Model parameter gradients in prediction
identification of state-space systems

P. Carrette and T. McKelvey

Department of Electrical Engineering
Linkping University, S-581 83 Linkping, Sweden
WWW: http://www.control.isy.liu.se
Email: carrette@isy.liu.se

March 5, 1998

Report no.: LiTH-ISY-R-2011
Submitted to the 37-th Conference on Decision and Control

Technical reports from the Automatic Control group in Linkping are available by anonymous ftp at the address ftp.control.isy.liu.se. This report is contained in the compressed postscript file 2011.ps.Z.
Model parameter gradients in prediction identification of state-space systems

P. Carrette* and T. McKelvey

Department of Electrical Engineering, Linköping University
S-58183 Linköping, Sweden

Abstract

The paper is devoted to the study of the gradient computation related to procedures for identifying state-space systems in the prediction error sense. The knowledge of these gradients is needed when iteratively estimating a state-space model for the system on the basis of data measurements. In classical estimation algorithm, any gradient signal is evaluated by running these data through a state-space dynamics corresponding to the model differentiation with respect to the related parameter. In order to reduce the computation burden of this estimation, the paper put into light the structure of the state-space gradient signals and, as a by product, propose a new method for computing them. The obtained improvement is based on exploiting the properties of matrices that commute with the prediction model state-feedback matrix.

Keywords: prediction identification, state-space system, controllability, matrix commutation

1 Introduction

In this paper, we consider MIMO systems whose input-to-output (I/O) measurements are dynamically described by

\begin{align*}
x(t + 1) &= Ax(t) + Bu(t) + Ke(t) \\
y(t) &= Cx(t) + Du(t) + e(t)
\end{align*}

where \(u(t) \in \mathbb{R}^m\) and \(y(t) \in \mathbb{R}^p\) respectively denote the input and output of the system while \(e(t) \in \mathbb{R}^p\) stands for perturbation samples, e.g. out of a white noise process with bounded moments. The dimension of the auxiliary state vector \(x(t)\), i.e. \(n\), is the order of the state-space realization of the system. Furthermore, \(A \in \mathbb{R}^{n \times n}\), \(B \in \mathbb{R}^{n \times m}\), \(C \in \mathbb{R}^{p \times n}\),

*Corresponding author: carrette@isy.liu.se
\( D \in \mathbb{R}^{p \times m} \) and \( K \in \mathbb{R}^{n \times p} \) are the state-space matrices in this realization.

A natural predictor for the system in state-space form (1) is given by [7]

\[
\begin{align*}
\dot{x}(t + 1) &= (A - KC)x(t) + Bu(t) + K(y(t) - Du(t)) \\
\dot{y}(t) &= Cx(t) + Du(t)
\end{align*}
\]

(2)

where \( \dot{x}(t) \) is an estimate of the state vector \( x(t) \) and \( \dot{y}(t) \) denotes the predictor for the output \( y(t) \) of the system.

When applying prediction error identification to such systems, a part of the elements of the state-space matrices are taken as (non-trivial) parameters that are tuned in order to make the predicted output \( \dot{y}(t) \) as close as possible to the system output \( y(t) \).

More precisely, the state-space model that is identified as the best description of the system dynamics in the prediction error sense on the only basis of data measurements, i.e.

\[
Z^N = \{(y(t), u(t)), 0 < t \leq N\},
\]

is the one whose parameters correspond to

\[
\hat{\theta}_N = \arg\min_{\theta \in \mathcal{D} \subseteq \mathbb{R}^d} V(\theta, Z^N)
\]

(3)

with

\[
V(\theta, Z^N) = \frac{1}{2N} \sum_{t=1}^{N} (y(t) - \dot{y}(t, \theta))^2
\]

where the vector \( \theta \) contains, put in a predefined order, the \( d \) elements that are considered as parameters in the state-space matrices. The subset \( \mathcal{D} \) stands for the set of parameter vectors for which the predictor dynamics (2) is stable, i.e. all the eigenvalues of the state-feedback matrix \((A - KC)\) are inside the unit disk.

Note that we have put into light the dependence of the predictor output upon the parameter values, i.e. \( \dot{y}(t) = \dot{y}(t, \theta) \).

**Remark 1** The choice of the elements in the state-space matrices that are considered as parameters is called the parameterization of the state-space model. Different parameterizations exist. For example, the full parameterization [7] considers all the elements as parameters, the canonical parameterization [5] has the minimal number of parameters, the tridiagonal parameterization [8] imposes the matrix \( A \) to have only three principal diagonals and the balanced parameterization [9] where an intricate parameterization of the matrices \( A, B \) and \( C \) having a total number of parameters identical to \( n(m + p) \) is used.

In Table 1, we present the number of parameters in each state-space matrix as well as their total number \( d \) for all these parameterizations.

Because of the dependence of the predictor output \( \dot{y}(t, \theta) \) in the model parameters, i.e. within \( \theta \), the minimization in (3) is really hard to solve: it refers to a multidimensional
nonlinear constrained problem. Such a nonlinear problem is classically worked out by use of iterative algorithms (see [6, 2] as reference books). More precisely, at the \( k \)-th iteration step, we generally intend to

1. provide a minimizing direction \( \zeta_k = \zeta(\hat{\theta}^{k-1}_N) \) where \( \hat{\theta}^{k-1}_N \) denotes the solution vector at the \((k-1)\)-th step.

2. solve the related scalar nonlinear problem, i.e. \( \alpha_k = \arg\min_{\alpha} V([\hat{\theta}^{k-1}_N + \alpha \zeta_k], Z^N) \) for \( \alpha \) lying in \([0, 1]\).

3. compute the \( k \)-th step solution as \( \hat{\theta}^k_N = \hat{\theta}^{k-1}_N + \alpha_k \zeta_k \).

And the iteration steps proceed until a desired stopping criterion is satisfied, i.e. either on the accuracy (e.g. \( \|\zeta_k\|_2 \) or \( V(\hat{\theta}^k_N, Z^N) \) small) or on the number of steps. In prediction error identification, the mostly used algorithm performs quasi-Newton steps for which the minimizing direction \( \zeta(\theta) \) is given by solving

\[
\zeta(\theta) = \arg\min_{\zeta \in \mathbb{R}^d} \| [\Psi(\theta)]^T \zeta - E(\theta) \|_2
\]

with

\[
E(\theta) = [\varepsilon^T(1, \theta), \ldots, \varepsilon^T(N, \theta)]^T \quad \text{and} \quad \Psi(\theta) = [\psi(1, \theta), \ldots, \psi(N, \theta)]
\]

where \( \varepsilon(t, \theta) \) denotes the prediction error, i.e. \( \varepsilon(t, \theta) = y(t) - \hat{y}(t, \theta) \in \mathbb{R}^p \), while \( \psi(t, \theta) \) stands for the gradient of the predicted output \( \hat{y}(t, \theta) \) with respect to the parameter vector \( \theta \), i.e. \( \psi(t, \theta) = \partial \hat{y}(t, \theta)/\partial \theta \) belonging to \( \mathbb{R}^{d \times p} \).

A well-conditioned solution for this least-squares (LS) equation is expressed in term of the pseudo-inverse [11] of the gradient matrix, i.e. \( \zeta(\theta) = [\Psi^T(\theta)]^+ E(\theta) \) where the symbol “\(^+\)” denotes the pseudo-inverse operator.

Hence, it is seen that the output prediction gradients \( \psi(t, \theta) \) are the basic ingredient of the model parameter estimation. Note that these gradients, evaluated at the (converged) solution, can also be used to provide statistical properties of the model estimate [5, chap. 9], e.g. covariance of the parameters and of the related model transfer functions.

From a numerical point of view, this means that it is valuable to end up with a method that can provide the successive prediction gradients at low cost. This has already been investigated by Gupta et al. in [3] without real numerical success.
With that in mind, the purpose of the present paper is twofold. First, we explore the structure of the predictor gradient when considering the full state-space parameterization. Then, we show how to practically take advantage of this structure for the low cost computation of the related prediction gradients.

The structure of the paper is as follows. In Section 2, we present the dynamics equation of state-space model gradients and, from it, extract the structure of the corresponding gradient signals. This makes use of matrices whose fundamental property is to commute with the predictor state-feedback matrix, i.e. \((A - KC)\). In Section 3, we detail this commutation property and show how to compute the related matrices without any structural assumptions on the state-feedback matrix. Numerical discussions are held in Section 4 where we compare our gradient computation and the one obtained through the simulation of the state-space dynamics obtained by differentiating of the predictor equations (2) for each parameter in \(\theta\) separately.

2 Gradients of state-space predictor models

As seen in the introduction, the \(l\)-th row of the state-space prediction gradient \(\psi(t, \theta) \in \mathbb{R}^{d \times p}\), evaluated at any value of the state-space “parameters”, i.e. at \(\theta\), is defined as

\[
\psi^T_l(t, \theta) = \partial \hat{y}(t, \theta) / \partial \theta_l
\]

In the full parameterization, \(\theta_l\) stands for any element of the state-space matrices \(A, B, C, D\) and \(K\), i.e. either \(a_{ij}, b_{ij}, c_{ij}, d_{ij}\) or \(k_{ij}\), respectively.

With the help of the state-space predictor formulation (2), the \(l\)-th gradient signal, i.e. when considering \(\psi^T_l(t, \theta)\) along \(t\), can be expressed by the following system of equations

\[
\frac{\partial \hat{x}(t+1)}{\partial \theta_l} = (A - KC) \frac{\partial \hat{x}(t)}{\partial \theta_l} + (A'_l - [K'_l C + K C'_l])\hat{x}(t) + (B'_l - [K'_l D + K D'_l])u(t) + K'_l \hat{y}(t)
\]

\[
\psi^T_l(t, \theta) = C \frac{\partial \hat{x}(t)}{\partial \theta_l} + C'_l \hat{x}(t) + D'_l u(t)
\] (4)

where \(A'_l, B'_l, C'_l, D'_l\) and \(K'_l\) respectively denote the partial derivative of \(A, B, C, D\) and \(K\) with respect to the element \(\theta_l\) belonging to one of these matrices only. There is \(d\) such equation set: one for each \(\theta_l\).

From equation (5), it is seen that the \(p\)-dimensional signal \(\psi^T_l(t, \theta)\) is a linear combination of the derivative of the estimated state vector with respect to \(\theta_l\), of the estimated state vector and of the input signals, i.e. \(\partial \hat{x}(t)/\partial \theta_l, \hat{x}(t)\) and \(u(t)\), respectively. Hence, apart the second (evaluated by simulating (2)) and the third signals, the only quantity that is needed for obtaining \(\psi^T_l(t, \theta)\) is the signal that is derived from equation (4), i.e. \(\partial \hat{x}(t)/\partial \theta_l\). Moreover, it seems that this equation must be simulated for each value of \(l\) with \(0 < l \leq d\). This means \(d = n(m+n+2p)+mp\) times in the full parameterization. In fact, this is these first equation simulations that make the computational cost of the gradient evaluation by
use of this technique so high.

In order to avoid this large amount of simulations, let us extract the links that possibly exist between the different signals $\partial \hat{x}(t)/\partial \theta$. This constitutes the main contribution of the paper. The different results are proven in Appendix.

To begin with, let us simplify equation (4) as

$$z(t + 1, \beta, s) = A z(t, \beta, s) + \beta s(t) \quad (6)$$

where $z(t, \beta, s), \beta \in \mathbb{R}^n$ and $s(t)$ is a scalar (as signal samples). Note immediately that, according to this equation, $(n+m+p)$ different “input” signals $s(t)$ are present in equation (4), i.e. $\hat{x}_j(t), u_j(t)$ and $y_j(t)$ for appropriate $j$.

As far as the equation (6) is concerned, a simple result is the following.

**Lemma 1** Let the matrix $A \in \mathbb{R}^{n \times n}$, the vector $\beta \in \mathbb{R}^n$ and the solution $z(t, \beta, s)$ of the equation (6) for a particular signal $s(t)$ be given. If there exists a matrix $R \in \mathbb{R}^{n \times n}$ commuting with $A$, i.e. $RA = AR$, such that $b = R \beta$, then $z(t, b, s) = R z(t, \beta, s)$.

From a practical point of view, this result implies that the signal $z(t, \beta, s)$ originating from the simulation of (6) can be used for accessing the solution of

$$z(t + 1, b, s) = A z(t, b, s) + b s(t)$$

provided $b = R \beta$ where $R$ commutes with $A$.

In fact, it would be more useful to characterize when, given the vector $\beta$, a particular vector $b$ can be reached by a commuting matrix $R$. Indeed, imagine the signal $z(t, \beta, s)$ is at hand, can it lead to the solution $z(t, b, s)$ according to Lemma 1? A partial answer for it is provided by the following statement in which we make use of the controllability matrix of the pair $(A, \beta)$, i.e. $C(A, \beta) = [\beta, A\beta, \ldots, A^{n-1}\beta]$.

**Theorem 2** Let the matrix $A \in \mathbb{R}^{n \times n}$ and the vectors $b, \beta \in \mathbb{R}^n$ be given. Then, there exists a matrix $R \in \mathbb{R}^{n \times n}$ commuting with $A$ such that $b = R \beta$ if and only if the condition

$$\ker C(A, \beta) \subseteq \ker C(A, b) \quad (7)$$

is satisfied.

The main drawback of this theorem lies in condition (7). Indeed, given $A$ and $\beta$, it restricts the set of vectors $b$ for which a matrix $R$ (commuting with $A$ such that $b = R \beta$) can be found. In fact, this restriction can be overcome by the choice of a vector $\beta$ for which the above condition is satisfied regardless of $b \in \mathbb{R}^n$. More formally, we state the following definition.

**Definition 1** A pair $(A, \beta)$, with $A \in \mathbb{R}^{n \times n}$ and $\beta \in \mathbb{R}^n$, is maximally controllable if condition (7) in Theorem 2 is satisfied for any $b \in \mathbb{R}^n$.

From this definition, the following result holds.
Let $A \in \mathbb{R}^{n \times n}$ and $\beta \in \mathbb{R}^n$. Then, for any vector $b \in \mathbb{R}^n$, there exists a matrix $R \in \mathbb{R}^{n \times n}$ commuting with $A$ so that $b = R \beta$ if and only if the pair $(A, \beta)$ is maximally controllable.

One question remains: given the matrix $A$, how to find a vector $\beta$ for which the pair $(A, \beta)$ is maximally controllable? The answer is achieved by applying controllability results obtained by Paige in [10]: namely,

1. Denote by $T$ the similarity transformation that puts the matrix $A$ in Hessenberg form, i.e. $A = T^{-1}AT$, for which the absolute value of the first lower sub-diagonal elements is decreasing, i.e. $|A_{2,1}| \geq \cdots \geq |A_{n,n-1}|$.

2. Let $|A_{l+1,l}|$ be the first zero (or numerically not relevant) element in that sub-diagonal. If no such element is found then $l = n$.

3. Then the pair $(A, T_1)$ is maximally controllable with a matrix $C(A, T_1)$ of rank $l$.

Hence, the choice of the first column of the similarity transformation $T$ for the vector $\beta$ does the job. Note that, in case the matrix $A$ is non-derogatory (i.e. all eigenvalues having a geometrical multiplicity identical to unity [4]), it is always possible to make $l = n$.

Now, let us come back to the problem of accessing the different state gradient signals $\partial \dot{x}(t)/\partial \theta_l$ satisfying equation (4) by the application of the above results. Therefore, assume that the pair $(A - KC, \beta)$ is maximally controllable and that, for $0 < i \leq n$, $R_i$ denotes the matrix commuting with the state-feedback matrix $(A - KC)$ and such that $R_i \beta = 1_i$ with $(1_i)_j = \delta_{ij}$. Furthermore, for each of the “inputs” appearing in equation (4), i.e. $\dot{x}_j(t)$, $u_j(t)$ and $y_j(t)$, suppose that the equation (6) has been simulated (with $A = (A-KC)$) for obtaining the solution signals $z(t, \beta, \dot{x}_j)$, $z(t, \beta, u_j)$ and $z(t, \beta, y_j)$, respectively.

Then, the fundamental result in Lemma 1 implies that the $d$ partial derivatives of the estimated state vector $\dot{x}(t)$, i.e. $\partial \dot{x}(t)/\partial \theta_l$ with $0 < l \leq d$, can be expressed only in terms of the $(n + m + p)$ simulated signals $z(t, \beta, \dot{x}_j)$, $z(t, \beta, u_j)$ and $z(t, \beta, y_j)$. Indeed, with the help of the matrix $R_i$, we easily obtain that

$$\partial \dot{x}(t)/\partial a_{ij} = R_i z(t, \beta, \dot{x}_j) \quad \text{and} \quad \partial \dot{x}(t)/\partial b_{ij} = R_i z(t, \beta, u_j)$$

For what concerns the partial derivatives with respect to the elements in the matrices $C$, $D$ and $K$, the superposition principle is applied as

$$\partial \dot{x}(t)/\partial c_{ij} = - \sum_{l=1}^n k_{li} R_l z(t, \beta, \dot{x}_j) \quad \text{and} \quad \partial \dot{x}(t)/\partial d_{ij} = - \sum_{l=1}^n k_{li} R_l z(t, \beta, u_j)$$

while

$$\partial \dot{x}(t)/\partial k_{ij} = - \left[ \sum_{l=1}^n c_{jl} R_l z(t, \beta, \dot{x}_i) + \sum_{l=1}^m d_{jl} R_l z(t, \beta, u_l) \right] + R_i z(t, \beta, y_j)$$
As a whole, the matrices \( R_i \) allow us to reduce the \( d \) simulations of the equation (4) to only \((n + m + p)\) simulations of the equation (6) with appropriate \( s(t) \) signals, i.e. \( \dot{x}_j(t) \), \( u_j(t) \) and \( y_j(t) \). Note that \((n + m + p)\) is exactly the number of different “inputs” acting on the dynamics of the prediction gradients \( \psi(t, \theta) \).

In case of full parameterization, the reduction factor is identical to \( n + p(n+m)/(n+m+p) \) while it becomes \( n + (mp - n^2)/(n+m+p) \) for the canonical parameterization.

**Remark 2** It is not difficult to take parameterized initial conditions for the estimated state vector \( \hat{x}(t) \) into account, i.e. \( x_0 \) for \( \hat{x}(0) \). Indeed, it only needs the simulation of equation (6) with a “Dirac” signal, i.e. \( z(t+1, \beta, \delta) = \mathcal{A} z(t, \beta, \delta) + \beta \delta(t) \). Then, the partial derivative of the estimated state with respect to any of these “initial” parameters is simply
\[
\frac{\partial \hat{x}(t)}{\partial x_{0i}} = R_i z(t, \beta, \delta)
\]

Of course, the simulation of the estimated state vector \( \hat{x}(t) \) should take care of the current value of these “initial” parameters, i.e. \( x_{0i} \) for \( 0 < i \leq n \).

### 3 State-feedback matrix commutation

In this section, we discuss about the evaluation of the matrices \( R_i \) that have appeared to play a central role in the result obtained before. Therefore, we examine the (generic) case of the matrix \( R \) presented in Lemma 1.

In this lemma, we have seen that, given a vector \( \beta \), the matrix \( R \) has to satisfy two conditions:
\[
R \mathcal{A} = \mathcal{A} R \quad \text{and} \quad R \beta = b
\]
where the matrix \( \mathcal{A} \) is identical to the predictor state-feedback matrix, i.e. \( \mathcal{A} = (A - KC) \), as soon as the matrices \( R_i \) are concerned (with \( R_i \beta = 1_i \)).

Before proposing a general algorithm for finding the matrix solution \( R \), let us present the special situation where the matrix \( \mathcal{A} \) is diagonalizable with distinct eigenvalues.

In that case, this matrix can be written as \( V \Lambda V^{-1} \) where \( \Lambda \) is a diagonal matrix containing the eigenvalues and \( V \) has the corresponding eigenvectors as columns. Hence, the matrix \( R \) is immediately expressed as
\[
R = V \text{diag}(w_1, \ldots, w_n) V^{-1}
\]
where \( w_i = (V^{-1}b)_i/(V^{-1}\beta)_i \).

**Remark 3** If the vector \( \beta \) is left free (but such that the pair \( (\mathcal{A}, \beta) \) is maximally controllable), then it is interesting to choose \( \beta = V (\sum_i 1_i) \). Indeed, it leads to a simplified expression for the \( w_i \)'s, i.e. \( w_i = (V^{-1}b)_i \).

Furthermore, in case the matrix \( \mathcal{A} \) is real-valued, so is such vector \( \beta \). This implies that the matrix \( R \) as well as any quantity \( z(t, \beta, s) \) in equation (6) are kept in real space. \( \square \)
In the general situations where the matrix $\mathcal{A}$ either has identical eigenvalues or is non-diagonalizable, the formula (9) can not be used anymore. In fact, a more sophisticated procedure is needed for evaluating the matrix $R$ out of the two equations in (8).

With the help of the matrix vectorization (or “vec”) operator [4, chap. 4], the two conditions defining the matrix $R$ can be put in the following form

$$
\begin{bmatrix}
\mathcal{A}^T \otimes I_n - I_n \otimes \mathcal{A} \\
\beta^T \otimes I_n
\end{bmatrix}
\begin{bmatrix}
\text{vec}(R)
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
b
\end{bmatrix}
$$

(10)

where $\otimes$ denotes the Kronecker product and $I_n$ stands for the identity matrix in $\mathbb{R}^n$. This constitutes a set of $n(n+1)$ equations with $n^2$ unknowns, i.e. $\text{vec}(R) \in \mathbb{R}^{n^2}$ being in one-to-one correspondence with $R \in \mathbb{R}^{n \times n}$.

Note that this set of equations can be interpreted as follows (see Carrette [1]). The solution $\text{vec}(R)$ lies in the null space of the matrix $[\mathcal{A}^T \otimes I_n - I_n \otimes \mathcal{A}]$ and satisfies the last $n$ equations, i.e $(\beta^T \otimes I_n)\text{vec}(R) = b$.

Now, let us show how to efficiently evaluate the solution $\text{vec}(R)$ by exploiting the structure of the Schur decomposition of the matrix $\mathcal{A}$, i.e. $\mathcal{A} = UT_uU^*$ where the symbol $*$ stands for the conjugate-transpose operator and the matrix $U$ is unitary while the matrix $T_u$ is upper-triangular.

Then, by putting the matrix $R$ in the form of $R = UXV^*$ with $X \in \mathbb{R}^{n \times n}$, we arrive at

$$
\begin{bmatrix}
T_u^T \otimes I_n - I_n \otimes T_u \\
(U^*\beta)^T \otimes I_n
\end{bmatrix}
\begin{bmatrix}
\text{vec}(X)
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
U^*b
\end{bmatrix}
$$

(11)

With the above interpretation in mind, we can easily write the solution of this set of equation, i.e. $\text{vec}(X)$, as

$$
\text{vec}(X) = Q q \quad \text{with} \quad q = \left((U^*\beta)^T \otimes I_n\right) Q^{-1} U^*b
$$

(12)

where the columns of $Q$ constitute a basis for the null space of $[T_u^T \otimes I_n - I_n \otimes T_u]$. Note that, for a matrix $\mathcal{A}$ having eigenvalues with geometrical multiplicities identical to unity, the dimension of this null space is exactly $n$ [1] so that $Q \in \mathbb{R}^{n^2 \times n}$. In other situations, the equation for $q$ must be understood in the generalized sense (using pseudo-inverse instead of matrix inverse [11]).

It is important to mention that the solution expression (12) has reduced the resolution of the $n(n+1)$ equation in (11) (or in the original expression for it) to finding a basis for the null space of a sparse and well-structured matrix, i.e. $[T_u^T \otimes I_n - I_n \otimes T_u]$, and to solving a set of $n$ equations, i.e. defining $q$.

Finally, let us refer to [1] for the complete description of an algorithm that leads to an orthogonal version of the matrix $Q$. Simply note that, by exploiting the structure of the matrix $[T_u^T \otimes I_n - I_n \otimes T_u]$, the proposed procedure achieves a low numerical cost, i.e. $\sim 80 \, n^4$. 

8
4 Numerical discussions

In this section, we intend to compare the amount of computations needed for evaluating the state-space gradient $p$-dimensional signals, i.e. $\psi_l(t, \theta)$ for $0 < l \leq d$ with $0 < t \leq N$, by the simulation of equation (4) for each state-space parameter and by use of the method presented in this paper.

Therefore, let us consider the evaluation of the gradient signals for a 5-inputs, 5-outputs and 10-states system, i.e. $p = m = 5$ and $n = 10$. The number of data measurements is chosen identical to $N = 1000$.

In Table 2, we have presented the numerical complexity (as given by the Matlab function “flops”) of the two method for computing the gradient signals in different parameterization situations; namely, in the full (F) and the canonical (C) parameterizations with four sets of state-space matrices, i.e. $A, B, C$ and/or $D$ and/or $K$. Furthermore, the labels “Grad. Dyn.” and “Grad. Struct.” refer to simulations of the gradient dynamics (for each state-space parameter) and gradient structure usages (i.e. the method presented in this paper with the assumption that the matrices $R_i$ are known beforehand), respectively, while “$z(t, \beta, s)$” stands for the simulation cost of the equation (6) over the different “input” signals $s(t)$, i.e. $\dot{x}_j(t)$, $u_j(t)$ and $y_j(t)$ for appropriate $j$.

A general comment is that the method presented here obviously outperforms the gradient dynamics simulation technique for computing the state-space model gradient signals in the two parameterization situations. In more details, we can remark the following.

- In gradient dynamics simulations, there is a linear relation between any additional parameter and the computational burden. This is not so clear in the present method because of the evaluation of the signals $z(t, \beta, s)$.

- These $z(t, \beta, s)$’s are only simulated for the “input” signals appearing in expression (4). That is to say that in the absence of parameters in the matrix $K$, as the output signal $y(t)$ disappear from this expression, none of the simulations of equation (6) corresponds to the $y_j(t)$’s. The effect of this is clearly seen in the last column of the table.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F</td>
<td>C</td>
</tr>
<tr>
<td>$A, B$ and $C$</td>
<td>18.6087</td>
<td>9.3024</td>
</tr>
<tr>
<td>$A, B, C$ and $D$</td>
<td>20.9359</td>
<td>11.6286</td>
</tr>
<tr>
<td>$A, B, C$ and $K$</td>
<td>23.2634</td>
<td>13.9550</td>
</tr>
<tr>
<td>$A, B, C, D$ and $K$</td>
<td>25.5911</td>
<td>16.2817</td>
</tr>
</tbody>
</table>

Table 2: Number of $10^7$-“flops” taken for computing the gradient signals in different state-space parameterization situations as well as the signals $z(t, \beta, s)$ simulated by equation (6) for the signals $s(t)$ present in expression (4).
The largest part of the computational cost in the present method is devoted to perform the linear combinations of the simulated signals \( z(t, \beta, s) \) for obtaining the state-space gradients. As these are numerically not too expensive, the obtained out-performance follows.

In Table 3, we have presented the “cpu”-time needed to accomplish the “flops” mentioned in Table 2 on a Sun Ultra 1/170E with 128 MB RAM by use of the Matlab platform for the matrix evaluations. When seconds are counted, the improvement performed by the present method takes a more real-life relevance.

Finally, let us mention that, in the present case of \( n = 10 \), it has taken less than one second for evaluating the matrices \( R_i \) commuting with the predictor state-feedback matrix \( (A - KC) \) and such that \( R_i \beta = 1_i \).

## 5 Conclusion

In this paper, we have studied a technical step of the procedure that deal with the identification of state-space systems in the prediction error sense. More precisely, we have considered the gradient computation of the related identification cost function. The knowledge of these gradients is needed when iteratively estimating a state-space model for the system on the basis of data measurements.

In classical estimation algorithm (see the Matlab-“Identification Toolbox” by Ljung), the gradient signal are evaluated through the simulations of state-space dynamics corresponding to the model differentiation with respect to the state-space parameters. In order to reduce the computation burden of this estimation, the paper is devoted to the study of the structure of the state-space gradient signals and, as a by product, to the elaboration of a new method for computing them. The obtained improvement is based on exploiting the properties of matrices that commute with the state-feedback matrix of the model predictor equation.

Numerical discussions are provided in order to illustrate the performances obtained by the proposed method.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F</td>
<td>C</td>
</tr>
<tr>
<td>( A, B ) and ( C )</td>
<td>13.3974</td>
<td>5.9893</td>
</tr>
<tr>
<td>( A, B, C ) and ( D )</td>
<td>15.6018</td>
<td>7.7679</td>
</tr>
<tr>
<td>( A, B, C ) and ( K )</td>
<td>17.9006</td>
<td>9.7570</td>
</tr>
<tr>
<td>( A, B, C, D ) and ( K )</td>
<td>20.1612</td>
<td>12.1638</td>
</tr>
</tbody>
</table>

Table 3: Time (in sec.) spent for computing the gradient signals in different state-space parameterization situations.
Appendix

Proof of Lemma 1:
We successively have that
\[
    z(t + 1, b, s) = R \begin{bmatrix} \mathcal{A} z(t, \beta, s) + \beta s(t) \end{bmatrix}
    = \mathcal{A} \begin{bmatrix} R z(t, \beta, s) + b s(t) \end{bmatrix} = \mathcal{A} z(t, b, s) + b s(t)
\]
so that \(z(t, b, s)\) obviously is a solution of equation (6).

Proof of Theorem 2:
(\(\Rightarrow\)) By assuming \(b = R \beta\) with \(R \mathcal{A} = \mathcal{A} R\), we have that \(C(\mathcal{A}, b) = R C(\mathcal{A}, \beta)\). Or, differently, \(|C(\mathcal{A}, b)|^T = |C(\mathcal{A}, \beta)|^T R^T\). This means that any column of \(|C(\mathcal{A}, b)|^T\) should lie in the range of \(|C(\mathcal{A}, \beta)|^T\) which is equivalent (see, e.g., [11, chap. 3]) to condition (7).

(\(\Leftarrow\)) Conversely, this condition implies that there exists a matrix \(R \in \mathbb{R}^{n \times n}\) so that
\[
    [b, \mathcal{A} b, \ldots, \mathcal{A}^{n-1} b] = [R \beta, R \mathcal{A} \beta, \ldots, R \mathcal{A}^{n-1} \beta]
\]
by the definition of the controllability matrix. By taking care of the first column of this matrix equation, we immediately have that \(b = R \beta\) as well as
\[
    [I_n R \beta, R \mathcal{A} \beta, \ldots, \mathcal{A}^{n-1} R \beta] = [R I_n \beta, R \mathcal{A} \beta, \ldots, R \mathcal{A}^{n-1} \beta]
\]
So, by applying Caley-Hamilton’s theorem [11], we derive that \((R \mathcal{A} - \mathcal{A} R) |C(\mathcal{A}, \beta)| \mu = 0\) for any vector \(\mu \in \mathbb{R}^n\).

Finally, it remains to prove that the matrix \(R\) can be chosen to commute with \(\mathcal{A}\), i.e. \((R \mathcal{A} - \mathcal{A} R) \mu = 0\). This is achieved as follows. As the condition (7) only imposes that the null space of such \(R\) does not belong to the range of \(C(\mathcal{A}, \beta)\), it suffices to independently impose that \(R\) further satisfies \(\ker R = \ker |C(\mathcal{A}, \beta)|^T\). Indeed, any vector \(\mu\) is made of two contributions: \(\mu = \mu_1 + \mu_2\) where, for any \(i \geq 0, \mathcal{A}^i \mu_1\) belong to the range of \(C(\mathcal{A}, \beta)\) and \(\mathcal{A}^i \mu_2\) does not. Hence, we have that, for such matrix \(R\),
\[
    (R \mathcal{A} - \mathcal{A} R) \mu = (R \mathcal{A} - \mathcal{A} R) \mu_1 = 0
\]
This completes the proof.

References


