

Enhancing MPC formulations by identification and estimation of valve stiction[☆]

Riccardo Bacci di Capaci, Marco Vaccari, Claudio Scali, Gabriele Pannocchia^a

Department of Civil and Industrial Engineering, University of Pisa, Pisa, Italy

^a*Corresponding author: gabriele.pannocchia@unipi.it*

Abstract

A common source of poor control performance in industrial processes is represented by stiction in control valves, which often induces offset, oscillating behavior, and even loss of stability. Recent studies have investigated the effectiveness of embedding stiction models into model predictive controller (MPC) schemes, moving from stiction unaware to different stiction aware formulations, which help to remove fluctuations and may guarantee higher set-point tracking ability. To this aim, along with the process model the controller needs to use a dynamic model of sticky valves. This paper proposes an efficient, computational approach to obtain both valve and process dynamics, under the framework of Hammerstein system identification, which is based on nonlinear, gradient-based, numerical optimization. In order to improve the computational behavior and effectiveness of the methodology, a recently proposed smoothed model of stiction is deployed. The proposed methodology is validated in several (single-input single-output, and multivariable) examples, where the effectiveness of the obtained stiction aware MPC regulator is also evaluated against a stiction unaware counterpart.

Keywords: Model predictive control, control valves, static friction, stiction modeling and estimation, numerical optimization.

1. Introduction

Industrial control loops often suffer from performance degradation due to valve stiction, i.e. excessive static friction between the stem and packing, which causes sustained oscillations of process variables, ultimately leading to shorter actuator life and inferior overall performance of the plant [2]. These oscillations are induced by controller integral action which forces the input to the actuator to change while the actual actuator position remains stuck until this position difference becomes so large that the valve jumps. In this way, the valve typically jumps between two extreme positions, above and below the desired operating point. A partial remedy to the problem is retuning the controller, slowing down the integral action, to reduce amplitude and frequency of oscillations [3]. However, this comes at expense of slower rate of disturbance rejection and set-point tracking. Furthermore, the behavior can become sluggish again depending on the operating conditions [4].

In the context of Model Predictive Control (MPC) systems, industrial offset-free formulations (i.e. which include integral action via disturbance estimation) also show a similar oscillating behavior, unless a dynamic model of the actuator malfunction is taken into account in the controller formulation. Indeed, stiction compensation via MPC and other advanced control methods is an area of recent activity, as witnessed by other several works published in the last years [5, 6, 7, 8, 9, 10].

In two recent works [6, 10] we showed that stiction embedding MPC, in which the MPC model is augmented with valves

dynamics, is a suitable approach to obtain good set-point tracking ability and stiction compensation. Relatively good robustness to valve model errors has been shown, but performance tend to deteriorate even significantly when such errors increase. It is therefore important to have reliable and possibly non-invasive methods to estimate the sticky valve dynamics.

Pneumatic sliding stem valves are the most spread final control elements. In such devices, stiction can be described both by detailed physical models and by empirical (data-driven) models [11]. For practical purposes, only the latter are actually useful, as few parameters and relatively simple algebra are involved. The most established data-driven models have been proposed in [12], [13] and [14].

The ability of providing an estimate of stiction amount is a crucial step for shortlisting the most critical valves, scheduling valve maintenance, and also performing on-line compensation. Methods available in the literature can be broadly divided into four main categories: apparent stiction techniques [15], Hammerstein-based methods (e.g., [16, 17, 18, 19, 20]), nonlinear process model-based methods (e.g., [21, 22]), mixed approaches (e.g., [23, 24, 25]). Some techniques perform detection and quantification of valve stiction in a single stage, while other methods can be applied only once stiction is clearly detected by suitable methods.

Among others techniques, the methods proposed in [18] and [26] used global and gradient-free optimization approaches. The first one implemented genetic and path search algorithms, but, despite being quite robust, high computational resources are required. The second one proved to be an improvement as one-stage identification is performed by means of a determin-

[☆]A preliminary version of this paper has been presented in [1].

istic algorithm that is no longer dependent on the initial guess of stiction parameters, obtained via the ellipse fitting method of [15].

The objective of this paper is to propose a reliable and efficient methodology to estimate valve stiction amount, on the basis of a recently proposed smoothed model [6]. This procedure, previously defined for SISO linear systems [1], is extended to MIMO systems in this work. Moreover, in the case of SISO processes, the time-delay is also estimated.

The remainder of the work is organized as follows. Preliminaries about problem definition and valve stiction modeling are given in Section 2. The proposed identification and estimation method is described in Section 3. Several numerical examples are then presented in Section 4, to test both the identification results and the enhancements obtained in the MPC. Finally, conclusions are drawn in Section 5.

2. Problem definition

The plant under study is formed by control valves followed by the process dynamics as depicted in Figure 1. In detail, χ represents the valve output, that is, the process input; y is the process output; u is the output of the model predictive controller, and, finally, v is white Gaussian noise.

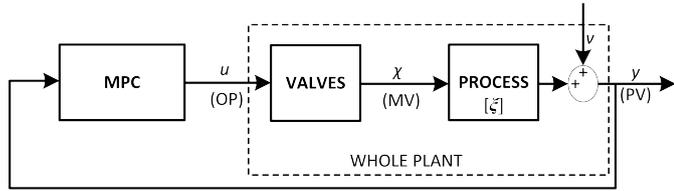


Figure 1: The closed-loop system with (sticky) control valves followed by the process.

The systems analyzed in the current work are both SISO and MIMO. The whole plant dynamics can be written as:

$$\begin{aligned} z_{k+1} &= f(z_k, u_k) \\ y_k &= h(z_k) + v_k \end{aligned} \quad (1)$$

where the valve output χ represents the first m components of the state vector of whole plant, that is, $z_k = [\chi_{k-1}, \xi_k]^T \in \mathbb{R}^{n_z}$.

When the valves stiction nonlinearity is followed by a linear process dynamics, the whole plant forms an *extended* Hammerstein structure [27]. Valves dynamics is described by a data-driven stiction model later presented, while the linear process dynamics can be expressed by ARX, ARMAX, or state-space models. In this latter case, we can write:

$$\begin{aligned} z_{k+1} &= \begin{bmatrix} \chi_k \\ \xi_{k+1} \end{bmatrix} = \begin{bmatrix} \varphi(\chi_{k-1}, u_k) \\ \mathbf{A}\xi_k + \mathbf{B}\varphi(\chi_{k-1}, u_k) \end{bmatrix} \\ y_k &= \mathbf{C}\xi_k + v_k \end{aligned} \quad (2)$$

where $\xi \in \mathbb{R}^n$ is the process state vector, $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$, $\mathbf{C} \in \mathbb{R}^{p \times n}$, being n the process model dimension, m the number of inputs (and valves) and p the number of outputs,

i.e. $u \in \mathbb{R}^m$, $\chi \in \mathbb{R}^m$, and $y \in \mathbb{R}^p$. The first m states χ of augmented state z is given by the stiction nonlinearity, expressed by the function $\varphi(\cdot): \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}^m$, later discussed. As previously explained in Section 1, the most used and established data-driven methods for valve stiction modeling adopt relatively simple algebra schemes. The standard empirical model by He and coworkers [14] and its revised version, the so-called *semi-physical* model [28], have been proved suitable to reproduce the valve response generated by physical stiction models without involving computationally intensive numerical integration. As a matter of fact, only two parameters are used: the dynamic (f_D) and the static (f_S) friction coefficients.

In these two stiction models, the transient valve dynamics is ignored, i.e. a fast response from the valve is assumed. On the other hand, a static – but with memory – nonlinear function is used, that is, only the stationary-state values of stem position are considered.

The discontinuous stiction model. He's standard data-driven model [14] can be rewritten so that the sticky valve has a nonlinear dynamics $\chi_k^{(j)} = \varphi(\chi_{k-1}^{(j)}, u_k^{(j)})$. Hence, the generic j -th valve is expressed by the following relation:

$$\chi_k^{(j)} = \begin{cases} \chi_{k-1}^{(j)} + [e_k^{(j)} - \text{sign}(e_k^{(j)})f_D^{(j)}] & \text{if } |e_k^{(j)}| > f_S^{(j)} \\ \chi_{k-1}^{(j)} & \text{if } |e_k^{(j)}| \leq f_S^{(j)} \end{cases} \quad (3)$$

where $f_S^{(j)}$ and $f_D^{(j)}$ are static and dynamic friction parameters, respectively, and $e_k^{(j)} = u_k^{(j)} - \chi_{k-1}^{(j)}$, with $j = 1, \dots, m$. Note that $e_k^{(j)}$ can be interpreted as the valve position error, while $f_S^{(j)} \geq f_D^{(j)}$ by definition. The valve dynamics (3) can be rewritten by expanding the nonlinear *sign* function, and finally by solving the inequalities, as follows:

$$\chi_k^{(j)} = \begin{cases} u_k^{(j)} - f_D^{(j)} & \text{if } u_k^{(j)} - \chi_{k-1}^{(j)} > f_S^{(j)} \\ u_k^{(j)} + f_D^{(j)} & \text{if } u_k^{(j)} - \chi_{k-1}^{(j)} < -f_S^{(j)} \\ \chi_{k-1}^{(j)} & \text{if } |u_k^{(j)} - \chi_{k-1}^{(j)}| \leq f_S^{(j)} \end{cases} \quad (4)$$

Therefore, the stiction nonlinearity $\varphi(\cdot)$ is formed by a set of three linear, mutually exclusive relations, thus constituting a sort of switching “multi-mode” model, which acts as a *discontinuous* model.

The smoothed stiction model. It is evident that the presence of such *if-else* statements implies two hard discontinuities in the input-output relation of each valve. Therefore, model (4) may affect the performance of the considered optimization problem, thus producing a bad quality parameters identification. Hence, to avoid this issue, in this work the *smoothed* stiction model introduced in [6] is used and a smoother optimization problem is obtained. Model (4) is expressly approximated by using a smoothing function $\varphi_S(\cdot)$ such that:

$$\chi_k^{(j)} = \eta_1(e_k^{(j)})\chi_{k-1}^{(j)} + (1 - \eta_1(e_k^{(j)}))u_k^{(j)} + \eta_2(e_k^{(j)})f_D^{(j)} \quad (5)$$

where $\eta_1(e_k^{(j)})$ and $\eta_2(e_k^{(j)})$ are the sum of two hyperbolic functions, defined as below:

$$\begin{aligned}\eta_1(e_k^{(j)}) &= \frac{1}{2} \tanh(\tau(e_k^{(j)} + f_S^{(j)})) + \frac{1}{2} \tanh(\tau(-e_k^{(j)} + f_S^{(j)})) \\ \eta_2(e_k^{(j)}) &= \frac{1}{2} \tanh(-\tau(e_k^{(j)} + f_S^{(j)})) + \frac{1}{2} \tanh(\tau(-e_k^{(j)} + f_S^{(j)}))\end{aligned}\quad (6)$$

where τ is a smoothing parameter. Extensive simulations have verified that, using $\tau \geq 10^4$, the valve signature given by the proposed smoothed model (5) matches precisely the original He's model results (see [6] for more details). It is to be noted that the proposed identification method can also be applied with other smoothing functions derived on the basis of the various discontinuous data-driven stiction models available in the literature, e.g. [13] and [29].

3. The proposed methodology

In this section the proposed stiction identification and quantification method is presented.

3.1. Hammerstein Model

As stated in Section 2, the linear part of the extended Hammerstein system to be identified can be expressed by various model. In the proposed approach, an AutoRegressive eXogenous (ARX) structure in discrete-time form is considered.

The approach used to build multiple input - multiple output (MIMO) models is to write a series of multiple input - single output (MISO) relations, for each output $y^{(i)}$, with $i = 1, \dots, p$, by identifying the parameters of the following linear difference equation:

$$\begin{aligned}y_k^{(i)} + a_1^{(i)} y_{k-1}^{(i)} + \dots + a_{n_a^{(i)}}^{(i)} y_{k-n_a^{(i)}}^{(i)} &= b_{1,1}^{(i)} \chi_{k-1}^{(1)} + \dots + \\ b_{1,n_{b_1}^{(i)}}^{(i)} \chi_{k-n_{b_1}^{(i)}}^{(1)} + \dots + b_{m,1}^{(i)} \chi_{k-1}^{(m)} + \dots + b_{m,n_{b_m}^{(i)}}^{(i)} \chi_{k-n_{b_m}^{(i)}}^{(m)} + v_k^{(i)}\end{aligned}\quad (7)$$

where $\chi_k^{(j)}$ is the j -th valve output, that is, the j -th process input, and $y_k^{(i)}$ is the i -th output; $n_a^{(i)}$ is the i -th output order, $n_{b_j}^{(i)}$ is the j -th input order associated with the i -th output.

Using the backward shift operator q^{-1} (i.e. such that $x_k^{(j)} = q^{-1} x_{k+1}^{(j)}$), the ARX system can be described by the following polynomials:

$$A^{(i)}(q) = 1 + a_1^{(i)} q^{-1} + \dots + a_{n_a^{(i)}}^{(i)} q^{-n_a^{(i)}} \quad (8a)$$

$$B_j^{(i)}(q) = b_{j,1}^{(i)} q^{-1-t_{d_j}^{(i)}} + \dots + b_{j,n_{b_j}^{(i)}}^{(i)} q^{-n_{b_j}^{(i)}-t_{d_j}^{(i)}} \quad (8b)$$

where $t_{d_j}^{(i)}$ is the j -th input time-delay of the process associated with the i -th output. Hence, the system (7) can be rewritten as:

$$A^{(i)}(q) y_k^{(i)} = \sum_{j=1}^m B_j^{(i)}(q) \chi_{k-t_{d_j}^{(i)}}^{(j)} + v_k^{(i)} \quad (9)$$

in which the valve position χ is calculated from the controller output u using the aforesaid smoothed stiction model $\varphi_S(\cdot)$ (5).

3.2. Optimization problem structure

The core of the proposed stiction quantification method is comprised by a nonlinear optimization problem. Let us define X as the vector of optimization variables, formed by stacking $2m$ stiction parameters, the static and dynamic friction of (5), with $N_a + N_b$ coefficients of the ARX process model (9), where $N_a = \sum_{i=1}^p n_a^{(i)}$ and $N_b = \sum_{i=1}^p \sum_{j=1}^m n_{b_j}^{(i)}$. The optimization variable vector can be thus written as:

$$X = [\hat{f}_S^{(1)}, \dots, \hat{f}_S^{(m)}, \hat{f}_D^{(1)}, \dots, \hat{f}_D^{(m)}, \hat{\theta}^T]^T \quad (10)$$

where:

$$\begin{aligned}\hat{\theta} &= [a_1^{(1)}, \dots, a_{n_a^{(1)}}^{(1)}, \dots, a_1^{(p)}, \dots, a_{n_a^{(p)}}^{(p)}, \\ &\quad b_{1,1}^{(1)}, \dots, b_{1,n_{b_1}^{(1)}}^{(1)}, \dots, b_{m,1}^{(p)}, \dots, b_{m,n_{b_m}^{(p)}}^{(p)}]^T\end{aligned}\quad (11)$$

The parameters of the optimization problem, instead, are the controller outputs u and corresponding plant outputs y sequences, i.e. $\mathbf{y} = [y_0, \dots, y_N] \in \mathbb{R}^{(p \times N)}$ and $\mathbf{u} = [u_0, \dots, u_N] \in \mathbb{R}^{(m \times N)}$ in which N is the number of data-points collected. For the sake of simplicity, the orders on the autoregressive and exogenous terms ($n_a^{(i)}, n_{b_j}^{(i)} \forall i = 1, \dots, p, \forall j = 1, \dots, m$) are assumed known a priori. If process orders are not known, they could be evaluated by applying an information criterion (see e.g. [30]).

Let us define the symbol $\mathbf{y}_{[k:k+\rho]}^{(i)} = [y_k^{(i)}, \dots, y_{k+\rho}^{(i)}] \in \mathbb{R}^{(1 \times \rho)}$, being ρ a generic time sample; therefore, the nonlinear optimizer finds an optimal solution solving the following one-stage problem:

$$X^* = \arg \min_X \frac{1}{p} \sum_{i=1}^p \text{SE}(\mathbf{y}_{[k:k+\rho]}^{(i)}, \hat{\mathbf{y}}_{[k:k+\rho]}^{(i)}) \quad (12a)$$

subject to:

$$\hat{\chi}_k^{(j)} = \varphi_S(\hat{\chi}_{k-1}^{(j)}, u_k^{(j)}) \quad \forall k = 1, \dots, N; j = 1, \dots, m \quad (12b)$$

$$\hat{\mathbf{y}}_{[k:k+\rho]}^{(i)} = \Phi^{(i)} \hat{\theta}^{(i)} \quad (12c)$$

$$f_{\min}^{(j)} \leq f_S^{(j)}, f_D^{(j)} \leq f_{\max}^{(j)} \quad \forall j = 1, \dots, m \quad (12d)$$

$$f_S^{(j)} \geq f_D^{(j)} \quad \forall j = 1, \dots, m \quad (12e)$$

$$\sigma^2(\hat{\chi}^{(j)}) \geq \sigma_{\min,j}^2 \quad \forall j = 1, \dots, m \quad (12f)$$

where $\varepsilon_i = \max(n_a^{(i)}, N_b^{(i)})$, with $N_b^{(i)} = \sum_{j=1}^m n_{b_j}^{(i)} + t_{d_j}^{(i)}$, represents the number of unidentifiable data-points for each output $y^{(i)}$ and $\hat{\chi}^{(j)} = [\hat{\chi}_1^{(j)}, \dots, \hat{\chi}_N^{(j)}] \in \mathbb{R}^{(1 \times N)}$ is the estimated j -th valve position sequence. The regressor matrix for the i -th output, $\Phi^{(i)} \in \mathbb{R}^{N - \varepsilon_i \times n_a^{(i)} + N_b^{(i)}}$, is built by stacking singular regressor vectors $\phi_k^{(i)}$ at each sample time k , with values of valves position estimated from the nonlinear model (5), i.e.:

$$\phi_k^{(i)} = [-y_{k-1}^{(i)}, \dots, -y_{k-n_a}^{(i)}, \hat{\chi}_{k-1-t_{d1}}^{(1)}, \dots, \hat{\chi}_{k-n_{b1}-t_{d1}}^{(1)}, \dots, \hat{\chi}_{k-1-t_{dm}}^{(m)}, \dots, \hat{\chi}_{k-n_{bm}-t_{dm}}^{(m)}] \quad (13)$$

The term $\hat{\theta}^{(i)}$ represents the subset of coefficients of $\hat{\theta}$ associated with the i -th output and defined consistently with (11).

The objective function is the Square Error (SE) between the output of actual process and of identified model \hat{y} , where:

$$\text{SE}(\mathbf{y}_{[\varepsilon_i:N]}^{(i)}, \hat{\mathbf{y}}_{[\varepsilon_i:N]}^{(i)}) = (\mathbf{y}_{[\varepsilon_i:N]}^{(i)} - \hat{\mathbf{y}}_{[\varepsilon_i:N]}^{(i)})^T (\mathbf{y}_{[\varepsilon_i:N]}^{(i)} - \hat{\mathbf{y}}_{[\varepsilon_i:N]}^{(i)}) \quad (14)$$

Remark 1. The only data points taken into consideration in the objective function are in the range $[\varepsilon_i : N]$. The components of the estimated output corresponding to the first ε_i data points, necessary to build the output predictor, are imposed equal to the real ones, i.e. $\hat{\mathbf{y}}_{[0:\varepsilon_i-1]}^{(i)} = \mathbf{y}_{[0:\varepsilon_i-1]}^{(i)}$.

Remark 2. It has to be underlined how the real valve positions χ are not known, since these measurements may not be available in many industrial plants. It is thus not possible to insert these variables into the objective function. Therefore, being model (2) an extended Hammerstein model composed by two parts, it may happen that a minimization on the process output error (12a) does not reflect a minimization on valve output error, that is, the quantity $\frac{1}{m} \sum_{j=1}^m (\boldsymbol{\chi}^{(j)} - \hat{\boldsymbol{\chi}}^{(j)})^T (\boldsymbol{\chi}^{(j)} - \hat{\boldsymbol{\chi}}^{(j)})$.

We note that constraints (12d) and (12e) describe the stiction parameters domain. As a matter of fact, this domain has a triangular shape: $0 = f_{\min}^{(j)} \leq f_D^{(j)} \leq f_S^{(j)} \leq f_{\max}^{(j)}$. Indeed, the so-called *overshoot* stiction cases are excluded, since waveforms generated by these parameters combinations are rarely observed in the industrial practice. The largest value of stiction parameters can be assumed equal to the span of sustained oscillation of the corresponding controller output: $f_{\max}^{(j)} = \Delta u^{(j)}$. As known, under boundary conditions, when $f_S^{(j)} + f_D^{(j)} = \Delta u^{(j)}$, the valve jumps between two extreme positions, thus generating an exactly square-shaped signal.

Constraint (12f) is imposed on the variance of the identified sequence of valve position, $\sigma^2(\hat{\boldsymbol{\chi}}^{(j)})$. This assures avoiding uncommon waveforms, that is, the stem position of a sticky valve cannot be fully or mostly steady when the input is oscillating, so that $\hat{\boldsymbol{\chi}}^{(j)}$ is forced to oscillate just because of the presence of stiction. A safe choice is considering the controller output variance, e.g. $\sigma_{\min,j}^2 = \alpha \sigma^2(\mathbf{u}^{(j)})$, with $\alpha = 0.1$.

3.3. Routine initialization and implementation

Problem (12) calculates the optimal solution starting from a suitable initial point X_0 . In particular, the coefficient vector $\hat{\theta}$ of linear process dynamics is initialized by performing a first-guess identification with an ARX model between controller outputs u and process variables y , that is, no stiction is

firstly assumed within the m valves. Hence, for the i -th output, by reversing (12c), $\hat{\theta}_0^{(i)}$ is calculated by:

$$\hat{\theta}_0^{(i)} = \text{pinv}(\Phi_0^{(i)}) \mathbf{y}_{[\varepsilon_i:N]}^{(i)} = [\Phi_0^{(i)T} \Phi_0^{(i)}]^{-1} \Phi_0^{(i)T} \mathbf{y}_{[\varepsilon_i:N]}^{(i)} \quad (15)$$

where $\Phi_0^{(i)} \in \mathbb{R}^{N-\varepsilon_i \times n_a^{(i)} + N_b^{(i)}}$ is the initial regressor matrix of the measurements, computed by stacking singular linear regressor vectors $\phi_{0,k}^{(i)}$ at each sample time k :

$$\phi_{0,k}^{(i)} = [-y_{k-1}^{(i)}, \dots, -y_{k-n_a}^{(i)}, u_{k-1-t_{d1}}^{(1)}, \dots, u_{k-n_{b1}-t_{d1}}^{(1)}, \dots, u_{k-1-t_{dm}}^{(m)}, \dots, u_{k-n_{bm}-t_{dm}}^{(m)}] \quad (16)$$

Considerations made about $\hat{\theta}^{(i)}$ stand also for $\hat{\theta}_0^{(i)}$, that is, $\hat{\theta}_0^{(i)}$ contains all the process dynamics coefficients related to the i -th output.

Initialization of the stiction parameters $f_S^{(j)}$ and $f_D^{(j)}$ is done by multiple iterations. As a matter of fact, in order to avoid to be stuck in a local minimum, a multiple starting algorithm has been implemented thus improving robustness of the optimization routine. The proposed approach (12) is hence iterated by setting M initial guesses. A good method is starting from the boundaries of the triangular-shaped domain of stiction parameters, by fixing a suitable step, e.g. $\Delta f_S^{(j)} = \Delta f_D^{(j)} = 0.5$. Further initial points can be obtained from the ellipse-fitting method [15]. An estimate of the so-called *apparent* stiction S_0 on the x-width of polar plot PV(OP), that is, $y^{(i)}(u^{(j)})$, is computed, and a limited set of combinations which satisfy $S_0 = f_S^{(j)} + f_D^{(j)}$ are added. Thus, respecting feasibility constraints and stability conditions of the identified linear process, the best solution in terms of objective function value is extracted.

Finally, note that, being integer variables, the various time-delays $t_{dj}^{(i)}$ cannot be considered as optimization variables in the proposed framework, but only as parameters, as they would require the redefinition of problem (12) in a Mixed-Integer Non Linear Programming (MINLP) way. Therefore, an iterative approach can be derived by repeating the aforesaid initialization and computing the optimization problem, for different time-delay matrices Θ_d , by varying the single elements within preset ranges. This approach is robust, but becomes computationally very expensive for large MIMO systems, as tested combinations are $T^{m \cdot p}$, being T the number of possible time-delays for each input-output channel.

In the current work, implementation of the above described algorithm, is performed on a code written in Python 3.6 with the use of symbolic framework CasADi 3.4 [31]. As nonlinear programming solver, the optimization problem implements IPOPT, a well-established interior point algorithm [32].

4. Simulation analysis

The performance of the proposed methodology is investigated on two numerical examples.

4.1. Example 1: SISO case with time-delay

For the linear process dynamics, a third order transfer function with time-delay is considered:

$$P(s) = \frac{1}{(10s+1)(5s+1)(s+1)} e^{-5s} \quad (17)$$

which corresponds to the following ARX process in discrete-time form with sample period $T_s = 1$ and $(n_a, n_b, t_d) = (3, 3, 5)$:

$$y_k = 2.0914y_{k-1} - 1.3749y_{k-2} + 0.2725y_{k-3} + 0.002445\chi_{k-1-t_d} + 0.007181\chi_{k-2-t_d} + 0.001278\chi_{k-3-t_d} + v_k \quad (18)$$

The output white noise v is a random sequence with normal distribution, zero-mean and standard deviation $\sigma = 10^{-2}$. Valve stiction is described by standard discontinuous He's model (3) with parameters $f_S = 5$ and $f_D = 2$.

Closed-loop data are generated in Python by using a model predictive controller (MPC) [33] and imposing some set-point changes. This controller formulation is *stiction unaware*, as valve dynamics $\varphi(\cdot)$ of (2) is completely neglected and only the process model $P(s)$ in the corresponding state-space discrete-time form is embedded.

The Finite Horizon Optimal Control Problem (FHOC) solved at each time k is defined as following:

$$\min_{\mathbf{x}, \mathbf{u}} \sum_{i=0}^{N_H-1} [(\xi_i - \xi_{s,k})^T Q (\xi_i - \xi_{s,k}) + (u_i - u_{i-1})^T S (u_i - u_{i-1})] + (\xi_{N_H} - \xi_{s,k})^T V_F (\xi_{N_H} - \xi_{s,k}) \quad (19a)$$

subject to:

$$\xi_0 = \hat{\xi}_k \quad (19b)$$

$$\xi_{i+1} = \mathbf{A}\xi_i + \mathbf{B}u_i \quad (19c)$$

$$y_i = \mathbf{C}\xi_i + \hat{d}_k \quad (19d)$$

$$\xi_i \in \mathbb{X}, \quad u_i \in \mathbb{U}, \quad y_i \in \mathbb{Y} \quad (19e)$$

where N_H is a positive integer representing the horizon length, Q , S , and V_F are the various penalty matrices, $\hat{\xi}$ and \hat{d} are the current state and disturbance estimate of the model (2), the triple $(\xi_{s,k}, u_{s,k}, y_{s,k})$ represents the steady-state values satisfying (2), and the triple $(\mathbb{X}, \mathbb{U}, \mathbb{Y})$ forms the constraint set. For more details see [6].

The smoothed stiction model of (5) is used in the identification stage, on the basis of controller output u and process variable y , vectors of length $N = 2500$. The ARX model orders are assumed known; the time-delay is estimated by using the iterative approach presented in Section 3.3, by testing $T = 11$ values in the interval $[0, 10]$.

The proposed nonlinear optimization method proves to be effective, as it obtains very accurate valve stiction parameters: $\hat{f}_S = 4.95$, $\hat{f}_D = 2.06$. The estimated process time-delay is equal to the true value $\hat{t}_d = 5$, and the other identified process parameters are not too far from the actual values:

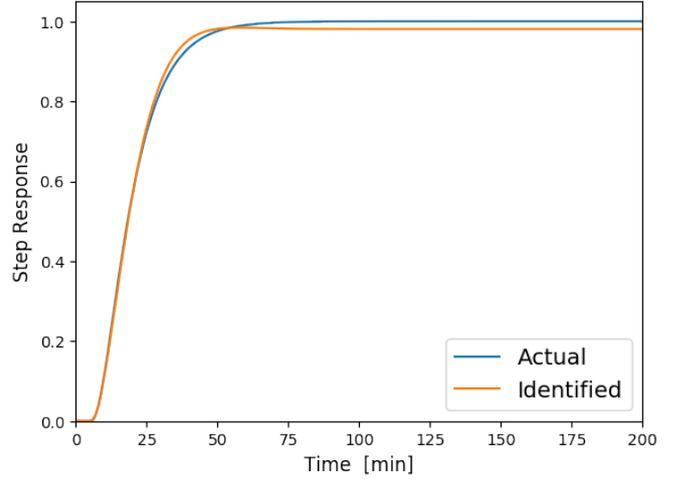


Figure 2: Example 1. Unit step response of the actual and the identified process dynamics.

$$\hat{\theta} = [-1.281, -0.1107, 0.413, 0.0030, 0.0073, 0.0099]$$

In particular, the step response of the actual linear process $P(s)$ and the identified dynamics $\hat{P}(s)$ are really close, as shown in Figure 2. Time-delay is perfectly estimated and the identified transitory response is very accurate; only the static gain is affected by a little error.

Note that the computational time required by the proposed method is quite short: about 195 s for $M = 7$. Figure 3 shows measured time trends and estimated signals for a data window of sustained oscillation with a set-point variation. It is to be observed that the fitting on process variable is high, and also the estimation of valve position is very accurate even when larger changes of input and valve position occur right due to the set-point change.

The same data set is then analyzed with a traditional grid-search method over the space of nonlinear model parameters, as explained by [20]. A triangular grid of stiction parameters (f_S, f_D) is built, and for each possible combination valve output is generated from measured controller output by using the smoothed stiction model (5). Then, ARX model coefficients are identified by least-squares regression on the basis of the generated valve output and the measured process output. The optimal combination of stiction parameter is evaluated as the one that minimizes the SE on process variable (14). The step size of the grid is set to $\Delta f_S = \Delta f_D = 0.1$, the largest possible value of stiction parameters are set equal to the sustained oscillation span of the controller output $f_{S,\max} = f_{D,\max} = \Delta u$, and the true combination is included. Not surprisingly, stiction parameters are identified exactly, but at the expense of much higher computational time (about 10500 s). In addition, the grid method finds the exact stiction parameter values as expected, although the estimated time-delay is incorrect ($t_d = 6$). This is due to the presence of white noise, which induces the minimum value of SE being calculated with a different value of time-delay.

It is evident that the computational time of the proposed optimization-based method is definitely shorter than grid-

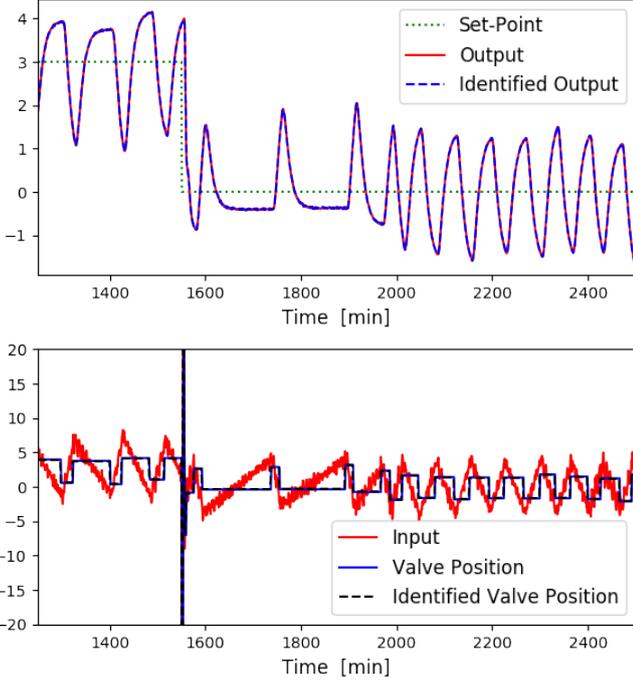


Figure 3: Example 1. Measured and estimated time trends.

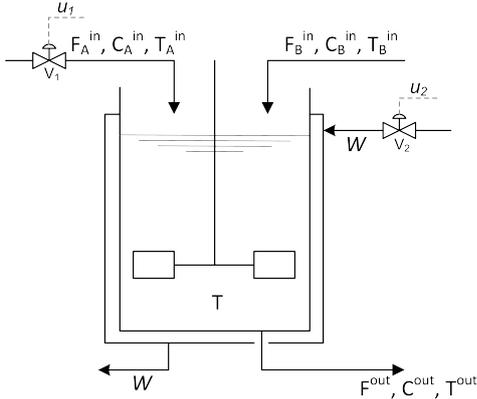


Figure 4: A multivariable system: the Continuous Stirred Tank.

search, thus offering a remarkable advantage in on-line applications. The validity of the proposed technique is further investigated in the following multivariable example.

4.2. Example 2: a MIMO case

As multivariable system, a continuous stirred tank (CST) mixer is here studied. A schematic diagram of the system is shown in Figure 4. The target is to control the output concentration C_{out} and the output temperature T_{out} by means of two control valves (V_1 , V_2), which can be subject to stiction, described by He's standard model (4). Valve V_1 is sticky, with $f_S^{(1)} = 8$, $f_D^{(1)} = 1$, while valve V_2 is healthy ($f_S^{(2)} = f_D^{(2)} = 0$), so that $\chi^{(2)} = u^{(2)}$. The process dynamics is described by the transfer matrix $G(z)$ reported in Table 1. Further details on the CST and the derivation of linear process model from the original nonlinear system dynamics are reported in Appendix A.

A standard model predictive controller based only on the linear process model of Table 1 is considered. As said in Section 4.1, this controller formulation is *stiction unaware*, as it totally disregards valve dynamics. A data set is obtained in closed-loop mode by imposing some set-point changes to process outputs. Each output of the system is corrupted by measurement white noise v , whose covariance matrix is set to $R_{wn} = 10^{-4}$. The prediction horizon and the sampling period are set to $N_H = 50$ and $T_s = 1$ min, respectively.

On the basis of the generated data, i.e. the sequences of controller outputs \mathbf{u} and process variables \mathbf{y} , the smoothed stiction model (5) and the MIMO ARX model (9) are used in the identification stage. The ARX model orders and the time-delay matrix ($T = \mathbf{0}^{m \times p}$) are supposed known for the sake of simplicity. The proposed nonlinear optimization method proves to be effective, as it obtains very accurate stiction parameters: $\hat{f}_S^{(1)} = 7.99$, $\hat{f}_D^{(1)} = 1.01$, and $\hat{f}_S^{(2)} = 0.0$, $\hat{f}_D^{(2)} = 0.0$. The corresponding process parameters are also reasonably close to their actual values, as shown in Table 2. As a matter of fact, the unit step response of the actual linear process $G(z)$ and the identified process dynamics $\hat{G}(z)$ are really close, as shown in Figure 5. It is important to remark that the actual transfer function g_{12} is essentially null, whereas the identified one shows practically zero steady-state gain and some minor, transient oscillations around the origin. Note that the computational time required is very short: about 43 seconds for $M = 5$ starting points.

This data set is also analyzed with the traditional grid-search method [20]. For a MIMO case, such approach has to be repeated for each $f_s^{(j)}$ and $f_d^{(j)}$, causing an exponential growth of the computational cost with the number of stiction parameter to be identified. The step size of the grid chosen is $\Delta f_S = \Delta f_D = 1$, ten times bigger than the SISO case, and the true combination is included. Each parameter $f_s^{(j)}$ or $f_d^{(j)}$ is spaced in a range of the maximum periodic oscillation of $\mathbf{u}^{(j)}$. Not surprisingly, stiction parameters are still identified exactly, but even with such a loose grid, the resulting computational time is about 6200 s. Decreasing the grid step size to 0.1, as used in the SISO case, makes the computational time nearly reaching a week.

Then, a stiction model with the newly identified parameters is implemented within two other formulations of MPC in order to build stiction aware controllers, as proposed in [6, 10].

- The first structure, named *stiction embedding* controller, is thus derived by augmenting the plant model with estimated valves dynamics with the smoothed model $\phi_S(\cdot)$.
- The second controller, named *stiction inversion*, is also aware of stiction, but it has an explicit model for the inverse dynamics of He's standard stiction model (ϕ^{-1}).

In stiction inversion MPC, the controller output \tilde{u} , subject to linear optimization, forms input to stiction inverse model, so that $u = \phi^{-1}(\tilde{u})$ is the output of the whole controller, as firstly proposed by [34]. Note that, for a perfect stiction inversion, one gets $\phi(\phi^{-1}(\tilde{u})) = \tilde{u}$, and then $\tilde{u} \equiv \chi$.

The following scenario is studied: the linear model of CST used within both stiction aware MPC formulations is the exact

Table 1: MIMO linear model $G(z)$ used for the nonlinear dynamics of CST.

	Input 1	Input 2
Output 1	$g_{11} = \frac{0.0024z^4 - 0.0086z^3 + 0.01167z^2 - 0.0070z + 0.0016}{z^5 - 4.5130z^4 - 8.1438z^3 - 7.3452z^2 + 3.3117z - 0.5969}$	$g_{12} = \frac{-9.0795 \cdot 10^{-6}z^4 + 3.2188 \cdot 10^{-5}z^3 - 4.2938z^2 + 2.5550z - 5.7232 \cdot 10^{-6}}{z^5 - 4.5130z^4 - 8.1438z^3 - 7.3452z^2 + 3.3117z - 0.5969}$
Output 2	$g_{21} = \frac{-0.0169z^4 + 0.0608z^3 - 0.0823z^2 + 0.0494z - 0.0111}{z^5 - 4.5130z^4 - 8.1438z^3 - 7.3452z^2 + 3.3117z - 0.5969}$	$g_{22} = \frac{0.0281z^4 - 0.1014z^3 + 0.1371z^2 - 0.0823z + 0.0185}{z^5 - 4.5130z^4 - 8.1438z^3 - 7.3452z^2 + 3.3117z - 0.5969}$

Table 2: Identified ARX MIMO system $\hat{G}(z)$.

	Input 1	Input 2
Output 1	$g_{11} = \frac{0.0022z^4 + 0.0020z^3 + 0.0009z^2 + 0.0014z + 0.0015}{z^5 - 0.1111z^4 - 0.1135z^3 - 0.1432z^2 - 0.1436z - 0.1584}$	$g_{12} = \frac{-0.0168z^4 - 0.0111z^3 - 0.0075z^2 - 0.044z - 0.0010}{z^5 - 0.1111z^4 - 0.1135z^3 - 0.1432z^2 - 0.1436z - 0.1584}$
Output 2	$g_{21} = \frac{-0.0160z^4 + 0.0168z^3 + 0.0213z^2 - 0.0229z + 0.0007}{z^5 - 0.2079z^4 - 0.2279z^3 - 0.1270z^2 - 0.1125z - 0.1104}$	$g_{22} = \frac{0.0905z^4 - 0.0653z^3 + 0.0495z^2 + 0.0590z - 0.0699}{z^5 - 0.2079z^4 - 0.2279z^3 - 0.1270z^2 - 0.1125z - 0.1104}$

one shown in Table 1, and only stiction embedding MPC has a globally nonlinear model, as it also implements valves nonlinearity, while stiction inversion MPC is purely linear.

The performance of these two MPC formulations is tested in closed-loop mode, under the same conditions in terms of state observer and disturbance model previously used for the stiction unaware controller. The prediction horizon and the sampling period are also the same. The peculiarities lay in a suitable stiction compensation sequence and in a revised cost function, which are respectively used within dynamic optimization module of the stiction embedding MPC formulation as warm-start and as objective function, as introduced in [6] and then further discussed in [10].

The warm-start for stiction-compensation. In order to get good tracking performance and move variables to their targets by avoiding sustained oscillations induced by valve stiction, a suitable *warm-start* is given to the stiction embedding MPC nonlinear programming solver. That is, the FHOCP (19) is solved by IPOPT starting from the warm-start computed as follows.

At each sampling time, the steady-state optimization module can compute a suitable steady-state target (χ_{ss}) also for the valve output:

$$z_{ss} = \begin{bmatrix} \chi_{ss} \\ \xi_{ss} \end{bmatrix} = \begin{bmatrix} \varphi(\chi_{ss}, u_{ss}) \\ \mathbf{A}\xi_{ss} + \mathbf{B}\varphi(\chi_{ss}, u_{ss}) \end{bmatrix} \quad (20)$$

$$y_{ss} = \mathbf{C}\xi_{ss} = y_{sp}$$

A general formulation of warm-start can be obtained by solving a dynamic optimization problem, as explained in [6, 10]. Alternatively, a particular input sequence could be used as first-guess trajectory within the dynamic module. This suitable warm-start is inspired by a new version of the well-known two-move stiction compensation method [35].

In the present work, the following sequence is given as warm-start to input signal of each m valve on the basis of standard

He's stiction model (4):

$$u_k = \begin{cases} u_{k-1} + af_S & \text{if } u_{k-1} \geq \chi_{ss} \\ u_{k-1} - af_S & \text{if } u_{k-1} < \chi_{ss} \end{cases}$$

$$u_{k+1} = \begin{cases} \chi_k + f_D & \text{if } u_k \geq \chi_k \\ \chi_k - f_D & \text{if } u_k < \chi_k \end{cases} \quad (21)$$

$$u_{k+2} = \begin{cases} \chi_{ss} - f_D & \text{if } u_{k+1} \geq \chi_{ss} \\ \chi_{ss} + f_D & \text{if } u_{k+1} < \chi_{ss} \end{cases}$$

$$u_{k+j} = u_{k+2} (= u_{ss}) \quad \text{if } j > 2$$

The *first* input u_k moves the valve stem away from its stuck position, if $a > 2$. By observing (4), it is evident that the maximum value of the difference between valve input and output that does not cause a movement in the valve is $|u_k - \chi_{k-1}| = f_S$. If $u_{k-1} \geq \chi_{ss}$, in the worst case $u_{k-1} - \chi_{k-1} = -f_S$. Therefore, if $a > 2$, one gets $|u_k - \chi_{k-1}| > f_S$ and moves the valve: $\chi_k \neq \chi_{k-1}$. Then, for the *second* movement u_{k+1} , the input is moved towards the actual valve position χ_k and set at a distance f_D . The *third* movement brings the stem position to its steady-state value (χ_{ss}) in order to eliminate error on control variable and set the input at distance f_D , so that $\chi_{k+2} = \chi_{ss}$. After that, the stem cannot move from steady-state position since the input signal u_{k+j} (with $j > 2$) is always kept constant.

The controller tuning. For all three MPC formulations, tuning parameters of the static module and the terminal penalty matrix are the same: $Q_{ss} = I_2$, $R_{ss} = 0$, and $V_F = 10^3 I_{n_\xi}$. Some differences lay in the dynamic modules. Both stiction unaware (SU) and stiction inversion (SI) MPC adopt the dynamic problem (19) with the following penalty values: $Q = \frac{\mathbf{C}^T \mathbf{C}}{\max c_{ij}^2} + 0.01 I_{n_\xi}$, where $\max c_{ij}^2$ is the maximum element of matrix \mathbf{C} , and $S = \text{diag}[10, 100]$.

In addition, within stiction embedding (SE) MPC, the objective function (19a) is modified as follows:

$$\ell(z_i, u_i) = (z_i - z_{ss})^T Q_z (z_i - z_{ss}) + \Delta \hat{\chi}_i^T Q_s \Delta \hat{\chi}_i + (u_i - u_{s,k})^T R (u_i - u_{s,k}) \quad (22)$$

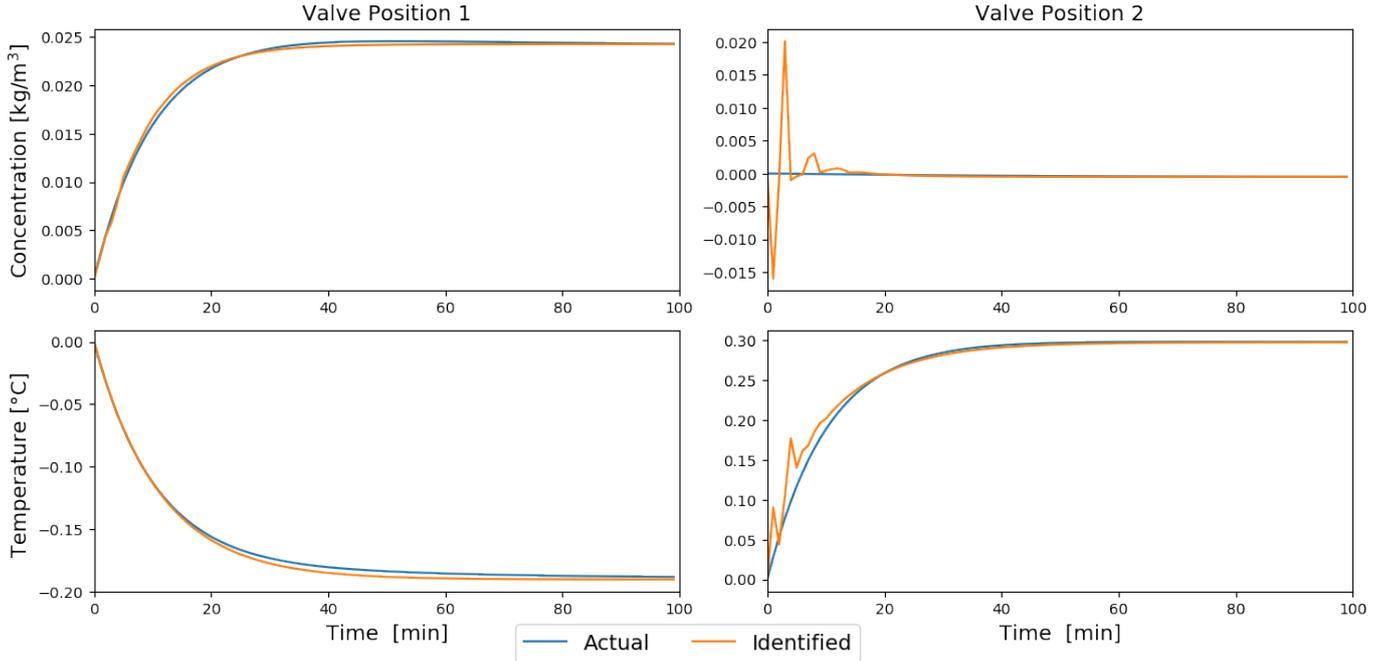


Figure 5: Example 2. Unit step response of the actual and the identified process dynamics.

where $R \in \mathbb{R}^{m \times m}$ is the control penalty matrix, $\Delta \hat{\chi}_i = \hat{\chi}_i - \hat{\chi}_{i-1}$ is the rate of change of the estimated valve position, and $Q_s \in \mathbb{R}^{m \times m}$ is the corresponding difference penalty matrix. Matrices values are: $Q_z = \text{diag}[10^3, 10^3, 1, 1, 1, 1]$, $Q_s = \text{diag}[1, 1]$, $R = \text{diag}[10, 10]$. Note that limitations on rate of changes of process input are now imposed directly on the estimated valve position and not on controller output. This refined approach has proved to further reduce offset and speed up response, but at the expense of a slightly larger input variation and wider valve movements in transitory dynamics [1].

Result discussion. Figure 6 shows time trends of the process outputs, controller outputs, and valve positions with the three stiction MPC formulations for the same set-points, comprised of sequences of step changes. It is evident that this traditional MPC does not remove fluctuations induced by stiction, as the disturbance estimate, obtained with the output disturbance model, cannot be brought to zero, but the state estimator (the *Luenberger* observer) keeps sustained oscillations which unavoidably propagate to the other variables. Stiction embedding formulation can guarantee very good tracking performance with negligible offsets on both process variables, thus an effective stiction compensation is possible. Note that valve stiction is compensated so well to reproduce the behavior of the stiction unaware controller in stiction-free environment, as shown in [10]. It has to be recalled that such high performance for stiction embedding MPC are not achievable only by augmenting plant model with the valve dynamics and an accurate estimation of stiction parameters. As a matter of fact, the appropriate warm-start and the revised cost function previously discussed are essential features to avoid the solver being stuck in local minima where the valve does not move at all [10].

On the opposite, stiction inversion MPC, despite being aware

of the valve malfunction, cannot yield good control, as sustained oscillations can arise. This happens since the conditions on input sequence defined in [6], which allow to get a perfect stiction inversion, are not verified in closed-loop operation. Note that fluctuations produced by this MPC formulation are caused by the disturbance estimate which is not zero, due to the approximation adopted in valve inverse dynamics [6]. Finally, note that controller retuning cannot completely remove these stable oscillations, but simply alters occurrences, amplitudes and frequencies.

5. Conclusions

This paper has presented a non-invasive, reliable and efficient method to identify stiction in control valves for industrial processes controlled by MPC. By the use of Hammerstein model and nonlinear optimization, the proposed approach can accurately estimate the valve stiction parameters of a recently proposed smoothed model. Applications to simulation case studies, both SISO and MIMO systems, have been presented to demonstrate the validity of the proposed method. This technique can be implemented in a on-line routine in order to improve the performance of a stiction unaware model predictive controller, which otherwise would exhibit sustained oscillations in the presence of valve stiction. In fact, it has been shown that by means of the identified stiction model, the MPC regulator can be eventually turned into stiction aware formulations, thus improving control behavior.

References

- [1] R. Bacci di Capaci, M. Vaccari, G. Pannocchia, C. Scali, Identification and estimation of valve stiction by the use of a smoothed model, IFAC-PapersOnLine 51 (18) (2018) 684–689.

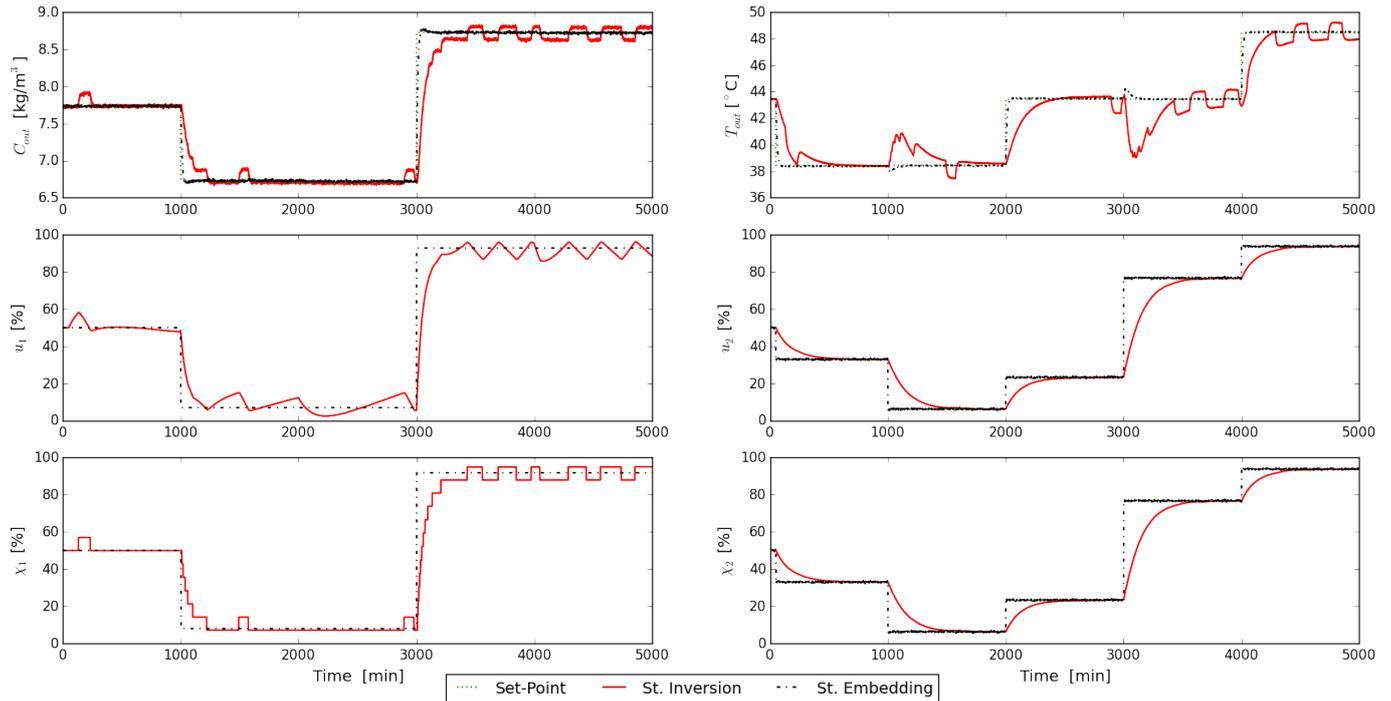


Figure 6: Process outputs (C^{out}, T^{out}), process inputs u , and valves position χ for three MPC formulations.

- [2] M. Jelali, B. Huang, Detection and Diagnosis of Stiction in Control Loops: State of the Art and Advanced Methods, Springer-Verlag, London, 2010.
- [3] M. Ale Mohammad, B. Huang, Compensation of control valve stiction through controller tuning, *J. Process Contr.* 22 (2012) 1800–1819.
- [4] R. Bacci di Capaci, C. Scali, Review and comparison of techniques of analysis of valve stiction: From modeling to smart diagnosis, *Chem. Eng. Res. Des.* 130 (2018) 230–265.
- [5] H. Durand, P. Christofides, Actuator stiction compensation via model predictive control for nonlinear processes, *AIChE Journal* 62 (2016) 2004–2023.
- [6] R. Bacci di Capaci, M. Vaccari, G. Pannocchia, A valve stiction tolerant formulation of MPC for industrial processes, *IFAC-PapersOnLine* 50 (1) (2017) 9044–9049.
- [7] J. L. Pitarch, P. Santos, C. de Prada, On stiction compensation methods for practical nonlinear MPC implementations, in: *Proceedings of the 22nd International Conference on System Theory, Control and Computing (IC-STCC)*, Sinaia, Romania, 10–12 October, 2018, pp. 747–752.
- [8] M. E. U. J. Arajo, J. R. T. Gadelha, W. M. Santos, A. L. Maitelli, F. M. U. Arajo, Nonlinear predictive control system for stiction compensation in electropneumatic control valves, *Journal of Control Science and Engineering* 1 (2018) 1–15.
- [9] M. Rodríguez, W. Heath, Controller structure for plants with combined saturation and stiction, *Proc IMechE Part I: J. Systems and Control Engineering* 1 (2018) 1–16.
- [10] R. Bacci di Capaci, M. Vaccari, G. Pannocchia, Model predictive control design for multivariable processes in the presence of valve stiction, *Journal of Process Control* 71 (2018) 25–34.
- [11] C. Garcia, Comparison of friction models applied to a control valve, *Contr. Eng. Pract.* 16 (2008) 1231–1243.
- [12] M. A. A. S. Choudhury, S. L. Shah, N. F. Thornhill, Modelling valve stiction, *Contr. Eng. Pract.* 13 (2005) 641–658.
- [13] M. Kano, M. Hiroshi, H. Kugemoto, K. Shimizu, Practical model and detection algorithm for valve stiction, in: *Proceedings of 7th IFAC DYCOPS*, Boston, USA, 2004, paper ID n. 54.
- [14] Q. P. He, J. Wang, M. Pottmann, S. J. Qin, A curve fitting method for detecting valve stiction in oscillating control loops, *Ind. Eng. Chem. Res.* 46 (2007) 4549–4560.
- [15] M. A. A. S. Choudhury, S. L. Shah, N. F. Thornhill, D. S. Shook, Automatic detection and quantification of stiction in control valves, *Contr. Eng. Pract.* 16 (2006) 1395–1412.
- [16] R. Srinivasan, R. Rengaswamy, S. Narasimhan, R. Miller, Control loop performance assessment 2. Hammerstein model approach for stiction diagnosis, *Ind. Eng. Chem. Res.* 44 (2005) 6719–6728.
- [17] M. A. A. S. Choudhury, S. L. Shah, N. F. Thornhill, D. S. Shook, Stiction - definition, modeling, detection and quantification, *J. Process Contr.* 18 (2008) 232–243.
- [18] M. Jelali, Estimation of valve stiction in control loops using separable least square and global search algorithms, *J. Process Contr.* 18 (2008) 632–642.
- [19] R. Bacci di Capaci, C. Scali, Stiction quantification: A robust methodology for valve monitoring and maintenance scheduling, *Ind. Eng. Chem. Res.* 53 (2014) 7507–7516.
- [20] R. Bacci di Capaci, C. Scali, G. Pannocchia, System identification applied to stiction quantification in industrial control loops: A comparative study, *J. Process Contr.* 46 (2016) 11–23.
- [21] G. Wang, J. Wang, Quantification of valve stiction for control loop performance assessment, in: *Proceedings of the 16th International Conference on Industrial Engineering and Engineering Management*, Beijing, China, 2123 October, 2009, pp. 1189–1194.
- [22] R. Romano, C. Garcia, Valve friction and nonlinear process model closed-loop identification, *J. Process Contr.* 21 (2011) 667–677.
- [23] H. Zabiri, A. Maulud, N. Omar, NN-based algorithm for control valve stiction quantification, *WSEAS Trans. Syst. Control* 4 (2009) 88–97.
- [24] A. Araujo, C. Munaro, M. R. Filho, Quantification of valve stiction and dead band in control loops based on the harmonic balance method, *Ind. Eng. Chem. Res.* 51 (2012) 14121–14134.
- [25] Q. He, J. Wang, Valve stiction quantification method based on a semi-physical valve stiction model, *Ind. Eng. Chem. Res.* 53 (2014) 12010–12022.
- [26] M. Farenzena, J. Trierweiler, Valve stiction evaluation using global optimization, *Contr. Eng. Pract.* 20 (2012) 379–385.
- [27] J. Wang, Q. Zhang, Detection of asymmetric control valve stiction from oscillatory data using an extended Hammerstein system identification method, *J. Process Control* 24 (2014) 1–12.
- [28] Q. P. He, J. Wang, Valve stiction modeling: First-principles vs. data-drive approaches, in: *Proceedings of the 7th American Control Conference*, Baltimore, MD, USA, 30 June–2 July, 2010, pp. 3777–3782.
- [29] S. L. Chen, K. K. Tan, S. Huang, Two-layer binary tree data-driven model for valve stiction, *Ind. Eng. Chem. Res.* 47 (2008) 2842–2848.
- [30] G. Armenise, R. Bacci di Capaci, M. Vaccari, G. Pannocchia, An open-

source system identification package for multivariable processes, in: Proceedings of the UKACC 12th International Conference on Control (CONTROL), Sheffield, UK, 5–7 September, 2018, pp. 152–157.

- [31] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, M. Diehl, CasADi – A software framework for nonlinear optimization and optimal control, *Mathematical Programming Computation* (2018) 1–36.
- [32] A. Wächter, L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Mathematical programming* 106 (1) (2006) 25–57.
- [33] M. Vaccari, G. Pannocchia, A multipurpose, easy-to-use Model Predictive Control design and simulation code, in: Proceedings of the 4th European Conference on Computational Optimization EUCCO, Leuven, Belgium, 2016.
- [34] M. Rodríguez, W. Heath, MPC for plants subject to saturation and dead-zone, backlash or stiction, in: Proceedings of the 4th IFAC Nonlinear Model Predictive Control Conference, Noordwijkerhout, The Netherlands, 2012, pp. 418–423.
- [35] R. Srinivasan, R. Rengaswamy, Stiction compensation in process control loops: A framework for integrating stiction measure and compensation, *Ind. Eng. Chem. Res.* 44 (2005) 9164–9174.

Appendix A. Details on the CST mixer

The meaning of variables and parameters are detailed in Table A.3. Seven variables could be considered as inputs to the system: two manipulated variables, the input flow rate of stream A, F_{in}^A , and the steam flow rate into the jacket, W ; and five disturbance variables, the input concentrations of stream A and B, (C_{in}^A, C_{in}^B) , the two input temperatures, (T_{in}^A, T_{in}^B) , and the input flow rate of stream B, F_{in}^B . Therefore, process inputs are u_1 and u_2 , that is, the output signals from MPC, while process states, which correspond to process outputs, are concentration C^{out} and temperature T^{out} .

Table A.3: Variables and parameters of Continuous Stirred Tank.

Parameter or Variable	Description [Unit]	Value or Constraints
V	tank volume [m^3]	10.0
ρ	solution density [kg/m^3]	1100.0
c_p	specific heat [$kJ/kg\ K$]	4.180
λ	latent heat [kJ/kg]	2272.0
C_i	concentration [$kg\ solute/m^3\ solution$]	$C_A^{in} = 10$ $C_B^{in} = 5$ $C^{out} = 0 \div 20$
T_i	temperature [$^{\circ}C$]	$T_A^{in} = 25$ $T_B^{in} = 30$ $T^{out} = 0 \div 100$
F_i	solution flow rate [m^3/min]	$F_A^{in} = 0 \div 1$ $F_B^{in} = 0.5$
W	steam flow rate [kg/min]	$0 \div 60$
$u^{(i)}$	valve input / controller output [%]	$0 \div 100$
$\chi^{(i)}$	valve position [%]	$0 \div 100$
$K^{(i)}$	proportional constant, valve position vs. flow rate	$K^{(1)} = F_A^{max}/100$
	valves have linear characteristic	$K^{(2)} = W^{max}/100$

time nonlinear process dynamics:

$$\begin{aligned} \rho F_A^{in} + \rho F_B^{in} &= \rho F^{out} \\ C_A^{in} F_A^{in} + C_B^{in} F_B^{in} - C^{out} F^{out} &= V \frac{dC}{dt} \\ \rho c_p F_A^{in} T_A^{in} + \rho c_p F_B^{in} T_B^{in} - \rho c_p F^{out} T^{out} + W\lambda &= V \rho c_p \frac{dT}{dt} \end{aligned} \quad (A.1)$$

Note that for the sake of simplicity, c_p , λ , ρ and then V are considered constant with temperature, and a perfect level control is assumed, so that $F_{out} = F_A^{in} + F_B^{in}$ at each time. In addition, no disturbance is active, so that, input concentration of stream A and B, two input temperatures, and the input flow rate of B, are all fixed at their steady-state values. Note also that manipulated variables can be rewritten as $F_A^{in} = K^{(1)} \chi^{(1)}$ and $W = K^{(2)} \chi^{(2)}$.

Continuous-time nonlinear dynamics of stirred tank (A.1) is integrated using explicit Runge-Kutta 4th order method, in order to match discrete-time dynamics of two valves:

$$\begin{aligned} z_{k+1}^P &= \begin{bmatrix} \chi_k \\ \zeta_{k+1} \end{bmatrix} = \phi_P(z_k^P, u_k) = \begin{bmatrix} \varphi(\chi_{k-1}, u_k) \\ f_P(\zeta_k, \varphi(\chi_{k-1}, u_k)) \end{bmatrix} \\ y_k &= C \zeta_k + v_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \zeta_k + v_k \end{aligned} \quad (A.2)$$

where two valves output and two nonlinear process states compose the state vector of the complete plant $z_k^P = [\chi_{k-1}^T, \zeta_k^T]^T$.

For the sake of simplicity, a linear model is identified in open-loop mode in order to describe the nonlinear dynamics $f_P(\cdot)$ of the CST. Therefore, the two valve inputs (u_1, u_2) are changed independently as pseudo-random binary sequences (PRBS) with a switch probability equal to 0.05 and amplitudes equal to 20% of the corresponding ranges around the following equilibrium point: $F_{A,0}^{in} = 0.4\ m^3\ min^{-1}$, $W_0 = 24\ kg\ min^{-1}$. Once data are collected, a recently developed package for system identification is employed [30]. The well-known N4SID method is used with a fixed model order ($n = 5$), so that n new linear process states are considered for the whole plant model 2. The following state-space model is thus identified in discrete-time and then used within the three different MPC formulations:

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 0.8987 & -0.0035 & -0.0204 & 0.0099 & 0.0065 \\ -0.0064 & 0.9038 & -0.0224 & -0.0269 & -0.0062 \\ 0.0307 & 0.0136 & 0.9801 & -0.0183 & 0.0365 \\ -0.0031 & 0.0013 & -0.0188 & 0.8319 & -0.1507 \\ 0.0038 & -0.0018 & 0.0020 & 0.0127 & 0.8985 \end{bmatrix}, \\ \mathbf{B} &= \begin{bmatrix} -2.7524 \cdot 10^{-3} & 1.7493 \cdot 10^{-4} \\ 1.1984 \cdot 10^{-3} & -2.5430 \cdot 10^{-3} \\ 6.4088 \cdot 10^{-4} & 2.7857 \cdot 10^{-4} \\ -6.1987 \cdot 10^{-5} & 1.3907 \cdot 10^{-6} \\ 1.0405 \cdot 10^{-4} & -3.4249 \cdot 10^{-5} \end{bmatrix}, \\ \mathbf{C} &= \begin{bmatrix} -0.9168 & -0.0684 & -0.0777 & 0.0360 & 0.0327 \\ 1.1241 & -11.0733 & -0.8478 & -1.5393 & -0.4602 \end{bmatrix} \end{aligned} \quad (A.3)$$

Mass and energy balances yield to the following continuous-