A Systematic Grey-Box Modeling Methodology via Data Reconciliation and SOS Constrained Regression

Authors:

José Luis Pitarch, Antonio Sala, César de Prada

Date Submitted: 2019-07-25

Keywords: process modeling, SOS programming, Machine Learning, grey-box model

Abstract:

> Developing the so-called grey box or hybrid models of limited complexity for process systems is the cornerstone in advanced control and real-time optimization routines. These models must be based on fundamental principles and customized with sub-models obtained from process experimental data. This allows the engineer to transfer the available process knowledge into a model. However, there is still a lack of a flexible but systematic methodology for grey-box modeling which ensures certain coherence of the experimental sub-models with the process physics. This paper proposes such a methodology based in data reconciliation (DR) and polynomial constrained regression. A nonlinear optimization of limited complexity is to be solved in the DR stage, whereas the proposed constrained regression is based in sum-of-squares (SOS) convex programming. It is shown how several desirable features on the polynomial regressors can be naturally enforced in this optimization framework. The goodnesses of the proposed methodology are illustrated through: (1) an academic example and (2) an industrial evaporation plant with real experimental data.

DOI of Published Version: https://doi.org/10.3390/pr7030170 *processes*

Article

A Systematic Grey-Box Modeling Methodology via Data Reconciliation and SOS Constrained Regression

José Luis Pitarch [1,*], Antonio Sala [2] and César de Prada [1,3]

[1] Systems Engineering and Automatic Control department, EII, Universidad de Valladolid, C/Real de Burgos s/n, 47011 Valladolid, Spain; prada@autom.uva.es

[2] Instituto Universitario de Automática e Informática Industrial (ai2), Universitat Politècnica de Valencia, Camino de Vera S/N, 46022 Valencia, Spain; asala@isa.upv.es

[3] Institute of Sustainable Processes (IPS), Universidad de Valladolid, C/Real de Burgos s/n, 47011 Valladolid, Spain

[*] Correspondence: jose.pitarch@autom.uva.es

Received: 26 February 2019; Accepted: 20 March 2019; Published: 23 March 2019

Abstract: Developing the so-called grey box or hybrid models of limited complexity for process systems is the cornerstone in advanced control and real-time optimization routines. These models must be based on fundamental principles and customized with sub-models obtained from process experimental data. This allows the engineer to transfer the available process knowledge into a model. However, there is still a lack of a flexible but systematic methodology for grey-box modeling which ensures certain coherence of the experimental sub-models with the process physics. This paper proposes such a methodology based in data reconciliation (DR) and polynomial constrained regression. A nonlinear optimization of limited complexity is to be solved in the DR stage, whereas the proposed constrained regression is based in sum-of-squares (SOS) convex programming. It is shown how several desirable features on the polynomial regressors can be naturally enforced in this optimization framework. The goodnesses of the proposed methodology are illustrated through: (1) an academic example and (2) an industrial evaporation plant with real experimental data.

Keywords: grey-box model; machine learning; SOS programming; process modeling

1. Introduction

Due to the increasing levels of digitalization, motivated by the ideas of the so-called Industry 4.0 [1], both academic researchers and technological companies search for methods to transform raw data in useful information. This information is expected to significantly impact in the decision-making procedures at all levels in a factory: from operation and maintenance to production scheduling and supply chain.

The process industries are not alien to this digital transformation, although the challenges to face are slightly different from the ones in other sectors. On the one hand, their activity is based in complex plants formed by very heterogeneous (usually expensive) equipment, performing complex processes such as (bio)chemical reactions, phase transformations, etc. On the other hand, their markets are quite constrained in terms of raw materials or product demands, whereas environmental regulations become tighter every year. In this context, improved efficiency thanks to smart-production systems can be achieved by: (1) transforming data in reliable information and use such information to optimize the operation [2] and (2) improve the coordination of tasks in a plant [3] and between plants [4]—in summary, to equip people (operators, engineers, managers, etc.) with support systems in order to make better tactical decisions.

Suitable models of a different nature are key to making better decisions through the above stages. Current trends encourage the use of pure data-driven approaches coming from the framework of artificial intelligence and big data (e.g., artificial neural networks [5] and machine learning [6]). These techniques are rather systematic and do not require a deep knowledge on the systems where applied. Nevertheless, the process industry is not characterized by a scarce knowledge on the involved physicochemical processes. Indeed, detailed models for some equipment/plants have already existed for the last two decades (e.g., distillation columns [7]). Therefore, throwing out all this deep knowledge and just relying on the decisions inferred by data-driven machines would be risky.

These high-fidelity models have normally been used in offline simulations, for making decisions about the process design. This drawback is due to the usually high computational complexity and the relatively limited degrees of freedom to fit the actual plants. Therefore, there is still a lack of suitable models able for online prediction at almost all tactical levels of the automation pyramid: from real-time predictive simulation and optimization [8] to production planning and scheduling [3]. Therefore, the concept of plant *digital twin* contains a (virtual replica of actual assets that matches their behavior in real time), playing an important role in decision-support systems.

Consequently, many people in the process control community have been devoting efforts during the last decade to develop efficient and reliable models to support operators and managers in their decisions [9,10]. The preferred option is building models that combine as much physical information as possible/acceptable with relationships obtained from experimental data collected from the plant [11]. In this way, these hybrid or *grey-box models* get a high level of matching with the actual plant and, importantly, they get improved prediction capabilities as their outputs will at least fulfil the basic physical laws considered.

There are many good reviews and publications on process modelling, both covering first principles [12] and data-based approaches [13], but, in the authors' opinion, there is still a lack of methodology for the systematic development of grey-box models. In addition, several different approaches have been proposed in the literature to identify the "black part" of the grey-box model from input–output data. Among them, least-squares (LS) regression with regularization in the model coefficients [14,15] is one of the most used. Nonetheless, although the obtained models with this family of methods are quite balanced in terms of fitness to data and model complexity, the guarantees of physical coherence are under discussion, as they mainly depend on the quality and quantity of the collected data for regression.

This paper proposes a two-stage methodology which combines robust data reconciliation [16] with improved constrained regression. In the first stage, one gets estimations for all process variables that are coherent with some basic physical laws. Then, in a second stage of experimental customization, sophisticated constrained regression is used to get reliable experimental relationships among variables (that are not necessarily physical inputs and outputs measured in the plant), which will complete the first-principles backbone [17].

In this context, the authors of [18,19] already proposed a useful concept for black-box modeling: a machine-learning approach which automatically selects the suitable model complexity among a set of basis functions by balancing some model-complexity criteria with the fitness to regression data. Thus, this approach can be used in the second stage (constrained regression) of our proposed grey-box model building methodology [17]. The goal in this stage is to include as much process knowledge as possible (bounds on the model response, valid input domain, monotonic responses, maximum slopes and curvatures, etc.) as constraints in the regression. However, as these types of constraints on the model need to be enforced on infinitely many points belonging to the input–output domain, the regression becomes a semi-infinite programming problem [20] where a set of finite decision variables (the model parameters) but an infinite set of constraints arise. To tackle this problem numerically, the authors of [18] break down the problem into two parts: first, a relaxation of the original problem over a finite subset $x \in \mathcal{X}^*$ of the inputs domain $\mathcal{X}^* \subset \mathcal{X}$ (typically the points in the regression dataset) is solved via mixed-integer programming (MIP). Once a solution (i.e., values of the n model coefficients $\beta \in \mathbb{R}^n$) for this problem is gathered, a subsequent maximum-violation problem needs to be solved—that is, basically, a maximization of the constraints violation over all $x \in \mathcal{X}$, with the model fixed from the previous stage. If the constraints are violated at some point, this is added to the regression dataset and the procedure repeats until no constraint violation is detected. In the general case, this procedure involves solving nonlinear optimization problems (except the MIP one if candidate basis functions and constraints are chosen to be linear in decision variables). Moreover, the problem of finding the point where maximum constraint violation takes place is generally *nonconvex*. Therefore, a *global* optimizer is required to guarantee that the best fit fulfilling constraints have been found. Altogether, this means that the constrained-regression problem can be very time-consuming and computationally demanding.

To overcome this issue, in this paper, we propose casting the constrained-regression problem as a sum-of-squares (SOS) polynomial programming one, a technique that emerged ten years ago as the generalization of the semidefinite programming to the polynomial optimization over semi-algebraic sets [21]. The great advantage of SOS programming is the ability of guaranteeing constraint satisfaction for all $x \in \mathcal{X}$ (infinitely-constrained problem) without the need for fine-sampling datasets and via *convex optimization*. Although SOS programming is quite popular now within the automatic-control community, it has not penetrated too much into other fields of application. In particular, the authors only know one work on SOS programming applied to constrained regression [22], where explicit equilibrium approximations of fast-reacting species are sought via polynomial regressors. This work is particularly interesting because its authors outlined ideas similar to ours about grey-box modeling: they searched for reduced-order representations of kinetic networks which were physically consistent. In this paper, such an initial approach is extended to pose a constrained regression problem with guaranteed satisfaction of more advanced constraints than just model positivity, e.g., boundary constraints and limits on the model (partial) derivatives.

The rest of the article is organized as follows: Section 2 presents the problem formulation and its context in a formal way. The necessary definitions and preliminary results supporting the methodology and/or the examples are summarized in Section 3. Subsequently, Section 4 presents our proposed grey-box modeling methodology and Section 5 goes deeper into the SOS constrained regression. The benefits of the proposal are illustrated in Section 6 with two examples, one academic and another based on an industrial case study. Finally, the results are discussed in the last section, providing final remarks as well as an overview for possible extensions of the method.

2. Problem Statement

Let us assume that some first-principles equations of a process are available:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x(t), u(t), z(t), \theta), \quad h(x(t), u(t), z(t), \theta) = 0, \tag{1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ are *known* process inputs (manipulated variables or measured disturbances taking arbitrary values independently of the rest of the variables), $z \in \mathbb{R}^q$ are algebraic variables (with not-yet-fixed roles: some of them may be arbitrary inputs, some others may be functions of other variables, as discussed below), $\theta \in \mathbb{R}^\rho$ are model parameters (assumed to be constant) and $f(\cdot) \in \mathbb{R}^n$, $h(\cdot) \in \mathbb{R}^l$ can be nonlinear functions of their arguments.

Let us also assume that the above model is "incomplete", meaning that the system (1) is not fully determined by only the inputs u. Formally, this means that there are $q - n - l - m > 0$ variables $z^* \subset z$ assumed arbitrarily time-varying. However, some of them are not *unknown inputs* actually, but must be a function of other variables $z^*(x, u, z)$, representing the not well-known parts of the process. Therefore, assuming no significant unmodeled dynamics, let us assert that some additional equations $r(x, u, z) = 0$ need to be identified from process experimental data. Note that, although the first-principle model was "complete", it incorporates parameters θ whose value is not perfectly known. Therefore, stages of data collection and parameter identification are always present in practice, so that the model outputs fit those of the plant.

The classical way to approach this problem is to set up a certain functional structure for the equations $r(x, u, z, \theta_z) = 0$ searched with some parameters θ_z left for identification, and then formulate the following least-squares (LS) constrained (nonlinear) regression problem with N data samples collected at time instants $t_1, t_2, \ldots, t_N$ [23]:

$$\begin{aligned} &\min_{\theta, x_0} \sum_{t=t_1}^{t_N} \left\| \left(\hat{y}(t) - y(t)\right)/\sigma \right\|_2^2, \\ &\text{s.t.:}\ \frac{\mathrm{d}x}{\mathrm{d}t} = f(x(t), u(t), z(t), \theta), \quad x(0) = x_0, \\ &\qquad h(x(t), u(t), z(t), \theta) = 0, \quad r(x(t), u(t), z(t), \theta_z) = 0, \end{aligned} \tag{2}$$

where $\hat{y}$ are the process measured outputs, normalized by their respective mean or standard deviation σ, and $y = c(x, z)$ are their corresponding model predictions.

Note that, in many cases, x_0 may itself be part of the adjustable parameters (initial condition fitting).

This approach assumes that the chosen structure (implicitly in whichever equations are asserted in the expression of $r(\cdot) = 0$) for $z^*(x, u, z)$ is correct and the parameters are being estimated in the right way. Therefore, a proper selection of this structure is key to obtaining a good fitness with the real plant. However, the above assumptions and expectations may fail due to the following reasons:

- Proposing a good candidate structure often implies certain knowledge of the interactions and phenomena taking place in the process, which are normally too complex to model or are not well understood.
- Some parameters θ_z may not be identifiable within reasonable precision with a scarce set of measured variables y.

Furthermore, as problem (2) is normally a nonlinear dynamic optimization one, the computational demands can exploit rapidly with the size of the dataset N, with the complexity of the model equations and the time scales of the involved dynamics [24]. Therefore, it is not recommended to unnecessarily go deep into the physical phenomena as long as the final aim does not require it, e.g., different model requirements arise for the development of digital twins than for the ones for control or optimization purposes.

Hence, as the initially proposed structure $r(\cdot)$ will likely not be correct in a complex identification setup, the fit will need to be repeated with a modified candidate structure, following a very time-consuming trial-and-error procedure, without any guarantee of optimality of course.

In these cases, it may be sensible to combine what is known with certainty about the model, such as mass or energy balances, with data-driven equations obtained from measurements, representing those parts of the process which are unknown or complex to model. This results in a *grey box* model being a mixture of first-principles and data-driven equations [25].

Pursued Goal

As these grey-box models are built to be used for interpolation and extrapolation in control and optimization routines, the data-driven parts must be in coherence with the process physics [26]. Hence, some properties on z^* and/or in their derivatives (bounds, monotony, curvature, convexity, etc.) would like to be ensured, not only in the regression data but in the entire expected region of operation.

Machine learning is thus recalled to identify such “black parts” with suitable complexity, but, as these constraints on the model outputs need to be enforced on *infinitely many points* belonging to the input–output variables domain, the constrained regression becomes a semi-infinite programming problem [20] where there is a set of finite decision variables (model parameters) but an *infinite* set of constraints. To tackle this problem numerically, the authors in [18] developed the tool ALAMO which performs a kind of two-stage procedure, where, in the first stage, there is a relaxation of the original fitting problem over a finite subset of the input variables (any variable in (1) considered to be input into the black-box submodel $z^*(x, u, z)$) is solved via mixed-integer linear programming (MILP) for suitable model feature selection. Then, once optimal values for the model parameters are gathered, a second stage of validation is performed. This step consists of solving a maximum-violation problem that is basically a nonlinear maximization of the constraint violation over the whole input region, with the model fixed from the previous stage. Then, if such maximum violation is not zero, the point where it happens is added to the regression dataset and the procedure repeats. This can be a very time-consuming process involving the resolution of several mixed-integer and nonconvex optimization problems, depending on how many re-samples are required to ensure constraint satisfaction in the whole operating region.

In this paper, we propose an alternative way to efficiently tackle the grey-box modeling problem via data reconciliation and regression sub-models based on polynomials with guaranteed constraint satisfaction.

3. Materials and Methods

The following notation and previous results will be used in the proposed methodology.

3.1. Dynamic Data Reconciliation

Data reconciliation (DR) is a well-known technique to provide a set of estimated values for process variables over time that are as close as possible to the measured values inferred by sensors, but that are coherent with the process underlying physics: fulfilling some basic first-principle laws such as mass and energy balances [27]. The approach is based on the assumption that redundant information (duplicated sensors and/or existence of extra algebraic constraints among variables) is available. Hence, an optimization problem is set up to minimize some weighted sum of the deviations between the values measured over a past time horizon H until the current time t_c and their corresponding estimations (decision variables), subject to the (usually nonlinear) model equations plus any other inequality constraint to bound the unmeasured internal variables of the model (see Figure 1).

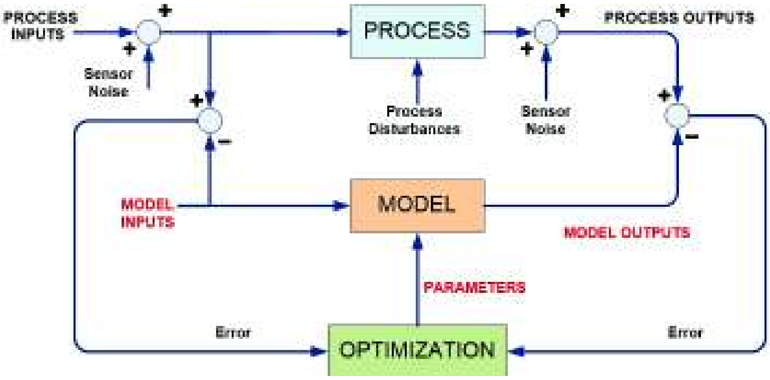

Figure 1. Standard DR scheme (decision variables highlighted in red).

The main obstacle of DR in industrial plants is the often scarce number of sensors, so that it is difficult to provide an acceptable level of redundancy with the collected data. Hence, in many cases, the approach is only able to calculate the unknown variables in the model, from perhaps some corrupted measurements that lead to incorrect estimations. In order to palliate this limitation, two courses of action are explored: (a) artificially increasing the system redundancy and (b) reducing the influence of gross errors in the measurements. Both aspects are covered in the following dynamic DR formulation:

$$\begin{aligned}\min_{u,z,w,\theta} \int_{t_c-H}^{t_c} & \left(K^2 \sum_{i=1}^{s} \left(\frac{|\epsilon_i(t)|}{K} - \log\left(1+\frac{|\epsilon_i(t)|}{K}\right)\right) + \sum_{j=1}^{r} w_j(t)^2 \right) \mathrm{d}t, \\ \text{s.t.: } & \frac{\mathrm{d}x}{\mathrm{d}t} = f(x(t),u(t),z(t),\theta), \quad x(t_c-H) = x_0, \\ & \frac{\mathrm{d}z^*}{\mathrm{d}t} = \omega_c \cdot z^*(t) + \kappa w(t), \quad z^*(t_c-H) = z_0^*, \\ & h(x(t),u(t),z(t),\theta) = 0, \quad g(x(t),u(t),z(t),\theta) \geq 0.\end{aligned} \tag{3}$$

In this (nonlinear) dynamic optimization problem:

- $\epsilon_i := (y_i - \hat{y}_i)/\sigma_i$, $\hat{y}$ being the process measured variables with $y = c(x,u,z) \in \mathbb{R}^s$ their analogies in the model, and σ are the sensors' standard deviations.
- $f(\cdot), h(\cdot), g(\cdot), c(\cdot)$ are vectors of possibly nonlinear functions comprising the model equations (f and h), the measured outputs (vector c) and additional constraints such as upper and lower bounds in some variables or/and their variation over time (vector g).
- z^* are the *free* model variables whose value will be estimated by the DR. These z^* are supposed to vary conforming a wide-sense stationary process w whose power spectral density is limited by bandwidths $\omega_c \in \mathbb{R}^+$. Bandwidths ω_c and gains κ can be set according to an engineering guess on the variation of the mean values of θ and via the sensitivity matrix of θ in y as proposed in ([28] Chap. 3), respectively. For instance, a limit case of $\omega_c \to 0$ and $\kappa \to 0$ would represent a constant parameter.
- $K \in \mathbb{R}^+$ is a user-defined parameter to tune the slope of the fair estimator [16], i.e., the insensitivity to outliers.

The initial states x_0 and z_0^* at $t = t_c - H$ may be either assumed known from the estimations provided at the previous reconciliation run, or also left as decision variables (with possible addition of some pondering of their deviations w.r.t. such previous estimations to the cost index).

Remark 1. *Note that inclusion of x_0 and z_0^* carry out all the system information from the past, thus avoiding the need of solving (3) for large H. In fact, z_0^* can be interpreted as "virtual measurements" for the unknown variables, increasing thus the system redundancy ([28] Chap. 3).*

3.2. Sum-of-Squares Programming

Sum-of-Squares (SOS) programming will lie at the heart of the proposed methodologies in this work. This section briefly reviews the basic concepts in it.

A multivariate real polynomial p in variables $x = (x_1, \ldots, x_n)$ and coordinate degree $d = (i_1, \ldots, i_n)$ is a linear combination of monomials in x with coefficients $c_i \in \mathbb{R}$

$$p(c,x) = \sum_{i \leq d} c_i \cdot x^i, \quad x^i = x_1^{i_1} \cdots x_n^{i_n}$$

and will be denoted by $p(c,x) \in \mathcal{R}_x$.

Definition 1 (SOS polynomials)**.** *An even-degree polynomial $p(c,x)$ is said to be SOS if it can be decomposed as a sum of squared polynomials $p(c,x) = \sum_i g_i(a,x)^2$, or, equivalently, iff $\exists\ Q(c) \succeq 0 \mid p(c,x) = z^T(x)Q(c)z(x)$, with $z(x)$ being a vector of monomials in x [29].*

Matrix Q is called the *Gram Matrix*, and checking if any $Q(c) \succeq 0$ (i.e., Q positive semidefinite) exists for a given p is a linear matrix inequality (LMI) problem [30]. In this way, checking if a polynomial $p(c,x)$ is SOS can be efficiently done via semidefinite (i.e., convex) programming (SDP) solvers [31].

The set of SOS polynomials in variables x will be denoted by Σ_x. E.g., stating that a polynomial $p(c, x)$, being c adjustable parameters, is SOS will be represented as $p(c, x) \in \Sigma_x$. Note that, evidently, all SOS polynomials are non-negative, but the inverse is not true [32].

Definition 2 (SOS polynomial matrices)**.** *Let* $F(c, x) \in \mathcal{R}_x^m$ *be an* $m \times m$ *symmetric matrix of real polynomials in* x*. Then,* $F(c, x)$ *is an SOS polynomial matrix if it can be decomposed as* $F(c, x) = H^T(a, x)H(a, x)$ *or, equivalently, if* $y^T F(c, x)y \in \Sigma_{x,y}$ *[33].*

An $m \times m$ SOS polynomial matrix F in variables x will be denoted by $F(c, x) \in \Sigma_x^m$. Analogously to SOS polynomials, if F is SOS, then $F(c, x) \succeq 0 \forall x$.

SOS Optimization

In the same way as certifying that a polynomial (matrix) is SOS, the minimization of a linear objective in decision variables β subject to some affine-in-β SOS constraints $F(\beta, x) \in \Sigma_x^m$ or positive-definiteness constraints $M(\beta) \succeq 0$ can be cast as an SDP problem. *Local* certificates of positivity on semialgebraic sets can be checked via the Positivstellensatz theorem [34]. The following lemmas are particular versions of such general result [35].

Lemma 1. *Consider a region* $\Omega(x)$ *defined by polynomial boundaries as follows:*

$$\Omega(x) := \{x \mid g_1(x) \geq 0, \ldots, g_q(x) \geq (0), k_1(x) = 0, \ldots, k_e(x) = 0\}$$

If polynomial multipliers $s_i(a_i, x) \in \Sigma_x$ *and* $v_j(b_j, x) \in \mathcal{R}_x$ *are found to be fulfilling*

$$p(c, x) - \sum_{i=1}^{q} s_i(a_i, x)g_i(x) + \sum_{j=1}^{e} v_j(b_j, x)k_j(x) \in \Sigma_x, \tag{4}$$

then $p(c, x)$ *is locally greater or equal to zero in* $\Omega(x)$*. Note that* $p(c, x)$ *can have an arbitrary (not necessarily even) degree, as long as* $\deg(s_i \cdot g_i)$ *and* $\deg(v_j \cdot k_j)$ *are even and greater than* $\deg(p)$*.*

Lemma 2. *A symmetric polynomial matrix* $F(c, x) \in \mathcal{R}_x^m$ *is locally positive semidefinite in* $\Omega(x)$ *if there exist polynomial matrices* $S_i(a_i, x) \in \Sigma_x^m$ *and* $V_j(b_j, x) \in \mathcal{R}_x^m$ *verifying:*

$$F(c, x) - \sum_{i=1}^{q} S_i(a_i, x)g_i(x) + \sum_{j=1}^{e} V_j(b_j, x)k_j(x) \in \Sigma_x^m. \tag{5}$$

By the previous discussion, checking the matrix condition (5) can be done via SDP optimization algorithms and SOS decomposition [31].

Lemma 3. *The set of (polynomial) matrix inequalities nonlinear in decision variables* $\beta := \{a, b, c\}$

$$R(c, x) \succ 0, \qquad Q(a, x) - S(b, x)^T R(c, x)^{-1} S(b, x) \succ 0 \tag{6}$$

with $Q(a, x) = Q(a, x)^T$ *and* $R(c, x) = R(c, x)^T$*, is equivalent to the following matrix expression:*

$$M(\beta, x) = \begin{bmatrix} Q(a, x) & S(b, x)^T \\ S(b, x) & R(c, x) \end{bmatrix} \succ 0. \tag{7}$$

This result is the direct extension of the well-known Schur Complement in the LMI framework [36] to the polynomial case. Condition (7) can be (conservatively) checked via SOS programming, as previously discussed in Lemma 2.

3.3. Polynomial Regression with Regularization

Our methodology proposal in this work will be compared to standard regularized regression [14,15], whose basic ideas are briefly summarized next.

Assume that a normalized (zero mean and $\sigma = 1$) set of input–output data $\mathcal{X}_\mathrm{T}\{X, Y\}$ for regression is available, where matrices X, Y have the N samples over time in columns, for the respective n_i input and n_o output variables in rows. Consider the candidate models for regression to be polynomials $p(c, x) \in \mathcal{R}_x^{n_o \times 1}$ of coordinate degree less than d in the inputs. Abusing notation, $P(c, X) \in \mathbb{R}^{n_o \times N}$ will represent the matrix resulting from evaluate $p(c, x)$ at the sampled points X.

Though polynomials are flexible candidate models, its use in machine-learning approaches is often limited to degrees $d \leq 3$ because they are very susceptible to overfitting, especially with a small number of samples. In order to palliate this drawback, a suitable *regularization* on the coefficients c of the high-degree monomials can be used, hence balancing the fitness to the training data with model complexity:

$$\min_{c} \quad \|Y - p(c, X)\|_l + \gamma \left\|\Gamma \cdot c^T\right\|_l, \tag{8}$$

where $\Gamma \in \mathbb{R}^{C_{n_i+d,n_i}}$ is a metaparameter matrix (usually diagonal) defining the regularization in each coefficient of c (i.e., its weighting structure in the objective function) and $\gamma \in \mathbb{R}^+$ is a tuning parameter to optimize training versus validation fit—see the next paragraph. Note that fitting errors as well as the regularization term may be formulated in any l-norm, typically the absolute ($l = 1$) or quadratic error ($l = 2$). In fact, the inclusion of bandwidth limits ω_c and random inputs w in (3) can also be understood as a type of regularization in a dynamic framework.

Of course, a further stage of cross validation of the "trained" model against a different dataset $\mathcal{X}_\mathrm{V}$ (or leave-one-out validation if few data are collected) is required. Thus, given a metaparameter Γ fixed a priori, the procedure to get the polynomial model which best fits the experimental data is solving (8) performing an exploration in γ (note that the evolution of the fitting error with γ can be non-monotonic, so bisection algorithms do not apply) and choosing the model which minimizes any desired weighted combination of the training and validation errors.

4. Proposed Modeling Methodology

Instead of a priori fixing a certain structure for the unknown equations $r(x, u, z, \theta_z)$ and solving (2) or, directly by brute force using a machine-learning approach to find a complete surrogate model $y = p(u, z)$ for the whole plant or individual equipment [37], we propose following the two-stage approach for grey-box modeling:

1. **Estimation.** With the partial model (1), use data reconciliation (3) to get coherent estimations over time of all variables x, u, z and parameters θ from process data.
2. **Regression.** Identify relationships between variables z^* with any x, u and/or z, and formulate a constrained regression problem to obtain algebraic equations $r(x, u, z) = 0$. Finally, these equations are added to the first-principles ones (1) in order to get a complete model of the process.

Stage 1 typically involves solving a nonlinear dynamic optimization problem, whose resolution can be done either via sequential or simultaneous approaches [24]: Depending on the problem structure, a combination of a dynamic simulator (e.g., IDAS, CVODES, etc. [38]) with an NLP optimization algorithm (rSQP like SNOPT [39] or an evolutionary one like spMODE [40]) can be a good choice, but modern optimization environments including algorithmic differentiation (like CasADi [41] or Pyomo [42]) offer excellent features in simultaneous (sparse) optimization problems, including automatic discretization of the system dynamics by orthogonal collocation, that facilitate the use of efficient interior-point NLP codes (e.g., IPOPT [43]). The outputs of this stage are coherent variables and parameter estimations according to the known physics of the process, including the estimations for the unknown inputs $z*$ whose hidden relations with other variables will be sought in Stage 2.

For Stage 2, different approaches from machine learning can be used. However, as mentioned in Section 2, not all can take advantage of the partial knowledge that one may have about z^*. Therefore, extra (local or global) conditions on the regression models are to be enforced in order to guarantee reliable interpolation, and also extrapolation to allow z^* taking values outside the range where experimental data was collected.

Although this concept is not novel [26], modern machine learning tools generalize the resolution of this constrained-regression problem. For instance, mixed-integer programming (MIP) and global optimization methods (e.g., BARON [44]) are employed to automatically select among a set of user-provided potential basis functions, a linear combination of those that provide the best fit taking into account such extra constraints to guarantee physical coherence. As briefly mentioned in Section 2, algebraic modelling environments like ALAMO offer a good support for this task using MIP solvers and adaptive-sampling procedures. However, their computational demands are high, even in the case where the MIP problem is restricted to be linear in decision variables.

Instead of the "ALAMO approach", an alternative way for solving Stage 2 via SOS constrained regression is proposed next. In this approach, the potential set of basis functions for regression are limited to be polynomial, but the resulting optimization problem is convex and extra constraints on the model response and/or in its derivatives are naturally enforced with full guarantee of satisfaction within a desired input–output region, no matter how many samples are to be fitted or which region was covered by the experiments. In this way, high-order polynomial regressors can be used with guarantees of well-behaved resulting function approximators, compared to most options in prior literature.

5. SOS Constrained Regression

Assume that a given dataset of N sampled (or estimated) values of some output variables (those z^* in Stage 2, Section 4) $Y \in \mathbb{R}^{n_o \times N}$ and some (x, u, z) inputs $X \in \mathbb{R}^{n_i \times N}$ is available. Abusing notation for simplicity, in this section, it is assumed that x represents any set of input variables x, u, z in Stage 2, Section 4. Thus, the problem to solve is building a polynomial model of coordinate degree at most d

$$z^* = p(c, x) \in \mathcal{R}_x^{n_o \times 1}, \qquad c \in \mathbb{R}^{n_o \times C_{n_i+d,n_i}}, \tag{9}$$

with the monomial coefficients c being parameters for regression, such that a measure of the error $\mathcal{E}$ (e.g., $\mathcal{L}_1$-regularized or least-squares) w.r.t. the data being minimized over a set of constraints on the model, locally defined in the parameter $c \in \mathcal{P}$ and input $x \in \mathcal{X}$ spaces:

$$\min_c \mathcal{E} := \|Y - P(c, X)\|_l\,, \tag{10}$$

$$\text{s.t.: } \Omega(\mathcal{X}) := \{c \in \mathcal{P} \mid g(c, x) \geq 0 \; \forall x \in \mathcal{X}\}. \tag{11}$$

The vector function $g(\cdot)$ here represent a general set of *polynomial* constraints to (locally) specify some desired robust features on the model response. Thus, Ref. (11) may range from standard (polynomial) bounds on z^* ensuring, for instance, non-negativity in $x \in \mathcal{X}$, to more complex bounds on its derivatives.

In this way, Refs. (10) and (11) are a semi-infinite constrained optimization problem, but it can be cast as a convex SOS problem if polynomials p and g are affine in decision variables c, $\mathcal{E}$ is linear in c and the region $\mathcal{X}$ is defined by polynomial boundaries on x. Details are given next for each of the entities involved in the above constrained regression problem.

Objective function. Note that $P(c, X)$ in (10) can be written as $P(c, X) = c \cdot F(X)^T$, where $F(X) \in \mathbb{R}^{C_{n_i+d,n_i} \times N}$ is the Vandermonde matrix containing all the monomials up to degree d evaluated at the sample points X. Then, as usually $N >> C_{n_i+d,n_i}$, the *economic* singular value decomposition $F(X) = S_1 V_1 D$ can be used to reduce the size of (10) [22]:

$$\mathcal{E} := \|Y - P(c, X)\|_l = \|YS_1 - cDV_1\|_l \,. \tag{12}$$

Now, the more common regressors based on the $\mathcal{L}_1$ and $\mathcal{L}_2^2$ norms (absolute error and least squares respectively) can be reformulated for SDP optimization as follows:

1. $\|YS_1 - cDV_1\|_1$ is enforced by:

$$\begin{aligned} &\min_{c,\tau} \sum_{i=1}^{n_o} \tau_i \\ &\text{s.t.: } \tau - YS_1 + cDV_1 \geq 0, \quad YS_1 - cDV_1 + \tau \geq 0, \quad \tau \in \mathbb{R}_+^{n_o}. \end{aligned} \tag{13}$$

2. Using Lemma 3, $\|YS_1 - cDV_1\|_2^2$ is enforced by:

$$\begin{aligned} &\min_{c,\tau} \tau \\ &\text{s.t.: } \begin{bmatrix} \tau & YS_1 - cDV_1 \\ S_1^T Y^T - V_1 D^T c^T & I \end{bmatrix} \succeq 0. \end{aligned} \tag{14}$$

Constraints on the input/output domain. Constraints on z^* are introduced in (11) with g of the form:

$$g(c, x) = \beta_l^T p(c, x) + k_l(x), \tag{15}$$

where vector β_l weights the model outputs and $k_l(x)$ is a vector of polynomial user-defined functions in x. Hence, depending on the degree of the components of k_l, upper and lower limits for z^* (zero-order constraints) can be stated, or more complex (higher order) constraints on the feasible output region too. Moreover, using SOS programming and Lemma 1, (11) with (15) can be locally enforced in $x \in \mathcal{X}$ as long as $\mathcal{X}$ is defined by polynomial boundaries.

Constraints on the model derivatives. Model slopes and curvatures w.r.t. x get the following functional form for g in (11):

$$g(c, x) = \alpha_d^T \nabla_x p(c, x) + k_d(x), \tag{16}$$

$$g(c, x) = \mathrm{A}^T \nabla_x^2 p(c, x) \mathrm{A} + B(x), \tag{17}$$

where ∇_x stands for the gradient operator w.r.t. x. ∇_x^2 denotes the Hessian matrix and α_d, $k_d(x)$, $B(x)$ and A are user-defined elements with suitable dimensions. As derivatives of polynomials are also polynomials, (11) with (16) and/or (17) can be locally checked for SOS in $x \in \mathcal{X}$ using the results in Section 3.2.

For example, suppose that *global* convexity is to be ensured in a regression candidate model $p(x_1, x_2) = c_0 + c_1 x_1 + c_2 x_2 + c_3 x_1 x_2^2 + c_4 x_1^2 x_2$. The Hessian matrix for it is:

$$\mathrm{H}(c, x_1, x_2) = \begin{bmatrix} 2c_4 x_2 & 2c_3 x_2 + 2c_4 x_1 \\ 2c_3 x_2 + 2c_4 x_1 & 2c_3 x_1 \end{bmatrix}.$$

The classical approach to ensure convexity in p is forcing the determinant of H to be non-negative. Unfortunately, $-c_3 c_4 x_1 x_2 - c_4^2 x_1^2 - c_3^2 x_2^2 \geq 0$ is nonconvex in c and would transform (10) and (11) into a quadratically constrained regression problem. However, global convexity on p can be easily enforced using SOS programming by just setting (11) to:

$$\begin{bmatrix} 2c_4 x_2 & 2c_3 x_2 + 2c_4 x_1 \\ 2c_3 x_2 + 2c_4 x_1 & 2c_3 x_1 \end{bmatrix} \in \Sigma_{x_1, x_2}^2. \tag{18}$$

Boundary constraints. Boundary conditions (Dirichlet, Neumann, Robin or Cauchy) require equality constraints in (11), enforced over some $x_i = \mathrm{x}_i^* \in \mathcal{X}$. In this case, the general representation for g is:

$$g(c,x) = \left(\beta_b^T p(c,x) + \alpha_b^T \nabla_x p(c,x) + \kappa^T \nabla_x^2 p(c,x)\kappa + k_b(x)\right)|_{x_i=\mathrm{x}_i^*} \tag{19}$$

and their local enforcement in $x \in \mathcal{X}$ can be proven again by Lemma 1 and SOS programming. Note that $g(c,x) = 0$ is equivalent to check $g(c,x) \in \Sigma_x$ jointly with $-g(c,x) \in \Sigma_x$. Moreover, $g(c,x) \in \Sigma_x$ is equivalent to $g(c,x) - s(x) = 0$ and $s(x) \in \Sigma_x$.

6. Illustrative Examples

Two examples to show the potential benefits of our proposed methodology are presented in this section. The first one is a simple academic example with artificially created data to face the SOS constrained regression against least-squares (LS) polynomial fitting with regularization, a basic approach in the machine-learning literature. The second one is an industrial example of grey-box modeling in an evaporation plant. In particular, the example shows how to build a model for the heat-transfer in a series of exchangers which suffer from fouling due to depositions of organic material.

6.1. SOS Constrained Regression versus Regularization

The purpose of this simple example is to demonstrate the improved features of our physics-based regression approach w.r.t. the "blind" regularization summarized in Section 3.3.

Assume that a dataset of 20 samples is collected from an ill-known SISO process, and that a polynomial model for it is to be sought. For building such model, the data is randomly divided in two sets, $\{X_\mathrm{T}, Y_\mathrm{T}\}$ with 11 samples for training and $\{X_\mathrm{V}, Y_\mathrm{V}\}$ with the rest for validation:

$$\begin{aligned}
X_\mathrm{T} &= [0.6978, 1.0811, -0.5991, 0.648, -0.3354, 1.3677, 1.3317, -0.9742, 0.4538, 0.329, -1.4],\\
Y_\mathrm{T} &= [0.1917, 0.5362, 0.554, 0.1629, 0.1718, 1.2121, 1.4415, 1.3438, 0.2583, -0.0378, 1.5],\\
X_\mathrm{V} &= [1.4798, -0.9409, -0.7277, -1.5231, 1.7593, 1.13, -0.0821, 0.5573, 0.1789],\\
Y_\mathrm{V} &= [1.64, 1.173, 0.8318, 1.6, 1.706, 0.64, 0.027, 0.2193, 0.1025].
\end{aligned}$$

Looking at the plotted data in Figure 2, one may infer that the "obscure" process could be convex, so fitting with quadratic candidate models would be satisfactory enough. However, this is not the case as we will explain later, and note that this visual inspection would not be possible in high-dimensional systems. Therefore, for the shake of better fitness, the candidate model will be a polynomial of, at most, degree $d = 8$:

$$p(c,x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + c_4 x^4 + c_5 x^5 + c_6 x^6 + c_7 x^7 + c_8 x^8. \tag{20}$$

As expected, using classical unconstrained LS with (20) and just 11 samples for training leads to a totally useless overfitted model (orange curve in Figure 2) with two local minima and drastically falling down around $|x| \geq 1.5$.

6.1.1. Least Squares with Regularization

In order to avoid overfitting, regularization in c is recalled (Section 3.3). In this approach, the user must set the metaparameter Γ a priori and then perform an exploration in γ to find the best fitting for such Γ. This means that the performance in this approach is very tailored to have a good guess for Γ. Unfortunately, the metaparameter cannot be easily related to any physical insight, but only to reduce the influence of some non-preferred monomials, normally the higher-degree ones. Following this idea, two typical alternatives for the metaparameter were tested:

$$\text{[M-1]}\ \ \Gamma = [0,0,0,1,1,10,10,100,100]^T; \qquad \text{[M-2]}\ \ \Gamma = [0,0,1,e^2,e^3,e^4,e^5,e^6,e^7]^T.$$

Note that coefficients of the zero-order and linear terms are not penalized in both alternatives (at least the best linear prediction will be found in the worst case). Moreover, the quadratic term is also freed due to such intuition of convexity from data visual inspection, whereas the higher-order monomials are progressively penalized. In M-2, the usual exponential penalty with the monomials degree is set in order to balance fitness to data with model complexity.

After exploration in γ for both setups, the model with less total fitting error (chosen to be training plus validation errors) is found at $\gamma = 0.4$ with the chosen exploration granularity. The best model (coefficients below $c < 10^{-4}$ are disregarded) obtained with the metaparameter choice [M-1] is a polynomial of degree 7 (dashed blue curve in Figure 2), whereas [M-2] is a polynomial of degree 5 (dotted pink curve in Figure 2). Table 1 gives the fitting error for these "best" models, as well as some values resulting from the exploration in the regularization scaling parameter *gamma*.

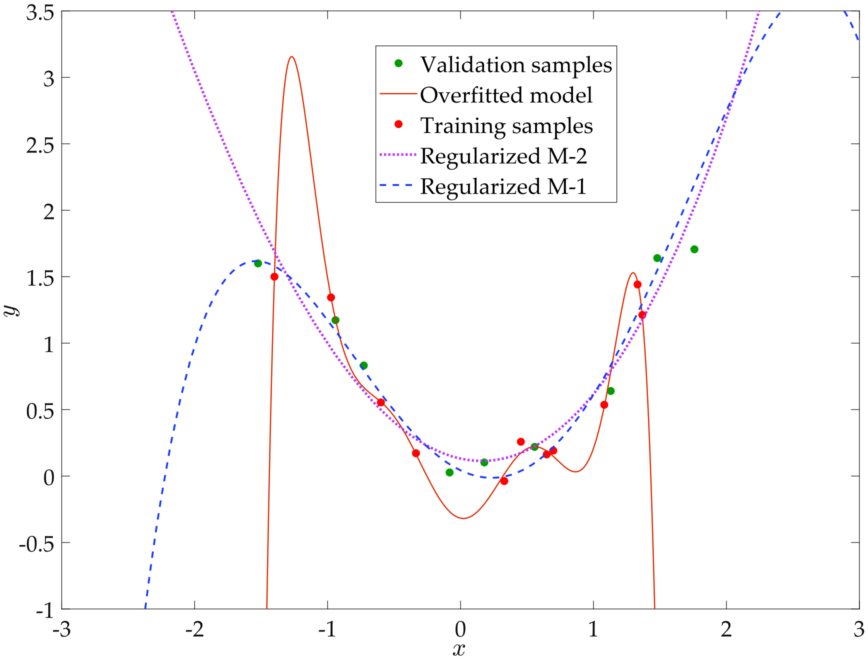

Figure 2. Sampled data and polynomial models fitted by standard LS approaches.

Table 1. Exploration in γ for M-1 and M-2 regularizations.

Γ	γ	Training Error	Validation Error	Total
M-1	0.01	0.1517	1.84	2
	0.1	0.206	0.366	0.572
	0.4	0.218	0.324	0.541
	1	0.23	0.372	0.602
	10	0.34	0.49	0.83
	100	0.416	0.55	0.967
M-2	0.001	0.184	1.021	1.2
	0.01	0.231	0.834	1.065
	0.5	0.405	0.422	0.826
	2	0.63	0.42	1.05
	10	1.671	1.698	3.37

Remark 2. *Looking at Figure 2, the model obtained by the usual exponential regularization [M-2] is preferable to the one obtained by the ad hoc [M-1] because it is quite symmetric and convex (at least in the depicted region), so it would be more "reliable" a priori for extrapolation in* $\mathcal{X} := \{x : 2 < |x| < 3\}$. *However, note that simple*

visual inspection is not available for high-dimensional systems. Thus, without visual information, one would have chosen the model by [M-1], as it is the one which best fit the data.

6.1.2. SOS Constrained Regression

Alternatively to the "blind" regularization, some desired features with physical insight on the model response could have been searched. Thus, as an initial idea, non-negativity and convexity were forced on (20) via SOS constrained regression (LS objective, Section 5) with the following constraints:

$$p(c,x) = c_0 + c_1x + c_2x^2 + c_3x^3 + c_4x^4 + c_5x^5 + c_6x^6 + c_7x^7 + c_8x^8 \quad \in \quad \Sigma_x, \tag{21}$$

$$\frac{\mathrm{d}^2 p(c,x)}{\mathrm{d}x^2} = 2c_2 + 6c_3x + 12c_4x^2 + 20c_5x^3 + 30c_6x^4 + 42c_7x^5 + 56c_8x^6 \quad \in \quad \Sigma_x. \tag{22}$$

The convex polynomial found to best fit the training data ($\mathcal{E} = 0.226$) incurs in a high error on the validation data ($\mathcal{E} = 14.46$). By inspecting the modeling error with the data points (visual inspection omitted, as this possibility is hardly available in models with multiple inputs) it was found that the highest deviations appear mainly around the boundaries of the training region. Thence, it might be inferred that the generating process flattens far away from the origin, so it is probably nonconvex (or not strongly convex at least).

A simple way to find a model whose response fits better with this insight of flatness in extrapolation is to set up *local* upper and lower bounds on $p(c,x)$: $\overline{y} - p(c,x) \geq 0 \forall x \in \{x : |x| < 3\}$; $p(c,x) - \underline{y} \geq 0 \forall x \in \{x : 2 < |x| < 3\}$ or better by locally bounding the slope to small values in $\psi := \{x : 2 < |x| < 3\}$. Using Lemma 1, this last condition is enforced by the following SOS constraints:

$$\begin{aligned} p(c,x) - s_1(a_1,x)\cdot(3^2 - x^2) &\in \Sigma_x, \\ 0.3 - \frac{\mathrm{d}p(c,x)}{\mathrm{d}x} - s_2(a_2,x)\cdot(3^2 - x^2) - s_3(a_3,x)\cdot(x^2 - 2) &\in \Sigma_x, \\ \frac{\mathrm{d}p(c,x)}{\mathrm{d}x} + 0.3 - s_4(a_4,x)\cdot(3^2 - x^2) - s_5(a_5,x)\cdot(x^2 - 2) &\in \Sigma_x, \end{aligned} \tag{23}$$

with $s_i(a_i,x) \in \Sigma_x$ being SOS polynomial multipliers whose highest degree is $d \geq 6$, as $p(c,x)$ can be of degree 8. Note that local non-negativity of p on $\mathcal{X} := \{x : |x| < 3\}$ is also enforced, as there is no need to force global positivity outside the region considered for extrapolation, thus reducing conservatism.

The model obtained with this approach is the solid orange curve in Figure 3, labelled as [P-1]. This desired response was obtained with a total regression error (training plus validation) of $\mathcal{E} = 0.41$, beating by 25% the best fit obtained by the regularization approach.

Nonetheless, the response shows several local minima in $\mathcal{X}$. If this surrogate model is to be integrated in a larger grey-box model for real-time optimization purposes, getting a quasi-monotonous model (single global minimum) could be more interesting than achieving the lowest fitting error, in order to reduce the probability of getting stuck in local optima with gradient-based NLP solvers. Several alternative ways are available to handle this issue via SOS constrained regression:

[P-2] Positive curvature in $\mathcal{X}$, tending to zero when $x \in \psi$ (dashed-dotted pink curve in Figure 3):

$$p(c,x) \geq 0,\ \frac{\mathrm{d}^2 p(c,x)}{\mathrm{d}x^2} \geq 0,\ \forall x \in \mathcal{X}; \qquad \frac{\mathrm{d}^2 p(c,x)}{\mathrm{d}x^2} \leq 0.25\ \forall x \in \psi,$$

[P-3] Upper bound on p in $\mathcal{X}$ and bounded negative curvature in $x \in \psi$ (dashed green curve):

$$2.5 \geq p(c,x) \geq 0\ \forall x \in \mathcal{X}; \qquad 0 \geq \frac{\mathrm{d}^2 p(c,x)}{\mathrm{d}x^2} > -0.8\ \forall x \in \psi,$$

[P-4] Symmetrically bounding the slope between two values in $x \in \psi$ (dotted blue curve):

$$p(c,x) \geq 0 \,\forall x \in \mathcal{X}; \qquad 0.1 < \frac{\mathrm{d}p(c,x)}{\mathrm{d}x} < 0.6 \,\forall x \in \{2 \leq x \leq 3\};$$
$$-0.1 \geq \frac{\mathrm{d}p(c,x)}{\mathrm{d}x} \geq -0.6 \,\forall x \in \{-2 \geq x \geq -3\}.$$

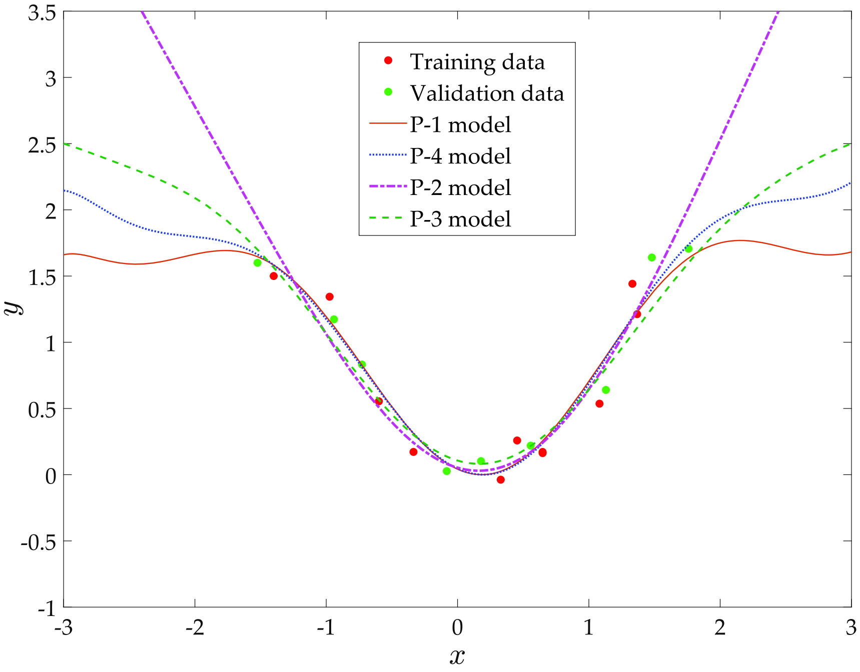

Figure 3. Sampled data and polynomial models fitted by SOS constrained regression.

As can be seen in Figure 3, the three approaches (P-2 to P-4) get *quasi-monotonous* surrogate models which are suitable for optimization purposes. The total regression error is quite similar in all the approaches (Table 2) and the small differences between them could be just a product of sheer luck (fitting the sensor noise). Thus, the choice of one over the other would only depend on the engineer physical intuition.

Table 2. Least-squared errors for the SOS constrained approaches.

Constraint	Training Error	Validation Error	Total
P-1	0.26	0.15	0.41
P-2	0.31	0.364	0.674
P-3	0.372	0.255	0.627
P-4	0.257	0.144	0.4

Remark 3. *Note that the standard LS regularization was not able to find these more feasible models obtained with the SOS approach, at least with the tested values for the metaparameter* Γ. *Anyway, although it may be found, there is no clear and direct relation between* Γ *and the features desired in the model response.*

6.2. Modeling the Heat-Transfer in an Evaporation Plant

In this example, we make use of the proposed methodology in Section 4 to build up a grey-box model for a multiple-effect evaporation plant of a man-made cellulose fiber production factory.

The plant is formed by several evaporation chambers and some heat exchangers in serial connection, a mixing steam condenser and a cooling tower, forming a multiple-effect evaporation system. See Figure 4, where individual equipment have been lumped together for confidentiality reasons and due to the lack of measurements in between. The plant receives a liquid input, mixture of water with chemical components and leftovers of organic material. The goal is to concentrate the liquid by removing a certain amount of water.

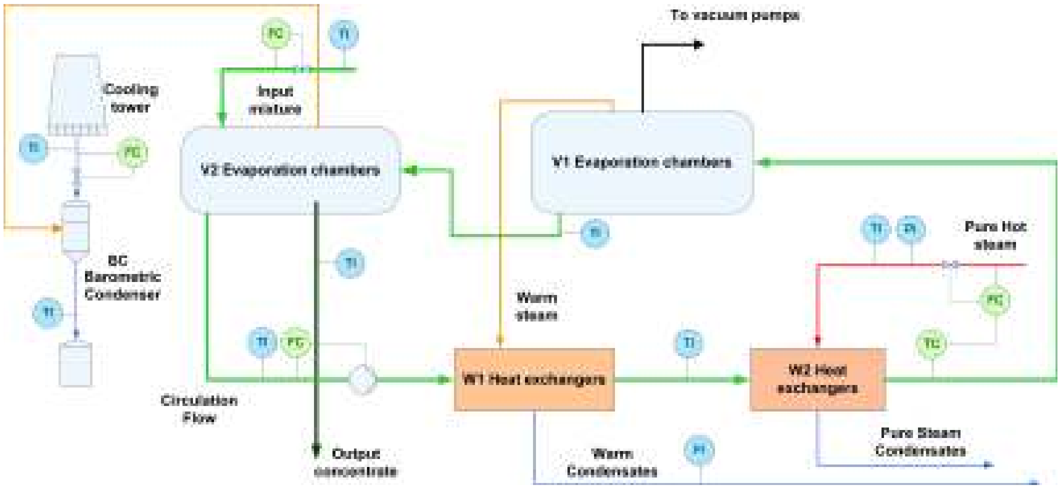

Figure 4. Simplified schema of the evaporation plant with existent instrumentation.

The process operates as follows: the liquid enters the system by chamber V_2 and then goes sequentially through the sets of heat exchangers W_1 and W_2 to increase its temperature up to a desired setpoint. In W_1, the temperature rise is achieved from saturated-steam flows recirculated from the evaporation chambers V_1. Then, the temperature setpoint is reached in W_2 thanks to a fresh steam inlet from boilers. Afterwards, the hot liquid enters sequentially into the low-pressure set of chambers V_1, where a partial evaporation of water takes place. The remaining evaporation is achieved in the last chambers V_2 thanks to the pressure drop in the mixing condenser BC, linked to the cooling tower. Finally, part of the concentrated solution leaves the plant by V_2 and the rest mixes again with the inlet, being recirculated to the heat exchangers.

6.2.1. Stage 1: Estimation

Our modeling approach starts from a nonlinear set of equations of the plant in steady state, obtained from first principles. These equations have been omitted here for brevity, but the reader is referred to the previous works of the authors [45,46] to get a detailed description of both the plant and the physical model equations. Then, in the *estimation* phase (Stage 1 of the proposed methodology), DR is performed to "clean" the process data from incoherent sensor values and to get suitable estimates for the internal-model variables and time-varying parameters, in particular for the average heat-transfer coefficient $UA(t)$ in the lumped heat exchangers. Note that this time-varying parameter depends on the conduction and convection effects plus the exchange surface, values that are not precisely known or complex to model.

Here, the focus is on UA because an accurate modeling of the long-term fouling dynamics in the heat-exchangers pipes is key for a realistic optimization of the operation, and the right scheduling of the maintenance tasks. Indeed, this issue is shared with other industrial systems like furnaces or catalyst deactivation in chemical reactors. All have in common a system-efficiency degradation, which may be palliated or worsened by the way the equipment operates.

Thus, a set of experiments were performed on site, running the plant in different operating conditions (setting different values for the main control variables: the circulation flow and the temperature setpoint). Moreover, in order to get significant information from the actual fouling process, the plant historian for several months of operation (including some stops for cleaning) has been also provided as experimental data for reconciliation. Figure 5 shows the estimated UA for exchangers W_1, provided by the DR (details omitted for brevity).

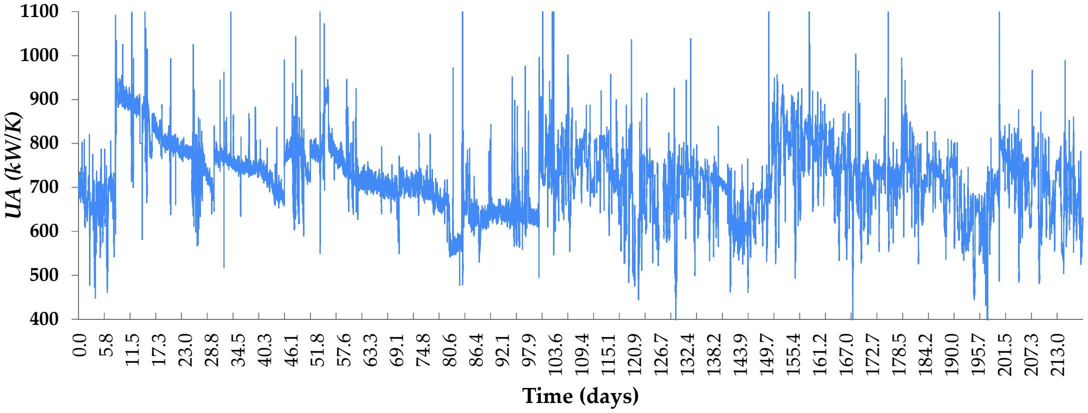

Figure 5. Estimated values of UA over seven months of operation.

6.2.2. Stage 2: Regression

The objective now in the *regression* stage is to build up a polynomial regression model $UA = p(c, F, t)$ to link/predict the heat-transfer coefficient UA with the circulating flow through the exchangers F and with the time t that the plant is in operation since last cleaning.

The first issue arises when selecting the samples for training and validation. Although the recorded dataset of seven months with a sampling time of 5 min may look huge, the quality of data is much more important that the quantity of samples. In addition, in this case, the plant was usually working at high circulating flows, except in the few experiments executed on purpose and in particular situations (product changeovers). Therefore, lots of data for the plant operating in a local region are available, but a significant amount of information of the convection and fouling behaviors at medium/low flows is missing.

Note importantly that, although there is no major computational issue in performing regularized or SOS constrained regression with hundreds of data, if lots of such data are agglomerated around the same operating point, the fitted model might specialize too much in such region, as the model structure for regression will not likely contain the same nonlinearities as the actual plant which generated the data. Hence, the prediction capabilities out of this region can be seriously compromised with such a model. Therefore, the data points must be “triaged” according to their degree of uniqueness (data containing almost-redundant information should get lower weights in the regression problem, or be directly removed from the training set) in order to prevent this possible model bias due to strong non-uniform data densities.

Consequently, after inspecting and analyzing the plant historian, we ended up with a selected subset of 22 samples (UA, F, t) for training plus 20 samples for validation. These data, depicted in Figure 6, contains nearly all the information available in the feasible region of operation:

$$\Omega := \{F, t \in \mathbb{R}^+ : 100 \leq F \leq 200 \text{ m}^3/\text{h},\ t \leq 60 \text{ days}\}. \tag{24}$$

As it can be observed by simple visual inspection, there are many samples covering Ω at high flows, but there is a significant lack of information at lower flows, especially when the plant works after cleaning and when it is in operation for more than 40 days.