SLICOT Working Note 1998-6

Subspace Model Identification of Linear Systems in SLICOT *

Wouter Favoreel[†]

Vasile Sima[‡]

Sabine Van Huffel[§]

Michel Verhaegen[¶]

Bart De Moor[‖]

* This paper presents research results of the European Community BRITE-EURAM III Thematic Networks Programme NICONET (contract number BRRT-CT97-5040) and is distributed by the Working Group on Software WGS. *WGS secretariat*: Mrs. Ida Tassens, ESAT - Katholieke Universiteit Leuven, K. Mercierlaan 94, 3001-Leuven-Heverlee, BELGIUM. This report is also available by anonymous ftp from *wgs.esat.kuleuven.ac.be* in the directory *pub/WGS/REPORTS/SLWN1998-6.ps.Z*

[†] Katholieke Universiteit Leuven - ESAT-SISTA/COSIC, K. Mercierlaan 94, 3001 Leuven, Belgium, tel 32/16/32 1808, fax 32/16/32 19 70, WWW: 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

[§] NICONET coordinator, Katholieke Universiteit Leuven - ESAT-SISTA/COSIC, K. Mercierlaan 94, 3001 Leuven, Belgium, tel 32/16/32 17 03, fax 32/16/32 19 70, WWW: http://www.esat.kuleuven.ac.be/sista, email: *sabine.vanhuffel@esat.kuleuven.ac.be*

[¶] Delft University of Technology, Department of Information Technology Systems, Control Engineering Group, P.O. Box 5031, 2600 GA Delft, The Netherlands

[‖] Katholieke Universiteit Leuven - ESAT-SISTA/COSIC, K. Mercierlaan 94, 3001 Leuven, Belgium, tel 32/16/32 17 15, fax 32/16/32 19 70, WWW: http://www.esat.kuleuven.ac.be/sista, email: *bart.demoor@esat.kuleuven.ac.be*

Abstract

Since the appearance of the first results on subspace system identification in the literature different subspace algorithms are available in practice. The most commonly used algorithms are N4SID, CVA and MOESP. Although they 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 these subspace 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. Also the influence of the identification parameters on the identification quality is studied and analysed.

Key Words: *subspace identification, numerical algorithms, control software*

Contents

1 Subspace identification - A brief overview

1.1 Problem formulation

The problem treated in linear subspace identification is that of finding an estimate of the system matrices A, B, C, D and the noise covariance matrix S, given measurements of the inputs u_k and y_k of an unknown linear time-invariant system of the form:

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

where the noise e_k has the following covariance matrix:

$$\mathbf{E}[e_p e_q^T] = S\delta_{pq}$$

1.2 Matrix input-output equations

The starting point of all subspace identification algorithms is the following set of matrix input-output equations:

$$\begin{aligned} Y_f &= \Gamma_i X_f + H_i^d U_f + H_i^s E_f, \\ Y_p &= \Gamma_i X_p + H_i^d U_p + H_i^s E_p. \end{aligned}$$

In these equations we have different terms which will be explained now. First of all there are the data Hankel matrices:

$$\begin{aligned} U_p &\stackrel{\text{def}}{=} \begin{pmatrix} u_0 & u_1 & \dots & u_{j-1} \\ u_1 & u_2 & \dots & u_j \\ \dots & \dots & \dots & \dots \\ u_{i-1} & u_i & \dots & u_{j+i-2} \end{pmatrix}, \\ U_f &\stackrel{\text{def}}{=} \begin{pmatrix} u_i & u_{i+1} & \dots & u_{i+j-1} \\ u_{i+1} & u_{i+2} & \dots & u_{i+j} \\ \dots & \dots & \dots & \dots \\ u_{2i-1} & u_{2i} & \dots & u_{2i+j-2} \end{pmatrix} \end{aligned}$$

where the indices p and f stand for past and future. The above matrices can also be defined in the same way for the outputs y_k and the output noise e_k as Y_p, Y_f and E_p, E_f respectively. We will also use the following short-hand notations:

$$W_p = \begin{pmatrix} Y_p \\ U_p \end{pmatrix}.$$

The past and future state sequences are defined as:

$$\begin{aligned} X_p &\stackrel{\text{def}}{=} \begin{pmatrix} x_0 & x_1 & \dots & x_{j-1} \end{pmatrix}, \\ X_f &\stackrel{\text{def}}{=} \begin{pmatrix} x_i & x_{i+1} & \dots & x_{i+j-1} \end{pmatrix}. \end{aligned}$$

Further we also have the following system related matrices:

$$\Gamma_i \stackrel{\text{def}}{=} \begin{pmatrix} C \\ CA \\ \dots \\ CA^{i-1} \end{pmatrix},$$

$$H_i^d \stackrel{\text{def}}{=} \begin{pmatrix} D & 0 & \dots & 0 \\ CB & D & \dots & 0 \\ \dots & \dots & \dots & \dots \\ CA^{i-2}B & CA^{i-3}B & \dots & D \end{pmatrix},$$

$$H_i^s \stackrel{\text{def}}{=} \begin{pmatrix} I & 0 & \dots & 0 \\ CK & I & \dots & 0 \\ \dots & \dots & \dots & \dots \\ CA^{i-2}K & CA^{i-3}K & \dots & I \end{pmatrix},$$

where Γ_i is the extended observability matrix, H_i^d the Toeplitz matrix containing the impulse response of the system and H_i^s the Toeplitz matrix containing the impulse response to a stochastic perturbation.

1.3 Orthogonal and oblique projections

A very commonly used tool in subspace identification are orthogonal and oblique projections of row spaces of matrices. Suppose we are interested in the projection of the row spaces of $\mathcal{A} \in \mathbb{R}^{p\times j}, \mathcal{B} \in \mathbb{R}^{q\times j}$ and $\mathcal{C} \in \mathbb{R}^{r\times j}$. The rows of $\mathcal{A}$ can be decomposed as a linear combination of the rows of $\mathcal{B}$, $\mathcal{C}$ and a third matrix orthogonal to $\mathcal{B}$ and $\mathcal{C}$. This can be written as:

$$\mathcal{A} = L_{\mathcal{B}}\mathcal{B} + L_{\mathcal{C}}\mathcal{C} + L_{\mathcal{B}^\perp,\mathcal{C}^\perp}\begin{pmatrix} \mathcal{B} \\ \mathcal{C} \end{pmatrix}^\perp.$$

where $L_{\mathcal{B}}$, $L_{\mathcal{C}}$ and $L_{\mathcal{B}^\perp,\mathcal{C}^\perp}$ are the respective weights of $\mathcal{B}$, $\mathcal{C}$ and $\begin{pmatrix} \mathcal{B} \\ \mathcal{C} \end{pmatrix}^\perp$ in the linear combination.

- The *orthogonal projection* of the row space of $\mathcal{A}$ into the row space of $\mathcal{B}$ is denoted by $\mathcal{A}/\mathcal{B}$ and defined as:

$$\mathcal{A}/\mathcal{B} \stackrel{\text{def}}{=} \mathcal{A}\mathcal{B}^\dagger\mathcal{B}.$$

- The *oblique projection* of the row space of $\mathcal{A}$ into the row space of $\mathcal{B}$ along the row space of $\mathcal{C}$ is denoted by $\mathcal{A}/_{\mathcal{C}}\,\mathcal{B} = L_{\mathcal{B}}\mathcal{B}$. This oblique projection can be interpreted through the following recipe: Project the row space of $\mathcal{A}$ orthogonally on the joint row space of $\mathcal{B}$ and $\mathcal{C}$ and decompose the result along the row space of $\mathcal{B}$. Mathematically, the orthogonal projection of the row space of $\mathcal{A}$ on the joint row space of $\mathcal{B}$ and $\mathcal{C}$ can be stated as:

$$\begin{aligned} A/\begin{pmatrix} \mathcal{B} \\ \mathcal{C} \end{pmatrix} &= \mathcal{A}\begin{pmatrix} \mathcal{B} \\ \mathcal{C} \end{pmatrix}^{\dagger}\begin{pmatrix} \mathcal{B} \\ \mathcal{C} \end{pmatrix} \\ &= L_{\mathcal{B}}\mathcal{B} + L_{\mathcal{C}}\mathcal{C} \\ &= \mathcal{A}/_{\mathcal{C}}\ \mathcal{B} + \mathcal{A}/_{\mathcal{B}}\ \mathcal{C}. \end{aligned}$$

This leads to the following definition for the oblique projection:

$$\mathcal{A}/_{\mathcal{C}}\ \mathcal{B} \stackrel{\text{def}}{=} \mathcal{A}\left[\begin{pmatrix} \mathcal{B} \\ \mathcal{C} \end{pmatrix}^{\dagger}\right]_{(:,1:q)} \mathcal{B}.$$

1.4 Basics of subspace identification

The subspace identification problem can also be interpreted as follows: given the past inputs and outputs W_p and the future inputs U_f we want to find an optimal prediction of the future outputs Y_f. If we use a linear predictor:

$$\widehat{Y}_f = L_w W_p + L_u U_f$$

the least squares prediction $\widehat{Y}_f$ of Y_f can be found from the following least squares problem:

$$\min_{L_w, L_u} ||Y_f - \begin{pmatrix} L_w & L_u \end{pmatrix}\begin{pmatrix} W_p \\ U_f \end{pmatrix}||_F^2.$$

The solution to this problem is the orthogonal projection of the row space of Y_f into the row space spanned by W_p and U_f:

$$\begin{aligned} \widehat{Y}_f &= Y_f/\begin{pmatrix} W_p \\ U_f \end{pmatrix} \\ &= \underbrace{Y_f/_{U_f}\ W_p}_{L_w W_p} + \underbrace{Y_f/_{W_p}\ U_f}_{L_u U_f}. \end{aligned}$$

A very important property of the projections in the right hand side is that it is a rank deficient term since:

$$Y_f/_{U_f}\ W_p = L_w W_p = \Gamma \widehat{X}_f.$$

In practice the above projection may be weighted on the left and the right by some weighting matrices W_1 and W_2 which gives us:

$$W_1 Y_f/_{U_f}\ W_p W_2 = W_1 \Gamma \widehat{X}_f W_2. \tag{1}$$

2 Data sets

To make the comparison as general as possible, data sets from different real-life systems have been considered. One of our primary concerns has been to make the present comparison completely accessible to other researchers. Therefore all data sets as well as the Matlab programs that were used have been made publicly available on the DAISY internet site [1]:

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

Every data set available on the DAISY internet site is labeled by a 5 digit number. The list of data sets is the given by the following list.

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. **cstr** Data of a continuous stirring tank reactor, where the reaction is exothermic and the concentration is controlled by regulating the coolant flow. [98-002].
4. **dryer 1** Laboratory setup acting like a hair dryer [96-06].
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** [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].

A more detailed description of the data sets can be found on the DAISY internet site.

It should be noted that for the identification tests all data sets were detrended which means that from all inputs and outputs the mean value was removed.

3 Subspace identification algorithms

In this section the 3 algorithms that were used are shortly described. For more technical details we refer to a paper where the method was described.

CVA Canonical variate analysis [3]. Constrained version of [4].

N4SID Numerical algorithm for subspace state space system identification [5]. The algorithm we used is the `subid.m` matlab function that comes with the previously cited book.

MOESP SMI-toolbox. The functions that have been used here are `dordpo.m`, `dmodpo.m` and `dac2bd.m` [2].

It is possible to interprete the above mentioned subspace identification algorithms in a unifying framework. Indeed, in [5] it was shown that they correspond to the weighted projection (1) with the appropiate weighting matrices W_1 and W_2. We present in Table 3 the acronyms and references of these algorithms and the weights to be plugged into (1). It should be noted that the different algorithms all have their particularities and have been used as provided by the different authors. For instance they do not all use a QR-decomposition. As a consequence the computational time and/or the precision of the algorithm will be affected by the way it is implemented. A more thorough study and analysis of the matlab-code would be necessary to try to explain this. The different algorithms have been implemented under Matlab 5.1 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 additional parameters i and n are the number of block rows in the data Hankel matrices and the chosen system order respectively. The number of block rows in the past and future matrices is supposed to be the same. The outputs of the function are the system matrices A, B, C, D and the matrix K as defined in section 1.

Acronym	Full name and Main ref.	W_1	W_2
N4SID	**N**umerical Algorithm for **S**ubspace **S**tate **S**pace **S**ystem **ID**entification	I_{li}	$I_{(m+l)i}$
CVA	**C**anonical **V**ariate **A**nalysis	$[(Y_f/U_f^\perp).(Y_f/U_f^\perp)^T]^{-1/2}$	$/U_f^\perp$
MOESP	**M**ultivariable **O**utput-**E**rror **S**tate s**P**ace	I_{li}	$/U_f^\perp$

Table 1: **Interpretation of the different subspace identification algorithms in a unifying framework. The weighting matrices W_1 and W_2 for the different subspace algorithms to be plugged in (1).**

4 Benchmark quality criteria

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

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

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

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

with $y_{k,i}$ the i-th output at time step k, y_k^p the one-step-ahead predicted output, l the number of outputs and val the number of validation data points. A data set is always split into an identification set and a validation set. With the identification set the model parameters are calculated while the validation set is used to find the prediction error.

Simulation error: is also calculated based on the validation data set and is defined as follows:

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

with $y_{k,i}$ the i-th output at time step k, y_k^s the simulated output and l the number of outputs.

The above three quality criteria are calculated for a wide range of the identification parameters i and n. i is the number of block rows in the data Hankel matrices. The chosen order of the system n always has to be such that $n < il$. The results for the different data sets and the different values of i and n are visualised in a 3D graph.

5 Simulation results

5.1 Simulation parameters

	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 2: **Parameters of the different data sets.** m **is the number of inputs,** l **the number of outputs,** 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 to 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.**

5.2 Overall results

The following tables give the overall results of the identification for the 15 data sets. Table 3 and 4 present the minimal value of the prediction error and the simulation error respectively found for the different values of i and n. Table 5 gives the computational load of the different subspace algorithms and for the different data sets with the i and n values where the prediction error of the CVA algorithm is minimal.

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 3: **The minimum value of the prediction error over the different values of i and n.**

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 4: **The minimum value of the simulation error over the different values of i and n.**

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 5: **The computational comlexity (in number of Mega flops) corresponding to the value of i and n where the prediction error of the CVA algorithm is minimal.**

5.3 Computational complexity

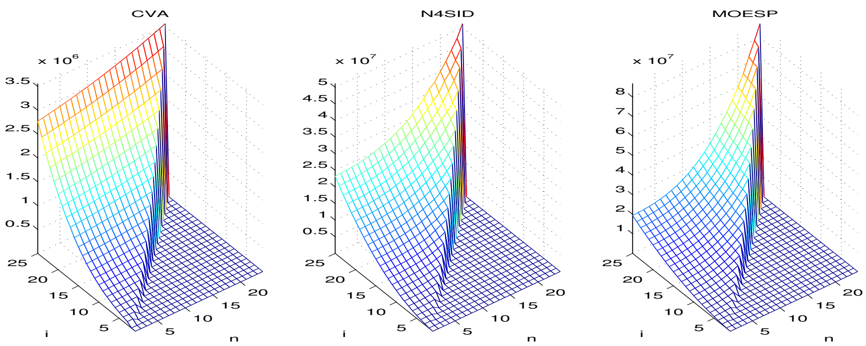

Figure 1: **Computational complexity of data set 1.**

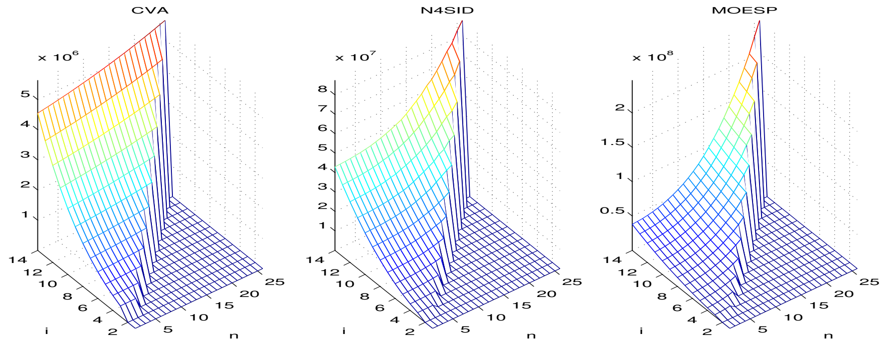

Figure 2: **Computational complexity of data set 2.**

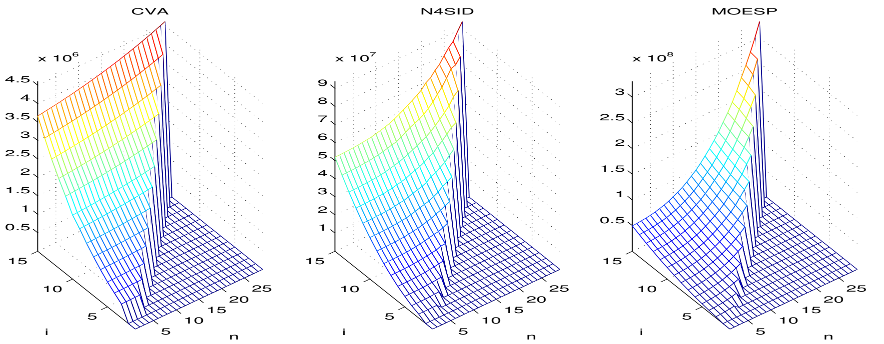

Figure 3: **Computational complexity of data set 3.**

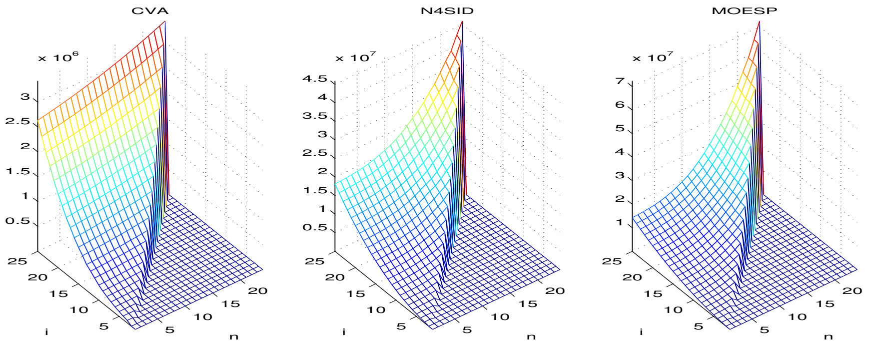

Figure 4: **Computational complexity of data set 4.**

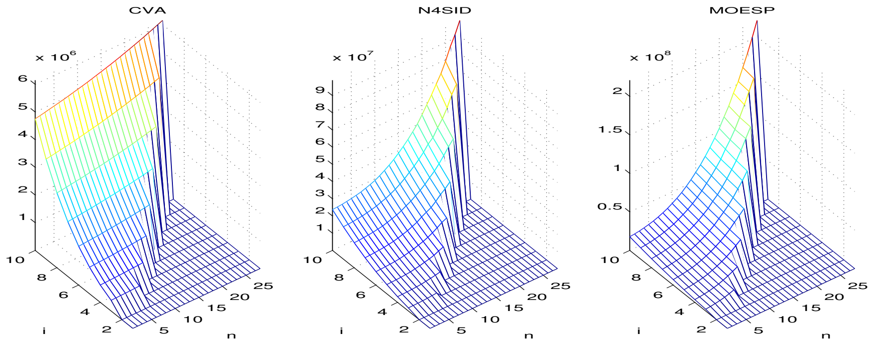

Figure 5: **Computational complexity of data set 5.**

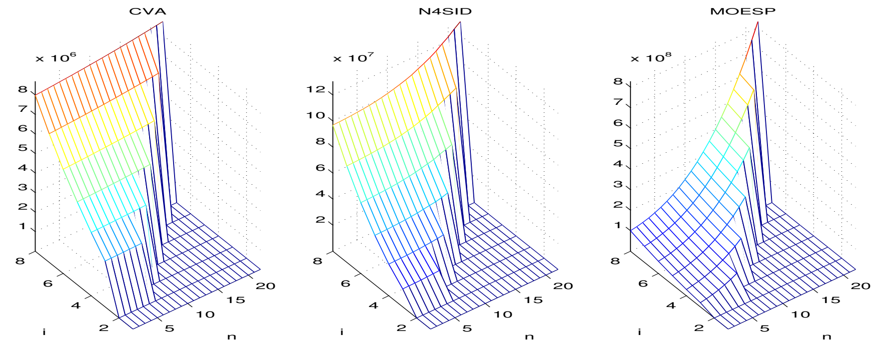

Figure 6: **Computational complexity of data set 6.**

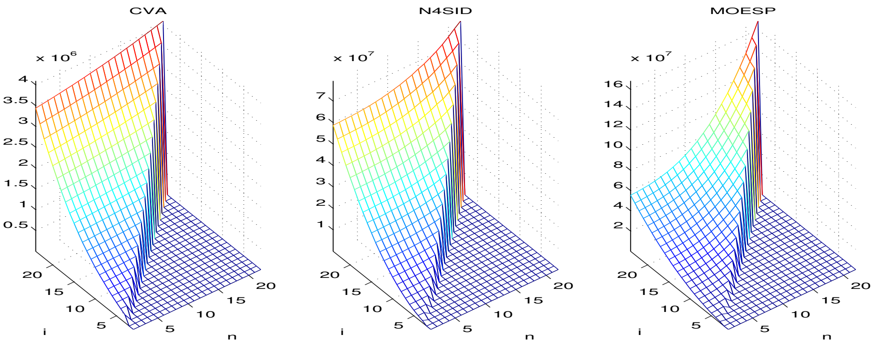

Figure 7: **Computational complexity of data set 7.**

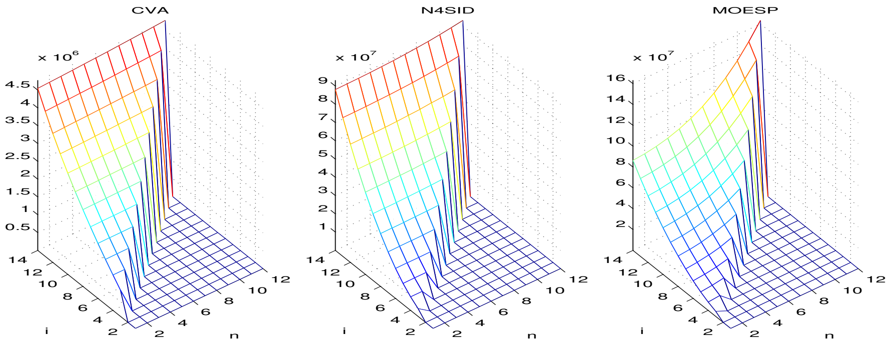

Figure 8: **Computational complexity of data set 8.**

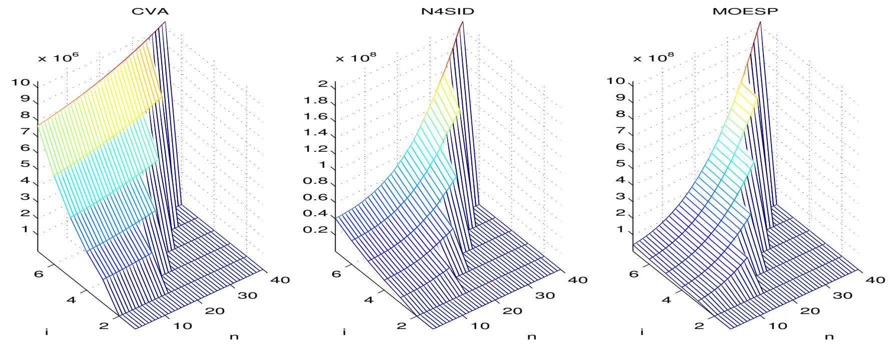

Figure 9: **Computational complexity of data set 9.**

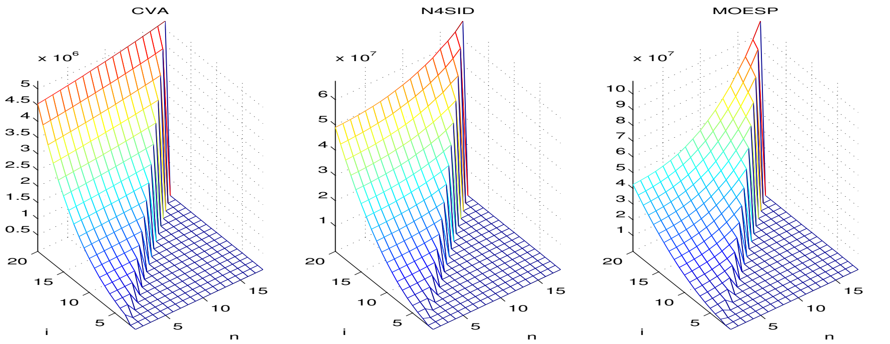

Figure 10: **Computational complexity of data set 10.**

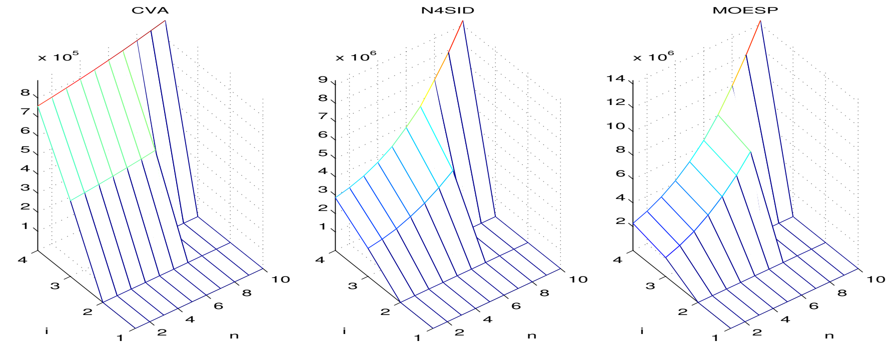

Figure 11: **Computational complexity of data set 11.**

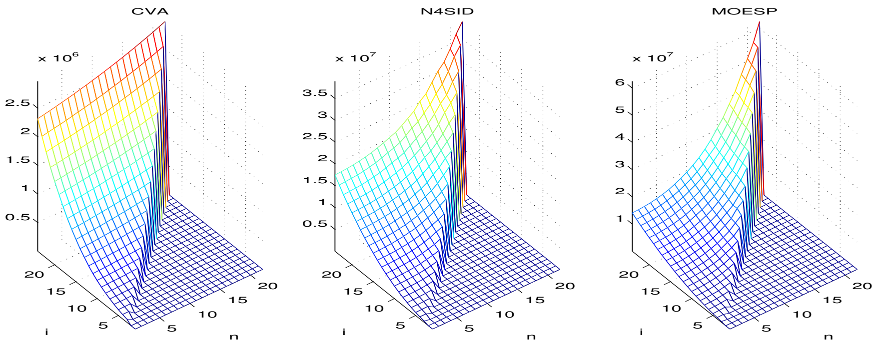

Figure 12: **Computational complexity of data set 12.**

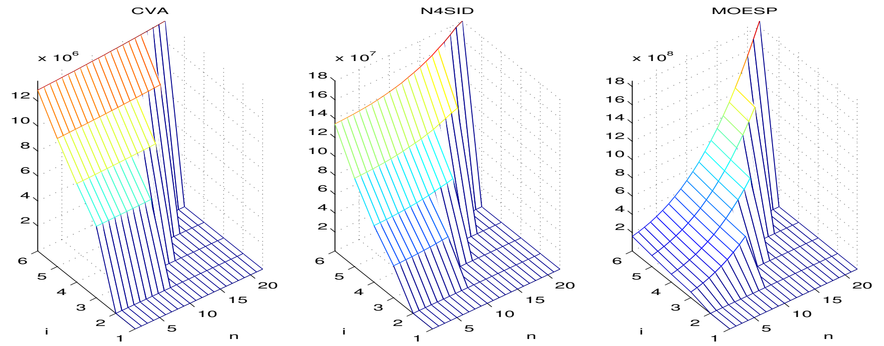

Figure 13: **Computational complexity of data set 13.**

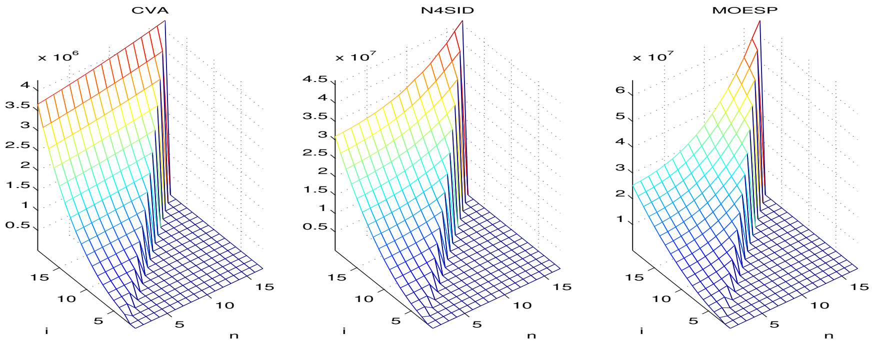

Figure 14: **Computational complexity of data set 14.**

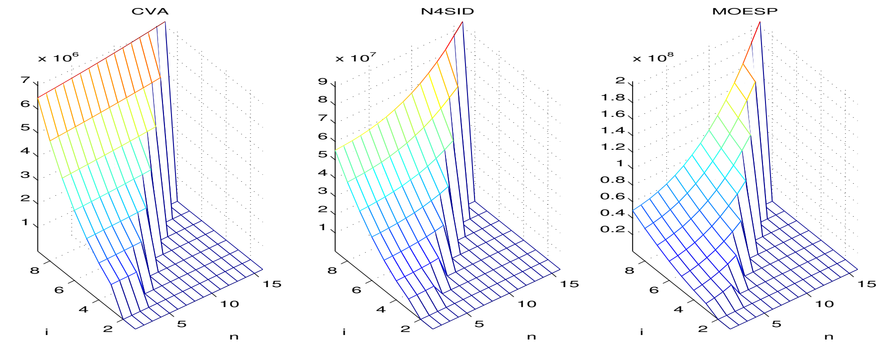

Figure 15: **Computational complexity of data set 15.**

5.4 Prediction errors

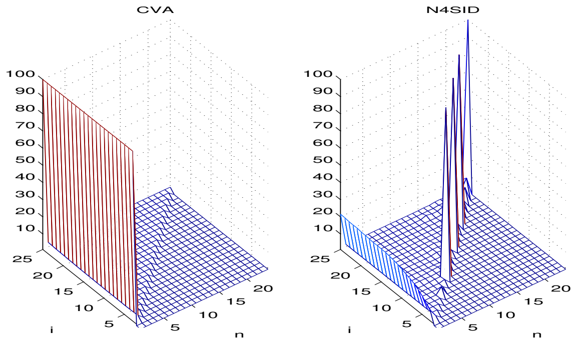

Figure 16: **Prediction error of data set 1.**

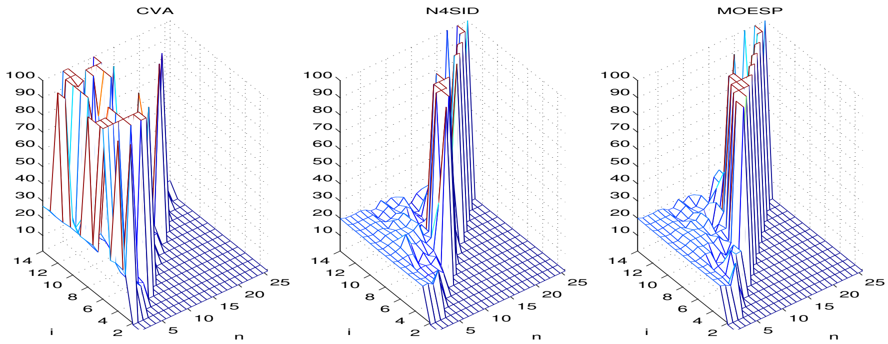

Figure 17: **Prediction error of data set 2.**

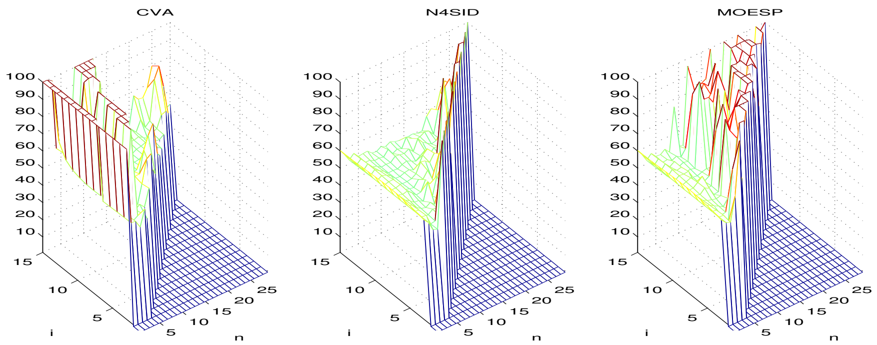

Figure 18: **Prediction error of data set 3.**

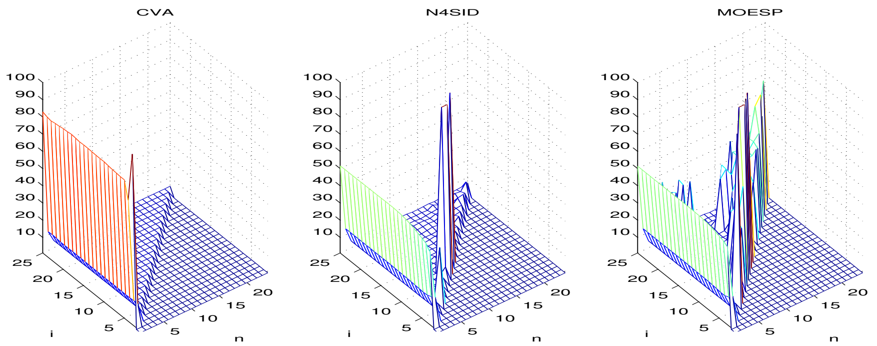

Figure 19: **Prediction error of data set 4.**

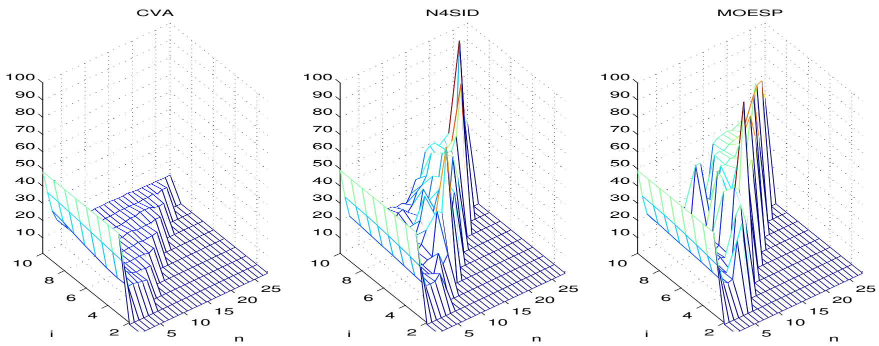

Figure 20: **Prediction error of data set 5.**

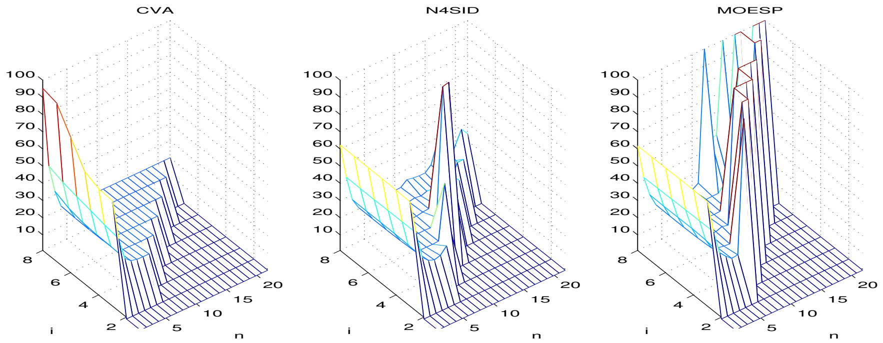

Figure 21: **Prediction error of data set 6.**

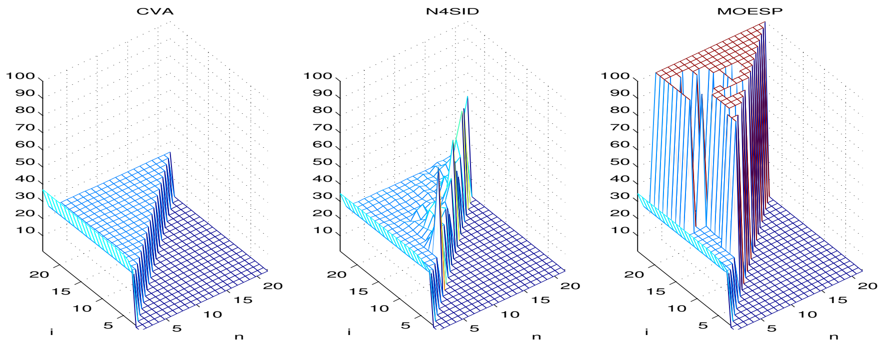

Figure 22: **Prediction error of data set 7.**

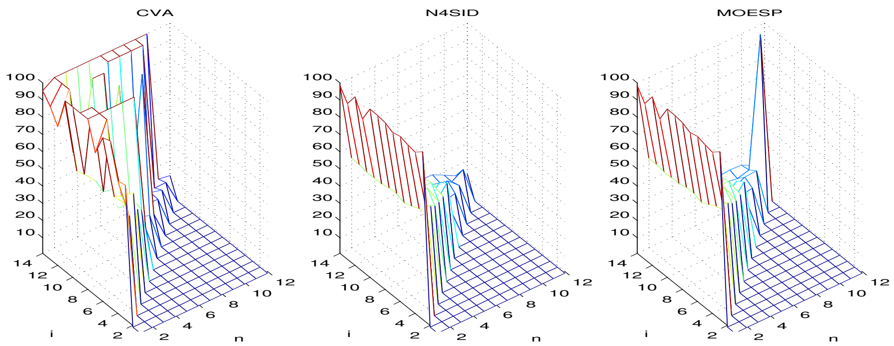

Figure 23: **Prediction error of data set 8.**

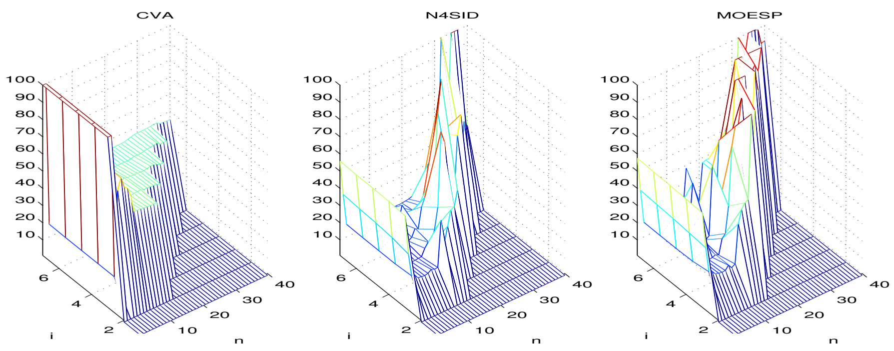

Figure 24: **Prediction error of data set 9.**

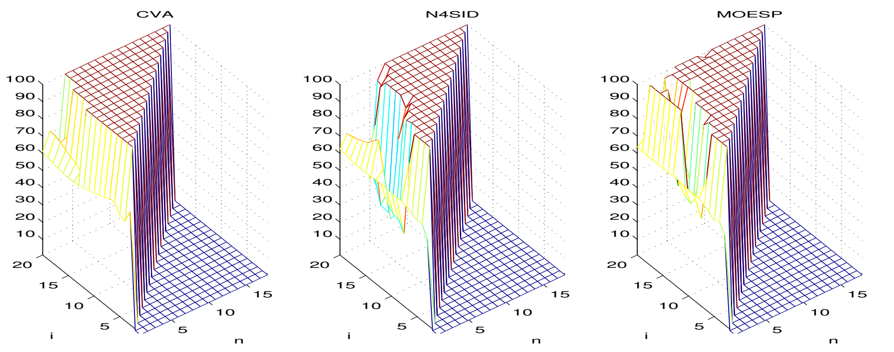

Figure 25: **Prediction error of data set 10.**

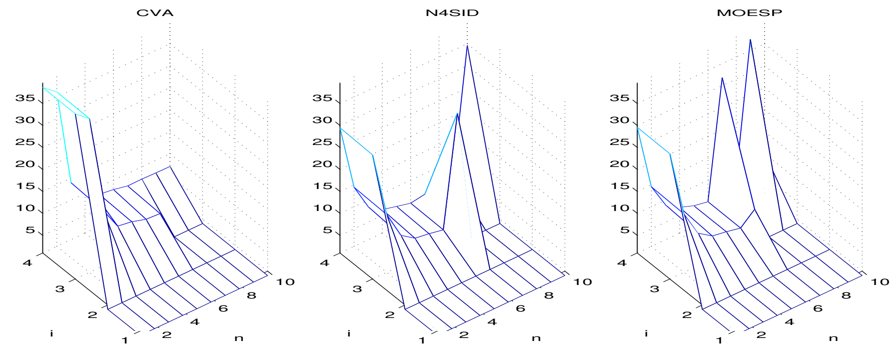

Figure 26: **Prediction error of data set 11.**

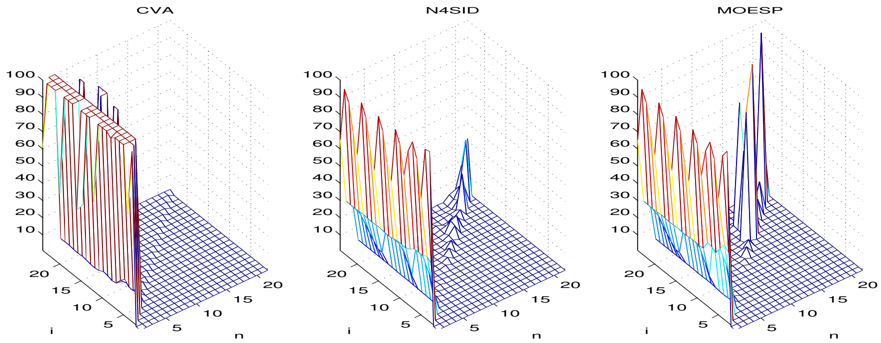

Figure 27: **Prediction error of data set 12.**

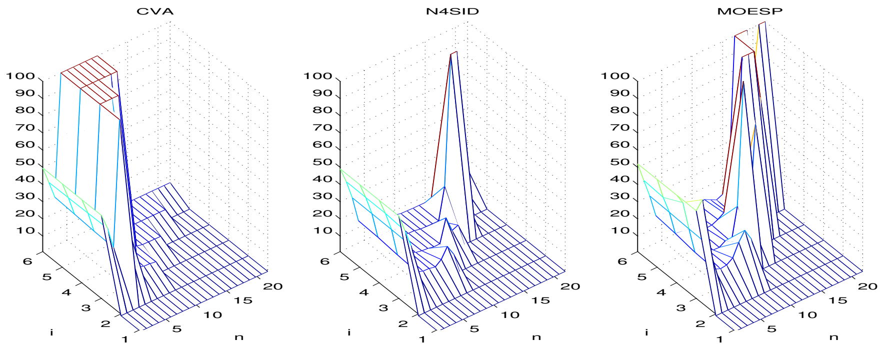

Figure 28: **Prediction error of data set 13.**

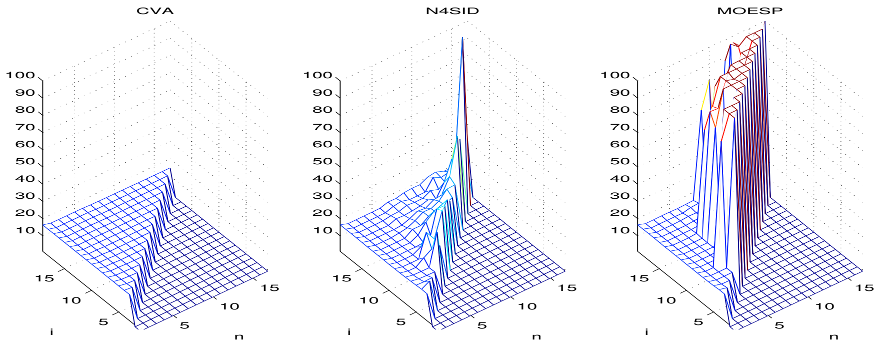

Figure 29: **Prediction error of data set 14.**

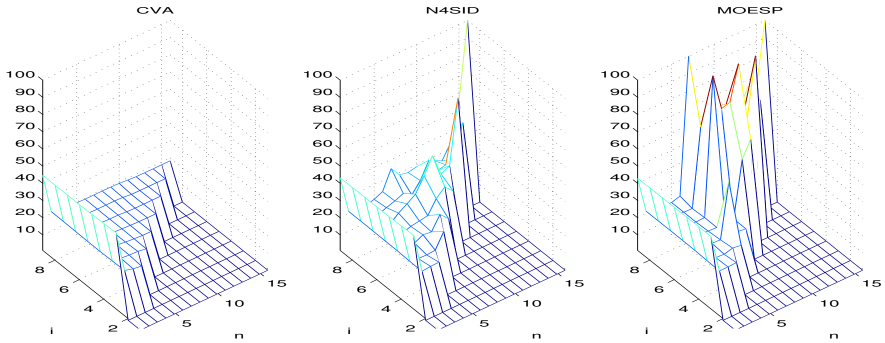

Figure 30: **Prediction error of data set 15.**

5.5 Simulation errors

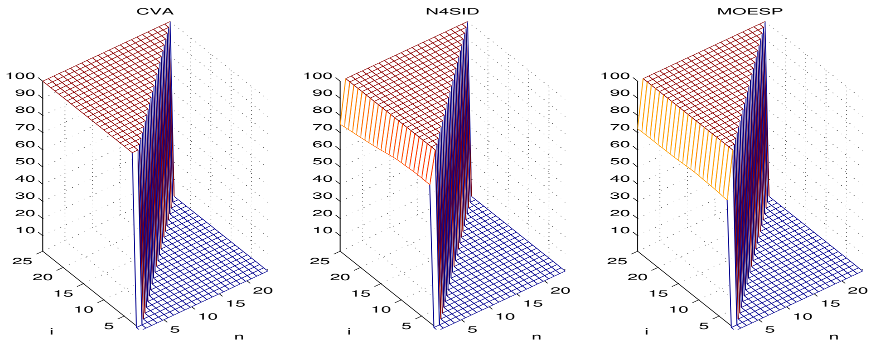

Figure 31: **Simulation error of data set 1.**

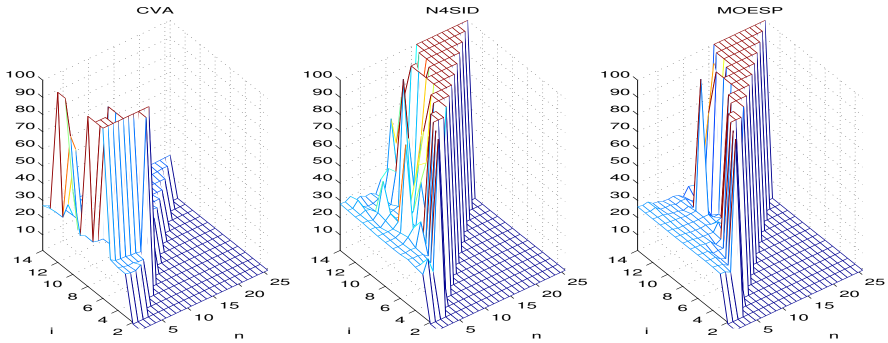

Figure 32: **Simulation error of data set 2.**

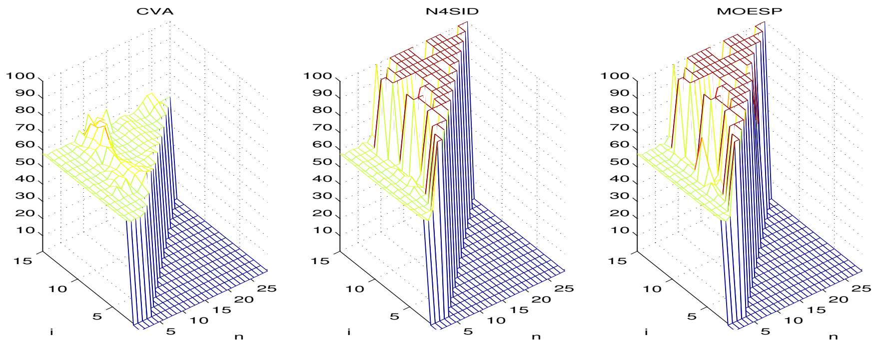

Figure 33: **Simulation error of data set 3.**

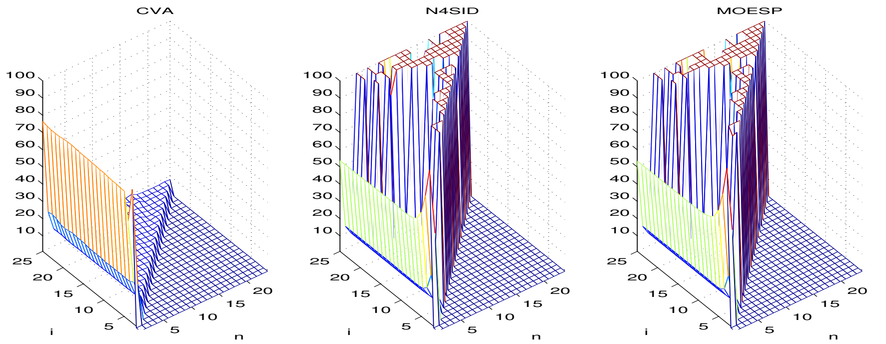

Figure 34: **Simulation error of data set 4.**

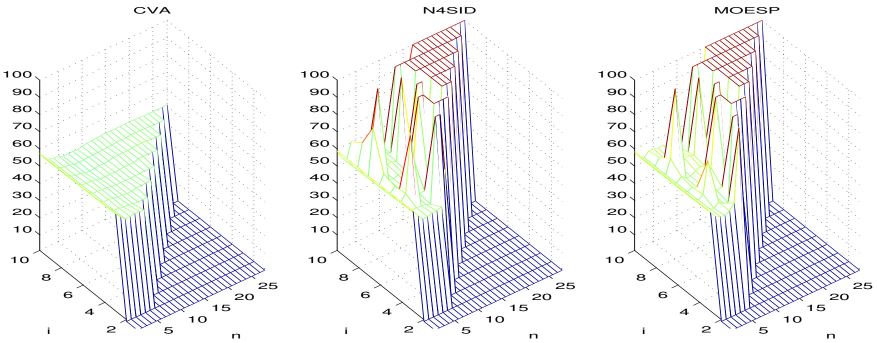

Figure 35: **Simulation error of data set 5.**

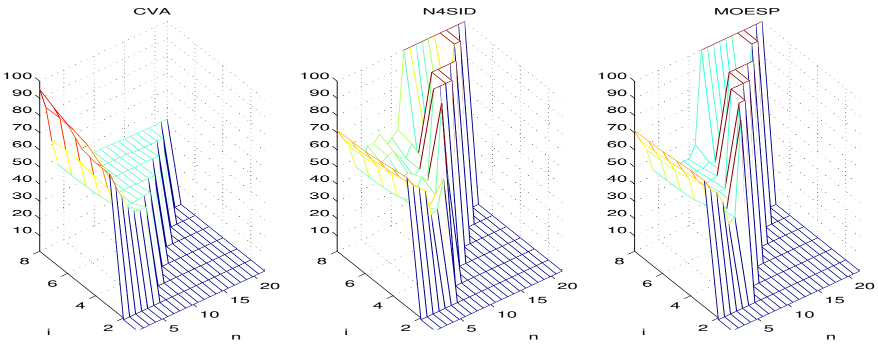

Figure 36: **Simulation error of data set 6.**

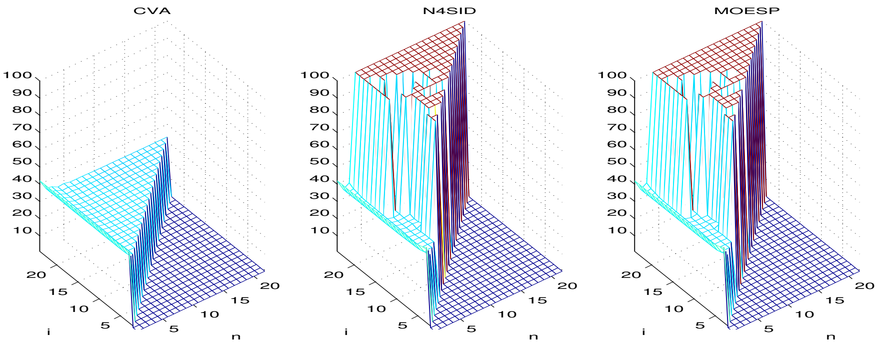

Figure 37: **Simulation error of data set 7.**

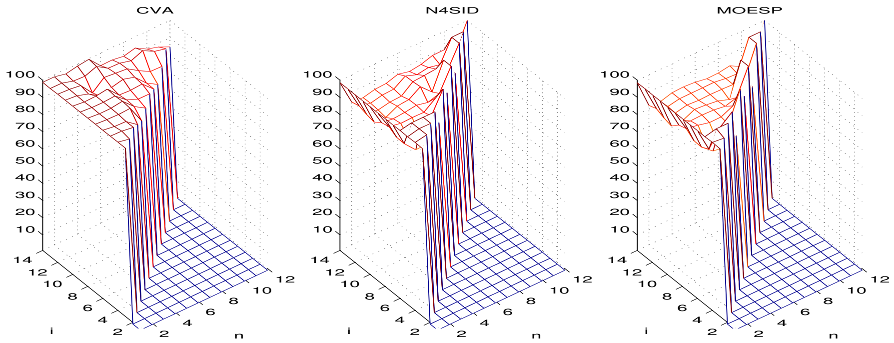

Figure 38: **Simulation error of data set 8.**

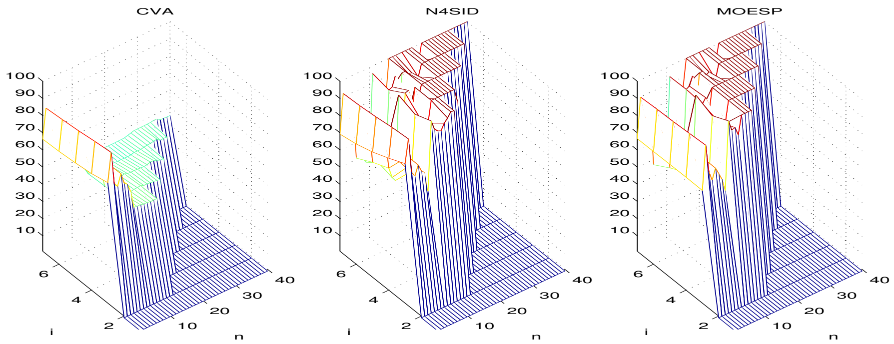

Figure 39: **Simulation error of data set 9.**

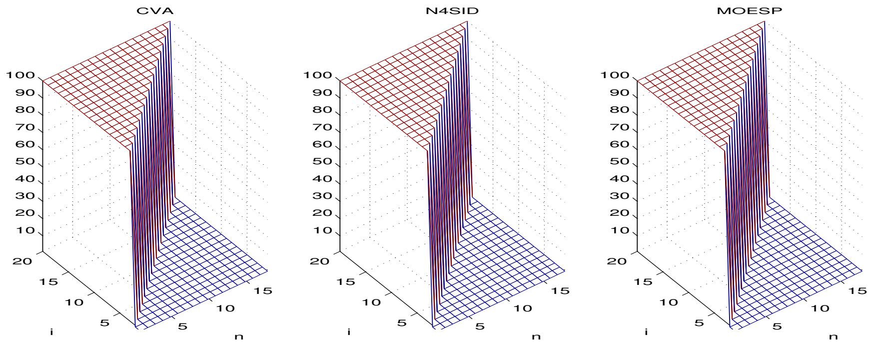

Figure 40: **Simulation error of data set 10.**

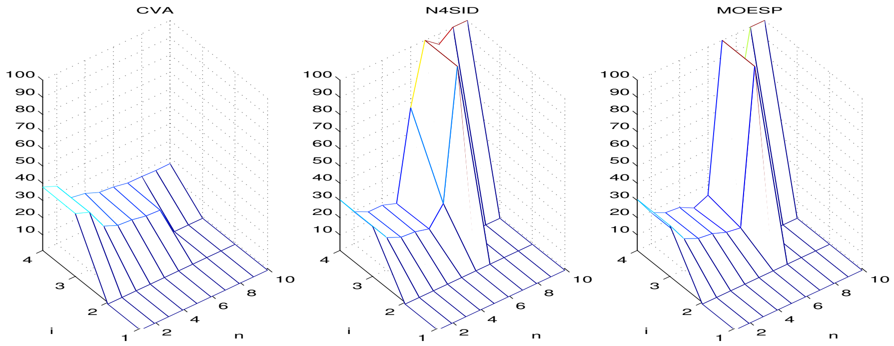

Figure 41: **Simulation error of data set 11.**

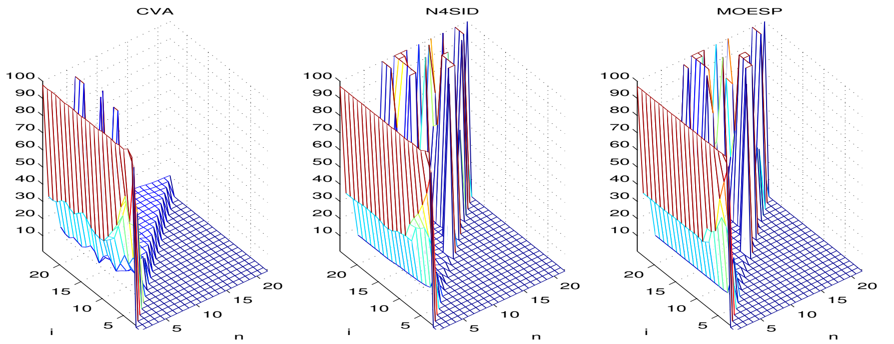

Figure 42: **Simulation error of data set 12.**

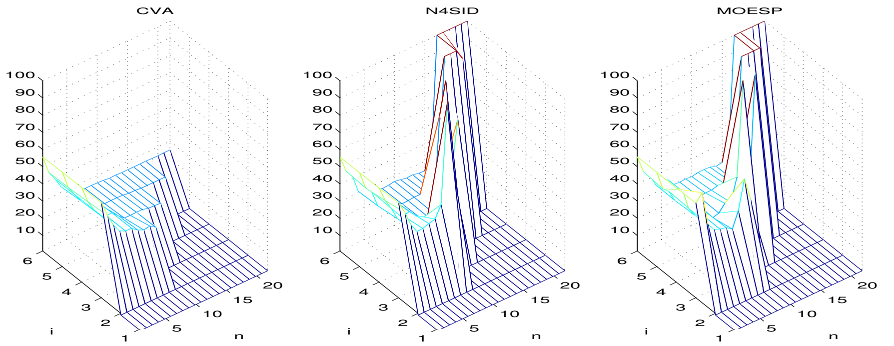

Figure 43: **Simulation error of data set 13.**

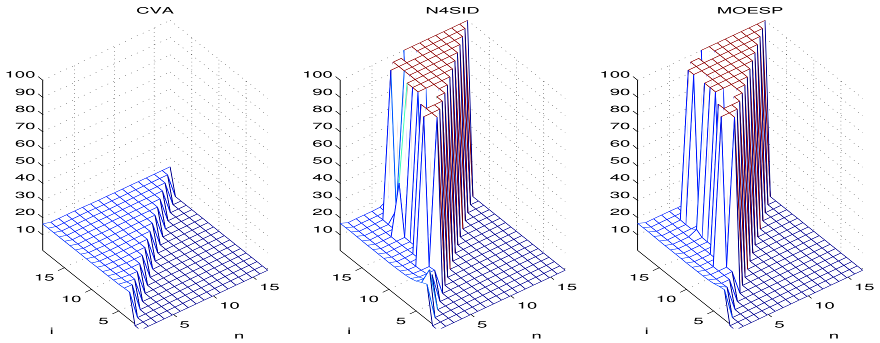

Figure 44: **Simulation error of data set 14.**

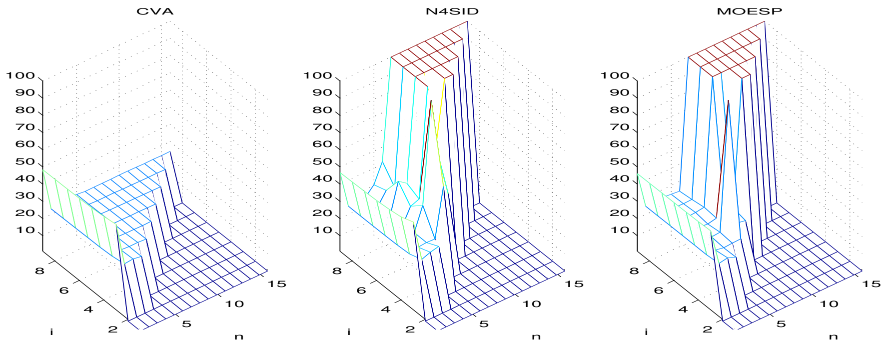

Figure 45: **Simulation error of data set 15.**

6 Conclusions

From the above presented results we can draw the following conclusions:

- The CVA algorithm and the N4SID algorithm have the smallest prediction errors.
- The MOESP algoritm has the smallest simulation errors. CVA and N4SID seem to perform equally well.
- The computational complexity of the CVA method is much smaller than for the other two algorithms. MOESP is clearly the slowest algorithm. The reason probably lies in the way the different methods are implemented. Since we did not compare the Matlab code of the algorithms with each other we can not give more specific reasons for this result.

- The CVA algorithms seems to be much more stable with respect to changes in the i (number of block rows in the data matrices) and n (system order) parameters than the N4SID and MOESP algorithm. Here we do not have a specific explanation. We suspect that the symetry of the CVA algorithm plays an important role in this.

7 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.

References

[1] 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)*, 38(3):4–5, Sep. 1997. http://www.esat.kuleuven.ac.be/sista/daisy.

[2] 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.

[3] W.E. Larimore. Canonical variate analysis in identification, filtering and adaptive control. In *Proc. 29th Conference on Decision and Control*, pages 596--604, Hawai, 1990.

[4] K. Peternell, W. Scherrer, and M. Deistler. Statistical analysis of novel subspace identification methods. *Signal Processing*, 52:161--177, 1996.

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