Identification of a Steel Subframe Flexible Structure Using SLICOT System Identification Toolbox

Vasile Sima
National Institute for Research & Development in Informatics
Bd. Mareşal Al. Averescu, Nr. 8–10
71316 Bucharest, Romania
eMail: vsima@iciadmin.ici.ro
phone +40-21-224-0765; fax +40-21-224-0539

Abstract— **This paper uses fast system identification algorithms for a challenging steel subframe flexible structure application. The input-output data are available in the DAISY collection (`http://www.esat.kuleuven.ac.be/sista/daisy`), and consist of 8523 samples, 2 inputs, and 28 outputs. Commercial system identification codes cannot be conveniently used on typical workstations to solve the associated problem, due to its large size. The recently developed, multivariable system identification toolbox—SLIDENT—incorporated in the Fortran 77 Subroutine Library in Control Theory (SLICOT) successfully handled this application, and enabled to perform various experiments in a timely manner. SLIDENT provides drivers, computational routines, and MATLAB or Scilab interfaces, which implement several algorithmic approaches, and use standard or fast techniques for data processing. Both linear and Wiener-type multivariable discrete-time systems are addressed. The results show that SLIDENT is reliable and able to solve large identification problems.**

Keywords— **Estimation, Identification for Control, Identification Methods, Signals and Systems, Subspace Methods.**

I. Introduction

Discrete-time linear multivariable systems are often identified using subspace-based techniques. These techniques are attractive mainly for the following reasons: state-space models are directly estimated; no parameterizations are needed; robust linear algebra tools like QR decomposition and singular value decomposition (SVD) are used; only one parameter has to be chosen. Two commonly used approaches are MOESP (Multivariable Output Error state SPace) [22], [23], and N4SID (Numerical algorithm for Subspace State Space System IDentification) [20], [21].

A subspace-based identification procedure is often significantly faster than an optimization-based procedure, since no iterative algorithms for parameter estimation are involved. Subspace techniques have, however, some limitations; for instance, they cannot guarantee the preservation of stability or real positivity of a physical system. Extensions of these techniques have been recently proposed (e.g., [19]) to remove such limitations.

This paper investigates a system identification procedure using the steel subframe flexible structure [1] input-output data, stored as Example [96-013] of the DAISY collection [5], available from

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

This is a large data set, with $t = 8523$ samples, $m = 2$ inputs, and $\ell = 28$ outputs. The input signals correspond to two shakers at two locations, and the 28 outputs are accelerations provided by accelerometers around the structure, used for measurements. The sampling period is 1/1024 seconds. Both inputs are white noise forces. Assuming that the outputs of the flexible structure are independent, it is expected that a dynamical system modelling the given input-output data should have the order at least 28. Since this identification problem is quite large, it seems reasonable to start with a smaller order, and analyze the modelling errors for increasing orders. Even if a strict upper bound s on the system order n is set to 21, the MATLAB codes[1], which we could try, either needed too much time (dozens of minutes, or even hours, on an IBM-PC computer at 500 MHz and 128 Mb memory), or ran into "Out of memory" errors. On the other hand, the fast functions available in the recently developed, multivariable system identification toolbox SLIDENT,[2] incorporated in the Fortran 77 **S**ubroutine **L**ibrary **i**n **Co**ntrol **T**heory (SLICOT) [3], solved the problem with $s = 21$ in about one minute. A much larger value for s, $s = 61$, was used for getting the results of this paper using SLIDENT tools.

SLIDENT toolbox provides drivers, computational routines, and MATLAB [7] or Scilab [4] interfaces, which implement several algorithmic approaches, and use standard or fast techniques for data processing. The implementations are based on the state-of-the-art linear algebra package LAPACK [2] and Basic Linear Algebra Subprograms (BLAS) collections, which enable to take advantage of the capabilities of modern high-performance computer architectures. LAPACK and BLAS have been already used in the previously developed codes [11], [12]. Details about abilities and performance of the SLICOT identification toolbox are given in several reports and papers [14]–[18].

This work was supported in part by the European Community BRITE-EURAM III *Thematic Networks Programme NICONET* (project BRRT–CT97-5040).

[1] MATLAB is a registered trademark of The MathWorks, Inc.

[2] Available from `ftp://ftp.esat.kuleuven.ac.be/pub/WGS/SLICOT`, directory `MatlabTools/Windows/SLToolboxes`, file `ident_mex.zip`, or from `http://www.win.tue.nl/niconet/NIC2/NICtask3A.html`, for PC-Windows platforms.

The toolbox has been recently extended [10] to cover identification of Wiener systems, consisting of a linear part and a static nonlinearity.

II. Linear System Identification of the Flexible Structure Application

Consider a linear time-invariant discrete-time state space model, described by

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

where $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$, and $y(k) \in \mathbb{R}^\ell$ are the state, input, and output vectors at time k, respectively, for $k = 1, \ldots, t$, A, B, C, and D are real matrices of appropriate dimensions, $\{w_k\}$ and $\{v_k\}$ are zero-mean stationary ergodic state and output disturbance or noise sequences, uncorrelated with $\{u_k\}$ and the initial state of (1), with covariances satisfying the relation

$$\mathcal{E}\left\{ \begin{bmatrix} w_p \\ v_p \end{bmatrix} \begin{bmatrix} w_q^T & v_q^T \end{bmatrix} \right\} = \begin{bmatrix} Q_w & S \\ S^T & R_v \end{bmatrix} \delta_{pq} \geq 0 \, , \qquad (2)$$

where $\mathcal{E}$ denotes the expected value operator and δ_{pq} is the Kronecker delta symbol. Both MOESP and N4SID associated algorithms start by building a large block-Hankel-block matrix H (which is a concatenation of two block-Hankel matrices in terms of the input and output data sequences, respectively), and perform a data compression by finding an upper triangular factor R of a QR factorization of H, $H = QR$, but the matrix Q is not needed subsequently. Then, a SVD of a certain matrix, built from R, reveals the order n of the system as the number of "nonzero" singular values. System matrices are computed from the right singular vectors, and other submatrices of R. The Kalman gain is obtained by solving a discrete-time algebraic matrix Riccati equation using the covariance matrices estimated based on the residuals of a least squares problem. Besides the standard QR algorithm for data compression, SLIDENT includes two fast algorithms, which exploit the special structure of the matrix H: Cholesky factorization of the efficiently built inter-correlation matrix [13], or fast QR factorization [6], based on the generalized Schur algorithm.

The use of the SLIDENT toolbox is illustrated below for the flexible structure data. The application data sequences are loaded and preprocessed using the following Matlab commands, which include detrending of input and output variables, for removing any linear trend

```
load flexible_structure_dat;
u  = flexible_structure_dat(:,1:2);
y  = flexible_structure_dat(:,3:30);
u  = detrend(u);  y = detrend(y);
```

Figure 1 illustrates the trajectories of the inputs and the first two outputs.

Finding a model based on all data can efficiently be done using SLIDENT function slmoen4—a combination of MOESP and N4SID—with a fast data compression algorithm, either Cholesky [13], for alg = 1 (default), or

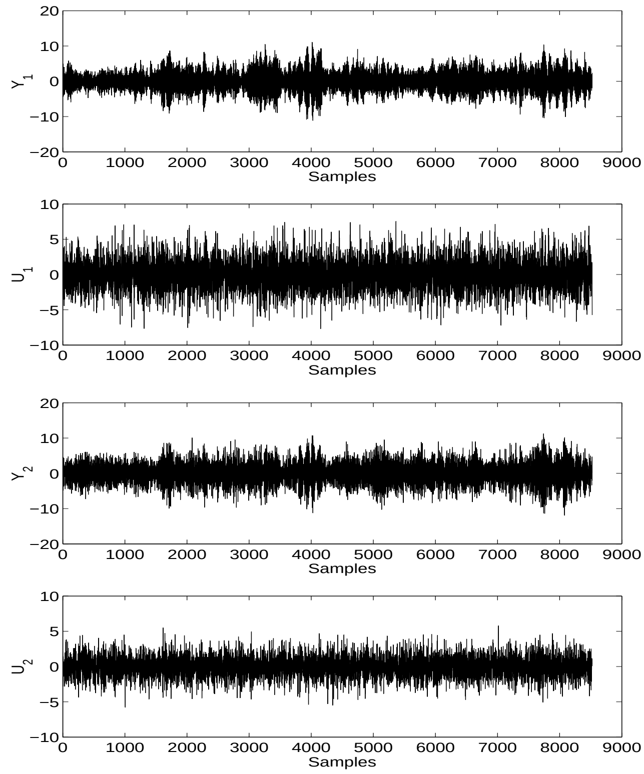

Fig. 1. The input and the first two output trajectories for the steel subframe flexible structure application (DAISY): u_1, y_1 (top half) and u_2, y_2 (bottom half).

fast QR factorization [6], for alg = 2. A value $s = 61$ was used for the "number of block rows" s, for getting the results, as follows,

```
s = 61;  n = [ ];  alg = 2;
[sys,K,rcnd,R] = slmoen4(s,y,u,n,alg);
```

The command above, calling slmoen4, computes and plots the $s\ell$ singular values, which in theory could give an indication of the system order.

The singular value plot, shown in Figure 2, reveals a very fast decay of the singular values around the abscissa value 30. (The figure also displays a similar plot for $s = 41$; it appears that this value can be used equally well for identification purposes, and it would allow faster computations.)

The system order defined by the default tolerance is $n = 28$. Any other desired value of n, less than s, may be specified after inspecting the singular values. The slmoen4 function then computes the system matrices (A, B, C, D) and stores them as a Matlab system object, sys, that is sys.a gives the matrix A, etc. The Kalman gain matrix K, and the upper triangular factor R of the QR factorization of the huge matrix H, are also computed. Matrix H has the size $N \times 2s(m+\ell)$, where $N = t-2s+1$. With the given and chosen values, H is 8462×3660, and R is 3660×3660. Matrix R can then be used for finding any models of order less than s. The output parameter rcnd contains the reciprocal condition numbers of linear algebraic systems solved by the identification algorithm. Therefore, it is possible

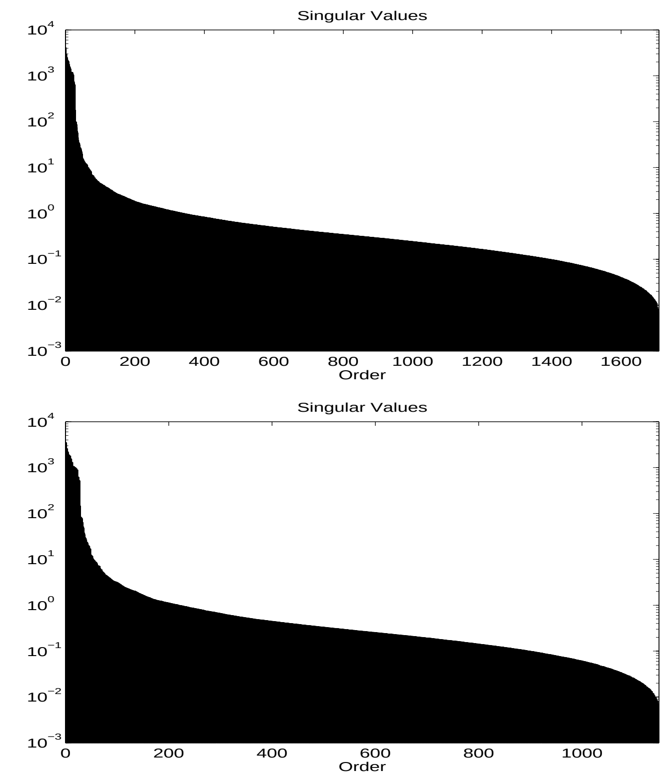

Fig. 2. Singular value plots for steel subframe flexible structure application, for $s = 61$ (top) and $s = 41$ (bottom).

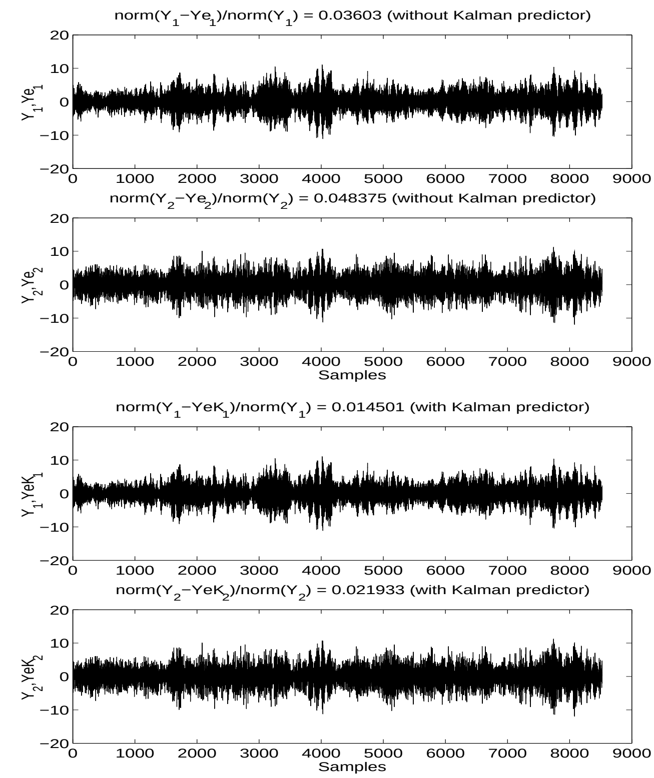

Fig. 3. The first two original and estimated output trajectories for the steel subframe flexible structure application ($n = 28$) without predictor (top half) and with predictor (bottom half).

to check the quality of the intermediate and final results by inspecting these numbers. The values delivered by the above calculations were larger than 0.0004.

The quality of the model can be assessed using the function `find_err`, included in the demonstration suite for the SLIDENT toolbox,[3]

```
[err(1),ye]  = find_err(y,u,sys);
[err(2),yeK] = find_err(y,u,sys,K);
```

The commands above compute the estimated output trajectories, `ye` and `yeK`, without and with a Kalman predictor, respectively, and their relative errors `err(1:2)`, in comparison with the given, original output trajectory, $\{y(k)\}_{k=1}^{t}$. The relative error 1-norms, without and with predictor,

```
norm(y - ye, 1)/norm(y,1)
norm(y - yeK,1)/norm(y,1)
```

have the values 0.0628 and is 0.0548, respectively. The estimated output, without and with predictor, together with the original output, can be plotted using the SLIDENT calling statements

```
plot_ye(y,ye);  plot_ye(y,yeK);
```

which give the plots in Figure 3 (for the first two outputs). A detailed plot of the beginning 200 samples appears in Figure 4. The estimated outputs cannot practically be distinguished from the original outputs.

[3]Available from `ftp://ftp.esat.kuleuven.ac.be/pub/WGS/SLICOT`, directory `MatlabTools/Windows/SLdemos`, file `slident_demo.zip`, or from `http://www.win.tue.nl/niconet/NIC2/NICtask3A.html`, for PC-Windows platforms.

The means and minimum values of the engineering measure of fit, called Variance-Accounted-For (VAF), in percentages,[4] for all outputs of the system of order 28 are:

	Without predictor	With predictor
mean(VAF)	99.7218	99.8622
min(VAF)	99.0379	99.3213

Comparatively, these values for order $n = 20$ (but computed with a too small value for s, chosen as 21) were:

	Without predictor	With predictor
mean(VAF)	17.0962	94.0542
min(VAF)	-74.6196	80.5374

which show that the 20-order model without predictor (determined for $s = 21$) is not good enough.

Using the SLIDENT commands below, one can check models of various orders, from 4 to $\min\{40, s-1\}$, with step 2, without and with Kalman predictor. The system is stable for all these orders. The trajectories are optionally plotted.

```
n_max  = min( 40, s - 1 );
list_n = [4 : 2 : n_max];  withK = 1;
[errs,VAFs]   = find_models(y,u,R,list_n);
[errsk,VAFsk] = find_models(y,u,R,list_n,withK);
```

[4]A perfect fit has VAF = 100 %.

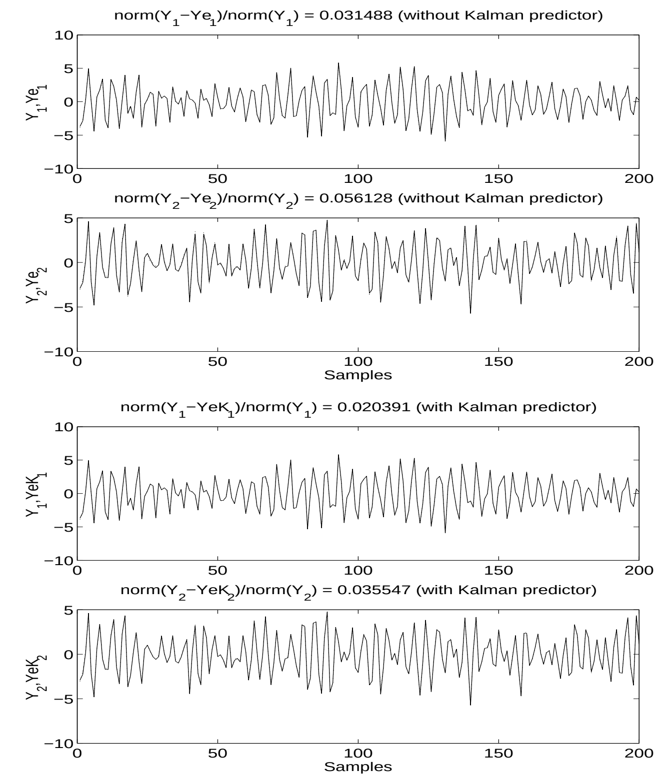

Fig. 4. The beginning 200 samples of the first two original and estimated output trajectories for the steel subframe flexible structure application without (top half) and with predictor (bottom half).

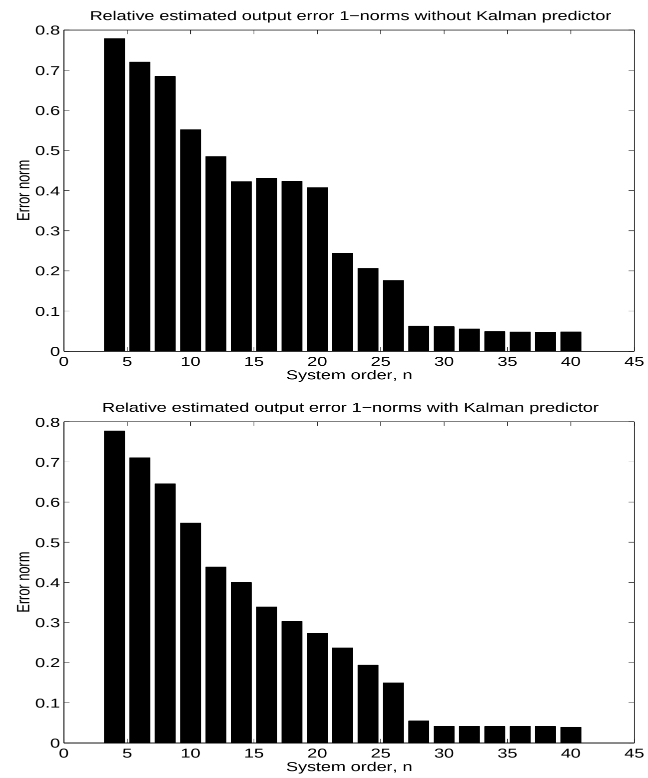

Fig. 5. Relative output errors for steel subframe flexible structure application for various orders ($s = 61$), without predictor (top) and with predictor (bottom).

These commands use the already computed matrix R, so that models of various orders are obtained quickly. Plotting the relative error 1-norms and the VAF values for all orders in `list_n` shows that the goodness of fit increases for increasing n, but no significant benefit could be obtained for n larger than 28 (but $n \leq 40$). The following MATLAB commands can be used

```
plot(list_n,errs);  plot(list_n,errsk);
plot(list_n,VAFs);  plot(list_n,VAFsk)
```

The relative output errors without and with a Kalman predictor are displayed in Figure 5, as bar graphs, and the VAF values are plotted similarly in Figure 6. Note that for each value in `list_n` of the system order, there are 28 vertical bars, differing in a gray scale. The mean and minimum VAF values, for all outputs and orders, were larger than 74.89 and 9.5, respectively, without predictor, and than 77.83 and 8.83, respectively, with predictor. Small VAF values arise, of course, for very small system orders. The models with order larger than 20 give good relative output errors and VAF values. Finally, Figure 7 represents the VAF values as trajectories depending on the system order. There are 28 trajectories, corresponding to each output of the system. This figure clearly indicates that some outputs are not so well identified if n is small, but for n approaching 28 all outputs are perfectly fitted.

III. Estimation of a Wiener System for the Flexible Structure Application

A discrete-time Wiener system, consisting in a linear part and a static nonlinearity, can be represented by

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ z(k) &= Cx(k) + Du(k), \\ y(k) &= f\,(z(k)) + v(k), \end{aligned} \qquad (3)$$

where $x(k)$, $u(k)$, and $y(k)$ are as for (1), $z(k)$ is the output of the linear part, and $f(\cdot)$ is a nonlinear vector function from $\mathbb{R}^\ell$ to $\mathbb{R}^\ell$. For identification, the linear part, found by subspace techniques, is parameterized using the output normal form [9]; the nonlinearity is modelled by a set of ℓ single layer neural networks,

$$f_r\,(z(k)) = \hat{f}_r\,(z(k)) + \epsilon_r(k)\,, \qquad r = 1, \ldots, \ell\,, \qquad (4)$$

$$\begin{aligned} \hat{f}_r\,(z(k)) \;:=\; &\sum_{i=1}^{\nu} \left(\alpha(r,i)\phi \left(\sum_{j=1}^{\ell} \beta(r,i,j) z_j(k) + b(r,i) \right) \right) \\ &+ b(r, \nu+1)\,, \end{aligned} \qquad (5)$$

where $f_r(\cdot)$ and $z_r(k)$ denote the r-th entry of the vector function $f(\cdot)$ and the vector $z(k) := z_k$, respectively, the vector $\epsilon(k)$ is the approximation error, the integer ν represents the number of neurons, and the coefficients $\alpha(r,i)$, $\beta(r,i,j)$, $b(r,i)$ and $b(r,\nu+1)$ are real numbers

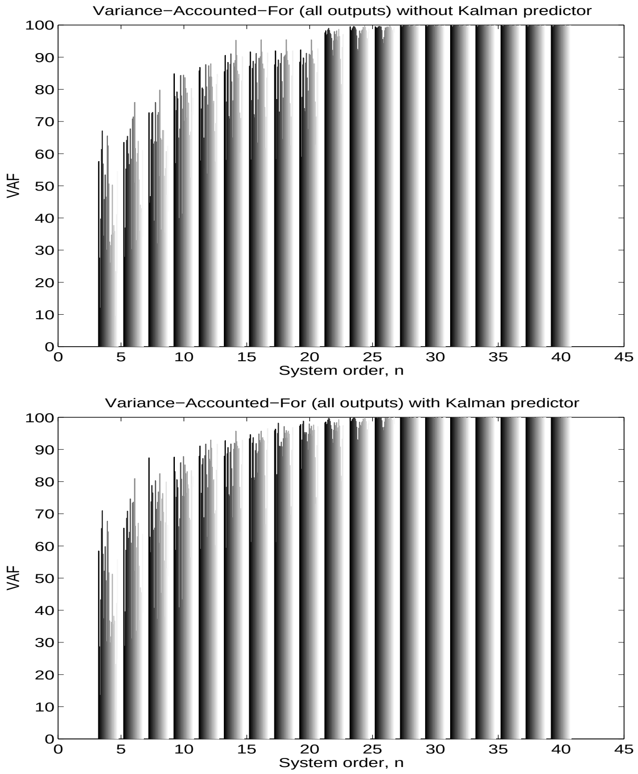

Fig. 6. Variance Accounted For (VAF) for steel subframe flexible structure application, for various orders ($s = 61$), without predictor (top) and with predictor (bottom).

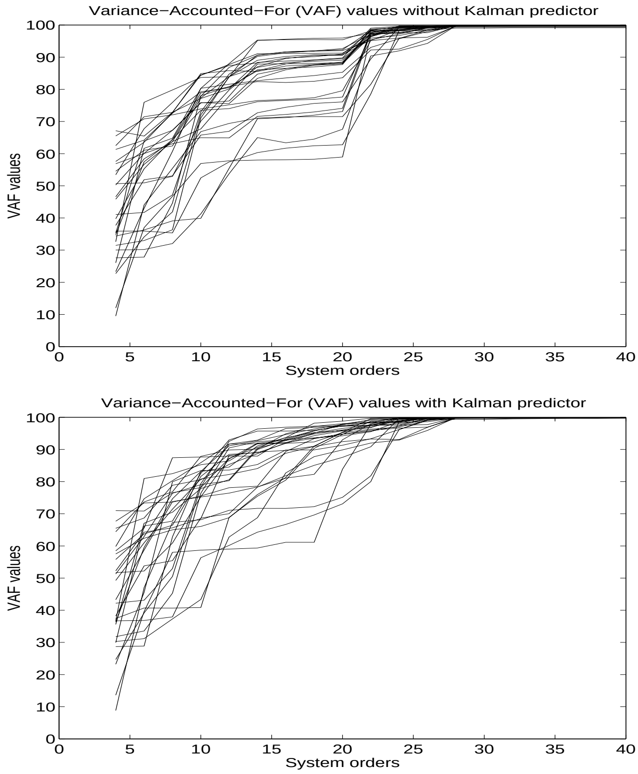

Fig. 7. Variance Accounted For (VAF) trajectories for steel subframe flexible structure application, for various orders ($s = 61$), without predictor (top) and with predictor (bottom).

to be estimated. The estimation problem is formulated as a structured nonlinear least squares problem which is solved in three steps (see [10] and the references therein). The first step identifies the linear part, using a subspace-based approach. The second step finds initial values of the weighting coefficients parameterizing the functions $\hat{f}_r$ in (5). (The hyperbolic tangent is used in implementation as a nonlinear function ϕ.) Specifically, all constants α, β, and b in (5) are stacked in the parameter vector θ, where $\theta = \left(\theta_1^T \,|\, \theta_2^T \,|\, \cdots \,|\, \theta_\ell^T \right)^T \in \mathbb{R}^{\ell((\ell+2)\nu+1)}$, and are estimated by solving the following nonlinear least squares problem

$$\min_{\theta} \sum_{k=1}^{N} \left\| \left[\begin{array}{c} y_1(k) - \hat{y}_1(k) \\ \vdots \\ y_\ell(k) - \hat{y}_\ell(k) \end{array} \right] \right\|^2, \qquad (6)$$

with $\hat{y}_r(k) := \hat{f}_r(\hat{z}_k)$, where $\hat{z}_k$ is the estimated output of the linear part. Clearly, (6) is equivalent to ℓ independent nonlinear least squares problems, which are solved separately. Finally, the parameters of the linear and nonlinear parts are refined by optimization calculations, starting with values corresponding to the results of the first two steps. Adding the parameters corresponding to the linear part at the end of the vector θ, the Jacobian matrix of the overall optimization problem has a block diagonal form with an additional right block column. Two specialized implementations of the Levenberg-Marquardt algorithm are provided: a standard implementation, which uses Cholesky factorization, or a conjugate gradients algorithm, for solving the symmetric positive-definite linear systems involved, and a MINPACK-like [8], but LAPACK-based, structure-exploiting implementation, which uses QR factorization.

Since the optimization problem for our application is too large for standard workstations, a simplified problem has been solved. Specifically, only the first half of the input and output data samples have been used for estimation (but all data samples were used for validation), the first 7 outputs only have been modelled, the system order was taken as $n = 20$, and the number of neurons for each output was chosen as 12. Even with this significant reduction in complexity, the corresponding optimization problem has 977 variables, and $7 \times \lfloor 8523/2 \rfloor = 29827$ nonlinear error functions.

The execution times needed for Wiener system identification were 7956.51, 3481.84, and 98595.72 seconds, for the QR-based, Cholesky-based, and conjugate gradients-based implementations, respectively. The "sum of squares" values, which the optimization algorithms minimize, were 155, 179, and 155, respectively, hence, the faster Cholesky implementation was somewhat less accurate. The error norms for all data samples were 225, 249, and 226, respectively, compared to 1000, for the linear model. Hence, again, the Cholesky implementation was less accurate.

The mean values of errors for linear and Wiener identification are plotted in Figure 8. The means have been computed on a moving window with a length of 40 samples.

The trajectories for Wiener system identification have been obtained using the algorithm based on structured QR factorization (with block-column pivoting). It is apparent that the Wiener model significantly reduces the prediction error, and has a smoothing effect. It is, however, expected that such improvements would not be possible for the model of order $n = 28$, estimated using $s = 61$, since the corresponding linear model is in that case very accurate.

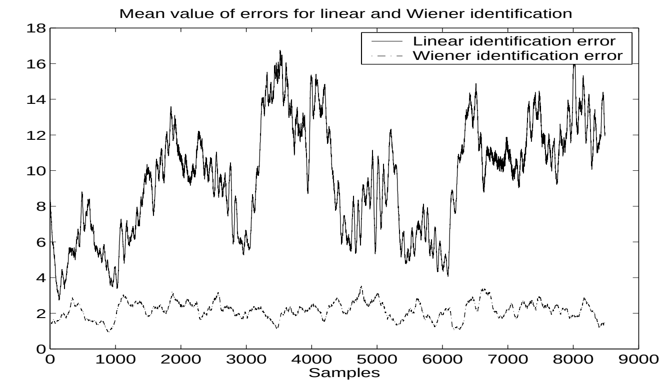

Fig. 8. The mean values of errors for linear and Wiener identification for steel subframe flexible structure application (the first 7 outputs only); the first half of the data set is used for estimation.

IV. Conclusions

System identification algorithms implemented in the recently developed toolbox SLIDENT, incorporated in the Fortran 77 library SLICOT, have been used for modelling a steel subframe flexible structure application. The toolbox provides a convenient and easy-to-use MATLAB interface. Using this interface and based on the incorporated fast, structure-exploiting algorithms, complex identification experiments can be quickly performed. The results show that SLIDENT is reliable and able to solve large identification problems.

Acknowledgement

The author is indebted to the anonymous reviewer for suggestions to improve the original manuscript.

References

[1] M. Abdelghani, M. Basseville, and A. Benveniste, "In-operation damage monitoring and diagnosis of vibrating structures, with application to offshore structures and rotating machinery", IMAC-XV, Feb. 3–6 1997, FL USA. Cited in [5].

[2] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide: Third Edition*, Software · Environments · Tools. SIAM, Philadelphia, 1999.

[3] P. Benner, V. Mehrmann, V. Sima, S. Van Huffel, and A. Varga, "SLICOT — A subroutine library in systems and control theory," in *Applied and Computational Control, Signals, and Circuits*, B.N. Datta, Ed., vol. 1, chapter 10, pp. 499–539. Birkhäuser, Boston, 1999.

[4] C. Gomez (Ed.), *Engineering and Scientific Computing with SciLab*, Birkhäuser, Boston, 1999.

[5] De Moor, B.L.R. (ed.), "DaISy: Database for the Identification of Systems," Department of Electrical Engineering, ESAT/SISTA, K.U. Leuven, Belgium. Available from URL: `http://www.esat.kuleuven.ac.be/sista/daisy/`.

[6] N. Mastronardi, D. Kressner, V. Sima, P. Van Dooren, and S. Van Huffel, "A fast algorithm for subspace state-space system identification via exploitation of the displacement structure," *J. Comput. Appl. Math.*, vol. 132, no. 1, pp. 71–81, 2001.

[7] The MathWorks, Inc, 24 Prime Park Way, Natick, Mass., 01760–1500, *Using MATLAB. Version 5*, 1999.

[8] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, "User's guide for MINPACK-1," Report ANL-80-74, Applied Math. Division, Argonne National Laboratory, Argonne, Illinois, 1980.

[9] R. Peeters, B. Hanzon, and M. Olivi. Balanced realizations of discrete-time stable all-pass systems and the tangential Schur algorithm. In *Proceedings of the European Control Conference*, 31 August–3 September 1999, Karlsruhe, Germany, 1999. Session CP-6, Discrete-time Systems.

[10] R. Schneider, A. Riedel, V. Verdult, M. Verhaegen, and V. Sima, "SLICOT system identification toolbox for nonlinear Wiener systems," SLICOT Working Note 2002-6, Katholieke Universiteit Leuven (ESAT/SISTA), Leuven, Belgium, June 2002. Available from `ftp://wgs.esat.kuleuven.ac.be/pub/WGS/REPORTS/`, file `SLWN2002-6.ps.Z`, 26 pages.

[11] V. Sima, "Algorithms and LAPACK-based software for subspace identification," in *Proceedings of The 1996 IEEE International Symposium on Computer-Aided Control System Design*, September 15–18, 1996, Dearborn, MI, U.S.A., 1996, pp. 182–187.

[12] V. Sima, "Subspace-based algorithms for multivariable system identification," *Studies in Informatics and Control*, vol. 5, no. 4, pp. 335–344, 1996.

[13] V. Sima, "Cholesky or QR factorization for data compression in subspace-based identification ?," in *Proceedings of the Second NICONET Workshop on "Numerical Control Software: SLICOT, a Useful Tool in Industry"*, December 3, 1999, INRIA Rocquencourt, France, 1999, pp. 75–80.

[14] V. Sima, "SLICOT linear systems identification toolbox," SLICOT Working Note 2000-4, Katholieke Universiteit Leuven (ESAT/SISTA), Leuven, Belgium, July 2000. Available from `ftp://wgs.esat.kuleuven.ac.be/pub/WGS/REPORTS/`, file `SLWN2000-4.ps.Z`, 30 pages.

[15] V. Sima, D. M. Sima, and S. Van Huffel, "SLICOT system identification software and applications," in *Proceedings of the 2002 IEEE International Conference on Control Applications and IEEE International Symposium on Computer Aided Control System Design, CCA/CACSD 2002*, September 18–20, 2002, Scottish Exhibition and Conference Centre, Glasgow, Scotland, U.K., P.R. Kalata (Ed.). Sept. 2002, pp. 45–50, Omnipress.

[16] V. Sima and S. Van Huffel, "SLICOT subspace identification toolbox," in *Proceedings CD of the UKACC International Conference on Control 2000*, University of Cambridge, United Kingdom, 4-7 September, 2000. 6 pages.

[17] V. Sima and S. Van Huffel, "Efficient numerical algorithms and software for subspace-based system identification," in *Proceedings of the 2000 IEEE International Conference on Control Applications and IEEE International Symposium on Computer-Aided Control Systems Design*, September 25–27, 2000, Anchorage, AK, U.S.A., Dr. L.K. Mestha (Ed.). Wilson Center for Research & Technology, Xerox Corporation, 800 Phillips Rd. MS 128-56E, Webster, NY 14580, Sept. 2000, pp. 1–6, Omnipress.

[18] V. Sima and S. Van Huffel, "Performance investigation of SLICOT system identification toolbox," in *Proceedings of the European Control Conference, ECC 2001*, 4–7 September, 2001, Seminário de Vilar, Porto, Portugal, Sept. 2001, pp. 3586–3591.

[19] T. Van Gestel, J.A.K. Suykens, P. Van Dooren, and B. De Moor, "Identification of stable models in subspace identification by using regularization," *IEEE Trans. Automat. Control*, vol. 46, no. 9, pp. 1416–1420, 2001.

[20] 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.

[21] P. Van Overschee and B. De Moor, *Subspace Identification for Linear Systems : Theory – Implementation – Applications*, Kluwer Academic Publishers, Boston/London/Dordrecht, 1996.

[22] 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.

[23] 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.