Application of SOS-constrained regression to model unknown reaction kinetics

José Luis Pitarch*, Daniel A. Montes*, César de Prada*,*, Antonio Sala****

**Systems Engineering and Automatic Control DPT, EII, Universidad de Valladolid.*
C/Real de Burgos s/n, 47011, Valladolid, Spain, (jose.pitarch@autom.uva.es, danielalberto.montes.lopez@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. C/Real de Burgos s/n, 47011, Valladolid, Spain, (prada@autom.uva.es)*

Abstract: The key idea of the fourth industrial revolution is to use the huge amount of data from the increased process digitalisation in order to make better decisions at all levels: from the design and control, to operation and management. However, advanced decision support systems usually rely on good plant models. Despite the increased popularity of machine learning, in the process industry many of these approaches may fail in building reliable prediction models: that is, models whose output can be trusted even out of the region where actual data was collected. This paper illustrates how to get a reliable grey-box model of a chemical plant for optimisation purposes via sum-of-squares (SOS) constrained regression, a method that guarantees full enforcement of physical features on the identified model, no matter the quality and quantity of the collected data. The approach is used here to identify a reliable model for the reaction kinetics in a hybrid CSTR, a pilot plant where the chemical reactions are emulated over a harmless fluid.

Keywords: Data-driven models, Process identification, Hybrid plant, SOS programming, sparse datasets.

1. MOTIVATION

Industry is at the doors of the full digital era, where the impressing amount of data that will be stored, as well as the speed at which they are recorded, is expected to significantly affect the decision-making procedures at all levels of a factory. However, in the process industries (those where bulk materials are submitted to complex physical and chemical processes), these expected advances will not come alone by just collecting huge amounts of data and presenting them in a nice view: data treatment and analytics is necessary. Moreover, models for reliable predictions need to be built upon such data (Aguilar, Torres & Martín, 2019), in order to be later used in advanced control, real-time optimisation and scheduling routines (Grossmann & Harjunkoski, 2019).

First, several drawbacks that torpedo the success of machine learning (ML) in the chemical industry already arise from the data collection side (Pitarch & de Prada, 2019):

- Measurements could be biased and/or corrupted.
- Plants are operating in the same point most of the time.
- Extensive experimentation is too expensive or limited due to production constraints.
- Some key variables are not accessible for measurement.

Concerning model building, there are also some problems to face, such as finding the right trade-off between achieving high fidelity and low computational cost, how to customise a physics-based model to really match the actual plant, or ensuring coherent behaviours in extrapolating with a data-driven model. In this context, the direct application of an ML approach in the process industry needs to be evaluated carefully, as this is not the first (unsuccessful) attempt in the history of process systems engineering (Venkatasubramanian, 2019).

In these industrial sectors, it is not sensible to throw away all the *deep knowledge* acquired by expert scientists and engineers for many years, just to replace it with *deep learning* algorithms. Thus, one of the key challenges of ML to successfully penetrate in the process industry is developing methods and tools that are able to naturally embed and combine the existent process knowledge with data. Consequently, the process control and chemical engineering communities devoted efforts to develop effective methods and tools for building grey-box models that get both high matching with the actual plant in current operation and, importantly, confidence for extrapolation (i.e., prediction beyond the explored operation zone). See for instance the works by Nauta et al. (2007) and Cozad, Sahinidis & Miller (2015), among others.

Recently, the authors proposed a systematic methodology for building grey-box models of process plants (Pitarch, Sala, & de Prada, 2019b). In this method, the two main sources of process knowledge (well-known first principles and collected data) are integrated into a data-reconciliation stage, which provides an extended set of *virtual data* that is coherent with the basic process physics. Then, ML is recalled to identify any required black-box relationship among variables that was difficult to model at first instance. Any ML procedure could be used in this second stage, but *constrained regression* is the suggested one in order to provide interpretability as well as the desired reliability in extrapolation with the identified black-box sub models.

Briefly summarising, constrained regression attempts to solve the following problem. Assume that N reliable data points are available for some outputs y and inputs x. Then, a form $f(x)$

♦ This research received funding from the EU and the Spanish MICINN through research projects PGC2018-099312-B-C31 and DPI2016-81002-R (AEI/FEDER).

is sought such that a p-measure of the error (e.g., L_1-regularized or least squares) w.r.t. the data are minimised subject to some extra constraints $c(x) \geq 0$ with physical meaning:

$$\underset{\alpha \in \mathbb{R}^n}{\text{minimise}} \sum_{t=1}^{N} \left\| y_{[t]} - f(\alpha; x_{[t]}) \right\|_p \qquad (1)$$
$$\text{s.t.}: c(\alpha; x) \geq 0 \; \forall \, x \in \mathcal{X}$$

Where the decision variables α are the coefficients of $f(x)$, i.e. the model parameters, and constraints $c(x)$ would like to hold in a local region $\mathcal{X}$ of the input space, that may be wider than the region to which the N experimental points $x_{[t]}$ belong.

Two main technologies have been developed to address (1) off-line in a computationally affordable way: *symbolic regression* via mixed-integer programming (Neumann et al., 2019) and the *SOS-constrained regression* via semidefinite programming (Pitarch, Sala, & de Prada, 2019a). The main drawback with symbolic regression is that constraints $c(x) \geq 0$ can only be checked in the *finite* number of points present in the dataset, which is certainly a subset of region $\mathcal{X}$. This important issue vanishes with the SOS approach where, due to the inherent features of semidefinite programming, $c(x) \geq 0$ is ensured in the whole region $\mathcal{X}$ independently of the data available. However, SOS programming can only handle polynomial $f(x)$ whereas nonlinear basis functions can be part of the candidate model in symbolic regression. Note that, although polynomials are flexible regressors, many of the internal mechanisms in (bio)chemical systems behave according to non-polynomial representations, so the models obtained by the SOS approach may lack physical interpretability.

Pilot plants of reduced scale are a way to test these novel modelling methods. However, despite needing less investment in instrumentation or equipment, they also involve raw materials and chemical reactions that can be expensive or dangerous. Moreover, these plants also require a careful setting up and maintenance. Here is where the concept of *hybrid plants* can be useful. In these pilot setups, the equipment and the hydrothermodynamics of the process are real (but limited to the properties of the involved raw materials) and the hazardous components (e.g, chemical reactions) are emulated via software (Kershenbaum & Kittisupakorn, 1994). In this line, we built a hybrid continuous-stirred tank reactor (CSTR) of laboratory scale. The aim is to use this hybrid plant as proof of concept for advanced modelling, control and optimisation solutions (e.g., Kalliski et al. (2019), among others).

Hence, the contributions in this paper are twofold: a practical one, by applying the SOS-constrained regression to build grey-box models, illustrated with the constructed hybrid CSTR; and a methodological one, extending the SOS approach to include non-polynomial basis functions, without losing full constraint satisfaction guarantees of course. To achieve this, the non-polynomial terms arising in $c(x)$ are bounded between polynomial vertex models of desired complexity via the *Taylor series* approach (Sala & Ariño, 2009).

In the next section, the reader will find the description of the constructed hybrid CSTR. Then, Section 3 resumes the SOS-modelling approach, whose application to the CSTR case is in Section 4 together with the obtained results, and a discussion section with remarks closes the paper.

2. THE HYBRID CSTR PILOT PLANT

2.1 Hardware components

In the hybrid CSTR of Fig. 1 the "reactants" (i.e. water) are stored in tank T-101 and fed to the reactor using pump P-101 (0.3 to 1.7 l/min), manipulated by a PID flow control loop. The volume of the reactor is constant as the products are extracted using P-102 mimicking overflow. The cooling fluid is water from a network and a PID controller acting over V-101 sets the flowrate (0.8 to 15 l/min). Exothermic reactions can be simulated by the computation block UX-100, so that the heat is generated in the vessel R-101 (AISI 316 stainless steel) by two electric resistances (3 kW each) feed with the computed signal amplified in J. The plant has four PT100 temperature probes and two magnetic flowmeters, shown in Fig. 1.

Perfect mixture is assumed inside the vessel R-101 (11.5 l volume), so the process can be modelled by mass and energy balances of lumped parameters. Note that the kinetic calculations in UX-100 are carried out with the actual reactor temperature (measured) and reactants inflow.

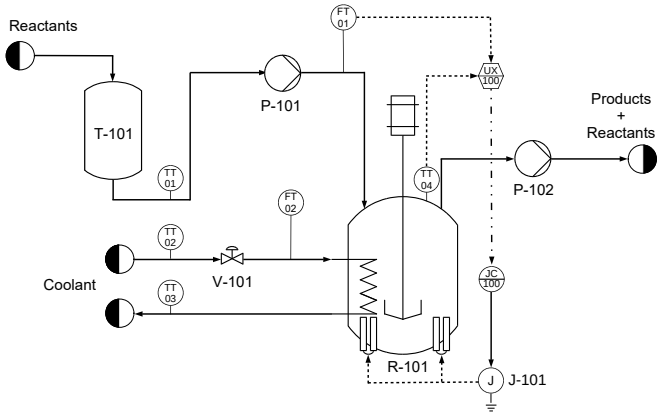

Fig. 1. P&ID of the hybrid CSTR pilot plant.

The data acquisition card MCC USB-1208HS-4A0, which reads and writes in the 0-5 V range, manages all control signals. However, instruments work in 4-20 mA range. So an additional conversion stage is needed, as shown in Fig. 2.

2.2 Software components

There are three main software components to run the plant: an OPC server that contains the simulation of the chemical reactions, OPC servers that implement two digital PID flow controllers and a SCADA system developed in the Wonderware InTouch environment (Fig. 2). This SCADA is in charge of managing all communications among the different OPC servers, visualising the information from plant sensors as well as allowing the user to manipulate actuators, tuning the PIDs and setting the kinetic parameters of the chemical reactions to be simulated.

The evolution of plant variables can be recorded from the SCADA via ReadOPC v1.6 software. The resulting data file is suitably processed by a Python script developed ad-hoc that converts it to a readable CSV file.

To facilitate the understanding of the methodology proposed in the next section, a toy first-order reaction A → B is coded in the simulation block.

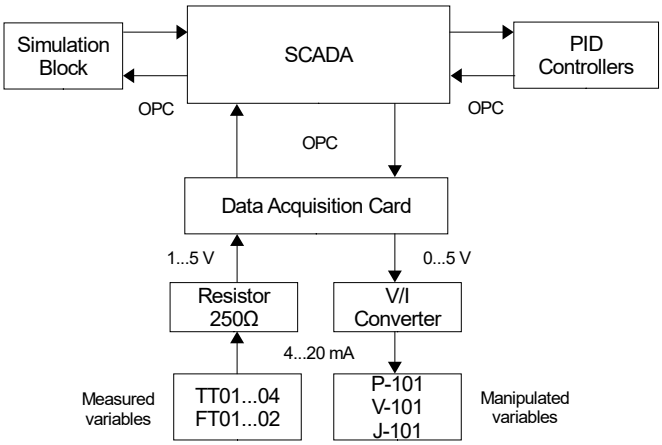

Fig. 2. Plant communications scheme.

The reaction rate (c_A is the concentration of reactant A and E the activation energy)

$$r_A := k_0 \cdot e^{-\frac{E}{R \cdot T}} \cdot c_A \tag{2}$$

appears in the mass balance (3). The reactor temperature T and volume V are used in the energy balances:

$$V\,\dot{c}_A = (c_{A0} - c_A)\,q_R - V\,r_A \tag{3}$$

$$\rho C_p V\,\dot{T} = (T_0 - T)q_R - (T - T_C)\,U\,A \tag{4}$$

$$\rho_C C_{pC} V_C\,\dot{T}_C = (T_{C0} - T_C)\rho_C C_{pC}\,q_C + (T - T_C)\,U\,A \tag{5}$$

The heat generated by the chemical reaction is computed by:

$$Q = V\,r_A\,\Delta H \tag{6}$$

Where ΔH is the standard enthalpy of reaction.

If this was an industrial setup, parameters such as volumes V, flow densities ρ, specific heats C_p, exchange area A, could be assumed known with enough precision. However, the kinetics of the actual chemical reaction(s) are not always well known. Similarly, it could happen with the actual expression of the heat-transfer coefficient U, dependent of the cooling flow q_R.

Remark 1. For the modelling in this paper, neither the parameters nor the true structure of (2) will be assumed fully known.

3. SOS-CONSTRAINED REGRESSION

First, we briefly recall some definitions and basic results on sum-of-squares programming.

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 $p(x) = z^T(x) Q z(x)$, with $z(x)$ being a vector of monomials in x. Then, checking if any $Q \succcurlyeq 0$ exist for a given p is a linear matrix inequality (LMI) problem (Parrilo, 2000).

In this way, if p is affine in decision variables (typically its coefficients), it can be checked for SOS via efficient SDP solvers (Papachristodoulou, et al., 2013). From now on, the set of SOS polynomials is denoted by the symbol Σ_x.

SOS optimisation. Analogously to the previous cases, the minimisation of a cost index linear in some decision variables α subject to SOS constraints $g(\alpha; x) \in \Sigma_x$ or SOS positive-definiteness constraints $G(\alpha; x) \in \Sigma_x^n$ with g, G affine in α can also be cast as a convex SDP problem. Scalar matrix linear constraints on α can be incorporated too, as they are zero-degree polynomials.

Local positivity of polynomials on semialgebraic sets can be checked via the well-known Putinar's Positivstellensatz theorem (Putinar, 1993). Lemmas 1 and 2 in the Appendix are simplified versions of such result.

Accordingly, the constrained regression (1) can be cast as an SOS optimisation problem provided that polynomials f and c are affine in decision variables, the objective function is linear, and the region $\mathcal{X}$ is defined by polynomial boundaries:

$$\mathcal{X} := \{x | g(x) \geq 0; k(x) = 0;\ g, k \in \mathcal{R}_x^n\} \tag{7}$$

For a given dataset $\mathcal{D} := \{Y \in \mathbb{R}^{n_o \times N}, X \in \mathbb{R}^{n_i \times N}\}$ of N samples of n_o output variables (measured or estimated) and n_i input ones, denote by $F(\alpha; X)$ to a matrix whose columns result from evaluating a polynomial vector form $f(\alpha; x) \in \mathcal{R}_x^{n_o}$ at each value $x_{i[t]} \in X$. In other words, F is the matrix of model predictions for the $n_i \times N$ input samples. Then, if α are the affine coefficients in the polynomial candidate form f, then $F(\alpha; X) = \alpha\, Z(X)^T$, where $Z(X) \in \mathbb{R}^{C_{n_i+d,n_i} \times N}$ is a matrix containing all monomials in $f(x)$ evaluated at the samples X. In the usual least-squares fitting ($\mathcal{L}_2^2$ norm), the fitness function in (1) can be efficiently expressed as:

$$\|Y\,U_1 - \alpha\,V\,\Sigma_1\|_l \tag{8}$$

Where the *economic* singular value decomposition $F(X) = U_1 \Sigma_1 V^T$ is used to reduce the problem size when $N \gg C_{n_i+d,n_i}$. Then, (1) with (8) is reformulated for SOS optimisation as:

$$\underset{\alpha,\beta,\gamma \in \mathbb{R}^n;\, \tau \in \mathbb{R}^+}{\text{minimise}} \quad \tau$$

$$\text{s.t.: } \eta^T \begin{bmatrix} \tau & YU_1 - \alpha V \Sigma_1 \\ U_1^T Y^T - \Sigma_1 V^T \alpha^T & I \end{bmatrix} \eta^T \in \Sigma_\eta \tag{9}$$

$$c(\alpha; x) - \sum_{i=1}^{l} s_i(\beta; x) g_i(x) + \sum_{j=1}^{q} v_j(\gamma; x) k_j(x) \in \Sigma_x \tag{10}$$

Where Lemma 3 (Appendix A) is recalled and used to equivalently express $\tau - \|YU_1 - \alpha V \Sigma_1\|_2^2 \geq 0$ as (9), and Lemma 1 to enforce the additional constraints $c(x) \geq 0$ locally in $\mathcal{X}$. The polynomial forms $c(x)$ in (10) can be chosen among:

- Constraints on the input/output domain.
- Constraints on the model (partial) derivatives, i.e., bounded slopes, curvatures, and/or enforcing convexity.
- Boundary constraints ensured by equality constraints $c(x)|_{x_{[t]}} = 0$, enforced over some $x_{[t]} = \bar{x}$.

The reader is referred to (Pitarch, Sala, & de Prada, 2019b) for extended details on the possible shapes of $c(\alpha; x)$.

Extension to non-polynomial basis functions

The goal now is to allow the inclusion of non-polynomial terms $h(\beta; x)$, such as $e^{\beta x}$, $\log(\beta x)$, $\sin(\beta x)$, $\sqrt{x}$, etc., in the candidate form $f(x)$. Unfortunately, the parameters β here cannot be decision variables for the optimisation. Furthermore, the method presented below requires that $h(x)$ is C^{on} so that

its Taylor expansion of order o_n exists and converges to $h(x)$ in a region $\mathcal{X}$ of the input space as $o_n \to \infty$.

Given these assumptions, now denote $\rho_i = h_i(\beta; x)$ to each i-th non-polynomial function that will be included in the candidate model f. Then, taking ρ as additional input variables, a polynomial model $f(\alpha; x, \rho)$ of degree d can be sought.

Note that the matrix $F(\alpha; X)$ is now computed by applying $f(\alpha; x, h(\beta; x))$ to each value $x_{i[t]}$ in X. Consequently, $Z(X)$ is now a matrix containing all monomials in x and ρ present in f, evaluated at the samples X. From here, the fitting objective (8), as well as its particular SOS implementation (9) can be formulated. However, $c(x)$ are no more polynomials in x, as the regressors $h(x)$ and/or other possible nonlinearities derived from them arise in c, depending on the type of constraint to be enforced. E.g., a constraint on the model slope involves

$$\frac{df}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial \rho} \cdot \frac{d\rho}{dx}$$

so that $h(x)$ can arise in $\partial f/\partial x$ and in $d\rho/dx$, and new non-polynomial forms $u(\beta; x)$ normally arise in $d\rho/dx$ as well.

Then, the idea proposed here is to embed each of these nonlinearities, $h(x)$ and $u(x)$, between *polynomial vertices* using Lemma 4 (Appendix A). Hence, $c(x)$ can be equivalently expressed as a convex combination of polynomials in region $\mathcal{X}$:

$$c(\alpha; x) = \sum_{k=1}^{r} \sigma_k(x)\, p_k(\alpha; x),\ \sum_k \sigma_k(x) = 1, x \in \mathcal{X} \quad (11)$$

Where $r = 2^v$ is the total number of vertex models in the general case, being v the number of defined nonlinearities $h(x_i)$ and $u(x_i)$ on a single variable x_i.

Following this idea and neglecting the shape dependency in the interpolating functions $\sigma(x)$, i.e., transforming $\sigma_k(x)$ into an arbitrary scalar in the standard simplex $\sigma_k \in \Delta$ (shape independency) as in (27), constraints (10) can be conservatively guaranteed by enforcing them in the k: $1, \dots, r$ vertices:

$$p_k(\alpha; x) - \sum_{i=1}^{l} s_{ki}(x) g_i(x) + \sum_{j=1}^{q} v_{kj}(x) k_j(x) \in \Sigma_x \quad (12)$$

4. APPLICATION TO THE HYBRID CSTR

The goal now is to apply the above modelling methodology to find an expression for the reaction rate $r_A(T, c_A)$ from experimental data. Note that the precise knowledge of (2) *is not known*. Therefore, a fully fixed functional structure will not be set for regression as in classical parameter identification. Instead, a more flexible one will be sought through SOS-constrained regression, so that certain coherence with the basic kinetic features of the class of chemical reactions taking place in the reactor will be assured, no matter the size, sparsity or diversity of the dataset recorded from plant sensors.

To this aim, an experiment of 305 min duration (5 sec sampling time) is run to collect data with sensible step variations in P-101 and V-101 in order to capture the plant state dynamics with precision, see Fig 3. This provides a dataset of 3660 samples, choosing randomly half for regression and half for validation for instance (other divisions can be done as well).

At this point, *dynamic data reconciliation* is recalled with system equations (3)-(5) to robustly estimate r_A from all available plant measurements (Pitarch, Sala, & de Prada, 2019b). Note that full state measurements are available in this case, as c_A computed in UX-100 is assumed measurable in real time at the reactor outlet.

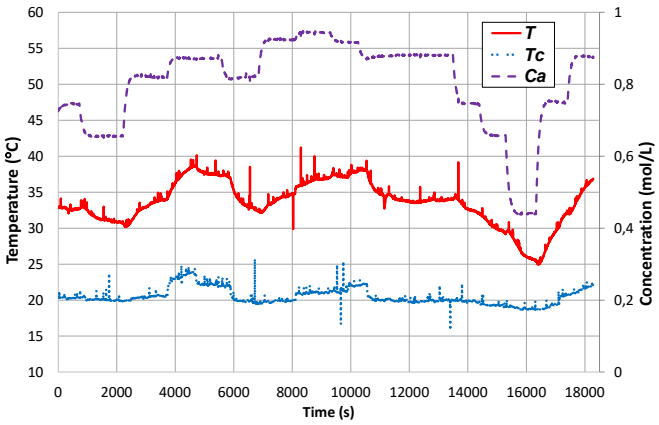

Fig. 3. Evolution of the CSTR state during the experiment.

This filtering approach resulted satisfactory. The proof is that a nonlinear parameter identification with the functional form (2) fully known could acceptably identify[1] parameters k_0 and E from data. See the filtered r_A compared to the true response of (2) in Fig. 4. Nonetheless, as the precise structure of (2) is not assumed known for this example, nor it is usually in actual practice, the methodology in Section 3 is applied to build a physically coherent model for $r_A(T, c_A)$ in the local region:

$$\mathcal{X} := \{T, c_A | 0.2 \le c_A \le 1, 10 \le T - 273 \le 80\} \quad (13)$$

Remark 2. Note that $\mathcal{X}$ goes beyond the region explored in the experiment (see Fig. 5). This is set on purpose in order to ensure certain physical coherence of the model predictions when extrapolating to physically possible operating conditions.

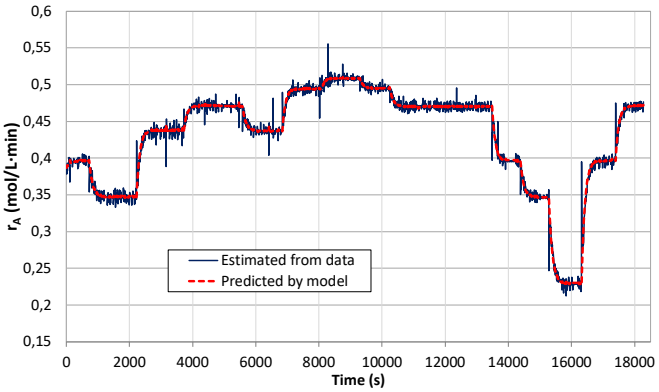

Fig. 4. Evolution of the reaction rate r_A over the time.

Now let's suppose that the chemical engineer has the insight that r_A necessarily increases with c_A and T. Furthermore, that the family of reactions $A \to B$ are governed by:

$$r_A = f\left(c_A^{\gamma}, e^{-\frac{\theta}{T}}\right);\quad \theta, \gamma \in \mathbb{R}^+; 0 < \gamma \le 2 \quad (14)$$

Where, $f(\cdot)$ is an unknown function of the above arguments. Note that c_A^{γ} and $e^{-\frac{\theta}{T}}$ (with fixed γ, θ) can be renamed as ρ_1 and ρ_2 so that a polynomial $f(c_A, T, \rho_1, \rho_2)$ could be directly sought with our proposed methodology. Nonetheless, as proposing a priori values for γ, θ close to the real ones would be sheer luck, the SOS optimisation would certainly remove the

[1] $k_0 = 1.38$ and $E = 2420$ were the values set for (2) in the UX-100 block, whereas $\hat{k}_0 = 1.367$ and $\hat{E} = 2398$ were identified by nonlinear regression.

basis ρ from the final form f. Thus, what is sensible to avoid this issue is searching for a *linear combination* between, at least, the two expected vertex values for each nonlinear basis. Consequently, we will search for a model of the form:

$$r_A = p_1(c_A) \cdot \mathrm{e}^{-\frac{120}{T}} + p_2(c_A) \cdot \mathrm{e}^{-\frac{600}{T}} \tag{15}$$

Where $p(c_A)$ are polynomial forms up to degree $d = 2$. The partial derivatives of (15) w.r.t. c_A and T are:

$$\frac{\partial r_A}{\partial c_A} = \frac{dp_1}{dc_A}\mathrm{e}^{-\frac{120}{T}} + \frac{dp_2}{dc_A}\mathrm{e}^{-\frac{600}{T}} \tag{16}$$

$$\frac{\partial r_A}{\partial T} = \mathrm{e}^{-\frac{120}{T}}\frac{120p_1}{T^2} + \mathrm{e}^{-\frac{600}{T}}\frac{600p_2}{T^2} \tag{17}$$

Now, denote the above non-polynomial terms by:

$$\begin{gathered}\rho_1 \coloneqq \mathrm{e}^{-\frac{120}{T}};\ \ \rho_2 \coloneqq \mathrm{e}^{-\frac{600}{T}};\\ \rho_3 \coloneqq \rho_1\frac{120}{T^2};\ \ \rho_4 \coloneqq \rho_2\frac{600}{T^2}\end{gathered} \tag{18}$$

Then, each of the above ρ is bounded in $\mathcal{X}$ between two linear (for simplicity) vertex models by computing the maximum and minimum values for the reminders of the Taylor expansion around $T_0 = 33.56 + 273$ K (mean value of the dataset):

$$\overline{\rho_1} = 8.988e^{-4}T + 0.1351;\ \underline{\rho_1} = 8.912e^{-4}T + 0.1327;$$

$$\overline{\rho_2} = 9.216e^{-4}T + 0.3928;\ \underline{\rho_2} = 7.707e^{-4}T + 0.4391;$$

$$\overline{\rho_3} = 9.624e^{-4} - 1.07e^{-6}T;\ \underline{\rho_3} = 9.452e^{-4} - 1.014e^{-6}T;$$

$$\overline{\rho_4} = 2.914e^{-3} - 4.758e^{-6}T;\ \underline{\rho_4} = 0.0027 - 4.06e^{-6}T;$$

Note importantly that, despite we have 4 nonlinearities that would give rise to $r = 2^4$ vertex models in the general case, they all depend on the same variable T and they reach their minimum and maximum values at the extremes of $\mathcal{X}$. Hence, they will never evolve independently, so we get full guarantees by just ensuring the desired features in the two vertices formed by all the $\overline{\rho}$ and $\underline{\rho}$ respectively. Finally, the desired positivity is locally enforced in the partial derivatives by setting (10) as:

$$\begin{aligned}&\frac{dp_1}{dc_A}\rho_{1k} + \frac{dp_2}{dc_A}\rho_{2k} - S_{1k}(353 - T)(T - 283) -\\ &\qquad S_{2k}(1 - c_A)(c_A - 0.2) \in \Sigma_{T,c_A}\ \forall \rho_{ik} \in \left\{\overline{\rho_i}, \underline{\rho_i}\right\}\end{aligned} \tag{19}$$

$$\begin{aligned}&p_1\rho_{3k} + p_2\rho_{4k} - S_{3k}(353 - T)(T - 283) -\\ &\qquad S_{4k}(1 - c_A)(c_A - 0.2) \in \Sigma_{T,c_A}\ \forall \rho_{ik} \in \left\{\overline{\rho_i}, \underline{\rho_i}\right\}\end{aligned} \tag{20}$$

With $S_{1k}, S_{2k} \in \mathbb{R}^+$ and quadratic $S_{3k}, S_{4k} \in \Sigma_{T,c_A}$ multipliers. Furthermore, including an upper bound τ on the $\partial r_A/\partial T$ is convenient too. This bound is chosen as the highest variation detected in the data w.r.t. the temperature, $\tau = 0.027$.

The resulting polynomials $p_1(c_A)$ and $p_2(c_A)$ after solving the SOS-constrained regression are:

$$\begin{aligned}p_1 &= -7.71c_A^2 + 9.8233c_A - 1.3925\\ p_2 &= 1.663c_A^2 - 1.3549c_A + 0.33276\end{aligned} \tag{21}$$

The model (15) with the computed polynomials (21) fits the data as good as a nonlinear parameter regression does if the *true structure* of r_A were known (see Table 1), while keeping coherent behaviour in the zone of $\mathcal{X}$ that was not covered by the performed experiment. Indeed, if the reader compares Fig. 5 with the shape of the true generating function (2) in $\mathcal{X}$, he/she will find little differences. That is of course due to the fact that (2) is not highly nonlinear in $\mathcal{X}$, but also because the proposed methodology imposes certain model coherence, no matter the degree of complexity and/or the features of the dataset.

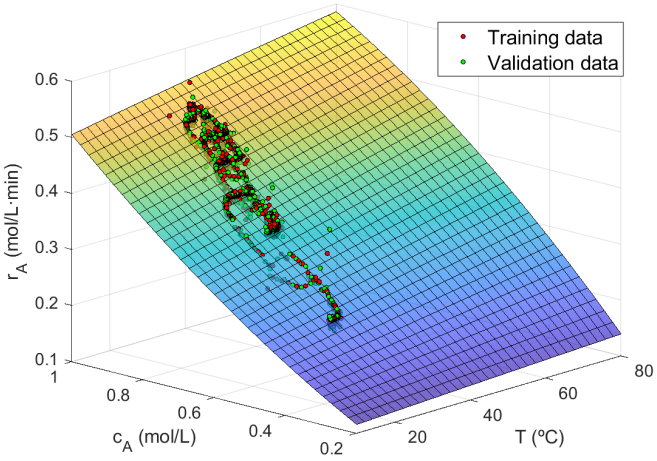

Fig. 5. Collected plant data over the computed model surface.

Remark 3. The SOS-constrained regression is solved in less than 3 seconds in a common laptop (Intel® i7-4510U CPU).

Table 1. Regression error

	Training	Validation	**Total**
Nonlinear ident.	0.10214	0.09092	**0.19306**
SOS Constrained	0.08965	0.10149	**0.19114**

5. REMARKS AND FURTHER EXTENSIONS

This paper shows how SOS-constrained regression is applied to find a reliable model from data collected in a chemical plant. Indeed, the methodology has proven effective with a sparse dataset and in presence of measurement noise. This proves that (conservatively) enforcing physical constraints does not necessarily penalise the total fitness to data.

Moreover, we outlined how the modelling approach based on SOS polynomials and semidefinite programming can be extended to include non-polynomial basis functions, while keeping full guarantees of constraint satisfaction. Derivations of the Positivstellensatz theorem and of the polynomial sector nonlinearity for fuzzy-systems modelling were key to build the proposed extension. However, the main drawback that remains is the impossibility of keeping the parameters in non-polynomial terms as decision variables in the SOS optimisation, a problem also common to other approaches based on symbolic regression (Cozad, Sahinidis, & Miller, 2015).

Although the SOS approach can be naturally combined with standard polynomial regularization methods, we admit that selecting the suitable model complexity by automatically activating the combination of basis functions is desirable. Thus, we foresee mixed-integer semidefinite programming (in particular MISOSP) as an interesting path to explore.

REFERENCES

Aguilar, R.M., Torres, J.M. & Martín, C.A. (2019). Automatic learning for the system identification. A case study in the prediction of power generation in a wind farm. *Revista Iberoamericana de Automática e Informática industrial, 16*(1), 114-127.

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

Grossmann, I., & Harjunkoski, I. (2019). Process Systems Engineering: Academic and industrial perspectives. *Computers & Chemical Engineering, 126*, 474-484.

Kalliski, M., Pitarch, J.L., Jasch, C., & 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.

Kershenbaum, L.S., & Kittisupakorn, P. (1994). The use of a partially simulated exothermic (PARSEX) reactor for experimental testing of control algorithms. *Chemical Engineering Research & Design*, *72*(1), 55-63.

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

Neumann, P., Cao, L., Russo, D., Vassiliadis, V., & Lapkin, A. (2019). A new formulation for symbolic regression to identify physico-chemical laws from experimental data. *Chemical Engineering Journal,* 123412.

Papachristodoulou, A., Anderson, J., Valmorbida, G., Prajna, S., Seiler, P., & 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: Caltech.

Pitarch, J.L., & de Prada, C. (2019). Machine learning and the digital era from a Process Systems Engineering perspective. *10th EUROSIM Congress* (p. 12). Logroño.

Pitarch, J.L., Sala, A., & de Prada, C. (2019a). A Sum-Of-Squares Constrained Regression Approach for Process Modeling. *IFAC-PapersOnLine, 52*(1), 754-759.

Pitarch, J.L., Sala, A., & de Prada, C. (2019b). A systematic grey-box modeling methodology via data reconciliation and SOS constrained regression. *Processes, 7*(3), 170.

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

Sala, A., & Ariño, C.V. (2009). Polynomial fuzzy models for nonlinear control: A Taylor series approach. *IEEE Transactions on Fuzzy Systems, 17*(6), 1284-1295.

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.

Venkatasubramanian, V. (2019). The Promise of Artificial Intelligence in Chemical Engineering: Is It Here, Finally? *AIChE Journal, 65*(2), 466-478.

Appendix A. AUXILIARY RESULTS

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). $F(x)$ being SOS implies $F(x) \succcurlyeq 0 \; \forall x$. The set of $n \times n$ symmetric SOS polynomial matrices is denoted by the symbol $\Sigma_x^{n \times n}$.

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

$$p(x) - \sum_{i=1}^{l} s_i(\beta; x) g_i(x) + \sum_{j=1}^{q} v_j(\gamma; x) k_j(x) \in \Sigma_x \quad (22)$$

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

Remark 4. With Lemma 1 we can prove local positivity of odd-degree polynomials via SOS programming, by appropriately choosing the multipliers degree such that $deg(s(x)g(x))$ and $deg(v(x)k(x))$ is even and greater than $deg\big(p(x)\big)$.

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

$$F(x) - \sum_{i=1}^{l} S_i(\beta; x) g_i(x) + \sum_{j=1}^{q} V_j(\gamma; x) k_j(x) \in \Sigma_x^n \quad (23)$$

■

Lemma 3. The set of nonlinear matrix inequalities

$$R(x) > 0, \qquad Q(x) - S(x)^T R(x)^{-1} S(x) > 0, \quad (24)$$

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:

$$M(x) = \begin{bmatrix} Q(x) & S(x)^T \\ S(x) & R(x) \end{bmatrix} > 0 \quad (25)$$

■

This is the direct extension of the well-known Schur Complement result in the LMI framework to the polynomial case (Scherer, 2005). Condition (25) can be checked (conservatively) via SOS programming, as previously discussed.

Lemma 4. (Sala & Ariño, 2009) Let $f(x)$ be a continuous and smooth enough function of a variable so that its Taylor expansion up to degree d around certain x_0, i.e. $f_d(x - x_0)$, exists. Assume that the d-th derivative of f is continuous in a compact region $\mathcal{X}$ including x_0. Then, the Taylor's reminder is:

$$T_d(x) \coloneqq \frac{\big(f(x) - f_d(x - x_0)\big)}{(x - x_0)^d},$$

Then, computing its maximum and minimum in $\mathcal{X}$ as

$$\psi_1 \coloneqq \sup_{x \in \mathcal{X}} T_d(x), \;\; \psi_2 \coloneqq \inf_{x \in \mathcal{X}} T_d(x),$$

such reminder can be equivalently expressed by:

$$T_d(x) = \mu_1(x) \cdot \psi_1 + \mu_2(x) \cdot \psi_2 \quad (26)$$

I.e., $f(x)$ can be embedded in a convex combination between two polynomial vertex models of desired complexity d:

$$f(x) \subset f_d(x - x_0) + (\mu_1 \psi_1 + \mu_2 \psi_2)(x - x_0)^d$$
$$\mu_1 + \mu_2 = 1, \;\; \forall \mu \in [0,1], \;\; x \in \mathcal{X} \quad (27)$$

■

This result guarantees local positivity of a nonlinear function by just checking it on the polynomial vertices via SOS programming. This method can be also applied to any function that can be written as an expression tree with functions of one variable, i.e., addition and multiplication.