Noname manuscript No.
(will be inserted by the editor)

Comparing MIMO process control methods on a pilot plant

E. Boeira · V. Bordignon · D. Eckhard · L. Campestrini

Received: date / Accepted: date

Abstract This work presents a comparison among three different control strategies for multivariable processes. The techniques were implemented in a pilot plant with coupled control loops, where all steps used to design the controllers were described allowing to establish a trade-off between algorithm complexity, information needed from the process and achieved performance. Two data-driven control techniques are used: Multivariable Ultimate Point Method (MUPM) to design a decentralized PID controller and Virtual Reference Feedback Tuning (VRFT) to design a centralized PID controller. A mathematical model of the process is obtained and used to design a model-based Generalized Predictive Controller (GPC). Experimental results allow us to evaluate the performance achieved for each method, as well as to infer on their advantages and disadvantages.

Keywords MIMO control · Ultimate point method · VRFT · System identification · MPC

1 Introduction

Multivariable systems are ubiquitous in Process Control specially in Oil & Gas and Pulp & Paper industries (Skogestad and Postlethwaite 2005; Al-Naumani and Rossiter 2015; Rojas et al 2012; Dumont 1986). Multiple-input multiple-output (MIMO) processes present interactions between control loops, such that operational changes in one sub-system disturb or affect properties of other sub-systems. Different strategies are used to control MIMO processes. When interactions are weak, simple strategies that completely disregard the multivariable nature of the process can be used. However, when coupling is strong, using single-input single-output (SISO) techniques would result in poor performance.

When high performance is expected usually a more complex control structure is used, which demands a larger amount of information from the process (sensors, models and experimental data), uses more complex algorithms (for instance model predictive, adaptive or nonlinear control) and finally requires more time to be designed (García et al 1989; Bodson and Groszkiewicz 1997; Krstic et al 1995). When low performance is accepted, it is usual to use simpler control structures (for instance a decentralized PID controller) that require less information about the process, are easier to implement and tune, resulting in smaller time to obtain an adequate performance (Campestrini et al 2006; Vu and Lee 2010; Jin et al 2013).

Control techniques can be divide into two large groups: data-based or data-driven and model-based. Data-based techniques do not use a mathematical model from the process to design the controller. All information used from the process is collected from experimental data. One example of this class is the classical Ziegler & Nichols technique (Ziegler and Nichols 1942), that uses data from experiments to tune PID controllers. Other examples are *data-driven* techniques developed after the 90's like Iterative Feedback Tuning (Hjalmarsson et al 1998), Virtual

E. Boeira, V. Bordignon, D. Eckhard and L. Campestrini are supported by CNPq.

E. Boeira · V. Bordignon · L. Campestrini
Departamento de Sistemas Elétricos de Automação e Energia
Universidade Federal do Rio Grande do Sul (UFRGS)
E-mail: {emerson.boeira,virginia.bordignon,luciola}@ufrgs.br
D. Eckhard
Departamento de Matemática Pura e Aplicada
Universidade Federal do Rio Grande do Sul (UFRGS)
E-mail: diegoeck@ufrgs.br

Reference Feedback Tuning (Campi et al 2002; Campestrini et al 2016) and Correlation based Tuning (Karimi et al 2004; Yubai et al 2009). These techniques are usually used to design low order controllers (SISO and MIMO), such as PID and lead-lag controllers. On the other hand, model-based methods use a mathematical model of the process to design the controllers and are mostly used with high order controllers like full-order state-feedback with state-observation or model predictive control (MPC) (García et al 1989).

In this work we compare three different MIMO control design methods, each one using different controller structures. Each control method requires different information from the process (model or experimental data), has different complexity in the implementation and results in different closed-loop performance. The objective of this work is to describe all necessary steps to implement the controllers in order to give the reader an overall complexity in using the technique and the limitations of each one. To achieve that, the chosen control methods are used to solve the same problem, which is common in process control: level control of liquids in coupled tanks. This process is present in many industrial applications, is easy to understand and therefore is useful for comparison of control techniques. The chosen controller structures vary from simple decentralized PID, through centralized PID and MIMO MPC and the implemented techniques are Multivariable Ultimate Point Method (Campestrini et al 2009), which is a data-driven technique inspired in the SISO Ziegler-Nichols method, MIMO Virtual Reference Feedback Tuning method (Campestrini et al 2016), that is also a data-driven method which is an extension of VRFT for multivariable systems, and Generalized Predictive Control (Clarke et al 1987) which is a variant of MPC.

The paper is organized as follows. Section 2 presents the Pilot Plant specifications. Section 3 describes the problem formulation, the control objective and the used controller structures. Data-driven control is described in Section 4, which presents the Multivariable Ultimate Point Method and the Virtual Reference Feedback Tuning method description and application to the Pilot Plant. Section 5 describes the Generalized Predictive Control algorithm, which is a model-based technique. A mathematical model of the plant is identified, and this model is used in the design of the controller. The closed-loop response is presented both in simulation and experiments. Section 6 presents a discussion and comparison among all control techniques applied and Section 7 is a brief conclusion.

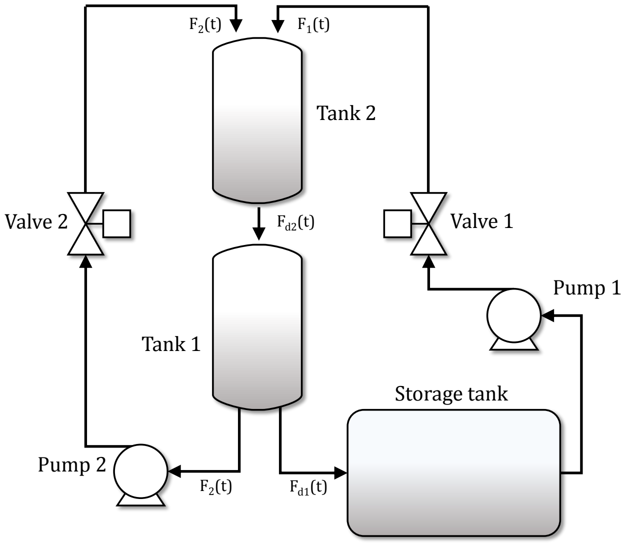

Fig. 1 Schematic diagram of the pilot plant.

2 Pilot plant specifications

The pilot plant is built with off-the-shelf equipments and it possesses typical industrial process' characteristics. In order to provide energy to the fluid (water) and dislocate it through the process, the plant has two centrifugal pumps driven by induction motors of 0.25 kW which are controlled by frequency inverters. Usually this pumps are kept with a fixed frequency. The system also features two globe valves used to regulate the liquid flow rate. These valves are intelligent equipments with internal PID positioners that ensures the opening of the valves are always in the correct position. The plant's tanks present cylindrical geometry with capacity of 70 liters each. Also, to supply water for the experiments, the plant has a 250 liter storage tank. Fig. 1 shows a schematic diagram of the pilot plant.

Communication between the plant's devices is made up in three different layers, as can be seen in Fig. 2. Communication between the frequency inverters and a transducer occurs on the first (lowest) layer, translating the command signals sent by the programmable logic controller (PLC) to $4 - 20$ mA signals. Transducers, positioners, sensors and the PLC are on the second (intermediate) layer. These equipments are intelligent, i.e. they can execute functions of control, mathematics and communicate via Foundation Fieldbus H1 protocol. The last (upper) layer holds the communication between the PLC and a computer via TCP/IP and OPC (OLE for Process Control) protocols. The interface with the user is made by *Elipse SCADA* supervisory. Through the supervisory the user can set the experiments up, configure controller parameters and acquire data

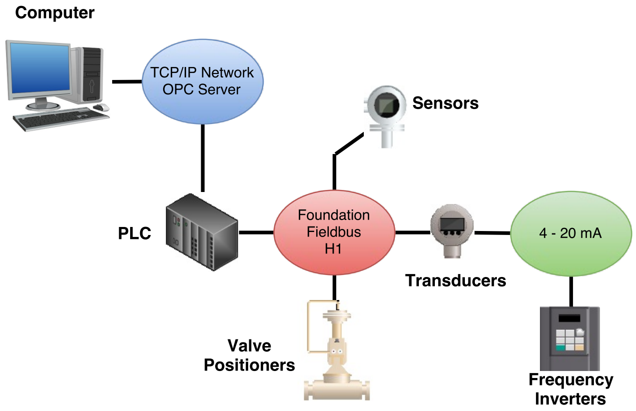

Fig. 2 Plant's network diagram.

from experiments. The flexibility of the system allows the user to establish a connection with MATLAB/Simulink via OPC server to apply input and/or reference signals like steps, ramps, pseudorandom binary sequences (PRBS) and sinusoidal signals. Also, it is possible to implement different control structures, from simple decentralized PID control to complex MPC.

3 Problem formulation

The objective of the control system is reference tracking of both tank levels. Tank 1 level is denoted $y_1(t)$ while $y_2(t)$ is the level of tank 2, where t is the continuous time variable measured in seconds. In order to change the liquid flow and control the tank levels the opening of valves is manipulated while the pumps are kept in a constant speed.

The opening percentage of valve 1 is denoted $u_1(t)$ while $u_2(t)$ is the opening of valve 2. Fig. 1 shows that the water is pumped from the storage to tank 2 through valve 1, from tank 1 to tank 2 through valve 2 and that the liquid returns to the storage by the effect of gravity. This configuration leads to a multivariable behavior of the system, since both inputs ($u_1(t)$ and $u_2(t)$) impact in the level of both tanks ($y_1(t)$ and $y_2(t)$).

The dynamics of this process can be approximated by a linear multivariable model either in continuous- or discrete-time using Laplace or Z transform. The model is written on the frequency domain as

$$Y = GU \tag{1}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \tag{2}$$

where $Y \in \mathbb{R}^2$ is the output vector (liquid levels), $U \in \mathbb{R}^2$ is the input vector (valve openings) and G is the MIMO process model.

The objective of the controller is to automatically choose the input vector U in order to ensure that the output Y respects some performance requirements, usually measured with some characteristics of the system's temporal response, as settling time t_s, rise time t_r, maximum overshoot $M_{o\%}$ or also described as some error norm between the output Y and a reference signal R, which specifies the desired trajectory for the output. Generically, the controller algorithm can be described as a function f that maps the output Y, the reference signal R and some other tuning parameters $P \in \mathbb{R}^p$ to the controller output U:

$$U = f(Y, R, P), \tag{3}$$

This algorithm may be some complex nonlinear model predictive control or some simple PID controller. When PID controllers are used, (3) is simplified and it is given by

$$U = C(P)(R - Y). \tag{4}$$

In other words, the controller signal U is the error between the reference R and the output Y, filtered by the controller transfer function C, which has some tuning parameters P. Usually the parameters comprise the proportional K_P, integral K_I and derivative K_D gains of the MIMO controller.

Since the pilot plant has two inputs and two outputs, the controller can be written as

$$C(P) = \begin{bmatrix} c_{11}(\rho_{11}) & c_{12}(\rho_{12}) \\ c_{21}(\rho_{21}) & c_{22}(\rho_{22}) \end{bmatrix} \tag{5}$$

where each subcontroller $c_{ij}, i = 1, 2; j = 1, 2$ is a PID controller in continuous or discrete time and

$$P = [\rho_{11}^T \; \rho_{12}^T \; \rho_{21}^T \; \rho_{22}^T]^T \tag{6}$$

are the tuning parameters. In continuous-time the controller is written in the frequency domain as

$$c_{ij}(\rho_{ij}) = K_p^{ij} + \frac{K_i^{ij}}{s} + K_D^{ij} s \tag{7}$$

while in discrete-time it is written as (Åström and Wittenmark 1997)

$$c_{ij}(\rho_{ij}) = K_p^{ij} + K_i^{ij} \frac{T_s}{1 - q^{-1}} + K_D^{ij} \frac{1 - q^{-1}}{T_s} \tag{8}$$

where q is the shift operator ($qx(k) = x(k+1)$ and $q^{-1}x(k) = x(k-1)$), $\rho_{ij} = [K_p^{ij} \;\; K_i^{ij} \;\; K_D^{ij}]^T$ and T_s is the sampling interval.

When the system is uncoupled (one process variable does not influence another) or when the performance requirements are not strict (it is allowed that one control-loop disturbs another), it is usual to use a decentralized PID controller that has restrictive structure with $c_{12} = c_{21} = 0$, such that

$$C(P) = \begin{bmatrix} c_{11}(\rho_{11}) & 0 \\ 0 & c_{22}(\rho_{22}) \end{bmatrix}. \tag{9}$$

When the MIMO process (2) is controlled by the MIMO PID controller (4), it is possible to compute the influence of the reference signal R on the output in closed-loop:

$$Y(P) = T(P)R \tag{10}$$

$$T(P) = (GC(P) + I)^{-1} GC(P). \tag{11}$$

where it is explicitly described that the controller parameter vector P affects the output of the closed loop system. The objective of the control methods is to tune the parameters P in order to ensure the outputs Y respect some desired performance requirements.

In this paper, three different control methods are used to control both levels of the pilot plant. Firstly, a simple decentralized PI controller is tuned using the *Multivariable Ultimate Point Method*, then a centralized PI controller is tuned using the *Virtual Reference Feedback Tuning* method. Finally a *Model Predictive Controller* is used to obtain a closed-loop system with strict performance requirements.

4 Data-driven control

Perhaps the most known data-driven control method is the classical Ziegler-Nichols (ZN) tuning method and related methods for SISO systems (Ziegler and Nichols 1942), which are still largely used in industrial applications. Either an open-loop - Step Response Method - or closed-loop - Frequency Response Method - experiment is performed and some features of process dynamics are obtained in order to tune PI or PID controllers through simple formulas. ZN formulas were obtained empirically with a large amount of processes and the expected result of their application is good response to load disturbances. Different sets of formulas were also proposed for enhancing the performance of closed-loop systems to reference changes, like Chien, Hrones and Reswick method (Åstrom and Hägglund 1995). However, using such formulas that are based in only two quantities, the user cannot choose the closed-loop performance, but expect a satisfactory result for cases where process variables' variations are not critical.

When process variables are coupled, the MIMO nature of the process must be taken into account for controller tuning, and SISO tuning methods are not appropriate. MIMO approaches of the frequency response method were proposed for tuning decentralized PID controllers (Loh et al 1993; Palmor et al 1995; Campestrini et al 2009), where a relay feedback experiment is performed to obtain the ultimate quantities, and the controller is obtained from simple formulas based on these quantities. Notice that, as in the SISO case, the user cannot choose a specific closed-loop performance.

For the cases where the performance obtained with ZN tuning is unsatisfactory, more information can be used to controller tuning. Since the 90's many data-driven tuning method were developed. Such methods use a batch or a set of batches of input/output data in an optimization problem which results in parameters of a fixed-structure controller. The most known are Iterative Feedback Tuning (IFT) (Hjalmarsson et al 1998), Correlation based Tuning (CbT) (Karimi et al 2004) and Virtual Reference Feedback Tuning (VRFT) (Campi et al 2002). The main differences between these methods are the optimization criterion used to obtain the controller parameters and the number of input/output collected batches. While VRFT is a non-iterative method that uses one or two batches of input/output data, methods like IFT and CbT are iterative and typically use more than 30 batches. The advantages of these newer methods are many: more information about the process is used, resulting in better controllers; there is a larger flexibility in choosing the structure of the controller, they are not restricted to PID; and the user can choose the *desired* performance for the closed-loop system, while with ZN methods the performance only depends on the chosen table.

In the next subsections we will describe a MIMO version of the ZN methods and a MIMO version of the Virtual Reference Feedback Tuning Method, with application to the pilot plant.

4.1 Multivariable ultimate point method

The multivariable ultimate point method, presented in (Campestrini et al 2009), extends the well-known ultimate point method (Åstrom and Hägglund 1995) for MIMO systems. In the well-known SISO case (Åstrom and Hägglund 1995), the ultimate point is defined as the system frequency response (amplitude and frequency) when its phase reaches ±180°. Usually, the controller tuning is performed using the following procedure:

Step 1: perform a relay feedback experiment so a sustained oscillation is obtained at the system output and collect the oscillation frequency (ultimate frequency ω_u);

Step 2: calculate the ultimate gain K_u with collected relay and output amplitudes;

Step 3: calculate the PI/PID controller gains through simple formulas, using the well known Ziegler-Nichols (Ziegler and Nichols 1942) or Tyreus-Luyben (Luyben 1986) tables.

The Nyquist analysis of the loop function $L(s) = G(s)C(s)$ shows that, in the SISO case, the ultimate point of the loop function is shifted to a specific point in the s-plane, depending on the used formula, far away from the point $-1 + j0$. Table 1 shows points in the s-plane to which the ultimate point is moved using Ziegler-Nichols and Tyreus-Luyben controllers. Notice that PI controllers can move the ultimate point to the second quadrant while PID controllers move it to the third quadrant of the plane.

Table 1 Points to which the ultimate point is moved using Ziegler-Nichols and Tyreus-Luyben controllers.

	Ziegler-Nichols	Tyreus-Luyben
PI	$-0.4 + j0.08$	$-0.31 + j0.023$
PID	$-0.6 - j0.28$	$-0.45 - j0.42$

The Multivariable Ultimate Point method is formulated using the idea of shifting the ultimate point on the complex plane to MIMO systems. The definitions for the ultimate quantities for MIMO systems are based on the multivariable version of Nyquist theorem, known as *Generalized Nyquist theorem* (Maciejowski 1989). As the pilot plant possesses two inputs and two outputs, we present the definition to the TITO case, but the concepts can be easily extended for the general case.

Definition 1 (Characteristic loci) Let the system be put under pure proportional control, i.e. $U = -KY$, where $K = diag\{K_{p_{11}}, K_{p_{22}}\}$. Denote $\lambda_i(s)$, $\forall i = 1, 2$ the eigenvalues of $G(s)K$. The graphs of $\lambda_i(s)$ as s goes around the Nyquist contour are called the characteristic loci. □

The stability limit is reached when at least one characteristic locus crosses the point $-1+j0$ for some $K_u = diag\{K_{u_{11}}, K_{u_{22}}\}$ and, if the process is coupled, both outputs oscillate with the ultimate frequency ω_u. Thus, the objective of the method is to shift both process characteristic loci far away from the point $-1+j0$, bringing them closer to the origin. This can be done by fixing the desired point Λ for both $\lambda_i(j\omega_u)$ or for the one that crosses the point $-1 + j0$ with controller K_u, considering that the other one will be closer to the origin.

Consider a TITO system. Inserting a decentralized controller into the loop results in the loop transfer function in the frequency domain

$$L(j\omega) = \begin{bmatrix} g_{11}(j\omega) & g_{12}(j\omega) \\ g_{21}(j\omega) & g_{22}(j\omega) \end{bmatrix} \begin{bmatrix} c_{11}(j\omega) & 0 \\ 0 & c_{22}(j\omega) \end{bmatrix}, \qquad (12)$$

and the eigenvalues of $G(j\omega)C(j\omega)$ define both characteristic loci for all frequencies:

$$\begin{aligned}\lambda_{1,2}(j\omega) &= \frac{1}{2}g_{11}(j\omega)c_{11}(j\omega) + g_{22}(j\omega)c_{22}(j\omega)\\ &\pm \frac{1}{2}[g_{11}^2(j\omega)c_{11}^2(j\omega) - 2g_{11}(j\omega)c_{11}(j\omega)g_{22}(j\omega)c_{22}(j\omega)\\ &+ g_{22}^2(j\omega)c_{22}^2(j\omega) + 4g_{12}(j\omega)c_{11}(j\omega)g_{21}(j\omega)c_{22}(j\omega)]^{\frac{1}{2}}.\end{aligned} \qquad (13)$$

In this work, we want to move one characteristic locus to the desired point Λ when $\omega = \omega_u$. Setting $\omega = \omega_u$, $\lambda_1(j\omega_u) = \Lambda$ and rearranging terms in (13) we have

$$\begin{aligned}&\Lambda^2 - \Lambda[g_{11}(j\omega_u)c_{11}(j\omega_u) + g_{22}(j\omega_u)c_{22}(j\omega_u)] =\\ &[g_{12}(j\omega_u)g_{21}(j\omega_u) - g_{11}(j\omega_u)g_{22}(j\omega_u)]c_{11}(j\omega_u)c_{22}(j\omega_u).\end{aligned} \qquad (14)$$

For a desired Λ and known $G(j\omega_u)$, (14) has two unknowns ($c_{11}(j\omega_u)$ and $c_{22}(j\omega_u)$). Setting

$$c_{22}(j\omega_u) = \alpha c_{11}(j\omega_u), \qquad (15)$$

where $\alpha = g_{11}(0)/g_{22}(0)$ gives the same relative significance to both loops, and substituting (15) into (14), yields:

$$\begin{aligned}&\alpha[g_{12}(j\omega_u)g_{21}(j\omega_u) - g_{11}(j\omega_u)g_{22}(j\omega_u)]c_{11}(j\omega_u)^2 +\\ &[\Lambda g_{11}(j\omega_u) + \alpha\Lambda g_{22}(j\omega_u)]c_{11}(j\omega_u) - \Lambda^2 = 0.\end{aligned} \qquad (16)$$

Because (16) has two different solutions for $c_{11}(j\omega_u)$, we choose the one where the real part has the same signal of $g_{11}(0)$. This procedure is adopted considering that the controller is decentralized and so it will be more effective for the main diagonal elements of the transfer function matrix.

The PI controller parameters are given by[1]

$$\begin{cases} K_p^{11} = \mathrm{Re}\{c_{11}(j\omega_u)\},\\ K_i^{11} = -\mathrm{Im}\{c_{11}(j\omega_u)\}\omega_u,\\ K_p^{22} = \mathrm{Re}\{c_{22}(j\omega_u)\},\\ K_i^{22} = -\mathrm{Im}\{c_{22}(j\omega_u)\}\omega_u. \end{cases} \qquad (17)$$

Notice that the controller parameters can be calculated if the ultimate frequency ω_u and the process frequency response at that frequency $G(j\omega_u)$ are known. Different configurations of relay feedback experiment can be performed in a MIMO system (Palmor et al 1995; Campestrini et al 2006). Through a decentralized relay feedback (DRF), the MIMO ultimate frequency can be identified. However, this experiment is not sufficiently informative to identify $G(j\omega_u)$ and a second experiment must be performed: either a second DRF, where the relay

[1] For PID controller parameters see (Campestrini et al 2009).

amplitudes are modified, with the same oscillation frequency, or an open-loop experiment, where the process is excited at such frequency (Campestrini et al 2009). The MIMO tuning procedure adopted in this work is as follows:

Step 1 : obtain $G(0)$ from step changes in an open or closed-loop experiment;

Step 2 : perform a DRF experiment in order to obtain a sustained oscillation, collect the input and output signals $u_{1r}(j\omega_u)$, $u_{2r}(j\omega_u)$, $y_{1r}(j\omega_u)$, $y_{2r}(j\omega_u)$ and the oscillation frequency - ultimate frequency ω_u;

Step 3 : perform an open-loop experiment, where the second input is excited with $u_2(t) = B\sin(\omega_u t)$ and collect the output signals $y_{1s}(j\omega_u)$, $y_{2s}(j\omega_u)$;

Step 4 : calculate $G(j\omega_u)$ using the collected signals on *Step 2* and *Step 3* into the set of equations

$$\begin{cases} y_{1r}(j\omega_u) = g_{11}(j\omega_u)u_{1r}(j\omega_u) + g_{12}(j\omega_u)u_{2r}(j\omega_u) \\ y_{2r}(j\omega_u) = g_{21}(j\omega_u)u_{1r}(j\omega_u) + g_{22}(j\omega_u)u_{2r}(j\omega_u), \\ y_{1s}(j\omega_u) = g_{12}(j\omega_u)B \\ y_{2s}(j\omega_u) = g_{22}(j\omega_u)B. \end{cases} \tag{18}$$

Step 5 : Define the desired complex point Λ to which one characteristic loci is moved;

Step 6 : Calculate the PI controller gains through (16) and (17).

For more authentic and accurate signal analysis and frequency response identification, an alternative technique to the common Describing Function method (Åstrom and Hägglund 1995) is applied: the fast fourier transform (FFT) algorithm as proposed in (Bi et al 1997; Wang et al 1997; Mehta 2013) is used to compute the discrete fourier transform (DFT) of each collected signal to be applied to (18).

4.1.1 Experimental Results

The first information needed to apply the Multivariable Ultimate Point method is the constant α. So, in order to obtain it, we performed an open loop experiment on the plant: we manipulated valve 1 from 75% to 65% and valve 2 from 35% to 75% and compared the steady state values for each level. The tank 1 level variation due to the step on valve 1 went from 23.66 cm to 17.21 cm. Tank 2 varied from 22.04 cm to 18.26 cm due to the manipulation of valve 2. That being said, we obtained $g_{11}(0) = 0.645$, $g_{22}(0) = 0.378$, resulting in $\alpha = 1.7$.

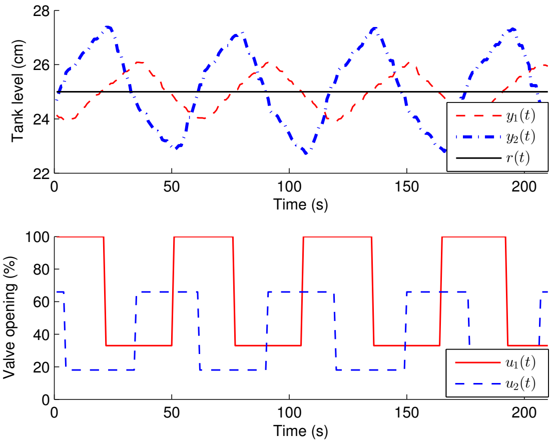

Fig. 3 Inputs and outputs of the DRF experiment on the pilot plant.

The following step is the application of the DRF experiment on the pilot plant to identify the ultimate quantities. We defined the reference signals as $r(t) = r_1 = r_2 = 25$ cm and closed the loop with the bang-bang controller:

$$u_i(t) = \begin{cases} \overline{u_i}, & e_i(t) > 0, \\ \underline{u_i}, & e_i(t) < 0, \ \forall i = 1,2, \end{cases} \tag{19}$$

with $\overline{u_i}$ and $\underline{u_i}$ chosen by the user as the control inputs maximum and minimum values. The objective of this experiment is to change $\overline{u_i}$ and $\underline{u_i}$ until a symmetric oscillation around the reference is reached for both outputs. In Fig. 3 we show the output and control signals obtained in the DRF experiment performed on the pilot plant, where the inputs' limits were chosen as

$$\begin{aligned} \overline{u_1} &= 100\%, & \overline{u_2} &= 66\%, \\ \underline{u_1} &= 33\%, & \underline{u_2} &= 18\%. \end{aligned} \tag{20}$$

The FFT algorithm was applied to the measured signals and the ultimate frequency was estimated as 0.11 rad s^{-1}. With the knowledge of the ultimate frequency we were able to implement the extra experiment, with the following inputs:

$$u_1(t) = 67\%, \tag{21}$$

$$u_2(t) = 42\% + 10sin(0.11t), \tag{22}$$

where the bias was inserted to maintain the process near the DRF experiment operation region. The output signals obtained from this experiment are shown in Fig. 4. Again, the FFT algorithm was applied to the collected signals, allowing the identification of $G(j\omega_u)$ by solving (18):

$$\begin{aligned} g_{11}(j0.11) &= -0.0046 + j0.0002 = 0.0046\underline{/178.06^\circ}, \\ g_{12}(j0.11) &= 0.0004 + j0.026 = 0.026\underline{/89.13^\circ}, \\ g_{21}(j0.11) &= -0.055 - j0.0433 = 0.0436\underline{/-97.3^\circ}, \\ g_{22}(j0.11) &= -0.0046 + j0.0195 = 0.02\underline{/-103.3^\circ}. \end{aligned} \tag{23}$$

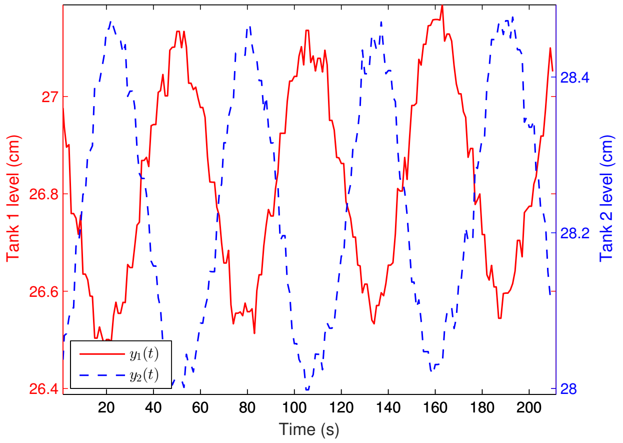

Fig. 4 Output signals of the process on the extra experiment: application of a sinusoidal signal.

To prove that the ultimate point was in fact identified, the critical gain was calculated:

$$
\begin{aligned}
k_{u_{11}} &= \frac{|u_{1r}(j0.11)|}{|y_{1r}(j0.11)|} = 45.47,\\
k_{u_{22}} &= \frac{|u_{2r}(j0.11)|}{|y_{2r}(j0.11)|} = 14.8,
\end{aligned} \tag{24}
$$

and from (13), the following characteristic loci positions were obtained:

$$
\begin{aligned}
\lambda_1(j0.11) &= 0.724 - j0.216 = 0.755\underline{/-16.9^\circ},\\
\lambda_2(j0.11) &= -1.0016 - j0.0657 = 1.0037\underline{/-176.25^\circ}.
\end{aligned} \tag{25}
$$

With the identification of the process frequency response at its ultimate frequency we proceed to the calculation of the PI controllers. We choose to move the second characteristic loci $\lambda_2(j0.11)$ to two distinct desired points: the Ziegler-Nichols point (i.e. $\varLambda = -0.4 + j0.08$) and the Tyreus-Luyben point (i.e. $\varLambda = -0.31 + j0.0023$). Solving (16) and (15) with the desired points and $G(j0.11)$, we get the following controllers for the Ziegler-Nichols point:

$$
c_{11_{ZN}}(s) = \frac{7.39(s + 0.005792)}{s} \tag{26}
$$

$$
c_{22_{ZN}}(s) = \frac{12.61(s + 0.005792)}{s}, \tag{27}
$$

and for the Tyreus-Luyben point:

$$
c_{11_{TL}}(s) = \frac{5.96(s + 0.004157)}{s} \tag{28}
$$

$$
c_{22_{TL}}(s) = \frac{12.61(s + 0.004157)}{s}. \tag{29}
$$

Finally, the controllers were implemented and the system was put to operate in closed loop. For both controller sets, we started the process with a reference of 30 cm and then applied at each reference a step of −10 cm at different instants. Fig. 5 shows the closed loop response of the levels for each case and Fig. 6 shows the control signals. Both methods resulted in stable closed-loop systems. As expected, the change of setpoint in one loop affects the level of the other loop since the designed PI controller is decentralized. In the next section we present the Virtual Reference Feedback Tuning which is also a data-driven method but can be used to design centralized controllers (with full C matrix), enhancing the closed-loop performance.

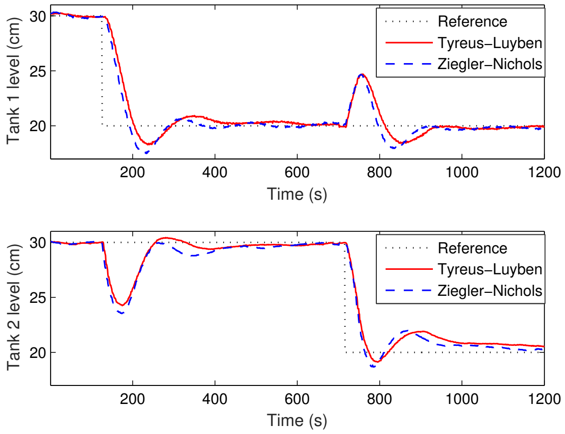

Fig. 5 Comparison between the closed loop responses for the Ziegler-Nichols and the Tyreus-Luyben PI tuning.

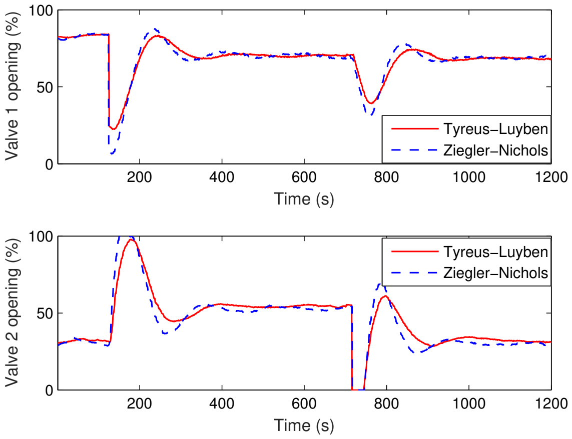

Fig. 6 Comparison between the control signals for the Ziegler-Nichols and the Tyreus-Luyben PI tuning.

4.2 Virtual Reference Feedback Tuning method

Virtual Reference Feedback Tuning is a non-iterative data-driven method that solves an optimization problem to tune the parameters of a fixed-structure linear controller (Campi et al 2002; Campestrini et al 2016). It assumes that a feedback controller is used as described in equation (4).

The MIMO controller $C(P)$ may have a full structure, as in (5), or be decentralized, as presented in (9). Each non-zero sub-controller is discrete in time and linearly parametrized, that is

$$c_{ij}(q, \rho_{ij}) = \rho_{ij}^T \bar{C}_{ij}(q), \rho_{ij} \in \mathbb{R}^p, \tag{30}$$

where ρ_{ij} is the vector of tuning parameters, $\bar{C}_{ij}(q)$ is a p-vector of fixed causal rational functions.

The controller parameters can be grouped in an extended vector P, which is given by (6) in the case of a full controller and represents all parameters to be obtained by the method.

Many data-driven methods, including VRFT, solve a model reference problem, where the controller parameters are adjusted so the obtained closed-loop response $Y(P)$ is as close as possible to a desired response

$$Y_d = T_d R, \tag{31}$$

where T_d is the *reference model* and represents the closed loop behavior chosen by the user. The model reference problem is usually described as

$$\min \|(T(P) - T_d)R\|, \tag{32}$$

which depends on the unknown plant G.

Instead, VRFT minimizes another cost function, which depends only on input-output collected data. Besides, since the controller is linear in the parameters, the VRFT cost function is quadratic and solved through linear least-squares, avoiding local minima.

Consider an open or closed-loop experiment, from where input/output data,

$$\mathcal{Z} = \{U(k), Y(k), \text{ for } k = 1, \ldots, N\} \tag{33}$$

are collected, where k denotes the discrete time variable, since the system is sampled and $t = kT_s$. From the output signal $Y(k)$ the user calculates a virtual reference signal $\bar{R}(k)$ that is defined such that $T_d(q)\bar{R}(k) = Y(k)$, and the virtual error, which is given by $\bar{E}(k) = \bar{R}(k) - Y(k)$.

Even though the plant G is unknown, when it is fed by $U(k)$ (the measured input signal), it generates $Y(k)$ as output. So, a "good" controller is one that generates $U(k)$ when fed by $\bar{E}(k)$. Since both signals $U(k)$ and $\bar{E}(k)$ are known, the controller design can be seen as the identification of the dynamical relation between $\bar{E}(k)$ and $U(k)$. As a result of this reasoning, the MIMO VRFT method minimizes the following criterion

$$P = \arg\min_P J_{VR}(P) \tag{34}$$

$$J_{VR}(P) = \sum_{k=1}^{N} \|F(q)[U(k) - C(q,P)\bar{E}(k)]\|_2^2 \tag{35}$$

where now $U(k)$ and $\bar{E}(k)$ are vectors, $C(q, P)$ is the controller matrix, $F(q)$ is a filter and $\| \cdot \|_2^2$ is the L-2 norm squared of a matrix. This filter is used to approximate the minima of (32) and (35). When data is collected in open loop, an usual choice for this filter is given by

$$F(q) = T_d(q)(I - T_d(q)).$$

When data is collected in closed loop, filter $F(q)$ is also a function of signal espectra, which should be estimated (Campestrini et al 2016).

The controller design through VRFT can be summarized as follows:

Step 1: perform an open-loop or closed-loop experiment and collect the process input and output signals;

Step 2: choose the controller structure, define both reference model $T_d(q)$ and filter $F(q)$;

Step 3: calculate the controller parameter vector P as the solution of (34), which is given by (Campestrini et al (2016)):

$$P = \left(\sum_{k=1}^{N} \varphi(k)\varphi^T(k) \right)^{-1} \sum_{k=1}^{N} \varphi(k)W(k), \tag{36}$$

where

$$\begin{aligned} &W(k) = F(q)U(k), \quad \varphi(k) = [A_1 \; A_2], \\ &A_x = \begin{bmatrix} F_{x1}E_1(k) \\ F_{x2}E_2(k) \end{bmatrix}, \quad E_x(k) = \begin{bmatrix} \bar{C}_{x1}(q)\bar{e}_1(k) \\ \bar{C}_{x2}(q)\bar{e}_2(k) \end{bmatrix} \end{aligned} \tag{37}$$

for $x = 1, 2$.

4.2.1 Experimental Results

In order to apply the MIMO-VRFT method presented above to tune the pilot plant controller, an open-loop experiment was performed, applying PRBS signals in both inputs, for 8000 s, where the sampling time was $T_s = 1$ s. This signal is widely used in system identification because it is deterministic, has a fixed amplitude and excites a large band of frequencies. The input signals are presented in Fig. 7 while the obtained output signals are presented in Fig. 8.

In this work, we used VRFT to design a centralized PI controller, such that

$$c_{ij}(q, \rho_{ij}) = \begin{bmatrix} K_p^{ij} & K_i^{ij} \end{bmatrix} \begin{bmatrix} 1 \\ \frac{1}{1-q^{-1}} \end{bmatrix}, \text{for } \begin{array}{l} i = 1, 2, \\ j = 1, 2. \end{array} \tag{38}$$

The reference model was chosen to obtain a closed-loop response that is faster than the open-loop one,

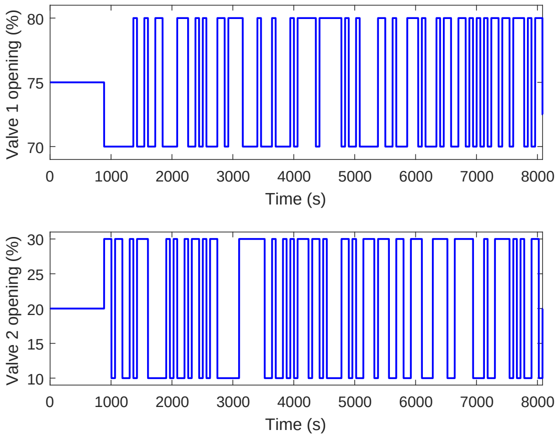

Fig. 7 PRBS input signals applied to the pilot plant.

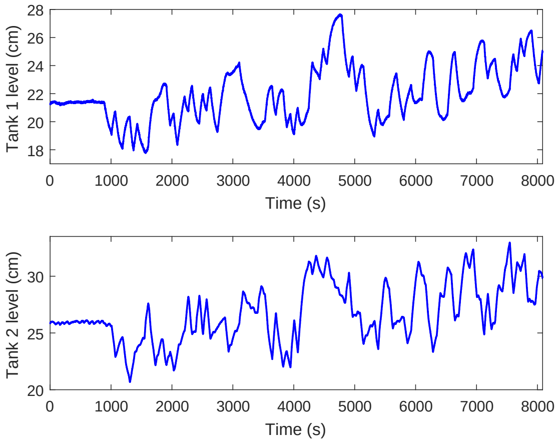

Fig. 8 Output signals of the pilot plant obtained with input signals presented in Fig. 7.

with no overshoot and null steady-state error. Based on that, we chose

$$T_d(q) = \begin{bmatrix} \frac{0.03q^{-1}}{1-0.97q^{-1}} & 0 \\ 0 & \frac{0.02q^{-1}}{1-0.98q^{-1}} \end{bmatrix}, \tag{39}$$

which represents performances with settling time of 128 s for the first output and 193 s for the second output. Filter $F(q)$ was chosen as $F(q) = T_d(q)(I - T_d(q))$ (Campestrini et al 2016).

The obtained controller is given by

$$C(q,P) = \begin{bmatrix} \frac{4.3893(1-0.9918q^{-1})}{(1-q^{-1})} & 3.12 \\ \frac{-10.493(1-0.9933q^{-1})}{(1-q^{-1})} & \frac{0.266(1-0.8282q^{-1})}{(1-q^{-1})} \end{bmatrix}, \tag{40}$$

which results in the closed-loop performance presented in Fig. 9, with control signals presented in Fig. 10. Notice that even though we have chosen to tune PI sub-controllers, in $C_{12}(q, \rho_{12})$ integrator term is null and the controller became only a Proportional one. The obtained result is very similar to the desired response, where the influence of one loop in the other one is practically unnoticed, for both outputs.

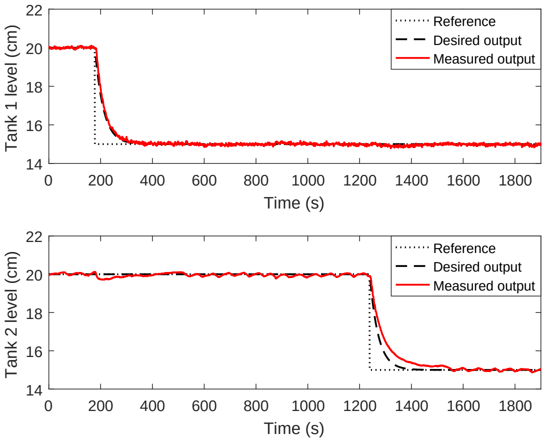

Fig. 9 Closed-loop response obtained with the estimated controller $C(q, P)$ through VRFT compared to the reference model response.

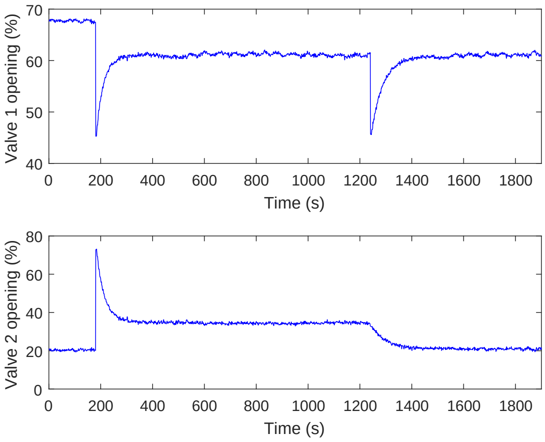

Fig. 10 Control signals of the closed-loop system with the estimated controller $C(q, P)$ through VRFT.

Notice that a closed-loop experiment could have been performed to collect data, the only restriction is that the collected data should be informative enough to estimate all the controller parameters (8 in the case of a centralized PI). In this case, the relay experiment would not be informative enough, resulting in a poor performance; however the closed loop data obtained with one of the previous controllers (ZN or TL) would suffice.

5 Model-based control

In the previous section we have presented two different data-driven methods to design linear-time invariant controllers for MIMO process. These methods use only data from experiments to find the controller parameters without using a model of the process. The methods may achieve reasonable performance, as shown by the experimental results, but they were not developed for difficult cases when there are hard limitations on control signals or when the settling time must be very short.

When high performance is desired, usually the designer needs to resort to more complex control structures that consider limitations on the plant and use a complete mathematical description of the process. Such methodology is classified as *model based control.* One of the most used high performance control algorithms is model predictive control (MPC), which often uses a model to predict the output of the system and optimizes the control signal to improve closed-loop performance. The control signal is optimized at each new collected sample, as described in (3), where it is explicitly shown that the control signal depends on the reference signal R, plant output Y and a set of parameters P, which describes here both the adjustable parameters of the algorithm and the model of the process.

An important element on these *model based control* methods is a model of the process that precisely describes the relations between input and output data. In the next subsection we describe how the model of the pilot plant was obtained and the use of MPC to design high performance controllers.

5.1 Mathematical modeling

Consider the liquid level plant diagram presented in Fig. 1. It is assumed that flow $F(t)$, defined in $\mathrm{cm}^3\,\mathrm{s}^{-1}$, at each valve is proportional to the percentage opening $u(t)$, and that valve dynamics can be neglected since they are much faster than process dynamics. Then we have:

$$F_i(t) = k_{vi} u_i(t),\ \forall\ i = 1,2, \tag{41}$$

where $F_{1,2}$ are the flow at each valve and $k_{v1,2}$ are the flow gains, in $\mathrm{cm}^3\,\mathrm{s}^{-1}$. Also, it is assumed that the discharge flow on both tanks ($F_{d1,d2}$) can be written as function of the levels by the flow equation through an orifice:

$$F_{di}(t) = C_{di} A_{0i} \sqrt{2 g y_i(t)} = k_{di} \sqrt{y_i(t)},\ \forall\ i = 1,2, \tag{42}$$

with C_{di} being the discharge coefficient of the orifice, A_{0i} the orifice's area in cm^2, g the gravity acceleration constant in $\mathrm{cm\,s}^{-2}$ and k_{di} the discharge gain in $\mathrm{cm}^{5/2}\,\mathrm{s}^{-1}$.

Developing the mass-balance equations of each tank, we obtain a system of differential equations describing the level variation at each tank:

$$\dot{y_1}(t) = \frac{-F_2(t) - F_{d1}(t) + F_{d2}(t)}{A_{t1}} \tag{43}$$

$$\dot{y_2}(t) = \frac{F_1(t) + F_2(t) - F_{d2}(t)}{A_{t2}}, \tag{44}$$

with A_{t1} and A_{t2} being the cross section area of tanks 1 and 2 respectively, in cm^2. Using (41) and (42) we get the following pair of nonlinear differential equations:

$$\dot{y_1}(t) = -\frac{k_{v2}}{A_{t1}} u_2(t) - \frac{k_{d1}}{A_{t1}} \sqrt{y_1(t)} + \frac{k_{d2}}{A_{t1}} \sqrt{y_2(t)} \tag{45}$$

$$\dot{y_2}(t) = \frac{k_{v1}}{A_{t2}} u_1(t) + \frac{k_{v2}}{A_{t2}} u_2(t) - \frac{k_{d2}}{A_{t2}} \sqrt{y_2(t)}, \tag{46}$$

where $\dot{y}_{1,2}(t)$ are defined in $\mathrm{cm\,s}^{-1}$.

This model is now linearized around the equilibrium point: y_{1e}, y_{2e}, u_{1e} and u_{2e}. The linearized model is very useful for posterior identification and control. Using the following change of variables:

$$y_{1\delta}(t) = y_1(t) - y_{1e} \tag{47}$$

$$y_{2\delta}(t) = y_2(t) - y_{2e} \tag{48}$$

$$u_{1\delta}(t) = u_1(t) - u_{1e} \tag{49}$$

$$u_{2\delta}(t) = u_2(t) - u_{2e} \tag{50}$$

and truncation of the Taylor Series on the linear terms results

$$\begin{bmatrix} \dot{y_{1\delta}}(t) \\ \dot{y_{2\delta}}(t) \end{bmatrix} = \begin{bmatrix} -a_{11} & a_{12} \\ 0 & -a_{22} \end{bmatrix} \begin{bmatrix} y_{1\delta}(t) \\ y_{2\delta}(t) \end{bmatrix} + \begin{bmatrix} 0 & -b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} u_{1\delta}(t) \\ u_{2\delta}(t) \end{bmatrix}, \tag{51}$$

where the constants are shown at Table 2 as function of the physical attributes.

Table 2 Constants of the hydraulic system.

Const.	Value	Unit	Const.	Value	Unit
a_{11}	$\frac{k_{d1}}{2A_{t1}\sqrt{y_{1e}}}$	s^{-1}	b_{12}	$\frac{k_{v2}}{A_{t1}}$	$\mathrm{cm\,s}^{-1}$
a_{12}	$\frac{k_{d2}}{2A_{t1}\sqrt{y_{2e}}}$	s^{-1}	b_{21}	$\frac{k_{v1}}{A_{t2}}$	$\mathrm{cm\,s}^{-1}$
a_{22}	$\frac{k_{d2}}{2A_{t2}\sqrt{y_{2e}}}$	s^{-1}	b_{22}	$\frac{k_{v2}}{A_{t2}}$	$\mathrm{cm\,s}^{-1}$

This model can also be written as a transfer function matrix

$$Y_\delta(s) = G_\delta(s) U_\delta(s), \tag{52}$$

$$\begin{bmatrix} y_{1\delta}(s) \\ y_{2\delta}(s) \end{bmatrix} = \begin{bmatrix} \frac{a_{12}b_{21}}{(s+a_{11})(s+a_{22})} & \frac{-b_{12}s}{(s+a_{11})(s+a_{22})} \\ \frac{b_{21}}{s+a_{22}} & \frac{b_{22}}{s+a_{22}} \end{bmatrix} \begin{bmatrix} u_{1\delta}(s) \\ u_{2\delta}(s) \end{bmatrix}. \tag{53}$$

The discrete-time version of the above model can be written, with the backward shift operator, as

$$Y_\delta(k) = G_\delta(q,\theta)U_\delta(k), \tag{54}$$

$$G_\delta(q,\theta) = \begin{bmatrix} \frac{\theta_1 q^{-2}}{(1+\theta_5 q^{-1})(1+\theta_6 q^{-1})} & \frac{-\theta_2(1-q^{-1})}{(1+\theta_5 q^{-1})(1+\theta_6 q^{-1})} \\ \frac{\theta_3 q^{-1}}{1+\theta_6 q^{-1}} & \frac{\theta_4 q^{-1}}{1+\theta_6 q^{-1}} \end{bmatrix}, \tag{55}$$

where parameters θ_i represent the constants of the system, and can be written in a vector form as

$$\theta^T = \begin{bmatrix}\theta_1\ \theta_2\ \theta_3\ \theta_4\ \theta_5\ \theta_6\end{bmatrix}. \tag{56}$$

In principle, constants θ_i can be calculated using information from Table 2 which depends on physical parameters k_{di}, k_{vi}, A_{ti} and operation point y_{ie}. However, these physical parameters are unknown and difficult to obtain. In this work, the parameter vector θ is estimated directly from input-output data, comparing the simulation of the model with data collected from experiments. The parameters are estimated solving the following optimization problem:

$$\hat{\theta} = \arg\min_\theta \sum_{k=1}^{N} \|Y_0(k) - G_\delta(q,\theta)U_0(k))\|^2 \tag{57}$$

where $Y_0(k), U_0(k), i = 1, 2, \ldots, N$ is a batch of input-output data collected from an experiment. The optimization problem is nonconvex and can be solved using the Newton-Raphson algorithm.

The same input-output data used by VRFT, presented in Fig. 7 and Fig. 8, were used to identify the parameters of the process model. The signals' mean values were subtracted from the input/output signals in order to take into account the assumption that the model is valid for that equilibrium point. The resulting parameters of the identification procedure are shown at Table 3. The RMS error between the experimental data and simulation with the model is 0.76 *cm*, which represents a good but not perfect model.

Table 3 Identified parameters.

Parameter	Value	Parameter	Value
θ_1	3.972×10^{-5}	θ_4	2.720×10^{-3}
θ_2	2.974×10^{-3}	θ_5	-9.926×10^{-1}
θ_3	6.331×10^{-3}	θ_6	-9.943×10^{-1}

5.2 Model Predictive Control

Model Predictive Control theory comprehends a set of controller design methods which provide an optimal control signal through minimizing an objective function over a future-time horizon. This strategy of dynamically optimizing the control action is also known as Receding Horizon Control, and it allows the user to consider limitations of the process in the form of numerical constraints (Camacho and Bordons 2007). In order to implement such method, an explicit model of the process is used to generate a prediction of the system's behavior considering past input and output measurements.

There exist multiple predictive control methods, which follow the same basic premises but use different predictor structures or cost functions in their formulation. Because of its intuitive character, one of the most popular MPC methods is the Generalized Predictive Control (GPC), which will be discussed in this paper.

The classical formulation of GPC was adapted for an autoregressive model with exogenous variables (ARX model), here described for a multivariable system of m inputs and n outputs, given as

$$A(q)y(k) = B(q)u(k-1) + \epsilon(k) \tag{58}$$

where $A(q)$ and $B(q)$ are polynomial matrices. Also, $u(k) \in \mathbb{R}^m$, $y(k) \in \mathbb{R}^n$ and $\epsilon(k) \in \mathbb{R}^n$ represent the input, the output and the white noise signals respectively.

From the system's model, it is possible to deduce a predictor of the outputs at a future instant $k+j$ based on information up until instant k, which will be expressed as $\hat{y}(k+j|k)$. In (Camacho and Bordons 2007), such predictor is developed by considering diophantine equations. It is also straightforward to do so by deriving the optimal predictor for the ARX model, as seen in (Söderström and Stoica 1988) for general linear models.

The expression obtained for the predictor is

$$\hat{y}(k+j|k) = S_j(q)\Delta u(k+j-1) + f_j \tag{59}$$

where S_j is a polynomial matrix whose coefficients correspond to the plant step response coefficients. Also vector f_j can be obtained recursively as

$$f_{j+1} = q(I - A(q))f_j + B(q)u(k+j) \tag{60}$$

where $f_0 = y(k)$ and $u(k+l) = u(k-1),\ \forall l \geq 0$.

The vector of control signal variations is obtained at each instant k by minimizing a quadratic cost function

$$\Delta u_F = \arg\min_{\Delta u} J^{GPC} \tag{61}$$

where

$$J^{GPC} = \sum_{j=N_1}^{N_2} ||\hat{y}(k+j|k) - r(k+j)||_R^2 + \sum_{j=1}^{N_u} ||\Delta u(k+j-1)||_Q^2 \tag{62}$$

where $r(k+j)$ corresponds to the future reference signals, $[N_1, N_2]$, the prediction horizon and N_u, the control horizon or yet the number of future control actions considered. It can be noted from (62) that both the reference tracking and the control rate are minimized at each instant k. Also R and Q are weighting matrices, which constitute the optimization parameters P, as seen in (3).

It should be noted that Δu_F consists of N_u control actions for each input signal of the process, i.e. $N_u \times m$ control actions. Only the first m future control actions calculated are actually applied to the inputs, the rest of them being discarded.

The following matrix representation can be derived from (59) (Camacho and Bordons 2007):

$$\mathbf{y} = \mathbf{G}\tilde{\mathbf{u}} + \mathbf{f} \tag{63}$$

where

$$\mathbf{y} = \left[\, \hat{y}(k+N_1)^T \, \ldots \, \hat{y}(k+N_2)^T \,\right]^T \tag{64}$$

$$\tilde{\mathbf{u}} = \left[\, \Delta u(k)^T \, \ldots \, \Delta u(k+N_u-1)^T \,\right]^T \tag{65}$$

$$\mathbf{f} = \left[\, f_{N_1}^T \cdots f_{N_2}^T \,\right]^T \tag{66}$$

and the matrix $\mathbf{G}$ corresponds to

$$\mathbf{G} = \begin{bmatrix} G_{N_1} & \ldots & G_{N_1-N_u+1} \\ \vdots & \ddots & \vdots \\ G_{N_2} & \ldots & G_{N_2-N_u+1} \end{bmatrix} \tag{67}$$

where G_j, $j = N_1, \ \ldots, \ N_2$, is a matrix of dimension $n \times m$, whose component $G_j(i,l)$ represents the *i-th* system output at instant $k+j$ for a step signal applied to the *l-th* input at instant k.

In order to obtain the control law, let us rewrite the objective function presented in (62) as

$$J^{GPC} = (\mathbf{y} - \mathbf{r})^T R (\mathbf{y} - \mathbf{r}) + \tilde{\mathbf{u}}^T Q \tilde{\mathbf{u}} \tag{68}$$

where

$$\mathbf{r} = \left[\, r(k+N_1)^T \, \ldots \, r(k+N_2)^T \,\right]^T \tag{69}$$

and $r(k) \in \mathbb{R}^n$ corresponds to the reference signal vector.

Substituting (63) in (68), we have

$$J^{GPC} = (\mathbf{G}\tilde{\mathbf{u}} + \mathbf{f} - \mathbf{r})^T R (\mathbf{G}\tilde{\mathbf{u}} + \mathbf{f} - \mathbf{r}) + \tilde{\mathbf{u}}^T Q \tilde{\mathbf{u}}. \tag{70}$$

For a free optimization problem, the optimal solution of $\tilde{\mathbf{u}}$ minimizing (70) is easily obtained as

$$\Delta u_F = (\mathbf{G^T} R \mathbf{G} + Q)^{-1} \mathbf{G}^T R (\mathbf{r} - \mathbf{f}). \tag{71}$$

The predictive controller design, for each instant k, can be summarized as

Step 1: measure the system's outputs $y(k)$;

Step 2: predict future behavior of the outputs over the prediction horizon based on the explicit model of the system, past values of $u(k)$ and $y(k)$ and present value of $y(k)$, as shown in (60);

Step 3: solve the minimization problem in (62) and find the vector of $N_u \times m$ control rates Δu_F;

Step 4: apply only the first m control signal rates of Δu_F to the process at instant k.

5.2.1 Experimental Results

The GPC algorithm was used to design a controller for the pilot plant using the identified model described in (55) with parameters given in Table 3. The choice of the prediction horizon will have an influence upon the response time of the closed loop system: the larger it is, the closer the closed-loop poles will be from the open-loop ones (Clarke et al 1987). For the application, we will consider weight matrices $R = 2 \times I$ and $Q = I$. The algorithm has been then set for a prediction horizon of $[N_1, N_2] = [1, 30]$ and a control horizon of $N_u = 30$.

Moreover, in order to avoid aggressive transient behaviors a reference filter $L(q)$ is used in the controller design, given by $L(q) = \frac{0.05}{(q-0.95)} \times I$. Therefore, in the GPC formulation, $r(k)$ is replaced by the filtered signal $r_L(k) = L(q)r(k)$.

From the process' model previously identified, its step response is easily obtained. Next, matrix $\mathbf{G}$ from (68) is generated, considering the chosen prediction and control horizons. For the purpose of comparing expected and real behavior, a scenario of set-point change has been simulated and then implemented on the actual pilot plant through OPC server with Simulink.

The scenario consisted of a step of -5 cm applied on the reference signals for the outputs $y_1(k)$ and $y_2(k)$ at different instants of time. The simulated and experimental results can be seen in Fig. 11 for the tanks' levels and in Fig. 12 for the valves' opening, with an initial operation point of 20 cm on tank 1 level and 35 cm on tank 2 level.

In Fig. 11, overall measured response is quite similar to the simulated one, which indicates a good correpondence between model and real behavior. Small differences can be noted, especially on the disturbance effect seen in output $y_2(k)$, due to slight imprecisions of the identified model. We can equally observe that noise is amplified in control signal $u_2(t)$, which reflects high performance demand (in our case, the weight on reference tracking is twice as important as the weight on control effort).

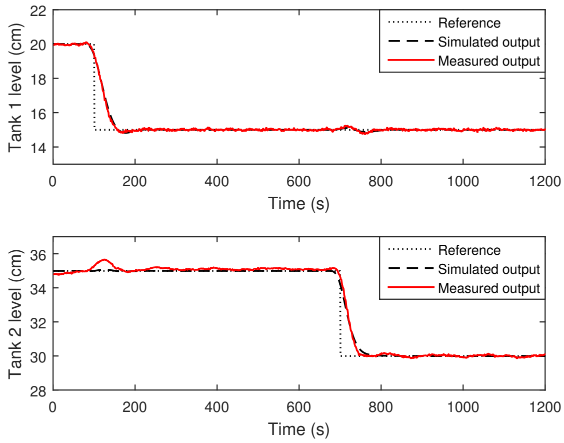

Fig. 11 Closed-loop response obtained with GPC in simulation and experiment.

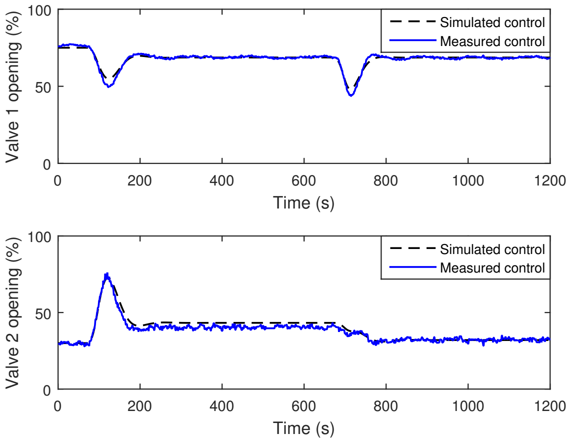

Fig. 12 Control signals of the closed-loop system with GPC in simulation and experiment.

6 Discussion

This paper has presented three different controllers for MIMO processes. In this section, a comparison between the three controllers is made where many aspects of each method are compared: structure of the controller, information needed from the MIMO process, design parameters and achieved performance.

6.1 Structure of controllers

Three different controller structures were used with the Pilot Plant: a decentralized PI controller, a centralized PI controller and an optimization based controller (GPC).

The PI controller designed with the MUPM method has the following decentralized structure:

$$C(P) = \begin{bmatrix} c_{11}(\rho_{11}) & 0 \\ 0 & c_{22}(\rho_{22}) \end{bmatrix}. \tag{72}$$

where the elements outside the diagonal are zero. This kind of controller is usually used when the interactions between loops are small, because the influence of one loop in another is disregarded. The Pilot Plant used to test the controller has significant interaction between loops, and as expected, a change of reference in one loop has affected the other loop.

The PI controller designed with the VRFT method has the following centralized structure:

$$C(P) = \begin{bmatrix} c_{11}(\rho_{11}) & c_{12}(\rho_{12}) \\ c_{21}(\rho_{21}) & c_{22}(\rho_{22}) \end{bmatrix}. \tag{73}$$

where the matrix is full. This kind of controller is usually used when the interactions between loops are significant, because the influence of one loop in another is considered. As seen in the experiments, this centralized controller can cope with the interactions between loops, and reference changes do not produce perturbations on the other loop.

The GPC controller has completely different structure. The control signal $u(t)$ is computed as the solution of an optimization problem, that considers both MIMO dynamics of the process and nonlinearities, for instance saturations. This kind of controller is usually used when high performance is demanded from the closed-loop system, which often saturates the control signal. In the experiments a very good performance was obtained without saturating the control signal.

6.2 Information needed to design the controller

The MUPM is a data-driven method that does not use a mathematical model of the process to design the gains of the controller. Instead, it uses collected data from two specific experiments. The first experiment is a closed-loop one with a relay controller which makes the system oscillate with the ultimate frequency. After an open-loop experiment is needed where the input is sinusoidal with the ultimate frequency. Although not using a model (which makes the design simpler), these two experiments are not so easy to implement, since both open-loop and closed-loop experiments are needed.

On the other hand, the VRFT method uses only input-output data from the process to design the controller. The experimental data can be obtained from historic data since no specific experiment is needed.

However, it is necessary that the experiment is informative enough. For instance, if an open loop experiment with constant input is realized, then the output is also constant, and there is no information about the dynamics of the system. So, good experimental data should reveal the dynamics of the process, so an open-loop experiment with PRBS is ideal, but closed-loop experiments where all references are changed are usually enough to design the controllers.

Finally, the GPC method needs a complete model of the process, which is usually identified from collected data. The identification of a good model is a hard task, and usually comprises several steps: obtaining the physical relation between variables, experimental data to adjust the constants of the model, selecting an algorithm to obtain the parameters and a performing a validation step. Since GPC uses a complete model of the process, it is usual that the obtained performance is better with this method, at the cost of consuming more resources to obtain the model. When high process performance is required, obtaining the model is worth it.

6.3 Design parameters

The only parameter of the MUPM method is the position to where the ultimate point is moved. This point is completely related to the used Table, where the most common are the Ziegler-Nichols and Tyreus-Luyben. Both are usually chosen when the control objective is focused on disturbance attenuation and the Ziegler-Nichols tuning usually results in a more aggressive controller.

With the VRFT method, the designer needs to choose the reference model $T_d(z)$ which describes the desired output of the closed-loop system for each reference signal. With this method the user has a larger flexibility to choose the settling time and desired overshoot. It is common to choose diagonal matrices because they describe that the influence of one loop in another should be null, as done in this work. However, different choices could also be made.

The GPC method has several parameters to be designed. The user needs to choose the control and prediction horizons N_u, N_1 and N_2, matrices R and Q that describe how control effort and reference tracking are emphasized, and reference filter $L(q)$ used to improve the reference tracking behavior. Despite being very flexible, the many choices the user has to make turns the method difficult to adjust. It is common that many of the parameters tuning is done by trial and error.

6.4 Obtained performance

In this subsection we compare the obtained performance with each method. This comparison is not intended to describe which method is better. The objective is to present a numerical comparison so the reader can better understand the limitations of each method.

We decided to use the Integral Time Absolute Error criterion to compare the methods where:

$$ITAE_i = \int_{t_j}^{t_j+400} \frac{t|e_i(t)|}{\Delta r_j} dt, \tag{74}$$

and $e_i(t) = r_i(t) - y_i(t)$ is the i-loop error, and $\Delta r_j(t)$ is a normalizing constant (the amplitude of the step used in the experiment). The criterion is computed after a step signal is applied to one loop. Time t_j is the instant when the step signal is applied to the loop.

Table 4 Comparison of the $ITAE$ performance index for the step on the reference $r_1(t)$

Method	$ITAE_1$	$ITAE_2$
MUPM ZN	4004.27	7166.32
MUPM TL	5048.47	4867.07
VRFT	1999.46	1157.35
GPC	1018.15	1674.38

Table 5 Comparison of the $ITAE$ performance index for the step on the reference $r_2(t)$

Method	$ITAE_1$	$ITAE_2$
MUPM ZN	3866.83	6611.14
MUPM TL	4149.79	8450.38
VRFT	792.64	4255.46
GPC	752.25	1105.07

Notice that first column of Table 4 describes the performance considering reference tracking while the second column describes the performance considering disturbance attenuation. Table 5 is similar, while first column accounts to disturbance attenuation, the second column accounts to reference tracking. It is possible to observe that the MUPM method has the worst performance when compared to VRFT and GPC. This is expected since the Pilot Plant presents strong interaction between loops and the MUPM method uses a decentralized controller that does not take into account this effect. Also, GPC presented the best performance at the cost of needing a mathematical model of the process. The VRFT method presented very good performance when compared to its reference model. Probably a lower ITAE would be obtained if a faster reference model had been chosen. Observe also that no experiment presented input saturation. If this was the case, it would be expected that the GPC method would generate the best performance since it considers the saturation effect in its formulation.

7 Concluding Remarks

We have successfully applied three different MIMO control methods in a liquid level pilot plant. The chosen methods require different knowledge about the process, use different controller structures and therefore result in different performance. We have compared the advantages and drawbacks of each method in a practical point of view. In summary, as more information is available the better is the response. MUPM method uses only information from the critical point and can tune properly a decentralized controller. VRFT uses a batch of data from an experiment and can tune a centralized controller with better performance. GPC needs a complete model from the process and can achieve very good performance using an optimization based controller.

References

Al-Naumani Y, Rossiter J (2015) Distributed MPC for upstream oil & gas fields – a practical view. vol 48, pp 325–330, 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015

Åstrom KJ, Hägglund T (1995) PID controllers: theory, design and tuning, vol 2. Isa Research Triangle Park, NC

Åström KJ, Wittenmark B (1997) Computer-controlled systems: theory and design. Prentice-Hall

Bi Q, Wang QG, Hang CC (1997) Relay-based estimation of multiple points on process frequency response. Automatica 33(9):1753–1757

Bodson M, Groszkiewicz JE (1997) Multivariable adaptive algorithms for reconfigurable flight control. IEEE Transactions on Control Systems Technology 5(2):217–229

Camacho EF, Bordons C (2007) Model Predictive control. Advanced Textbooks in Control and Signal Processing, Springer London, London

Campestrini L, Barros P, Bazanella A (2006) Auto-tuning of PID controllers for MIMO processes by relay feedback. In: IFAC Symposium on Advance Control of Chemical Processes, Gramado - RS - Brazil, pp 451–456

Campestrini L, Stevanatto Filho LC, Bazanella AS (2009) Tuning of multivariable decentralized controllers through the ultimate-point method. IEEE Transactions on Control Systems Technology 17(6):1270–1281

Campestrini L, Eckhard D, Chía LA, Boeira E (2016) Unbiased MIMO VRFT with application to process control. Journal of Process Control 39:35–49

Campi M, Lecchini A, Savaresi S (2002) Virtual reference feedback tuning: a direct method for the design of feedback controllers. Automatica 38:1337–1346

Clarke DW, Mohtadi C, Tuffs PS (1987) Generalized predictive control – part I. the basic algorithm. Automatica 23(2):137–148

Dumont GA (1986) Application of advanced control methods in the pulp and paper industry – a survey. Automatica 22(2):143–153

García C, Prett D, Morari M (1989) Model predictive control: Theory and practice – a survey. Automatica 25(3):335–348

Hjalmarsson H, Gevers M, Gunnarsson S, Lequin O (1998) Iterative feedback tuning: theory and applications. IEEE Control Systems Magazine 18(4):26–41

Jin Q, Hao F, Wang Q (2013) A multivariable IMC-PID method for non-square large time delay systems using npso algorithm. Journal of Process Control 23(5):649–663

Karimi A, Mišković L, Bonvin D (2004) Iterative correlation-based controller tuning. International Journal of Adaptive Control and Signal Processing 18:645–664

Krstic M, Kanellakopoulos I, Kokotovic PV (1995) Nonlinear and adaptive control design. Wiley

Loh AP, Hang CC, Quek CK, Vasnani VU (1993) Autotuning of multiloop proportional-integral controllers using relay feedback. Industrial & Engineering Chemistry Research 32(6):1102–1107

Luyben WL (1986) Simple method for tuning SISO controllers in multivariable systems. Industrial & Engineering Chemistry Process Design and Development 25:654–660

Maciejowski JM (1989) Multivariable feedback design. Electronic systems engineering series, Wokingham, England Reading, Mass. Addison-Wesley

Mehta U (2013) Fast fourier transform for estimating process frequency response. In: 2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA), IEEE, pp 7–10

Palmor ZJ, Halevi Y, Krasney N (1995) Automatic tuning of decentralized pid controllers for TITO processes. Automatica 31(7):1001–1010

Rojas J, Morilla F, Vilanova R (2012) Multivariable PI control for a boiler plant benchmark using the virtual reference feedback tuning. In: 2nd IFAC Conference on Advances in PID Control, 2012, pp 376–381

Skogestad S, Postlethwaite I (2005) Multivariable Feedback Control: Analysis and Design. Wiley

Söderström T, Stoica P (1988) System identification. Prentice-Hall, Inc.

Vu TNL, Lee M (2010) Independent design of multi-loop PI/PID controllers for interacting multivariable processes. Journal of Process Control 20(8):922–933

Wang QG, Bi Q, Zou B (1997) Use of FFT in relay feedback systems. Electronics Letters 33(12):1099–1100

Yubai K, Usami H, Hirai J (2009) Correlation-based direct tuning of MIMO controllers by least-squares and its application to tension-and-speed control apparatus. In: ICCAS-SICE, 2009, IEEE, Fukuoka, Japan, pp 931–936

Ziegler JG, Nichols NB (1942) Optimum settings for automatic controllers. Journal of Dynamic Systems, Measurement, and Control 115(2B):220–222