LAPACK-Based Software for Subspace Identification

Vasile Sima* Andras Varga† Michel Verhaegen‡

June 28.1996

1 Introduction

Computer-aided control system design is usually based on linear time-invariant (LTI) state space models. Surprisingly enough, multivariable state space system identification has only recently become a topic of intense research. In contrast with classical input-output (I/O) identification approaches, the newly proposed Subspace Model Identification (SMI) techniques essentially find a state sequence, or a column space approximation, and then determine the system matrices by solving some least squares problems. These techniques have promising advantages over the classical ones. One advantage is that there is no need for parameterizations, which are notoriously difficult to choose or analyse, or could led to ill-conditioned problems. Another advantage is that robust numerical linear algebra techniques, like QR factorization and singular value decomposition (SVD), can be largely used; this contrasts with the iterative optimization schemes required in the parametric model identification approach, documented e.g. in [1]. The attractiveness of SMI techniques is further increased by the small number of parameters (essentially only one) to be selected for determining the model structure, without any restriction on model generality. See, for instance, [2]–[7] for a further discussion of the SMI features.

MATLAB codes based on SMI algorithms have been recently developed, e.g. in [8]. For efficiency, it is clearly useful to implement some algorithms in Fortran, using the state-of-the-art, public-domain linear algebra package LAPACK. This allows to exploit the potential parallelism of many modern computer architectures.

This paper briefly describes the basic approaches and LAPACK-based Fortran routines for multivariable system identification by subspace techniques. The developed software package [9] includes implementations of two classes of SMI techniques to identify an LTI system. The first one is referred to as the State Intersection (SI) class of SMI techniques. The implemented method, described in [2], is known as the N4SID (Numerical algorithm for Subspace State Space System IDentification) approach. The second one is referred to as the Multivariable Output Error state SPace (MOESP) class of techniques. The implemented MOESP methods are described in [3], [5] and [6]. Deterministic and combined deterministic-stochastic

*Research Institute for Informatics, Bd. Mareşal Averescu, Nr. 8–10, 71316 Bucharest 1, Romania. e-mail: vsima@roearn.ici.ro

†DLR-Oberpfaffenhofen, German Aerospace Research Establishment, Institute for Robotics and System Dynamics, D-82234 Weßling, FRG. e-mail: Andreas.Varga@dlr.de.

‡Delft University of Technology, Dept. of Electrical Engineering, P.O.Box 5031, NL-2600 GA Delft, The Netherlands. e-mail: verhaege@dutentb.et.tudelft.nl

identification problems are dealt with. A state space model is computed from I/O data sequences. It is possible to handle multiple I/O sequences, each one being collected by a possibly independent identification experiment. Sequential processing of large data sets is provided as an option. See [9], for further details. In addition to the main identification facilities, auxiliary routines are available to detect dead-times and shift the I/O signals appropriately, estimate initial states, identify systems operating in closed-loop, or simulate discrete-time LTI systems.

The theoretical algorithms and their MATLAB realizations have been largely reorganized. For instance, the N4SID algorithm makes use of the inverse of a part of the triangular factor of an RQ factorization of a Hankel-like matrix constructed from I/O sequences. (This is needed before computing the SVD that gives the system order and then the quadruple of system matrices.) The MATLAB implementations solve several standard least squares problems to obtain the associated, so-called "oblique projection". While the new, LAPACK-based implementation only involves some orthogonal transformations for obtaining the various projections and residuals.

One objective of the Fortran implementation has been to use the LAPACK routines as much as possible. Preference has generally been given to the block variants and the Level III BLAS codes, which are responsible for efficient use of parallelism. Advantage was taken of the problem structure, whenever possible. For instance, the calculations have been organized such that singular value decompositions were required for triangular matrices only, and a dedicated, very compact and efficient routine has been written for that purpose. Moreover, both in theory and in the MATLAB implementations, the computation of the system matrices involves a very large matrix. Fortunately, this matrix can be brought to a special block-triangular structure. A structure-exploiting QR row compression scheme has been devised, reducing the storage requirements and computational effort.

Another related objective has been to allow performing the computations with as reduced memory requirements as possible, but without sacrificing the speed, when enough memory is available. For instance, there is an option for sequential calculation of the triangular factor in QR factorization of the Hankel-like matrix. The user can specify that I/O data are entered in batches (independent or not). Moreover, when there is not enough working memory for processing a given data batch, the codes automatically perform an inner sequential processing of that batch, to accommodate with the available memory space. No provisions for using the displacement rank techniques (based on the block-Hankel structure) are provided in the current implementations.

2 Basic Approaches

The basic LTI state space models considered are described by

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k, \\ y_k &= Cx_k + Du_k + v_k, \end{aligned} \tag{1}$$

where x_k is the n-dimensional state vector at time k, u_k is the m-dimensional input (control) vector, y_k is the ℓ-dimensional output vector, $\{w_k\}$ and $\{v_k\}$ are state and output disturbance or noise sequences, and A, B, C, and D are real matrices of appropriate dimensions. The system order, n, and the quadruple of system matrices (A, B, C, D) are not known; the only available information is given by an upper bound, s, on n, and by the input and output data sequences, $\{u_k\}$ and $\{y_k\}$, $k = 1, \ldots t$.

2.1 The State Intersection Approach

The main feature of the SI class of SMI techniques is the determination of the state sequence of the LTI system to be identified, or of an observer to reconstruct its state sequence, via the intersection of the row spaces of the Hankel-like matrices constructed from "past" and "future" I/O data. The basic idea was first described in [4]. An extension of this idea led to the N4SID algorithm developed in [2]. This algorithm identifies LTI state space models in the so-called innovation form, where the additive disturbance w_k in (1) is generated by an innovation model of the form $w_k = Kv_k$, $\{v_k\}$ is a zero-mean white noise sequence and $\{u_k\}$ is a deterministic input sequence (perfectly known to the user). Both the basic and extended variants of this class produce statistically consistent and efficient estimates when certain assumptions hold.

2.2 The MOESP Approach

The main feature of this class of SMI techniques is the determination of an extended observability matrix, Γ_s, of the deterministic part of the model (1). Note that the upper bound, s, on n, is used. The basic idea was first described in [5] and [6]. The simplest algorithm derived from this idea, called the ordinary MOESP scheme, allows to identify, in a statistically consistent and efficient way, LTI systems that can be described by (1), with $w_k \equiv 0$ and $\{v_k\}$ a zero-mean white noise sequence, independent of the input.

Extensions based on using past input quantities and/or reconstructed state variables as instrumental variables have been proposed, allowing to consistently identify a model in (1), for $w_k \equiv 0$, and $\{v_k\}$ a zero-mean arbitrary stochastic disturbance, independent of the input. The price paid for this increased applicability is that the estimates are no longer efficient. These variants are referred to as the PI or RS schemes, when past inputs or reconstructed state variables are used as instrumental variables, respectively.

When the additive disturbance w_k in (1) is generated by an innovation model of the form $w_k = Kv_k$, and v_k is a zero-mean white noise independent of the input, it is again possible to obtain both consistent and efficient estimates when, besides the past input, past output quantities are also used as instrumental variables. The scheme derived from this extension of ordinary MOESP scheme is known as the PO scheme [3].

For the deterministic-stochastic identification problem, it is possible to estimate the deterministic part by using the PO scheme, and the stochastic part, that is the noise covariance matrices, by using the SI approach [2]. Further, the corresponding Kalman gain K can be computed to allow the design of state observers [3].

The algorithms belonging to the MOESP class have the striking feature of being highly streamlined, in the sense that the sequence of computations performed by these schemes is almost independent from the type of problems to be analyzed. This has led to highly modular implementations of the algorithms within this class.

3 LAPACK-Based Software

The collection of implemented subroutines forms a library called RASP-IDENT [9] which is part of the RASP package [13]. All implementations were done according to the RASP-SLICOT mutual compatibility concept introduced in [14]. The RASP-IDENT library consists of six driver routines wich are compatible with the RASP implementation and documentation standards, five auxiliary routines for system identification, ten SLICOT compatible driver routines, and nineteen SLICOT compatible computational and auxiliary routines.

3.1 Driver Routines

The following driver routines are available for MOESP class of SMI:

RPIMOE computes consistent and statistically efficient estimates of the matrices (A, B, C, D) in (1) when $w \equiv 0$ and v is zero-mean white noise independent of the input, using the ordinary MOESP scheme.

RPIMPI computes consistent estimates of the matrices (A, B, C, D) in (1) when $w \equiv 0$ and v is zero-mean, but of arbitrary statistical color, using the ordinary MOESP scheme extended with instrumental variables based on past input quantities.

RPIMPO computes consistent and statistically efficient estimates of the matrices (A, B, C, D) in (1) when w is generated by an innovation model and v is zero-mean white noise independent of the input, using the ordinary MOESP scheme extended with instrumental variables based on past input and past output quantities.

RPIMRS computes consistent estimates of the matrices (A, B, C, D) in (1) when $w \equiv 0$ and v is zero-mean, but of arbitrary statistical color, using the ordinary MOESP scheme extended with instrumental variables based on reconstructed state variables.

RPIMKG performs the computations in RPIMPO routine, but also estimates the noise covariance matrices and the Kalman gain (to allow the design of a state observer as a state predictor), using the SI approach.

The following driver subroutine is available for the SI class of SMI:

RPIMN4 computes consistent and statistically efficient estimates of the matrices (A, B, C, D) in (1) when w is generated by an innovation model and v is a zero-mean white noise sequence, using the basic N4SID algorithm. This routine also estimates the noise covariance matrices and the Kalman gain (to allow the design of a state observer as a state predictor).

All driver routines have an option for sequential I/O data processing.

3.2 Computational and Auxiliary Routines

To validate and extend the use of various subspace identification schemes, a number of computational and auxiliary routines have been implemented. The main ones are listed below:

IMMPID estimates a finite set of Markov parameters from I/O data (no constraints on input sequence to be white noise, or initial conditions be zero).

IMSHFT shifts the I/O data sequences to compensate for the dead-times in the input vector components.

IMICID estimates the initial state x_0 of a LTI system, given the matrix quadruple (A, B, C, D) and the input and output trajectories of the system.

IMCLID computes the parameters of a LTI system operating in closed-loop with a LTI controller.

RPIMIU computes the output trajectory of a LTI discrete-time system, given the input trajectory, the initial state and/or the state and output disturbances (or noise) trajectories.

4 Conclusions

LAPACK-based Fortran software for multivariable system identification by subspace techniques have been briefly described. The underlying approaches seem very promising, and the software components proved robust, flexible, and easy of use, even when the available memory is quite limited. The software has been extensively tested; the results obtained using various algorithms (including their MATLAB versions) have been compared. New versions of the codes, with improved modularity and efficiency, are under development. Implementation of new algorithms is also considered.

References

[1] L. Ljung, *System Identification Toolbox.* Natick, MA: The MathWorks, Inc., 1992.

[2] P. Van Overschee, and B. De Moor, *N4SID: Two subspace algorithms for the identification of combined deterministic-stochastic systems*, Automatica, vol. 30, no. 1, pp. 75–93, 1994.

[3] M. Verhaegen, *Identification of the deterministic part of MIMO state space models given in innovations form from input-output data*, Automatica, vol. 30, no. 1, pp. 61–74, 1994.

[4] M. Moonen, B. De Moor, L. Vandenberghe, and J. Vandewalle, *On- and off-line identification of linear state-space models*, Int. J. Control, vol. 49, no. 1, pp. 219–232, 1989.

[5] M. Verhaegen, and P. Dewilde, *Subspace Model Identification. Part 1: The output-error state- space model identification class of algorithms*, Int. J. Control, vol. 56, no. 5, pp. 1187–1210, 1992.

[6] M. Verhaegen, *Subspace Model Identification. Part 3: Analysis of the ordinary output-error state-space model identification algorithm*, Int. J. Control, vol. 58, no. 3, pp. 555–586, 1993.

[7] P. Van Overschee, *Subspace Identification : Theory – Implementation – Applications*, Katholieke Universiteit Leuven, Ph. D. Thesis, 1995.

[8] M. Verhaegen, *State Space Model Identification Toolbox*, Delft University of Technology, Tech. Rep., Nov. 1993.

[9] V. Sima, and A. Varga, *RASP–IDENT Subspace Model Identification Programs*, DLR – Oberpfaffenhofen, Tech. Rep. TR R888–94, Oct. 1994.

[10] S. Van Huffel, and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis*. Philadelphia, PA: SIAM, 1991.

[11] M. Verhaegen, and A. Varga, *Some Experience with the MOESP Class of Subspace Model Identification Methods in Identifying the BO105 Helicopter*, DLR – Oberpfaffenhofen, Tech. Rep. TR R165–94, June 1994.

[12] V. Sima, *Algorithms and LAPACK-Based Software for Subspace Identification*, Proceedings of The IEEE International Symposium on Computer-Aided Control System Design—CACSD'96, September 15–18, 1996, Dearborn, Michigan, U.S.A. (to appear).

[13] G. Grübel, and H.-D. Joos, *RASP and RSYST - Two Complementary Program Libraries for Concurrent Control Engineering*, Prepr. 5th IFAC/IMACS Symp. CADCS'91, Swansea, UK, Pergamon Press, Oxford, pp. 101-106, 1991.

[14] G. Grübel, A. Varga, A. van den Boom, and A. J. Geurts, *Towards a coordinated development of numerical CACSD software: the RASP/SLICOT compatibility concept*, Prepr. of IEEE/IFAC Symp. CACSD'94, Tucson, Arizona, 1994