EFFICIENT IMPLEMENTATION OF SEPARABLE LEAST SQUARES FOR THE IDENTIFICATION OF COMPOSITE LOCAL LINEAR STATE-SPACE MODELS

José Borges [*] **Vincent Verdult** [**] **Miguel Ayala Botto** [*]

[*] *Technical University of Lisbon, Instituto Superior Técnico*
Department of Mechanical Engineering, GCAR
Avenida Rovisco Pais, 1049-001 Lisboa, Portugal,
Email: jborges@ist.utl.pt
[**] *Delft Center for Systems and Control,*
Delft University of Technology, The Netherlands,
Email: v.verdult@dcsc.tudelft.nl

Abstract: The efficient implementation of separable least squares identification of nonlinear systems using composite local linear state-space models is discussed in this paper. A full parametrization of system matrices combined with projected gradient search is used to identify the model. This combined approach reduces the number of iterations and improves the numerical condition of the optimization algorithm. Further enhancements result from using numerical tools, e.g. the QR-decomposition, and efficient approaches to compute the separable least squares matrices and gradients of the cost functions. Monte-Carlo simulations are used to show the effectiveness of the proposed approach.

Keywords: identification, nonlinear systems, state-space models, least squares algorithm, efficient algorithms

1. INTRODUCTION

A feasible methodology for system identification has to generate reliable models, which are capable of reproducing the dynamical behavior of a system with given accuracy, while allowing the efficient implementation of its algorithms. Considering the available computational resources, the *efficiency* of an implementation is characterized by the computational burden/time that is used to complete a given identification experiment while handling arbitrarily large data-sets.

The main contribution of this paper is, therefore, the development of computationally efficient and numerically reliable tools to implement the principle of Separable Leats Squares (SLS) for the identification of composite local linear state-space models. The method is based on the use of numerically efficient tools, e.g. the QR-decomposition, and efficient algorithms to compute the gradients for the cost functions and the main matrices of the SLS method. As discussed e.g. in Bergboer et al. (2002), the use of tools like the QR-decomposition allows a straightforward handling of large data sets while avoiding computational problems arising from dimensionality growth.

Verdult et al. (2002) proposed to use composite local linear state-space models to approximate nonlinear systems. This approach fits into the broader class denominated by operating regime decomposition methods (Murray-Smith and Johansen 1997) and it can be interpreted as the division of the operating range of the nonlinear system into smaller regimes, in which the nonlinear system is approximated by a linear model.

Recently, Borges et al. (2004) proposed an approach, which consists of combining the principle of SLS with the projected gradient search method, for the identification of composite local linear state-space models. Similar approaches have been proposed by Verdult (2002) in the case of bilinear state-space systems, and by Ribarits et al. (2004) in the case of linear time-invariant systems.

The principle of SLS (Golub and Pereyra 1973) allows to solve nonlinear optimization problems in two consecutive steps: initially part of the parameters are estimated using a nonlinear optimization procedure. Afterwards, the remaining parameters are estimated using linear least squares optimization and the knowledge of the optimal parameters estimated in the first step. A key issue is the definition of a mapping from the space of parameters estimated by linear least squares into the space of parameters estimated using nonlinear optimization. Bruls et al. (1999) proposed the identification of linear time-invariant systems using the principle of SLS and specific parameterizations of system matrices.

The concept of projected gradient search (McKelvey and Helmersson 1997; Lee and Poolla 1999) is used here to deal with the problem of parameterizing the system matrices. Rather than choosing a specific parametrization, this method allows to use fully parameterized system matrices by restricting the update parameters in the optimization process to directions that result in an effective update of the cost function.

This paper is organized as follows: Section 2 briefly reviews the use of SLS combined with projected gradient search for the identification of composite local linear state-space models. Section 3 discusses an efficient implementation of this methodology, as well as some efficient tools to compute the gradients for the cost functions. The discussion of these efficient implementations is presented in Section 4 using simulations results. Finally, in Section 5 some conclusions are drawn.

2. COMPOSITE LOCAL LINEAR MODELS

Similarly to Verdult et al. (2002), the composite local linear state-space models are defined as:

$$x_{k+1} = \sum_{i=1}^{s} p_i\left(\phi_k\right)\left(A_i x_k + B_i u_k + O_i\right) \qquad (1)$$

$$y_k = C x_k + v_k \qquad (2)$$

where s is the number of local models, $x_k \in \mathbb{R}^n$ is the state vector, $u_k \in \mathbb{R}^m$ is the input, $y_k \in \mathbb{R}^\ell$ is the output, $v_k \in \mathbb{R}^\ell$ is a white-noise sequence and $p_i\left(\phi_k\right) \in \mathbb{R}^s$ are the weighting vectors. The weighting vectors $p_i\left(\phi_k\right) \in \mathbb{R}^s$ are parameterized using normalized radial basis functions:

$$p_i\left(\phi_k\right) = \frac{r_i\left(\phi_k; c_i, w_i\right)}{\sum_{j=1}^{s} r_j\left(\phi_k; c_j, w_j\right)} \qquad (3)$$

with,

$$r_i = \exp\left(-\left(\phi_k - c_i\right)^T diag\left(w_i\right)^2\left(\phi_k - c_i\right)\right) \qquad (4)$$

where $\phi_k \in \mathbb{R}^q$ is the scheduling vector, c_i is the center and w_i the width of the i-th radial basis function.

The goal is to determine, from a finite number of measurements of the input u_k and output y_k, the matrices A_i, B_i, O_i, C, and the centers c_i and widths w_i that describe the radial basis functions. Let the vectors c and w consist of the centers c_i and w_i, respectively. Let θ parameterize the system matrices A_i, B_i, O_i and C. The estimation of parameters θ, c and w, is based on the minimization of the following cost function:

$$J_N(\theta, c, w) = E_N^T E_N \qquad (5)$$

where $E_N = Y_N - \widehat{Y}_N$ denotes the error vector, with Y_N a vector containing N samples of the measured outputs and $\widehat{Y}_N$ a vector containing the outputs of the estimated model.

2.1 Separable least squares

Given the state-space system (1)–(2), the estimated output for a given time instant k is given by:

$$\hat{y}_k = C\left(\prod_{i=0}^{k-1}\mathcal{A}_i\right)x_0 + \sum_{j=0}^{k-1} C\left(\prod_{h=j+1}^{k-1}\mathcal{A}_h\right) G\left(p(\phi_j)\otimes\tilde{u}_j\right) \qquad (6)$$

where the symbol $\otimes$ denotes the Kronecker product, $\mathcal{A}_k := \sum_{i=1}^{s} A_i p_i(\phi_k)$, $\widetilde{u}_j := \left[u_j^T\ 1\right]^T$, and

$$p(\phi_k) := \left[p_1(\phi_k)\ p_2(\phi_k)\ \cdots\ p_s(\phi_k)\right]$$
$$G := \left[B_1\ O_1\ B_2\ O_2\ \ldots\ B_s\ O_s\right]$$

Equation (6) can be further simplified by considering $k \in [1, \ldots, N]$ and assuming the initial state equal to zero, as follows,

$$\widehat{Y}_N = \Phi\left(\eta\right)\theta_\ell \qquad (7)$$

where

$$\eta := \begin{bmatrix}\theta_n \\ w \\ c\end{bmatrix}, \quad \theta_\ell := \text{vec}\left(G\right), \quad \theta_n := \begin{bmatrix}\text{vec}\left(A\right) \\ \text{vec}\left(C\right)\end{bmatrix}$$

with $A := \left[A_1\ A_2\ \ldots\ A_s\right]$. Matrix $\Phi\left(\eta\right)$ is defined as,

$$\Phi\left(\eta\right) := \begin{bmatrix} 0 \\ \left(p(\phi_0)\otimes\widetilde{u}_0\right)^T \otimes C \\ \vdots \\ \sum_{j=0}^{N-2}\left(p(\phi_j)\otimes\widetilde{u}_j\right)^T \otimes C\left(\prod_{h=j+1}^{N-2}\mathcal{A}_h\right)\end{bmatrix} \qquad (8)$$

From (7) E_N can be written as,

$$E_N = Y_N - \Phi(\eta)\theta_\ell \qquad (9)$$

For a fixed η, minimizing the norm of E_N with respect to θ_ℓ yields

$$\widehat{\theta}_\ell\left(\eta\right) = \Phi(\eta)^\dagger Y_N \qquad (10)$$

with $\Phi^\dagger = \left(\Phi^T\Phi\right)^{-1}\Phi^T$. By evaluating equation (9) for $\theta_\ell = \widehat{\theta}_\ell\left(\eta\right)$, it is possible to define a new error vector,

$$\widetilde{E}_N\left(\eta\right) = E_N\left(\theta_\ell, \eta\right)\Big|_{\theta_\ell = \widehat{\theta}_\ell(\eta)} = \left(I - P(\eta)\right)Y_N \qquad (11)$$

where $P(\eta) = \Phi(\eta)\Phi(\eta)^{\dagger}$. The principle of SLS from Golub and Pereyra (1973) states that the optimization problem of estimating $\left(\widehat{\eta}, \widehat{\theta}_\ell\right) = \left(\widehat{\theta}, \widehat{w}, \widehat{c}\right)$ using the cost function (5) is equivalent to, first, estimate $\widehat{\eta}$ by minimizing

$$\widetilde{J}_N(\eta) := \widetilde{E}_N^T \widetilde{E}_N \tag{12}$$

which is independent of θ_ℓ and, then, estimate $\widehat{\theta}_\ell$ using the linear least squares optimization (10).

2.2 Projected gradient search

Full parameterized A and C matrices are used along the optimization process. A major drawback of using full parameterized matrices for the identification of composite local linear state-space models, is the redundancy that is introduced in the optimization procedure. This redundancy results from the nonuniqueness of the cost function values (Borges et al. 2004; Lee and Poolla 1999; McKelvey and Helmersson 1997). A way to cope with this problem is to use projected gradient search. Consider the following matrix

$$M_{(A,C)} := \sum_{i=1}^{s} \Pi_i^T \otimes \begin{bmatrix} A\Pi_i^T \\ C\Pi_i^T \end{bmatrix} - A \otimes \begin{bmatrix} I_n \\ 0_{\ell\times n} \end{bmatrix} \tag{13}$$

where $\Pi_i := \begin{bmatrix} 0_{n\times(i-1)n} & I_n & 0_{n\times(s-i)n} \end{bmatrix}$. Assuming (A_i, C) is observable, with $i \in \{1, \ldots, s\}$, then the matrix $M_{(A,C)}$ has full column rank and its left null-space $\mathcal{Q}_2$ is given as

$$M_{(A,C)} = \begin{bmatrix} \mathcal{Q}_1 & \mathcal{Q}_2 \end{bmatrix} \begin{bmatrix} \mathcal{R}_1 \\ 0 \end{bmatrix} \tag{14}$$

The matrix $\mathcal{Q}_2$ can be used in a projected gradient search using a Levenberg-Marquardt type of update equation $\eta^{(i+1)} = \eta^{(i)} + d^{(i)}$. Taking into account the projection of the gradient, the parameter update is given by:

$$d^{(i)} = -\overline{\mathcal{Q}}_2 \left(\overline{\mathcal{Q}}_2^T \Psi_N^T \Psi_N \overline{\mathcal{Q}}_2 + \lambda I\right)^{-1} \overline{\mathcal{Q}}_2^T \Psi_N^T E_N \tag{15}$$

with,

$$\Psi_N := \frac{\partial E_N}{\partial \eta^T}, \quad \overline{\mathcal{Q}}_2 := \begin{bmatrix} \mathcal{Q}_2 & 0 \\ 0 & I \end{bmatrix}^T$$

Note that $\mathcal{Q}_2$ only operates on the θ_n part of η.

3. EFFICIENT IMPLEMENTATION

In Section 2 the combined use of separable least squares with projected gradient is proposed for the identification of composite local linear models. In the present section the efficient implementation of such approach is discussed. This discussion will consider the computation of matrix Φ, including the handling of large data sets, and the computations of the value and gradients for the SLS cost function.

3.1 Computation of the matrix Φ

Kronecker products are used in equation (8) to compute the matrix $\Phi(\eta)$. An efficient implementation should avoid the direct use of such products. Below an alternative method to compute the columns of Φ by simulating a set of linear systems is proposed. The output of the state-space model (1)–(2) can be written as:

$$y_k = C\left(\prod_{i=0}^{k-1} \mathcal{A}_i\right) x_0 + \sum_{\alpha=1}^{ms}\sum_{\beta=1}^{n}\left(\underbrace{\sum_{j=0}^{k-1}\left(C\prod_{h=j+1}^{k-1}\mathcal{A}_h\right) E_{\alpha,\beta}\left(p_j \otimes \tilde{u}_j\right)}_{y_{\alpha,\beta}(k)}\right) b_{\alpha,\beta} \tag{16}$$

where $E_{\alpha,\beta}$ is an auxiliary input matrix, with dimensions $n \times (m+1)s$, that is equal to zero everywhere, except for $E_{\alpha,\beta}(i,j)|_{(i=\alpha, j=\beta)} = 1$. The terms in equations (16) and (6) are related such that, if several realizations of $y_{\alpha,\beta}(k)$ are stacked in a column-wise manner, and the values of $b_{\alpha,\beta}$ are stacked as a column-vector, then the following construction for matrices Φ and θ_ℓ of equation (7) results:

$$\Phi(\eta) = \begin{bmatrix} y_{1,1} & y_{2,1} & y_{1,2} & \cdots & y_{n,s(m+1)} \end{bmatrix} \tag{17}$$

$$\theta_\ell = \begin{bmatrix} b_{1,1} & b_{2,1} & b_{1,2} & \ldots & b_{n,s(m+1)} \end{bmatrix}^T \tag{18}$$

Therefore, Φ can be constructed by performing $(m+1)ns$ simulations of the following systems:

$$x_{\alpha,\beta}(k+1) = \sum_{i=1}^{s} p_i(\phi_k)\left(A_i\, x_{\alpha,\beta}(k) + E_{\alpha,\beta}\widetilde{u}_k\right) \tag{19}$$

$$y_{\alpha,\beta}(k) = C x_{\alpha,\beta}(k) \tag{20}$$

where $x_{\alpha,\beta}(0) = 0$ are used as initial conditions, and the pair of indices (α, β) varies as indicated in (17).

3.2 Computation of matrices with large data sets

Matrix $\Phi(\eta)$ has dimensions $Nl \times ns(m+1)$. Therefore, depending on the number of variables and data samples, the handling of $\Phi(\eta)$ along the optimization process can be troublesome. A way to cope with this dimensionality problem is to factorize the matrix $\Phi(\eta)$ using, e.g., the QR-decomposition of $\Phi(\eta)$, as follows,

$$\Phi = QR = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1 R_1 \tag{21}$$

Using matrices Q_1 and R_1, the computation of e.g. the vector of parameters $\widehat{\theta}_\ell$ is quite straightforward,

$$\widehat{\theta}_\ell = \Phi^{\dagger} Y_N = R_1^{-1} Q_1^T Y_N \tag{22}$$

This approach can be further refined if the following QR-decomposition is considered,

$$[\Phi \; Y_N] = \overline{Q}\,\overline{R} = \begin{bmatrix} Q_1 & \overline{Q}_2 \end{bmatrix} \begin{bmatrix} R_1 & r \\ 0 & 0 \end{bmatrix} \tag{23}$$

According to (Golub and Van Loan 1996), an estimative for the number of flops used in this factorization is presented in Table 1. The optimum value for θ_ℓ is now given as,

$$\widehat{\theta}_\ell = R_1^{-1} r \tag{24}$$

where $r = Q_1^T Y_N$. The inversion of R_1 in (24) is computationally expensive and should be avoided. Since R_1 results from the QR-decomposition directly with an upper triangular shape, $\widehat{\theta}_\ell$ can be computed using backward replacement in the following equation,

$$R_1 \widehat{\theta}_\ell = r \tag{25}$$

When huge data sets are used, the computation of $\Phi\left(\eta\right)$, or its QR-decomposition, may render the identification procedure unfeasible. In these cases it is still possible to overcome this problem by exploiting the special structure proposed in Section 3.1 to compute $\Phi\left(\eta\right)$ using an iterative procedure. The approach then is to compute the matrix $\Phi\left(\eta\right)$ by block-rows of size N_b and, from one batch computation to the other, to store the last value of the state vector $x_{\alpha,\beta}$ and the matrices Q_1 and R_1, as follows:

- The value of the state vector computed in the last iteration of the previous batch computation is stored, and then used as the initial state for the next batch computations, i.e.,

$$x_{\alpha,\beta}^{\text{batch }\#(j+1)}\left(0\right) = x_{\alpha,\beta}^{\text{batch }\#(j)}\left(N_b\right)$$

- The matrices Q_1 and R_1 are updated using the iterative procedure described in Algorithm 1, where MATLAB based notation and functions are used.

Algorithm 1: Iterative procedure to compute the matrices Q_1 and R_1.

```
$[Q_1, R_1] = \mathbf{qr}(\Phi(1:N_b,:),0)$
$[n_r, n_c] = \mathbf{size}(R_1)$
for $i = N_b + 1 : N_b : N - N_b$
    $[Q_a, R_1] = \mathbf{qr}([R_1; \Phi(i:i+N_b,:)],0)$
    $Q_1 = [Q_1 * Q_a(1:n_r,:); Q_a(n_r+1:end,:)]$
end
```

Due to the computational burden that results from computing the blocks-rows of matrix Φ, and their QR-decomposition, the procedure described in Algorithm 1 should be used only if a huge number of data samples is to be used. Actually, this iterative approach is efficient only in those cases when the limit of the computer physical memory is reached and the software has to run using the virtual, or swap, memory. In this cases, the delay due to the access to the swap medium causes an increase in the computational time, which necessarily results in a decrease of performance for the algorithm. In these cases the iterative procedure is much more efficient than the non-iterative procedure.

3.3 Computation of the SLS-cost function value

Using the QR-decomposition of Φ, which is given in (23), the computation of the SLS-cost function value becomes straightforward. The error vector can be computed as follows,

$$\widetilde{E}_N = Y_N - Q_1 Q_1^T Y_N = Y_N - Q_1 r \tag{26}$$

where the products are performed from right-to-left. The cost is then computed by replacing the result of (26) in equation (12).

3.4 Computation of gradients

A main step in any gradient based optimization is the computation of cost function gradients. Since a least squares optimization problem is considered in this paper, the computation of gradients for the error vector are rather treated. The relation of gradients for the cost function and the error vector is the following:

$$\frac{\partial \widetilde{J}_N}{\partial \eta_j} = \widetilde{E}_N \frac{\partial \widetilde{E}_N}{\partial \eta_j} \tag{27}$$

In this part two distinct approaches to compute the gradients of $\widetilde{E}_N$ are proposed:

(1) To compute directly the gradient of $\widetilde{E}_N$;

(2) To compute the gradient of $\widetilde{E}_N$ using the gradient of E_N and computing $\widehat{\theta}_\ell$.

Computing the gradients using Approach (1) The gradient of $\widetilde{E}_N$ with respect to η_j is given by:

$$\begin{aligned}\frac{\partial \widetilde{E}_N}{\partial \eta_j} &= -\left(I - P\right) \frac{\partial \Phi}{\partial \eta_j} \Phi^{\dagger} Y_N \\ &\quad - \left(\Phi^{\dagger}\right)^T \frac{\partial \Phi^T}{\partial \eta_j} \left(I - P\right) Y_N\end{aligned} \tag{28}$$

A more efficient procedure to compute the gradient of $\widetilde{E}_N$ is to use,

$$\begin{aligned}\frac{\partial \widetilde{E}_N}{\partial \eta_j} &= -\left(I - Q_1 Q_1^T\right) \frac{\partial \Phi}{\partial \eta_j} R_1^{-1} Q_1^T Y_N \\ &\quad - \left(R_1^{-1} Q_1^T\right)^T \frac{\partial \Phi^T}{\partial \eta_j} \left(I - Q_1 Q_1^T\right) Y_N\end{aligned} \tag{29}$$

where $\Phi^{\dagger}$ is computed as in (22) and $P = Q_1 Q_1^T$. The products are calculated from right-to-left.

Based on the efficient approach introduced in Section 3.1, the gradients of $\Phi\left(\eta\right)$ are computed using,

$$\frac{\partial \Phi}{\partial \eta_j} = \left[\, Y_{1,1}^{\eta_j} \; Y_{2,1}^{\eta_j} \; Y_{1,2}^{\eta_j} \; Y_{2,2}^{\eta_j} \; \cdots \; Y_{n,(m+1)s}^{\eta_j} \,\right] \tag{30}$$

Each column of (30) is the result of simulating the following state-space system:

$$\begin{aligned}X_{\alpha,\beta}^{\eta_j}\left(k+1\right) &= \sum_{i=1}^{s} \frac{\partial p_i\left(\phi_k\right)}{\partial \eta_j} \left(A_i \, x_{\alpha,\beta}\left(k\right) + E_{\alpha,\beta} \widetilde{u}_k\right) \\ &\quad + p_i\left(\phi_k\right) A_i X_{\alpha,\beta}^{\eta_j}\left(k\right) + p_i\left(\phi_k\right) \frac{\partial A_i}{\partial \eta_j} x_{\alpha,\beta}\left(k\right)\end{aligned} \tag{31}$$

$$Y_{\alpha,\beta}^{\eta_j}\left(k\right) = \frac{\partial C}{\partial \eta_j} x_{\alpha,\beta}\left(k\right) + C \, X_{\alpha,\beta}^{\eta_j}\left(k\right) \tag{32}$$

where the following notation is used,

$$X_{\alpha,\beta}^{\eta_j}(k) = \frac{\partial x_{\alpha,\beta}(k)}{\partial \eta_j} \tag{33}$$

$$Y_{\alpha,\beta}^{\eta_j}(k) = \frac{\partial y_{\alpha,\beta}(k)}{\partial \eta_j} \tag{34}$$

the state $X_{\alpha,\beta}^{\eta_j}(0)$ in (31) is initialized with zero.

The computation of (30) can be done efficiently if the state-space system (31)–(32) is implemented sequentially by blocks according to the computed gradient, e.g., if η_j is an element of A then the computations in (31)–(32) simplify to,

$$X_{\alpha,\beta}^{\eta_j}(k+1) = \sum_{i=1}^{s} p_i(\phi_k) A_i X_{\alpha,\beta}^{\eta_j}(k) + p_i(\phi_k) \frac{\partial A_i}{\partial \eta_j} x_{\alpha,\beta}(k) \tag{35}$$

$$Y_{\alpha,\beta}^{\eta_j}(k) = C\, X_{\alpha,\beta}^{\eta_j}(k) \tag{36}$$

The same reasoning is applied to the computation of gradients with respect to the remaining variables of the parameter vector η. If the input data is used as scheduling vector in equations (1)–(2), i.e. $\phi_k = u_k$, then the computations of the gradients for $p_i(u_k)$ are simplified, as shown by Verdult (2002). An estimative of the number of flops used to compute the gradient (29) is presented in Table 1.

Computing the gradients using Approach (2) From equation (11) the following relation for the gradients of the error vectors trivially result:

$$\frac{\partial \widetilde{E}_N(\eta)}{\partial \eta_j} = \left. \frac{\partial E_N(\theta_\ell, \eta)}{\partial \eta_j} \right|_{\theta_\ell = \widehat{\theta}_\ell} \tag{37}$$

A similar expression is presented by Ribarits et al. (2004), but only for the case of LTI systems and using the cost functions. In equation (37) the gradient of $E_N(\theta_\ell, \eta)$ with respect to η_j is computed using the chain rule,

$$\frac{\partial E_N(\theta_\ell, \eta)}{\partial \eta_j} = \frac{\partial E_N(\theta_\ell, \eta)}{\partial \eta_j} + \frac{\partial E_N(\theta_\ell, \eta)}{\partial \theta_\ell} \frac{\partial \theta_\ell(\eta)}{\partial \eta_j} \tag{38}$$

Since the principle of SLS considers $\theta_\ell = \widehat{\theta}_\ell(\eta)$, the gradient of $E_N(\theta_\ell, \eta)$ with respect to θ_ℓ is equal to zero at the optimum $\widehat{\theta}_\ell(\eta)$. Thus (37) simplifies to:

$$\frac{\partial \widetilde{E}_N}{\partial \eta_j} = \frac{\partial E_N\left(\widehat{\theta}_\ell(\eta), \eta\right)}{\partial \eta_j} \tag{39}$$

Based on this equality, two distinct approaches to compute the gradients of $\widetilde{E}_N$ with respect to η_j are proposed:

a. To compute the gradient of E_N using,

$$\frac{\partial E_N\left(\widehat{\theta}_\ell(\eta), \eta\right)}{\partial \eta_j} = -\frac{\partial \Phi}{\partial \eta_j} \widehat{\theta}_\ell \tag{40}$$

where $\partial\Phi/\partial\eta_j$ is constructed as in the previous section and $\widehat{\theta}_\ell$ is computed using backward replacement in (25). An estimative of the number of flops used to compute (40) is presented in Table 1.

b. To compute the gradient of E_N using,

$$\frac{\partial E_N\left(\widehat{\theta}_\ell(\eta), \eta\right)}{\partial \eta_j} = -\frac{\partial \widehat{Y}_N}{\partial \eta_j} \tag{41}$$

The gradient of $\widehat{Y}_N$ consists of the outputs from the following state-space systems:

$$X^{\eta_j}(k+1) = \sum_{i=1}^{s} \frac{\partial p_i(\phi_k)}{\partial \eta_j} \left(A_i x_k + \widehat{B} u_k + \widehat{O}\right) + p_i(\phi_k) A_i X^{\eta_j}(k) + p_i(\phi_k) \frac{\partial A_i}{\partial \eta_j} x_k \tag{42}$$

$$\widehat{Y}^{\eta_j}(k) = \frac{\partial C}{\partial \eta_j} x(k) + C\, X^{\eta_j}(k) \tag{43}$$

where the following notation has been used,

$$X^{\eta_j}(k) = \frac{\partial x(k)}{\partial \eta_j} \tag{44}$$

$$\widehat{Y}^{\eta_j}(k) = \frac{\partial \hat{y}(k)}{\partial \eta_j} \tag{45}$$

and where $\widehat{B}$ and $\widehat{O}$ are obtained from $\widehat{\theta}_\ell(\eta)$.

These computations also require the state vector x_k, which results from simulating the composite local linear state-space model (1)–(2), where $B(\eta)$ is used instead. The matrix $B(\eta)$ is built up by reshaping the elements of $\widehat{\theta}_\ell$, which are computed at each iteration using backward replacement in (25). An estimative of the number of flops to compute (41) is presented in Table 1.

4. SIMULATION EXPERIMENTS

In this part Monte-Carlo experiments are used to show the performance of the implementations proposed in Section 3. Consider the following MIMO system from (Narendra and Parthasarathy 1990),

$$y_{k+1}^{(1)} = \frac{0.8\left(y_k^{(1)}\right)^3 + \left(u_k^{(1)}\right)^2 u_k^{(2)}}{2 + \left(y_k^{(2)}\right)^2}$$

$$y_{k+1}^{(2)} = \frac{y_k^{(1)} + y_k^{(1)} y_k^{(2)} + \left(u_{k-1}^{(1)} - 0.5\right)\left(u_k^{(2)} + 0.8\right)}{2 + \left(y_k^{(2)}\right)^2}$$

The system has, therefore, two inputs and two outputs. The input signals for the Monte-Carlo experiment, $u_k^{(1)}$ and $u_k^{(2)}$, are uncorrelated white-noise input sequences that are generated at each experiment. The initial models used in the output-error optimization algorithm were obtained using the PI-MOESP method of Verhaegen (1994). Initially, a global state-space linear model is estimated using this subspace method. Afterwards, each local linear model is initialized using this global linear model and the algorithm proceeds as described in Section 2. The weights were uniformly initialized in the interval between -1 and 1. The size of

Table 1. Estimative for the number of flops used in several computations of Section 3.

	Flops
Compute Φ, Eq. (17)	$2\ell\, nN\left((m+n+5)\,s+\ell\right)$
QR-decomposition, Eq. (23)	$2\ell\, nN\left(m^2ns^2\right)$
Compute $Y_{\alpha,\beta}^{\eta_j}$, Eq. (32), or $\widehat{Y}_N^{\eta_j}$, Eq. (43)	$2\ell\, nN\left(2sn^3+\left({}^3\!/\!_2 s+\ell\right)n^2+\ell^2n+(4n+2m)s^2\right)$
Compute $\widehat{\theta}_\ell$, Eq. (25)	$m^2n^2s^2$
Gradient using Appr. 1.), Eq (29)	$2\ell\, nN\left(\left(2sn^3+\left({}^3\!/\!_2 s+\ell\right)n^2+\ell^2n+(4n+2m)s^2\right)mns+8ms\right)$
Gradient using Appr. 2.a), Eq (40)	$2\ell\, nN\left(2sn^3+\left({}^3\!/\!_2 s+\ell\right)n^2+\ell^2n+(4n+2m)s^2\right)mns$
Gradient using Appr. 2.b), Eq (41)	$2\ell\, nN\left(2sn^3+\left({}^3\!/\!_2 s+\ell\right)n^2+\ell^2n+(4n+2m)s^2\right)$

Table 2. Mean times to compute the gradients using approaches of Section 3.4.

	Appr. 1	Appr. 2.a)	Appr. 2.b)
Mean time [s]	7.208	5.744	0.162

the data samples for each experiment is $N = 1\ 000$. At each experiment the value of the cost function and the gradients are computed and the computational-time it takes is stored. Similarly to Verdult (2002), a composite local linear state-space model consisting of four third-order local linear models was selected for the identification experiments. This means that there is a total of $\#(\eta) = 50$ variables for the nonlinear part of the SLS optimization, and $\#(\theta_\ell) = 36$ variables for the linear part. The experiments were simulated in MATLAB 6.5, over a Linux operating system. The processor is Pentium 4, 2.8 GHz, with 512 KB of RAM. Table 2 shows the mean computation times for the Monte-Carlo experiments. It is shown that Approach 2.b) is the methodology that presents better performance in the computation of gradients for the error vector. This result is coherent with the values for the flops presented in Table 1. The mean time of Approach 2.a) is higher than Approach 2.b) because the computation of the mns columns $Y_{\alpha,\beta}^{\eta_j}$, for matrix $\frac{\partial\Phi}{\partial\eta_j}$, constitutes the main computational burden in the algorithm. In Approach 1.), additionally to the burden of computing the gradient of Φ, also the projection described in (29) has to be computed, resulting in a even longer mean computation time for this approach.

5. CONCLUSIONS

An efficient implementation of the separable least squares identification of composite local linear state-space systems is described in this paper. Due to the large size of the SLS optimization matrices, the memory required by the algorithms, as well as the respective computational-times, are the bottleneck of an effective implementation of such approach. These problems were addressed in this paper using an implementation based on the use of efficient algorithms to compute the SLS matrices, as well as on the use of mathematically efficient tools to handle large data-sets in the identification procedure. Furthermore, also the computation of gradients for the cost functions was addressed and three different methods were proposed. By means of simulation experiments the most efficient method to compute the gradients for the SLS optimization method was selected. The final conclusion is that the proposed efficient approach results in an increase of performance by reducing the computation times of the SLS optimization.

ACKNOWLEDGMENTS

Partially supported by the Marie Curie network Control Training Site, HPMT-CT-2001-00278, and programme POCTI, FCT, Ministério da Ciência e Tecnologia, Portugal, SFRH / BD / 16370 / 2004, and ESF through the III Quadro Comunitário de Apoio.

REFERENCES

N. Bergboer, V. Verdult, and M. Verhaegen. An efficient implementation of maximum likelihood identification of LTI state-space models by local gradient search. In *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 616–621, Las Vegas, Nevada USA, December 2002.

J. Borges, V. Verdult, M. Verhaegen, and M. Ayala Botto. Separable least squares for projected gradient identification of composite local linear state-space models. In *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems*, Leuven, Belgium, July 2004.

J. Bruls, C. Chou, B. Haverkamp, and M. Verhaegen. Linear and non-linear system identification using separable least-squares. *European Journal of Control*, 5(1):116–128, 1999.

G. H. Golub and V. Pereyra. The differentiation of pseudo-inverse and nonlinear least squares problems whose variables separate. *SIAM J. of Numerical Analysis*, 10(2):413–432, April 1973.

G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins, Baltimore, 3rd edition, 1996.

L. H. Lee and K. Poolla. Identification of linear parameter-varying systems using nonlinear programming. *Journal of Dynamic Systems, Measurement and Control*, 121(1):71–78, March 1999.

T. McKelvey and A. Helmersson. System identification using an over-parametrized model class: Improving the optimization algorithm. In *Proceedings of the 36th IEEE Conference on Decision and Control*, pages 2984–2989, San Diego, California, December 1997.

R. Murray-Smith and T. A. Johansen. *Multiple Model Approaches to Modelling and Control*. Taylor and Francis, London, 1997.

K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1):4–27, 1990.

T. Ribarits, M. Deistler, and B. Hanzon. On new parametrization methods for the estimation of linear state-space models. *forthcoming in the International Journal of Adaptive Control and Signal Processing*, 2004.

V. Verdult. *Nonlinear System Identification: A State-Space Approach*. PhD thesis, University of Twente, Faculty of Applied Physics, Enschede, The Netherlands, 2002.

V. Verdult, L. Ljung, and M. Verhaegen. Identification of composite local linear state-space models using a projected gradient search. *International Journal of Control*, 75(16/17):1385–1398, 2002.

M. Verhaegen. Identification of the deterministic part of MIMO state space models given in innovations form from input-output data. *Automatica*, 30(1):61–74, 1994.