batch in Matlab, which enables to automatically simulate the plant in different conditions and to plot graphs of different responses, in order to compare the behaviour of distinct situations. To exemplify that and validate the model, tests were performed considering set point variations (servo mode) and disturbances (regulatory mode).

2. Process description

The P&ID of the pH neutralization plant is shown in Figure 1 (ISA, 2009). In it, pH is affected by the variations in acid and base flows, where the first is considered a disturbance and the second the manipulated variable. Level is affected by the input and output flows, where the input values are considered disturbances and the output flow is the manipulated variable. This model includes the following items:

a. modeling of pH in the CSTR;

b. modeling of level in the CSTR;
c. modeling of pH (AE/AITY-10) and level (LIT-20) meters;
d. modeling of two kinds of actuators for the pH loop: dosing pump (FZ-11) and control valve (FV-12) driven by an I/P converter (FY-12);
e. modeling of a solenoid valve (LV-20) as the actuator of the level loop;
f. modeling of the meters for measuring the acid flow disturbance (FIT-30) and the base flow (FIT-10); and
g. inclusion of digital PI regulators to control pH (AIC-10) and level (LIC-20).

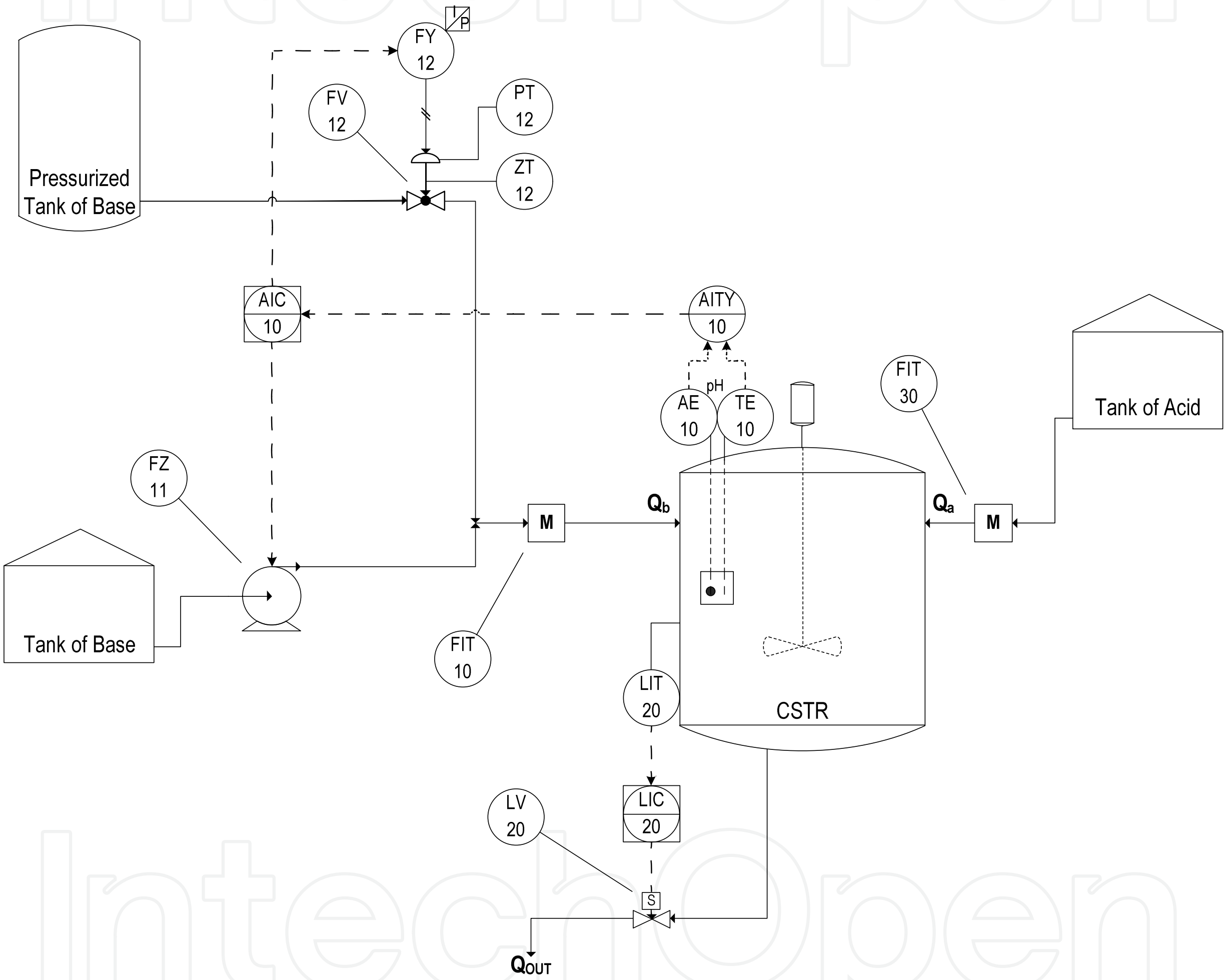

Fig. 1. P&ID of the pH neutralization plant

One important point to be emphasized is that the two kinds of actuators for the pH loop represent a linear one (dosing pump) and a non-linear actuator, as the control valve is modeled considering that it has large friction coefficients, so representing a problematic valve. The idea is to show the effects in the closed loop variability of an actuator (Rinehart & Jury, 1997) which is performing well and another one which needs maintenance. To enable analyzing the friction in the control valve, it is assumed that its stem position (ZT-12) and the actuator pressure (PT-12) are measured.

The P&ID diagram in Figure 1 is converted in the Simulink diagram in Figure 2.

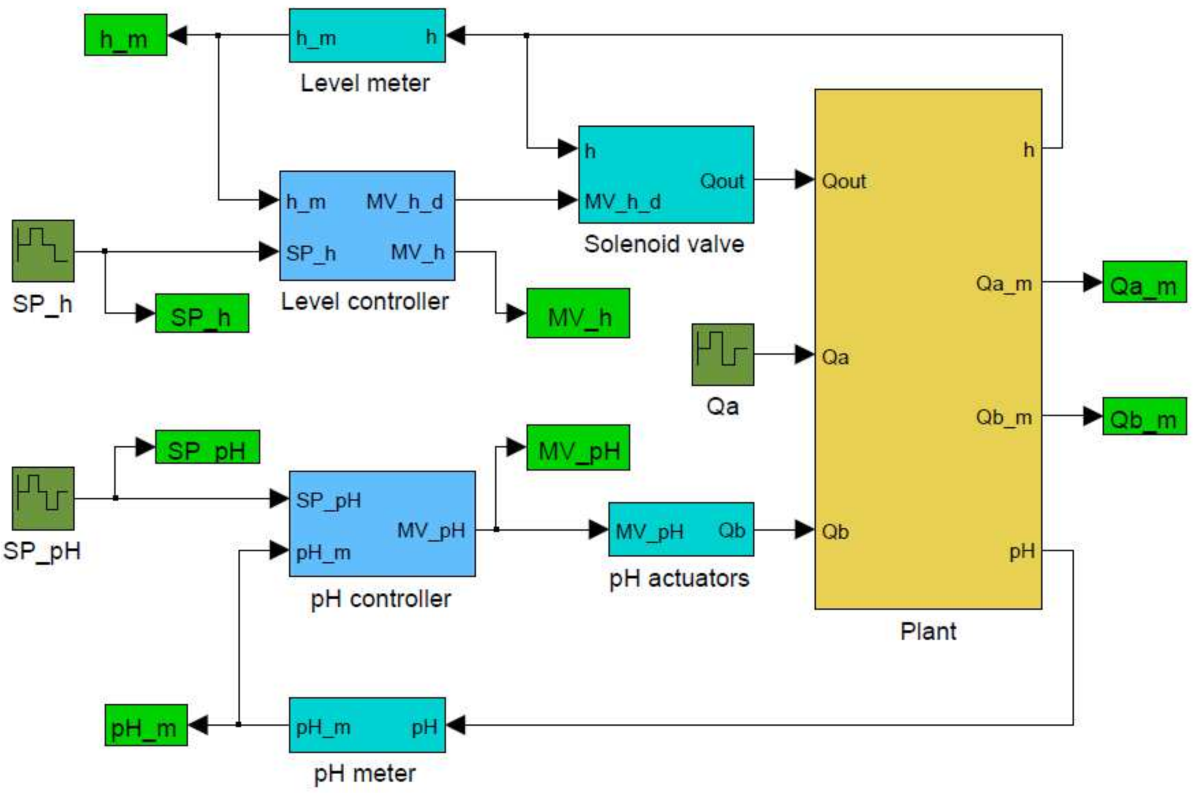

Fig. 2. Representation of the pH neutralization plant model in Simulink.

Each block of this model is individually presented in the next section.

3. Mathematical modelling and implementation in Simulink

Each element of the plant is next modeled (Garcia, 2005).

3.1 pH neutralization process

pH is related to the concentration of the ions [H+] through the following logarithmic function:

$$pH \equiv -\log_{10}[H^+] \tag{1}$$

The process here investigated is the neutralization of a strong acid effluent (HCl) in a CSTR by a strong base (NaOH). This process is modeled according to (Jacobs et al., 1980). They used a first order dynamics model with titration curve as the nonlinearity. The reactions that occur are:

$$\begin{cases} HCl + NaOH \rightarrow H_2O + NaCl \\ H^+ + OH^- \rightarrow H_2O \end{cases} \tag{2}$$

The possible amount of effluent to be neutralized is defined mainly by the concentration of the reactants. If the mixture is perfect and instantaneous, the ionic concentrations $[Cl^-]$ and $[Na^+]$ in the CSTR can be related to the flows of acid Q_a and of base Q_b and to the input concentrations $[Cl_{in}^-]$ and $[Na_{in}^+]$, according to the following equations:

$$V\frac{d}{dt}[Cl^-] = [Cl_{in}^-] \cdot Q_a - [Cl^-] \cdot Q_{out} \tag{3}$$

$$V\frac{d}{dt}[Na^+] = [Na^+_{in}]\cdot Q_b - [Na^+]\cdot Q_{out} \tag{4}$$

where V corresponds to the volume of fluid inside the CSTR.
The concentrations must also satisfy the electro-neutrality equation:

$$[Na^+] + [H^+] = [Cl^-] + [OH^-] \tag{5}$$

which, together with the dissociation equation for water:

$$[H^+]\cdot[OH^-] = k_W = 10^{-14} \tag{6}$$

relates these concentrations to $[H^+]$ and therefore to pH. This relationship is expressed in terms of the difference of the ionic concentrations X:

$$X \equiv [OH^-] - [H^+] \tag{7}$$

that combined with equation (5) results in:

$$X = [Na^+] - [Cl^-] \tag{8}$$

Combining (6) and (7) results in:

$$\begin{cases} [H^+] = \frac{X}{2}\cdot\left(\sqrt{1+\frac{4.k_W}{X^2}}-1\right) & \text{if } X > 0 \\ [H^+] = -\frac{X}{2}\cdot\left(\sqrt{1+\frac{4.k_W}{X^2}}+1\right) & \text{if } X < 0 \\ [H^+] = \sqrt{k_W} & \text{if } X = 0 \end{cases} \tag{9}$$

The equation describing the process dynamics is obtained by subtracting (3) from (4) and using (8), resulting in:

$$V\frac{dX}{dt} = [Na^+_{in}]\cdot Q_b - [Cl^-_{in}]\cdot Q_a - X\cdot Q_{out} \tag{10}$$

The time constant τ of the process is dependent on the residence time and is given by:

$$\tau = \frac{V}{Q_{out}} \tag{11}$$

Equations (1), (9) and (10) correspond to the pH neutralization model. It is considered that the CSTR is at room temperature.
This model, implemented in Simulink, is shown in Figure 3.

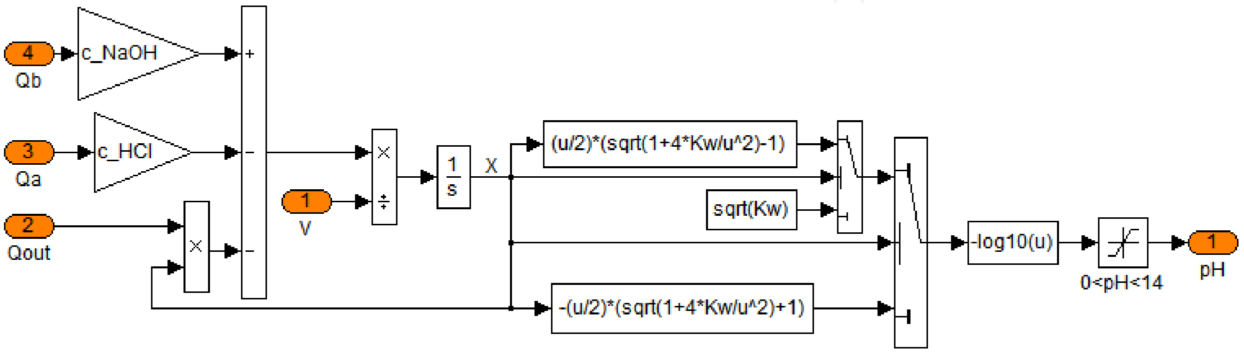

Fig. 3. Model of the pH neutralization process.

3.2 CSTR level

The level h in the CSTR is modeled through a mass balance:

$$\rho \cdot \frac{dV}{dt} = A \cdot \rho \cdot \frac{dh}{dt} = \rho_a \cdot Q_a + \rho_b \cdot Q_b - \rho \cdot Q_{out} \tag{12}$$

where A is the surface area of the CSTR, ρ is the specific mass of the mixture inside the CSTR, ρ_a is the specific mass of the acid solution associated with the acid flow Q_a, ρ_b is the specific mass of the base solution associated with the base flow Q_b and Q_{out} is the output flow of the CSTR.

The implementation of this model in Simulink is presented in Figure 4.

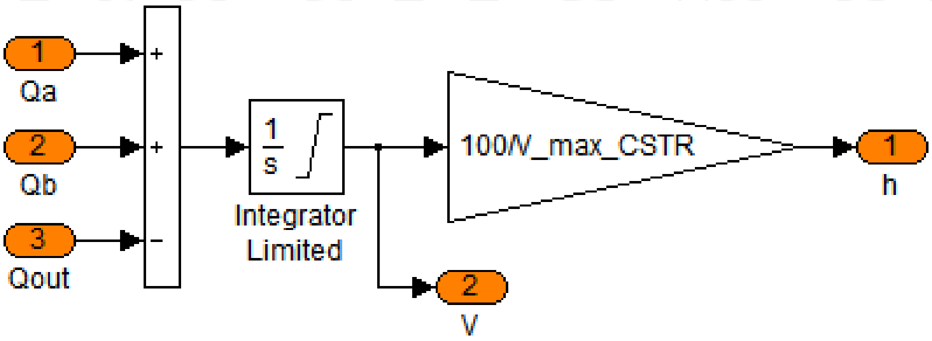

Fig. 4. Model of the level in the CSTR.

The integration of the pH and level models derives the plant model, as shown in Figure 5.

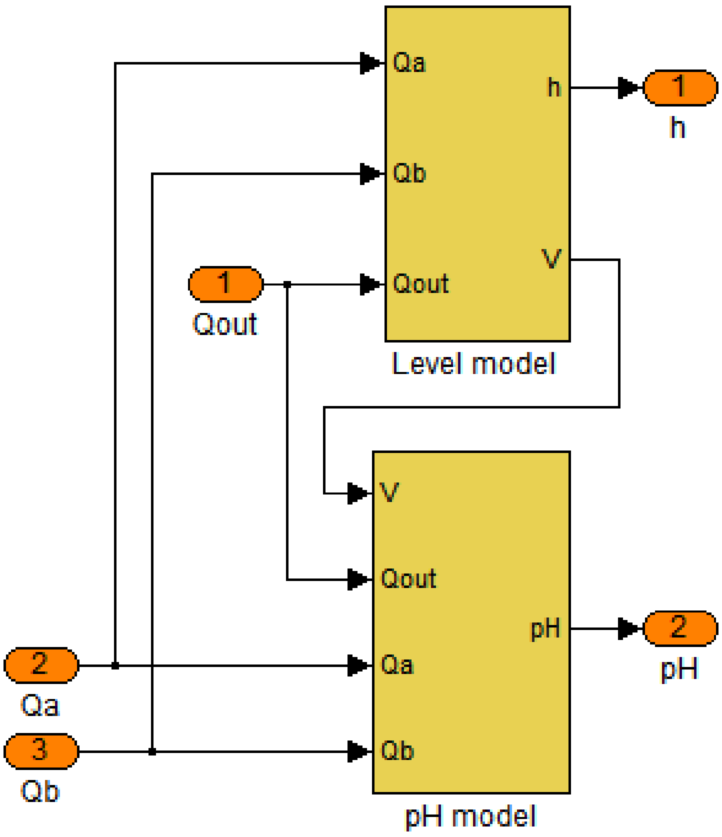

Fig. 5. Model of the plant.

3.3 pH, level and flow meters

All the measuring instruments are modeled considering that their dynamics is described by a first order system:

$$G_{meter}(s) = \frac{Y_{meas}(s)}{Y(s)} = \frac{K_{meter}}{\tau_{metr} \cdot s + 1} \quad (13)$$

In (13) Y is the variable to be measured, Y_{meas} is the measured value of the variable Y, K_{meter} is the meter gain and τ_{meter} is the meter constant time. For instance, the level meter is modeled as shown in Figure 6.

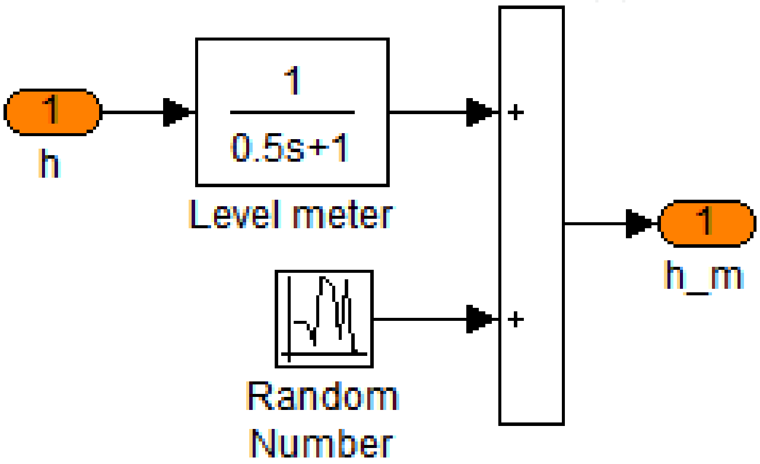

Fig. 6. Model of the lever meter.

It can be noticed in Figure 6 that it is added to the output of the meter a random number, representing measurement noise. In order to show other available forms in Simulink to represent a first order system, the pH meter in Figure 7 is represented through state space.

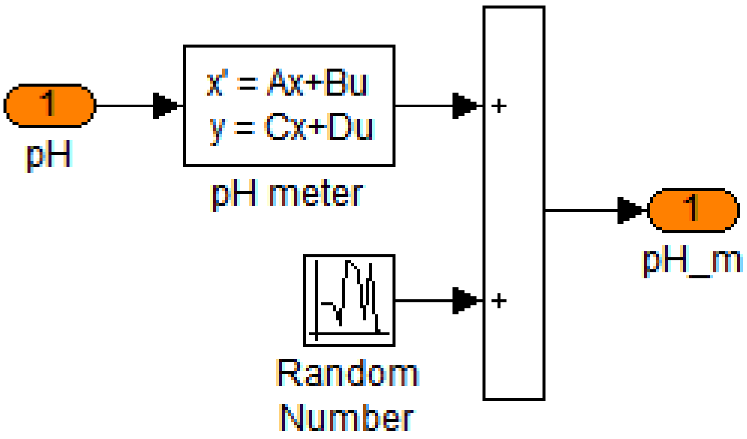

Fig. 7. Model of the pH meter.

An expanded view of the plant in Figure 5 is shown in Figure 8, where the meters of the flows Q_a and Q_b are included.

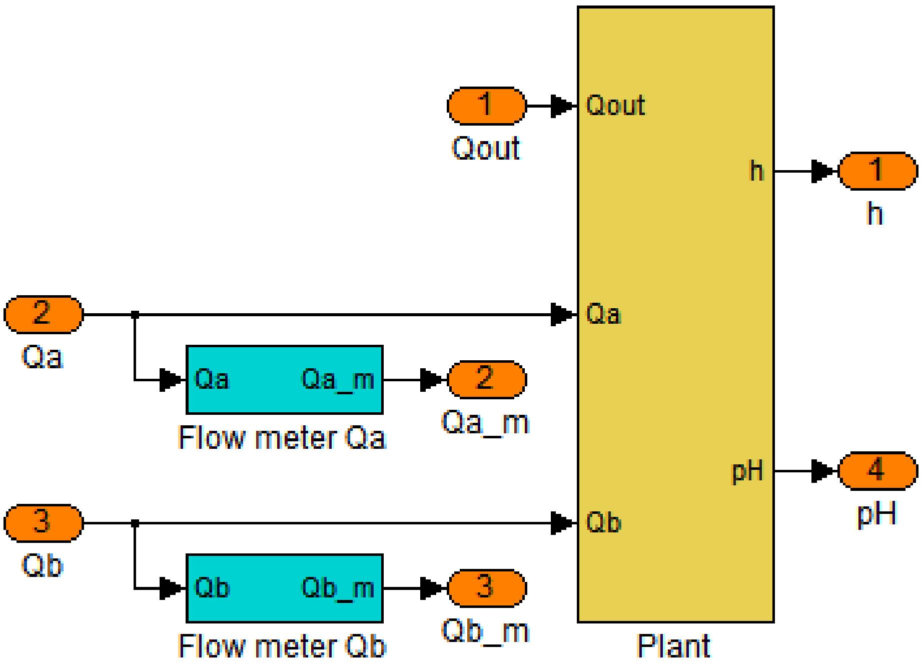

Fig. 8. Model of the plant including the flow meters for flows Q_a and Q_b.

The specific values of K_{meter} and τ_{meter} for each meter is presented in the Matlab file in Section 6.2 that describes the parameters of the plant and of the instruments.

3.4 pH actuators

As seen in Figure 1, the manipulation of the base flow occurs in two different modes: through a pump or via a control valve, as represented in Figure 9.

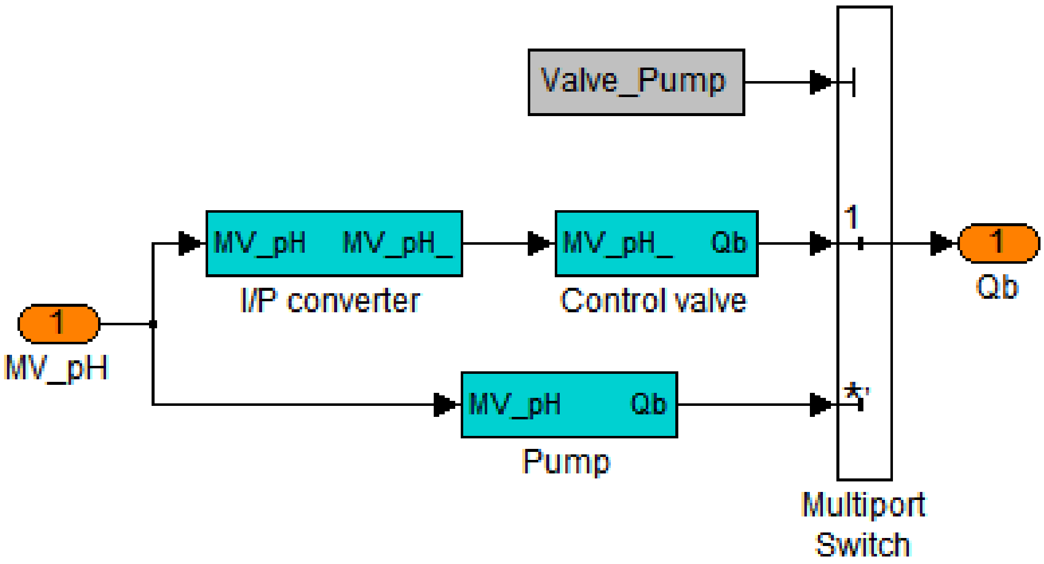

Fig. 9. Representation of the forms of manipulation of the base flow Q_b.

The pump has a linear behavior and its dynamics is very fast, so that it is considered negligible. Its model is presented in Figure 10.

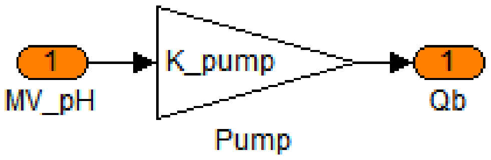

Fig. 10. Model of the pump.

The control valve is pneumatic, thus it needs an I/P converter to transform current into pressure. The model of this I/P converter is shown in Figure 11.

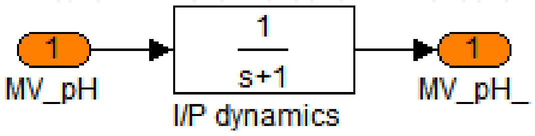

Fig. 11. Model of the I/P converter.

The control valve is modeled in two parts: the first represents its actuator, which is responsible for converting the pressure input signal into the valve stem movement, here denoted by x. This variable is assumed to vary in the range 0 to 1, that is, it is represented in p.u. (per unit). The second part is its body and it converts the valve stem movement x into flow Q_b. The model of the control valve is depicted in Figure 12.

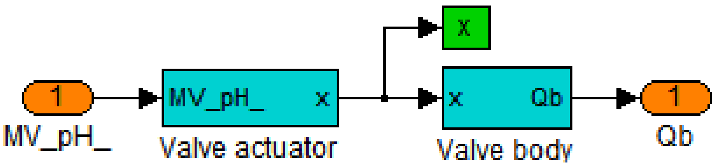

Fig. 12. Parts of the model of the control valve.

The valve actuator model is divided in two parts, as seen in Figure 13. The first part represents the actuator dynamics whereas the second part depicts the effect of the friction in the control valve.

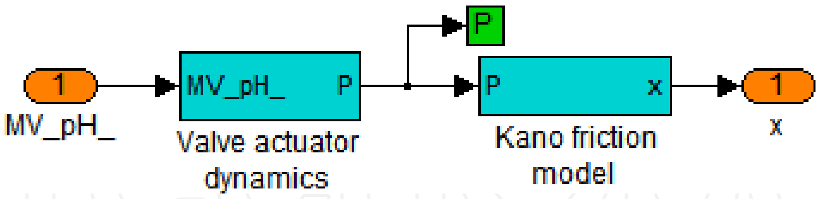

Fig. 13. Parts of the model of the control valve actuator.

The dynamics of the valve actuator is represented by a first order system, as shown in Figure 14.

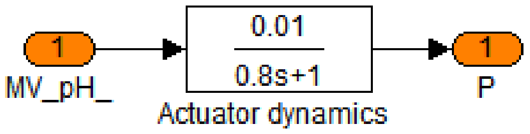

Fig. 14. Dynamics of the control valve actuator.

The Kano friction model, as presented in Figure 15, is one of the possible forms to represent the effects of friction in control valves. For more details about it, consult (GARCIA, 2008).

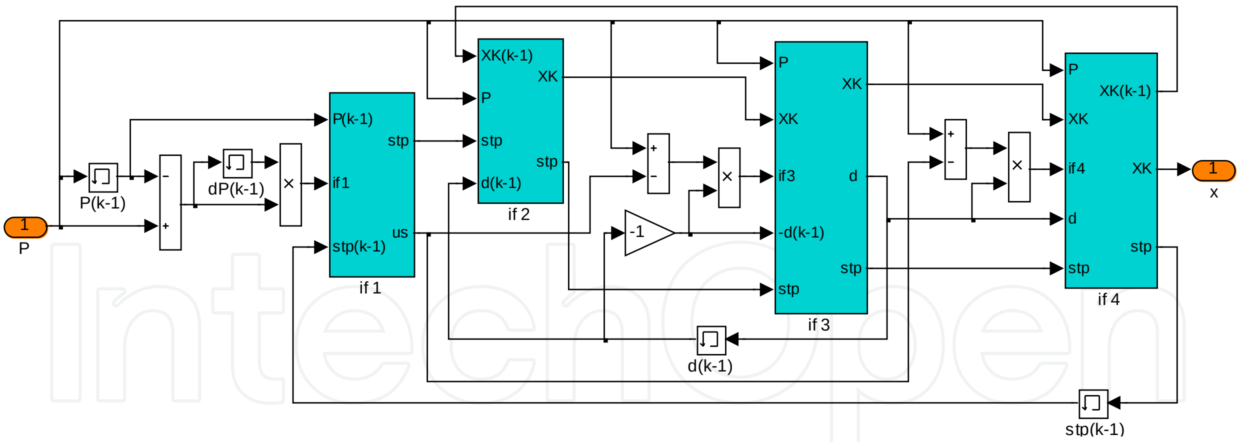

Fig. 15. Kano friction model of the control valve.

To complete the model of the control valve, its body is shown in Figure 16.

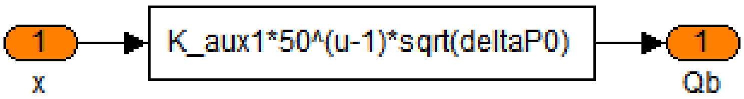

Fig. 16. Equation representing the control valve body.

The equation present in Figure 16 represents an equal percentage valve, which, when dealing with liquids, is described by the following equation:

$$Q_b = k_V \cdot C_V \cdot f(x) \cdot \sqrt{\frac{\Delta P}{\rho}} \tag{14}$$

where k_v is a term to adapt the engineering units of the equation, C_v is the flow coefficient of the valve, $f(x)$ represents how the flow varies as the valve stem position x changes, ΔP is the pressure drop in the control valve and ρ is the specific mass of the fluid in the flowing conditions. As the valve used is an equal percentage, it can be described by $f(x) = R^{x-1}$, where R is the valve rangeability.

3.5 Level actuator

The final control element of the level control loop is a solenoid valve, which model is depicted in Figure 17. It is supposed that its actuation is so fast, that its dynamics is considered negligible.

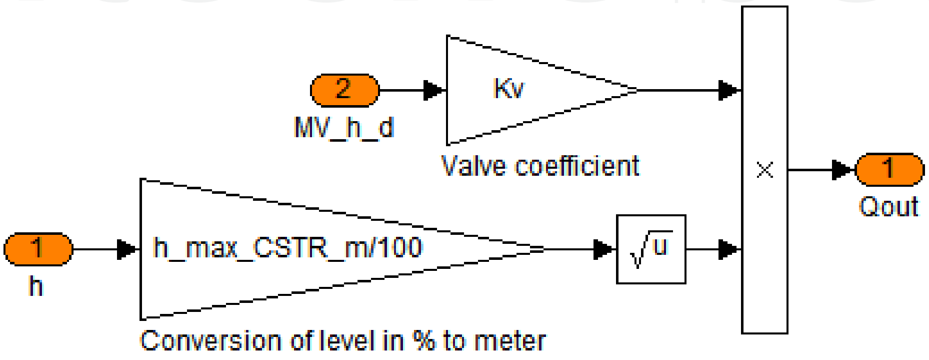

Fig. 17. Model of the solenoid valve.

3.6 pH controller

The acid solution is neutralized by NaOH that enters the CSTR with a flow governed by one of the pH actuators. These actuators are manipulated by a PI digital controller, which receives the pH set point (*SP_pH*) and the pH measured value (*pH_m*). This controller has reverse action and it is implemented through individual blocks of the Simulink, showing that it is possible to mount your own controller. It has the possibility to operate in automatic or manual. Besides, it is included an anti-reset windup algorithm, to avoid saturation of the integral action. Its algorithm is shown in Figure 18.

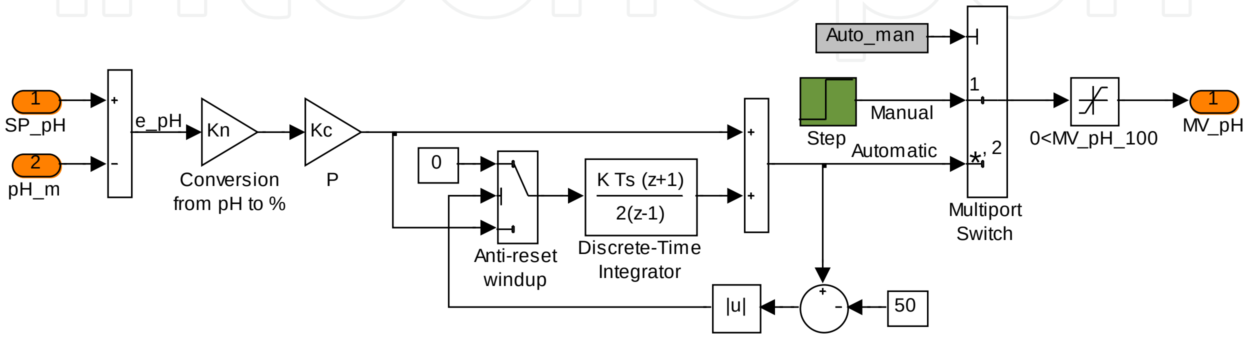

Fig. 18. Algorithm of the PI digital controller of pH.

The tuning parameters of the pH controller are K_c=1.34 and T_i=620 sec/rep.

3.7 Level controller

The level in the CSTR is controlled through the output flow using a solenoid valve, which is manipulated by a PI digital controller, which receives the level set point (*SP_h*) and the level measured value (*h_m*). It has direct action and is implemented through a single block of the Simulink. As the final control element of this loop is a solenoid valve, the continuous output of the PI controller has to converted to a time-discrete signal, operating in two levels: 1 (valve open) and 0 (valve closed). It is performed through a PWM converter. The level controller is shown in Figure 19.

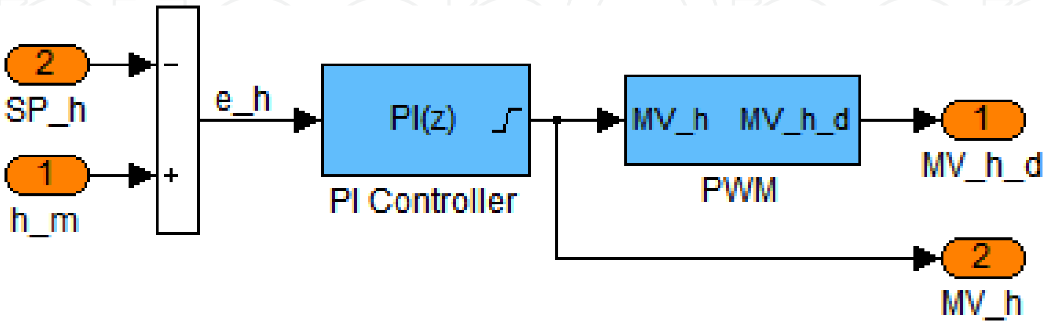

Fig. 19. Level controller with PI and PWM algorithms.

The PI controller is configured to have anti-reset windup by back-calculation and saturation in the output. The tuning parameters of the level controller are K_c=11.65 and T_i=271.52 sec/rep. Its algorithm is shown in Figure 20.

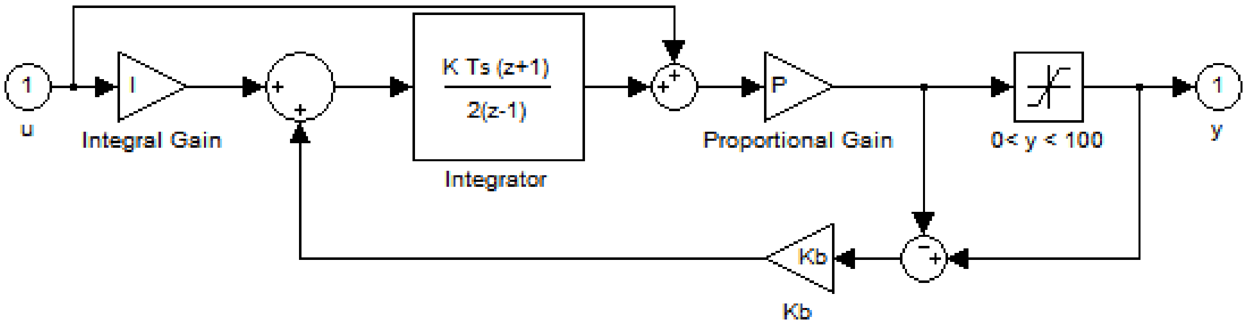

Fig. 20. Algorithm of the PI digital controller of level.

As the final control element of this loop is a solenoid valve, the continuous output of the PI controller has to be converted to a time-discrete signal, operating in two levels: 1 (valve open) and 0 (valve closed). It is performed through the PWM algorithm, as presented in Figure 21.
The PWM algorithm operates as the following equation:

$$\begin{cases} MV\_hd = 1 \text{ for } 0 < t_{PWM} \leq MV\_h \cdot T \\ MV\_hd = 0 \text{ for } MV\_h \cdot T < t_{PWM} \leq T \end{cases} \qquad (15)$$

where MV_hd is the valve position (open or closed), T is the PWM duty cycle, set to 10 seconds, MV_h is the continuous signal being transformed, the PI controller output, and t_{PWM} is the internal clock of the PWM.

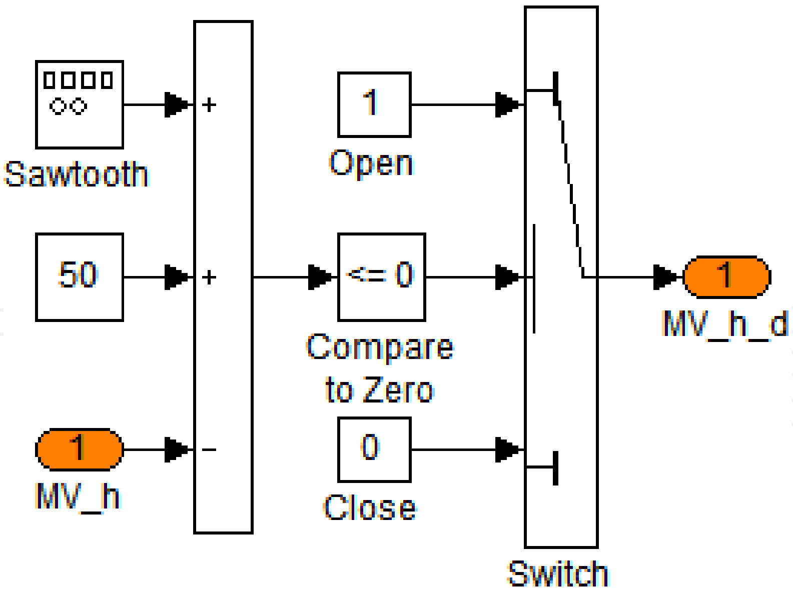

Fig. 21. Algorithm of the PWM block.

4. Model simulation

Here are presented some guidelines on how to create a batch file that configures and runs the model and also manages and presents the simulations results.

First of all, it is recommended to reset Matlab in order to avoid previously loaded data from interfering in the simulation.

Next, the model parameters, as presented in a Matlab script *(Configuring_pH_and_level_model.m)* in Section 6.2, should be loaded to the Workspace.

Following that, configuration parameters for the simulation should be loaded to the Workspace, such as simulation interval, step size of the simulation, set points for the controllers and operation mode selection variables.

The main difference between these two sets, that is, model parameters and simulation parameters, is that the first defines the system and generally is only changed when it is desired to change some characteristic of the model itself, while the later defines an operation mode for the model and are more frequently changed in order to run simulations in different conditions.

With the model implemented in Simulink and its parameters loaded to the Workspace, it is possible to run a simulation directly from Simulink with the start simulation command, or from the Matlab Workspace, with the sim function. However, a more interesting way of simulating the model is from a batch file, automatically running different simulations.

In order not to overwrite simulation results, after each simulation all the relevant variables should be copied to new unique variables.

Another useful hint is to save all generated data, mainly when running long simulations. This can be done through the save function.

After all the simulations are run or after each simulation is concluded, the generated results can be presented in many ways. A very concise and helpful method of presenting results is in graphs, which can be created with the plot function and its variants.

A batch was developed following these guidelines and is presented in a Matlab script (Simulating_pH_and_level_model) in Section 6.1.

5. Simulations results

For the presented model, two simulations are run, one with the dosing pump as the actuator in the pH loop and the other one with the control valve as its actuator. In both simulations the level and pH controllers are tested in servo and regulatory mode. For that, steps are applied to the set point of each controller and to the acid flow, the model disturbance variable.

Figures 22 and 23 show the simulated pH, level and acid flow for the plant with dosing pump and control valve as actuator, respectively. Each simulation was done for a time interval of 500,000 seconds, which correspond to a total of 278 hours or 11.6 days of continuous plant operation. This is one of the main advantages of using a simulator: to generate several hours of estimated system behavior in some seconds or minutes.

As can be seen in Figure 22, the level controller performs quite well, being capable of following the set point (servo mode) and rejecting disturbances (regulatory mode) with a low error. This controller is analyzed in greater detail ahead. It also shows that the pH controller with the dosing pump can follow set points and reject disturbances with no steady state error and fast response.

In Figure 23, the simulation results with the control valve active, it can be seen that the pH control cannot settle in the desired set point, presenting a high variability. This problem is better analyzed ahead. It can also be seen in this figure that the level controller performs well, even with an oscillatory base flow caused by the variability in the pH loop.

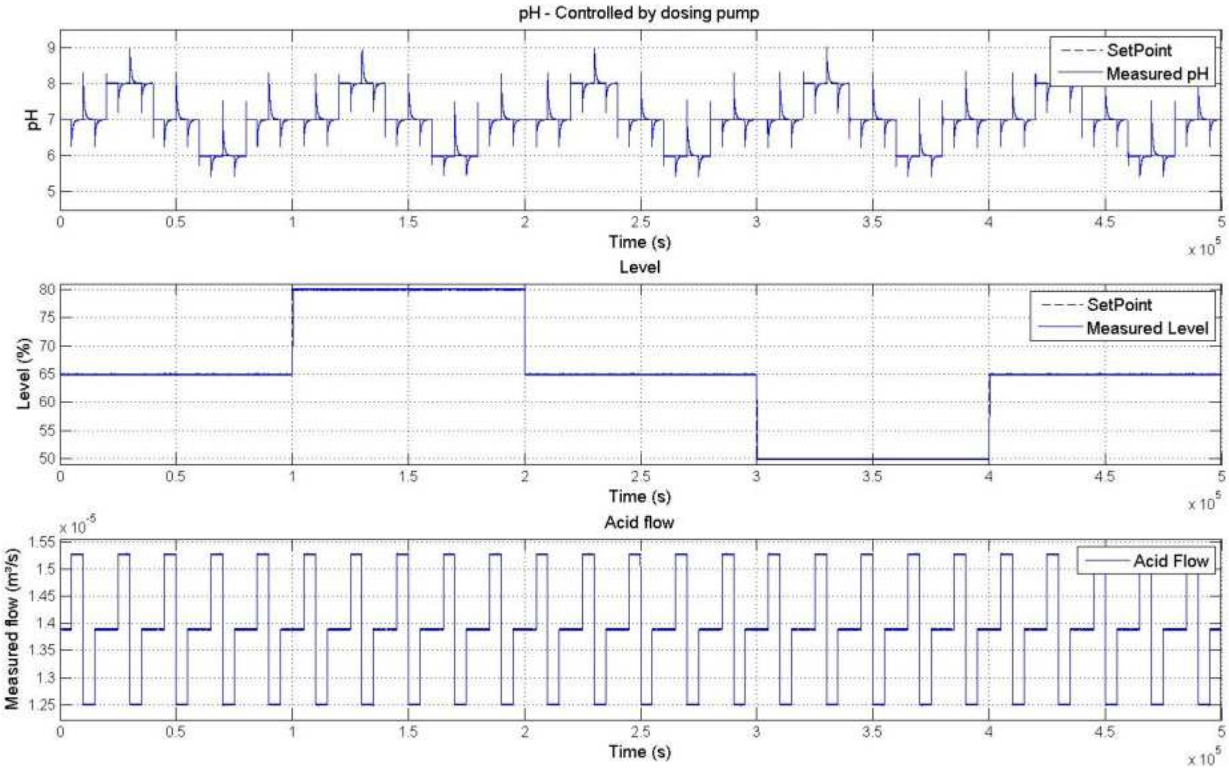

Fig. 22. All simulation results for pH, level and acid flow with dosing pump active.

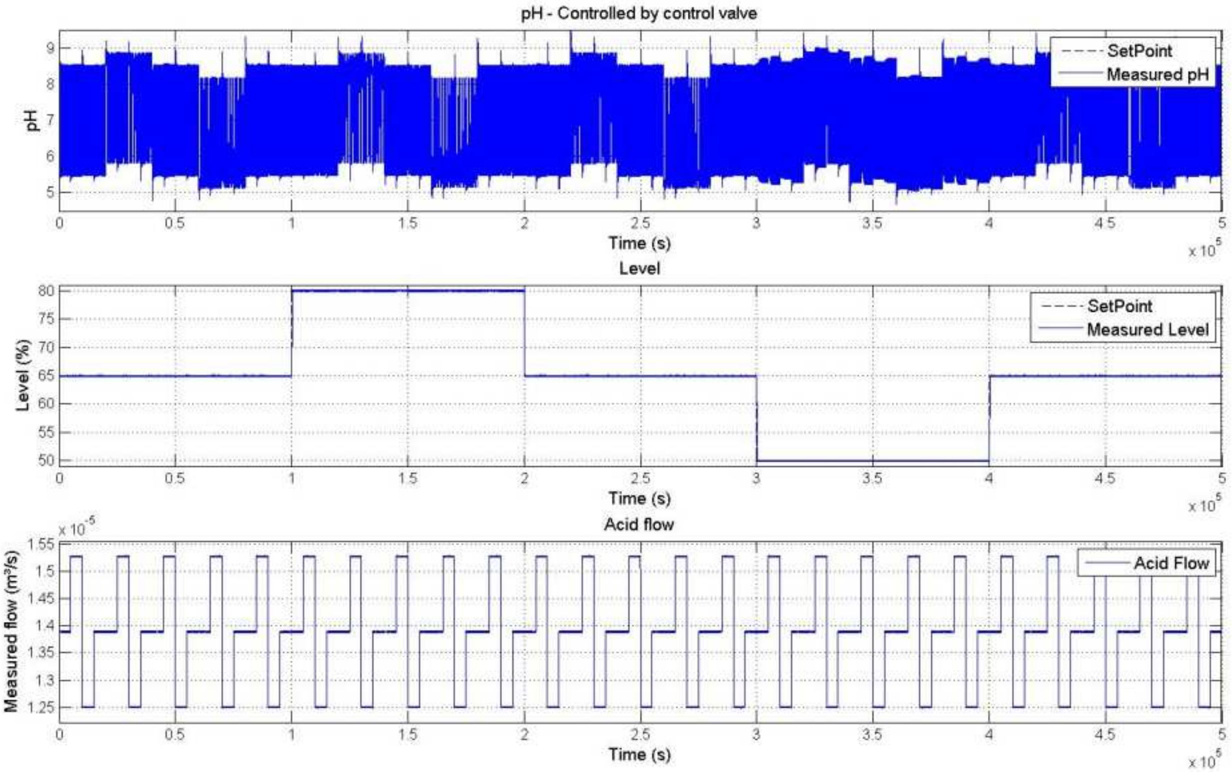

Fig. 23. All simulation results for pH, level and acid flow with control valve enable.

In order to better analyze the level controller performance, its responses with both actuators are plotted and zoomed at the steps in set point in Figure 24. As can be seen in this figure, the level controller performs almost identically in both simulations and does not deviate from the set point when disturbances are applied. This demonstrates a very good performance, either in servo or regulatory mode. The instants of step application are zoomed below the complete response in Figure 24. It can be seen that the settling time of the loop depends on the step direction and the CSTR level. It can also be noticed that there is no overshoot or undershoot in the level response.

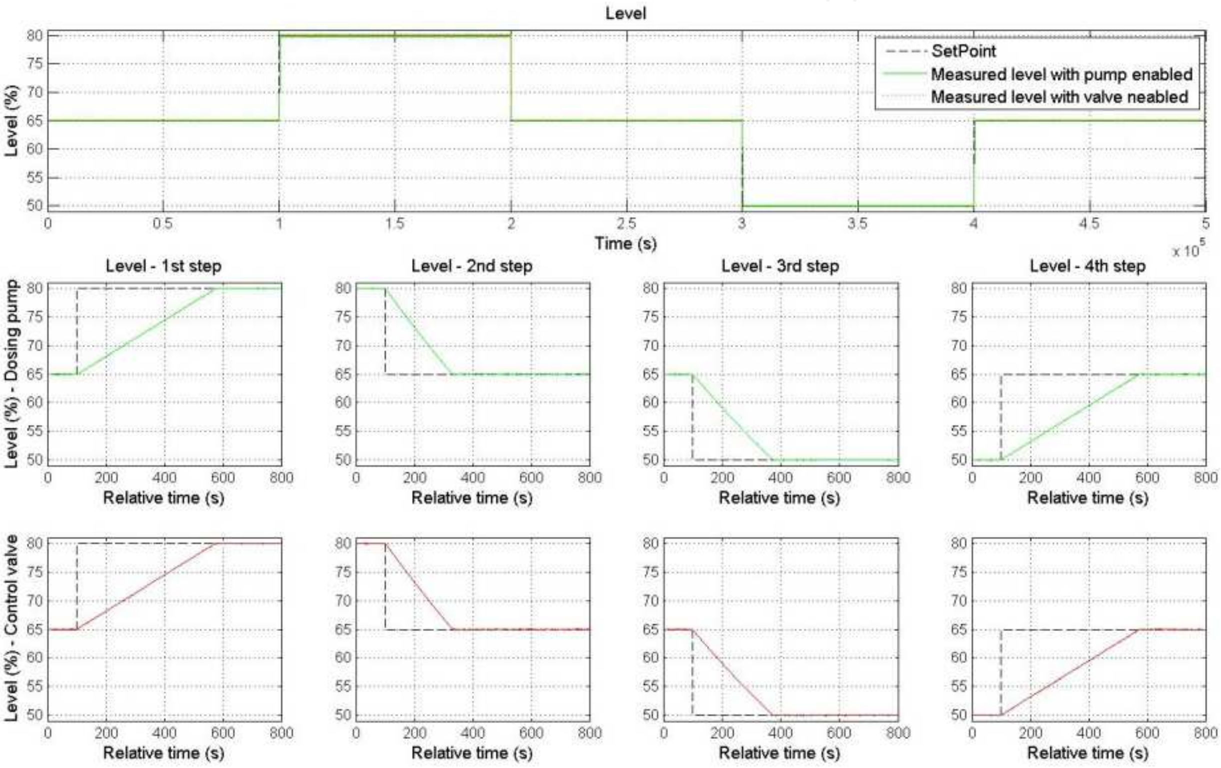

Fig. 24. Level controller response.

In order to analyze the level influence in the pH loop, this loop response with different levels (50%, 65% and 80%) is plotted in Figure25.

As can be noted in the upper graph in Figure 25, the pH response with the pump enabled is not significantly affected by variations in the level in the operation region, since the controller is capable of compensating these changes. With the control valve enabled, however, the pH response is more affected by the CSTR level. As the level decreases, oscillations in the measured pH become greater in amplitude and frequency. It happens because the valve has high friction and does not stop in a fixed position, but remains oscillating around a certain overture.

For a better analysis of the pH controller, Figures 26 and 27 show its response with the CSTR level constant at its nominal value. These figures also show the measured flows of acid and base and the control effort.

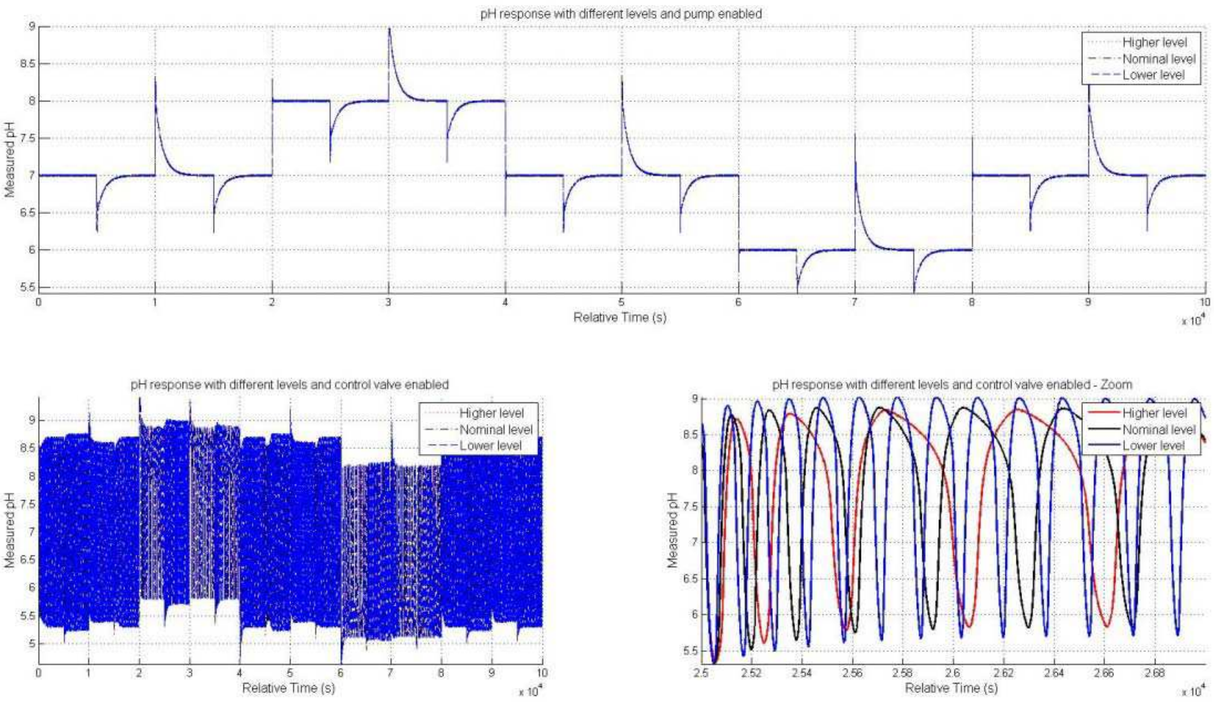

Fig. 25. Level influence in the pH response.

As can be observed in Figure 26, the controller output varies to compensate the pH error and is followed by the base flow, because the dosing pump is a linear and fast actuator.

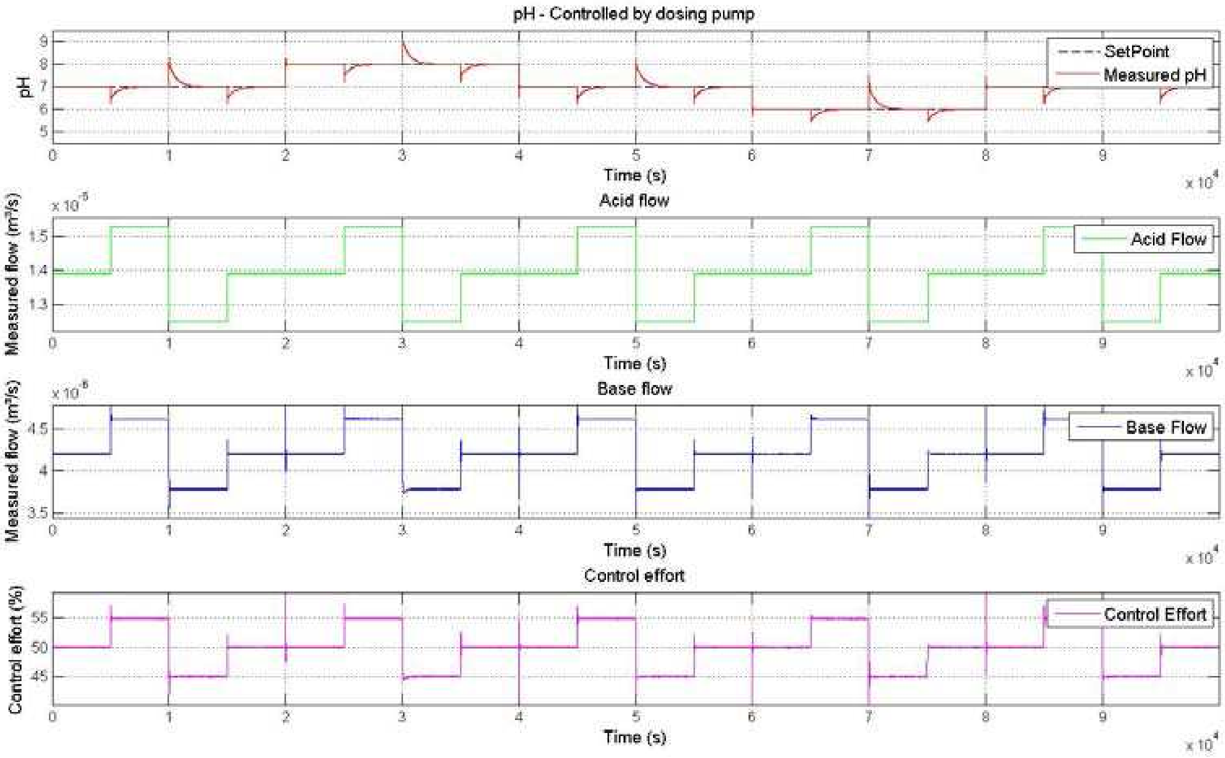

Fig. 26. pH controller response and control effort with dosing pump enabled.

The pH response with the control valve as actuator, shown in Figure 27, is oscillatory due to variability in the loop. As can be noted in the two lower plots, the controller output varies to compensate the pH error but the base flow does not change proportionally.

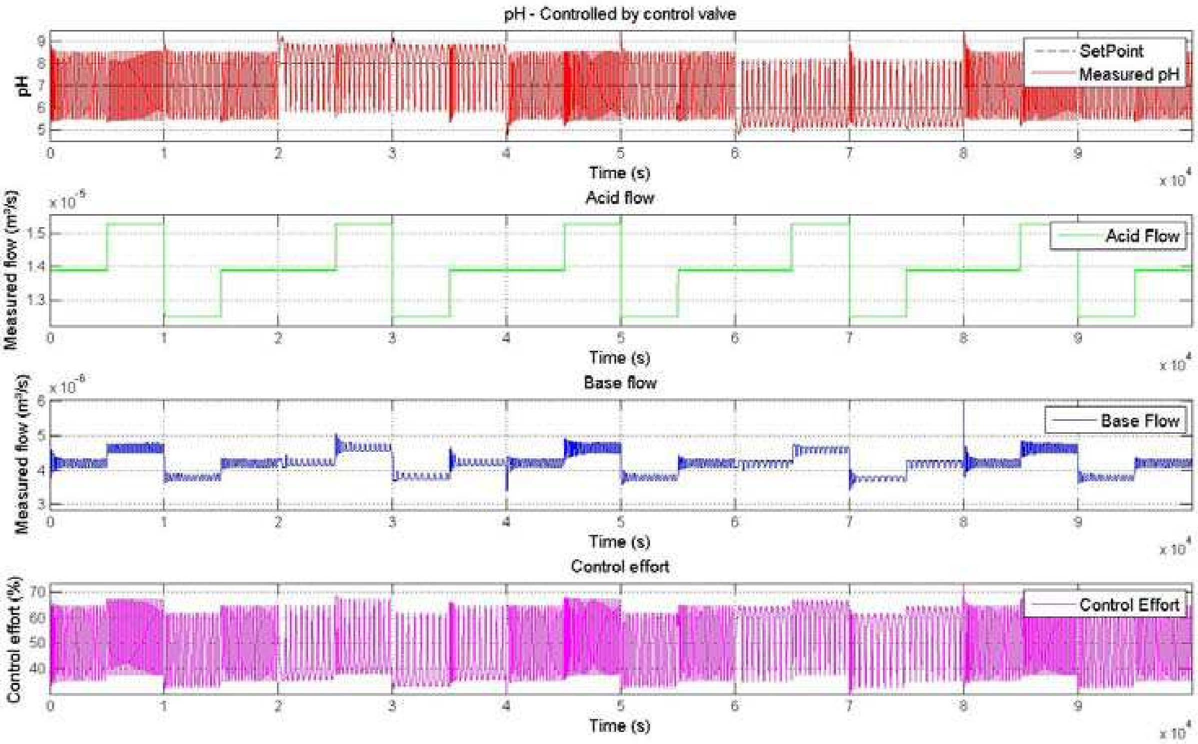

Fig. 27. pH controller response and control effort with control valve enabled.

The effect of the variability in the control loop is observed in greater detail in Figure 28.

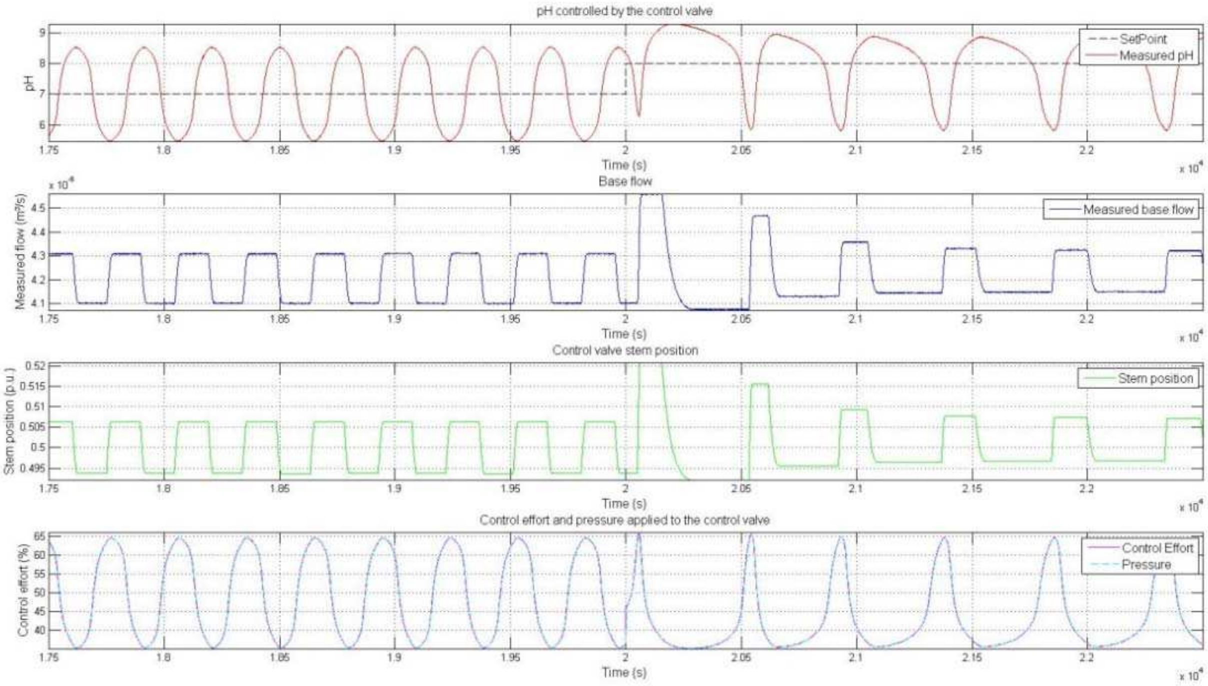

Fig. 28. Control variability due to valve friction.

The pH controller output is shown in the lower plot in Figure 28 together with the pressure applied to the valve by the I/P converter. Since this converter is linear and has a fast response, the pressure virtually follows the controller output.

Despite the efficient operation of the I/P converter, the base flow, shown in the second plot in Figure 28, does not follow the controller output, becoming constant on certain intervals, what causes the variability in the control loop. This effect can be better understood observing the valve stem position, shown in the third plot in this figure. The base flow follows the stem position, since the valve opening is proportional to it, considering that the stem movements are very small, around 1%. The problem of variability arises because the stem position does not strictly follow the changes in the pressure applied to the valve. This happens because of the friction in the valve, which prevents the stem from moving for low changes in pressure, causing the movement seen in the plots and the variability observed in the pH response.

6. Matlab code

Here are presented the Matlab .m files developed to configure and simulate the presented model, and to save and present the results of the simulations.

Seven Matlab files were created, where one of them is the main script that runs all the other files, including the Simulink model. Another file is a script to load the model parameters to the Workspace. Finally, there are five function files that generate graphics with the simulations results. Each of them will be presented and briefly commented, since all the code is documented in detail.

6.1 Simulating pH and level model

Here is presented the main Matlab script (*simulatin_pH_and_level_model.m*). It initializes the Workspace, loads the model parameters, simulates the model and saves and presents the results.

The main point in writing a script instead of manually simulating the model is that it facilitates the model use, making it easier to run several simulations and manage the resulting data.

```
%% Simulation of the pH neutralization plant
% Batch for the simulation of the model
% Authors: Claudio Garcia and Rodrigo Juliani
% -------------------------------------------------------------------------------------------------------------------------------
%% Initialization
% Matlab Workspace is initialized
matlabrc;                                          % Clears all variables in the workspace and reset Matlab
close all;                                         % Closes all figures
clc;                                               % Clears the command window
tic;                                               % Starts Matlab timer
scrsz = get(0,'ScreenSize');                       % Gets the size of the screen
fig_size = [1 1 scrsz(3) scrsz(4)];                % Sets a default size for figures to fit the screen
% -------------------------------------------------------------------------------------------------------------------------------
%% Model Configuration
Configuring_pH_and_level_model                     % Model parameters are loaded to the Workspace
% -------------------------------------------------------------------------------------------------------------------------------
%% Simulations Configuration
% Parameters of the simulation
```

```
delta_sim = 0.05;                                                % [s] Fixed step size of the simulation
Ts = 0.5;                                                        % [s] Sampling time of the digital controllers
Decim = Ts/delta_sim;                                            % [adim.] Decimation of the recorded variables
Tsim = 500000;                                                   % [s] Simulation interval
% Setpoints and disturbance for the simulation
delta_h = 15;                                                    % [%] Step size for the level controller
exc_h = [h_nom h_nom+delta_h h_nom h_nom-delta_h].';             % [%] Vector of setpoins for the level controller
T_h = Tsim/5;                                                    % [s] Step times for the level controller setpoints
delta_pH = 1;                                                    % [pH] Step size for the pH controller
exc_pH = [pH_nom pH_nom+delta_pH pH_nom pH_nom-delta_pH pH_nom].';% [pH] Vector of setpoints for the pH controller
T_pH = T_h/5;                                                    % [s] Step times for the pH controller setpoints
delta_Qa = Qa_nom*0.1;                                           % [m³/s] Step size for the acid flow disturbance
exc_Qa = [Qa_nom Qa_nom+delta_Qa Qa_nom-delta_Qa Qa_nom].';      % [m³/s] Vector of acid feed flow
T_Qa = T_pH/4;                                                   % [s] Step times for the acid flow disturbances
% Control mode
Auto_man = 2;                                                    % pH controller: Manual =1; Automatic = 2;
% ------------------------------------------------------------------------------------------------------------------------
%% Simulations
% Process response with pump enabled
Valve_Pump = 2;                                                  % pH actuator: Valve = 1; Pump = 2;
sim ('Model_pH_and_level');                                      % Simulates the model
 % Simulation results are copied to unique variables
h_pump = h_m;                                                    % Measured level
pH_pump = pH_m;                                                  % Measured pH
Qa_pump = Qa_m;                                                  % Measured acid flow
Qb_pump = Qb_m;                                                  % Measured base flow
MV_h_pump = MV_h;                                                % Level control effort
MV_pH_pump = MV_pH;                                              % pH control effort
% Process response with valve enabled
Valve_Pump = 1;                                                  % pH actuator: Valve = 1; Pump = 2;
sim ('Model_pH_and_level');                                      % Simulates the model
% Simulation results are copied to unique variables
h_valve = h_m;                                                   % Measured level
pH_valve = pH_m;                                                 % Measured pH
Qa_valve = Qa_m;                                                 % Measured acid flow
Qb_valve = Qb_m;                                                 % Measured base flow
MV_h_valve = MV_h;                                               % Level control effort
MV_pH_valve = MV_pH;                                             % pH control effort
x_valve = x;                                                     % Valve stem position
P_valve = P;                                                     % Pressure applied to the control valve
% Saves simulated data
save Simulated_Data;                                             % Saves the simulated data in a .mat file
% ------------------------------------------------------------------------------------------------------------------------
%% Results
% Complete results
plot_all_data(t, SP_pH, pH_pump, SP_h, h_pump, Qa_pump, Tsim, h_nom,...   % Plots all the simulation data with the
   delta_h, Qa_nom, delta_Qa, 'dosing pump', fig_size);                   % dosing pump enabled
print -djpeg All_data_pump;                                      % Saves the figure as a .jpg file
plot_all_data(t, SP_pH, pH_valve, SP_h, h_valve, Qa_valve, Tsim, h_nom,... % Plots all the simulation data with the
   delta_h, Qa_nom, delta_Qa, 'control valve', fig_size);                 % control valve enabled
print -djpeg All_data_valve;                                     % Saves the figure as a .jpg file
% Level responses
plot_level_response(t, SP_h, h_pump, h_valve, h_nom, delta_h, Tsim, fig_size);
print -djpeg Level_response;                                     % Saves the figure as a .jpg file
% Level influense analysis
plot_level_influence(t, pH_pump, pH_valve, fig_size);
print -djpeg Level_influence;                                    % Saves the figure as a .jpg file
% pH responses for nominal level
```

```
plot_pH_response(t(1:length(t)/5), SP_pH(1:length(t)/5), pH_pump(1:length(t)/5), Qa_pump(1:length(t)/5), ...
   Qa_nom, delta_Qa, Qb_pump(1:length(t)/5), MV_pH_pump(1:length(t)/5), 'dosing pump', fig_size);
print -djpeg pH_response_pump;                                          % Saves the figure as a .jpg file
plot_pH_response(t(1:length(t)/5), SP_pH(1:length(t)/5), pH_valve(1:length(t)/5), Qa_valve(1:length(t)/5), ...
  Qa_nom, delta_Qa,Qb_valve(1:length(t)/5), MV_pH_valve(1:length(t)/5), 'control valve', fig_size);
print -djpeg pH_response_valve;                                         % Saves the figure as a .jpg file
% Control valve friction analysis
plot_valve_friction_analysis(t(length(t)/5*0.175:length(t)/5*0.225), SP_pH(length(t)/5*0.175:length(t)/5*0.225),...
   pH_valve(length(t)/5*0.175:length(t)/5*0.225), Qb_valve(length(t)/5*0.175:length(t)/5*0.225), ...
   MV_pH_valve(length(t)/5*0.175:length(t)/5*0.225), x(length(t)/5*0.175:length(t)/5*0.225), ...
   P(length(t)/5*0.175:length(t)/5*0.225), fig_size);
print -djpeg valve_friction_analysis;                                   % Saves the figure as a .jpg file
% ---------------------------------------------------------------------------------------------------------------------------------------------
%% Execution Time
Execution_time = toc;                                                   % Registers the current value in Matlab timer
Execution_time_minutes = floor(Execution_time/60);                      % Calculates the number of minutes elapsed
Execution_time_seconds = Execution_time-Execution_time_minutes*60;      % Calculates the number of seconds elapsed
Elapsed_time = [num2str(Execution_time_minutes) ' minutes ' ...
   num2str(Execution_time_seconds) ' seconds'];                         % Creates a string with elapsed time
msgbox(Elapsed_time, 'Elapsed Time');                                   % Shows a message box with the elapsed time
```

6.2 Configuring pH and level model

This file (*Configuring_pH_and_level_model.m*) loads all the model parameters to the Workspace.

Although it is easier to set all model parameters directly in the Simulink model, generating it with literal parameters and creating a Matlab script with the parameter values has some advantages, such as making it simpler to locate any parameter, to change its value or to make simulations with different values for certain parameters, as was done for the pH loop actuator.

```
%% Configuring pH and level model
% Data for the simulation of the pH neutralization plant
% Authors: Claudio Garcia and Rodrigo Juliani
% ---------------------------------------------------------------------------------------------------------------------------------------------
%% Model parameters are loaded to the Workspace
% Parameters of the reactants
c_NaOH = 0.0185;                                                        % [kmol/m³] Molar concentration of the base
c_HCl = 0.0056;                                                         % [kmol/m³] Molar concentration of the acid
Kw = 10^-14;                                                            % [adim.] Dissociation constant of water
% ---------------------------------------------------------------------------------------------------------------------------------------------
% Parameters of the reaction tank (CSTR)
D_tank = 0.4;                                                           % [m] Reaction tank (CSTR) diameter
D_pHmeter = 0.03;                                                       % [m] pHmeter diameter
D_cond_ext = 0.035;                                                     % [m] Instrument external diameter
D_cond_int = 0.030;                                                     % [m] Instrument internal diameter
D_cond_inst = 0.01;                                                     % [m] Instrument diameter
D_resist = 0.01;                                                        % [m] Equipment diameter
D_agit = 0.01;                                                          % [m] Agitator diameter
D_pH_cond = 0.05;                                                       % [m] Instrument diameter
A_tank = pi*D_tank^2/4;                                                 % [m²] Area of the base of the CSTR
A_inst = pi*(D_pHmeter^2+D_cond_ext^2-D_cond_int^2+D_cond_inst+2*D_resist^2+D_agit^2+D_pH_cond^2)/4;
                                                                        % [m²] Area of the instrumentation in the CSTR
A_CSTR = A_tank - A_inst;                                               % [m²] Effective area of the CSTR
h_max_CSTR = 100;                                                       % [%] Maximum CSTR level, in percentage
h_max_CSTR_m = 0.5;                                                     % [m] Maximum CSTR level, in meters
```

```
h_nom = 65;                                          % [%] Nominal CSTR level
h0_CSTR = h_nom;                                     % [%] Initial CSTR level
V_max_CSTR = A_CSTR*h_max_CSTR_m;                    % [m³] Effective maximum CSTR volume
V0_CSTR = V_max_CSTR*h0_CSTR/100;                    % [m³] Initial CSTR volume
% Parameters of the pressurized base tank
D_press = 0.65;                                      % [m] Pressurized base tank diameter
A_press = pi*D_press^2/4;                            % [m²] Base area of the pressurized base tank
H_press_max = 1.05;                                  % [m] Maximum height of the base tank
rho = 998.21;                                        % [kg/m3] Mass density of the base at 20oC
Patm = 9.247e5;                                      % [Pa] Local atmospheric pressure
P_press = 2.13e4;                                    % [Pa] Gauge internal pressure of the pressurized tank
% Parameters of the I/P converter
K_ip = 1;                                            % [adim.] Gain of the I/P converter
tau_ip = 1;                                          % [s] Time constant of the I/P converter
% Parameters of the pHmeter
tau_meter_pH = 10;                                   % [s] Time constant of the pH meter
K_meter_pH = 1;                                      % [pH/pH] Gain of the pH meter
[A,B,C,D] = tf2ss(K_meter_pH,[tau_meter_pH 1]);      % Estimates the state-space representation of the
pHmeter
var_noise_pH = 1E-5;                                 % [adim.] Variance of noise in the pHmeter
seed_noise_pH = 67890;                               % [adim.] Seed of the random number generator
% Parameters of the level meter
K_meter_level = 1;                                   % [%/%] Gain of the level meter
tau_meter_level = 0.5;                               % [s] Time constant of the level meter
T_noise_level = -1;                                  % [s] Period of noise in the level meter
var_noise_level = 1E-4;                              % [adim.] Variance of noise in the level meter
seed_noise_level = 12345;                            % [adim.] Seed of the random number generator
% Parameters of the flow meters
K_meter_flow = 1;                                    % [(m³/s)/(m³/s)] Gain of the flow meters
tau_meter_flow = 3;                                  % [s] Time constant of the flow meters
var_noise_Qa  = 1E-17;                               % [adim.] Variance of the acid flow meter
seed_noise_Qa = 357;                                 % [adim.] Seed fot the random number generator
var_noise_Qb  = 1E-18;                               % [adim.] Variance of the base flow meter
seed_noise_Qb = 159;                                 % [adim.] Seed fot the random number generator
% Parameters of the pump
K_pump = 30.3/1000/3600/100;                         % [(m³/s)/%] Gain of the pump
% Parameters of the solenoid valve
Kv = 9.238E-5;                                       % [m³/s/sqrt(m)] Flow coefficient of the valve
% Parameters of the control valve of the pressurized base tank
g = 9.80665;                                         % [m/s²] Standard gravity
Cv = 0.22;                                           % [sqrt(psig)/gpm] Flow coefficient of the valve
deltaP0 = P_press+rho*g*H_press_max;                 % [Pa] Maximum differential pressure in the valve
K_aux1 = 2.40153e-5*Cv/sqrt(rho);                    % Auxiliar parameter
% Parameters of the control valve actuator
K_at = 0.01;                                         % [p.u./%] Gain of the valve actuator
tau_at = 0.8;                                        % [s] Time constant of the valve actuator
% Nomimal variable values of the pH plant
Qa_nom = 13.8889e-6;                                 % [m³/s] Nominal flow of acid
Qb_nom = 4.204e-6;                                   % [m³/s] Nominal flow of base
pH_nom = 7;                                          % [pH] Nominal value of pH
X_nom = 0;                                           % [kmol/m³] Nominal value of the
                                                     % ionic concentration difference

% Inicialization of the plant variables
mv0 = 50;                                            % [%] Initial output of the pH controller
pH_m0 = pH_nom*K_meter_pH;                           % [%] Initial output of the pH meter
mv_p0 = K_ip*mv0;                                    % [%] Initial output of the I/P converter
x0 = mv0*K_at;                                       % [p.u.] Initial opening of the control valve
% Inicialization of the Kano friction model
```

```
J = 0.0;                                        % [p.u.] Value of the friction parameter J
S = 0.28;                                       % [p.u.] Value of the friction parameter S
d0 = 1;                                         % [adim.] Initial value of the friction direction
                                                % d=1: positive friction; d=-1: negative friction
u0 = x0;                                        % [p.u.] Initial input signal
us0 = u0;                                       % [p.u.] Initial value for wich the input moved before
% the stem stopped
u_10 = u0;                                      % [p.u.] Initial value for the input signal at t=-1
stp0 = 1;                                       % [adim.] Initial value for the stem movement condition
                                                % (stp=0: moving stem; stp=1: stopped tem)
% Tuning parameters of the pH controller (PI)
Kn = 100/14;                                    % [%/pH] Adimensionalization gain
Kc = 1.34;                                      % [adim.] Gain of the PI controller
Ti = 620;                                       % [adim.] Ti parameter of the PI controller
% Tuning parameters of the level controller (PI)
P_l = 11.65;                                    % [adim.] Proportional gain of the PI controller
I_l = 271.52;                                   % [adim.] Integrative gain parameter of the PI controller
T_PWM = 10;                                     % [s] Period of the PWM in the level controller
```

6.3 Plot all data

This function (*plot_all_data.m*) plots the controlled variables, pH and level, and the disturbance for both loops, acid flow.

Since it is a Matlab function, it can be used more than once, avoiding repetition of code and making changes to the plots easier to perform.

```
%% Plot all data
% Plots all the simulation data for pH, level and acid flow
% Authors: Claudio Garcia and Rodrigo Juliani
% ------------------------------------------------------------------------------------------------------------------------------------
function [] = plot_all_data(t, SP_pH, pH, SP_h, h, Qa, Tsim, h_nom, delta_h, Qa_nom, delta_Qa, actuator, fig_size)
figure_complete = figure('OuterPosition',fig_size);      % Generates a figure with specified size and location
subplot(3,1,1);                                          % Creates a plot in the upper part of the figure
stairs(t,SP_pH,'k--');                                   % Plots the setpoint for the pH controller
hold on;                                                 % Holds the plot so that other plots can be added to it
plot(t,pH);                                              % Plots measured values of the pH
axis ([0 Tsim 4.5 9.5]);                                 % Sets the plot axis to desired dimensions
grid;                                                    % Inserts a grid on the plot
title (['pH - Controlled by ' actuator], 'FontSize', 12); % Inserts a title in the plot
xlabel ('Time (s)', 'FontSize', 12);                     % Inserts a label in the horizontal axis
ylabel ('pH', 'FontSize', 12);                           % Inserts a label in the vertical axis
leg = legend ('SetPoint','Measured pH');                 % Inserts a legend in the plot
set (leg, 'FontSize', 12);                               % Sets the font size in the legend
subplot(3,1,2);                                          % Creates a plot in the middle part of the figure
stairs(t,SP_h,'k--');                                    % Plots the setpoints for the level controller
hold on;                                                 % Holds the plot for another plot can be added
plot(t,h);                                               % Plots measured values of the level
axis ([0 Tsim h_nom-delta_h-1 h_nom+delta_h+1]);         % Sets the plot axis so that data fits the plot area
grid;                                                    % Inserts a grid on the plot
title ('Level', 'FontSize', 12);                         % Insert a title in the plot
xlabel ('Time (s)', 'FontSize', 12);                     % Inserts a label in the x axis
ylabel ('Level (%)', 'FontSize', 12);                    % Inserts a label in the vertical axis
leg = legend ('SetPoint','Measured Level');              % Inserts a legend in the plot
set (leg, 'FontSize', 12);                               % Sets the font size in the legend
subplot(3,1,3);                                          % Creates a plot in the lower part of the figure
plot(t,Qa);                                              % Plots measured values of acid flow
axis ([0 Tsim Qa_nom-delta_Qa*1.2 Qa_nom+delta_Qa*1.2]); % Sets the plot axis so that data fits the plot area
```

```
grid;                                                          % Inserts a grid on the plot
title ('Acid flow', 'FontSize', 12);                           % Insert a title in the plot
xlabel ('Time (s)', 'FontSize', 12);                           % Inserts a label in the x axis
ylabel ('Measured flow (m³/s)', 'FontSize', 12);               % Inserts a label in the vertical axis
leg = legend ('Acid Flow');                                    % Inserts a legend in the plot
set (leg, 'FontSize', 12);                                     % Sets the font size in the legend
```

6.4 Plot level response

This function (*plot_level_response.m*) generates graphs of the level with both actuators enabled, dosing pump and control valve, for comparison. It also generates plots zoomed on the steps applied to the level controller set point.

```
%% Plot level controller response
% Plots the level controller response for different conditions
% Authors: Claudio Garcia and Rodrigo Juliani
% ------------------------------------------------------------------------------------------------------------------------------------------
function [] = plot_level_response(t, SP_h, h_pump, h_valve, h_nom, delta_h, Tsim, fig_size)
figure_complete = figure('OuterPosition',fig_size);            % Generates a figure with specified size and location
%% Level response comparison
subplot(3,1,1);                                                % Creates a plot in the upper part of the figure
stairs(t,SP_h,'k--');                                          % Plots the setpoints for the level controller
hold on;                                                       % Holds the plot for another plot to be added
plot(t,h_pump, 'g', t, h_valve, 'r:');                         % Plots measured values of the level
axis ([0 Tsim h_nom-delta_h-1 h_nom+delta_h+1]);               % Sets the plot axis with desired dimensions
grid;                                                          % Adds a grid to the plot
title ('Level', 'FontSize', 12);                               % Inserts a title in the plot
xlabel ('Time (s)', 'FontSize', 12);                           % Inserts a label in the x axis
ylabel ('Level (%)', 'FontSize', 12);                          % Inserts a label in the vertical axis
leg = legend ('SetPoint','Measured level with pump enabled',
   'Measured level with valve neabled');                       % Inserts a legend in the plot
set (leg, 'FontSize', 12);                                     % Sets the font size in the legend
%% Parameters for the step response plots
slice = length(h_pump)/5;                                      % Determines reference points for the steps
relative_t = t(1:slice*0.008+1);                               % Creates a time vector for the step response plots
axis_values = [0 800 h_nom-delta_h-1 h_nom+delta_h+1];         % Creates a parameter for the following plots axis
%% Pump
subplot(3,4,5);                                                % Creates a plot for the first step
stairs(relative_t,SP_h(slice*0.999:slice*1.007),'k--');        % Plots the setpoint for the level controller
hold on;                                                       % Holds the plot for another plot to be added
plot(relative_t,h_pump(slice*0.999:slice*1.007), 'g');         % Plots measured values of the level
axis (axis_values);                                            % Sets the plot axis to desired dimensions
grid;                                                          % Adds a grid to the plot
title ('Level - 1st step', 'FontSize', 12);                    % Inserts a title in the plot
xlabel ('Relative time (s)', 'FontSize', 12);                  % Inserts a label in the horizontal axis
ylabel ('Level (%) - Dosing pump', 'FontSize', 12);            % Inserts a label in the vertical axis
subplot(3,4,6);                                                % Creates a plot for the second step
stairs(relative_t,SP_h(slice*1.999:slice*2.007),'k--');        % Plots the setpoint for the level controller
hold on;                                                       % Holds the plot for another plot to be added
plot(relative_t,h_pump(slice*1.999:slice*2.007), 'g');         % Plots measured values of the level
axis (axis_values);                                            % Sets the plot axis to desired dimensions
grid;                                                          % Adds a grid to the plot
title ('Level - 2nd step', 'FontSize', 12);                    % Inserts a title in the plot
xlabel ('Relative time (s)', 'FontSize', 12);                  % Inserts a label in the horizontal axis
subplot(3,4,7);                                                % Creates a plot for the third step
stairs(relative_t,SP_h(slice*2.999:slice*3.007),'k--');        % Plots the setpoint for the level controller
hold on;                                                       % Holds the plot for another plot to be added
plot(relative_t,h_pump(slice*2.999:slice*3.007), 'g');         % Plots measured values of the level
```

```
axis (axis_values);                                          % Sets the plot axis to desired dimensions
grid;                                                        % Adds a grid to the plot
title ('Level - 3rd step', 'FontSize', 12);                  % Inserts a title in the plot
xlabel ('Relative time (s)', 'FontSize', 12);                % Inserts a label in the horizontal axis
subplot(3,4,8);                                              % Creates a plot for the fourth step
stairs(relative_t,SP_h(slice*3.999:slice*4.007),'k--');      % Plots the setpoint for the level controller
hold on;                                                     % Holds the plot for another plot to be added
plot(relative_t,h_pump(slice*3.999:slice*4.007), 'g');       % Plots measured values of the level
axis (axis_values);                                          % Sets the plot axis to desired dimensions
grid;                                                        % Adds a grid to the plot
title ('Level - 4th step', 'FontSize', 12);                  % Inserts a title in the plot
xlabel ('Relative time (s)', 'FontSize', 12);                % Inserts a label in the horizontal axis
%% Control Valve
subplot(3,4,9);                                              % Creates a plot for the first step
stairs(relative_t,SP_h(slice*0.999:slice*1.007),'k--');      % Plots the setpoint for the level controller
hold on;                                                     % Holds the plot for another plot to be added
plot(relative_t,h_valve(slice*0.999:slice*1.007), 'r');      % Plots measured values of the level
axis (axis_values);                                          % Sets the plot axis to desired dimensions
grid;                                                        % Adds a grid to the plot
xlabel ('Relative time (s)', 'FontSize', 12);                % Inserts a label in the horizontal axis
ylabel ('Level (%) - Control valve', 'FontSize', 12);        % Inserts a label in the vertical axis
subplot(3,4,10);                                             % Creates a plot for the second step
stairs(relative_t,SP_h(slice*1.999:slice*2.007),'k--');      % Plots the setpoint for the level controller
hold on;                                                     % Holds the plot for another plot to be added
plot(relative_t,h_valve(slice*1.999:slice*2.007), 'r');      % Plots measured values of the level
axis (axis_values);                                          % Sets the plot axis to desired dimensions
grid;                                                        % Adds a grid to the plot
xlabel ('Relative time (s)', 'FontSize', 12);                % Inserts a label in the horizontal axis
subplot(3,4,11);                                             % Creates a plot for the third step
stairs(relative_t,SP_h(slice*2.999:slice*3.007),'k--');      % Plots the setpoint for the level controller
hold on;                                                     % Holds the plot for another plot to be added
plot(relative_t,h_valve(slice*2.999:slice*3.007), 'r');      % Plots measured values of the level
axis (axis_values);                                          % Sets the plot axis to desired dimensions
grid;                                                        % Adds a grid to the plot
xlabel ('Relative time (s)', 'FontSize', 12);                % Inserts a label in the horizontal axis
subplot(3,4,12);                                             % Creates a plot for the fourth step
stairs(relative_t,SP_h(slice*3.999:slice*4.007),'k--');      % Plots the setpoint for the level controller
hold on;                                                     % Holds the plot for another plot to be added
plot(relative_t,h_valve(slice*3.999:slice*4.007), 'r');      % Plots measured values of the level
axis (axis_values);                                          % Sets the plot axis to desired dimensions
grid;                                                        % Adds a grid to the plot
xlabel ('Relative time (s)', 'FontSize', 12);                % Inserts a label in the horizontal axis
```

6.5 Plot level influence

This funciont (*plot_level_influence.m*) plots the pH response with different levels for both actuators.

```
%% Plot level influence analysis
% Plots the pH response with different levels in the CSTR
% Authors: Claudio Garcia and Rodrigo Juliani
% ----------------------------------------------------------------------------------------------------------------
function [] = plot_level_influence(t, pH_pump, pH_valve, fig_size)
figure_level_influence = figure('OuterPosition',fig_size);   % Generates a figure with specified size and location
slice = length(pH_pump)/5;                                   % Determines a slice to separate data with constant level
subplot(2,1,1);                                              % Creates a plot in the upper half of the figure
hold on;                                                     % Holds the plot so that other plots can be added to it
plot(t((1:slice*1+1)),pH_pump(slice*1:slice*2),'r:');        % Data for high level
plot(t((1:slice*1+1)),pH_pump(slice*2:slice*3),'k-.');                   % Data for nominal level
plot(t((1:slice*1+1)),pH_pump(slice*3:slice*4),'b--');                   % Data for low level
axis 'tight'                                                              % Sets the plot axis to desired dimensions
grid;                                                                     % Inserts a grid on the plot
xlabel ('Relative Time (s)', 'FontSize', 12);                             % Inserts a label in the horizontal axis
ylabel ('Measured pH', 'FontSize', 12);                                   % Inserts a label in the vertical axis
title ('pH response with different levels and pump enabled', 'FontSize', 12);  % Inserts a title in the plot
leg = legend ('Higher level','Nominal level', 'Lower level');             % Inserts a legend in the plot
set (leg, 'FontSize', 12);                                                % Sets the font size in the legend
subplot(2,2,3);                                                           % Creates a plot in the left part lower half of the figure
hold on;                                                                  % Holds the plot so that other plots can be added to it
plot(t((1:slice*1+1)),pH_valve(slice*1:slice*2),'r:');                    % Data for high level
plot(t((1:slice*1+1)),pH_valve(slice*2:slice*3),'k-.');                   % Data for nominal level
plot(t((1:slice*1+1)),pH_valve(slice*3:slice*4),'b--');                   % Data for low level
axis 'tight'                                                              % Sets the plot axis to desired dimensions
grid;                                                                     % Inserts a grid on the plot
xlabel ('Relative Time (s)', 'FontSize', 12);                             % Inserts a label in the horizontal axis
ylabel ('Measured pH', 'FontSize', 12);                                   % Inserts a label in the vertical axis
title ('pH response with different levels and control valve enabled', 'FontSize', 12);% Inserts a title in the plot
leg = legend ('Higher level','Nominal level', 'Lower level');             % Inserts a legend in the plot
set (leg, 'FontSize', 12);                                                % Sets the font size in the legend
subplot(2,2,4);                                                           % Creates a plot in the lower half of the figure
hold on;                                                                  % Holds the plot so that other plots can be added to it
plot(t((slice*.25:slice*.27)),pH_valve(slice*1.25:slice*1.27),'r','LineWidth',1.5);   % Data for high level
plot(t((slice*.25:slice*.27)),pH_valve(slice*2.25:slice*2.27),'k','LineWidth',1.5);   % Data for nominal level
plot(t((slice*.25:slice*.27)),pH_valve(slice*3.25:slice*3.27),'b','LineWidth',1.5);   % Data for low level
axis 'tight'                                                              % Sets the plot axis to desired dimensions
grid;                                                                     % Inserts a grid on the plot
xlabel ('Relative Time (s)', 'FontSize', 12);                             % Inserts a label in the horizontal axis
ylabel ('Measured pH', 'FontSize', 12);                                   % Inserts a label in the vertical axis
title ('pH response with different levels and control valve enabled - Zoom', 'FontSize', 12);
                                                                          % Inserts a title in the plot
leg = legend ('Higher level','Nominal level', 'Lower level');             % Inserts a legend in the plot
set (leg, 'FontSize', 12);                                                % Sets the font size in the legend
```

6.6 Plot pH response

This function (*plot_pH_response.m*) plots the pH response and respective control effort with constant level.

```
%% Plot pH response
% Plots the pH step and disturbance responses
% Authors: Claudio Garcia and Rodrigo Juliani
% -----------------------------------------------------------------------------------------------------------------------------------------------
function [] = plot_pH_response(t, SP_pH, pH, Qa, Qa_nom, delta_Qa, Qb, MV_pH, actuator, fig_size)
figure_complete = figure('OuterPosition',fig_size);                       % Generates a figure with specified size and location
subplot(4,1,1);                                                           % Creates a plot in the upper part of the figure
stairs(t,SP_pH,'k--');                                                    % Plots the setpoint for the pH controller
hold on;                                                                  % Holds the plot so that other plots can be added to it
plot(t,pH,'r');                                                           % Plots measured values of the pH in red
axis ([0 t(length(t)) 4.5 9.5]);                                          % Sets the plot axis to desired dimensions
grid;                                                                     % Inserts a grid on the plot
title (['pH - Controlled by ' actuator], 'FontSize', 12);                 % Inserts a title in the plot
xlabel ('Time (s)', 'FontSize', 12);                                      % Inserts a label in the horizontal axis
ylabel ('pH', 'FontSize', 12);                                            % Inserts a label in the vertical axis
leg = legend ('SetPoint','Measured pH');                                  % Inserts a legend in the plot
set (leg, 'FontSize', 12);                                                % Sets the font size in the legend
subplot(4,1,2);                                                           % Creates a plot for the acid flow
```

```
plot(t,Qa,'g');                                                   % Plots measured values of acid flow
axis ([0 t(length(t)) Qa_nom-delta_Qa*1.2 Qa_nom+delta_Qa*1.2]);  % Sets the plot axis to desired dimensions
grid;                                                             % Inserts a grid on the plot
title ('Acid flow', 'FontSize', 12);                              % Insert a title in the plot
xlabel ('Time (s)', 'FontSize', 12);                              % Inserts a label in the x axis
ylabel ('Measured flow (m³/s)', 'FontSize', 12);                  % Inserts a label in the vertical axis
leg = legend ('Acid Flow');                                       % Inserts a legend in the plot
set (leg, 'FontSize', 12);                                        % Sets the font size in the legend
subplot(4,1,3);                                                   % Creates a plot for the base flow
plot(t,Qb,'b');                                                   % Plots measured values of base flow
axis 'tight';                                                     % Sets the plot axis so that data fits the plot area
grid;                                                             % Inserts a grid on the plot
title ('Base flow', 'FontSize', 12);                              % Insert a title in the plot
xlabel ('Time (s)', 'FontSize', 12);                              % Inserts a label in the x axis
ylabel ('Measured flow (m³/s)', 'FontSize', 12);                  % Inserts a label in the vertical axis
leg = legend ('Base Flow');                                       % Inserts a legend in the plot
set (leg, 'FontSize', 12);                                        % Sets the font size in the legend
subplot(4,1,4);                                                   % Creates a plot in the lower part of the figure
plot(t,MV_pH,'m');                                                % Plots the control efforts
axis 'tight';                                                     % Sets the plot axis so that data fits the plot area
grid;                                                             % Inserts a grid on the plot
title ('Base flow', 'FontSize', 12);                              % Insert a title in the plot
xlabel ('Time (s)', 'FontSize', 12);                              % Inserts a label in the x axis
ylabel ('Control effort (%)', 'FontSize', 12);                    % Inserts a label in the vertical axis
leg = legend ('Control Effort');                                  % Inserts a legend in the plot
set (leg, 'FontSize', 12);                                        % Sets the font size in the legend
```

6.7 Plot valve friction analysis

This function (*plot_valve_friction_analyis.m*) plots the pH with the control valve enabled and respective base flow, valve stem position, pressure applied to the valve actuator and control effort.

```
%% Plot valve friction analysis
%  Generates plots for a valve friction analysis
% Authors: Claudio Garcia and Rodrigo Juliani
% ----------------------------------------------------------------------------------------------------------------------------
function [] = plot_valve_friction_analysis(t, SP_pH, pH, Qb, MV_pH, x, P, fig_size)
figure_complete = figure('OuterPosition',fig_size);               % Generates a figure with specified size and location
subplot(4,1,1);                                                   % Creates a plot in the upper part of the figure
stairs(t,SP_pH,'k--');                                            % Plots the setpoint for the pH controller
hold on;                                                          % Holds the plot so that other plots can be added to it
plot(t,pH,'r');                                                   % Plots measured values of the pH in red
axis 'tight';                                                     % Sets the plot axis so that all data can be seen
grid;                                                             % Inserts a grid on the plot
title ('pH controlled by the control valve', 'FontSize', 12);     % Inserts a title in the plot
xlabel ('Time (s)', 'FontSize', 12);                              % Inserts a label in the horizontal axis
ylabel ('pH', 'FontSize', 12);                                    % Inserts a label in the vertical axis
leg = legend ('SetPoint','Measured pH');                          % Inserts a legend in the plot
set (leg, 'FontSize', 12);                                        % Sets the font size in the legend
subplot(4,1,2);                                                   % Creates a plot for the base flow
plot(t,Qb,'b');                                                   % Plots measured values of base flow
axis 'tight';                                                     % Sets the plot axis so that data fits the plot area
grid;                                                             % Inserts a grid on the plot
title ('Base flow', 'FontSize', 12);                              % Insert a title in the plot
xlabel ('Time (s)', 'FontSize', 12);                              % Inserts a label in the x axis
ylabel ('Measured flow (m³/s)', 'FontSize', 12);                  % Inserts a label in the vertical axis
leg = legend ('Measured base flow');                              % Inserts a legend in the plot
```

```
set (leg, 'FontSize', 12);                                                    % Sets the font size in the legend
subplot(4,1,3);                                                               % Creates a plot for the stem position
plot(t,x,'g');                                                                % Plots the valve stem position
axis 'tight';                                                                 % Sets the plot axis so that data fits the plot area
grid;                                                                         % Inserts a grid on the plot
title ('Control valve stem position', 'FontSize', 12);                        % Insert a title in the plot
xlabel ('Time (s)', 'FontSize', 12);                                          % Inserts a label in the x axis
ylabel ('Stem position (p.u.)', 'FontSize', 12);                              % Inserts a label in the vertical axis
leg = legend ('Stem position');                                               % Inserts a legend in the plot
set (leg, 'FontSize', 12);                                                    % Sets the font size in the legend
subplot(4,1,4);                                                               % Creates a plot in for the control effort
plot(t,MV_pH,'m',t,P*100,'c--');                                              % Plots the control efforts and pressure applied to
                                                                              % the control valve
axis 'tight';                                                                 % Sets the plot axis so that data fits the plot area
grid;                                                                         % Inserts a grid on the plot
title ('Control effort and pressure applied to the control valve', 'FontSize', 12); % Inserts a title in the plot
xlabel ('Time (s)', 'FontSize', 12);                                          % Inserts a label in the x axis
ylabel ('Control effort (%)', 'FontSize', 12);                                % Inserts a label in the vertical axis
leg = legend ('Control Effort', 'Pressure');                                  % Inserts a legend in the plot
set (leg, 'FontSize', 12);                                                    % Sets the font size in the legend
```

7. Conclusion

In this chapter it was demonstrated how to build a Simulink model from a P&ID diagram of a plant and from the mathematical model of each of its components.

It has also been shown how to simulate the model, changing operation conditions and grouping several tests.

To test the built model, simulations were made and the results were analyzed, being similar to the expected from the real plant that was used as basis for the modeling method.

The process of building the model was described with enough detail in order to enable the reproduction of the model by the reader and with sufficient generalization so that it can be used as a guide for creating models for other systems.

Finally, all Simulink diagrams and Matlab codes developed were presented for easy consultation, understanding and reproduction of the model and obtained results.

8. References

ANSI/ISA-5.1-2009 (2009). Instrumentation symbols and identification. *American National Standard*. ISBN 978-1-936007-29-5.

Asuero, A. G. & Michalowski, T. (2011). Comprehensive formulation of titration curves for complex acid-base systems and its analytical implications. *Critical Reviews in Analytical Chemistry*, Vol.41, No.2, (May 2011), pp. 151-187, ISSN 1547-6510.

Garcia, C. (2005). *Modelagem e simulação*. (2nd edition), EDUSP, ISBN 85-314-0904-7 , São Paulo, Brazil. /In Portuguese/

Garcia, C. (2008). Comparison of friction models applied to a control valve. *Control Engineering Practice*, Vol.16, No.10, (October 2008), pp. 1231-1243, ISSN 0967-0661.

Jacobs, O. L. R., Hewkin, M. A. & While, C. (1980). Online computer control of pH in an industrial process. *IEE Proceedings D*, Vol.127, No.4, (July 1980), pp. 161-168, ISSN 0143-7054.

Rinehart, N. & Jury, F. (1997). How control valves impact process optimization. *Hydrocarbon Processing*, Vol.76, No.6, (June 1997), pp.53-58, ISSN 0018-8190.

APPLICATIONS OF MATLAB IN SCIENCE AND ENGINEERING
Edited by Tadeusz Michalowski
INTECH

Applications of MATLAB in Science and Engineering
Edited by Prof. Tadeusz Michalowski

ISBN 978-953-307-708-6
Hard cover, 510 pages
Publisher InTech
Published online 09, September, 2011
Published in print edition September, 2011

The book consists of 24 chapters illustrating a wide range of areas where MATLAB tools are applied. These areas include mathematics, physics, chemistry and chemical engineering, mechanical engineering, biological (molecular biology) and medical sciences, communication and control systems, digital signal, image and video processing, system modeling and simulation. Many interesting problems have been included throughout the book, and its contents will be beneficial for students and professionals in wide areas of interest.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821