LINEAR-MODEL CASE

Norton, John
Department of Electronic, Electrical and Computer Engineering, School of Engineering, University of Birmingham, UK

Keywords: Bound-based identification, set-membership identification, parameter bounding, unknown but bounded uncertainty, bounded noise, regression models, parameter estimation, ellipsoids, polytopes, algorithms.

Contents

1. Bounding a linear model: the simplest case
2. Computation of the exact feasible set
3. Approximate parameter bounding
   3.1. Limited-complexity polytopes
   3.2. Ellipsoidal bounding
   3.3. Box bounding
   3.4. Parallelootope bounding
   3.5. Hybrid algorithms
4. Parameter bounding with unknown output-error bound
5. Parameter bounding with uncertain explanatory-variables vector
6. Clashes and outliers
7. Parameter bounds for time-varying linear systems
   7.1. Heuristic recursive bounding of time-varying parameters using ellipsoids
   7.2. Bounding of time-varying parameters treated as state variables
8. Conclusions
Bibliography
Biographical Sketch

Summary

This article presents a selection of techniques for computation of bounds on the parameters of a system model that is linear in the parameters and has specified bounds on the errors between the model output and observations of the system output. Exact and approximate parameter bounds are considered, and important special situations are described: parameter bounding when the output-error bound or the explanatory variables are uncertain, when the data clash with the model specification, and when the parameters vary with time.

1. Bounding a linear model: the simplest case

Bounding uses measurements from a system to reduce uncertainty in the unknowns in a model of the system. It aims to find the feasible set of all values of the unknowns that are consistent, in some precisely defined way, with all the measurements. First consider the simplest possible case, where

- the unknowns are constant parameters in the model,
• the relation between the parameters and observations is linear
• parameter values are consistent with the measurements if, for the same input values, they make the model output match the corresponding output observations to within a specified tolerance.

The linear model

\[ y_k = f_k^T \theta + e_k \]  

relates the \( k \)th scalar observation \( y_k \in \mathbb{R} \) to a vector \( \theta \in \mathbb{R}^n \) of unknown parameters, through known \( f_k \). The model may or may not be dynamical. For a static system, \( f_k \) might consist of simultaneous samples of \( n \) distinct system inputs, while for a dynamical system it might consist of \( n \) successive samples of a single input leading up to time \( k \), with the parameters in \( \theta \) comprising the unit-pulse response. [Note that the independent variable indexed by \( k \) need not be time, nor need the observations be at equal intervals in the independent variable. If, for instance, a vector of \( m \) output observations is taken at each point in time, we can treat the \( i \)th vector observation as \( y_{(i-1)m+1} \) to \( y_{im} \), with a different \( f \) for each of the \( m \) output variables].

At each \( k \), we require the model-output error \( e_k \) to be between \( \pm e_k \) (symmetrical about zero, without loss of generality). Thus to be consistent with observation \( y_k \), \( \theta \) must satisfy

\[ y_k - e_k \leq f_k^T \theta, \quad f_k^T \theta \leq y_k + e_k \]  

Geometrically, (2) requires \( \theta \) to be in both of two half spaces, bounded by hyperplanes with common normal \( f_k \). That is, \( \theta \) must be in the strip

\[ S_k = \left\{ \theta : y_k - e_k \leq f_k^T \theta \leq y_k + e_k \right\} \]  

between two parallel hyperplanes \( 2e_k / \|f_k\| \) apart. To be consistent with a set of observations \( y_k, \quad k = 1, 2, \ldots, N \), the parameters must be in the intersection of \( N \) strips, as in Figure 1, so the feasible set for \( \theta \) is

\[ \mathcal{P}_N = \left\{ \theta : \theta \in S_k, \quad k = 1, 2, \ldots, N \right\} = \bigcap_{k=1}^{N} S_k \]  

which is a polytope, convex and compact so long as \( n \) of the normals \( f_k, \quad k = 1, 2, \ldots, N \) are linearly independent.
So far, the picture is very simple. The boundary of the feasible set consists of those segments of the $2N$ hyperplanes (2) for $k = 1, 2, ..., N$ which are active bounds. Often $\theta$ is also subject to linear prior bounds, which are likely to come in parallel pairs, e.g. when each parameter can be assumed to be in a known range, confining $\theta$ to a box. Any unpaired linear prior bound can be treated as having a partner distant enough to be inactive. Thus linear prior bounds do not alter the picture but merely increase the effective $N$.

![Polytope feasible set with boundary-\(1\), formed by intersection of strips defined by hyperplane bounds due to three observations, for \(n = 2\).](image)

Figure 1. Polytope feasible set with boundary-\(1\), formed by intersection of strips defined by hyperplane bounds due to three observations, for \(n = 2\).

The feasibility of any particular value of $\theta$ can be tested simply by checking whether (2) is true for all the $k$’s; the smallest output-error tolerance $\varepsilon_k$ for which $\theta$ remains feasible is just the largest $\varepsilon_k$ yielded by $\theta$ in (1). However, explicit identification of the feasible set $\mathcal{P}_N$ may be difficult when $N$ is large, even if $n$ is quite small, because much computing may be necessary to establish which of the $2N$ hyperplane bounds are active as the faces of $\mathcal{P}_N$. The next two sections will address this problem.

Several useful generalizations of the linear parameter-bounding problem so far outlined will be considered in later sections: allowing unknown $\varepsilon_k$, allowing the parameters $\theta$ to vary with $k$, dealing with uncertain $f_k$, and coping with bad data.

### 2. Computation of the exact feasible set

Polytope $\mathcal{P}_N$ has at most $2N$ hyperplane faces, given by (2). They are conveniently rewritten as
Any one of them, say \( h_j^T \theta \leq z_j \), is active if and only if it contains the extremal points of \( P_N \) in the direction of \( h_j \). A linear programming solution maximizing \( h_j^T \theta \) subject to (5) can check this. All the active bounds can be identified by at most \( 2N \) such LP solutions, each with up to \( 2N - 1 \) constraints. The LP computing load increases with \( N \) and sooner or later becomes excessive if new observations continue to arrive. Instead of recomputing the feasible set from scratch each time a new observation is received, it is more economical to update a list of the active bounds of \( P_{N-1} \) to a list for \( P_N \) when observation \( N \) arrives, i.e., to compute the boundary of the feasible set recursively. Several closely related algorithms exist for recursive computation of \( P \) (Broman and Shensa, 1986; Walter and Piet-Lahanier, 1989; Mo and Norton, 1990). They differ in what details of the polytope they store and update. A basic version will now be described.

A non-minimal description of the polytope is stored, to economize in the computing effort required to update the description. It consists of a list of vertices and, for each vertex, two lists, of its supporting hyperplanes (those intersecting at it) and its adjacent vertices. To incorporate a new hyperplane bound \( h_j^T \theta \leq z_j \) due to a new observation, the updating procedure is:

- test each vertex \( v_i \) to see whether it is cut off by the hyperplane, i.e., whether \( h_j^T v_i > z_j \).
- if all vertices are cut off, the feasible set becomes empty (no parameter values remain feasible) so updating stops.
- if no vertex is cut off, no updating is necessary: the new hyperplane is redundant.
- for each vertex \( v_i \) not cut off, test all vertices on its adjacent-vertex list to see which are cut off.
- for each pair of adjacent vertices, \( v_i \) not cut off and \( v_l \) cut off, create a new vertex on the edge joining them, at \( v = (1 - \lambda)v_i + \lambda v_l \) where
  \[
  \lambda = \frac{h_j^T v_i - z_j}{h_j^T (v_i - v_l)}
  \]
- create the supporting-hyperplane list for each new vertex, consisting of the new hyperplane and those hyperplanes common to the lists of vertices \( v_i, v_l \).
- for retained vertices, update their adjacent-vertex lists by replacing cut-off...
vertices by the corresponding new ones

- for each new vertex \( v \), create an adjacent-vertex list consisting of the retained vertex \( v_i \) and all new vertices which share at least \( n - 1 \) supporting hyperplanes with \( v \).

Often relatively few of the \( 2N \) hyperplanes contribute to the boundary of the feasible set, many being redundant. Even so, an excessive computing load is likely to be incurred, particularly if the number of dimensions is large, because the numbers of vertices and edges of the polytope may become very large as more observations arrive. At some point the exact feasible set can no longer be computed and we have to resort to an approximate version that is cheaper to update and less complicated to work with. The next section describes the most popular approximations.

### 3. Approximate parameter bounding

A number of algorithms to compute simpler approximations to the exact feasible set have been developed. They all employ outer-bounding approximations, conservative in the sense that no feasible value is excluded but some infeasible ones are included. All the approximations are of complexity independent of the number of observations processed. The approximating set may be a polytope with a fixed number of faces, an orthotope (box) with \( n \) mutually orthogonal pairs of parallel faces (Milanese and Belforte, 1982), a parallelotope with \( n \) generally non-orthogonal pairs of parallel faces (Chisci, Garulli and Zappa, 1996, Vicino and Zappa, 1996), or an ellipsoid (Schweppe, 1968, Fogel and Huang, 1982). An ellipsoidal set is described by

\[
\mathcal{E}_k = \left\{ \theta : (\theta - \hat{\theta}_k)^T P_k^{-1} (\theta - \hat{\theta}_k) \leq 1 \right\}
\]

where \( \hat{\theta}_k \) is the centre and positive-definite matrix \( P_k \) describes the size and shape.

Of these approximations, a box aligned with the coordinate axes has the simplest description (by \( 2n \) numbers) but is a poor fit if the feasible set is narrow in a direction not close to an axis direction; a box not aligned with the axes requires \( n(n + 3)/2 \) numbers to specify it and can fit many convex, near-symmetrical sets well; an ellipsoid also requires \( n(n + 3)/2 \) numbers and also often fits convex, near-symmetrical sets well; a parallelotope is described by \( n(n + 1) \) numbers and is thus more complicated but can fit a somewhat wider range of sets well. The nature of the updating for each type of approximation is discussed below. Batch processing (all together) and recursive processing (one at a time) of the observations will be considered. In every case, the tightest computable bounds are required, giving the approximating object with, for instance, smallest volume.

#### 3.1. Limited-complexity polytopes

A bounding polytope defined by a fixed, restricted number \( n_p \geq n + 1 \) of hyperplanes can be computed recursively using the algorithm given in Section 2, together with a
criterion for discarding one of the \( n_p + 1 \) bounds in play once a new observation-induced bound is added. To minimize the volume of the updated polytope, the volumes of the trial polytopes found by deleting each bound in turn can be compared, in principle. In practice, the volumes are easily computed for \( n_p = n + 1 \) (i.e. updating a simplex) but heavy to compute for larger \( n_p \) (Lassere, 1983). Instead, Piet-Lahanier and Walter (1993, 1996) discard the bound farthest from the Chebyshev centre of the \((n_p + 1)\)-face polytope, while Broman and Shensa (1990), in a slightly different formulation, choose among new bounds cutting an existing polytope by maximizing the depth of the cut, as measured by the largest of the distances from the new bound to the excluded vertices. In both cases the computation is simple provided the vertices of the existing polytope have been recorded: each element of the Chebyshev centre is the mean of the extrema in the corresponding coordinate direction, and the distance of a point \( \bar{\Theta} \) from \( h^T \Theta = z \) is \( \| h^T \bar{\Theta} - z \| / \| h \| \).


process noise and observation noise. A performance comparison of state bounding with Kalman filtering is also provided.

Maksarov D., and Norton J. P. (1996b) Tuning of noise bounds in state bounding. Proc. CES’96 IMACS Multiconference, Symp. on Modeling, Analysis and Simulation, Lille, France, 2, 837-842. [Tackles the problem of adapting the specified noise bounds to match observed behaviour, analogously to adaptive Kalman filtering].


Pronzato L., and Walter E. (1990) Experiment design for bounded-error models. Math. & Comput. In Simulation 32, 571-584. [One of the motives for bounded-error formulation of identification is to avoid probabilistic assumptions about small data sets; for such sets, experiment design may be crucial in delivering useable results].
Pronzato L., and Walter E. (1994) Minimum-volume ellipsoids containing compact sets: application to parameter bounding. *Automatica* **30**, 1731-1739. [Computation of the smallest ellipsoidal outer approximation to boundary of the exact polytope feasible set of a linear-in-parameters model. This ellipsoid is distinct from (and generally smaller than) that found by a sequence of updates, each yielding the smallest ellipsoid for that update in isolation].


Veres S. M., and Norton J. P. (1991b) Parameter-bounding algorithms for linear errors-in-variables models, *9th IFAC/IFORS Symp. on Identification & System Parameter Estimation*, Budapest, 1038-1043. [*] [Discusses parameter bounding for linear models when the input, as well as the output, is observed with bounded error. Distinguishes static and dynamic cases and shows how non-linear parameter bounds arise as the envelope of linear bounds].


Walter E., and Piet-Lahanier L. (1991) Recursive robust minimax estimation for models linear in their parameters, *9th IFAC/IFORS Symp. on Identification & System Parameter Estimation*, Budapest, 763-768. [*] [Points out that, as a linear, additive-error model is linear in output error as well as parameters, the minimax-output-error parameter estimate is the lowest vertex of the feasible set in the space of parameters and output error].


Biographical Sketch

John Norton graduated in Mechanical Sciences at Cambridge, then worked in digital design in industry. Following a PhD in control of DC power systems at Imperial College, London, he worked as a research fellow at the National Physical Laboratory and Warren Spring Laboratory from 1967 to 1971, working mainly on modeling aspects of control problems in the steel and paper industries. Between 1971 and 1979 he was in the Dept. of Electrical Engineering, University of Tasmania, Australia. During that time he developed new methods of handling complex dynamics by estimating linear, time-varying models. Since 1979 he has been at the University of Birmingham, where he has been Professor since 1991. Work on biomedical modeling led to an early involvement in bound-based methods for identifying parametric models of dynamical systems, and since the mid-1980s he has published many papers in that area. He has also been active in state-estimation techniques for target tracking, latterly using Monte Carlo Bayesian estimation. Other recent work has been in two areas: methods for incorporating collateral information into parameter estimation and state estimation, and techniques for exploring the characteristics of complex digital simulation models (in collaboration with the Centre for Resource & Environmental Studies at the Australian National University). He has published over 100 papers, the book An Introduction to Identification, Academic Press, the co-edited collection of papers Bounding Approaches to System Identification, Plenum Press and contributions to numerous books and collections of papers on control and signal-processing topics. He was Editor for Adaptive Control, Int. J. of Adaptive Control & Signal Processing 1992-95 and editorial board member of IJACSP 1992-2001, J. of System & Control Engineering 1998-2001, and Environmental Modeling & Software 1997-present. He has been a member of the IFAC Technical Committee on Modeling, Identification and Signal Processing since 1995.