Comparative study between three subspace identification algorithms

Wouter Favoreel, Sabine Van Huffel, Bart De Moor*,
Vasile Sima†,
Michel Verhaegen‡

Abstract

Since the appearance of subspace system identification in literature different subspace algorithms are available in practice. The most commonly used algorithms are CVA, N4SID and MOESP. Although they all follow the same philosophy, their practical implementation can differ considerably. In the present paper it is our intention to make a direct comparison between these different subspace system identification algorithms. The comparison is made on the basis of 15 publicly available practical data sets on which the algorithms have been applied. The quality measure for the identification we considered is the computational complexity of the method on the one hand and the prediction/simulation error on the other hand. The influence of the identification parameters on the identification quality is studied and analyzed.

1 Introduction

Subspace identification is the name for a general class of relatively new system identification methods. Although the origins of subspace identification can be traced back to different fields of system theory, we can state that the first paper on subspace system identification appeared at the end of the eighties [1]. We give the following representative references to the subspace identification literature [2], [3], [4], [5]. Many papers appeared where subspace identification was successfully applied [6]. One of the main reasons that made subspace identification so popular is that they have some very interesting advantages when compared to other system identification methods. For instance they are very well suited for the identification of systems with multiple inputs and multiple outputs. They are essentially non-iterative (so no convergence problems arise), fast and numerically robust (since only based on numerically stable techniques of linear algebra). Especially in the field of process industry the method is very popular.

Since the appearance of subspace system identification in literature different subspace algorithms are available in practice. The most commonly used algorithms are N4SID [3], CVA [2] and MOESP [4]. Although they all follow the same philosophy, their practical implementation can differ considerably. In the present paper it is our intention to make a direct comparison between these different subspace system identification algorithms. The comparison is made on the basis of 15 publicly available practical data sets to which the different subspace algorithms are applied. The quality measure for the identification we consider is the computational complexity of the method on the one hand and the prediction/simulation error on the other hand. The influence of the identification parameters on the identification quality is studied and analyzed.

The current research was done in the framework of the European BRITE/EURAM thematic networks project NICONET coordinated by WGS (Working Group on Software http://www.win.tue.nl/wgs). The main objective of WGS is to develope a numerically reliable and freely available software library for systems and control theory, called SLICOT [7]. Another very important objective of this library is to make links with the area of industrial applications. Such software is an essential ingredient in modern computer aided control system design (CACSD) and thus for the future of high-tech European industry. In particular, NICONET wants to include subspace identification software into SLICOT. For more information we refer to the web site: http://www.win.tue.nl/wgs/niconet.html.

*Katholieke Universiteit Leuven - ESAT/SISTA, K. Mercierlaan 94, 3001 Leuven, Belgium, Tel: +32-16-321808, Fax: +32-16-321970, URL: http://www.esat.kuleuven.ac.be/sista, email: wouter.favoreel@esat.kuleuven.ac.be

†Research Institute for Informatics, Bd. Maresal Averescu Nr. 8–10, 71316 Bucharest 1, Romania, Tel: +401-6656060/ext 156, Fax: +401-3128539, URL: http://www.ici.ro/homepages/vsima.htm, email: vsima@u3.ici.ro

‡Delft University of Technology, Department of Information Technology Systems, Control Engineering Group, P.O. Box 5031, 2600 GA Delft, The Netherlands, Tel: +31-15-2786152, Fax: +31-15-2786679, URL: http://lcewww.et.tudelft.nl/ verhaege/ email: M.Verhaegen@its.tudelft.nl

2 Subspace identification

The linear subspace system identification problem can be stated as:

Given measurements of the inputs u_k and outputs y_k of an unknown linear time-invariant system of the form:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Ke_k, \\ y_k &= Cx_k + Du_k + e_k \end{aligned}$$

with the white Gaussian noise e_k having the covariance matrix $\mathbf{E}[e_p e_q^T] = S\delta_{pq}$, **find** an estimate of the system matrices A, B, C, D and the noise related matrices S and K.

The usual steps in a subspace identification algorithm are the following:

1. Construct a data block Hankel matrix from the measured input and output data.
2. Perform a well-defined QR-decomposition with this matrix.
3. Perform an SVD on the low dimensional rank-deficient R-component to find the extended observability matrix and/or the Kalman filter state sequence of the system.

Although these three steps are common to most of the subspace identification algorithms, the way they are finally implemented in practice can differ considerably. The next steps in the subspace identification algorithms, i.e. that of finding the actual system matrices A, B, C, D, K and S, can be completely different according to the method. For instance certain methods use the observability matrix of the system to find the system matrices whereas other methods use an estimate of the state sequence (see [5] for further details). The three subspace identification algorithms that we will consider in this paper are:

CVA **C**anonical **V**ariate **A**nalysis [8], [2].

N4SID **N**umerical algorithm for **S**ubspace **S**tate **S**pace **S**ystem **ID**entification. The algorithm we used is the `subid.m` matlab function that comes with [3].

MOESP **M**ultivariable **O**utput **E**rror **S**tate s**P**ace. The functions that have been used here are `dordpo.m`, `dmodpo.m` and `dac2bd.m` of the SMI-toolbox [9].

The purpose of the present paper is **not** to analyze the implementational differences between those methods. What we tried to do is to test and compare the performance of the algorithms, according to certain criteria that will be explained in the next section. We used the Matlab-code as provided by the authors without making any modifications.

3 Description of the benchmark

In order to increase the accessibility and the reproducibility of the results presented in this paper, all the data and the Matlab-code that have been used are made publicly available on the DAISY internet site [10]:

http://www.esat.kuleuven.ac.be/sista/daisy

Every data set available on that site is labeled by a 5 digit number. The list of data sets is:

1. **ball & beam** Data of the ball and beam practicum at ESAT-SISTA [96-004].
2. **cd player arm** Data from the mechanical construction of a CD player arm [96-007].
3. **distillation** Data of a simulation related to the identification of an ethan-ethylene distillation column [96-001].
4. **dryer 1** Laboratory setup acting like a hair dryer [96-006].
5. **dryer 2** Data from an industrial dryer (by Cambridge Control Ltd) [96-016].
6. **evaporator** A four-stage evaporator to reduce the water content of a product, for example milk [96-010].
7. **heat exchanger** A liquid-saturated steam heat exchanger, where water is heated by pressurized saturated steam through a copper tube [97-002].
8. **flexible structure** Experiment on a steel subframe flexible structure performed at LMS-International, Leuven-Belgium [96-013].
9. **glass furnace** Data of a glassfurnace (Philips) [96-002].
10. **pH data** Simulation data of a pH neutralization process in a constant volume stirring tank [98-002].
11. **powerplant** Data of a power plant (Pont-sur-Sambre (France)) of 120 MW [96-003].
12. **robot arm** Data from a flexible robot arm [96-009].
13. **steam generator** Steam generator at Abbott Power Plant in Champaign IL [98-003].
14. **thermic wall** Heat flow density through a two layer wall (brick and insulation layer) [96-011].

15. **winding process** Test setup of an industrial winding process [97-003].

The different algorithms have been implemented under Matlab 5.1 on a Sun Ultra Sparc 5 workstation (266 MHz) as an m-file of the following format:

$$[A, B, C, D, K] \quad = \quad \text{subspace}(u_{id}, y_{id}, i, n)$$

where u_{id} and y_{id} are the inputs and the outputs of the identification data set. The i-parameters is the number of block rows in the past and the future data block-Hankel matrices as defined in [3]. n is the chosen order of the system. "subspace" is the subspace identification algorithm (CVA, N4SID or MOESP). The number of block rows in the past and future matrices is supposed to be the same and equal to i. The outputs of the function are the system matrices A, B, C, D and the noise related matrix K.

A data set is always split up into an identification part and a validation part. With the identification set the model parameters are calculated while the validation set is used to find the prediction/simulation error. The parameters of the simulation for every data set are given in Table 1. The number of inputs and outputs are m and l respectively. id is the number of data points used for the identification whereas val is the length of the validation data set.

We considered the following factors to be important for the quality of a system identification method:

Computational complexity: quantified by the number of Mega flops required by the identification step in Matlab 5.1.

Prediction error: based on the validation data set and defined as follows:

$$\epsilon^p \quad \stackrel{\text{def}}{=} \quad \frac{1}{l} \sum_{q=1}^{l} \sqrt{\frac{\sum_{k=1}^{val} (y_{k,q} - y_{k,q}^p)^2}{\sum_{k=1}^{val} y_{k,q}^2}}$$

with $y_{k,q}$ the q-th output at time step k and y_k^p the one-step-ahead predicted output.

Simulation error: also based on the validation data set and defined as the prediction error but by replacing the one-step-ahead predicted output y_k^p by the simulated output y_k^s.

The above three quality criteria are calculated for a wide range of the identification parameters i and n. The chosen order n of the system must satisfy $n < il$.

	m	l	id	val
1.	1	1	1000	0
2.	2	2	1500	548
3.	1	2	3000	979
4.	1	1	750	250
5.	3	3	600	267
6.	3	3	5000	1305
7.	1	1	3000	1000
8.	2	1	6000	2523
9.	3	6	1000	247
10.	2	1	1500	501
11.	5	3	200	0
12.	1	1	800	224
13.	4	4	7000	2600
14.	2	1	1000	680
15.	5	2	1500	1000

Table 1: **Parameters of the different data sets.** m **is the number of inputs,** l **the number of inputs,** id **the number of data points used for the identification and** val **the number of data points used for validation of the obtained model. If** val **is zero, the data set was too short to be split up into an identification and a validation part. In that case, the identification data set was used for the validation of the model.**

4 Simulation results

For each data set we calculated the simulation/prediction error and computational load for different values of the number of block rows in the data Hankel matrices i and the order of the system n. It is then possible to represent these results as function of i and n in a three dimensional plot. Due to space limitations we only give these plots for data set number 14 (see Figures 1, 2 and 3). The interested reader can find the complete results in [11]. From these figures, the following conclusions can be drawn:

- The computational complexity of the CVA method is about 20 times smaller than for the MOESP method and 10 times smaller than the N4SID method. We presume that the reason for this mainly is due to the way the algorithm is implemented. Instead of doing a QR-decomposition on the data, the data is first reduced to the corresponding covariances. As a consequence the matrices that are involved are much smaller which results in a faster algorithm. Although we did not observe this experimentally, it is generally assumed that starting from covariances is less accurate than starting from the data. The reason for this is that the calculation of the covariances involves the multiplication of data matrices which squares up the condition number and may lead to an inevitable loss of numerical accuracy in case of ill-conditioning. Also interesting to note is that for the CVA algorithm the computational complexity increases more or less linearly with the order n whereas for the other two methods (and especially for MOESP) there seems to be an exponential increase. The difference between

Data set	CVA	N4SID	MOESP
1	2.4198	**2.3220**	2.4186
2	6.8151	**6.5376**	8.9553
3	**48.7454**	48.7784	49.1773
4	4.8265	4.7682	**4.7666**
5	11.5455	**10.7647**	12.6140
6	**20.0718**	20.1193	20.6335
7	**23.8987**	24.1221	24.2227
8	**11.3536**	12.2794	20.2570
9	17.0438	**13.6495**	14.4347
10	52.3222	**31.7654**	46.3633
11	6.8093	**4.6228**	5.0161
12	0.9642	0.1170	**0.0998**
13	**6.3111**	6.3273	9.2190
14	**14.6819**	14.8387	14.8210
15	**18.2243**	19.7667	19.4947

Table 2: **The minimum value of the prediction error over the different values of i and n for each data set. CVA and N4SID seem to perform slightly better than MOESP. However, no final conclusions may be drawn from this table.**

the computational complexity of MOESP and N4SID is essentially due to the algorithms used for the calculation of system matrices (`dac2bd.m` in the SMI toolbox). Previous comparisons of MOESP and N4SID approaches [12], using the Fortran implementations described in [13], revealed that they are computationally equivalent.

- Concerning the prediction errors and the simulation errors think that it is not possible to draw any hard conclusions from the present study. Perhaps one might say that CVA and N4SID have a smaller prediction error whereas MOESP has a smaller simulation error.

- Another important quality of an identification method is the stability of the prediction/simulation errors when the i and n parameters change. The CVA algorithm seems to be less sensitive to system orders n that are close to the number of block rows in the data matrices i.

The following tables give the overall results for the 15 data sets. In each table the minimum value of the corresponding quality criterion for the three algorithms is indicated in bold-face. Table 2 and 3 present the minimum value of the prediction and the simulation errors over i and n. The errors that are larger than 100% have been clipped at 100. Table 4 gives the computational load of the different subspace algorithms that correspond to the i and n values where the prediction error of the CVA algorithm is minimal.

Data set	CVA	N4SID	MOESP
1	100.0000	74.9221	**72.3515**
2	19.6774	23.5653	**18.0015**
3	55.6503	55.8385	**55.6116**
4	**7.5086**	7.8874	7.9014
5	**49.2288**	49.4671	50.3791
6	43.0268	42.9225	**37.4237**
7	32.5908	**32.5660**	32.8408
8	82.7149	77.6466	**75.9364**
9	43.2657	**38.0679**	41.6424
10	100.0000	100.0000	100.0000
11	16.3818	12.8543	**9.9283**
12	5.7773	**3.5258**	3.9485
13	24.8750	**24.4981**	28.6154
14	**14.8218**	14.9181	14.8399
15	**23.3648**	24.8119	24.3589

Table 3: **The minimum values of the simulation error over the different values of i and n for each data set. MOESP seems to perform slightly better than CVA and N4SID. However, no final conclusions may be drawn from this table.**

Data set	CVA	N4SID	MOESP
1	**1.1**	14.5	23.5
2	**4.7**	72.1	201.8
3	**1.9**	27.1	58.0
4	**1.0**	12.6	18.9
5	**6.1**	92.0	202.2
6	**8.5**	113.8	497.8
7	**3.0**	58.6	118.4
8	**4.8**	93.2	163.4
9	**1.5**	7.1	19.1
10	**4.6**	49.4	43.4
11	**0.5**	3.6	7.1
12	**2.5**	19.6	18.4
13	**13.5**	171.5	1541.7
14	**4.2**	44.2	60.4
15	**5.6**	67.1	154.7

Table 4: **The computational complexity (in number of Mega flops) corresponding to the value of i and n where the prediction error of the CVA algorithm is minimal. One can see the CVA algorithm is about a factor 20 faster than the MOESP algorithm. The N4SID in turn is twice as fast as the MOESP algorithm.**

Figure 1: **Computational complexity for data set 14 as a function of i and n. An interesting thing is that the computational load for the CVA algorithm increases linearly with the chosen order where as for the other two algorithms it seems to raise exponentially.**

Figure 2: **Prediction error of data set 14 as a function of i and n. The CVA algorithm seems to be less sensitive for values of the order n that are close to the values of the number i of block rows in the data matrices.**

Figure 3: **Simulation error of data set 14 as a function of i and n. Same remark as in Figure 2.**

5 Conclusions

In the present paper it was our intention to compare different subspace system identification methods with each other. The comparison was made by applying them, under the same conditions, to 15 different practical data sets. The quality criteria that were considered are computational complexity, prediction and simulation errors. Although the performance of the methods is very comparable, CVA and N4SID tend to have smaller prediction errors whereas MOESP has smaller simulation errors. Due to the way the algorithm is implemented the CVA algorithm has a much smaller computational complexity than the other two algorithms. It also seems to be less sensitive to the choice of the system order and the number of block rows in the data matrices.

6 Acknowledgements

We would like to thank Dietmar Bauer from the Technische Universität Wien, Bert Haverkamp from the Delft University of Technology and Peter Van Overschee from the Katholieke Universiteit Leuven, for providing us with the necessary Matlab-code.

Work supported by the **Flemish Government** (Administration of Science and Innovation (Concerted Research Action MIPS: *Model-based Information Processing Systems*, Bilateral International Collaboration: *Modeling and Identification of nonlinear systems*, IWT-Eureka SINOPSYS: *Model-based structural monitoring using in-operation system identification*), FWO-Vlaanderen: *Analysis and design of matrix algorithms for adaptive signal processing, system identification and control, based on concepts from continuous time system theory and differential geometry*, *Numerical algorithms for subspace system identification: Extension towards specific applications*, FWO-Onderzoeksgemeenschappen: *Identification and Control of Complex Systems*, *Advanced Numerical Methods for Mathematical Modeling*); **Belgian Federal Government**(Interuniversity Attraction Pole IUAP IV/02: *Modeling, Identification, Simulation and Control of Complex Systems*, Interuniversity Attraction Pole IUAP IV/24: IMechS: Intelligent Mechatronic Systems); **European Commission:** (Human Capital and Mobility: *SCIENCE-ERNSI: European Research Network for System Identification*; *NICONET:* Brite Euram Programme, Thematic networks BRRT-CT97-5040.)

References

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

[2] W. Larimore, "Canonical variate analysis in identification, filtering and adaptive control," in *Proc. of the 29th Conference on Decision and Control, CDC 90, Hawaii, US*, pp. 596–604, 1990.

[3] P. Van Overschee and B. De Moor, *Subspace identification for linear systems: theory, implementation, applications.* Dordrecht: Kluwer Academic Publishers, 1996.

[4] M. Verhaegen and P. Dewilde, "Subspace identification, part I: The output-error state space model identification class of algorithms," *Internat. J. Control*, vol. 56, pp. 1187–1210, 1992.

[5] B. De Moor, P. Van Overschee, and W. Favoreel, *Numerical algorithms for subspace state space system identification - An overview.* Birkhauser, 1998.

[6] W. Favoreel, B. De Moor, and P. Van Overschee, "Subspace state space system identification for industrial processes," in *Proc. of the 5th IFAC Symposium on Dynamics and Control of Process Systems, June 8-10, Corfu, Greece*, pp. 322–330, 1998.

[7] P. Benner, V. Mehrmann, V. Sima, S. Van Huffel, and A. Varga, *A Subroutine Library in Systems and Control Theory*, ch. Accepted for publication. Birkhäuser, 1998.

[8] K. Peternell, W. Scherrer, and M. Deistler, "Statistical analysis of novel subspace identification methods," *Signal Processing*, vol. 52, pp. 161–177, 1996.

[9] B. Haverkamp and M. Verhaegen, *SMI Toolbox: State space Model Identification software for multivariable dynamical systems.* Technical University Delft, 1997. http://lcewww.et.tudelft.nl/˜haver/smi.html.

[10] B. De Moor, P. De Gersem, B. De Schutter, and W. Favoreel, "Daisy: A database for identification of systems," *Journal A, Special Issue on CACSD (Computer Aided Control Systems Design)*, vol. 38, pp. 4–5, Sep. 1997. `http://www.esat.kuleuven.ac.be/sista/daisy`.

[11] W. Favoreel, B. De Moor, S. Van Huffel, V. Sima, and M. Verhaegen, "Benchmark for subspace system identification algorithms," Tech. Rep. 98-14, Katholieke Universiteit Leuven - ESAT/SISTA, 1998.

[12] V. Sima, "High-performance numerical software for control systems analysis and design, and subspace-based system identification. final report.," tech. rep., Katholieke Universiteit Leuven, Leuven, Belgium, March 1997. `ftp://wgs.esat.kuleuven.ac.be`, directory `/pub/WGS/REPORTS/rep97-2.ps.Z`.

[13] V. Sima, "Algorithms and lapack-based software for subspace identification," in *Proc. of The 1996 IEEE International Symposium on Computer-Aided Control System Design, September 15–18, Dearborn, MI, US*, pp. 182–187, 1996.