

Interval Predictor Models for Data with Measurement Uncertainty

Márcio J. Lacerda¹

Luis G. Crespo²

Abstract—An interval predictor model (IPM) is a computational model that predicts the range of an output variable given input-output data. This paper proposes strategies for constructing IPMs based on semidefinite programming and sum of squares (SOS). The models are optimal in the sense that they yield an interval valued function of minimal spread containing all the observations. Two different scenarios are considered. The first one is applicable to situations where the data is measured precisely whereas the second one is applicable to data subject to known biases and measurement error. In the latter case, the IPMs are designed to fully contain regions in the input-output space where the data is expected to fall. Moreover, we propose a strategy for reducing the computational cost associated with generating IPMs as well as means to simulate them. Numerical examples illustrate the usage and performance of the proposed formulations.

I. INTRODUCTION

This paper proposes a parametric metamodeling technique for predicting the range of an output variable given input-output data. In contrast to most metamodeling techniques, we do not use a measurement error term, nor do we make assumptions on the distribution of the data. This paper focuses on models depending linearly on the parameters and polynomially on the input. This structure makes possible to rigorously characterize the range of the predicted output and the uncertainty in the parameters of the model.

A few remarks on the significance and practical use of interval predictor models (IPMs) are in order. There are several conceptual differences between standard metamodeling techniques and IPMs. One of them is that while some techniques aim at interpolating the data, IPMs aim at describing their range. For instance, while each of the functions comprising a Gaussian Process model for a zero-noise level interpolates all the data points, there is at least one member of the family of functions comprising an IPM passing through each data point. None of these functions however, interpolate all the data. In [2], IPMs with a Gaussian radial basis have been used to describe a radiation shielding application for space exploration.

A solution for the IPM problem based on a Linear Matrix Inequality formulation has been presented in [1]. In that paper the data are not affected by uncertainties. The Sum of

Squares (SOS) formulation is a method that allows the characterization of positive polynomials (see [3] and references therein). In the last years, many robust control problems have been solved with the help of SOS formulations, one may cite control design problems [4], [5], filter design problems [6], [7], stability problems [8] and domain of attraction estimation problems [9], [10]. It is also important to cite [11] which have studied the data fitting problem using SOS polynomials. A brief introduction to the SOS method is presented in the Appendix.

The main contribution of this paper is a new methodology for calculating IPMs based on the solution of a semidefinite program (SDP) with SOS constraints. Two formulations are proposed. First, an IPM is designed for a set of data that is known precisely. This formulation is then extended to the case in which the data are subjected to measurement error. This error, which might be caused by biases and randomness in the instrumentation, is bounded by a hyper-rectangular set in the input-output space. The properties of this error might well vary with the input and the output. Two strategies for generating IPMs in the presence of measurement error are developed, one in which measurement error in all data points is considered upfront and another one requiring the consideration of measurement error for the worst-case data points only. The latter approach, which substantially relaxes the computational cost of the alternative method, entails (i) computing the IPM without considering data uncertainty, (ii) determining the data points whose uncertainty set is not fully contained by the IPM, (iii) solving for a second IPM that contains such uncertainty sets, and (iv) prescribing the desired IPM as the union of the two. This practice considerably reduces the number of SOS constraints required.

The paper is organized as follows. Section II introduces the problem formulation and provides some preliminary information. The main results are presented in the Section III. Section V presents a few numerical examples and Section VI concludes the paper.

II. PROBLEM FORMULATION AND PRELIMINARIES

A *Data Generating Mechanism* (DGM) is postulated to act on a vector of input variables, $x \in \mathbb{R}^{n_x}$, to produce an output, $y \in \mathbb{R}^{n_y}$. In this paper the focus will be on the single-input ($n_x = 1$), single-output ($n_y = 1$) case. Assume that N input-output pairs are obtained from the DGM, and denote this by $z = \{z(t)\}$, with $z(t) = (x(t), y(t))$ for $t = 1, \dots, N$, the corresponding data sequence.

¹M. J. Lacerda is with the Department of Electrical Engineering, Federal University of São João del-Rei - UFSJ, 36307-352 São João del-Rei, MG, Brazil. lacerda@ufsj.edu.br

²L. G. Crespo is a Research Aerospace Engineer with the Dynamic Systems and Control Branch, NASA Langley Research Center, Hampton, VA, 23681, USA. luis.g.crespo@nasa.gov

It is desired to build a mathematical model of the DGM based on z which will predict the output corresponding to an unobserved realization of the input. Let $X \subseteq \mathbb{R}^{n_x}$ be a set of input variables, and $Y \subseteq \mathbb{R}^{n_y}$ be a set of outputs which might result from evaluating the model at elements of X . The presence of intrinsic variability, and parametric- and model-form-uncertainty makes it unrealistic to build a model that will predict a single output for a fixed input. Instead, an IPM will predict an interval into which unobserved data is expected to fall. Engineering judgment is used to select a computational model, $y = M(x, p)$ depends polynomially on x , where $p \in \mathbb{R}^{n_p}$ is a parameter. Instead of the standard practice of trying to fit all of the data as closely as possible with M evaluated at a fixed p , the thrust in this work is to provide an interval-valued function of the input where unobserved data is expected to fall.

A. Interval Predictor Models

An IPM is fully prescribed by its upper and lower boundaries. In this paper, such boundaries are prescribed by two polynomial functions, so $I(x) = [f_l(x), f_u(x)]$ where $f_l(x) < f_u(x)$ for all $x \in X$. These functions will be described as

$$f_u(x) = \sum_{i=0}^{d_u} u_i x^i, \quad f_l(x) = \sum_{i=0}^{d_l} l_i x^i \quad (1)$$

where d_u and d_l are the maximum degrees of the upper and lower function respectively. The decision variables are the coefficients of the upper function u_i , $i = 1, \dots, d_u$, and the coefficients of the lower function l_i , $i = 1, \dots, d_l$. Note that the IPM boundaries might not have the same polynomial degree.

III. MAIN RESULTS

The conditions for designing an IPM based on N input-output data points, $x(t)$, $y(t)$ with $t = 1, \dots, N$, exempt from measurement uncertainty are given by:

Optimization Program 1: If there exist $\gamma > 0$, polynomials $f_u(x)$ and $f_l(x)$ as in (1), and a SOS multiplier $\lambda(x)$ for a given polynomial $p(x) \leq 0$ describing the input domain of interest, such that

$$\min \gamma \quad \text{s.t.} \quad (2)$$

$$\mathbb{E}_x [f_u(x) - f_l(x)] < \gamma \quad (2)$$

$$y(t) - f_l(x(t)) > 0, \quad t = 1, \dots, N \quad (3)$$

$$y(t) - f_u(x(t)) < 0, \quad t = 1, \dots, N \quad (4)$$

$$f_u(x) - f_l(x) + \lambda(x)p(x) \quad \text{is SOS} \quad (5)$$

where $\mathbb{E}_x[\cdot]$ is the expected value with respect to x , then

$$I_{y(x)} = \{(x, y) : x \in X, f_l(x) \leq y(x) \leq f_u(x)\}$$

is the IPM of minimal expected spread containing the data.

When x is a standard joint random vector with joint probability density $f_x(x)$, the cost function in (2) can be

calculated analytically. Otherwise, the sample mean based on the data in z should be used. In the former case, the expected value can be computed analytically for each of the monomials as

$$\mathbb{E}[x^n] = \int_{\underline{x}}^{\bar{x}} x^n f_x(x) dx$$

where $X = \{x : \underline{x} \leq x \leq \bar{x}\}$. Conditions (3) and (4) ensure that all the data points are inside the IPM, i.e., all the measurements $y(t)$ are between the upper function $f_u(x)$ and the lower function $f_l(x)$. Condition (5) assures that the difference between the functions $f_u(x)$ and $f_l(x)$ is positive in the interval of interest. Remember that for univariate polynomials the SOS constraint is equivalent to positivity¹. The polynomial $p(x)$ can be constructed as

$$p(x) = (x - \underline{x})(x - \bar{x}) \quad (6)$$

In this way $p(x)$ is guaranteed to be negative or equal to zero in X . Note that Equation (5) might lead to an IPM for which $f_u - f_l$ is negative outside X . This would not have been the case if we would have made $f_u - f_l$ SOS. As such, Equation (5) renders IPM with an improved performance. It is important to remember that optimization program 1 is a convex problem that can be solved using standard semidefinite software such as SeDuMi [12].

Let us now consider the case in which the data is subject to measurement error. Assume that each data point is expected to fall within a rectangular input-output region defined as

$$\underline{x}(t) \leq x(t) \leq \bar{x}(t), \quad \text{and} \quad \underline{y}(t) \leq y(t) \leq \bar{y}(t) \quad (7)$$

where $t = 1, \dots, N$. An IPM that contains such a region for all data points can be identified by solving the following optimization program:

Optimization Program 2: If there exist $\gamma > 0$, polynomials $f_u(x)$ and $f_l(x)$, and the SOS multipliers $\lambda(x)$, $\bar{\lambda}_t(x)$, $\underline{\lambda}_t(x)$, $t = 1, \dots, N$; for the given polynomials $p(x) < 0$ (describing X) and $p_t(x) \leq 0$ (describing the input domain of measurement error $[\underline{x}(t), \bar{x}(t)]$), such that

$$\min \gamma \quad \text{s.t.}$$

$$\mathbb{E}_x [f_u(x) - f_l(x)] < \gamma \quad (8)$$

$$f_u(x) - \bar{y}(t) + \bar{\lambda}_t(x)p_t(x) \text{ for } t = 1, \dots, N \text{ is SOS} \quad (9)$$

$$\underline{y}(t) - f_l(x) + \underline{\lambda}_t(x)p_t(x) \text{ for } t = 1, \dots, N \text{ is SOS} \quad (10)$$

$$f_u(x) - f_l(x) + \lambda(x)p(x) \text{ is SOS} \quad (11)$$

then

$$I_{y(x)} = \{(x, y) : x \in X, f_l(x) \leq y(x) \leq f_u(x)\}$$

is the IPM of minimal expected spread containing the data.

¹We refer the reader to the Appendix for more details about the SOS formulation

Optimization program 2 requires setting a polynomial $p_t(x)$ for each $t = 1, \dots, N$. Following the same rationale used in (6), one can write

$$p_t(x) = (x - \underline{x}(t))(x - \bar{x}(t)), \quad t = 1, \dots, N$$

The only difference between optimization program 1 and optimization program 2 is the requirement for two SOS multipliers for each data point. As such, large values of N and large polynomial degrees can create computational difficulties. An alternative approach for mitigating such difficulties is presented in Algorithm 1.

Algorithm 1

- 1) Use Program 1 to calculate an IPM after ignoring uncertainty in the data. Denote the resulting IPM as $I_{y(x),1}$.
- 2) Use Bernstein polynomials [13] to determine which uncertainty regions are not fully contained in $I_{y(x),1}$. Denote by N_v be the number of such regions.
- 3) Use Program 2 to solve for an IPM that contains the N_v regions. Denote the resulting IPM as $I_{y(x),2}$.
- 4) An IPM fully enclosing the N uncertainty regions is given by:

$$I_{y(x)} = I_{y(x),1} \cup I_{y(x),2} \quad (12)$$

Note that this equation leads to an IPM containing the union of two IPMs. Even though this IPM predicts a single interval for each value of x as desired, the resulting prediction might be overly conservative since such an interval might include large input-output regions where no data is present. Alternatively, one might consider adding the constraint $I_1(x) \subseteq I_2(x)$ when solving for $I_2(x)$. This algorithm requires solving optimization programs 1 and 2. However, in this case, optimization program 2 is solved only with N_v uncertain regions where $N_v \leq N$. The resulting IPM in (12) is given by the union of two IPMs. As such, the boundaries of this IPM have derivative discontinuities where the boundaries of I_1 and I_2 intersect.

Remark 1: The image of a hyper-rectangle when mapped by a multivariable polynomial is a bounded interval. By expanding that polynomial using a Bernstein basis over that rectangle, rigorous bounds to such an interval can be calculated using mere algebraic manipulations. Bernstein polynomials can be used to determine if a hyper-rectangle in the input-output space is fully contained within an IPM or not. The tightness of the bounds, as well as the outcome of the set containment determination, depend exclusively on the level of refinement used to describe the rectangle, i.e., a partition of sub-rectangles constituting the rectangle. The bounds can be made arbitrarily tight and the set containment test conclusive by using a sufficiently fine partition of the rectangle.

Means to give a functional representation to the proposed IPMs is presented next. This is a setting in which the IPM is comprised by infinitely many input output functions fully and tightly enclosed by the IPM envelopes. This representation enables the systematic generation of input-output functions used for simulating the IPM. Until this point there are no constraints imposed directly in the parameter space of the coefficients of the polynomial functions f_u and f_l . In [14] a parametric approach to obtain a functional representation of the IPM is presented. The idea was generating an IPM by letting the coefficients of a polynomial vary within a bounded range. The cartesian product of such ranges yield an hyper-rectangular set, which in contrast to the ellipsoids considered in [1] have the advantage that the range of an individual parameter is not affected by the value taken by the others. In particular, if $y = p^\top \phi(x)$ is a linear combination of monomials, the boundaries of the IPM corresponding to $\underline{p} \leq p \leq \bar{p}$ are the non-polynomial functions

$$\bar{y}(x, \bar{p}, \underline{p}) = \bar{p}^T \left(\frac{\phi(x) + |\phi(x)|}{2} \right) + \underline{p}^T \left(\frac{\phi(x) - |\phi(x)|}{2} \right) \quad (13)$$

and

$$\underline{y}(x, \bar{p}, \underline{p}) = \bar{p}^T \left(\frac{\phi(x) - |\phi(x)|}{2} \right) + \underline{p}^T \left(\frac{\phi(x) + |\phi(x)|}{2} \right) \quad (14)$$

The developments that follow enable giving a functional representation to the IPMs derived here. To achieve this one must have $f_u(x) = \bar{y}(x, \bar{p}, \underline{p})$ and $f_l(x) = \underline{y}(x, \bar{p}, \underline{p})$. When $x > 0$, Equations (13) and (14) yield $f_u(x) = \bar{p}^T \phi(x)$ and $f_l(x) = \underline{p}^T \phi(x)$ that is the same representation (1) used in this paper. So, in order to construct the same type of IPMs in [14] using the proposed formulations, we need to use the same basis for f_u and f_l , and add the constraint $\underline{p} \leq p \leq \bar{p}$. The next Corollary summarizes this process

Corollary 1: By solving optimization program 1 (or 2) for $x > 0$ and $u_k \geq l_k$, $k = 0, \dots, d$ we obtain an IPM such that, the family of all polynomial functions with coefficients lying between u_k and l_k , $k = 0, \dots, d$ are within the IPM. An input transformation can be used to map a non-positive set X into the first orthant (see example 3 for details).

Remark 2: It is important to emphasize that the bound γ present in both optimization programs 1 and 2 is the variable used to measure how good the approximation is. The bound γ can be used as an indicative of when we should consider using higher order polynomial functions in the IPM.

IV. RELIABILITY

The reliability of an IPM is defined as the probability of a future scenario falling outside the predicted interval. When data is measured precisely an scenario is a data point. When the data is subject to measurement error an scenario is an uncertainty box. Scenario optimization enables to rigorously bound the reliability of an IPM using a non-asymptotic,

distribution-free formula [1], [14]. This relationship describes the interplay between the reliability of the IPM, the complexity of the model (measured by the number of design variables), and the amount of information available (measured by the number of scenarios N). This framework enables assessing the reliability of the IPMs derived herein. The results of such an analysis however, are omitted due to space limitations.

V. NUMERICAL EXPERIMENTS

The numerical experiments that follow illustrate the optimization programs proposed above. The routines were implemented in MATLAB, version 8.3.0.532 (R14) using the packages Yalmip [15], [16] and SeDuMi [12] in an Intel(R) Core(TM) i5-4210, 1.7 GHz, 8 GB RAM, Windows 10.

Example 1. The DGM considered in [14] and given by

$$y(t) = x^2 \cos(x) - \sin(3x) \exp(-x^2) - x - \cos(x^2) + xg \quad (15)$$

will be used in the examples that follow. The input x is independent and identically distributed (i.i.d.) with a uniform distribution over the support $X = [-5.5, 5.5]$ and g is i.i.d. with standard normal distribution. A set of $N = 50$ observations have been drawn and used to form the data sequence \mathbf{z} . The structure of the DGM is provided for the sake of clarity and no information about it will be used to construct the IPMs.

First we consider the case in which there is no uncertainty in the data. To this end we will use optimization program 1 under two settings: i) polynomials $f_u(x)$ and $f_l(x)$ with degrees $d_u = d_l = 6$, and ii) polynomials $f_u(x)$ and $f_l(x)$ with degrees $d_u = d_l = 8$. In both cases we use a SOS multiplier $\lambda(x)$ of degree $d_s = 2$. The solution to this programs yields $\gamma = 47.20$ and $\gamma = 18.06$ respectively. Figure 1 shows the resulting IPMs. The data are marked with \times , the black dashed lines depict the first IPM and the solid red lines depict the second one. All the observations are enclosed by the IPMs, but second IPM does so much tightly by virtue of the higher polynomial degree. This is reflected in the resulting value of γ .

Table I presents the γ values obtained by optimization program 1 for different values of $d_u = d_l$ while keeping everything else the same. As expected, the bound γ decreases rapidly when the degree of the polynomial functions increases. Table I also lists the time in seconds required by SeDuMi [12] to solve the respective SDP and the number of scalar decision variables S_v for the different degrees. It can be seen that the time required to solve the convex problem is reasonable (less than one second) for all cases.

Example 2. We now consider the case in which the data generated by the DGM (15) is subjected to measurement error. For the sake of simplicity, the limiting vertices of the rectangular region (7) describing the uncertainty set are

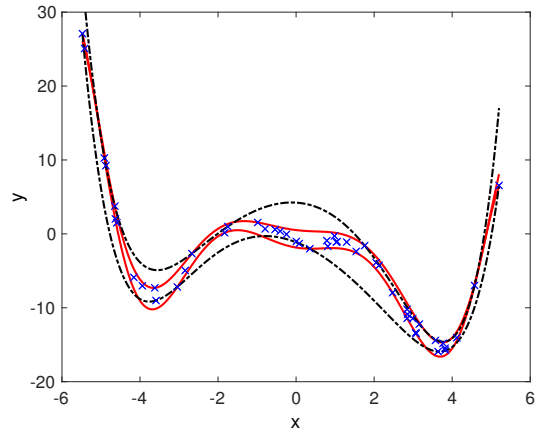


Fig. 1. Optimization program 1 using a SOS multiplier $\lambda(x)$ of degree $d_s = 2$ and polynomial functions with $d_u = d_l = 6$ (black dashed lines) and $d_u = d_l = 8$ (red solid lines).

TABLE I
BOUNDS γ OBTAINED BY OPTIMIZATION PROGRAM 1 WITH THE RESPECTIVE TIME REQUIRED TO SOLVE THE PROBLEM AND NUMBER OF SCALAR DECISION VARIABLES FOR DIFFERENT DEGREES $d_u = d_l$.

$d_u = d_l$	2	3	4	5	6	7	8
γ	178.19	169.45	61.59	61.56	47.20	42.37	18.06
Time(s)	0.099	0.100	0.157	0.149	0.172	0.216	0.318
S_v	19	23	27	31	35	39	43

$\underline{x}(t) = 0.9x(t)$, $\bar{x}(t) = 1.1x(t)$, $\underline{y}(t) = 0.9y(t)$, and $\bar{y}(t) = 1.1y(t)$.

We now apply optimization program 2 to solve for an IPM. Figure 2 shows the resulting IPM. The uncertainty regions are shown as rectangles. The IPM was designed with $d_u = d_l = 5$, SOS multiplier $\lambda(x)$ of degree $d_s = 6$ and SOS multipliers $\bar{\lambda}_t(x)$ and $\underline{\lambda}_t(x)$ of degree $d_t = 2$. As intended, the IPM fully contains all the regions where the data is expected to be.

Next, we use Algorithm 1 to solve the very same problem. This approach yields to $I_1(x)$, which is the IPM shown in Figure 3. Note that some of the uncertainty regions are not fully contained by the IPM. In particular, the $N_v = 19$ regions shown as red boxes extend beyond the IPM boundaries of I_1 .

The application of Step 3 in Algorithm 1 with the inclusion of these N_v uncertainty regions leads to $I_2(x)$ (not shown). The final IPM, computed via (12), is shown in Figure 4. The IPM in Figure 4 is very similar to the IPM in Figure 2. However, the main advantage of the sequential algorithm is the smaller number of decision variables employed to solve the problem. For instance, in this example, optimization program 2 makes use of 944 scalar decision variables while the procedure presented in Algorithm 1 involves 418 scalar decision variables, i.e., a reduction of more than 50% in the

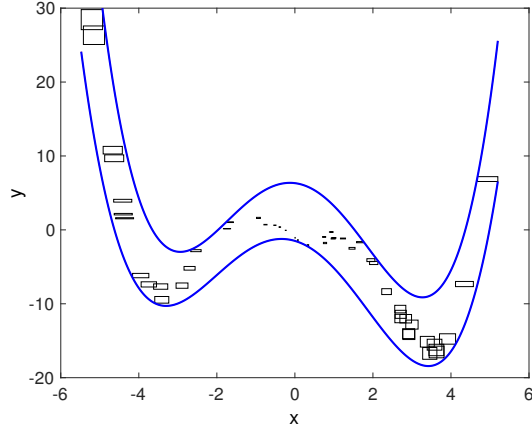


Fig. 2. Optimization program 2 using a SOS multiplier $\lambda(x)$ of degree $d_s = 6$, polynomial functions with $d_u = d_l = 5$ and SOS multipliers $\bar{\lambda}_t(x)$ and $\underline{\lambda}_t(x)$ of degree $d_t = 2$.

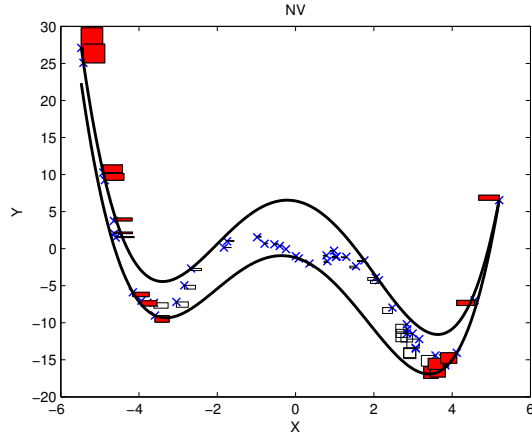


Fig. 3. Step 1 of Algorithm 1 using a SOS multiplier $m_s(x)$ of degree $d_s = 2$, polynomial functions with $d_u = d_l = 5$. The uncertain regions that violate the IPM_1 are colored.

number of variables.

Example 3. In this example we use Corollary 1 to give a functional representation to one of the IPMs proposed. Remember that Corollary 1 requires $x(t)$ to be positive, so we apply a constant shift to the original data so $x(t) = x(t) + 6$. An IPM for the same data sequence in Example 1 for $d_u = d_l = 5$, $u_k \geq l_k$ for $k = 0, \dots, d$, and a SOS multiplier $\lambda(x)$ of degree $d_s = 2$ leads to an IPM with boundaries

$$f_u(x) = 50.6268 - 53.7433x + 18.0955x^2 - 2.4302x^3 + 0.1248x^4 - 0.0015x^5$$

and

$$f_l(x) = 45.9672 - 54.0068x + 18.0955x^2 - 2.4303x^3 + 0.1248x^4 - 0.0015x^5$$

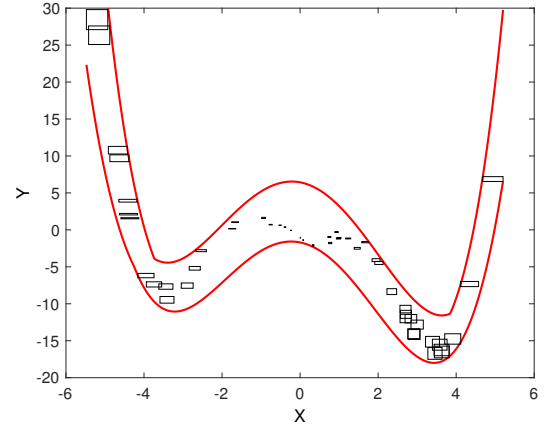


Fig. 4. IPM obtained from Algorithm 1.

Observing the boundaries of the IPM given above one can say that the monomials of order zero and one contribute the most to the spread in the IPM. A Monte Carlo simulation with 1000 different polynomials of order five, with coefficients between \underline{p} , i.e., the coefficients of $f_u(x)$, and \bar{p} , i.e., the coefficients of $f_l(x)$, are shown in Figure 5. The green area represents the range of the 1000 functions and indicates that all the generated polynomials lie inside the IPM.

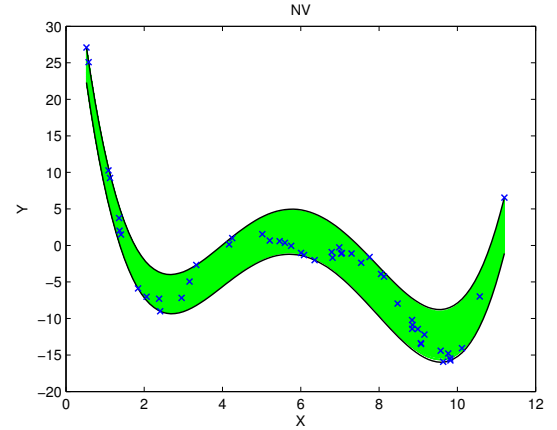


Fig. 5. Corollary 1 with optimization program 1 using a SOS multiplier $\lambda(x)$ of degree $d_s = 2$, polynomial functions with $d_u = d_l = 5$. Monte Carlo simulation with 1000 different polynomial functions.

IPMs capture the discrepancy between the unknown structure of the DGM and the model as parametric uncertainty. This discrepancy might be caused by measurement noise and model-form uncertainty. IPMs can be used to characterize the prediction error of physics-based models and to prescribe the input domain where a models' prediction is acceptable. Let $y = f(x)$ be a calibrated physics-based model of a DGM, and $I(x)$ be an IPM based on the sequence $z = \{(x(t), y(t) - f(x(t)))\}$ for $t = 1, \dots, N$. The model $y = f(x) + I(x)$ describes the phenomenon of interest. Furthermore, the set

$\hat{X} = \{x : \delta_y < \hat{\delta}\} \subset X$ for a suitable value of $\hat{\delta}$, where $\delta_y = f_u(x) - f_l(x)$, prescribes the input domain where the prediction is an acceptable representation of the DGM.

VI. CONCLUSIONS

This paper proposes optimization-based strategies for calculating interval predictor models based on a limited number of observations. We consider the cases in which the data is known precisely, and the case in which the data is subject to measurement error. This error might be caused by the intrinsic limitations of the metrology system used to measure the data. The solution strategies were formulated as semidefinite programs with sum of squares constraints, where the coefficients of the IPM boundaries are polynomials and the corresponding coefficients are the design variables. Extensions to the multi-input case as well as the use of the proposed technique for a realistic application are under investigation.

ACKNOWLEDGMENT

The authors acknowledge helpful discussions with Peter Seiler. This work was supported by the São Paulo Research Foundation (FAPESP) grant 2015/00269-5 and the NASA Langley administration.

REFERENCES

- [1] M. C. Campi, G. Calafiore, and S. Garatti, "Interval predictor models: Identification and reliability," *Automatica*, vol. 45, no. 2, pp. 382–392, February 2009.
- [2] L. G. Crespo, S. P. Kenny, D. P. Giesy, R. B. Norman, and S. R. Blattnig, "Application of interval predictor models to space radiation shielding," in *Proceedings of the 18th AIAA Non-Deterministic Approaches Conference*, San Diego, CA, USA, January 2016, pp. 1–23.
- [3] P. A. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization," Ph.D. dissertation, California Institute of Technology, Pasadena, CA, 2000.
- [4] F. Wu and S. Prajna, "A new solution approach to polynomial LPV system analysis and synthesis," in *Proceedings of the 2004 American Control Conference*, Boston, MA, USA, June/July 2004, pp. 1362–1367.
- [5] G. Valmorbida, S. Tarbouriech, and G. Garcia, "Design of polynomial control laws for polynomial systems subject to actuator saturation," vol. 58, no. 7, pp. 1758–1770, July 2013.
- [6] P. Li, J. Lam, and G. Chesi, "On the synthesis of linear \mathcal{H}_∞ filters for polynomial systems," vol. 61, no. 1, pp. 31–36, January 2012.
- [7] M. J. Lacerda, G. Valmorbida, and P. L. D. Peres, "Linear filter design for continuous-time polynomial systems with \mathcal{L}_2 -gain guaranteed bound," in *Proceedings of the 54th IEEE Conference on Decision and Control*, Osaka, Japan, December 2015, pp. 5026–5030.
- [8] A. A. Ahmadi and P. A. Parrilo, "Non-monotonic Lyapunov functions for stability of discrete time nonlinear and switched systems," in *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, December 2008, pp. 614–621.
- [9] U. Topcu, A. Packard, P. Seiler, and G. Balas, "Robust region-of-attraction estimation," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 137–142, 2010.
- [10] D. Henrion and M. Korda, "Convex computation of the region of attraction of polynomial control systems," vol. 59, no. 2, pp. 297–312, February 2014.
- [11] A. Magnani, S. Lall, and S. Boyd, "Tractable fitting with convex polynomials via sum-of-squares," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Seville, Spain, December 2005, pp. 1672–1677.

- [12] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11, no. 1–4, pp. 625–653, 1999, <http://sedumi.ie.lehigh.edu/>.
- [13] L. G. Crespo, S. P. Kenny, and D. P. Giesy, "Reliability analysis of polynomial systems subject to p-box uncertainties," *Mechanical Systems and Signal Processing*, vol. 37, no. 1–2, pp. 121–136, May/June 2013.
- [14] —, "Interval predictor models with a linear parameter dependency," *ASME Journal of verification, validation and uncertainty quantification*, vol. 1, no. 2, pp. 1–10, January 2016.
- [15] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proceedings of the 2004 IEEE International Symposium on Computer Aided Control Systems Design*, Taipei, Taiwan, September 2004, pp. 284–289.
- [16] —, "Pre- and post-processing sum-of-squares programs in practice," *IEEE Transactions on Automatic Control*, vol. 54, no. 5, pp. 1007–1011, May 2009.
- [17] S. Prajna, A. Papachristodoulou, P. Seiler, and P. A. Parrilo, "SOSTOOLS and its control applications," in *Positive Polynomials in Control*, ser. Lecture Notes in Control and Information Sciences, D. Henrion and A. Garulli, Eds. Berlin: Springer-Verlag, 2005, vol. 312, pp. 273–292.
- [18] A. Papachristodoulou, J. Anderson, G. Valmorbida, S. Prajna, P. Seiler, and P. A. Parrilo, *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*, <http://arxiv.org/abs/1310.4716>, 2013, available from <http://www.eng.ox.ac.uk/control/sostools>.

APPENDIX

A polynomial $F(x)$ is a SOS if it can be written as

$$F(x) = \sum_{i=1}^k f_i^2(x)$$

If $F(x)$ is SOS, then $F(x) \geq 0$ and $F(x)$ is of even degree. Suppose $F(x)$, of degree $2d$ and let $z(x)$ be a vector of all monomials of degree less than or equal to d . $F(x)$ is SOS if and only if there exist Q such that $Q > 0$ and $F(x) = z(x)^T Q z(x)$. If Q is a feasible solution of the SDP, then factorize $Q = VV^T$, and write $V = [v_1 \ \dots \ v_r]$ so

$$F(x) = z(x)^T VV^T z(x) = \|V^T z(x)\|^2 = \sum_{i=1}^r (v_i^T z(x))^2$$

where r is the rank of Q . For univariate polynomials of even degree, the SOS constraint is equivalent to the non-negativity constraint.

Example 4. Consider the polynomial $f(x) = 4x^4 - 4x^3 + 6x^2$. $f(x)$ as a quadratic function of monomials

$$f(x) = \begin{bmatrix} x^2 \\ x \end{bmatrix}^T \begin{bmatrix} 4 & -2 \\ -2 & 6 \end{bmatrix} \begin{bmatrix} x^2 \\ x \end{bmatrix}$$

As $Q = \begin{bmatrix} 4 & -2 \\ -2 & 6 \end{bmatrix}$ is positive definite it can be factorized, for instance using a Cholesky factorization yields

$$f(x) = \begin{bmatrix} x^2 \\ x \end{bmatrix}^T \begin{bmatrix} 2 & -1 \\ 0 & \sqrt{5} \end{bmatrix}^T \begin{bmatrix} 2 & -1 \\ 0 & \sqrt{5} \end{bmatrix} \begin{bmatrix} x^2 \\ x \end{bmatrix}$$

or

$$f(x) = (2x^2 - x)^2 + 5x^2$$

that is a SOS.

Several numerical tools are available to solve this problem [17], [16], [18].