2015 IEEE 54th Annual Conference on Decision and Control (CDC)
December 15-18, 2015. Osaka, Japan

Non-convex scenario optimization with application to system identification

Marco C. Campi[1], Simone Garatti[2], and Federico A. Ramponi[1]

Abstract— **Convex scenario optimization is a well-recognized approach to data-based optimization where the solution comes accompanied by precise generalization guarantees. It has been used in system identification as a driving methodology to construct interval prediction models. With this paper, scenario optimization breaks into the realm of non-convex optimization. In non-convex optimization, the number of scenarios that determine the solution - the so-called support scenarios - cannot be bounded beforehand, and one has to wait until the solution is computed to evaluate the size of the support scenario set. A theory is developed in this paper such that the generalization property of the solution is *a-posteriori* evaluated based on the registered number of support scenarios. This new perspective empowers the method and opens up new important possibilities for it to be applied to system identification involving non-convex optimization.**

I. INTRODUCTION

In a previous paper by the same authors, [8], Interval Predictor Models (IPMs) for use in identification have been introduced. An IPM can be identified from a data set according to schemes introduced in [8] and can be used to predict future values of the output of a data generation mechanism. IPMs abandon the traditional perspective that a model returns a single value as output. Instead, unlike standard models in system identifications, an IPM returns an interval of possible outputs. The IPM selection is driven by the principle that the model correctly describes the seen data set and, among models correctly describing the data set, the one returning on average the smallest prediction interval is preferred. It has to be noted that models with interval outputs have been used in other contexts. Their origin lies in the theory of differential inclusions and set-valued dynamical systems, [3], [4] and [5]. Moreover, interval models have been also adopted in identification problems along lines different from the IPM theory of [8], e.g. [12], [13], [14], [15], [16], and [10]

While *a-priori* information has a fundamental role in selecting the class in which the IPM is identified so that the interval is small and practically useful, the main theoretical strength of the IPM's theory developed in [8] is that, under hypotheses of stationarity and independence, the reliability of the IPM can be determined with no prior knowledge, that is, it holds independently of the data generation mechanism at hand. Hence, by the use of IPMs a key conceptual separation is obtained: the reliability is always guaranteed, and *a-priori* information only impacts on the size of the prediction intervals, a quantity which can be assessed after the identification process has been completed.

The key tool used in [8] in the derivation of the IPM theory is the scenario optimization framework developed in [6], [7], [9]. In these papers, deep results on the generalization properties of data-based optimization programs have been established that hold true independently of the distributional properties of uncertainty. In an identification context, this latter fact is recast into saying that the reliability guarantees are valid independently of the data generation mechanism. However, one fundamental limitation occurs when the results of [6], [7], [9] are used, and it is that the theory is inherently based on the assumption of convexity of the function being optimized as well as on the convexity of constraints. When applied to identification, this assumption severely limits the freedom in the model class selection. For instance, in parameterizing the central line of an IPM with constant interval size, see [11], due to convexity the central line has to be assigned as a linear combination of fixed basis functions, while more flexible choices of tunable basis functions are ruled out. This e.g. excludes linear combinations of sinusoids with tunable frequency and phase such as $\sum_{k=1}^{p} \alpha_k \sin(\omega_k u + \phi_k)$, where the α_ks, ω_ks and ϕ_ks are estimated, or a single-layer neural network with structure $\sum_{k=1}^{p} \alpha_k \sigma(a_k u + b_k) + c$, where $\sigma(x) := 1/(1 + \mathrm{e}^{-x})$ is the sigmoid function and the α_ks, a_ks, b_ks, and c are estimated. Resorting e.g. to the sigmoid central line leads to constructing an IPM model by the optimization program

$$\begin{aligned} &\arg\min_{\alpha_k, a_k, b_k, c, h} \quad h \\ &\text{subject to } \left| y(i) - \sum_{k=1}^{p} \alpha_k \sigma(a_k u(i) + b_k) - c \right| \leq h \qquad (1) \\ &\qquad \text{for all } i = 1, \cdots, N, \end{aligned}$$

where the IPM's width h is minimized under the constraint that the data points $u(i), y(i)$, $i = 1, \cdots, N$, are in the IPM. Clearly, this program is not convex.

This paper breaks up with this limitation, and scenario optimization is generalized so that it becomes suitable to deal with non-convex optimization. A new scenario theory is presented to this purpose for the first time in this paper.

[1]Marco C. Campi, marco.campi@unibs.it, and Federico A. Ramponi, federico.ramponi@unibs.it, are with the Dipartimento di Ingegneria dell'Informazione, Università di Brescia, Italy.

[2]Simone Garatti, simone.garatti@polimi.it, is with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy.

One fundamental aspect is that, in a convex context, the maximum number of data points that determine the solution, the so-called support points, see [8], is *a-priori* known and it equals the number of optimization variables. In a non-convex context, this fact fails to be true, and the new perspective of this paper is that one first computes the solution and then evaluates the number of support points. It turns out that an *a-posteriori* judgment compensates for the lack of *a-priori* knowledge and leads to sharp and useful evaluations.

In this paper, we limit ourselves to consider IPMs with constant interval size and such that all seen points are included in the IPM as it is done in (1). The proposed methodology, however, can be applied more generally to various IPM structures. Moreover, one can conceive discarding data points that are left out of the IPM because they are considered as outliers. All these extensions will be presented in a subsequent publication.

The paper is organized as follows: in the next Section non-convex scenario optimization is presented, followed by the proof of its main result in Section III. In Section IV we resume system identification and show how the result from the previous sections can be applied.

II. Non-convex scenario optimization

Suppose that Δ is a probability space, endowed with a σ-algebra $\mathcal{D}$ and a probability P. Let, moreover, $(\Delta^N, \mathcal{D}^N, \mathsf{P}^N)$ be the N-fold Cartesian product of Δ equipped with the product σ-algebra $\mathcal{D}^N$ and the probability $\mathsf{P}^N = \mathsf{P} \times \cdots \times \mathsf{P}$ (N times). A point in $(\Delta^N, \mathcal{D}^N, \mathsf{P}^N)$ is thus a sample $(\delta^{(1)}, \cdots, \delta^{(N)})$ of N components extracted independently from Δ according to the same probability P. Each $\delta^{(i)}$ is called a "scenario"; in a system identification endeavor, a scenario is an input-output pair, i.e. $\delta^{(i)} = (u(i), y(i))$.
Let $\mathcal{X}$ be a subset of $\mathbb{R}^d$, $f : \mathcal{X} \to \mathbb{R}$ be a function and, for each $\delta \in \Delta$, let $\mathcal{X}_\delta$ be a subset of $\mathbb{R}^d$. For any sample $(\delta^{(1)}, \cdots, \delta^{(N)})$, we consider the corresponding sets $\mathcal{X}_{\delta^{(1)}}, \cdots, \mathcal{X}_{\delta^{(N)}}$, and we build the following program:

$$\begin{aligned} \mathrm{NCSP}_N : \ & \arg\min_{x \in \mathcal{X}} f(x) \\ & \text{subject to } x \in \mathcal{X}_{\delta^{(i)}} \text{ for all } i = 1, \cdots, N. \end{aligned} \tag{2}$$

The acronym NCSP stands for Non-Convex Scenario Program reflecting the fact that, unlike all the previous literature on scenario optimization where the subsets $\mathcal{X}$ and $\mathcal{X}_\delta$, as well as the function f, were convex, here no assumptions of convexity are made. NCSP_N is a *random* program, and hence its solution is also random. Finding the optimal solution of (2) is in general a difficult task. Here, we assume that an algorithm $\mathcal{A}$ is available, which maps $\{\delta^{(1)}, \cdots, \delta^{(N)}\}$ to a possibly sub-optimal solution x_N^* of (2). Interestingly, the generalization result of this paper is applicable to any $\mathcal{A}$ independently of its accuracy, provided the following assumption applies:

Assumption 1: P^N-almost surely, for all N, the solution $x_N^* := \mathcal{A}(\delta^{(1)}, \cdots, \delta^{(N)})$ exists, it is unique, and it satisfies all the constraints, that is, $x_N^* \in \mathcal{X}_{\delta^{(i)}}$, $i = 1, \cdots, N$. Moreover, x_N^* is invariant with respect to any permutation of the sample $(\delta^{(1)}, \cdots, \delta^{(N)})$. ■

The following definition is central to our discussion.

Definition 1: Let x be a given point in $\mathcal{X}$. The *violation probability* of x is defined as

$$\mathsf{V}(x) := \mathsf{P}\{\delta \in \Delta \ : \ x \notin \mathcal{X}_\delta\}.$$

■

Now consider a fixed reliability parameter $\varepsilon \in (0, 1)$. We say that $x \in \mathcal{X}$ is ε-feasible if $\mathsf{V}(x) \leq \varepsilon$. Although $\mathsf{V}(x)$ is a number for any fixed value of x, if $\mathsf{V}(\cdot)$ is computed corresponding to the solution x_N^*, then $\mathsf{V}(x_N^*)$ is a random variable over Δ^N. Our goal is to provide, for small values of ε, a guarantee on the ε-feasibility of x_N^*.

In a convex setup, i.e. with respect to a program where $f(x) = c^\top x$, and the sets $\mathcal{X}$ and $\mathcal{X}_\delta$ are convex, one can provide such a guarantee in the form

$$\mathsf{P}^N \{\mathsf{V}(x_N^*) > \varepsilon\} \leq \beta; \tag{3}$$

stated another way, (3) claims that

$$\mathsf{P}^N \{x_N^* \text{ is } \varepsilon\text{-feasible}\} \geq 1 - \beta.$$

This result is deeply grounded on the concept of *support constraint* [7, Definition 4], and on the fact that the number of support constraints in a d-dimensional convex program is at most d [7, Theorem 3]. When β is very small, $1 - \beta$ should be interpreted as a "very high confidence", or "practical certainty". The reader is referred to [9] for a study on how β depends on ε and N.

In the non-convex scenario program NCSP_N no upper bound on the number of support constraints is available, and it is easy to find examples where the number of support constraints can be as large as the number N of constraints; moreover, resorting to alternative routes based on the VC-theory, [1], leads to conservative results. Prompted by this fact, in this paper we address the feasibility of the solution of (2) along a different approach. We abandon the idea that there always exists a subset of constraints of *a-priori* bounded cardinality sufficient to support the solution; we suppose, instead, that after the extraction of $(\delta^{(1)}, \cdots, \delta^{(N)})$ the experimenter will be able to isolate a subset of constraints sufficient to support the solution. The reliability guarantee will depend on the *a-posteriori* assessed cardinality of that set, and *the smaller the cardinality, the better the guarantee*.

The following definition formalizes the concept of subset of constraints sufficient to support the solution.

Definition 2: Consider a sample $(\delta^{(1)}, \cdots, \delta^{(N)}) \in \Delta^N$, and let x_N^* be the corresponding solution of program

NCSP_N. A *support set* for NCSP_N is a subset of elements $S = \{\delta^{(i_1)}, \cdots, \delta^{(i_k)}\} \subseteq \{\delta^{(1)}, \cdots, \delta^{(N)}\}$ such that the program obtained by removing from NCSP_N all the constraints except $x \in \mathcal{X}_{\delta^{(i_1)}}, \cdots, x \in \mathcal{X}_{\delta^{(i_k)}}$ has the same solution x^*_N as NCSP_N. An *irreducible* support set for NCSP_N is a support set $S = \{\delta^{(i_1)}, \cdots, \delta^{(i_k)}\}$ such that no constraint can be removed from S leaving the solution unchanged. ∎

Clearly, the set $\{\delta^{(1)}, \cdots, \delta^{(N)}\}$ corresponding to the whole sample is a support set. The goal of the experimenter is to find a support set of as small cardinality as possible, possibly one of minimal cardinality. However, we stress that, for the purposes of applying the theory, minimality is not required. Among various approaches, a simple greedy algorithm to find an irreducible support set is as follows.

1) Set $L \leftarrow (\delta^{(1)}, \cdots, \delta^{(N)})$ and compute $x^*_N \leftarrow$ solution of the corresponding program NCSP_N;
2) For all $i = 1, \cdots, N$:
   - let $L' \leftarrow L \backslash \delta^{(i)}$, form the program $\mathrm{NCSP}^{(L')}$ with the constraints in L', and let x^* be its solution;
   - if $x^* = x^*_N$, set $L \leftarrow L'$;
3) Output the set $\{i_1, \cdots, i_k\}$ of the indices of the elements in L.

Remark 1: An algorithm like the one above will be regarded as a function $\mathcal{B} : (\delta^{(1)}, \cdots, \delta^{(N)}) \mapsto \{i_1, \cdots, i_k\}$ such that $\{\delta^{(i_1)}, \cdots, \delta^{(i_k)}\}$ is a support set. If $(\delta^{(1)}, \cdots, \delta^{(N)})$ is a random sample, then the cardinality

$$s^*_N := |\mathcal{B}(\delta^{(1)}, \cdots, \delta^{(N)})|$$

is a random variable over Δ^N. Note that s^*_N may depend on the order in which the elements $(\delta^{(1)}, \cdots, \delta^{(N)})$ appear. For example, if both $\{\delta^{(1)}, \delta^{(2)}\}$ and $\{\delta^{(3)}, \delta^{(4)}, \delta^{(5)}\}$ are *irreducible* support sets, then the above greedy algorithm gives $\mathcal{B}(\delta^{(1)}, \delta^{(2)}, \delta^{(3)}, \delta^{(4)}, \delta^{(5)}) = \{3, 4, 5\}$ and $\mathcal{B}(\delta^{(5)}, \delta^{(4)}, \delta^{(3)}, \delta^{(2)}, \delta^{(1)}) = \{1, 2\}$. ∎

We are ready to state our main result.

Theorem 1: Suppose that Assumption 1 holds, and let $\beta \in (0, 1)$. Define the function $\varepsilon : \{0, \cdots, N\} \to [0, 1]$ as follows:

$$\varepsilon(k) := \begin{cases} 1 & \text{if } k = N, \\ 1 - \sqrt[N-k]{\frac{\beta}{N\binom{N}{k}}} & \text{otherwise.} \end{cases}$$

Suppose that some algorithm $\mathcal{B} : \Delta^N \to 2^{\{1, \cdots, N\}}$ designed to select a support set is provided, and let $s^*_N = |\mathcal{B}(\delta^{(1)}, \cdots, \delta^{(N)})|$. Then,

$$\mathsf{P}^N \{\mathsf{V}(x^*_N) > \varepsilon(s^*_N)\} \leq \beta. \quad (4)$$

∎

The function $\varepsilon(k)$ in Theorem 1 is profiled in Figure 1 for $N = 200$ and various values of β. When k is close to 200, $\varepsilon(k)$ is near 1, leading to poor reliability guarantees when the support set cardinality s^*_N is close to the size of the data set. The conclusion becomes progressively stronger as s^*_N becomes smaller. From the figure, it also appears that the quantitative role of the confidence parameter β is minor, so that β can be set to very small values without significantly affecting $\varepsilon(k)$. This is similar to what happens in a convex context where ε does not depend on k, see e.g. [2].

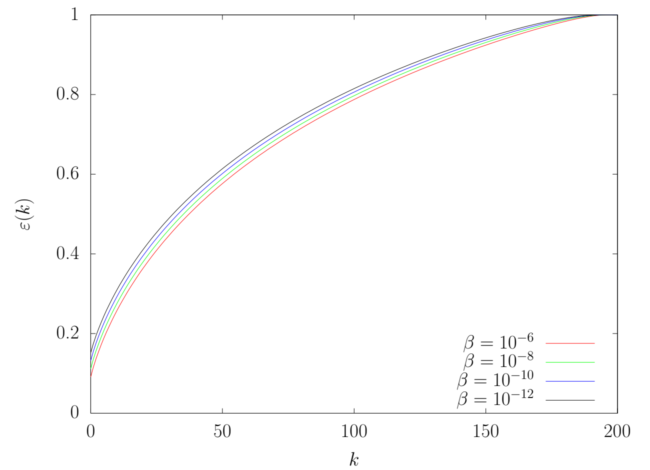

Fig. 1. Plot of $\varepsilon(k)$ versus k for $N = 200$ and for $\beta = 10^{-6}$, 10^{-8}, 10^{-10}, 10^{-12}.

III. Proof of the Main Result

Let $\mathcal{I}_k$ denote the set of all possible $\binom{N}{k}$ selections of k indices $I_k = \{i_1, \cdots, i_k\}$ from $\{1, \cdots, N\}$. For any such selection I_k, let $x^*_{I_k}$ be the solution to the following program:

$$\mathrm{NCSP}^{(I_k)} : \arg\min_{x \in \mathcal{X}} f(x)$$
$$\text{subject to } x \in \mathcal{X}_{\delta^{(i_j)}} \text{ for all } i_j \in I_k.$$

Let us define the subsets $\Delta^N_0, \cdots, \Delta^N_N$ according to the following principle: $(\delta^{(1)}, \cdots, \delta^{(N)}) \in \Delta^N_k$ if and only if $|\mathcal{B}(\delta^{(1)}, \cdots, \delta^{(N)})| = k$. The subsets $\Delta^N_0, \cdots, \Delta^N_N$ form a partition of Δ^N. Let us refine such partition: for each $k = 0, \cdots, N$ and for all $I_k \in \mathcal{I}_k$ define $\Delta^N_{k,I_k} \subseteq \Delta^N_k$ according to the following rule: $(\delta^{(1)}, \cdots, \delta^{(N)}) \in \Delta^N_{k,I_k}$ if and only if $\mathcal{B}(\delta^{(1)}, \cdots, \delta^{(N)}) = I_k$. It holds

$$\Delta^N = \bigcup_{k=0}^{N} \bigcup_{I_k \in \mathcal{I}_k} \Delta^N_{k,I_k}.$$

Let moreover

$$B = \{(\delta^{(1)}, \cdots, \delta^{(N)}) \in \Delta^N \ : \ \mathsf{V}(x^*_N) > \varepsilon(s^*_N)\}$$

and

$$B_{I_k} = \{(\delta^{(1)}, \cdots, \delta^{(N)}) \in \Delta^N \ : \ \mathsf{V}(x^*_{I_k}) > \varepsilon(k)\}.$$

We have

$$\begin{aligned} B = \Delta^N \cap B &= \bigcup_{k=0}^{N} \bigcup_{I_k \in \mathcal{I}_k} \Delta^N_{k,I_k} \cap \{\mathsf{V}(x^*_N) > \varepsilon(s^*_N)\} \\ &= \bigcup_{k=0}^{N} \bigcup_{I_k \in \mathcal{I}_k} \Delta^N_{k,I_k} \cap \{\mathsf{V}(x^*_{I_k}) > \varepsilon(k)\} \\ &= \bigcup_{k=0}^{N-1} \bigcup_{I_k \in \mathcal{I}_k} \Delta^N_{k,I_k} \cap \{\mathsf{V}(x^*_{I_k}) > \varepsilon(k)\} \\ &= \bigcup_{k=0}^{N-1} \bigcup_{I_k \in \mathcal{I}_k} \Delta^N_{k,I_k} \cap B_{I_k}, \end{aligned}$$

where the last-but-one equality holds since $\varepsilon(N) = 1$, hence $\{\mathsf{V}(x^*_{I_k}) > 1\} = \varnothing$. Now focus on any selection I_k of k indices; to fix ideas, consider $I_k = \{1, \cdots, k\}$. Since the definition of B_{I_k} only involves the first k components, B_{I_k} is a cylinder with base in the Cartesian product of the first k domains Δ. Suppose that $(\bar{\delta}^{(1)}, \cdots, \bar{\delta}^{(k)})$ is a point in the base of such cylinder; then a point $(\bar{\delta}^{(1)}, \cdots, \bar{\delta}^{(k)}, \delta^{(k+1)}, \cdots, \delta^{(N)})$ belongs to $\Delta^N_{k,\{1,\cdots,k\}} \cap B_{\{1,\cdots,k\}}$ only if the constraints $x^*_{\{1,\cdots,k\}} \in \mathcal{X}_{\delta^{(k+1)}}, \cdots, x^*_{\{1,\cdots,k\}} \in \mathcal{X}_{\delta^{(N)}}$ are satisfied (this is by definition of $\Delta^N_{k,\{1,\cdots,k\}}$). On the other hand, for any such point

$$\mathsf{V}(x^*_{\{1,\cdots,k\}}) = \mathsf{P}\left\{\delta \in \Delta \ : \ x^*_{\{1,\cdots,k\}} \notin \mathcal{X}_\delta\right\} > \varepsilon(k)$$

by definition of $B_{\{1,\cdots,k\}}$. Therefore, since the components $\delta^{(k+1)}, \cdots, \delta^{(N)}$ are independent,

$$\begin{aligned} &\mathsf{P}^{N-k}\Big\{(\delta^{(k+1)}, \cdots, \delta^{(N)}) \ : \\ &\quad (\bar{\delta}^{(1)}, \cdots, \bar{\delta}^{(k)}, \delta^{(k+1)}, \cdots, \delta^{(N)}) \in \Delta^N_{k,\{1,\cdots,k\}} \cap B_{\{1,\cdots,k\}}\Big\} \\ &= \mathsf{P}^{N-k}\left\{\bigcap_{i=k+1}^{N} \left\{(\bar{\delta}^{(1)}, \cdots, \bar{\delta}^{(k)}, \delta^{(k+1)}, \cdots, \delta^{(N)}) \ : \ x^*_{\{1,\cdots,k\}} \in \mathcal{X}_{\delta^{(i)}}\right\}\right\} \\ &= \prod_{i=k+1}^{N} \mathsf{P}\left\{(\bar{\delta}^{(1)}, \cdots, \bar{\delta}^{(k)}, \delta^{(i)}) \ : \ x^*_{\{1,\cdots,k\}} \in \mathcal{X}_{\delta^{(i)}}\right\} \\ &\le \prod_{i=k+1}^{N} (1 - \varepsilon(k)) = (1 - \varepsilon(k))^{N-k}. \end{aligned}$$

Integrating over the base of the cylinder $B_{\{1,\cdots,k\}}$, we obtain

$$\begin{aligned} &\mathsf{P}^N\left\{\Delta^N_{k,\{1,\cdots,k\}} \cap B_{\{1,\cdots,k\}}\right\} \\ &\quad \le (1 - \varepsilon(k))^{N-k} \, \mathsf{P}^k\left\{\text{base of } B_{\{1,\cdots,k\}}\right\} \\ &\quad \le (1 - \varepsilon(k))^{N-k}. \end{aligned}$$

Recall that, up to this point, the choice of $I_k = \{1, \cdots, k\}$ has been arbitrary. In fact, reasoning exactly in the same way, we obtain $\mathsf{P}^N\left\{\Delta^N_{k,I_k} \cap B_{I_k}\right\} \le (1 - \varepsilon(k))^{N-k}$ for all $I_k \in \mathcal{I}_k$. Therefore, by sub-additivity,

$$\begin{aligned} \mathsf{P}^N\{B\} &\le \sum_{k=0}^{N-1} \sum_{I_k \in \mathcal{I}_k} (1 - \varepsilon(k))^{N-k} \\ &= \sum_{k=0}^{N-1} \binom{N}{k} (1 - \varepsilon(k))^{N-k}. \end{aligned}$$

Finally, substituting the expression for $\varepsilon(k)$ provided in the statement of the theorem we get

$$\mathsf{P}^N\{\mathsf{V}(x^*_N) > \varepsilon(s^*_N)\} = \mathsf{P}^N\{B\} \le \beta.$$

■

IV. Examples

Suppose that u and y are two scalar variables and that the $N = 1250$ i.i.d. observations $(u(i), y(i))$, $i = 1, \cdots, 1250$, depicted in Figure 2 have been collected[1]. The objective is that of constructing an IPM able to provide a prediction interval for future outputs of the system. Specifically, we resort to program (1) with $p = 10$, that is, the central line is a single-layer neural network with vertical offset

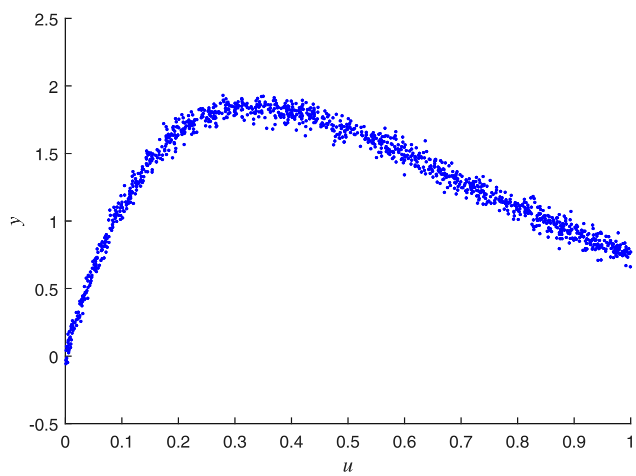

Fig. 2. Data generated by a linear×exponential model plus noise.

$$\sum_{k=1}^{10} \alpha_k \sigma(a_k u + b_k) + c,$$

where $\sigma(x) = 1/(1 + \mathrm{e}^{-x})$ and α_k, a_k, b_k, c, along with the IPM width h, are tunable parameters. Program (1) is a program with linear cost function and non-convex constraints, one for each observation $(u(i), y(i))$. In order to solve it, one has to rely on numerical algorithms for constrained nonlinear programming. We opted for

[1]For the sake of completeness, we let the reader know that the data record was generated according to the model

$$y(i) = 15u(i) \cdot \exp(-3u(i)) + w(i), \tag{5}$$

where the $u(i)$'s were i.i.d. with uniform distribution on $[0, 1]$ and the $w(i)$'s were i.i.d. with Gaussian distribution with zero mean and variance $2.5 \cdot 10^{-3}$. Note, however, that in no way this knowledge about the data generation mechanism is used in the constructions discussed in this example section.

the fmincon function of the Optimization Toolbox of MATLAB. The following incremental usage of fmincon was employed to find an approximate solution to program (1) and, simultaneously, compute a support set.

Incremental procedure

0. Set the initial solution to an arbitrary initialization; we chose

$$\begin{cases} c &= 1 \\ \alpha_k &= 1/k, \\ a_k &= 1/k, \quad k = 1, \ldots, 10. \\ b_k &= 1/k, \\ h &= 0. \end{cases} \tag{6}$$

Set $L \leftarrow \varnothing$;

1. For each data point and in correspondence of the current solution, compute

$$d_i \leftarrow \left| y(i) - \sum_{k=1}^{10} \alpha_k \sigma(a_k u(i) + b_k) - c \right|.$$

If $d_i \leq h$ for all $i = 1, \cdots, N$, then stop and return L and the current solution;

2. Otherwise, let $\bar{i} = \arg\max_i d_i$ and update $L \leftarrow L \cup \{\bar{i}\}$. Run fmincon with initialization as in (6) and with the sole constraints corresponding to the indices in L in place. Update the current solution with the output of the fmincon function. Go to 1.

In words, the procedure progressively updates the solution by adding one data point at a time, until the computed IPM contains all the data points ("if" clause in step 1). The data point to be added at each step is the outermost – and hence most indicative of the way the solution has to be changed to contain all the data points – from the currently computed IPM. By construction, the returned L is a support set, because if the procedure is run with a data set restricted to the data points whose indices are in the obtained L, then the final solution returned by the procedure keeps unchanged. Note that the procedure always terminates, because step 1 is run at most N times, in which case the IPM contains all the data points. Clearly, the hope is that the procedure halts well before this extreme condition, and experimental trials reveal that this is indeed the case.

Based on experience, the incremental procedure described above is able to return solutions with cost close to optimal. The procedure was run for the data record at hand, and the IPM depicted in Figure 3 was obtained. The returned support set, which is also depicted in the figure through red crosses, had cardinality[2] equal to 13. Note that the support points need not be on the boundary of the IPM.

After that the IPM is computed, its reliability is evaluated by Theorem 1. We took $\beta = 10^{-6}$, which gives high confidence

[2]Multiple experiments revealed that when the data are generated according to (5) it is likely that the obtained support set has cardinality between 9 and 15.

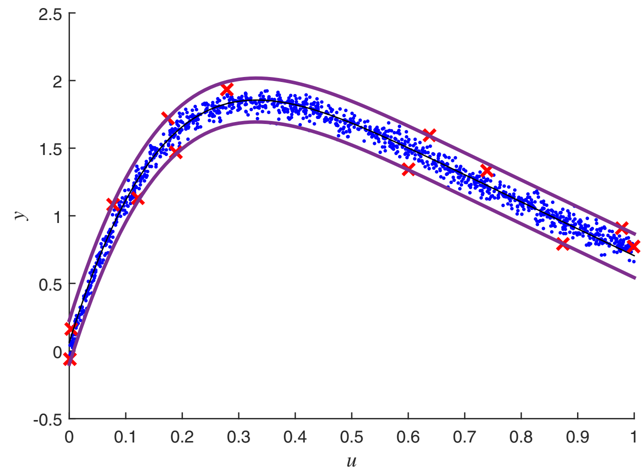

Fig. 3. IPM for the data in Figure 2. A support set of 13 points is marked with red crosses.

$1 - 10^{-6}$, and for the present instance, where $s_N^* = 13$, we drew the conclusion

$$\mathsf{P}\left\{(u,y) : \left| y - \sum_{k=1}^{10} \alpha_{k,N}^* \sigma(a_{k,N}^* u + b_{k,N}^*) - c_N^* \right| > h_N^* \right\}$$
$$\leq \varepsilon(13) = 0.071,$$

where $(\alpha_{k,N}^*, a_{k,N}^*, b_{k,N}^*, c_N^*, h_N^*$ is the solution of the incremental procedure. This means that only 7.1% of the pairs (u, y) are outside the obtained IPM. This result is drawn without resorting to any information other than that carried by the available data set. In particular, no cross-validation is required.

Suppose now that the data record depicted in Figure 4 has been obtained from a different system[3]. Using the incremental procedure described before led to the IPM in Figure 5. The cardinality of the support set was this time 21. Applying Theorem 1 gives with high confidence $1 - \beta = 1 - 10^{-6}$

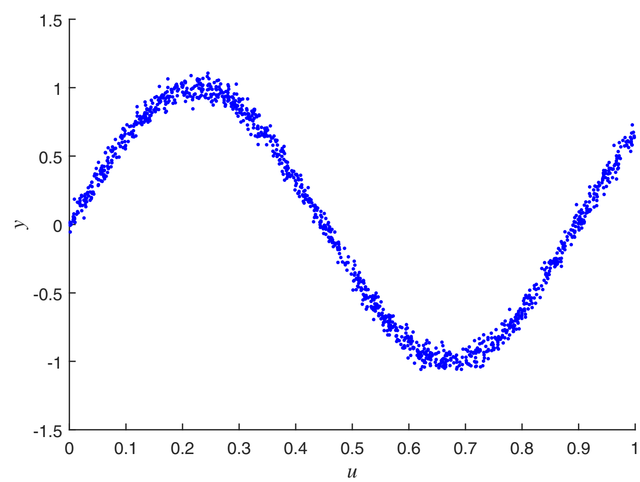

Fig. 4. Data generated by a sinusoidal model plus noise.

[3]This data record was generated with u and w as in the previous experiment by the model $y(i) = \sin(7u(i)) + w(i)$.

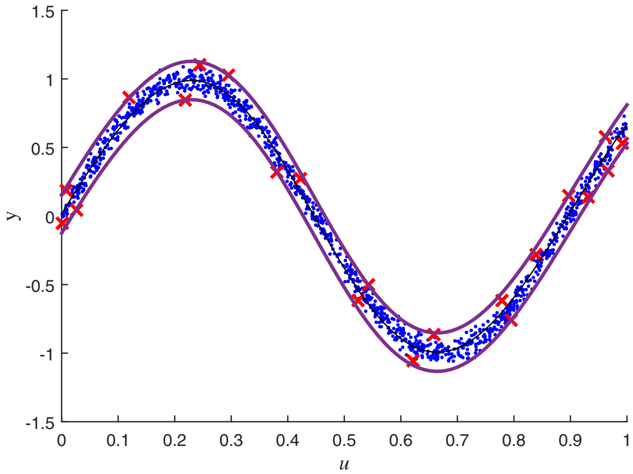

Fig. 5. IPM for the data in Figure 4. A support set of 21 points is marked with red crosses.

that only the $9.7\% = \varepsilon(21)$ of the pairs (u, y) lie outside the obtained IPM. The probability being higher than in the first example reflects that this second construction turned out to be more complex in terms of the support set generating the IPM.

REFERENCES

[1] T. Alamo, R. Tempo, and E. F. Camacho. A randomized strategy for probabilistic solutions of uncertain feasibility and optimization problems. *IEEE Transactions on Automatic Control*, 54:2545–2559, 2009.

[2] T. Alamo, R. Tempo, A. Luque, and D. R. Ramirez. Randomized methods for design of uncertain systems: Sample complexity and sequential algorithms. *Automatica*, 52:160–172, 2015.

[3] J.P. Aubin. *Set-valued analysis*. Birkhäuser, Boston, MA, 1990.

[4] J.P. Aubin and A. Cellina. *Differential inclusions*. Springer-Verlag, Berlin, Germany, 1984.

[5] J.P. Aubin, J. Lygeros, M. Quincampoix, S. Sastry, and N. Seube. Impulse differential inclusions: a viability approach to hybrid systems. *IEEE Transactions on Automatic Control*, 47(1):2–20, 2002.

[6] G. Calafiore and M. C. Campi. Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102, no.1:25–46, 2005.

[7] G. Calafiore and M. C. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, pages 742–753, 2006.

[8] M. C. Campi, G. Calafiore, and S. Garatti. Interval predictor models: identification and reliability. *Automatica*, 45:382–392, 2009.

[9] M. C. Campi and S. Garatti. The exact feasibility of randomized solutions of uncertain convex programs. *SIAM Journal on Optimization*, 19, no.3:1211–1230, 2008.

[10] L.G. Crespo, D.P. Giesy, and S.P. Kenny. Interval predictor models with a formal characterization of uncertainty and reliability. In *Proceedings of the 53rd IEEE Conference on Decision and Control (CDC)*, Los Angeles, USA, 2014.

[11] S. Garatti and M. C. Campi. L-infinity layers and the probability of false prediction. In *Proceedings of the 2009 IFAC Symposium on System Identification (SYSID'09)*, Saint Malo, France, 2009.

[12] L. Jaulin, M. Kieffer, I. Braems, and E. Walter. Guaranteed nonlinear estimation using constraint propagation sets. *International Journal of Control*, 74(18):1772–1782, 2001.

[13] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer, London, UK, 2001.

[14] M. Kieffer, L. Jaulin, and E. Walter. Guaranteed recursive nonlinear state bounding using interval analysis. *International Journal of Adaptive Control and Signal Processing*, 6(3):193–218, 2002.

[15] M. Milanese and C. Novara. Set-membership identification of nonlinear systems. *Automatica*, 40(6):957–975, 2004.

[16] M. Milanese and C. Novara. Set-membership prediction of nonlinear time series. *IEEE Transactions on Automatic Control*, 50(11):1655–1669, 2005.