A Sum-Of-Squares Constrained Regression Approach for Process Modeling

José Luis Pitarch* Antonio Sala César de Prada*,*****

**Systems Engineering and Automatic Control DPT, EII, Universidad de Valladolid.*
C/Real de Burgos s/n, 47011, Valladolid, Spain, (jose.pitarch@autom.uva.es)
***Instituto Universitario de Automática e Informática Industrial, Universitat Politècnica de València.*
Camino de Vera S/N, 46022, Valencia, Spain, (asala@isa.upv.es)
**** Institute of Sustainable Processes (IPS), Universidad de Valladolid.*
C/Real de Burgos s/n, 47011, Valladolid, Spain, (prada@autom.uva.es)

Abstract: Combining empirical relationships with a backbone of first-principle laws allow the modeler to transfer the available process knowledge into a model. In order to get such so-called grey-box models, data-reconciliation methods and constrained regression algorithms are key to obtain reliable process models that will be used later for optimization. However, the existent approaches require solving a semi-infinite constrained regression nonlinear problem, which is usually done numerically by an iterative procedure alternating between a relaxed problem and an a posteriori check for constraint violation. This paper proposes an alternative one-stage efficient approach for polynomial regression models based in sum-of-squares (convex) programming. Moreover, it is shown how several desirable features on the regression model can be naturally enforced in this optimization framework. The effectiveness of the proposed approach is illustrated through an academic example provided in the related literature.

Keywords: Constrained regression, Process models, Grey-box models, SOS programming.

1. INTRODUCTION

The increasing levels of digitalization motivated by the concepts stated in the so-called Industry 4.0 (Davies, 2015) force the companies to search for methods to transform raw data in useful information. This information is expected to significantly impact in the decision-making processes at all levels in the companies.

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, they base their operation (and, therefore, economic margins) 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 not very variable in terms of raw materials or product demands, whereas environmental restrictions are tighter every year. This causes fierce competition. In this context, what Industry 4.0 can bring in terms of improved efficiency can be summarized in three main aspects: 1) transform data in information via the definition of suitable indicators in real time (Kujanpää, et al., 2017), 2) use the information to optimize the plant operation (Krämer and Engell, 2017) and 3) improve coordination between plants operation and its link with the production planning and scheduling (Palacín, et al., 2018).

Suitable models, possible of different nature, are required through these steps to achieve the expected goals. Current trends in Industry 4.0 push to the excessive use of pure data-driven approaches coming from the world of artificial intelligence and big data (e.g., artificial neural networks (Afram and Janabi-Sharifi, 2015) and machine learning (Witten et al., 2016)). These approaches have been demonstrated useful to extract information from systems whose behaviour is practically unknown and/or have a large variability, e.g. in companies devoted to IT services (Golovin, et al., 2017). However, the process industry is characterized neither by these levels of uncertainty nor by a scarce knowledge in the involved physicochemical processes. Indeed, quite detailed models for some equipment/plants already exist since around 10-20 years (e.g. distillation columns (Olsen, Endrestøl and Sira, 1997)). However, because of their complexity and difficulties to match the actual plants, usually these models have been used in offline simulation for taking decision about process design, or for very specific control purposes. Therefore, there is still lack of suitable models, able for prediction, almost at all levels of the automation pyramid: from real-time plants optimization to production planning and scheduling.

In consequence, several researchers 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 (Kar, 2015), (Kalliski et al., 2019). There is a quite defined consensus around the option of developing models which combine as much physical information as possible/suitable with relationships obtained from real data collected from the plant (Zorzetto et al., 2000). In this way, these so-called grey-box models get a high level of matching with the actual plant in terms of current operation regimes and, importantly, are quite confident for ex-

♦ This research received funding from the European Union, Horizon 2020 research and innovation programme under grant agreement No 723575 (CoPro), and from EU plus the Spanish Ministry of Economy, grant DPI2016-81002-R (AEI/FEDER).

trapolation (i.e. prediction capabilities), as their outputs will never violate basic first-principle laws.

Several approaches have been proposed (or reused from the literature) to identify such “black” part of the grey-box models from input-output data. Among them, least-squares (LS) regression based (linear or nonlinear) is the most employed, but statistical methods like principal-component analysis based (Wang, Sun and Jia, 2016) are also very popular. Nonetheless, in the author’s opinion, one of the best approaches is combining robust data reconciliation (Llanos, Sanchéz and Maronna, 2015) with constrained regression (Cozad, Sahinidis and Miller, 2015). In this way, one firstly gets estimations of unmeasured variables that are coherent with basic physical principles, so that, in combination with subsequent LS constrained regression for instance, more reliable experimental relationships are obtained among variables that are not necessarily measured inputs and outputs (de Prada et al., 2018).

In this framework, (Wilson and Sahinidis, 2017) proposed a novel and very useful concept (based on a software tool for black-box modeling: ALAMO) which, given a dataset, automatically selects the right model complexity among a set of basis functions by balancing the Akaike information criteria with the regression performance (model fitness to experimental data). ALAMO was already used in (Cozad, Sahinidis and Miller, 2015) to cope somehow with the problem of grey-box model building via constrained regression. The idea is to include the a priori modeler knowledge (bounds on the model response, valid input domain, model slope and curvature, etc.) as constraints in the LS regression. However, as this type of constraints on the model need to be enforced on infinitely many points belonging to the input-output variables domain, the above LS regression becomes a *semi-infinite programming problem* (Reemtsen and Rückmann, 1998) where we have a set of finite decision variables (the model parameters) but an infinite set of constraints.

To tackle this problem numerically, the authors in (Cozad, Sahinidis and Miller, 2015) propose a two-step iterative procedure where, in the first phase, a relaxation of the original problem over a finite subset of the input variables $x \in X^l \subset \mathcal{X}$ is solved. Once a solution (i.e. values of the model parameters β^l) for this problem is gathered, a second phase of validation is performed. This step consists of solving a *maximum violation problem* that is basically a maximization of the constraint violation over $x \in \mathcal{X}$ with the model fixed from the previous step. In the general case both steps involve solving nonlinear optimization problems (except perhaps the first one with SISO models or simple linear constraints) and, what is worse, the second one is generally nonconvex. Thus, if one prioritizes speed versus chance of reaching global optimality[1], gradient-based interior point algorithms can be used. Otherwise, a global optimizer is required. See the above cited reference for more details.

[1] Note that the probability of falling stuck in a local optimum increases with the number of experimental samples to fit.

Since 25 years ago approximately, semidefinite (convex) programming (SDP) has become the main tool to solve control synthesis problems that were intractable in the past (Boyd et al., 1994). More recently, the sum-of-squares (SOS) programming emerged as a generalization of the semidefinite one to perform polynomial optimization over semi-algebraic sets (Parrilo 2003). Although it has been quite used within the automatic control community in stability and disturbance analysis, control and observation of nonlinear systems (Henrion and Garulli, 2005), (Pitarch and Sala, 2014), (Pitarch et al., 2018), it has not penetrated too much in other fields of application. In particular for SOS programming applied to constrained regression, the authors only know the work by Nauta et al. (2007), where explicit equilibrium approximations of fast-reacting species are sought. This work is particularly interesting because the authors share the same outlined ideas about grey-box modeling: they searched for reduced-order representations of kinetic networks which were physically consistent. To ensure so, the polynomial approximations were constrained to be positive within a local range of validity in the regression problem.

The goal of this paper is to highlight how SOS optimization can be used to solve the above presented semi-infinite constrained regression problems in one step, drastically reducing the computational load with respect to the existing two-phase procedures (at least for problems with a reasonable number of regression variables involved). Moreover, we aim to go beyond the work of Nauta et al. (2007), by extending the type of constraints that can be imposed on the regression model in order to get desirable features such as (local) convexity, monotony, smoothness, etc.

The rest of the article is organized as follows: Next, some necessary definitions and lemmas are recalled to support the proposal, presented later in Section 3. Then, the effectiveness of the approach is illustrated in Section 4 with a toy example adapted from the literature. Finally, some remarks as well as an overview of possible further extensions are outlined in the last section.

2. SOS PROGRAMMING

On the following, we recall some important definitions and preliminary results on sum-of-squares polynomials.

Definition 1: SOS polynomials. An even-degree polynomial $p(x) \in \mathcal{R}_x$ in variables x is SOS iff $\exists Q \succcurlyeq 0$ such that $p(x) = z^T(x)Qz(x)$, with $z(x)$ being a vector of suitable monomials in x. Q is called the “Gram Matrix” and checking if any $Q \succcurlyeq 0$ exist for a given p is a linear matrix inequality (LMI) problem (Parrilo, 2000).

In this way, if the polynomial p is affine in decision variables (typically its coefficients), it can be checked for SOS via efficient SDP solvers (Papachristodoulou, et al., 2013). Evidently, all SOS polynomials are nonnegative, but the inverse is not true. From now on, the set of SOS polynomials is denoted by the symbol Σ_x.

Definition 2: SOS polynomial matrix. Let $F(x) \in \mathcal{R}_x^n$ be an $n \times n$ symmetric polynomial matrix of degree $2d$ in x. Then,

$F(x)$ is an SOS polynomial matrix if $F(x) = H^T(x)H(x)$, or equivalently if $y^T F(x) y \in \Sigma_{x,y}$ (Scherer, 2005). Analogously to the previous case, if F is an SOS polynomial matrix, $F(x) \succcurlyeq 0\ \forall x$. The set of $n \times n$ symmetric SOS polynomial matrices is denoted by the symbol Σ_x^n.

SOS programming. In the same way as certifying that a polynomial $F(x)$ is SOS, minimization of a linear cost index in decision variables β subject to SOS constraints $F(x,\beta) \in \Sigma_x$ or SOS positive-definiteness constraints $F(x,\beta) \in \Sigma_x^n$ with F affine in β can also be cast as an convex SDP problem. Scalar linear constraints on β can, too, be easily incorporated (they can be considered as zero-degree polynomials).

Local positivity of polynomials on semialgebraic sets can be checked via the well-known Putinar's Positivstellensatz theorem (Putinar, 1993). The following lemmas are a reduced version of such result (Pitarch, 2013).

Lemma 1. Consider a region defined by polynomial boundaries $\mathcal{X} := \{x | g_1(x) \geq 0, \dots, g_l(x), k_1(x) = 0, \dots, k_r(x) = 0\}$. If polynomial multipliers $s_i(x) \in \Sigma_x$ and $v_j(x) \in \mathcal{R}_x$ can be found fulfilling:

$$p(x) - \sum_{i=1}^{l} s_i(x) g_i(x) + \sum_{j=1}^{r} v_j(x) k_j(x) \in \Sigma_x \qquad (1)$$

Then $p(x)$ is locally greater or equal than zero in $\mathcal{X}$. ∎

Remark. Note that, thanks to Lemma 1, we can also check local positivity of odd-degree polynomials via SOS programming, by just appropriately choosing the degree of multipliers s, v such that $deg(s(x)g(x))$ and $deg(v(x)k(x))$ is even and greater than $deg(p(x))$.

Lemma 2. The polynomial matrix $F(x)$ is locally positive semidefinite in the region $\mathcal{X}$ if there exist polynomial matrices $S_i(x) \in \Sigma_x^n$, $V_j(x) \in \mathcal{R}_x^n$ verifying the following condition:

$$F(x) - \sum_{i=1}^{l} S_i(x) g_i(x) + \sum_{j=1}^{r} V_j(x) k_j(x) \in \Sigma_x^n \qquad (2)$$

∎

By the previous discussion, the computational check of conditions (2) can be done via SDP algorithms and SOS tools (Papachristodoulou, et al., 2013). This is the key to solve the constrained regression problem stated in the next section.

Lemma 3. The set of nonlinear matrix inequalities

$$R(x) > 0, \qquad Q(x) - S(x)^T R(x)^{-1} S(x) > 0, \qquad (3)$$

where $Q(x) = Q(x)^T$, $R(x) = R(x)^T$ and $S(x)$ are polynomial matrices in x, is equivalent to the following polynomial matrix expression:

$$M(x) = \begin{bmatrix} Q(x) & S(x)^T \\ S(x) & R(x) \end{bmatrix} > 0 \qquad (4)$$

∎

This is the direct extension of the well-known Schur Complement result in the LMI framework (Boyd et al. 1994) to the polynomial case. Condition (4) can be checked (conservatively) via SOS programming, as previously discussed.

3. SOS CONSTRAINED REGRESSION

Now, back to the main topic of this paper, we first formally state the problem we attempt to solve.

3.1 Problem statement

Given a dataset of N sampled points of an output[2] y and some m input channels $x_1, \dots, x_m$, we aim to build an n-degree polynomial model

$$\hat{y} = f(\beta; x_1, \dots, x_m), \qquad \beta \in \mathbb{R}^{C_{m+n,m}} \qquad (5)$$

with $C_{m+n,m}$ parameters β that minimize a measure of the regression error J (e.g. $\mathcal{L}_1$-regularized error or squared error) with the data over a set of constraints on the parameter space $\beta \in \wp$, on the inputs $x \in \mathcal{X}$ and on the model response $\hat{y}$:

$$\underset{\beta \in \wp}{\text{minimize}} \quad \sum_{i=1}^{N} J\big(y_i - f(\beta; x_{1i}, \dots, x_{mi})\big) \qquad (6)$$

$$\text{s.t.:} \quad \Omega(\mathcal{X}) := \{\beta \in \mathbb{R}^{C_{m+n,m}} | c(x, \hat{y}) \geq 0, x \in \mathcal{X}\} \qquad (7)$$

Where the function $c(\cdot)$ represents a general set of polynomial constraints to specify bounds for local search and/or desired robust model features. Hence, (7) can represent constraints that may range from standard polynomial bounds on the output/inputs (ensuring non-negativity of a model in a region for example) to the more complex n-order ones, such as guaranteeing that model derivatives obey thermodynamic principles (see next section). Thus, (7) makes the regression become a semi-infinite constrained optimization problem.

3.2 SOS-programming reformulation

The above optimization (6)-(7) can be cast as convex SOS problem if polynomials f, c are affine in decision variables β, J is linear in β, and the region $\mathcal{X}$ is defined by polynomial bounds on x.

Objective function. The more usual regression measures based on the $\mathcal{L}_1$ and $\mathcal{L}_2^2$ norms (absolute error and least squares approach respectively) can be reformulated for SDP optimization as follows.

I. The $\mathcal{L}_1$ norm $|y_i - f(\beta; x_{1i}, \dots, x_{mi})|$ is enforced by:

$$\underset{\beta \in \wp,\ \tau \in \mathbb{R}^+}{\text{minimize}} \quad \sum_{i=1}^{N} \tau_i \qquad (8)$$

$$\text{s.t.:} \quad \tau_i - y_i + f(\beta; x_{1i}, \dots, x_{mi}) \geq 0 \quad i: 1, \dots, N \qquad (9)$$

$$y_i - f(\beta; x_{1i}, \dots, x_{mi}) + \tau_i \geq 0 \quad i: 1, \dots, N \qquad (10)$$

II. The $\mathcal{L}_2^2$ norm $\big(y_i - f(\beta; x_{1i}, \dots, x_{mi})\big)^2$ is enforced by:

$$\underset{\beta \in \wp,\ \tau \in \mathbb{R}^+}{\text{minimize}} \quad \sum_{i=1}^{N} \tau_i \qquad (11)$$

[2] A single output variable is considered for simplicity, but the results apply for MIMO system identification as well.

$$\text{s.t.: } \tau_i - \left(y_i - f(\beta; x_{1i}, \dots, x_{mi})\right)^2 \geq 0 \;\; i: 1, \dots, N \tag{12}$$

Which, using Lemma 3, (12) is equivalently expressed as:

$$\begin{bmatrix} \tau_i & e_i \\ e_i & 1 \end{bmatrix} \geq 0 \;\; i: 1, \dots, N; \;\; e_i \coloneqq y_i - f(\beta; x_{1i}, \dots, x_{mi}) \tag{13}$$

Constraints on the input/output domain. Constraints on the model output are represented in (7) by c of the form:

$$c(x, \hat{y}) = \alpha \cdot f(\beta, x) + h(x) \geq 0 \tag{14}$$

Where $\alpha \in \mathbb{R}$ scales the model response[3] and $h(x)$ is a polynomial user-defined function in x. Thus, depending on the degree of h we can state upper and lower limits on $\hat{y}$ (zero-order constraints), or more complex (higher order) constraints on the feasible region. Then, by Lemma 1, (7) with (14) is ensured by the sufficient SOS condition:

$$\alpha \cdot f(\beta, x) + h(x) - \sum_{i=1}^{l} s_i(x) g_i(x) + \sum_{j=1}^{r} v_j(x) k_j(x) \in \Sigma_x \tag{15}$$

With polynomial multipliers $s_i(x) \in \Sigma_x$, $v_j(x) \in \mathcal{R}_x$ of appropriate degree in x, and its coefficients being additional decision variables.

Constraints on the response derivatives. Model slopes and curvatures w.r.t. x get the following functional form for c:

$$c(x, \hat{y}) = \alpha^T \nabla_x f(\beta, x) + h(x) \geq 0 \tag{16}$$

$$c(x, \hat{y}) = \mathrm{A}^{\mathrm{T}} \cdot \nabla_x^2 f(\beta, x) \cdot \mathrm{A} + B(x) \geq 0 \tag{17}$$

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

For example, suppose that we desire to ensure the convexity of a regression candidate model $f(x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2^2 + \beta_4 x_1^2 x_2$. The Hessian for this model is:

$$H(\beta, x_1, x_2) = \begin{bmatrix} 2\beta_4 x_2 & 2\beta_3 x_2 + 2\beta_4 x_1 \\ 2\beta_3 x_2 + 2\beta_4 x_1 & 2\beta_3 x_1 \end{bmatrix} \tag{18}$$

The classical approach to ensure this is setting a constraint on the model curvature via the determinant of H being nonnegative. Unfortunately this constraint is nonconvex in the β space:

$$\Omega \coloneqq \{\beta \in \mathbb{R}^5 | -\beta_3 \beta_4 x_1 x_2 - \beta_4^2 x_1^2 - \beta_3^2 x_2^2 \geq 0, x \in \mathcal{X}\} \tag{19}$$

Assuming, for instance, least squares in (6), the inclusion of (19) will transform a quadratic problem into a quadratically constrained quadratic problem. Nonetheless, global model convexity can be easily enforced using SOS programming by just setting the constraint:

$$\begin{bmatrix} 2\beta_4 x_2 & 2\beta_3 x_2 + 2\beta_4 x_1 \\ 2\beta_3 x_2 + 2\beta_4 x_1 & 2\beta_3 x_1 \end{bmatrix} \in \Sigma_{x_1 x_2}^2 \tag{20}$$

Boundary constraints. Standard boundary conditions require equality constraints $c(x, \hat{y}) = 0$ in (7), enforced over some $x_i = x_i^*$. In this case, the general representation for c is:

$$c(x, \hat{y}) = \left(f(\beta, x) + \alpha^T \nabla_x f(\beta, x) + b^T \nabla_x^2 f(\beta, x) + h(x)\right)\big|_{x_i = x_i^*} \tag{21}$$

And their local enforcement in $x \in \mathcal{X}$ is:

$$c(x, \hat{y})|_{x_i = x_i^*} - \sum_{i=1}^{l} s_i(x) g_i(x) + \sum_{j=1}^{r} v_j(x) k_j(x) = 0 \tag{22}$$

Which can be directly introduced in SOS programming tools[4].

4. ILLUSTRATIVE EXAMPLE

In this toy example, adapted from (Cozad, Sahinidis and Miller, 2015), we model data sampled from $y = x^2 - 0.4x + 0.04 + \epsilon$ over $x \in [-1, 1]$, where ϵ is sampled from a uniform random distribution $\epsilon \in [-0.25, 0.25]$, using a regression model of the form:

$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^4 + \beta_5 x^5 + \beta_6 x^6 \tag{23}$$

Because we want to get a model with some reliable extrapolation capabilities, we enforced a bound (15) on the model output over an extended domain $x \in [-2, 2]$, with the following features:

$$\alpha = 1, \quad h(x) = -0.5x^2 + 0.3, \quad g_1(x) = x + 2, \quad g_2(x) = 2 - x \tag{24}$$

Moreover, since we know that the underlying distribution is convex, we would also like to enforce the generation of a globally convex surrogate model. So (17) with $\alpha = 1$, $h(x) = 0$ is set up in SOS programming form.

The training dataset consists of 25 points randomly sampled over the original domain $x \in [-1, 1]$. For the shake of comparison, we begin by solving unconstrained fitting problems using the $\mathcal{L}_1$-norm and LS regression measures respectively. Next, we compare this solution to the constrained cases, with convexity enforcement not only in the extended domain $x \in [-2, 2]$, but in the whole input space $x \in \mathbb{R}$.

Figure 1 depicts the regression data as well as the obtained surrogate models which minimize the above norms in the domain $x \in [-2, 2]$. The more remarkable aspect is both unconstrained surrogate models become negative when they barely have left the sampled domain, though they get a slight-

[3] Constraints on multiple outputs $\hat{y}$ could also be enforced in (15) as long as their relationship $d(\hat{y}(\beta; x))$ is expressed as a polynomial affine in β. Typically, linear relations, or quadratic ones reformulated via Schur Complement.

[4] Note that $c(x) = 0$ is equivalent to $c(x) \in \Sigma_x$ jointly with $-c(x) \in \Sigma_x$. Moreover, $c(x) \in \Sigma_x$ is equivalent to $c(x) - s(x) = 0$ and $s(x) \in \Sigma_x$.

ly better fitting (see Table 1 below). As expected, the constrained surrogate models respect the constraint imposed with (24).

Next, Figure 2 depicts the models curvature (second-order derivative) within the domain $x \in [-2, 2]$. As expected from the analysis of Figure 1, the unconstrained regression models get a very negative second derivative, even before leaving the sampling region $x \in [-1, 1]$. However, the constrained cases never get a nonnegative curvature in the whole input space.

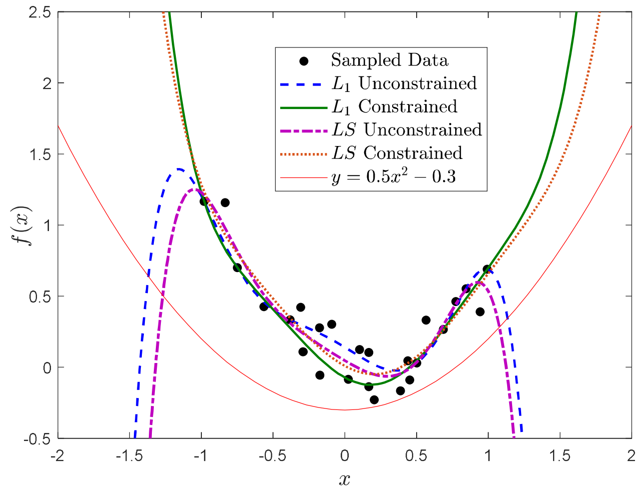

Fig. 1. Resulting surrogate models with LS and $\mathcal{L}_1$-norm regression approaches.

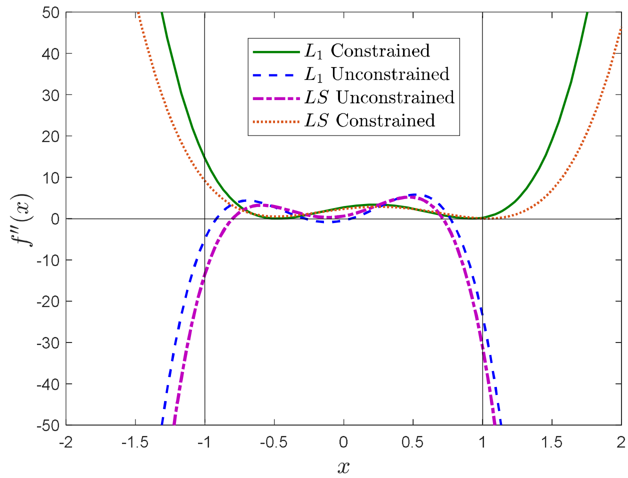

Fig. 2. Curvature corresponding to the identified models.

The obtained surrogate models are:

$$\hat{y}_{L1}^{U} = 0.1409 - 0.6197x - 0.2272x^2 + 1.181x^3 + 2.484x^4 - 0.8203x^5 - 1.452x^6 \quad (25)$$

$$\hat{y}_{L1}^{C} = -0.0721 - 0.5529x + 1.367x^2 + 0.925x^3 - 0.8096x^4 - 0.6359x^5 + 0.4786x^6 \quad (26)$$

$$\hat{y}_{LS}^{U} = 0.0465 - 0.596x + 0.304x^2 + 0.9854x^3 + 2.157x^4 - 0.7367x^5 - 1.625x^6 \quad (27)$$

$$\hat{y}_{LS}^{C} = 0.0093 - 0.5465x + 1.175x^2 + 0.6592x^3 - 0.4762x^4 - 0.4256x^5 + 0.2648x^6 \quad (28)$$

Where superscript letters U and C mean "unconstrained" and "constrained" respectively.

Table 1. Regression error

	$\mathcal{L}_1$ norm	Least Squares
Unconstrained	2.703	0.431
Constrained	2.836	0.4765

Regarding the computational effort, no one of the above regressions elapses more than one second (indeed the LS ones are solved in less than half a second) in a common laptop (Intel® i7-4510U CPU).

5. REMARKS AND FURTHER EXTENSIONS

This paper shows how sum-of-squares decompositions of polynomials can be used to cast many semi-infinite constrained regression problems in convex ones, and highlights the powerfulness of the semidefinite programming to efficiently tackle problems of grey-box model building.

The resulting models can be bounded in value, as well as its gradient (slope constraints) and Hessian (model convexity) while maintaining convexity on the underlying optimization problem. These constraints can, thus, be used to ensure desirable features of the final regression model (similar to other regularization options in literature). As in any nonlinear model-fitting application, standard training/test set validation or leave-one-out techniques must be carried out if few data are present in realistic setups.

The main drawback is that candidate models are limited to be polynomials, though polynomial basis functions are flexible and used in practice. Nevertheless, we will study the possible extension of the approach to cope with other nonpolynomial basis functions via multimodel polynomial bounding. In addition, although SOS programming is *convex* optimization, its scalability is limited by the number of independent variables x and the degree of polynomials. This fact may represent an issue in applying the proposed ideas to complex chemical-reaction problems with half a dozen or more components involved (overall complexity is problem dependent of course). Nonetheless, it is worth noting that we do not aim to get complete-plant surrogate models with the ideas in this paper, but just few-to-few local relationships among process variables to complete a grey-box model based on physics.

The great feature of automatic modeling tools like ALAMO, able to select the suitable model complexity by deactivating unnecessary basis functions, is of interest for future work too. In this context, the extension of the SOS programming to deal with mixed integer SOS problems would be desirable.

REFERENCES

Afram, A., and Janabi-Sharifi, F. (2015). Black-box modeling of residential HVAC system and comparison of gray-box and black-box modeling methods. *Energy and Buildings, 94*, 121-149.

Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V. (1994). *Linear matrix inequalities in system and control theory* (Vol. 15). SIAM.

Cozad, A., Sahinidis, N., and Miller, D. (2015). A combined first-principles and data-driven approach to model building. *Computers and Chemical Engineering, 73*, 116-127.

Davies, R. (2015). *Industry 4.0: Digitalization for productivity and growth.* document PE 568.337: European Parliamentary Research Service.

de Prada, C., Hose, D., Gutierrez, G., and Pitarch, J.L. (2018). Developing Grey-box Dynamic Process Models. *IFAC-PapersOnLine, 51*(2), 523-528.

Golovin, D., Solnik, B., Moitra, S., Kochanski, G., Karro, J., and Sculley, D. (2017). Google Vizier: A Service for Black-Box Optimization. *Proc. 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1487-1495). Halifax: ACM New York.

Henrion, D., and Garulli, A. (Eds.). (2005). *Positive polynomials in control* (Vol. 312). Springer Science and Business Media.

Kalliski, M., Pitarch, J.L., Jasch, C., and de Prada, C. (2019). Support to Decision-Making in a Network of Industrial Evaporators. *Revista Iberoamericana de Automática e Informática industrial, 16*(1), 26-35.

Kar, A. (2015). A hybrid group decision support system for supplier selection using analytic hierarchy process, fuzzy set theory and neural network. *Journal of Computational Science, 6*, 23-33.

Krämer, S., and Engell, S. (2017). *Resource Efficiency of Processing Plants: Monitoring and Improvement.* Weinheim, Germany: Wiley-VCH.

Kujanpää, M., Hakala, J., Pajula, T., Beisheim, B., Krämer, S., Ackershott, D., Kalliski, M., Engell, S., Enste, U. and Pitarch, J.L. (2017). *Successful Resource Efficiency Indicators for process industries* (Vol. 290). Espoo: VTT Technology.

Llanos, C., Sanchéz, M., and Maronna, R. (2015). Robust Estimators for Data Reconciliation. *Industrial and Engineering Chemistry Research, 54*(18), 5096-5105.

Nauta, K.M., Weiland, S., Backx, A., and Jokic, A. (2007). Approximation of fast dynamics in kinetic networks using non-negative polynomials. *16th IEEE Inter. Conf. on Control Applications,* 1144-1149, Singapore.

Olsen, I., Endrestøl, G., and Sira, T. (1997). A rigorous and efficient distillation column model for engineering and training simulators. *Computers & Chemical Engineering, 21*, S193-S198.

Palacín, C.G., Pitarch, J.L., Jasch, C., Méndez, C., and de Prada, C. (2018). Robust Integrated Production-Maintenance Scheduling for an Evaporation Network. *Computers and Chemical Engineering, 110*, 140-151.

Papachristodoulou, A., Anderson, J., Valmorbida, G., Prajna, S., Seiler, P., and Parrilo, P. (2013). *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*.

Parrilo, P. (2000). *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization.* PhD Thesis: California Institute of Technology.

Parrilo, P. (2003). Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming, 96*(2), 293-320.

Pitarch, J.L. (2013). *Contributions to fuzzy polynomial techniques for stability analysis and control.* PhD Thesis: Universitat Politècnica de València.

Pitarch, J.L., and Sala, A. (2014). Multicriteria fuzzy-polynomial observer design for a 3DoF nonlinear electromechanical platform. *Engineering Applications of Artificial Intelligence, 30*, 96-106.

Pitarch, J.L., Rakhshan, M., Mardani, M.M., and Shasadeghi, M. (2018). Distributed Saturated Control for a Class of Semilinear PDE Systems: An SOS Approach. *IEEE Transactions on Fuzzy Systems, 26*(2), 749-760.

Putinar, M. (1993). Positive Polynomials on Compact Semi-algebraic Sets. *Indiana Univerisity Mathematics Journal, 42*(3), 969-984.

Reemtsen, R., and Rückmann, J.J. (1998). *Semi-infinite programming.* Springer Science & Business Media.

Scherer, C. (2005). Relaxations for Robust Linear Matrix Inequality Problems with Verifications for Exactness. *SIAM Journal on Matrix Analysis and Applications, 27*(2), 365-395.

Wang, Y., Sun, F., and Jia, M. (2016). Online monitoring method for multiple operating batch processes based on local collection standardization and multi-model dynamic PCA. *The Canadian Journal of Chemical Engineering, 94*(10), 1965-1976.

Wilson, Z., and Sahinidis, N. (2017). The ALAMO approach to machine learning. *Computers and Chemical Engineering, 106*, 785-795.

Witten, I., Frank, E., Hall, M., and Pal, C. (2016). *Data Mining: Practical machine learning tools and techniques.* Elsevier.

Zorzetto, L., Maciel Filho, R., and Wolf-Maciel, M. (2000). Processing modelling development through artificial neural networks and hybrid models. *Computers and Chemical Engineering, 24*(2-7), 1355-1360.