Published version: http://dx.doi.org/10.1016/j.jprocont.2015.12.010

Unbiased MIMO VRFT with application to process control ☆

Lucíola Campestrini[a], Diego Eckhard[b], Lydia Andrea Chía[c], Emerson Boeira[a]

[a]*Departamento de Sistemas Elétricos de Automação e Energia - Universidade Federal do Rio Grande do Sul*
[b]*Departamento de Matemática Pura e Aplicada - Universidade Federal do Rio Grande do Sul*
[c]*Departamento de Engenharia Elétrica e Informática Industrial - Universidade Tecnológica Federal do Paraná*

Abstract

Continuous process industries usually have hundreds to thousands of control loops, most of which are coupled, i.e. one control loop affects the behavior of another control loop. In order to properly design the controllers and reduce the interactions between loops it is necessary to consider the multivariable structure of the process. Usually MIMO (multiple-input, multiple-output) controllers are designed using MIMO models of the process, but obtaining these models is a task very demanding and time consuming. Virtual Reference Feedback Tuning (VRFT) is a *data-driven* technique to design controllers which do not use a model of the process; all the needed information is collected from input/output data from an experiment. The method is well established for SISO (single-input, single-output) systems and there are some extensions to MIMO process which assume that all the outputs should have the same closed-loop performance. In this paper we develop a complete framework to MIMO VRFT which provides unbiased estimates to the optimal MIMO controller (when it is possible) even when the closed-loop performances are distinct to each loop. When it is not possible to obtain the optimal controller because the controller class is too restrictive (for example PID controllers) then we propose the use of a filter to reduce the bias on the estimates. Also, when the data is corrupted by noise, the use of instrumental variables to eliminate the bias on the estimate should be considered. The article presents simulation examples and a practical experiment on a tree tank system where the goal is to control the level of two tanks.

Keywords: PID Control, MIMO processes, VRFT, Data-driven control

1. Introduction

A large amount of industrial processes can be considered multivariable, due to the interaction between different variables involved. In order to obtain a desired output performance, control design must take into account the MIMO (multiple-input, multiple-output) nature of the process, without simply using SISO (single-input, single-output) tuning rules. One way of tuning PID controllers is to use Ziegler-Nichols methods, which are based in few information on the process. MIMO approaches for tuning decentralized PID controllers are presented in [1, 2]. However, since these methods use few information about the process, the obtained results may be unsatisfactory. A proper way to do that would be first obtaining a MIMO model of the process, then using a MIMO tuning rule, as the ones presented in [3, 4], for example. By doing so, one could find the controller that perfectly matches the desired output performance, usually given by adequate time responses of each controlled variable and no or low interactions between them.

However, there are some drawbacks in applying this procedure: obtaining appropriate MIMO models is usually very demanding and time consuming; even if the obtained MIMO process model is a low order transfer matrix, the controller that would yield the desired response tends to be a high order one, which will probably not be appropriate for implementation. If the goal is to tune PID controllers, which are still widely used in industrial applications, then a controller model reduction should also be applied in order to obtain an implementable controller. Among different methods that can be used to tune multivariable controllers, data-driven methods present an interesting characteristic: they *do not* use a process model. The information about the process is obtained through input and output data collected from process operation or some extra experiment on the system. The controller is tuned through an optimization procedure based on these data, a chosen controller structure (PID, for example) and the closed-loop performance requirements, which are translated into a transfer function matrix, but a mathematical model of the process is not used.

There are several data-driven methods developed for SISO control problems in the literature, some of them are iterative [5, 6, 7, 8], some are one-shot [9, 10, 11]. The main advantage of one-shot methods is that operation data can be used, without the need of performing specific experiments. However, SISO methods are not proper to be used when interactions between variables are significant, and some effort has been put in developing the extensions of these methods to the MIMO case: some are iterative [12, 13] and present the disadvantage of a higher number of experiments needed, others are one-shot [14, 15, 16], based on only one experiment. Besides, the complexity of MIMO one-shot data based methods, considering tuning decentralized or full controllers, is only a bit higher then the SISO

☆This work was partially supported by CNPq/Brasil, CAPES/Brasil and Fapergs/Brasil.

design, and it does not increase as the number of outputs to be controlled increases. These characteristics make one-shot data-driven methods very appealing to process control.

In this work we propose an extension of the Virtual Reference Feedback Tuning (VRFT) method to the MIMO case. The SISO VRFT method was introduced in [9] and its formulation can be seen as the identification of the ideal controller, considering that the user has access to input and output signals of the controller and that the ideal controller belongs to a controller class chosen by the user (a common assumption in data-driven design). Our formulation presents the same properties.

An extension of the VRFT method to MIMO processes is presented in [15], and a deeper investigation of that extension is presented in [17, 18]. As pointed out by the authors, the extension they propose provides the ideal controller when some restrictions are satisfied, including the one of equal closed-loop performance for all variables involved, which is not a restriction of the extension presented in this work. We show that, using our method, different closed-loop performances for each output signal are allowed and the method still results in an unbiased estimate for the ideal controller, if it is in the controller class chosen. When the ideal controller is not in the controller set, a filter can be used to approximate the minima of the VRFT and the model reference criteria. Finally, if signals are corrupted by noise, an instrumental variable should be used. In any of these cases, the main characteristic of VRFT is maintained: it is a one-shot data driven method, where the controller parameters are obtained through the solution of a least squares problem.

The paper is organized as follows. The problem statement is presented in Section 2. Section 3 briefly presents the state of the art of the VRFT method: the formulation of the SISO VRFT and the MIMO VRFT method found in the literature, pointing out its restrictions. The proposed formulation, developed in order to obtain an unbiased estimate of the ideal controller is presented in Section 4, where closed solutions of the problem considering the use of a filter when the ideal controller is not in the controller set and an instrumental variable in the case data are corrupted by noise are presented. Section 5 presents an illustrative example of the proposed method through a simulated process and an application to the level control of a two-input-two-output pilot plant. Conclusions are presented at the end of the paper.

2. Problem Statement

Consider a linear time-invariant discrete-time process

$$y(t) = G(q)u(t) + v(t), \tag{1}$$

where q is the forward-shift operator, $G(q)$ is an $n \times n$ rational transfer function matrix, $u(t)$ and $y(t)$ represent respectively input and output signal of the plant, both represented by an n-dimension column vector, and $v(t)$ is the noise vector.

This process is controlled by an $n \times n$ linear time-invariant controller, which belongs to a given - user specified - class C of transfer function matrices. The controller is parameterized by a parameter vector $P \in \mathbb{R}^p$, so that the control action $u(t)$ can be written as

$$u(t) = C(q, P)(r(t) - y(t)), \tag{2}$$

where $r(t)$ is a reference signal vector, which is assumed to be quasi-stationary.

The system (1)-(2) in closed loop becomes

$$\begin{aligned} y(t,P) &= T(q,P)r(t) + S(q,P)v(t) \\ S(q,P) &= (G(q)C(q,P) + I)^{-1}, \\ T(q,P) &= S(q,P)G(q)C(q,P) = G(q)C(q,P)S(q,P), \end{aligned}$$

where we have now made the dependence on the controller parameter vector P explicit in the set of output signals $y(t, P)$.

The controller class C is defined as

$$C = \{C(q, P) : P \in \Omega \subseteq \mathbb{R}^p\},$$

where the structure of the controller to be designed is defined as follows,

$$C(q,P) = \begin{bmatrix} C_{11}(q,\rho_{11}) & C_{12}(q,\rho_{12}) & \cdots & C_{1n}(q,\rho_{1n}) \\ \vdots & \vdots & & \vdots \\ C_{n1}(q,\rho_{n1}) & C_{n2}(q,\rho_{n2}) & \cdots & C_{nn}(q,\rho_{nn}) \end{bmatrix}, \tag{3}$$

where $P = [\rho_{11}^T \ \rho_{12}^T \ \ldots \ \rho_{n1}^T \ \ldots \ \rho_{nn}^T]^T$. It is also assumed that each subcontroller has a linear parametrization, i.e. they can be written as

$$C_{ij}(q, \rho_{ij}) = \rho_{ij}^T \bar{C}_{ij}(q), \rho_{ij} \in \mathbb{R}^m, \tag{4}$$

where $\bar{C}_{ij}(q)$ is an m-vector of fixed causal rational functions. Besides each subcontroller can have a different structure, provided that each one is linear in the parameters.

As an example, a full PID controller can be written as (3) with

$$C_{ij}(q,\rho_{ij}) = \begin{bmatrix} K_{pij} \ K_{iij} \ K_{dij} \end{bmatrix} \begin{bmatrix} 1 \\ \frac{q}{q-1} \\ \frac{q-1}{q} \end{bmatrix}, \quad \text{for} \quad \begin{matrix} i = 1, ..., n, \\ j = 1, ..., n. \end{matrix} \tag{5}$$

The closed-loop performance is specified through a “desired” closed loop transfer function matrix $T_d(q)$, also known as the *reference model*,

$$y_d(t) = T_d(q)r(t),$$

where $T_d(q)$ describes the relation between the reference signal and the desired output of the closed-loop system.

The controller parameters may be tuned solving the model reference (MR) optimisation problem

$$\hat{P} = \arg\min_P J^{MR}(P) \tag{6}$$

$$J^{MR}(P) \triangleq \sum_{t=1}^{N} \|(T_d(q) - T(q,P))r(t)\|_2^2. \tag{7}$$

The *ideal controller* $C_d(q)$ is the one that allows the closed loop system to match exactly $T_d(q)$ and is given by

$$C_d(q) = G(q)^{-1}T_d(q)(I - T_d(q))^{-1}. \tag{8}$$

We can then define the bias as the error between the ideal controller and the estimated one:

$$BIAS = E[C_d(q) - C(q, \hat{P})], \tag{9}$$

where E means the expected value.

In general, the ideal controller can not be achieved solving the optimization problem (6) because the controller structure C may be restrictive, which results in $BIAS$ different from zero. However, this is the best controller that could run in closed-loop and whenever is possible it should be used. When the structure C is such that the controller $C_d(q)$ can be represented we say that the ideal controller belongs to the controller class.

Assumption 1. *$C_d(q) \in C$: There is a P_0 such that $C(q, P_0) = C_d(q)$.*

When Assumption 1 is respected, the parameter vector P_0 solves the optimisation problem (6) and it may be computed using (8). However, (8) depends on the process model $G(q)$, which is assumed to be unknown by the user, so it can not be used to solve the optimisation problem (6). In this work, we want to solve the optimisation problem using only input/output data as information about the process - no other information as the transfer function model is available.

There are some techniques in the literature that aim to solve the optimisation problem (6) using only input/output data from the process, which are known as *data-driven* methods in contrast to the *model-based* techniques, which rely on the process model. The Iterative Feedback Tuning (IFT) [5] is probably the most known method. It is an iterative method which needs data from two specific experiments at each iteration to solve the optimisation problem with a gradient-based algorithm. On the other hand, there is the Virtual Reference Feedback Tuning (VRFT) [9] which is a direct method that do not need iterations to solve the optimisation problem. This method was initially developed to SISO systems and after there were some extensions to the MIMO case. In this work we present a more complete framework for the VRFT which copes both with SISO and MIMO problems with the same formulation and is able to find the minimum of (7) for a broader class of problems.

3. State of the Art - VRFT

The VRFT method is a one-shot data based method, that is, with one batch of data, the method searches for a controller that makes the closed-loop system as close as possible to the reference model. The method was introduced in [9], and some extensions were presented in [19, 20, 21, 11, 22, 23, 24] for SISO systems and in [15, 17, 18, 25, 26, 27] for MIMO systems.

The main idea of the method is to find the minimum of $J^{MR}(\rho)$ criterion without the knowledge of the process model and without using iterative algorithms. The user defines the reference model $T_d(q)$ and the controller structure, then the controller parameters are found through a least squares minimization. We briefly explain the characteristics of the method, for SISO and MIMO cases.

3.1. VRFT SISO

Consider the noise free case, that is $v(t) = 0$ in (1), and the single-input, single-output case, that is $n = 1$.Through either an open-loop or a closed-loop experiment, input data $u(t)$ and output data $y(t)$ are collected on the process. Given the measured $y(t)$, the virtual reference signal $\bar{r}(t)$ is defined such that $T_d(q)\bar{r}(t) = y(t)$, and the virtual error is given by $\bar{e}(t) = \bar{r}(t) - y(t)$.

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

$$\begin{aligned} J^{VR}_{SISO}(P) &= \sum_{t=1}^{N} \{L(q)[u(t) - C(q, P)\bar{e}(t)]\}^2, \qquad (10)\\ &= \sum_{t=1}^{N} \left\{L(q)[u(t) - C(q, P)(T_d(q)^{-1} - 1)y(t)]\right\}^2 \end{aligned}$$

where $L(q)$ is a filter used to approximate the minima of $J^{VR}_{SISO}(P)$ and $J^{MR}_{SISO}(P)$ when Assumption 1 is not satisfied. If the controller is linearly parametrized, $J^{VR}_{SISO}(P)$ is quadratic and can be easily minimized, which is one of the main advantages over other data-driven methods.

In order to analise the properties of $J^{VR}_{SISO}(P)$, when $N \to \infty$ Parseval's theorem can be applied to (10) yielding [28]

$$\begin{aligned} J^{VR}_{SISO}(P) = \frac{1}{2\pi} \int_{-\pi}^{\pi} |L(e^{J\omega})|^2 \frac{|G(e^{J\omega})|^2 |S_d(e^{J\omega})|^2}{|T_d(e^{J\omega})|^2} \\ \times |C_d(e^{J\omega}) - C(e^{J\omega}, P)|^2 \Phi_u(e^{J\omega}) d\omega, \qquad (11) \end{aligned}$$

where $\Phi_u(e^{J\omega})$ is the spectrum of the applied signal $u(t)$. From (11) it is easy to see that in spite of $J^{VR}_{SISO}(P)$ be different from $J^{MR}(P)$, when Assumption 1 is respected, $J^{VR}_{SISO}(P)$ is minimized by $C_d(q)$, which is the minimum of $J^{MR}(P)$. Notice that in the absence of noise, the minimum of both cost function evaluate to zero when Assumption 1 is satisfied. In this case, the filter $L(q)$ does not change the minimum of the cost function, it only shapes its form. Therefore, the SISO VRFT method solves an easier (convex) optimization problem, without the knowledge of the process model $G(q)$, and achieves the same controller than MR optimization.

3.1.1. Filter design

When Assumption 1 is not satisfied, the filter $L(q)$ is designed such that the minimum of $J^{VR}_{SISO}(P)$ becomes closer to the minimum of $J^{MR}_{SISO}(P)$ [9, 28]. Applying Parseval's Theorem to $J^{MR}_{SISO}(P)$ and comparing its expression to (11), it can be seen that there is a choice for the filter so that both functions are alike. This filter is given by

$$|L(e^{J\omega})|^2 = |T_d(e^{J\omega})|^2 |S(e^{J\omega}, P)|^2 \frac{\Phi_r(e^{J\omega})}{\Phi_u(e^{J\omega})}, \quad \forall \omega \in [-\pi, \pi] \tag{12}$$

where $\Phi_r(e^{j\omega})$ is the spectrum of the reference signal $r(t)$ we want to apply to the closed-loop system. If both functions are alike, so are their minimum. However, since $S(q,P)$ is unknown, the filter is approximated by

$$|L(e^{j\omega})|^2 = |T_d(e^{j\omega})|^2|1-T_d(e^{j\omega})|^2\frac{\Phi_r(e^{j\omega})}{\Phi_u(e^{j\omega})},\ \ \forall\omega\in[-\pi,\pi] \quad (13)$$

where the approximation $|S(e^{j\omega},\rho)|^2 \approx |S_d(e^{j\omega})|^2$ was made. If the chosen controller class C is not far from the ideal controller class, then this approximation is valid, and the filter will approximate the minima. Besides that, when the data is collected in open loop, and the applied signal $u(t)$ is the same type as the reference signal $r(t)$ is usually applied on the process, than $\frac{\Phi_r(e^{j\omega})}{\Phi_u(e^{j\omega})} = 1$, and the filter is only dependent on $T_d(q)$, which is known.

3.1.2. Dealing with noise

All the formulation of the VRFT method considers the ideal situation where signals are noise-free. When data are corrupted by noise, the minimization of (10) is done using an instrumental variable [9, 28], in order to cope with the bias error introduced by the noise in $\bar{e}(t)$, the input signal of the identification problem.

3.2. MIMO case

In [15] the VRFT method was extended to MIMO processes and the works [17, 18] presented a deeper investigation of the method and its application to a diesel engine. An application to a boiler plant is presented in [16] while [25] presents its application to the wastewater treatment control. The method's formulation considers again the noise free case, that is $v(t) = 0$ in (1), and that the system has the same number of inputs and outputs.

In the related works ([15, 17, 18]), the parameters of the controller are chosen as the solution of an optimization problem where the following objective criterion is minimized:

$$J^{VR}(P) = \sum_{t=1}^{N} ||F(q)u(t) - C(q,P)(T_d^{-1}(q) - I)F(q)y(t)||_2^2. \quad (14)$$

If the controller is linearly parametrized, $J^{VR}(P)$ is convex, and therefore the optimization problem can be easily solved, as in the SISO case.

However, notice that (14) is the MIMO version of (10) only in the case where $F(q)$ can comute in the expression $C(q,P)(T_d^{-1}(q)-I)F(q)$. Since all the terms in the expression are matrices, the filter can comute only if its given by $F(q) = f(q)I$, that is, a SISO transfer function multiplied by the identity matrix. If the filter can not comute, then even when Assumption 1 is satisfied, the minimum of $J^{VR}(P)$ is not $C_d(q)$.

Again, in all works, the authors set $F(q) = T_d(q)$, so (14) becomes

$$J^{VR}(P) = \sum_{t=1}^{N} ||T_d(q)u(t) - C(q,P)(I - T_d(q))y(t)||_2^2. \quad (15)$$

By setting the filter equal to $T_d(q)$, the restriction on filter $F(q)$ is now a restriction on $T_d(q)$, which means that the reference model should be

$$T_d(q) = \begin{bmatrix} T_1(q) & 0 & \cdots & 0 \\ 0 & T_1(q) & & 0 \\ \vdots & & \ddots & \\ 0 & 0 & & T_1(q) \end{bmatrix}. \quad (16)$$

So when Assumption 1 is satisfied and $T_d(q)$ is as in (16), the minimum of the objective function $J^{VR}(P)$ corresponds to the ideal controller $C_d(q)$. If $T_d(q)$ is not as in (16), then it is not possible to obtain exactly the desired response, as it occurs when applying VRFT SISO (in the noise free case). We show that in a simple example. Consider a process given by

$$G(q) = \begin{bmatrix} \frac{0.09516}{q-0.9048} & \frac{0.03807}{q-0.9048} \\ \frac{-0.02974}{q-0.9048} & \frac{0.04758}{q-0.9048} \end{bmatrix}, \quad (17)$$

which is the same process used in [17], with sampling time of 1 second. Consider also a closed-loop model given by

$$T_d(q) = \begin{bmatrix} \frac{1-a_1}{q-a_1} & 0 \\ 0 & \frac{1-a_2}{q-a_2} \end{bmatrix}. \quad (18)$$

In this case, the ideal controller $C_d(q)$ is a full controller where each subcontroller is a PI controller. In order to respect Assumption 1, we choose C a full controller PI class, where the parameters to be estimated are K_p and K_i of each subcontroller. To obtain input and output data from the process, we perform an open-loop experiment, collecting 600 samples of each signal: a unitary step is applied to the first input in the beginning of the experiment and after 300 samples a unitary step is applied to the second input.

We minimize (15) for two different options of $T_d(q)$:

1. $T_{d1}(q)$ is given by (18), where $a_1 = a_2 = 0.9$;
2. $T_{d2}(q)$ is given by (18), where $a_1 = 0.9$ and $a_2 = 0.6$.

Table 1 compares the obtained results with the ideal controller for each case. When $T_d(q)$ is chosen as (16) – $T_{d1}(q)$ in the example – the literature method is able to find the ideal controller since $C_{d1}(q) = C(q,P_1)$. However in the case where the reference model is not diagonal with all elements alike – $T_{d2}(q)$ in the example – the method is not able to find the ideal controller ($C_{d2}(q) \neq C(q,P_2)$), even when Assumption 1 is satisfied. Notice that the bias is more significant in the decouplers, which means that, in this case, the closed-loop response will be different from the desired response especially in the disturbance between the loops.

In several industrial problems, the settling time of each loop is very different and to force the desired closed-loop performance to be the same for all loops is undesirable. In this work, we will develop a similar method that even when the user chooses *any* reference model $T_d(q)$, it is still possible to obtain $C_d(q)$, considering that Assumption 1 is satisfied. Besides that, we also obtain a filter to approximate the minima of $J^{MR}(P)$ and $J^{VR}(P)$ when Assumption 1 is not satisfied, based on the SISO approach.

Table 1: Control parameters obtained with two different choices for $T_d(q)$.

	$T_{d1}(q)$		$T_{d2}(q)$	
P	$C_{d1}(q)$	$C(q,\hat{P}_1)$	$C_{d2}(q)$	$C(q,\hat{P}_2)$
K_{p11}	0.7606	0.7606	0.7606	0.7613
K_{i11}	0.08	0.08	0.08	0.08
K_{p12}	-0.6086	-0.6086	-2.434	-0.468
K_{i12}	-0.064	-0.064	-0.256	-0.256
K_{p21}	0.4754	0.4754	0.475	0.899
K_{i21}	0.05	0.05	0.05	0.049
K_{p22}	1.521	1.521	6.085	5.975
K_{i22}	0.16	0.16	0.64	0.632

As in the SISO case, the formulation of the MIMO VRFT considers that the signals are noise free. When they are corrupted with noise, the authors in [17, 18] suggest the use of an instrumental variable to cope with the bias problem caused by the noise.

4. Unbiased MIMO VRFT

In this work we propose the use of a new objective criterion to be used with the MIMO VRFT. Instead of criterion (15) described in [15], we propose the following optimisation problem inspired in the SISO VRFT:

$$\min_P J^{VRF}(P) \tag{19}$$

$$J^{VRF}(P) = \sum_{t=1}^{N} \|F(q)[u(t) - C(q,P)\bar{e}(t)]\|_2^2. \tag{20}$$

where now $u(t)$ and $\bar{e}(t)$ are vectors, $C(q,P)$ is the controller matrix and $F(q)$ is a filter that can be used as an additional degree of freedom by the user.

An important property of $J^{VRF}(P)$ is that, when Assumption 1 is respected, the ideal controller $C_d(q)$ is the minimum of $J^{VRF}(P)$, no matter which filter $F(q)$ is chosen. If we substitute the virtual error $\bar{e}(t)$ in $J^{VRF}(P)$ and consider that the noise is null, the objective criterion becomes

$$\begin{aligned} J^{VRF}(P) &= \sum_{t=1}^{N} \|F(q)[u(t) - C(q,P)(T_d^{-1}(q) - I)G(q)u(t)]\|_2^2, \\ &= \sum_{t=1}^{N} \|F(q)[I - C(q,P)(T_d^{-1}(q) - I)G(q)]u(t)\|_2^2. \end{aligned}$$

Substituting $T_d^{-1}(q) = [(G(q)C_d(q) + I)^{-1}G(q)C_d(q)]^{-1} = I + C_d^{-1}(q)G^{-1}(q)$ the criterion is then written as

$$J^{VRF}(P) = \sum_{t=1}^{N} \|F(q)[C_d(q) - C(q,P)]C_d^{-1}(q)u(t)\|_2^2, \tag{21}$$

where it is clear that if $C(q,\hat{P}) = C(q,P_0) = C_d(q)$ then $J^{VRF}(P)$ evaluates to zero, so the parameter vector P_0 is the minimum of the objective criterion for any chosen filter $F(q)$. Thus, the proposed optimisation problem (19) results in an unbiased estimate of P_0 when Assumption 1 is respected and there is no noise, that is, $BIAS = 0$.

Observe that if $C(q,P)$ is linearly parametrized then $J^{VRF}(P)$ is quadratic on the parameters and a closed solution exists to the optimization problem, which will be described in the sequence. Another advantage of the proposed cost function (20) is that the structure of the criterion is the same of the SISO VRFT, so that they are exactly the same in the case of a SISO process. By doing this, the proposed method can be classified as an extension of the VRFT to the MIMO case that preserves all the properties of the SISO VRFT when the process is SISO, but can also be used to MIMO process and results in unbiased estimates to the minimum of criterion (6) when Assumption 1 is respected and the signals are noise free.

In the sequence we present how to deal with the cases where Assumption 1 is not respected and when there is noise affecting data. But before that, let us present a closed solution to the problem (19).

Theorem 1. *The solution of the optimisation problem (19) is P_F where*

$$\hat{P}_F = \begin{bmatrix} \rho_{11} \\ \rho_{12} \\ \vdots \\ \rho_{1n} \\ \rho_{21} \\ \rho_{22} \\ \vdots \\ \rho_{2n} \\ \vdots \\ \rho_{n1} \\ \rho_{n2} \\ \vdots \\ \rho_{nn} \end{bmatrix} = \left(\sum_{t=1}^{N} \varphi(t)\varphi^T(t) \right)^{-1} \sum_{t=1}^{N} \varphi(t)w(t), \tag{22}$$

and

$$w(t) = F(q)u(t), \quad \varphi(t) = [A_1\ A_2\ \cdots\ A_n],$$

$$A_x = \begin{bmatrix} F_{x1}E_x(t) \\ F_{x2}E_x(t) \\ \vdots \\ F_{xn}E_x(t) \end{bmatrix}, \quad E_x(t) = \begin{bmatrix} \bar{C}_{x1}(q)\bar{e}_1(t) \\ \bar{C}_{x2}(q)\bar{e}_2(t) \\ \vdots \\ \bar{C}_{xn}(q)\bar{e}_n(t) \end{bmatrix} \tag{23}$$

for $x = 1, 2, \cdots, n$.

Proof: See the Appendix. ■

4.1. Filter design

As previously stated, when Assumption 1 is satisfied, $\min J^{VRF}(P) = \min J^{MR}(P)$ independently of the filter choice. However, Assumption 1 is rarely satisfied, and in the case it is not, the filter should be chosen to approximate the minima of both cost functions, as done in the SISO case, to reduce the bias.

To do so, consider the MR criterion,

$$J^{MR}(P) = \sum_{t=1}^{N} \|[S_d(q)G(q)C_d(q) - G(q)C(q,P)S(q,P)]r(t)\|_2^2,$$
$$= \sum_{t=1}^{N} \|S_d(q)[G(q)C_d(q) - S_d^{-1}(q)G(q)C(q,P)S(q,P)]r(t)\|_2^2.$$

Using $S_d^{-1}(q) = G(q)C_d(q) + I$ the cost criterion is then written as

$$J^{MR}(P) = \sum_{t=1}^{N} \|S_d(q)[G(q)C_d(q)S(q,P) - G(q)C(q,P)S(q,P))]r(t)\|_2^2$$
$$= \sum_{t=1}^{N} \|S_d(q)G(q)[C_d(q) - C(q,P)]S(q,P)r(t)\|_2^2. \quad (24)$$

Comparing (21) to (24) it is easy to see that if we could use the optimal filter

$$F(q) = S_d(q)G(q) \quad (25)$$

and could apply the signal

$$u(t) = C_d(q)S(q,P)r(t) \quad (26)$$

to obtain experimental data, then minimizing (21) would result in the same controller as if minimizing (24).

However, observe that the filter (25) depends on the process model, which is unavailable to the user. Of course, one can always try to identify the process model and use it on the expression, but the procedure is not attractive to a data-driven approach. Also, (26) depends on $C_d(q)$ and $S(q,P)$, which are also unknown. So, in the next subsection we develop a practical choice for the filter, which approximates some unknown information and finally does not depend on the process model.

4.1.1. Practical choice

Observe that $J^{VRF}(P)$ (21) can be rewritten as

$$J^{VRF}(P) = \sum_{t=1}^{N} \|F(q)C_d[I - C_d^{-1}(q)C(q,P)]C_d^{-1}(q)u(t)\|_2^2, \quad (27)$$

and that $J^{MR}(P)$ can be written as

$$J^{MR}(P) = \sum_{t=1}^{N} \|T_d(q)[I - C_d^{-1}(q)C(q,P)]S(q,P)r(t)\|_2^2. \quad (28)$$

Suppose we could commute the term $[I - C_d^{-1}(q)C(q,P)]$ in the cost functions above, the same procedure proposed for MIMO IFT [12]. In this case, $J^{VRF}(P)$ would be given by

$$\tilde{J}^{VRF}(P) = \sum_{t=1}^{N} \|[I - C_d^{-1}(q)C(q,P)]F(q)u(t)\|_2^2,$$

and $J^{MR}(P)$ would be given by

$$\tilde{J}^{MR}(P) = \sum_{t=1}^{N} \|[I - C_d^{-1}(q)C(q,P)]T_d(q)S(q,P)r(t)\|_2^2.$$

When $N \to \infty$, both criteria can be expressed by their frequency domain expressions using Parseval's Theorem. In this case,

$$\tilde{J}^{VRF}(P) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \mathrm{Tr}\left\{[(I - C_d^{-1}C(P))F]\Phi_u^{1/2} \times \Phi_u^{1/2}[(I - C_d^{-1}C(P))F]^H\right\} d\omega, \quad (29)$$

and

$$\tilde{J}^{MR}(P) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \mathrm{Tr}\left\{[(I - C_d^{-1}C(P))T_d S(P)]\Phi_r^{1/2} \times \Phi_r^{1/2}[(I - C_d^{-1}C(P))T_d S(P)]^H\right\} d\omega, \quad (30)$$

where Tr is the trace operator, the superscript H indicates the hermitian conjugate of a complex expression, Φ_r and Φ_u are the power spectrum of $r(t)$ and $u(t)$ respectively, and $\Phi_x^{1/2}$ denotes a spectral factor of Φ_x.

Let the filter be chosen as

$$F(e^{j\omega}) = T_d(e^{j\omega})S(e^{j\omega},P)\Phi_r^{1/2}(\omega)\Phi_u^{-1/2}(\omega), \quad \forall\omega \in [-\pi,\pi], \quad (31)$$

then $\tilde{J}^{VRF}(P) = \tilde{J}^{MR}(P)$. Notice that the above filter is very similar to the SISO VRFT filter (12), and that it depends on the unknown quantity $S(e^{j\omega},P)$. Now, if we approximate $S(e^{j\omega},P) \approx S_d(e^{j\omega})$, the same procedure of the SISO VRFT, then

$$F(e^{j\omega}) = T_d(e^{j\omega})(I - T_d(e^{j\omega}))\Phi_r^{1/2}(\omega)\Phi_u^{-1/2}(\omega), \quad \forall\omega \in [-\pi,\pi], \quad (32)$$

which is an implementable filter and equivalent to the filter used in the SISO case (see (13)). So, we recommend the use of the filter (32) when Assumption 1 is not satisfied to approximate the minimum $J^{VRF}(P)$ to the minimum of $J^{MR}(P)$. Observe that the estimate will be biased (compared to the MR controller) since the proposed practical filter is developed using some approximations. In order to obtain unbiased estimates, one should use the optimal filter (25)-(26), which depend on the unknown quantities. We will demonstrate the use of the proposed practical filter on Section 5.

4.2. Dealing with noise

Under Assumption 1 and considering that the process noise is zero, the solution (22) is such that $C(z,P_F)$ is equal to the ideal controller $C_d(q)$. In the practical situation, where the signals are corrupted by noise, the solution (22) is biased even in the case when Assumption 1 is satisfied, and we propose the use of an instrumental variable technique to cope with the bias[1]. In order to do so, an extra signal $y'(t)$, which is correlated to $y(t)$ but uncorrelated to the noise present in the output, should be obtained. Thus, the *instrumental variable* $\zeta(t)$ is defined as

$$\zeta(t) = [A'_1 \;\; A'_2 \cdots A'_n] \quad (33)$$

[1]Notice that if the ideal controller is not in the controller class, the bias will only be reduced, but not eliminated.

where $A'_x, x = 1, 2, \cdots, n$ are similar to A_x (from $\varphi(t)$), but formed with the signal $y'(t)$ instead of $y(t)$.

The solution of the optimization problem using the instrumental variable is

$$\hat{P}_{IV} = \left(\sum_{t=1}^{N} \zeta(t)\varphi(t)^T \right)^{-1} \sum_{t=1}^{N} \zeta(t)w(t). \tag{34}$$

Observe that again the proposed approach to cope with the bias introduced by the noise is very similar to the approach used in the SISO VRFT. There are several choices to the extra signal $y'(t)$ and we recommend the SISO VRFT articles for more information [19, 20, 21, 11].

5. Application to Process Control

In order to illustrate the improvements of the VRFT method proposed in this work for multivariable systems, a numerical example and a practical application are presented. The goal is to show that using the proposed method, the VRFT can be used to tune multivariable controllers for a broader class of systems, where the performance requirements can be chosen differently for each output. We show that the proposed method yields the ideal controller when Assumption 1 is satisfied, and that when it is not, a proper filter can be used to enhance the results. We also show the applicability of the method to a pilot plant.

5.1. Numerical example

The process considered in the example is a linear time-invariant discrete-time process described by

$$G(q) = \begin{bmatrix} \frac{0.095q}{(q-0.92)(q-0.8)} & \frac{0.04q}{(q-0.9)(q-0.85)} \\ \frac{-0.03q}{(q-0.92)(q-0.8)} & \frac{0.05q}{(q-0.9)(q-0.85)} \end{bmatrix}, \tag{35}$$

and the closed-loop reference model is chosen as

$$T_d(q) = \begin{bmatrix} \frac{0.2}{q-0.8} & 0 \\ 0 & \frac{0.4}{q-0.6} \end{bmatrix}, \tag{36}$$

both with sampling time of 1 second. Observe that this reference model represents different performance requirements for each output. The literature method is not able to find the ideal controller for this choice. Even though, we use the literature method in order to visualize the improvements of our formulation.

The ideal controller $C_d(q)$ which leads to perfect matching is given by

$$C_d(q) = \begin{bmatrix} \frac{1.681q^2-2.891q+1.237}{q^2-q} & \frac{-2.689q^2+4.625q-1.979}{q^2-q} \\ \frac{1.008q^2-1.765q+0.7714}{q^2-q} & \frac{6.387q^2-11.18q+4.886}{q^2-q} \end{bmatrix}, \tag{37}$$

where each subcontroller is Proportional-Integral-Derivative (PID).

The effectiveness of the methods are appreciated by calculating a cost function which represents the cost of the error between the obtained closed-loop output $y(t, P)$ and the reference model response $y_d(t)$, which, for a TITO (two-input two-output) system, is given by:

$$J(P) = \sum_{t=1}^{N} \left\| \begin{bmatrix} y_1(t,P) \\ y_2(t,P) \end{bmatrix} - \begin{bmatrix} y_{d1}(t) \\ y_{d2}(t) \end{bmatrix} \right\|_2^2 .$$

In this case, the closed-loop collected signals are obtained applying reference signals of 200 samples to both inputs. In the first input, a unitary step is applied on sample 10 and in the second input, the unitary step is applied on sample 100.

In this example, we consider the noise free case and explore the choice of the filter $F(q)$ to approximate the minima of $J^{VRF}(P)$ and $J^{MR}(P)$ when Assumption 1 is not satisfied. To do so, we compare the results obtained with:

- *Design 1:* the literature method ($J_1(\hat{P})$), where (15) is minimized;
- *Design 2:* the proposed method with a designed input signal $u(t) = C_d(q)S_d(q)r(t)$ (see (26)) and the optimal filter (25) ($J_2(\hat{P})$);
- *Design 3:* the proposed method with the filter that depends only on known quantities (32) ($J_3(\hat{P})$).

The proposed method with the designed input signal and the optimal filter is presented only to show the consequences of commuting terms in the cost function, that is, the use of filter (32) in *Design 3*. For that reason, we use the exact model of $G(q)$ and $C_d(q)$ to compute the filter and also approximate $S(q, P) \approx S_d(q)$ to compute the input signals in *Design 2*. The filter used in the literature method is $T_d(q)$, but notice that it is used to filter the input and output signals, differently from the proposed method in this work.

For *Design 1* and *Design 3* we perform open-loop experiments of 200 samples. In the first input, a unitary step is applied on sample 10 and in the second input, the unitary step is applied on sample 100 (the same signals used to compute the resulting cost function). In *Design 2*, these signals are filtered by $C_d(q)S_d(q)$ and then applied as the input of the process. We perform these three designs for different controller structures and compute the obtained cost. The results are shown in Table 2.

Table 2: Calculated costs of closed-loop systems with different controllers obtained using the literature method and the proposed method.

Controller	$J_1(\hat{P})$	$J_2(\hat{P})$	$J_3(\hat{P})$
[PID PID; PID PID]	0.1097	0	0
[PID PID; PI PID]	0.1225	0.0110	0.0220
[PID PI; PI PID]	0.2704	0.1822	0.2061
[PI PI; PI PI]	1.4153	1.0602	1.1053
[PID 0; 0 PID]	1.9354	1.5141	1.2949
[PI 0; 0 PI]	6.1916	4.5490	4.8738

First, our methodology has presented better results than the literature one, in all the cases presented in Table 2. In the first example, the controller class (all subcontrollers are PID) is such that Assumption 1 is satisfied, and our methodology is able to find the ideal controller, differently from the literature. In the subsequent controllers, Assumption 1 is no longer satisfied, and we present cases where it goes from "almost satisfied" to "far from satisfied". Remember that, in all proposed filters, the approximation $S(q,P) \approx S_d(q)$ was done, which is not valid when the ideal controller is far from the chosen controller class. In all cases presented in Table 2, using the "Practical choice" for the filter have resulted in costs that are not far from the optimal choice. Notice that we have commuted $[I - C_d^{-1}(q)C(q,P)]$ in equations (27) and (28) to obtain an implementable filter. Observe, by the costs $J_1(\hat{P})$ and $J_3(\hat{P})$, that commuting terms only to obtain and implementable filter (Design 3) is less sensitive than commuting terms to obtain a different cost function. Notice also that a decentralized PI controller is not adequate to control this system and its response is deteriorated, since the costs are much larger than the ones presented with other controller structures.

Fig. 1 presents the closed-loop step responses with the controllers obtained from the literature method and the proposed method for the second case presented in Table 2, that is, the ideal controller is close to the controller class chosen. The literature method yields the controller

$$C_1(q,\hat{P}) = \begin{bmatrix} \frac{1.6807(q-0.92)(q-0.8)}{q(q-1)} & \frac{-1.3445(q-0.92)(q-0.6)}{q(q-1)} \\ \frac{0.28001(q-0.9467)}{q(q-1)} & \frac{6.1259(q-0.9089)(q-0.828)}{q(q-1)} \end{bmatrix},$$

while the proposed method with the practical filter results in

$$C_3(q,\hat{P}_F) = \begin{bmatrix} \frac{1.6807(q-0.92)(q-0.8)}{q(q-1)} & \frac{-2.6891(q-0.92)(q-0.8)}{q(q-1)} \\ \frac{0.26312(q-0.9698)}{q(q-1)} & \frac{6.0758(q-0.9135)(q-0.823)}{q(q-1)} \end{bmatrix}.$$

Notice that the proposed method presents better decoupling, which has resulted in a lower cost when comparing the response with the desired one. The closed-loop responses with a decentralized PID controller is presented in Fig. 2, for the literature method and the proposed one. Notice that again the proposed method yields better results. There is no decoupling, but the disturbance from one output variable in the other one is smaller. Besides, when there is a step change in one reference, the related output is closer to the reference model response for the proposed method, compared to the response obtained with the literature method. In this case, the literature method yields the controller

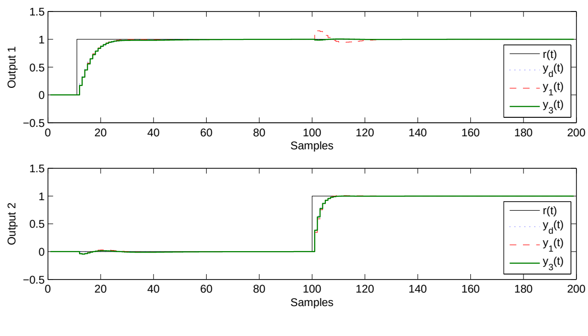

Figure 1: Output responses obtained with the design of a full controller compared with the desired response.

$$C_1(q,\hat{P}) = \begin{bmatrix} \frac{1.5057(q-0.9574)(q-0.6072)}{q(q-1)} & 0 \\ 0 & \frac{6.4516(q-0.966)(q-0.5956)}{q(q-1)} \end{bmatrix},$$

while the proposed method with the practical filter results in

$$C_3(q,\hat{P}_F) = \begin{bmatrix} \frac{1.7514(q-0.919)(q-0.8038)}{q(q-1)} & 0 \\ 0 & \frac{5.9889(q-0.9065)(q-0.8376)}{q(q-1)} \end{bmatrix}.$$

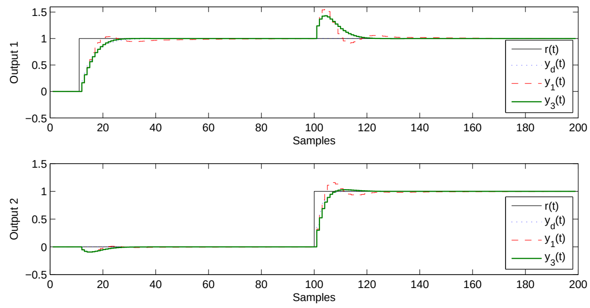

Figure 2: Output responses obtained with the design of decentralized controller compared with the desired response.

5.2. Practical application

We have also applied the proposed methodology to design controllers of a pilot plant, where the goal is to control the level of two tanks in a three tank plant. The same plant was used in [28], where a SISO controller was obtained using the SISO VRFT method considering the flow control of one tank. The schematic diagram in Fig. 3 describes the main parts of the process, considering a multivariable approach. The whole process is built with of-the-shelf industrial equipments (pumps, valves, sensors and tanks). The equipments in the process are intelligent, since they are connected through the Foundation Fieldbus protocol [29]. Tanks 1 and 2 are 70 liters each, while tank 3 is a 250 liters container.

The water is pumped up from Tank 3 to Tank 2 through Valve 1, from Tank 1 to Tank 2 through Valve 2 and back to Tank 3 by gravity. The liquid level of Tank 1 is $y_1(t)$ and the opening of the Valve 1 is the manipulated variable $u_1(t)$. Accordingly, the liquid level of Tank 2 is $y_2(t)$ and the opening of the Valve 2 is the manipulated variable $u_2(t)$.

The objective of the control system is to control the level of tanks 1 and 2 through the application of the MIMO VRFT proposed in this work. In order to do that, an open-loop experiment was performed, applying pseudo-random binary signals (PRBS) in both inputs, for 8000 s, where the sampling time was $T_s = 1$ s. The input signals are presented in Fig. 4 while the output signals are presented in Fig. 5.

After collecting process data, we have to choose the controller class and the reference model. Since we can only design PI controllers, which is a constraint imposed by the system, we choose a full controller, where all subcontrollers are PI. The reference model was chosen to obtain a closed-loop response that

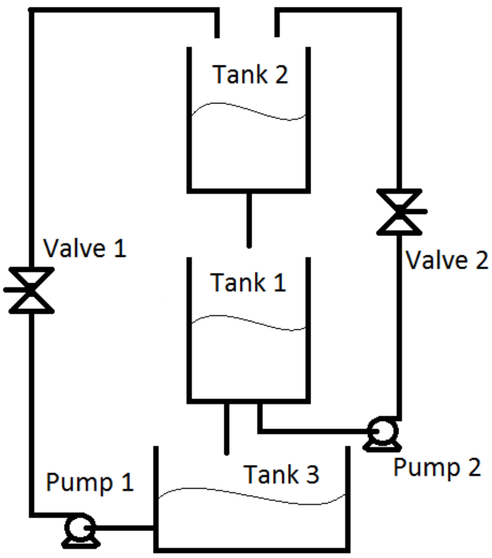

Figure 3: Schematic diagram of the pilot plant.

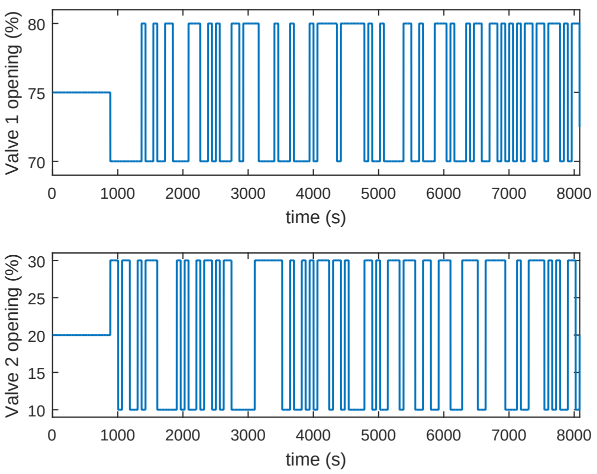

Figure 4: Input signals applied to the pilot plant.

is faster than the open-loop one. From a past open-loop experiment, we know that the open-loop settling time for Tank 1 is 900 s and for Tank 2 is 700 s. Based on that, we chose

$$T_d(q) = \begin{bmatrix} \frac{0.03}{q-0.97} & 0 \\ 0 & \frac{0.02}{q-0.98} \end{bmatrix}, \tag{38}$$

which represents performances with zero steady-state error and settling time of 128 s for the first output and 193 s for the second output. As in the practical case Assumption 1 is rarely satisfied, we have used the practical filter proposed in Section 4.1.1. Besides, since the output noise is not significant, we have not used instrumental variables in the estimate of the controller. The obtained controller is

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

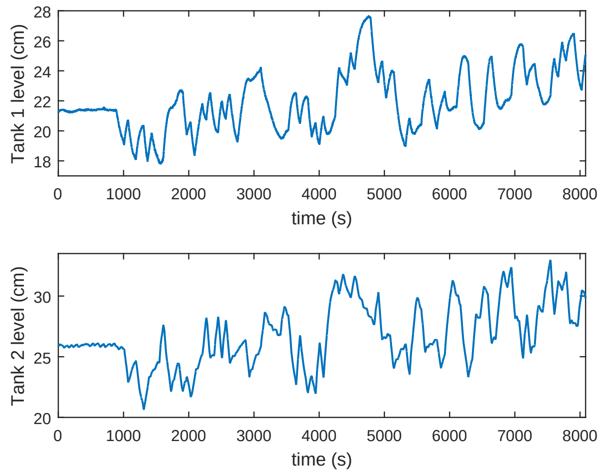

Figure 5: Output signals of the pilot plant obtained with input signals presented in Fig 4.

which results in the closed-loop performance presented in Fig. 6, with control signals presented in Fig. 7. Notice that even though we have chosen to tune PI subcontrollers, in $C_{12}(q, \rho_{12})$ the estimated zero canceled the integrator, and the controller became only a Proportional controller. 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. The obtained cost was $J(\hat{P}) = 31.64$ cm^2, which was calculated based on the signals presented in Fig. 6.

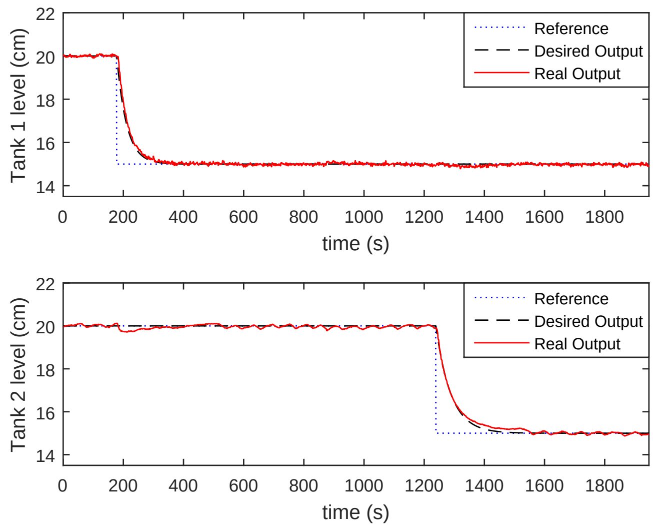

Figure 6: Closed-loop response obtained with the estimated controller $C(q, P_F)$ compared to the reference model response.

6. Conclusion

In this work we have presented an extension of the VRFT methods to MIMO systems, which presents the same characteristics of the SISO VRFT. The method presented here estimates

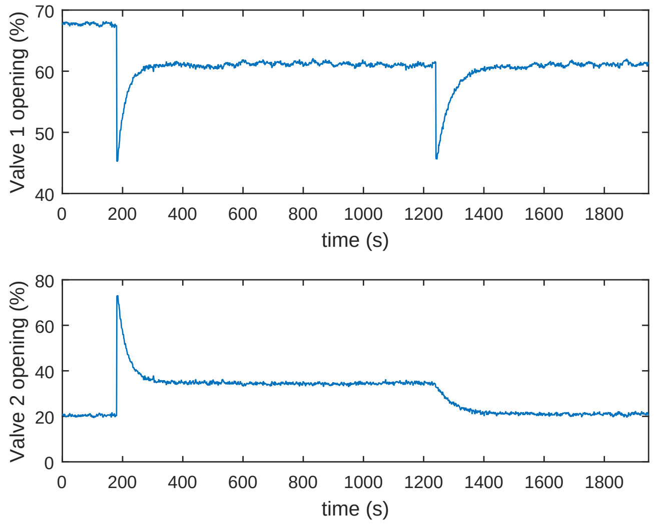

Figure 7: Control signals of the closed-loop system with the estimated controller $C(q, P_F)$.

the ideal controller if it belongs to the controller class chosen for any choice of the the reference model, different from previous methods presented in the literature. In the non ideal case, that is, when the ideal controller does not belong to the chosen controller class, a filter should be used to approximate te minima of the VRFT cost function to the MR cost function. The filter suggested in this work is equivalent to the SISO filter, and can be used even when the ideal controller belongs to the controller class. Besides that, equivalent to the SISO design, if the signals are corrupted with noise, an instrumental variable should be used.

A simulated example shows that when the ideal controller is in the controller class, its estimation is unbiased through the proposed method. It also shows that the results obtained with the proposed practical filter is not far from what could be obtained with the optimal filter, and it is suitable to approximate the minima of $J^{VRF}(P)$ to $J^{MR}(P)$. A practical application shows the applicability of the method to real processes.

Appendix A. Proof of theorem 1

In the MIMO case, the structure of the controller (3) can be defined as:

$$C(q,P) = \begin{bmatrix} \rho_{11}^T\bar{C}_{11}(q) & \rho_{12}^T\bar{C}_{12}(q) & \cdots & \rho_{1n}^T\bar{C}_{1n}(q) \\ \rho_{21}^T\bar{C}_{21}(q) & \rho_{22}^T\bar{C}_{22}(q) & \cdots & \rho_{2n}^T\bar{C}_{2n}(q) \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{n1}^T\bar{C}_{n1}(q) & \rho_{n2}^T\bar{C}_{n2}(q) & \cdots & \rho_{nn}^T\bar{C}_{nn}(q) \end{bmatrix}, \quad \text{(A.1)}$$

where P is the controller parameter vector to be calculated and $\bar{C}_{xy}$ are column vectors of fixed causal rational functions.

The filter is defined as

$$F(q) = \begin{bmatrix} F_{11}(q) & F_{12}(q) & \cdots & F_{1n}(q) \\ F_{21}(q) & F_{22}(q) & \cdots & F_{2n}(q) \\ \vdots & \vdots & \ddots & \vdots \\ F_{n1}(q) & F_{n2}(q) & \cdots & F_{nn}(q) \end{bmatrix}. \quad \text{(A.2)}$$

The objective function (20) can then be written as,

$$J^{VRF}(P) = \sum_{t=1}^{N} \left\| \begin{bmatrix} w_1(t) \\ w_2(t) \\ \vdots \\ w_n(t) \end{bmatrix} - \begin{bmatrix} F_{11}(q) & F_{12}(q) & \cdots & F_{1n}(q) \\ F_{21}(q) & F_{22}(q) & \cdots & F_{2n}(q) \\ \vdots & \vdots & \ddots & \vdots \\ F_{n1}(q) & F_{n2}(q) & \cdots & F_{nn}(q) \end{bmatrix} \cdot \begin{bmatrix} \rho_{11}^T\bar{C}_{11}(q) & \rho_{12}^T\bar{C}_{12}(q) & \cdots & \rho_{1n}^T\bar{C}_{1n}(q) \\ \rho_{21}^T\bar{C}_{21}(q) & \rho_{22}^T\bar{C}_{22}(q) & \cdots & \rho_{2n}^T\bar{C}_{2n}(q) \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{n1}^T\bar{C}_{n1}(q) & \rho_{n2}^T\bar{C}_{n2}(q) & \cdots & \rho_{nn}^T\bar{C}_{nn}(q) \end{bmatrix} \begin{bmatrix} \bar{e}_1(t) \\ \bar{e}_2(t) \\ \vdots \\ \bar{e}_n(t) \end{bmatrix} \right\|_2^2 \quad \text{(A.3)}$$

where $\bar{e}(t) = (T_d^{-1}(q) - I)y(t)$.

The minimum of (A.3) will occur when the gradient of the objective function is zero, that is

$$\nabla J^{VRF}(P) = \begin{bmatrix} \frac{\partial J(P)}{\partial \rho_{11}} \\ \frac{\partial J(P)}{\partial \rho_{12}} \\ \vdots \\ \frac{\partial J(P)}{\partial \rho_{1n}} \\ \vdots \\ \frac{\partial J(P)}{\partial \rho_{21}} \\ \frac{\partial J(P)}{\partial \rho_{22}} \\ \vdots \\ \frac{\partial J(P)}{\partial \rho_{2n}} \\ \vdots \\ \frac{\partial J(P)}{\partial \rho_{n1}} \\ \frac{\partial J(P)}{\partial \rho_{n2}} \\ \vdots \\ \frac{\partial J(P)}{\partial \rho_{nn}} \end{bmatrix} = 0. \quad \text{(A.4)}$$

Observe that (A.3) is quadratic and then (A.4) is a linear system of equations:

$$\nabla J^{VRF}(P) = 2\sum_{t=1}^{N} \varphi(t)\varphi^T(t)P - 2\sum_{t=1}^{N} \varphi(t)w(t) = 0,$$

with $w(t)$ and $\varphi(t)$ defined in the theorem statement.

The solution of the system of equations (A.4) is given by (22). Q.E.D.

References

[1] L. Campestrini, L. Stevanatto Filho, A. Bazanella, Tuning of multivariable decentralized controllers through the ultimate-point method, Control Systems Technology, IEEE Transactions on 17 (2009) 1270–1281.

[2] Z. J. Palmor, Y. Halevi, N. Krasney, Automatic tuning of decentralized PID controllers for TITO processes, Automatica 31 (1995) 1001–1010.
[3] R. Dittmar, S. Gill, H. Singh, M. Darby, Robust optimization-based multi-loop PID controller tuning: A new tool and its industrial application, Control Engineering Practice 20 (2012) 355–370.
[4] S. Titouche, P. Spiteri, F. Messine, M. Aidene, Optimal control of a large thermic process, Journal of Process Control 25 (2015) 50–58.
[5] M. Gevers, A decade of progress in iterative control design: from theory to practice, Journal of Process Control 12 (2002) 519–531.
[6] H. Hjalmarsson, Iterative feedback tuning - an overview, International Journal of Adaptive Control and Signal Processing 16 (2002) 373–395.
[7] L. C. Kammer, R. R. Bitmead, P. L. Bartlett, Direct Iterative Tuning via Spectral Anaysis, Automatica 36 (2000) 1301–1307.
[8] A. Karimi, L. Miskovic, D. Bonvin, Iterative correlation-based controller tuning, International Journal of Adaptive Control and Signal Processing 18 (2004) 645–664.
[9] M. C. Campi, A. Lecchini, S. M. Savaresi, Virtual reference feedback tuning: a direct method for the design of feedback controllers, Automatica 38 (2002) 1337–1346.
[10] A. Karimi, K. van Heusden, D. Bonvin, Noniterative data-driven controller tuning using the correlation approach, in: European Control Conference 2007, pp. 5189–5195.
[11] L. Campestrini, D. Eckhard, M. Gevers, A. Bazanella, Virtual Reference Feedback Tuning for non-minimum phase plants, Automatica 47 (2011) 1778 – 1784.
[12] H. Jansson, H. Hjalmarsson, Gradient approximations in iterative feedback tuning for multivariable processes, International Journal of Adaptive Control and Signal Processing 18 (2004) 665–681.
[13] L. Miskovic, A. Karimi, D. Bonvin, M. Gevers, Correlation-based tuning of linear multivariable decoupling controllers, in: Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on, pp. 7144–7149.
[14] K. Yubai, H. Usami, J. Hirai, Correlation-based direct tuning of MIMO controllers by least-squares and its application to tension-and-speed control apparatus, in: ICCAS-SICE, 2009, pp. 931–936.
[15] M. Nakamoto, An Application of the Virtual Reference Feedback Tuning for a MIMO Process, in: SICE 2004 Annual Conference, volume 3, pp. 2208–2213.
[16] J. Rojas, F. Morilla, R. Vilanova, 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.
[17] S. Formentin, S. Savaresi, Noniterative data-driven design of multivariable controllers, in: Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on, pp. 5106–5111.
[18] S. Formentin, S. Savaresi, L. del Re, Non-iterative direct data-driven controller tuning for multivariable systems: theory and application, Control Theory Applications, IET 6 (2012) 1250–1257.
[19] M. C. Campi, A. Lecchini, S. M. Savaresi, An application of the virtual reference feedback tuning method to a benchmark problem, European Journal of Control 9 (2003) 66–76.
[20] M. C. Campi, S. M. Savaresi, Direct nonlinear control tuning design: the virtual reference feedback tuning approach, IEEE Transactions on Automatic Control 51 (2006) 14–26.
[21] A. Lecchini, M. C. Campi, S. M. Savaresi, Virtual reference feedback tuning for two degrees of freedom controllers, International Journal of Adaptive Control and Signal Processing 16 (2002) 355–371.
[22] A. Esparza, A. Sala, P. Albertos, Neural networks in virtual reference tuning, Engineering Applications of Artificial Intelligence 24 (2011) 983 – 995.
[23] Y. Ding, N. Xu, L. Ren, K. Hao, Data-driven neuroendocrine ultrashort feedback-based cooperative control system, Control Systems Technology, IEEE Transactions on 23 (2015) 1205–1212.
[24] S. Formentin, A. Karimi, S. M. Savaresi, Optimal input design for direct data-driven tuning of model-reference controllers, Automatica 49 (2013) 1874 – 1882.
[25] J. D. Rojas, X. Flores-Alsina, U. Jeppsson, R. Vilanova, Application of multivariate virtual reference feedback tuning for wastewater treatment plant control, Control Engineering Practice 20 (2012) 499 – 510.
[26] P. Yan, D. Liu, D. Wang, H. Ma, Data-driven controller design for general {MIMO} nonlinear systems via virtual reference feedback tuning and neural networks, Neurocomputing 171 (2016) 815 – 825.
[27] M. Radac, R. Precup, Optimal behaviour prediction using a primitive-based data-driven model-free iterative learning control approach, Computers in Industry 74 (2015) 95 – 109.
[28] A. S. Bazanella, L. Campestrini, D. Eckhard, Data-driven Controller Design: The H_2 Approach, Springer, Netherlands, 2012.
[29] J. M. Gomes da Silva Jr, M. A. L. Silveira, C. E. Pereira, A web-based remote multivarialbe control experiment, in: 11th IFAC Symposium on Information Control Problems in Manufacturing (INCOM'04), Salvador.