Optim Eng (2006) 7:459–478
DOI 10.1007/s11081-006-0350-4

Numerical approaches for collaborative data processing

Pete Seiler · Michael Frenklach · Andrew Packard · Ryan Feeley

Received: 7 January 2004 / Revised: 10 February 2005

Abstract We present an approach to uncertainty propagation in dynamic systems, exploiting information provided by related experimental results along with their models. The approach relies on a solution mapping technique to approximate mathematical models by polynomial surrogate models. We use these surrogate models to formulate prediction bounds in terms of polynomial optimizations. Recent results on polynomial optimizations are then applied to solve the prediction problem. Two examples which illustrate the key aspects of the proposed algorithm are given. The proposed algorithm offers a framework for collaborative data processing among researchers.

Keywords Model validation · Prediction · Sums-of-squares polynomials · Semidefinite programming

1 Introduction

The goal of many scientific fields is to understand a complex physical process. A common mode of investigation is for individual researchers to study semi-isolated aspects of this complicated process. The community then faces the problem of systematically combining data from many researchers. Optimization techniques for this data processing problem will be discussed in this paper.

In our previous work (Frenklach et al., 2002, 2004), we addressed the data processing problem in the context of a "real-world" example: chemical kinetics of pollutant formation in combustion of natural gas. This example was drawn from the latest GRI-

This work was supported by the National Science Foundation, Information Technology Research Program, Grant No. CTS-0113985.

P. Seiler (✉)
Honeywell Labs in Minneapolis, MN
e-mail: peter.j.seiler@gmail.com

M. Frenklach · A. Packard · R. Feeley
Department of Mechanical Engineering at the University of California, Berkeley

Mech release (Smith et al., http://www.me.berkeley.edu/gri_mech/) which includes a process model consisting of 325 reversible reactions among 53 chemical species, trained on 77 well-documented and expert-evaluated experimental observations. Each experimental result, when coupled with a model, implicitly contains information that could reduce the uncertainty in the process model (e.g., uncertainty in some of the thermochemical parameters). The extent of reduction depends on the specific data processing method. We introduced a collaborative data processing algorithm and used the GRI-Mech dataset to compare it to alternative forms of data processing. A collaborative data processing algorithm is one which makes predictions using data from many researchers within a given scientific community. We concluded that the proposed collaborative algorithm extracts more information from experimental data and prevents false "controversies" arising between researchers.

This paper generalizes the algorithm presented in our previous work. This algorithm can be used to predict the range of possible outcomes of a modeled physical process, knowing to some accuracy the outcomes of several related, but different processes. We take a deterministic approach to prediction that only requires knowledge of an upper bound on the experimental error.

This paper differs from (Frenklach et al., 2004) mainly in two respects. First, this paper provides full details on the numerical implementation of the proposed algorithm. This leads to a more general algorithm which is applicable to data processing problems outside the field of chemical kinetics. Second, this paper utilizes simple examples to illustrate aspects of the algorithm which are not readily apparent in the large-scale chemical kinetics problem.

The paper has the following outline. In the next section, we formulate the model validation and prediction problems. In Section 3, we use a solution mapping technique to approximate the mathematical models by polynomial *surrogate* models. We use these surrogate models to recast the problems as polynomial optimizations. Recent results on polynomial optimizations (Parrilo, 2000, 2003; Lasserre, 2001) are discussed in Section 4 and then applied to the collaborative data processing problem in Section 5. Two examples of the proposed algorithm are given in Section 6.

2 Formulation of the model validation and prediction problems

In this section, we describe a mathematical formalism for model validation and prediction using experimental data. First, we introduce some basic terminology concerning models and experimental data. Then we pose the validation and prediction problems within a deterministic framework.

Let P denote a *physical process* and Y a *variable of interest* related to this process. An *experiment*, denoted E, is a realization of P where y, the value of Y, is measured with some uncertainty. Associated with each experiment is a *dataset unit*, (d, u, M), which consists of the measured value, reported uncertainty, and a mathematical model. The *reported uncertainty*, $u \in \mathbb{R}$, is a hard bound on the experimental error, so the measured value d and actual value y are related by $|d - y| \leq u$. In addition to d and u, an unambiguous description of the experiment is necessary to complete the analysis. In our work this takes the form of a *mathematical model* of the process. This function, $M : \mathbb{R}^n \to \mathbb{R}$, models the influence of the *parameter vector*, $x \in \mathbb{R}^n$, on the variable of interest Y.

In most practical problems, we have prior information concerning the parameter vector. For example, we may know that a particular parameter, based on physical considerations, must be positive. We define the *prior information* to be a set, $\mathcal{H} \subset \mathbb{R}^n$, known to contain all meaningful values of the parameter vector. We make the mild assumption that the prior information can be specified by a set of c polynomial inequalities: $\mathcal{H} \doteq \{x \in \mathbb{R}^n \,:\, g_k(x) \leq 0,\ 1 \leq k \leq c\}$, e.g., the prior information $-1 \leq x_1 \leq 1$ and $x_2 \leq x_1$ written as $-1 - x_1 \leq 0, x_1 - 1 \leq 0, x_2 - x_1 \leq 0$.

Each $x \in \mathcal{H}$ yields a possible outcome for the quantity of interest Y, namely $M(x)$. If $|M(x) - d| \leq u$ then the outcome, uncertainty and model (i.e., the dataset unit) are consistent at x. On the other hand, if $|M(x) - d| > u$ then the dataset unit is inconsistent at x. The model validation problem is to determine if a single parameter vector can make a set of dataset units consistent. Two key assumptions are made in the proposed approach. First, the physical processes have models with known structure, i.e. all the uncertainty lies in the values of a finite set of parameters. Second, the models are interrelated through their dependence on a common set of uncertain parameters.

Problem 1 (Model validation). Let $\{P_k\}_{k=1}^m$ denote physical processes, $\{E_k\}_{k=1}^m$ the associated experiments, and $\{(d_k, u_k, M_k)\}_{k=1}^m$ the dataset units. Assume that the mathematical models, $\{M_k\}_{k=1}^m$, are all defined on the common domain, $\mathcal{H}$. The model validation problem is to find $\bar{x} \in \mathcal{H}$ such that $|M_k(\bar{x}) - d_k| \leq u_k$ for all k, or prove that no such vector exists.

The set of parameter vectors that are consistent with all of the current information is defined as $\mathcal{F} \doteq \cap_{k=1}^m \{x \in \mathcal{H} \,:\, |M_k(x) - d_k| \leq u_k\}$. Each set in the intersection represents the parameters that are consistent with one dataset unit. The model validation problem is just a feasibility problem: Find $\bar{x} \in \mathcal{F}$ or prove $\mathcal{F} = \emptyset$. Therefore $\mathcal{F}$ is referred to as the *feasible set of parameters*. If $\mathcal{F} = \emptyset$, then an inconsistency exists among the available information. However, the existence of $\bar{x} \in \mathcal{F}$ provides some confidence in the fidelity of the models/data.

We can then use the available experimental results to make a prediction about an arbitrary process P_0, for which no measured outcome exists. We refer to P_0 as the *predicted process* and denote its quantity of interest and mathematical model by y_0 and M_0, respectively. Any $x \in \mathcal{F}$ is consistent with all the current information and hence it yields a meaningful prediction, $M_0(x)$. The prediction problem posed below is to find the range of possible outcomes of the predicted process that are consistent with all available information.

Problem 2 (Prediction). The prediction problem is to find the range of possible outcomes of the predicted process: $R \doteq [L, U]$ where:

$$L \doteq \min_{x \in \mathcal{F}} M_0(x), \qquad U \doteq \max_{x \in \mathcal{F}} M_0(x). \tag{1}$$

In the next section, we make the validation and prediction problems more concrete, and address issues involving approximation and nonconvexity.

3 An approach to solve the validation and prediction problems

Our approach to the validation and prediction problem involves two steps. First, we create polynomial approximations to the actual models. Then we use these polynomial approximations to ascertain emptiness and/or compute bounds on L and U. The polynomial approximations reduce the cost in evaluating the mathematical models and we can exploit the algebraic properties of polynomials to formulate convex relaxations.

3.1 Solution mapping

We use a solution mapping technique (Frenklach, 1984; Miller and Frenklach, 1983; Frenklach and Miller, 1985) to approximate a mathematical model with a polynomial. As a brief explanation of this technique, consider a mathematical model, $y = M(x)$, specified by a parametrized differential equation model, parametrized initial conditions, and an output functional:

$$\dot{z}(t) = f(t, z(t), x), \quad z(0) = \Phi(x)$$
$$y = h(z)$$

Here, z is the state vector, t is time, and x is a parameter vector, taking values in a known set $\mathcal{H}$, which parametrizes the differential equations model and/or the initial conditions. f and Φ are known functions. h is a known functional evaluated on the state trajectory, (e.g. the value of a state at a given time; the peak value of a given state; the peak location, t_{peak} of a certain state). In general, such equations do not possess a closed-form solution (i.e., y as a function of x, which we denoted $y = M(x)$). The essence of the solution mapping technique is twofold: over $\mathcal{H}$, determine which subset of parameters in x have measurable influence on $M(x)$, referred to as the *active parameters*, and again over $\mathcal{H}$, approximate M by a simple algebraic expression depending only on these parameters.

Usually, the active parameters are a small subset of the model parameters. Only these parameters need to be considered in the analysis since M has extremely low sensitivity to the remaining (nonactive) parameters. The active parameters are identified by ranking the absolute values of response sensitivities computed from a modest number of simulation runs. Parameters whose sensitivities are an order of magnitude less than the highest ranked parameter are candidates to be neglected. While this cut-off criterion is problem dependent and hence somewhat arbitrary in nature, past experience in methane combustion chemistry reaction modeling (Frenklach et al., 1992) indicates that taking the top-ranking 9 to 13 variables for each response is more than sufficient for accurate representation of an individual response.

The approximating functions for the responses are then obtained using the general methodology of the response surface technique (Box et al., 1978; Box and Draper, 1987). A relatively small number of computer simulations, referred to as *computer experiments*, are performed at combinations of pre-selected active parameter values and the entire set of these combinations is called a *design* of computer experiments. The computer experiments are performed using the mathematical model, $M(x)$. Finally, a simple function is selected from a prespecified class of functions (e.g., linear,

quadratic, exponential) to closely approximate M. The function obtained in this manner is referred to as a *surrogate* model, denoted $S(x)$. Once developed, the surrogate model replaces the solution of the original dynamic model whenever evaluation of the latter is required, decreasing the computational cost of evaluating the objective function by orders of magnitude.

While there is, in principle, no restriction on the mathematical form of the surrogate model, we will make extensive use of low order polynomial surrogate models, whose coefficients are determined via computer experiments arranged in a special order, called a factorial design (Box and Draper, 1987; Myers and Montgomery, 2002). These designs originate from a rigorous analysis of variance, with the objective of minimizing the number of computer experiments to be performed to gain the required information.

In the following, we assume that an upper bound on the surrogate modeling error is known: for each k, there exists e_k such that $\max_{x\in\mathcal{H}} |M_k(x) - S_k(x)| \leq e_k$. In practice, it is not possible to know such a bound with certainty. However, this bound can be approximated by sampling the parameter space and measuring the error between the mathematical model and surrogate model. Sampling and statistical assumptions can give some estimates of the error bounds.

The fitting error is controlled by the size of $\mathcal{H}$, the domain over which the surrogate model should approximate M, and the chosen polynomial order. The prior information $\mathcal{H}$ plays an important role at this stage. For a given polynomial degree, the fit of the response surface tends to improve as the size of the parameter domain (i.e. $\mathcal{H}$) decreases. Thus better prior information allows us to either use lower degree polynomials for surrogate models and/or reduce the fitting error. Both of these improvements ultimately lead to better predictions by the algorithms which follow.[1]

In Section 6, we use solution mapping to develop surrogate models for toy systems with a few parameters. However, the techniques can be applied to large systems with many parameters. In Frenklach et al. (2004), we developed surrogate models for a combustion process involving 102 active parameters.

3.2 Bounds for the validation problem

The surrogate models can be used to obtain bounds for the validation and prediction problems. Define two approximations of the feasible set using the surrogate models:

$$\mathcal{F}_I \doteq \{x \in \mathcal{H} \ : \ |S_k(x) - d_k| \leq u_k - e_k \ \forall k\}$$
$$\mathcal{F}_O \doteq \{x \in \mathcal{H} \ : \ |S_k(x) - d_k| \leq u_k + e_k \ \forall k\}$$

The subscripts emphasize inner and outer approximations, $\mathcal{F}_I \subseteq \mathcal{F} \subseteq \mathcal{F}_O$, which can be easily verified. Since $\mathcal{H}$ is described by c polynomial inequalities, both $\mathcal{F}_I$ and $\mathcal{F}_O$ are sets described by $2m + c$ polynomial inequalities.

[1] The Stone-Weierstrass Theorem (Folland, 1999) ensures that a continuous, multivariate function can be uniformly approximated to any degree of accuracy by a polynomial if $\mathcal{H}$ is closed and bounded. If prior information is not available, the response surface technique can still be applied but there are no guarantees that the model can be uniformly approximated by a polynomial over the entire parameter space.

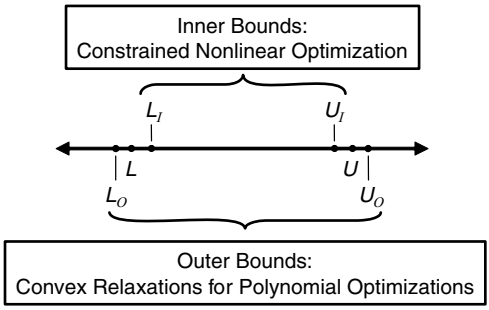


Fig. 1 A summary of the approach to the prediction problem: L and U are the min/max values of M_0 which are consistent with all available data. The proposed algorithm is to compute inner/outer bounds on L and U

The model validation problem is to find $\bar{x} \in \mathcal{F}$ or prove $\mathcal{F} = \emptyset$. In the next section, we describe a sufficient condition to prove that a set described by polynomial inequalities is empty. We use this condition to try to prove that $\mathcal{F}_O$ is empty. The containment $\mathcal{F} \subseteq \mathcal{F}_O$, ensures that $\mathcal{F} = \emptyset$ would follow from $\mathcal{F}_O = \emptyset$. Verifying the condition can be done with modest computational cost. In the model validation problem, if $\mathcal{F} = \emptyset$, there is an inconsistency among all dataset units. On the other hand, in order to not invalidate, a constrained nonlinear optimization routine can be used to search for $\bar{x}$ in $\mathcal{F}_I$. The containment $\mathcal{F}_I \subseteq \mathcal{F}$ ensures that $\bar{x} \in \mathcal{F}_I$ implies $\bar{x} \in \mathcal{F}$.

3.3 Bounds for the prediction problem

The prediction problem is to compute

$$L \doteq \min_{x \in \mathcal{F}} M_0(x), \qquad U \doteq \max_{x \in \mathcal{F}} M_0(x) \tag{2}$$

Figure 1 summarizes our approach. We compute four numbers (L_O, L_I, U_I, U_O) which bound (L, U): $L_O \leq L \leq L_I$ and $U_I \leq U \leq U_O$. The bounds with subscript I are referred to as inner bounds because they yield an inner approximation to the prediction interval. Similarly, the bounds with subscript O are referred to as outer bounds. If the inner and outer bounds are close, then we have approximately solved the original prediction problem. The numerical methods used to obtain the bounds are described in the remainder of the paper. Briefly, the bounds are obtained by using the surrogate models to convert the optimizations in Eq. (2) into polynomial optimizations. The inner bounds are computed by attempting to solve the resulting polynomial optimizations with constrained nonlinear optimization routines. The outer bounds are computed by solving related optimizations which are proven to bound the given polynomial optimizations.

The surrogate models, and their error bounds, along with the inner and outer bounds to $\mathcal{F}$ naturally lead to optimization expressions that bound L (and in an analogous manner, U). We have

$$\min_{x \in \mathcal{F}_O} (S_0(x) - e_0) \leq \underbrace{\min_{x \in \mathcal{F}} M_0(x)}_{L} \leq \min_{x \in \mathcal{F}_I} (S_0(x) + e_0) \tag{3}$$

which follows from $\mathcal{F}_I \subseteq \mathcal{F} \subseteq \mathcal{F}_O$ and the bound on the surrogate modeling error.

Recall that the function S_0 is polynomial, and the sets $\mathcal{F}_O$ and $\mathcal{F}_I$ are described by polynomial inequalities. Thus the two optimizations which bound L in Eq. (3) have a polynomial objective function and polynomial constraints. It is important to note that polynomial functions can have local minima which are not global minima. As a result, constrained nonlinear optimization routines may find a solution to the polynomial minimization which is a local but not global minima. Let L_I be "defined" as the result obtained using a nonlinear minimization routine on the function $S_0(x) + e_0$ constrained to $x \in \mathcal{F}_I$. Since L_I may only be a local minima, we can only conclude that $\min_{x\in\mathcal{F}_I}(S_0(x) + e_0) \leq L_I$. But this still guarantees that L_I is an upper bound on L.

The situation with the lower bound on L is more difficult. Suppose we "defined" α^* to be the result obtained using a nonlinear minimization routine on the function $S_0(x) - e_0$ constrained to $x \in \mathcal{F}_O$. We can only conclude that $\min_{x\in\mathcal{F}_O}(S_0(x) - e_0) \leq \alpha^*$ and hence there is no guarantee that this provides a lower bound on L. In the next section, we describe a convex relaxation that yields global lower bounds on polynomial minimization problems. A well-chosen convex relaxation yields a convex optimization problem whose solution is guaranteed to yield a lower bound on the original polynomial minimization. Convex optimization problems have the property that there are no local optima which are not global optima (i.e. the algorithm should converge to the global minimizer). The technique described in the next section is used to compute a number L_O which is guaranteed to satisfy $L_O \leq \min_{x\in\mathcal{F}_O}(S_0(x) - e_0)$. More importantly, this guarantees that L_O is a lower bound on L.

4 Convex relaxations for polynomial optimizations

In this section we discuss computational methods for polynomial optimizations. Let $\{p_k\}_{k=0}^r$ be a collection of polynomials. In what follows, p_0 is an objective function, and $p_1, \ldots, p_r$ describe constraints. Define $\mathcal{P} \doteq \{x \in \mathbb{R}^n \ : \ p_k(x) \leq 0, \ k = 1, \ldots, r\}$. Throughout this section, we focus on the following problems:

$$\text{Feasibility: Attempt to prove } \mathcal{P} = \emptyset. \tag{4}$$

$$\text{Optimization: Find } \underline{\mathrm{L}} \text{ such that } \min_{x\in\mathcal{P}} \ p_0(x) \geq \underline{\mathrm{L}}. \tag{5}$$

The approach presented in this section relies heavily on connections between semidefinite programming and polynomial optimizations (Parrilo, 2000, 2003; Lasserre, 2001). After presenting some background material, we restrict $\{p_k\}_{k=0}^r$ to be quadratic functions and discuss a well-known technique for the feasibility and optimization problems. Then we generalize this technique to handle polynomials of any degree.

4.1 Background

4.1.1 Polynomial notation

$\mathbb{N}$ denotes the set of nonnegative integers, $\{0, 1, \ldots\}$, and $\mathbb{N}^n$ is the set of n-dimensional vectors with entries in $\mathbb{N}$. For $\alpha \in \mathbb{N}^n$, a monomial in variables $\{x_1, \ldots, x_n\}$ is given

by $x^\alpha \doteq x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$. The degree of a monomial is defined as $\deg x^\alpha \doteq \sum_{i=1}^n \alpha_i$. A polynomial in variables $\{x_1, \ldots, x_n\}$ is a finite linear combination of monomials:

$$p \doteq \sum_{\alpha \in \mathcal{A}} c_\alpha x^\alpha = \sum_{\alpha \in \mathcal{A}} c_\alpha x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n} \tag{6}$$

where $c_\alpha \in \mathbb{R}$ and $\mathcal{A}$ is a finite collection of vectors in $\mathbb{N}^n$. Using the definition of deg for a monomial, the degree of p is defined as $\deg p \doteq \max_{\alpha \in \mathcal{A}} [\deg x^\alpha]$. The set of all polynomials in variables $\{x_1, \ldots, x_n\}$ with real coefficients is denoted $\mathbb{R}[x_1, \ldots, x_n]$. Unless specified otherwise, a polynomial is assumed to be in $\mathbb{R}[x_1, \ldots, x_n]$. A distinction is typically made between the polynomial as an object in $\mathbb{R}[x_1, \ldots, x_n]$ and the associated function from $\mathbb{R}^n$ to $\mathbb{R}$. The only relevant point to this distinction is that if p is a polynomial in $\mathbb{R}[x_1, \ldots, x_n]$, then $p(x_0)$ will denote the polynomial function evaluated at a specific point $x_0 \in \mathbb{R}^n$.

4.1.2 Semidefinite programming

This brief review of semidefinite programming (SDP) is based on a survey by Vandenberghe and Boyd (1996) and a monograph by Boyd et al. (1994). A symmetric matrix $F \in \mathbb{R}^{n \times n}$ is positive semidefinite if $x^T F x \geq 0$ for all $x \in \mathbb{R}^n$. Positive semidefinite matrices are denoted by $F \succeq 0$. A semidefinite program is an optimization problem of the following form:

$$\begin{aligned} \min_{\lambda} \quad & c^T \lambda \\ \text{subject to :} \quad & F_0 + \sum_{k=1}^r \lambda_k F_k \succeq 0 \end{aligned} \tag{7}$$

The symmetric matrices $F_0, \ldots, F_r \in \mathbb{R}^{n \times n}$ and the vector $c \in \mathbb{R}^r$ are given data. The vector $\lambda \in \mathbb{R}^r$ is the decision variable and the constraint, $F_0 + \sum_{k=1}^r \lambda_k F_k \succeq 0$, is called a linear matrix inequality (LMI). We refer to Eq. (7) as the primal problem. The dual associated with this primal problem is:

$$\begin{aligned} \max_{Z} \quad & -\mathbf{Tr}\,[F_0 Z] \\ \text{subject to :} \quad & \mathbf{Tr}\,[F_k Z] = c_k \quad k = 1, \ldots, r \\ & Z \succeq 0 \end{aligned} \tag{8}$$

where $Z = Z^T \in \mathbb{R}^{n \times n}$ is the decision variable for the dual problem. $\mathbf{Tr}\,[\cdot]$ denotes the trace of a matrix. This dual problem can be recast in the form of Eq. (7) and thus it is also a semidefinite program. While the primal and dual forms may look restrictive, these formulations are quite versatile and SDPs find applications in many problems of interest. Moreover, SDPs are convex and quality software exists to solve these problems. In particular, SeDuMi (Sturm, 2001) is a freely available MATLAB toolbox that simultaneously solves both the primal and dual forms of a semidefinite program. The asymptotic computational complexity to solve a semidefinite program with SeDuMi is $O(v^2 d^{2.5} + d^{3.5})$ where v is the number of variables and d is the dimension of the LMI (Labit et al., 2002).

In some cases, our only goal is to find a decision variable that satisfies the constraints. These are semidefinite programming feasibility problems. The following is

an example:

$$\text{Find } \lambda_1, \ldots, \lambda_r \in \mathbb{R} \text{ such that } F_0 + \sum_{k=1}^{r} \lambda_k F_k \succeq 0 \tag{9}$$

4.2 Relaxations for quadratic optimizations

If all $\{p_k\}_{k=0}^r$ are affine, then the feasibility and optimization problems (Eqs. (4) and (5)) are linear programs. A natural, nontrivial, and useful extension is to consider quadratic surrogate models. Consider the case where $\{p_k\}_{k=0}^r$ are all quadratic functions. For any quadratic function p in n real variables, there is a unique symmetric matrix $F \in \mathbb{R}^{(1+n)\times(1+n)}$ such that $p(x) = \left[\begin{smallmatrix}1\\x\end{smallmatrix}\right]^T F \left[\begin{smallmatrix}1\\x\end{smallmatrix}\right]$ for all x. Let $\mathbb{R}_+$ denote the set of nonnegative real numbers. The following theorem, a simple consequence of the *S-procedure* (Boyd et al., 1994), gives deceptively simple conditions that can be applied to the quadratic feasibility and optimization problems:

Theorem 1. *Assume* $p_k(x) \doteq \left[\begin{smallmatrix}1\\x\end{smallmatrix}\right]^T F_k \left[\begin{smallmatrix}1\\x\end{smallmatrix}\right]$ *where* $F_k^T = F_k \in \mathbb{R}^{(1+n)\times(1+n)} (k = 0, \ldots, r)$. *Define* $\mathcal{P} \doteq \{x \in \mathbb{R}^n \ : \ p_k(x) \le 0, \ k = 1, \ldots, r\}$.

(A) If there exists $\{\lambda_k\}_{k=1}^r \in \mathbb{R}_+$ *such that* $-I + \sum_{k=1}^r \lambda_k F_k \succeq 0$ *then* $\mathcal{P} = \emptyset$.
(B) If there exists $\{\lambda_k\}_{k=1}^r \in \mathbb{R}_+$ *such that* $F_0 - \left[\begin{smallmatrix}\gamma & 0\\ 0 & 0_n\end{smallmatrix}\right] + \sum_{k=1}^r \lambda_k F_k \succeq 0$ *then* $\min_{x\in\mathcal{P}} \ p_0(x) \ge \gamma$.

Proof: (A): If there exists $\bar{x} \in \mathcal{P}$, then we obtain an immediate contradiction:

$$0 \overset{(a)}{\le} \left[\begin{smallmatrix}1\\\bar{x}\end{smallmatrix}\right]^T \left[-I + \sum_{k=1}^{r} \lambda_k F_k \right] \left[\begin{smallmatrix}1\\\bar{x}\end{smallmatrix}\right] = -1 - \bar{x}^T \bar{x} + \sum_{k=1}^{r} \lambda_k \left[\begin{smallmatrix}1\\\bar{x}\end{smallmatrix}\right]^T F_k \left[\begin{smallmatrix}1\\\bar{x}\end{smallmatrix}\right] \overset{(b)}{<} 0$$

Inequality (a) follows because $-I + \sum_{k=1}^r \lambda_k F_k \succeq 0$. Inequality (b) follows because $\bar{x} \in \mathcal{P}$ and each $\lambda_k \ge 0$.

(B): For any $\bar{x} \in \mathcal{P}$,

$$0 \le \left[\begin{smallmatrix}1\\\bar{x}\end{smallmatrix}\right]^T \left[F_0 - \left[\begin{smallmatrix}\gamma & 0\\ 0 & 0_n\end{smallmatrix}\right] + \sum_{k=1}^{r} \lambda_k F_k \right] \left[\begin{smallmatrix}1\\\bar{x}\end{smallmatrix}\right] \le \left[\begin{smallmatrix}1\\\bar{x}\end{smallmatrix}\right]^T F_0 \left[\begin{smallmatrix}1\\\bar{x}\end{smallmatrix}\right] - \gamma \tag{10}$$

Since Eq. (10) holds for all $\bar{x} \in \mathcal{P}$, we conclude that $\min_{x\in\mathcal{P}} p_0(x) \ge \gamma$. □

Conditions (A) and (B) in Theorem 1 are applications of the *S-procedure* (Boyd et al., 1994). The use of the S-procedure in control theory dates back to Lur'e and Postnikov in the 1940's (see (Boyd et al., 1994) for a brief historical account). The nonnegative scalars, $\{\lambda_k\}_{k=1}^r$, are often referred to as *multipliers*. The two conditions in Theorem 1 can be implemented and efficiently solved as SDPs. In (A), the search

for $\lambda_k \geq 0$ such that $-I + \sum_{k=1}^{r} \lambda_k F_k \succeq 0$ can be placed in the form of an SDP feasibility problem (Eq. (9)). [2] Similarly, condition (B) yields a lower bound on quadratic optimizations and we can find the "best" lower bound by solving an SDP:

$$\begin{aligned} &\min_{x \in \mathcal{P}} \; p_0(x) \geq \max_{\lambda_k \in \mathbb{R}_+, \gamma} \; \gamma \\ &\text{subject to: } F_0 - \gamma \left[\begin{smallmatrix} 1 & 0 \\ 0 & 0_n \end{smallmatrix}\right] + \sum_{k=1}^{r} \lambda_k F_k \succeq 0 \end{aligned} \tag{11}$$

The lower bound has a useful statistical interpretation (Vandenberghe and Boyd, 1996; Frazzoli et al., 2001). This interpretation is based on the dual of lower bound SDP in Eq. (11):

$$\begin{aligned} &\min_{\Sigma, \bar{x}} \mathbf{Tr}\left[F_0 \left[\begin{smallmatrix} 1 & \bar{x}^T \\ \bar{x} & \Sigma \end{smallmatrix}\right]\right] \\ &\text{subject to: } \begin{cases} \mathbf{Tr}\left[F_k \left[\begin{smallmatrix} 1 & \bar{x}^T \\ \bar{x} & \Sigma \end{smallmatrix}\right]\right] \leq 0 \quad k = 1, \ldots, r \\ \left[\begin{smallmatrix} 1 & \bar{x}^T \\ \bar{x} & \Sigma \end{smallmatrix}\right] \succeq 0. \end{cases} \end{aligned} \tag{12}$$

Under weak technical conditions, strong duality holds and the SDPs in Eqs. (11) and (12) yield the same lower bound. We can reinterpret the dual SDP by considering a random variable X of dimension $n \times 1$ with $E[X] = \bar{x}$ and $E\left[XX^T\right] = \Sigma$. Since $E[\left[\begin{smallmatrix} 1 \\ X \end{smallmatrix}\right]^T F_k \left[\begin{smallmatrix} 1 \\ X \end{smallmatrix}\right]] = \mathbf{Tr}[F_k \left[\begin{smallmatrix} 1 & \bar{x}^T \\ \bar{x} & \Sigma \end{smallmatrix}\right]]$, Eq. (12) is equivalent to:

$$\begin{aligned} &\min_{X} E\left[\left[\begin{smallmatrix} 1 \\ X \end{smallmatrix}\right]^T F_0 \left[\begin{smallmatrix} 1 \\ X \end{smallmatrix}\right]\right] \\ &\text{s.t. : } \begin{cases} E\left[\left[\begin{smallmatrix} 1 \\ X \end{smallmatrix}\right]^T F_k \left[\begin{smallmatrix} 1 \\ X \end{smallmatrix}\right]\right] \leq 0 \;\; k = 1, \ldots, r \\ X \text{ is a rand. var. with } E[X] = \bar{x}, \;\; E\left[XX^T\right] = \Sigma \end{cases} \end{aligned} \tag{13}$$

The constraint $\left[\begin{smallmatrix} 1 & \bar{x}^T \\ \bar{x} & \Sigma \end{smallmatrix}\right] \succeq 0$ can be dropped because it is satisfied by all random variables. The optimization in Eq. (13) is similar to $\min_{x \in \mathcal{P}} \; p_0(x)$ except that the optimization variable is a random variable, X, rather than a deterministic vector, x. Moreover, the constraints in Eq. (13) are only required to be satisfied on average. Let $(\bar{x}_0, \Sigma_0)$ denote the optimal values found in the dual SDP. In our experience, vectors drawn from a Gaussian distribution with mean $\bar{x}_0$ and second moment Σ_0 can provide good initial conditions to start the nonlinear optimization for computing an upper bound.

4.3 Relaxations for polynomial optimizations

In this section, we return to the general case where $\{p_k\}_{k=0}^{r}$ is a collection of polynomials which are not necessarily quadratic. One additional concept is needed to generalize the *S*-procedure. A polynomial p is a *sum of squares* (SOS) if there exist polynomials

[2] There is one detail to place this problem in the form of Eq. (9). A single LMI constraint is obtained by diagonally augmenting the additional constraints, $\lambda_k \geq 0$, to the main semidefiniteness constraint, $-I + \sum_{k=1}^{r} \lambda_k F_k \succeq 0$ (Vandenberghe and Boyd, 1996).

$\{f_i\}_{i=1}^m$ such that $p = \sum_{i=1}^m f_i^2$. We note that if p is a sum of squares then $p(x) \geq 0$ $\forall x \in \mathbb{R}^n$. In general, the existence of a sum of squares decomposition is sufficient but not necessary for global nonnegativity. This is related to a famous problem in mathematics, first posed by Hilbert in 1900 (Reznick, 2000). Let $\mathcal{S}$ denote the set of SOS polynomials, i.e. write $p \in \mathcal{S}$ if p is a sum of squares. The restriction of $\mathcal{S}$ to polynomials of degree $\leq d$ is denoted $\mathcal{S}_d$. The following theorem generalizes the the results in Theorem 1:

Theorem 2. *Assume* $\{p_k\}_{k=0}^r$ *are polynomials and define* $\mathcal{P} \doteq \{x \in \mathbb{R}^n \ : \ p_k(x) \leq 0, \ k = 1, \ldots, r\}$.

(A) If there exists $\{\lambda_k\}_{k=1}^r \in \mathcal{S}$ *such that* $-1 + \sum_{k=1}^r \lambda_k p_k \in \mathcal{S}$ *then* $\mathcal{P} = \emptyset$.
(B) If there exists $\{\lambda_k\}_{k=1}^r \in \mathcal{S}$ *such that* $p_0 - \gamma + \sum_{k=1}^r \lambda_k p_k \in \mathcal{S}$ *then* $\min_{x \in \mathcal{P}} \ p_0(x) \geq \gamma$.

The proof of this theorem essentially involves notational changes to the proof of Theorem 1 and hence it is omitted. The *multipliers*, $\{\lambda_k\}_{k=1}^r$, are now polynomials which are constrained to be sums of squares. In principle, conditions (A) and (B) in Theorem 2 can be directly applied to the feasibility and optimization problems (Eqs. (4) and (5)). For example, if we restrict the multipliers to have degree $\leq d$, then the analogue of Eq. (11) is:

$$\begin{aligned} &\min_{x \in \mathcal{P}} \ p_0(x) \geq \max_{\lambda_k \in \mathcal{S}_d, \gamma} \ \gamma \\ &\text{subject to: } p_0 - \gamma + \sum_{k=1}^r \lambda_k p_k \in \mathcal{S} \end{aligned} \tag{14}$$

The maximization on the right involves the search for polynomial multipliers subject to various SOS constraints. This type of problem is called an SOS optimization (Prajna et al., 2002). The decision variables in the SOS optimization are γ and the coefficients of the multipliers. Since a polynomial with n variables and degree d has $\binom{n+d}{d}$ terms, there are $r\binom{n+d}{d} + 1$ decision variables. The only issue at this point is to write the SOS constraints in a meaningful form. This is done by exploiting ties between SOS polynomials and semidefinite matrices. In the remainder of this section, we review the elementary aspects of recent work by Parrilo (2000, 2003) that can be used to implement conditions (A) and (B) as SDPs. Connections to algebraic geometry are omitted in this review. For instance, if some additional technical conditions are satisfied and we search for multipliers of arbitrary degree, then the $\geq$ in Eq. (14) can be strengthened to $=$ by applying Putinar's theorem (Lemma 4.1 in Putinar (1993)).

Theorem 3 below gives a concrete statement of the ties between sums of squares and positive semidefinite matrices. We require two facts that follow from Theorem 1 and its preceding Lemma in Reznick (1978):

1. If p is a sum of squares then p must have even degree.
2. If p is degree $2d$ ($d \in \mathbb{N}$) and $p = \sum_{i=1}^m f_i^2$ then $\deg f_i \leq d \ \forall i$.

Next, we define z as the column vector of all monomials in variables $\{x_1, \ldots, x_n\}$ of degree $\leq d$: [3]

$$z \doteq \left[1,\ x_1,\ x_2,\ \ldots,\ x_n,\ x_1^2,\ x_1x_2,\ \ldots,\ x_n^2,\ \ldots,\ x_n^d\right]^T \tag{15}$$

There are $\binom{k+n-1}{k}$ monomials in n variables of degree k. Thus z is a column vector of length $l_z \doteq \sum_{k=0}^{d}\binom{k+n-1}{k} = \binom{n+d}{d}$. If f is a polynomial in n variables with degree $\leq d$, then f is a finite linear combination of monomials of degree $\leq d$. Consequently, there exists $a \in \mathbb{R}^{l_z}$ such that $f = a^T z$. The proof of the following theorem, introduced as a "Gram Matrix" method by Choi et al. (1995), is enlightening and is included for completeness. This result can be found more recently in Powers and Wörmann (1998).

Theorem 3. *Suppose $p \in \mathbb{R}[x_1, \ldots, x_n]$ is a polynomial of degree $2d$ and z is the $l_z \times 1$ vector of monomials defined above. Then p is a SOS if and only if there exists a symmetric matrix $Q \in \mathbb{R}^{l_z \times l_z}$ such that $Q \succeq 0$ and $p = z^T Qz$.*

Proof:

($\Rightarrow$) If p is a SOS, then there exists polynomials $\{f_i\}_{i=1}^m$ such that $p = \sum_{i=1}^m f_i^2$. As noted above, deg $f_i \leq d$ for all i. Thus, for each f_i there exists a vector, $a_i \in \mathbb{R}^{l_z}$, such that $f_i = a_i^T z$. Define the matrix, $A \in \mathbb{R}^{l_z \times m}$, whose ith column is a_i and define $Q \doteq AA^T \succeq 0$. Then $p = z^T Qz$.

($\Leftarrow$) Assume there exists $Q = Q^T \in \mathbb{R}^{l_z \times l_z}$ such that $Q \succeq 0$ and $p = z^T Qz$. Define $m \doteq rank(Q)$. There exists a matrix $A \in \mathbb{R}^{l_z \times m}$ such that $Q = AA^T$. Let a_i denote the ith column of A and define the polynomials $f_i \doteq z^T a_i$. By definition of f_i, $p = z^T(AA^T)z = \sum_{i=1}^m f_i^2$.

□

Theorem 3 provides the bridge to convert an SOS optimization, such as the maximization in Eq. (14), into an SDP. For example, the constraint $p_0 - \gamma + \sum_{k=1}^r \lambda_k p_k \in \mathcal{S}$ can be equivalently written as:

$$p_0 - \gamma + \sum_{k=1}^{r} \lambda_k p_k = z^T Qz \tag{16}$$

$$Q \succeq 0 \tag{17}$$

Q is a new matrix of decision variables that is introduced when we convert an SOS constraint to an LMI constraint. Equating the coefficients of $z^T Qz$ and $p_0 - \gamma + \sum_{k=1}^r \lambda_k p_k$ imposes linear equality constraints on the decision variables. Thus, Eq. (16)

[3] Any ordering of the monomials can be used to form z. In Eq. (15), x^α precedes x^β in the definition of z if:

$$\deg x^\alpha < \deg x^\beta \quad \text{or} \quad \deg x^\alpha = \deg x^\beta \text{ and the first nonzero entry of } \alpha - \beta \text{ is } > 0.$$

can be rewritten as a set of linear equality constraints on the decision variables. All SOS constraints in Eq. (14) can be replaced in this fashion with linear equality constraints and LMI constraints. As a result, the maximization in Eq. (14) can be written in the SDP dual form (Eq. (8)). While this may appear cumbersome, it is elementary, and software can perform the conversion. For example, SOSTOOLS (Prajna et al., 2002) is a freely available MATLAB toolbox for solving SOS optimizations. Currently, this package allows the user to specify the polynomial constraints using a symbolic toolbox. SOSTOOLS then converts the SOS optimization into an SDP which is then solved with SeDuMi (Sturm, 2001). SOSTOOLS then converts the solution of the SDP back to a polynomial solution. A drawback is that the size of the resulting SDP grows rapidly if the SOS optimization involves polynomials with many variables and/or high degree. While various techniques can be used to exploit the problem structure (Gatermann and Parrilo, 2004) this computational growth is a generic trend in SOS optimizations.

5 Computational aspects of the validation/prediction problems

In this section, we apply the results of the previous sections to the validation and prediction problems. First consider model validation: find $\bar{x} \in \mathcal{F}$ or prove $\mathcal{F}$ is empty. The inner and outer approximations, $\mathcal{F}_I$ and $\mathcal{F}_O$, are described by $2m + c$ polynomial inequalities and satisfy $\mathcal{F}_I \subseteq \mathcal{F} \subseteq \mathcal{F}_O$. We use a constrained nonlinear optimization in an attempt to find $\bar{x}$ in $\mathcal{F}_I$. Alternatively, we apply Theorem 2 in an attempt to prove that $\mathcal{F}_O$ is empty. Let $\{p_k\}_{k=1}^{2m+c}$ denote the polynomials that describe the outer approximation: $\mathcal{F}_O \doteq \{x \in \mathbb{R}^n \ : \ p_k(x) \leq 0, \ k = 1, \ldots, 2m + c\}$. By Theorem 2, if there exists sum of squares polynomials, $\{\lambda_k\}_{k=0}^{2m+c}$, such that $-1 + \sum_{k=1}^{2m+c} \lambda_k p_k \in \mathcal{S}$ then $\mathcal{F}_O$ is empty. If we restrict the multipliers to have degree $\leq d$, then the search for multipliers that prove $\mathcal{P} = \emptyset$ can be performed as an SDP feasibility problem. In the special case where $\mathcal{F}_O$ is described by quadratic inequalities, and we restrict the multipliers to be nonnegative constants (degree=0), then the conversion to an SDP feasibility problem is immediate by applying the S-procedure (Theorem 1).

There are three scenarios in which the convex relaxation fails to prove $\mathcal{F}_O = \emptyset$ and the nonlinear optimization fails to find $\bar{x} \in \mathcal{F}_I$, hence no conclusions regarding model validation can be drawn. In each case, there are partial remedies to resolve the difficulty at the expense of additional computation. One possibility is that $\mathcal{F}_O$ is not empty, though $\mathcal{F}$ is. In this situation, the approximation of $\mathcal{F}$ is the culprit, due to errors in fitting the surrogate models to the mathematical models. The fitting errors can be reduced by increasing the polynomial degree of the surrogate models and refitting the mathematical models. A second possibility is that $\mathcal{F}_I$ is nonempty, but the constrained nonlinear optimization routine may fail to find $\bar{x} \in \mathcal{F}_I$. The only remedy is to restart the feasibility search at a new initial point(s). Finally, $\mathcal{F}_O$ may be empty, but the convex relaxation is unable to prove emptiness. A partial remedy is to increase the degree of the multipliers and re-solve the SDP. The condition in Theorem 2 is only sufficient and hence we may fail to find multipliers even if we allow the degree to be arbitrarily large. However, Theorem 2 can be generalized to yield a necessary and sufficient condition. This result, known as the Positivstellensatz, is due to Stengle (1974) and a recent presentation can be found in Section 4 of Parrilo (2003). The

generalization of Theorem 2 to yield necessary and sufficient conditions is certainly of theoretical interest. However, the utility for moderate to large sized problems (10's of parameters and constraints) is currently limited by computation.

Next, we consider the prediction problem: compute $R = [L, U]$ as defined in (1). As in Section 3, we focus on L. The inner bound, $L \leq L_I$, is found using nonlinear optimization. The outer bound, L_O^d ($\leq L$) is obtained applying Theorem 2,

$$\begin{aligned} L_O^d \doteq & \max_{\lambda_k \in \mathcal{S}_d, \gamma} \ \gamma \\ & \text{subject to: } p_0 - \gamma + \sum_{k=1}^{2m+c} \lambda_k p_k \in \mathcal{S} \end{aligned} \tag{18}$$

The superscript, d, denotes the restriction on the degree of the multipliers. Since the multipliers, λ_k, are sums of squares, d should be an even integer. We now have inner and outer bounds that satisfy $L_O^d \leq L \leq L_I$. If L_I and L_O^d are "close", then we have approximately solved the original minimization for L. As in the validation problem, there are three reasons these bounds may fail to be close. Remedies are analogous to those discussed for the validation problem. For example, we can attempt to improve both bounds by refitting the mathematical models. Alternatively, we can try to reduce the inner bound by restarting the optimization from a different initial condition. Finally, we can try to improve the outer bound by increasing the degree of the multipliers from d to $d + 2$. Since $L_O^d \leq L_O^{d+2} \leq L$, we have a hierarchy of relaxations to obtain outer bounds. At the lowest ($d = 0$) the multipliers are just positive constants and this corresponds to a standard Lagrangian relaxation.[4]

Finally, we comment on the optimization tools which can be employed for large scale problems. We have used two nonlinear optimization routines to solve for the inner bounds on the combustion problem (Frenklach et al., 2002, 2004): `fmincon` which is the routine included in MATLAB's optimization toolbox and the commercial software NPSOL running within MATLAB. Both solvers perform reasonably but NPSOL tends to produce "better" inner bounds with less computation time. The outer bounds, as noted above, reduce to an SOS optimization problem. The solution of SOS optimization is an area of active research and the currently available tools can only solve problems with a few parameters and constraints. Small problems can be solved with SOSTOOLS (Prajna et al., 2002) in conjunction with the freely available SDP solver SeDuMi (Sturm, 2001). For the special case where the polynomials are quadratic, the S-procedure (Theorem 1) can be applied to obtain an SDP. SeDuMi is more efficient at solving this SDP than the solver contained in MATLAB's LMILAB toolbox because it exploits sparsity in the matrices. We have used SeDuMi to solve for outer bounds on the combustion problem (Frenklach et al., 2002, 2004).

[4] If the technical conditions in Putinar's theorem (Putinar, 1993) are satisfied, then these outer bounds converge to the correct answer, $L_O^d \to \min_{x \in \mathcal{F}_O}(S_0(x) - e_0)$, as $d \to \infty$. But since there is still a gap between L and $\min_{x \in \mathcal{F}_O}(S_0(x) - e_0)$, this convergence property is not crucial for the prediction problem.

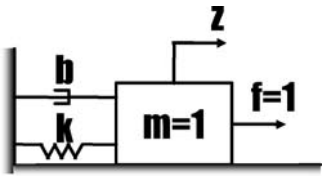


Fig. 2 A mass-spring-damper system

6 Examples

We recently demonstrated the viability of the proposed algorithm on a combustion process involving 77 experiments and 102 active parameters (Frenklach et al., 2002, Submitted). In this section, we illustrate the steps of the proposed algorithm through two examples. Due to their simplicity, we can describe and visualize certain aspects of the proposed algorithm.

6.1 Mass-spring-damper

In this example, we consider a mass-spring-damper system with a unit of force applied to the mass (Fig. 2). The goal is to use a single dataset unit containing an uncertain measurement and model of the integrated error (IE) to predict the peak velocity (PV). Using the techniques discussed in this paper we will calculate a prediction interval that contains PV. The constraints imposed by the dataset unit allow us to form a smaller prediction interval than would be otherwise possible.

The damper applies a force proportional to the velocity of the mass while the spring applies a force proportional to the position. Thus, Newton's second law for the mass-spring-damper is:

$$m\ddot{z} = -b\dot{z} - kz + f \tag{19}$$

The differential equation model is parameterized by the mass m, the spring constant k, and the damping coefficient b. We assume the $m = 1$, but k and b are unknown. Therefore the parameter vector $x = (b, k)$ reflects the uncertainty in the differential equation model. The prior information on this uncertainty is $\mathcal{H} = \{(b, k) : 0.5 \leq b \leq 1,\ 1 \leq k \leq 2\}$. Information for the experimental and predicted processes is given in Tables 1 and 2. We generated d_{IE} by computing the integrated error at $x = (0.75, 1.5)$ and then adding an error term randomly selected from $[-0.2,\ 0.2]$. The feasible set

Table 1 Definitions for MSD experiment process (P_{IE})

Quantity of Interest, IE	Integrated Error
Math. Model, M_{IE}	$\ddot{z}(t) + b\dot{z}(t) + kz(t) = 1$ $z(0) = 0, \dot{z}(0) = 0$ $IE = \lim_{T\to\infty} \int_0^T \lvert z(s) - z(T)\rvert ds$
Surrogate Model, S_{IE}	$S_{IE}([^b_k]) = 8.57 - 6.78b - 4.42k$ $+2.56b^2 + 0.91bk + 0.86k^2$
Surr. Model Error, e_{IE}	0.082
Meas./Unc., (d_{IE}, u_{IE})	(1.37, 0.20)

Table 2 Definitions for MSD predicted process (P_{PV})

Quantity of Interest, PV	Peak Velocity
Math. Model, M_{PV}	$\ddot{z}(t) + b\dot{z}(t) + kz(t) = 1$ $z(0) = 0, \dot{z}(0) = 0$ $PV = \max_{t \geq 0} \lvert\dot{z}(t)\rvert$
Surrogate Model, S_{PV}	$S_{PV}([^b_k]) = 1.25 - 0.63b - 0.37k$ $+0.12b^2 + 0.12bk + 0.05k^2$
Surr. Model Error, e_{PV}	0.004
Meas./Unc., (d_{PV}, u_{PV})	———

for this measurement is $\mathcal{F} = \{x \in \mathcal{H} \ : \ |M_{IE}(x) - d_{IE}| \leq u_{IE}\}$. Hereafter, the *true* system refers to the parameterized model evaluated at $x_{true} = (0.75, 1.5)$. We note that in a real problem, (d_{IE}, u_{IE}) would be obtained from an experiment.

We briefly discuss the computation of the surrogate models. We used a 2^2 composite orthogonal design containing 9 points in the parameter space, $\{x_i\}_{i=1}^9 \subset \mathcal{H}$. These points consisted of the 4 corners of $\mathcal{H}$, the midpoints of the 4 sides of $\mathcal{H}$, and the center of $\mathcal{H}$. At each point in the parameter space, we calculated peak velocity by evaluating M_{PV}. Specifically, we simulated the mass-spring damper system with the chosen values for the spring/damping constants and then computed the maximum velocity obtained during the simulation. We restricted the surrogate model, S_{PV}, to be a quadratic function and used an unweighted least-squares criterion to determine the coefficients. The same procedure was used to generate the surrogate model, S_{IE}, for IE. To compute the surrogate modeling errors we sampled the parameter space at an additional 100 points chosen randomly from $\mathcal{H}$. The maximum deviation between the mathematical models and the corresponding surrogate models is e_{PV} and e_{IE}, respectively.

Computing the prediction interval for PV involves specifying four quantities: L_O, L_I, U_I, and U_O. We note that the parameter constraints can be written as quadratic inequalities, e.g. $(b - 0.75)^2 \leq 0.125$. Since the surrogate models are also quadratic the outer bounds were computed by applying Theorem 1(B) and using SeDuMi to solve the resulting SDP. `fmincon`, an optimization routine in MATLAB, was used to find the inner bounds.

These computations led to the following bounds: $0.478 \leq L \leq 0.504$ and $0.583 \leq U \leq 0.602$. The inner bounds guarantee that there are parameter values, consistent with all available information which would generate a peak velocity is as low as 0.504 and as great as 0.583. The outer bounds guarantee that the true peak velocity can be no less than 0.478 and no greater than 0.602. For comparison, Fig. 3 shows the response of the true system. The solid lines on this figure are the outer bounds found by applying Theorem 1. As expected, the peak velocity of the true system lies between the outer bounds.

For this simple example, we can also visualize the various sets and models involved in the prediction algorithm (Fig. 4). Figure 4(a) shows that the dataset unit carves a diagonal swath across the parameter space. $\mathcal{F}_I$ and $\mathcal{F}_O$ provide good approximations of $\mathcal{F}$ because the surrogate modeling error, e_{IE}, is small. The labeled points lie within $\mathcal{F}_I$ and satisfy $M_{PV}(x_L) = L_I$, $M_{PV}(x_U) = U_I$. Figure 4(b) shows the surface generated by $PV = M_{PV}(x)\ \forall x \in \mathcal{H}$. The tick marks on the z-axis can be used to compare the prediction bounds (L_O, L_I, U_I, U_O) with the min/max values of PV on $\mathcal{H}$. If we only

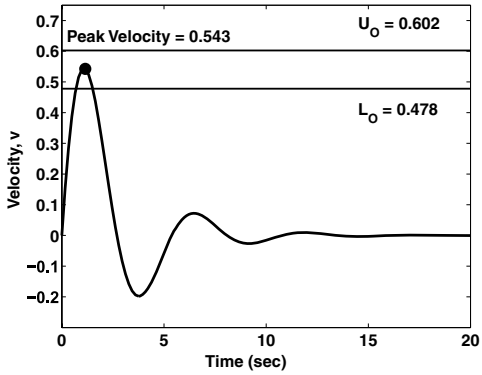
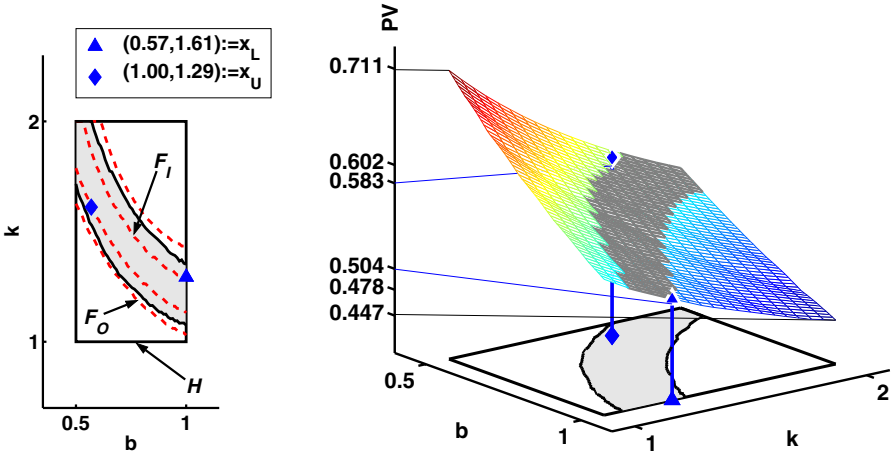


Fig. 3 Velocity of true system due to a unit step in applied force. The true peak velocity lies between the outer bounds



(a) Parameter Sets: $\mathcal{F}_I/\mathcal{F}_O$ approximate $\mathcal{F}$ and are dashed in red.

(b) PV Surface: The surface shows $M_{PV}(b, k)$. Lines emanate from surface at heights of L_I, U_I, and the min/max values of PV. Tick marks for L_O and U_O are also shown.

Fig. 4 Visualization of sets and models: In both subplots, the actual feasible set, $\mathcal{F}$, is shaded gray and the parameter values achieving the inner bounds (L_I/U_I) are blue symbols

know that $x \in \mathcal{H}$, then it is possible for the true peak velocity to lie between 0.447 and 0.711. Conceptually, the proposed algorithm uses the dataset unit to constrain the possible parameter vectors. As a result, the true peak velocity must lie in the gray shaded region on the surface. The minimum (maximum) value of PV on the gray region of the surface lies between L_O and L_I (U_I and U_O).

6.2 Cell division

We consider the cell division control system in frog eggs (Novak and Tyson, 1993; Tyson et al., 2001). Many biochemical processes occur as a cell grows and divides into two daughter cells. Two molecules, M-phase promoting factor (MPF) and cyclin, play a key role during cell division. In this example, we pose a model for the interactions of

Table 3 Definitions for cell division: Dataset units ($i = 1, 2$) generated at $k_{w,1} = 3.5$, $k_{w,2} = 6.0$

Quantity of Interest, y_i	MPF after 60 minutes
	$\dot{z}_1 = \frac{0.01}{1.1} - \left(\frac{\bar{k}_2}{10} + k_{w,i}\right) z_1 + \left(0.04 + 100 z_2^2\right)\left(\frac{z_2}{1.1} - z_1\right)$
Math. Model, M_i	$\dot{z}_2 = 0.01 - \frac{\bar{k}_2}{10} z_2$
$(k_{w,1} = 3.5, k_{w,2} = 6.0)$	$z_1(0) = 0.01, \;\; z_2(0) = 0.75$
	$y_i = z_1(60)$
Surrogate Model, S_i	$S_1(\bar{k}_2) = -1.509\bar{k}_2^3 + 1.340\bar{k}_2^2 - 0.40\bar{k}_2 + 0.05$
	$S_2(\bar{k}_2) = -0.097\bar{k}_2^3 + 0.11\bar{k}_2^2 - 0.048\bar{k}_2 + 0.01$
Surr. Model Error, e_i	$e_1 = 8.98e{-}4, \;\; e_2 = 1.91e{-}5$
Meas./Unc., (d_i, u_i)	$(d_1, u_1) = (9.8e{-}3, 5e{-}4), \;\; (d_2, u_2) = (6.5e{-}3, 5e{-}4)$

MPF and cyclin. Two measurements of MPF are then used to invalidate this model. This example demonstrates the model validation techniques in a general (i.e. nonquadratic) setting.

A simple model for the interactions of MPF and cyclin is given by Novak and Tyson (1993):

$$\begin{aligned} \dot{z}_1 &= \frac{0.01}{1.1} - \left(\frac{\bar{k}_2}{10} + k_w\right) z_1 + \left(0.04 + 100 z_2^2\right)\left(\frac{z_2}{1.1} - z_1\right) \\ \dot{z}_2 &= 0.01 - \frac{\bar{k}_2}{10} z_2 \\ z_1(0) &= 0.01, \;\; z_2(0) = 0.75 \end{aligned} \tag{20}$$

where z_1 and z_2 represent (non-dimensional) concentrations of MPF and total cyclin, respectively. k_w depends on the concentration of another molecule, Wee1, and it is treated as an input to the system. The only uncertainty lies in the rate constant, $\bar{k}_2$. The prior information for this parameter is $\mathcal{H} = \{\bar{k}_2 \;:\; 0.1 \le \bar{k}_2 \le 0.4\}$.[5]

Measurements of MPF concentration at $t = 60$ minutes are taken for two values of k_w: $k_{w,1} = 3.5$ and $k_{w,2} = 6.0$. The information for the two experimental processes is given in Table 3. The surrogate models were computed using data from 5 points in $\mathcal{H}$. Note that cubic surrogate models were required to adequately fit the mathematical models. The two measurements, d_1 and d_2, were generated by simulating a *true* system and then adding an error term randomly selected from $[-5e-4, \;\; 5e-4]$. The true system is obtained by replacing the uncertain constant $\bar{k}_2$ in Eq. (20) with $k_2(z_1) \doteq 0.1 + 1000 z_1^2$. This function represents a negative feedback in the production of MPF and appears to more closely model the cell division process.

The model in Eq. (20) can be invalidated by using Theorem 2(A) to prove that $\mathcal{F}_O$ is empty. The outer approximation of the feasible set is given by $\mathcal{F}_O \doteq \{x \in \mathbb{R} \;:$

[5] To improve numerical conditioning, $\bar{k}_2$ has been rescaled from the model given in Novak and Tyson (1993).

$p_k(x) \leq 0,\ k = 1, \ldots, 5\}$ where:

$$p_1(x) = S_1(x) - d_1 - (u_1 + e_1) \qquad p_2(x) = -S_1(x) + d_1 - (u_1 + e_1)$$
$$p_3(x) = S_2(x) - d_2 - (u_2 + e_2) \qquad p_4(x) = -S_2(x) + d_2 - (u_2 + e_2)$$
$$p_5(x) = (x - 0.25)^2 - 0.0225$$

Using SOSTOOLS, we found the following multipliers, $\{\lambda_k\}_{k=1}^5 \in \mathcal{S}$, such that $-1 + \sum_{k=1}^5 \lambda_k p_k \in \mathcal{S}$:

$$\lambda_1(x) = 4054.8x^2 - 36.7x + 4350.2 \qquad \lambda_2(x) = 3976.4x^2 - 1202.1x + 387.5$$
$$\lambda_3(x) = 3853.5x^2 - 1068.2x + 785.1 \qquad \lambda_4(x) = 5073.7x^2 + 6488x + 17675$$
$$\lambda_5(x) = 7439.2x^2 - 1772.8x + 139.5$$

By Theorem 2(A), $\mathcal{F}_O$ is empty and hence $\mathcal{F}$ is empty. Thus, the proposed model has been invalidated using the two dataset units.

7 Conclusions

In this paper, we presented a numerical approach for model validation and prediction. The proposed algorithm relies on a solution mapping technique as well as recent results for polynomial optimizations. We then illustrated the approach to prediction and validation via two simple examples. We believe that this approach provides a framework for collaborative data processing among researchers and that it can be successfully applied to other scientific fields.

Acknowledgment We gratefully acknowledge support from the NSF as well as helpful discussions with John Doyle, Pablo Parrilo and Stephen Prajna.

References

Box GEP, Draper NR (1987) Empirical model-building and response surfaces. Wiley

Box GEP, Hunter WG, Hunter JS (1978) Statistics for experimenters. an introduction to design, data analysis, and model building. Wiley

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

Choi M, Lam T, Reznick B (1995) Sums of squares of real polynomials. Proc Symp Pure Math 58(2):103–126

Folland G (1999) Real analysis: modern techniques and their applications. Wiley

Frazzoli E, Mao Z-H, Oh J-H, Feron E (2001) Resolution of conflicts involving many aircraft via semidefinite programming. AIAA J Guidance, Control, Dynamics 24(1):79–86

Frenklach M (1984) 'Chapter 7: modeling'. In: Gardiner JWC (ed) Combustion chemistry. Springer-Verlag

Frenklach M, Miller D (1985) 'Statistically rigorous parameter estimation in dynamic modeling using approximate empirical models'. AIChE J 31, 498–500

Frenklach M, Packard A, Seiler P (2002) Prediction uncertainty from models and data. In: Proceedings of the american control conference pp. 4135–4140

Frenklach M, Packard A, Seiler P, Feeley R (2004) Collaborative data processing in developing predictive models of complex reaction systems. J Chemical Kinetics 36(1):57–66

Frenklach M, Wang H, Rabinowitz M (1992) Optimization and analysis of large chemical kinetic mechanisms using the solution mapping method—combustion of methane. Prog Energy Combust Sci 18(47)

Gatermann K, Parrilo P (2004) Symmetry groups, semidefinite programs, and sums of squares. Journal of Pure and Appl. Algebra 192(1-3):95–128

Labit Y, Peaucelle D, Henrion D (2002) 'SeDuMi interface 1.02: a tool for solving LMI problems with SeDuMi'. In: IEEE international symposium on computer aided control system design proceedings. pp. 272–276

Lasserre J (2001) Global optimization with polynomials and the problem of moments. SIAM J Optim 11(3):796–817

Miller D, Frenklach M (1983) Sensitivity analysis and parameter estimation in dynamic modeling of chemical kinetics. Int J Chem Kinetics 15:677–696

Myers R, Montgomery D (2002) Response surface methodology. Wiley, New York

Novak B, Tyson J (1993) 'Modeling the cell division cycle: M-phase trigger, oscillations, and size control'. J Theor Biol 165:101–134

Parrilo P (2000) Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. Ph.D. thesis, California Institute of Technology

Parrilo P (2003) Semidefinite programming relaxations for semialgebraic problems. Mathematical Programming Ser. B 96(2):293–320

Powers V, Wörmann T (1998) An algorithm for sums of squares of real polynomials. J Pure Appl Alg 127:99–104

Prajna S, Papachristodoulou A, Parrilo PA (2002), SOSTOOLS: sum of squares optimization toolbox for MATLAB. Available from `http://www.cds.caltech.edu/sostools` and `http://www.aut.ee.ethz.ch/~parrilo/sostools`.

Putinar M (1993) Positive polynomials on compact semi-algebraic sets. Indiana Univ Math J 42(3):969–984

Reznick B (1978) Extremal PSD forms with few terms. Duke Math J 45(2):363–374

Reznick B (2000) Some concrete aspects of Hilbert's 17th problem. Contemp Math 253:251–272

Smith G, Golden D, Frenklach M, Moriarty N, Goldenberg BEM, Bowman C, Hanson R, Song S, Gardiner JWC, Lissianski V, Qin Z, 'GRI-Mech 3.0'. http://www.me.berkeley.edu/gri_mech/.

Stengle G (1974) A nullstellensatz and a positivstellensatz in semialgebraic geometry. Math Ann 207:87–97

Sturm J (2001) SeDuMi version 1.05. http://fewcal.kub.nl/sturm/software/sedumi.html

Tyson J, Borisuk M, Chen K, Novak B (2001) Computational modeling of genetic and biochemical networks, Chapt. 10: Analysis of complex dynamics in cell cycle regulation. MIT Press

Vandenberghe L, Boyd S (1996) Semidefinite programming. SIAM Rev 38(1):49–95