Optimization-based Domain Reduction in Guaranteed Parameter Estimation of Nonlinear Dynamic Systems

Radoslav Paulen [*,**] **Mario Villanueva** [**] **Benoît Chachuat** [**]

* *Process Dynamics and Operations Group, Department of Biochemical and Chemical Engineering, Technische Universität Dortmund, Emil-Figge-Str. 70, 44221 Dortmund, Germany (e-mail: radoslav.paulen@bci.tu-dortmund.de).*
** *Centre for Process Systems Engineering, Department of Chemical Engineering, Imperial College London, South Kensington Campus, London SW7 2AZ, UK (e-mail: {mario.villanueva10, b.chachuat}@imperial.ac.uk)*

Abstract: This paper is concerned with guaranteed parameter estimation in nonlinear dynamic systems in a context of bounded measurement error. The problem consists of finding—or approximating as closely as possible—the set of all possible parameter values such that the predicted outputs match the corresponding measurements within prescribed error bounds. An exhaustive search procedure is applied, whereby the parameter set is successively partitioned into smaller boxes and exclusion tests are performed to eliminate some of these boxes, until a prespecified threshold on the approximation level is met. In order to enhance the convergence of this procedure, we investigate the use of optimization-based domain reduction techniques for tightening the parameter boxes before partitioning. We construct such bound-reduction problems as linear programs from the polyhedral relaxation of Taylor models of the predicted outputs. When applied to a simple case study, the proposed approach is found to reduce the computational burden significantly, both in terms of CPU time and number of iterations.

Keywords: Parameter estimation · Dynamic systems · Bounded-error estimation · Measurement noise · Taylor models · Polyhedral relaxations · Domain reduction

1. INTRODUCTION

Process model development has become an integral part of modern process design methodologies as well as for control system design and operations optimization. A typical model development procedure is divided into two main phases, namely specification of the model structure and estimation of the unknown/uncertain model parameters. The latter phase, also known as model fitting, normally proceeds by determining parameter values for which the model predictions closely match the observed process. Failure to find an acceptable agreement calls for a revision of the model structure, and the parameter estimation is then repeated.

Most commonly, the parameter estimation problem is posed as an optimization problem that determines the parameter values minimizing the gap between the measurements and the model predictions, for instance in the least-square sense. Nonetheless, several factors can jeopardize a successful and reliable estimation procedure. First of all, structural model mismatch is inherent to the modeling exercise, and it would be an illusion to seek for the 'true' parameter values in this context. Even in the absence of model mismatch, fitting a set of experimental data exactly is generally not possible due to various sources of uncertainty. A measurement's accuracy is always tied to

the resolution of its corresponding measuring apparatus. Moreover, measured data are typically corrupted with noise, for instance Gaussian white noise or more generally colored noise.

Among the possible approaches that account for uncertainty in parameter estimation, the focus in this paper is on *guaranteed* parameter estimation [Jaulin and Walter, 1993], namely the determination of *all* parameter values—referred to as the solution set subsequently—that are consistent with the measurements under given uncertainty scenarios. Specifically, we consider the case that the uncertainty enters the estimation problem in the form of bounded measurement errors. In small-scale applications, the problem of approximating the solution set by a box partition, at an arbitrary precision, has been shown to be tractable by Walter and coworkers [Jaulin and Walter, 1993, Kieffer and Walter, 2011] using a set-inversion algorithm based on exhaustive search.

The same authors [Kieffer and Walter, 2011] also identify that the main bottlenecks, in terms of convergence speed and accuracy, of such a set-inversion algorithm are: (i) the need to compute tight bounds on the solutions of the dynamic system; and, (ii) the need to apply efficient domain-reduction strategies as a part of the exclusion tests. The first issue was addressed recently, e.g., in Lin

and Stadtherr [2007b] and Paulen et al. [2013], by applying novel bounding techniques for parametric dynamic systems based on Taylor model arithmetic or ellipsoidal calculus. Large improvements of the rate of convergence of the set-inversion algorithm could be obtained this way.

The main focus in this paper is on domain-reduction strategies as a means to enhance the convergence of the set-inversion procedure. Interval contractor schemes were presented in [Jaulin et al., 2001] and applied to dynamic guaranteed parameter estimation problems in [Kieffer and Walter, 2011]. Instead, the approach used here is inspired from recent developments in global dynamic optimization Sahlodin [2012]. We construct bound-reduction optimization problems in the form of linear programs (LPs) from the polyhedral relaxation of Taylor models of the predicted outputs. In other words, we exploit the available measurement bounds so as to indirectly exclude parts of the corresponding parameter subset.

The paper is organized as follows. The problem of guaranteed parameter estimation and the set-inversion algorithm are reviewed in Sect. 2. The Taylor model-based approach for ODE bounding and the optimization-based domain reduction approach are described in Sect. 3. The benefits of using optimization-based domain reduction are then illustrated with numerical case studies and discussed in Sect. 4. Finally, Sect. 5 concludes the paper.

2. GUARANTEED PARAMETER ESTIMATION

2.1 Problem Statement

Consider a model of the observed process described by parametric ODEs of the form

$$\dot{\boldsymbol{x}}(t;\boldsymbol{p}) = \boldsymbol{f}(\boldsymbol{x}(t;\boldsymbol{p}),\boldsymbol{p}), \qquad \boldsymbol{x}(t_0;\boldsymbol{p}) = \boldsymbol{h}(\boldsymbol{p}), \qquad (1a)$$
$$\hat{\boldsymbol{y}}(t;\boldsymbol{p}) = \boldsymbol{g}(\boldsymbol{x}(t;\boldsymbol{p}),\boldsymbol{p}), \qquad (1b)$$

where $\boldsymbol{x}$ denotes the n_x-dimensional vector of process states; $\boldsymbol{p}$, the n_p-dimensional vector of process (a priori unknown) parameters; and $\hat{\boldsymbol{y}}$, the n_y-dimensional vector of model outputs (predictions).

Given a set of output measurements $\boldsymbol{y}_m$ at N time points $t_1, \ldots, t_N$, *classical* parameter estimation seeks for *one* particular instance $\boldsymbol{p}_e$ of the parameters for which the (possibly weighted) normed difference between these measurements and the corresponding model outputs $\hat{\boldsymbol{y}}$ is minimized. In contrast, *guaranteed* (bounded-error) parameter estimation accounts for the fact that the actual process outputs, $\boldsymbol{y}_p$, are only known within some bounded measurement errors $\boldsymbol{e} \in \boldsymbol{E} := [\boldsymbol{e}^L, \boldsymbol{e}^U]$, so that

$$\boldsymbol{y}_p(t_i) \in \boldsymbol{y}_m(t_i) + [\boldsymbol{e}^L, \boldsymbol{e}^U] =: \boldsymbol{Y}_i, \qquad (2)$$

where the superscripts ^{L} and ^{U} represent the lower and upper bounds of an interval box (understood component-wise throughout).

Depicted in red on the top plot of Figure 1 is the set of all output trajectories satisfying $\hat{\boldsymbol{y}}(t_i;\boldsymbol{p}) \in \boldsymbol{Y}_i$, $i = 1, \ldots, N$. Here, the main objective is to estimate the set $\boldsymbol{P}_e$ of *all* possible parameter values $\boldsymbol{p}$ that yield those trajectories; that is,
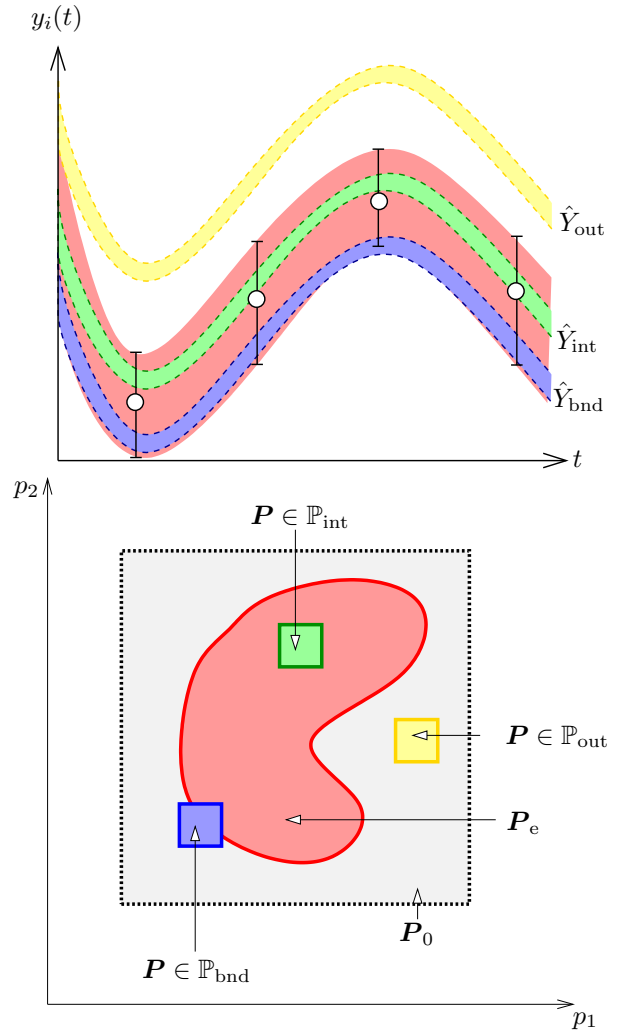

Fig. 1. Illustration of guaranteed parameter estimation concepts in the space of output trajectories (top plot) and parameters (bottom plot).

$$\boldsymbol{P}_e := \left\{ \boldsymbol{p} \in \boldsymbol{P}_0 \left| \begin{array}{l} \exists \boldsymbol{x}, \hat{\boldsymbol{y}} \text{ such that:} \\ \dot{\boldsymbol{x}}(t;\boldsymbol{p}) = \boldsymbol{f}(\boldsymbol{x}(t;\boldsymbol{p}),\boldsymbol{p}), \\ \boldsymbol{x}(t_0;\boldsymbol{p}) = \boldsymbol{h}(\boldsymbol{p}), \\ \hat{\boldsymbol{y}}(t_i;\boldsymbol{p}) = \boldsymbol{g}(\boldsymbol{x}(t_i;\boldsymbol{p}),\boldsymbol{p}), \\ \hat{\boldsymbol{y}}(t_i;\boldsymbol{p}) \in \boldsymbol{Y}_i, \\ \forall t \in [t_0, t_N], \forall i \in \{1, \ldots, N\} \end{array} \right. \right\}. \quad (3)$$

The projection of $\boldsymbol{P}_e$ onto the (p_1, p_2) space is illustrated on the bottom plot of Figure 1. Obtaining an exact characterization of the set $\boldsymbol{P}_e$ is not possible in general, and one has to resort to approximation techniques to make the problem computationally tractable. The focus in the remainder of the paper is on algorithms that provide approximations of $\boldsymbol{P}_e$ by exploiting set-inversion techniques.

2.2 Algorithmic Procedure

We consider a variant of the Set Inversion Via Interval Analysis (SIVIA) algorithm by [Jaulin and Walter, 1993] in order to approximate the solution set $\boldsymbol{P}_e$ to a desired accuracy; see also Kieffer and Walter [2011] for applications of this algorithm in the context of guaranteed parameter estimation.

Let $\boldsymbol{F}_t$ denote the mapping associated to the ODE model (1), such that

$$\forall \boldsymbol{p} \in \boldsymbol{P}_0 : \quad \boldsymbol{F}_t(\boldsymbol{p}) = \hat{\boldsymbol{y}}(t; \boldsymbol{p}), \tag{4}$$

for each $t \in [t_0, t_N]$. It follows that characterizing $\boldsymbol{P}_e$ via (3) is equivalent to intersecting the inverse image sets $\boldsymbol{F}_{t_i}^{-1}(\boldsymbol{Y}_i)$ of $\boldsymbol{Y}_i$ for each $i = 1, \ldots, N$; that is,

$$\boldsymbol{P}_e = \left(\bigcap_{i=1}^{N} \boldsymbol{F}_{t_i}^{-1}(\boldsymbol{Y}_i) \right) \cap \boldsymbol{P}_0. \tag{5}$$

The set-inversion algorithm proceeds as follows:

Input: Termination tolerances $\epsilon_{\text{box}} > 0$ and $\epsilon_{\text{bnd}} > 0$
Initialization: Set partitions $\mathbb{P}_{\text{bnd}} = \{\boldsymbol{P}_0\}$, $\mathbb{P}_{\text{int}} = \emptyset$, $\mathbb{P}_{\text{out}} = \emptyset$; Set iteration counter $k = 0$
Main Loop:
  (1) Select a parameter box $\boldsymbol{P}$ in the partition $\mathbb{P}_{\text{bnd}}$ and remove it from $\mathbb{P}_{\text{bnd}}$
  (2) Compute enclosures $\hat{\boldsymbol{Y}}(t_i; \boldsymbol{P}) \supseteq \boldsymbol{F}_{t_i}(\boldsymbol{P})$, for each $i = 1, \ldots, N$ and $\forall \boldsymbol{p} \in \boldsymbol{P}$
  (3) Exclusion Tests:
     (a) **If** $\hat{\boldsymbol{Y}}(t_i; \boldsymbol{P}) \subseteq \boldsymbol{Y}_i$ for all $i \in \{1, \ldots, N\}$, insert $\boldsymbol{P}$ into $\mathbb{P}_{\text{int}}$
     (b) **Else if** $\hat{\boldsymbol{Y}}(t_i; \boldsymbol{P}) \cap \boldsymbol{Y}_i = \emptyset$ for some $i \in \{1, \ldots, N\}$, insert $\boldsymbol{P}$ into $\mathbb{P}_{\text{out}}$
     (c) **Else** bisect $\boldsymbol{P}$ and insert subsets back into $\mathbb{P}_{\text{bnd}}$
  (4) Termination Tests:
     (a) **If** $\mathbb{P}_{\text{bnd}} = \emptyset$, **stop**
     (b) **If** $V_{\text{bnd}} := \sum_{\boldsymbol{P} \in \mathbb{P}_{\text{bnd}}} \text{volume}(\boldsymbol{P}) < \epsilon_{\text{bnd}}$, **stop**
     (c) **If** width$(\boldsymbol{P}) < \epsilon_{\text{box}}$ for all $\boldsymbol{P} \in \mathbb{P}_{\text{bnd}}$, **stop**
  (5) Increment counter $k {+}{=} 1$; return to step 1
Output: Partitions $\mathbb{P}_{\text{int}}$ and $\mathbb{P}_{\text{bnd}}$; Iteration count k

---

An illustration of parameter subboxes belonging to the partitions $\mathbb{P}_{\text{int}}$, $\mathbb{P}_{\text{bnd}}$, and $\mathbb{P}_{\text{out}}$ is shown on the bottom plot in Figure 1, together with the corresponding output trajectories on the top plot using the same color code. Upon termination, this algorithm returns partitions $\mathbb{P}_{\text{int}}$ and $\mathbb{P}_{\text{bnd}}$ such that

$$\bigcup_{\boldsymbol{P} \in \mathbb{P}_{\text{int}}} \boldsymbol{P} \subseteq \boldsymbol{P}_e \subseteq \left(\bigcup_{\boldsymbol{P} \in \mathbb{P}_{\text{int}}} \boldsymbol{P} \right) \cup \left(\bigcup_{\boldsymbol{P} \in \mathbb{P}_{\text{bnd}}} \boldsymbol{P} \right). \tag{6}$$

A number of remarks are in order:

- Multiple heuristics can be used regarding the selection of a parameter box in step 1 or the bisection in step 3c. Here, we select the widest parameter box in step 1 and bisect it at the mid-point along the least reduced axis in step 3c.
- Step 2 involves bounding the solution set of the parametric ODEs (1) for the current box $\boldsymbol{P}$. In this study, we employ verified ODE methods based on Taylor models to compute such over-approximations [Sahlodin and Chachuat, 2011]. Note that the resulting enclosures $\hat{\boldsymbol{Y}}(t_i; \boldsymbol{P})$ shrink as width$(\boldsymbol{P}) \to 0$ and the set-inversion algorithm thus terminates *finitely* for any finite tolerances $\epsilon_{\text{box}} > 0$ and $\epsilon_{\text{bnd}} > 0$.
- Test 4b is an addition to the original SIVIA algorithm [Jaulin and Walter, 1993], which interrupts the iterations when a specified level of approximation of the solution set $\boldsymbol{P}_e$ is reached. The level of approxi-

mation is measured here as the total volume V_{bnd} of the boxes in the partition $\mathbb{P}_{\text{bnd}}$, with corresponding threshold ϵ_{bnd}. In contrast, stopping the algorithm when a minimum box width is reached cannot give any guarantee on the approximation level because of the overestimation in step 2.

3. BOUNDS FOR PARAMETRIC ODE AND OPTIMIZATION-BASED DOMAIN REDUCTION

3.1 Taylor Model-based Bounds

Given a non-empty set $\boldsymbol{P} \in \mathbb{R}^{n_p}$ and a $\mathcal{C}^{q+1}$ function $\phi : \boldsymbol{P} \to \mathbb{R}$, with $q \geq 0$, $\mathcal{T}_{\phi, \boldsymbol{P}}^q := \mathcal{P}_{\phi, \boldsymbol{P}}^q + \mathcal{R}_{\phi, \boldsymbol{P}}^q$ is called a *qth-order Taylor model of ϕ on $\boldsymbol{P}$* if [Bompadre et al., 2012]

- the n_p-variate polynomial $\mathcal{P}_{\phi, \boldsymbol{P}}^q$ is such that

$$\forall \boldsymbol{p} \in \boldsymbol{P} : \quad \mathcal{P}_{\phi, \boldsymbol{P}}^q(\boldsymbol{p}) = \sum_{\substack{\boldsymbol{\kappa} \in \mathbb{N}^{n_p}, \\ |\boldsymbol{\kappa}| \leq q}} \frac{\partial^{\boldsymbol{\kappa}} \phi(\boldsymbol{p}^*)}{\boldsymbol{\kappa}!} (\boldsymbol{p} - \boldsymbol{p}^*)^{\boldsymbol{\kappa}}, \tag{7}$$

  for some $\boldsymbol{p}^* \in \boldsymbol{P}$ (using multi-index notation);

- the remainder interval $\mathcal{R}_{\phi, Y}^q$ is such that

$$\forall \boldsymbol{p} \in \boldsymbol{P} : \quad \phi(\boldsymbol{p}) - \mathcal{P}_{\phi, \boldsymbol{P}}^q(\boldsymbol{p}) \in \mathcal{R}_{\phi, \boldsymbol{P}}^q. \tag{8}$$

Taylor models can be constructed recursively for factorable functions, which are defined by a finite recursive composition of binary sums, binary products, and a given library of univariate intrinsic functions such as $\exp(\cdot)$, $\sqrt{\cdot}$, etc. This recursive procedure is initiated with a known Taylor model, which can be the Taylor model of a variable or, in the case of a composite function, of the inner function.

Verified ODE integration based on Taylor models [Lin and Stadtherr, 2007a, Sahlodin and Chachuat, 2011] constructs a Taylor model $\mathcal{T}_{\boldsymbol{x}(t_i), \boldsymbol{P}}^q$ of the state variables $\boldsymbol{x}(t_i; \cdot)$ on $\boldsymbol{P}$ at given times $t_i \in [t_0, t_N]$; that is,

$$\forall \boldsymbol{p} \in \boldsymbol{P} : \quad \boldsymbol{x}(t_i; \boldsymbol{p}) \subseteq \mathcal{T}_{\boldsymbol{x}(t_i), \boldsymbol{P}}^q. \tag{9}$$

This technique proceeds in two steps:

(1) **Predictor step.** Given a Taylor model $\mathcal{T}_{\boldsymbol{x}(t_j), \boldsymbol{P}}^q$ of $\boldsymbol{x}(t_j; \cdot)$ on $\boldsymbol{P}$, this step computes a stepsize h_j and an a priori (interval) enclosure $\tilde{\boldsymbol{X}}_j$ of $\boldsymbol{x}$ on $[t_j, t_{j+1}] \times \boldsymbol{P}$, with $t_{j+1} := t_j + h_j$. Existence and uniqueness of the parametric ODE solutions on $[t_j, t_{j+1}]$ is also verified.

(2) **Corrector step.** Given a Taylor model $\mathcal{T}_{\boldsymbol{x}(t_j), \boldsymbol{P}}^q$ of $\boldsymbol{x}(t_j; \cdot)$ on $\boldsymbol{P}$ and an a priori enclosure $\tilde{\boldsymbol{X}}_j$ of $\boldsymbol{x}$ on $[t_j, t_{j+1}] \times \boldsymbol{P}$, this step computes a Taylor model $\mathcal{T}_{\boldsymbol{x}(t_{j+1}), \boldsymbol{P}}^q$ of $\boldsymbol{x}(t_{j+1}; \cdot)$ on $\boldsymbol{P}$, based on a high-order Taylor expansion of the solution in combination with the mean-value theorem for wrapping mitigation.

In turn, the required enclosures $\hat{\boldsymbol{Y}}(t_i; \boldsymbol{P})$ over a given box $\boldsymbol{P}$ can be obtained in the form of qth-order Taylor models $\hat{\boldsymbol{Y}}(t_i; \boldsymbol{P}) = \mathcal{T}_{\hat{\boldsymbol{y}}(t_i), \boldsymbol{P}}^q := \mathcal{P}_{\hat{\boldsymbol{y}}(t_i), \boldsymbol{P}}^q + \mathcal{R}_{\hat{\boldsymbol{y}}(t_i), \boldsymbol{P}}^q$, via Taylor model propagation through the output function (1b). A key feature of these enclosures is their explicit dependence in the parameters $\boldsymbol{p}$, thereby making it possible to apply advanced domain-reduction techniques.

3.2 Optimization-based Domain Reduction

Variants of the set-inversion algorithm have been developed in order to improve its convergence properties. For instance, domain contractors based on interval gradients have been considered in [Kieffer and Walter, 2011]. The downside of this approach, however, is the need to compute enclosures of the first-order sensitivities of the ODEs (1), which can cause a significant computational overhead.

In this paper, we propose optimization-based domain reduction techniques that exploit the Taylor models of the predicted model outputs. Given a parameter box $\boldsymbol{P}$ as well as qth-order Taylor models $\mathcal{T}^q_{\hat{\boldsymbol{y}}(t_i),\boldsymbol{P}}$ of the model outputs $\hat{\boldsymbol{y}}(t_i;\cdot)$ on $\boldsymbol{P} =: [\boldsymbol{p}^L,\boldsymbol{p}^U]$, the bounds $\boldsymbol{p}^L$ and $\boldsymbol{p}^U$ can be tightened by solving optimization problems of the form

$$p_j^{L/U} = \max_{\boldsymbol{p}\in\boldsymbol{P}}/\min_{\boldsymbol{p}\in\boldsymbol{P}} p_j \tag{10a}$$

$$\text{s.t. } \mathcal{P}^q_{\hat{\boldsymbol{y}}(t_i),\boldsymbol{P}}(\boldsymbol{p}) + r^{q,\mathrm{L}}_{\hat{\boldsymbol{y}}(t_i),\boldsymbol{P}} \leq \boldsymbol{Y}^L_i, \quad i=1,\dots,N \tag{10b}$$

$$\mathcal{P}^q_{\hat{\boldsymbol{y}}(t_i),\boldsymbol{P}}(\boldsymbol{p}) + r^{q,\mathrm{U}}_{\hat{\boldsymbol{y}}(t_i),\boldsymbol{P}} \geq \boldsymbol{Y}^U_i, \quad i=1,\dots,N. \tag{10c}$$

This way, a reduced box $\boldsymbol{P}$ is obtained after solving $2\times n_p$ optimization problems—one problem for the lower bound and one problem for the upper bound of each parameter.

Since the polynomial part of a qth-order Taylor model is in general nonconvex whenever $q\geq 2$, the bound-reduction problems (10) are themselves nonconvex in general. Instead of solving these problems directly, we construct polyhedral relaxations in the form of linear programs (LPs), similar to the approach used in deterministic global optimization for computing lower/upper bounds—see, e.g., [Tawarmalani and Sahinidis, 2004, Sahlodin, 2012]. This relaxation procedure follows three steps:

(1) **Decomposition.** The multivariate polynomials $\mathcal{P}^q_{\hat{\boldsymbol{y}}(t_i),\boldsymbol{P}}$ are decomposed into elementary factors (binary sums and products and univariate compositions) via the introduction of auxiliary variables. The constraints in the reformulated optimization problem are either linear or contain a single bilinear term $p_j\times p_k$ or integer power term $(p_j)^k$.

(2) **Relaxation.** The nonconvex terms in the reformulated problem are relaxed so as to obtain convex optimization problems. This relaxation involves replacing bilinear and power terms with their convex/concave envelopes.

(3) **Polyhedral outer approximation.** Since the convex/concave envelopes of power terms are nonlinear in general, polyhedral outer-approximation are constructed via linearization at a number of points—usually chosen so as to meet a given level of accuracy.

By construction, the relaxed optimization problems are fully linear, making it possible to exploit the robustness, efficiency and speed of state-of-the-art LP solvers such as GUROBI or CPLEX.

In practice, this domain reduction procedure can be performed as an extra step between the steps (2) and (3) of the set-inversion algorithm in Sect. 2.2. Moreover, in case the reduction of a parameter box $\boldsymbol{P}$ is larger than a given threshold, for instance $\geq 20\%$ in volume, it can be repeated multiple times. Nonetheless, repeating the reduction several times requires recomputing the Taylor models $\mathcal{T}^q_{\hat{\boldsymbol{y}}(t_i),\boldsymbol{P}}$ of the model outputs on the reduced box $\boldsymbol{P}$. This defines a trade-off between the extra computational burden and the reduction in the size of the partition, which is clearly problem dependent.

4. CASE STUDIES

This section illustrates the benefits of using optimization-based domain reduction within the guaranteed parameter estimation algorithm with two simple numerical case studies. We note that the current implementation does not account for round-off and numerical errors, e.g., when LP problems are solved.

The set-inversion algorithm presented in Sect. 2.2 is implemented in a C++ program, and linked to the MC++ library (http://www3.imperial.ac.uk/people/b.chachuat/research), which features a collection of C++ classes for computations in interval and Taylor model arithmetics. The verified ODE integration method described in Sect. 3.1 and the domain-reduction technique described in Sect. 3.2 are also implemented in this library. The resulting LP problems are solved with the solver GUROBI (version 5.1.0) [Gurobi Optimization, 2012], and all the computations are performed on an Intel Core i7-3770 workstation with 3.4 GHz CPU using GCC (version 4.6.3) compiler.

4.1 Case Study 1

A dynamic model involving two state variables $\boldsymbol{x} = (x_1,x_2)^\mathsf{T}$ and three uncertain model parameters $\boldsymbol{p} = (p_1,p_2,p_3)^\mathsf{T} \in [0.01,1]^3 =: \boldsymbol{P}_0$ is considered [Kieffer and Walter, 2011]:

$$\dot{x}_1 = -(p_1+p_3)x_1 + p_2x_2, \qquad x_1(0)=1, \tag{11a}$$

$$\dot{x}_2 = p_1x_1 - p_2x_2, \qquad x_2(0)=0. \tag{11b}$$

The system has a single output variable $\hat{y}$, which corresponds to the state variable x_2, $\hat{y}(t;\boldsymbol{p}) = x_2(t;\boldsymbol{p})$, with $N=15$ measurements corresponding to the times $t_i = 1,\dots,15$. Synthetic experimental data are generated by simulating the model (11) with parameter values $\boldsymbol{p}^0 = (0.6,0.15,0.35)^\mathsf{T}$, and then rounding the output $\hat{y}(t_i;\boldsymbol{p})$ up or down to the nearest value by retaining two significant digits only.

The guaranteed parameter estimation algorithm is applied with and without the use of domain reduction (reduction threshold of 20% and maximum of 10 reduction loops), and Taylor models of orders $q=2,3,4$ are considered as enclosures for the model outputs $\hat{\boldsymbol{Y}}(t_i;\boldsymbol{P})$. In order to allow for fair comparisons, the termination criterion is defined in terms of the level of accuracy ϵ_{bnd} of the solution set, here in the range 5×10^{-6} to 10^{-3}, while the termination criterion in terms of the minimum box size ϵ_{box} is set to 0.

The number of iterations and the CPU time required by the set-inversion algorithm to terminate with different Taylor model orders and with or without the use of domain reduction are reported on the top and bottom plots of Fig. 2, respectively, as a function of the termination tolerance ϵ_{bnd}. It is evident that the number of iterations is significantly reduced when using domain reduction—here by
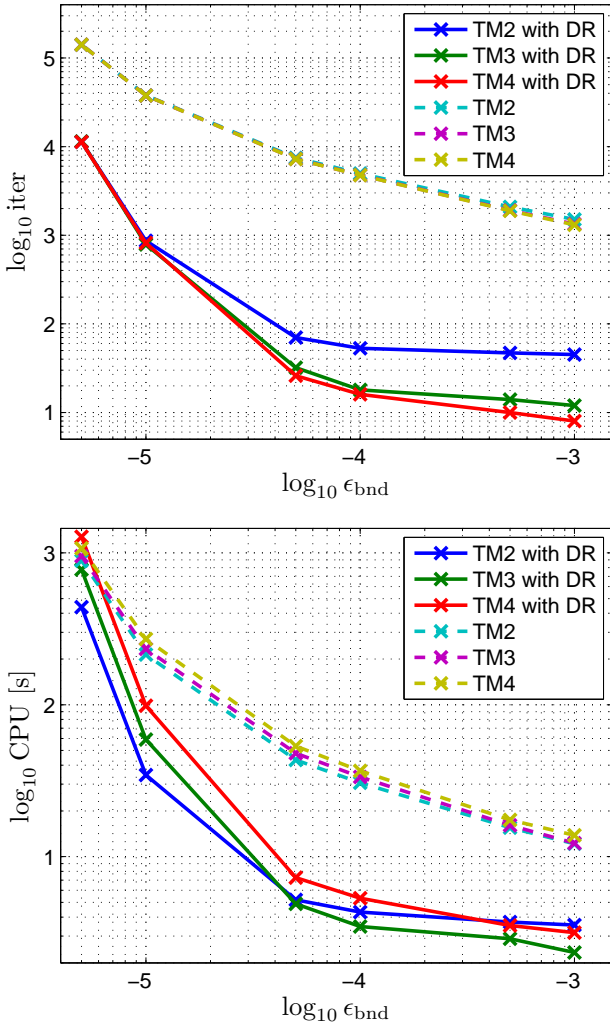
Fig. 2. Comparison of guaranteed parameter estimation algorithm for various Taylor model orders and with or without the use of domain reduction (DR). *Top:* Number of iterations vs. convergence threshold. *Bottom:* CPU time vs. convergence threshold.

one order of magnitude at least for all considered tolerance levels ϵ_{bnd}. Moreover, the higher the order of the Taylor model, the smaller the number of iterations required by the algorithm to converge. Nonetheless, a certain trade-off is observed in terms of CPU time on the lower plot of Fig. 2, as higher-order Taylor models can incur a possibly large computational overhead. Although higher-order Taylor models allow tighter approximations, lower-order Taylor models become computationally advantageous as the parameter boxes shrink. Another trade-off is observed in terms of the overhead incurred by the application of domain reduction (LP problem construction and solution). The observed trend clearly calls for the development of a hybrid strategy that would apply domain reduction until a given level of approximation on the parameter set $\boldsymbol{P}_{\mathrm{e}}$ is reached.

The top and bottom plots of Fig. 3 show the projections of the solution set outer-approximation in the (p_1, p_2) and (p_2, p_3) subspaces, respectively, for increasing accuracy levels of $\epsilon_{\mathrm{bnd}} = 5 \times 10^{-4}$, 5×10^{-5}, and 5×10^{-6}—second-order Taylor model and domain reduction are used here.
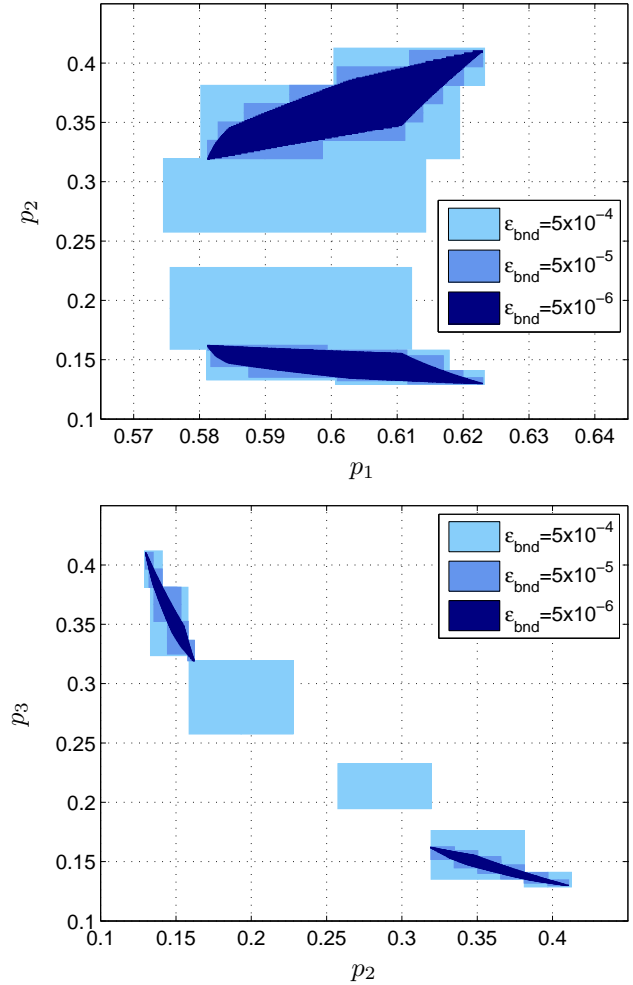

Fig. 3. Guaranteed parameter set approximations corresponding to different termination tolerances with 2nd-order Taylor models and domain reduction. *Top:* Projections into (p_1, p_2) space. *Bottom:* Projections into (p_2, p_3) space.

Note first that the parameter values $\boldsymbol{p}^0 = (0.6, 0.15, 0.35)^{\mathsf{T}}$ used to generate the experimental data are part of the solution set. Moreover, the solution set turns out to be disconnected in this problem, thereby suggesting a potential structural identifiability problem—this was already reported by Kieffer and Walter [2011]. In comparing different approximation levels of the boundary of set $\boldsymbol{P}_{\mathrm{e}}$, it is seen that setting $\epsilon_{\mathrm{bnd}} = 5 \times 10^{-5}$ already provides a tight approximation of the actual solution set $\boldsymbol{P}_{\mathrm{e}}$, with only 34 boxes and a CPU time of 5 sec. As expected, a much tighter approximation is obtained by setting $\epsilon_{\mathrm{bnd}} = 5 \times 10^{-6}$, yet this is at the price of a much finer partition comprising of 11,250 boxes and a corresponding CPU time of over 400 sec. For the sake of comparison, we also note that, when no domain reduction is used, the partition comprises of over 2,200 boxes with $\epsilon_{\mathrm{bnd}} = 5 \times 10^{-5}$ and over 70,000 boxes with $\epsilon_{\mathrm{bnd}} = 5 \times 10^{-6}$. These results also suggest that the efficiency of the guaranteed parameter estimation algorithm in computing highly-accurate set approximations could be improved significantly if affine cuts were enabled in addition to simple bounds contraction during the domain-reduction procedure. Such cuts would provide

the extra flexibility needed to closely approximate the actual parameter set, thereby further reducing the number of iterations and the CPU time.

4.2 Case Study 2

Consider the same dynamic system and estimation problem as in Sect. 4.1, with the difference that the initial condition of the state x_2 is now uncertain:

$$\dot{x}_2 = p_1 x_1 - p_2 x_2, \quad x_2(0; \boldsymbol{p}) = p_4, \quad p_4 \in [0, 0.1]. \quad (12)$$

In order to allow for fair comparisons with this extra parameter, the termination tolerance ϵ_{bnd} in the set-inversion algorithm is taken in the range 5×10^{-7} to 10^{-4}. Moreover, a maximum CPU time of 1 hour is considered.

In the case that no domain reduction is used, the computations show a dramatic increase in the number of iterations—more than 200-fold compared to the 3-parameter problem. In contrast, an increase of around 15 times is observed when applying optimization-based domain reduction. These results confirm that domain reduction techniques hold much promise in the context of guaranteed parameter estimation, especially for dynamic systems with a larger number of parameters, and might become an essential feature in the future.
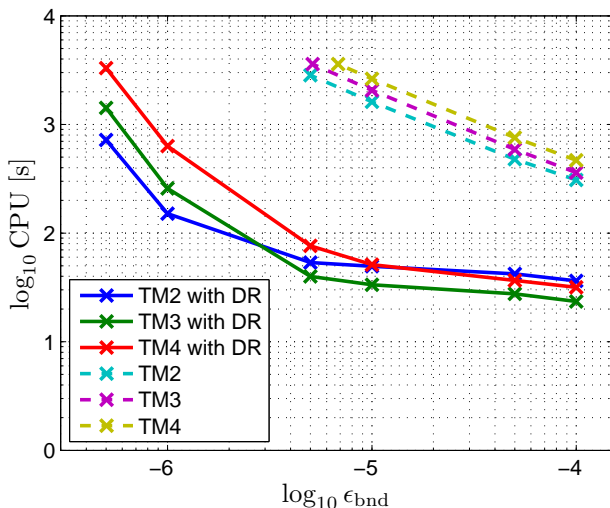

Fig. 4. Comparison of guaranteed parameter estimation algorithm for various orders of Taylor models and with or without the use of domain reduction (DR). CPU time vs. convergence threshold.

This improvement is reflected in Fig. 4, which compares the performance of Taylor models of different orders $q = 2, 3, 4$, with or without the use of domain reduction, in terms of CPU time as a function of the termination tolerance ϵ_{bnd}. The CPU times for these algorithmic variants is seen to follow the same trend as in the first case study. In particular, the computation of tight set approximations within a reasonable time is only possible here when domain reduction is applied.

5. CONCLUSIONS

This paper was concerned with the problem of guaranteed parameter estimation, which seeks for all parameter values of a dynamic model that are consistent with some experimental data, within specified error bounds. Set-inversion techniques based on exhaustive search are considered, with special emphasis on the use of optimization-based domain reduction techniques to enhance the algorithm convergence. Bound-reduction problems in the form of LPs are constructed from the polyhedral relaxation of Taylor models enclosing the predicted outputs. The potential of this approach, in terms of reducing the number of iterations and the CPU time, is illustrated with two simple case studies. The results suggest that further improvements could be obtained by switching off domain reduction during the algorithm, as well as by extending the domain reduction procedure to encompass affine cuts that better approximate the actual parameter set.

REFERENCES

A. Bompadre, A. Mitsos, and B. Chachuat. Convergence analysis of Taylor and McCormick-Taylor models. *Journal of Global Optimization*, in press, 2012. doi: 10.1007/s10898-012-9998-9.

Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2012. URL http://www.gurobi.com.

L. Jaulin and E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4):1053–1064, 1993.

L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer Verlag, 2001.

M. Kieffer and E. Walter. Guaranteed estimation of the parameters of nonlinear continuous-time models: contributions of interval analysis. *International Journal of Adaptive Control and Signal Processing*, 25(3):191 – 207, 2011.

Y. Lin and M. A. Stadtherr. Validated solutions of initial value problems for parametric ODEs. *Applied Numerical Mathematics*, 57(10):1145–1162, 2007a.

Y. Lin and M. A. Stadtherr. Guaranteed state and parameter estimation for nonlinear continuous-time systems with bounded-error measurements. *Industrial & Engineering Chemistry Research*, 46(22):7198–7207, 2007b.

R. Paulen, M. Villanueva, M. Fikar, and B. Chachuat. Guaranteed parameter estimation in nonlinear dynamic systems using improved bounding techniques. In *European Control Conference*, 2013. submitted.

A. M. Sahlodin. *Global Optimization of Dynamic Process Systems using Complete Search Methods*. PhD thesis, McMaster University, December 2012.

A. M. Sahlodin and B. Chachuat. Convex/concave relaxations of parametric ODEs using Taylor models. *Computers & Chemical Engineering*, 35(5):844 – 857, 2011.

M. Tawarmalani and N. V. Sahinidis. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, 99(3):563–591, 2004.