Maximum likelihood estimation of linear SISO models subject to missing output data and missing input data

Ragnar Wallin and Anders Hansson

Maximum Likelihood Estimation of Linear SISO Models Subject to Missing Output Data and Missing Input Data

Ragnar Wallin and Anders Hansson
Division of Automatic Control
Linkoping University
SE–581 83 Linkoping, Sweden

March 19, 2014

Keywords: System identification, Maximum likelihood estimation, Missing data

Abstract

In this paper we describe an approach to maximum likelihood estimation of linear SISO models when both input and output data are missing. The criterion minimized in the algorithms is the Euclidean norm of the prediction error vector scaled by a particular function of the covariance matrix of the observed output data. We also provide insight into when simpler and in general sub-optimal schemes are indeed optimal. The algorithm has been prototyped in Matlab, and we report numerical results that support the theory.

1 Introduction

Missing data is a commonly occurring phenomenon in industrial applications. As identification experiments are often both time consuming and expensive it is usually not an option to discard data sets containing missing measurements. Instead, people frequently use different ad hoc methods to fill in the missing data. This will most often result in biased parameter estimates, see [15] for examples of methods used and the effect they cause. The more data that is missing the more harmful influence the use of these ad hoc methods has. This explains the great interest in algorithms that can handle missing data.

Data can be missing at random time instants or according to periodical patterns. Examples of randomly missing data are outliers and random sensor failures. Periodically missing data appear for example in time sharing of sensors, radar scans and multirate sampling. When the input is sampled at a faster rate than the output a periodic missing output data problem results [9], [29].

Missing data does not only occur in industrial applications. Much work has been done in statistics and econometrics where time series are considered. A very good survey of the work done in statistics is presented in the book [18]. Other references also considering time series are [13], [22], [23],[24] and [25] where AR models are studied and [1], [6], [16] and [28] where ARMA models are studied. Some approaches, based on the EM method [4] or the Kullback-Liebler information measure, for different model types include [14], [34] and [36]. A very promising approximate EM method is presented in [32]. In [9] and [29] least squares methods are used for periodic ARMAX and ARX models, respectively. In [33] neural networks are used and in [26] a Bayesian learning approach is taken for ARMAX models. Frequency domain methods are used in [27]. Recently nuclear norm minimization based on ideas from subspace identification has been proposed, [5, 10, 21, 19]. This is not a complete reference list. For all references mentioned above either the methods do not consider exact maximum

likelihood estimation, and are hence not optimal, or the methods do only consider special cases of models, i.e. no unified treatment of all common models in system identification is provided.

The algorithms presented in this article can handle most commonly used models in system identification, such as Box-Jenkins, ARMAX, ARX, FIR, OE, ARMA, AR and MA models. Both data missing at random time instants and data missing according to periodic patterns can be taken care of. Moreover, both missing output data and missing input data is treated. However, the algorithms cannot handle cases when the missing data pattern is dependent on the signal. If for example all output data with a magnitude greater than two is missing the algorithms may not produce the correct answer. To be more precise, the distributions of the signals and the data missing mechanism have to be independent, see e.g. [18]. Unfortunately the statistics literature, [18], calls this condition on the data missing mechanism to be "missing at random". However it is possible to have the data be missing deterministically, e.g. according to a pre-described periodic pattern independent of the signals. Of course it is not sufficient that the times when the data is missing are described by a distribution, since this distribution is not allowed to be dependent on the distribution of the signal. In this work we assume that we know when the data is missing. In case this is not the case the problem becomes much more complicated.

This work relies heavily of the work presented in [11, 12], where maximum likelihood estimation of general Gaussian models with missing data is presented. There it is shown that the algorithm we are proposing to use is more efficient than the EM algorithm, which is a popular choice of algorithm when data is missing. The approach to solve the problem is based on linear algebra. This is the vehicle to a unified treatment, and it is the opinion of the authors that this approach simplifies the presentation of the theory and adds insight into the problem. Of course, the algorithms can be implemented using state-space models, fixed-interval Kalman smoothers et cetera. It is hinted throughout the article what matrix multiplication, data estimation and so on corresponds to in other frameworks. However, we will show that sparse linear algebra techniques provide a reasonably efficient implementation. It should be stressed that straight forward application of the results in [11, 12] will not provide an efficient implementation.

The novel contributions of this article are mainly threefold: 1) A unified and efficient treatment of system identification for missing data for many commonly used linear SISO models; 2) Handling of missing outputs and inputs in a seamless way; 3) Insight into when simple, commonly used, and in general sub-optimal schemes indeed are optimal. It is possible to extend the results any model structure that is linear in the inputs, the outputs and the noise, including MIMO systems.

The rest of the article is organized as follows. In Section 2 the models considered are presented. Section 3 first recapitulates how system identification is performed when all data is observed. Then the criterion to minimize to obtain the maximum likelihood estimate when output data is missing is discussed. Two ways to handle missing input data are presented in Section 4. How to actually solve the resulting optimization problems is shown in Section 5. In Section 6 computational efficiency is discussed. Some illustrative numerical examples can be found in Section 7. Finally, conclusions are drawn in Section 8.

2 Models

Assume that input data and output data from a linear SISO system is collected from time $k = 1$ to time $k = n$ and that all signals are zero for times $k \leq 0$. If this assumption does not hold in an application it has negligible influence on the estimated model in case n is large. However, if this is not the case other methods than the one we propose should be used. The output and input data are stacked into vectors $y = [y_1,\ y_2,\ \cdots,\ y_n]^T$ and $u = [u_1,\ u_2,\ \cdots,\ u_n]^T$ respectively. Then a general model, linear in data, describing the relationship between the input signal and the output signal can be written as

$$\mathcal{A}y - \mathcal{B}u = e \tag{1}$$

where $\mathcal{A}$ and $\mathcal{B}$ are $n \times n$ matrices that can have a complex nonlinear dependence on the model parameters. The zero mean random vector e has uncorrelated entries and covariance matrix λI_n where I_n is

the $n \times n$ identity matrix. In this article the matrices $\mathcal{A}$ and $\mathcal{B}$ are either restricted to be on the form

$$\mathcal{A} = \mathcal{A}_1 = A_1 C_1^{-1} D_1 \tag{2}$$

$$\mathcal{B} = \mathcal{B}_1 = B C_1^{-1} D_1 F^{-1} S_n^{n_k}. \tag{3}$$

or the model has $\mathcal{B} = 0$ and $\mathcal{A}$ is of the form

$$\mathcal{A} = \begin{bmatrix} \mathcal{A}_1 & \mathcal{B}_1 \\ 0_n & \mathcal{A}_2 \end{bmatrix} = \begin{bmatrix} A_1 C_1^{-1} D_1 & B C_1^{-1} D_1 F^{-1} S_n^{n_k} \\ 0_n & \beta A_2 C_2^{-1} D_2 \end{bmatrix}$$

where 0_n is the $n \times n$ zero matrix and β is a scalar. The matrix S_n is the $n \times n$ shift matrix with zeros except on the first sub-diagonal which consists of ones, and n_k is the time delay between the input signal and the output signal. The matrices A_i, B, C_i, D_i and F are defined as

$$A_i = \sum_{k=1}^{n_{ai}} a_{ik} S_n^k + I_n \qquad B = \sum_{k=1}^{n_b} b_k S_n^{k-1} \tag{4}$$

$$C_i = \sum_{k=1}^{n_{ci}} c_{ik} S_n^k + I_n \qquad D_i = \sum_{k=1}^{n_{di}} d_{ik} S_n^k + I_n \tag{5}$$

$$F = \sum_{k=1}^{n_f} f_k S_n^k + I_n \tag{6}$$

for $i = 1, 2$, where S_n^0 is defined to be I_n. This is general enough to describe the models commonly used in system identification such as Box-Jenkins, ARMAX, ARX, FIR, OE, ARMA, AR and MA models. They have $\mathcal{A}$ and $\mathcal{B}$ as in (2)-(3) and the choices of A_1, B, C_1, D_1 and F for these special cases are given in Table 1. More general model classes may be considered by considering other choices of $\mathcal{A}$ and $\mathcal{B}$. The matrices defined in (4)-(6) are banded lower triangular Toeplitz matrices. This class of matrices has several interesting properties. Multiplication of two lower triangular Toeplitz matrices is commutative and the resulting matrix is also lower triangular Toeplitz. The inverse of a lower triangular Toeplitz is lower triangular Toeplitz. Both C_i, $i = 1, 2$, and F can be written as the sum of the identity matrix and a strictly lower triangular matrix L. The matrix L is linear in the parameters and as it is strictly lower triangular it is nilpotent. It is easy to show that the inverse of such a matrix is $(I_n + L)^{-1} = I_n + \sum_{k=1}^{n-1} (-L)^k$, and hence both the inverse matrix and the entire model are polynomial in the parameters. Multiplication of two lower triangular Toeplitz matrices corresponds to the convolution of two linear filters. The filter coefficients are the entries of the first column of the matrices. Filtering a signal vector is done by multiplying it with a lower triangular Toeplitz matrix. Inverse filtering corresponds to multiplication with the inverse of a lower triangular Toeplitz matrix. The derivatives of $\mathcal{A}_i$, $i = 1, 2$, and $\mathcal{B}_1$ with respect to the parameters are given in the appendix. Note that multiplying a signal by these derivatives also corresponds to linear filtering. Also time-varying models can be written as (2)-(3). The matrices will still be lower triangular but as the parameters vary they will no longer be Toeplitz.

Table 1: Special cases of the model in (1). When a matrix is listed as free the parameters in (4)-(6) for that matrix are not fixed. Note how choosing a filter to be one or zero, when system identification is treated in a filtering framework, corresponds to choosing a matrix to be the identity matrix or the zero matrix.

Model	A_1	B	C_1	D_1	F
Box-Jenkins	I_n	free	free	free	free
ARMAX	free	free	free	I_n	I_n
ARX	free	free	I_n	I_n	I_n
FIR	I_n	free	I_n	I_n	I_n
OE	I_n	free	I_n	I_n	free
ARMA	free	0_n	free	I_n	I_n
AR	free	0_n	I_n	I_n	I_n
MA	I_n	0_n	free	I_n	I_n

3 Identification when only output data is missing

When all data is observed, the parameters of the models listed in Table 1 can be estimated with a prediction error method [20], [31]. The estimate is obtained by finding the parameters that minimize the Euclidean norm of the prediction error vector. This is equivalent to solving the nonlinear least squares

problem

$$\min_{\theta} \|\mathcal{A}y - \mathcal{B}u\|_2^2 \tag{7}$$

where θ are the parameters that appear in the definitions of the matrices in (4)-(6). Prediction error methods yield the maximum likelihood estimate of the parameters if the random vector e is Gaussian.

Let T_m be a matrix containing entries that are either one or zero. The entries in T_m are chosen such that they pick out the missing data in y. The transpose of T_m will then pick out the columns in $\mathcal{A}$ that are multiplied with the missing output data. Similarly, let T_o be a matrix that picks out the observed data in y. Such matrices have the property that $T_m^T T_m + T_o^T T_o = I_n$. Then the model (1) can be written as

$$\begin{aligned}\mathcal{A}y - \mathcal{B}u &= \underbrace{\mathcal{A}T_m^T}_{\mathcal{A}_m}\underbrace{T_m y}_{y_m} + \underbrace{\mathcal{A}T_o^T}_{\mathcal{A}_o}\underbrace{T_o y}_{y_o} - \mathcal{B}u \\ &= \mathcal{A}_m y_m + \mathcal{A}_o y_o - \mathcal{B}u.\end{aligned}$$

This form is convenient when missing output data is to be estimated.

It can be shown that the criterion to minimize for missing output data is

$$\min_{\theta, y_m} \left\| \frac{\mathcal{A}y - \mathcal{B}u}{\det(\mathcal{A}_o^T P_{\mathcal{A}_m}^{\perp} \mathcal{A}_o)^{\frac{1}{2n_o}}} \right\|_2^2 \tag{8}$$

where $P_{\mathcal{A}_m}^{\perp} = I_n - \mathcal{A}_m \mathcal{A}_m^{\dagger}$ is a projection matrix and and n_o is the number of observed output data points. See [11, 12] for a derivation of this result. The calculations are based on writing down the likelihood function of the observed data, but too long to present here again. Note that, for the models listed in Table 1, (8) reduces to (7) when all output data is observed. The residual of the separable nonlinear least squares problem in (8) is proportional to the determinant of the covariance matrix of the observed output data which is

$$\mathrm{E}(y_o y_o^T) = \lambda(\mathcal{A}_o^T P_{\mathcal{A}_m}^{\perp} \mathcal{A}_o)^{-1}.$$

Based on the results in [11, 12] it is possible to show that (8) can be written as

$$\min_{\theta, y_m} \left\| \left(\frac{\det(\mathcal{A}_m^T \mathcal{A}_m)}{\det(\mathcal{A}^T \mathcal{A})} \right)^{\frac{1}{2n_o}} (\mathcal{A}y - \mathcal{B}u) \right\|_2^2 \tag{9}$$

As $\det(\mathcal{A}^T\mathcal{A}) = 1$ for the models listed in Table 1 the criterion (9) can be further simplified to

$$\min_{\theta, y_m} \| \det(\mathcal{A}_m^T \mathcal{A}_m)^{\frac{1}{2n_o}} (\mathcal{A}y - \mathcal{B}u) \|_2^2. \tag{10}$$

Solving this separable nonlinear least squares problem yields the maximum likelihood estimate of the parameters if the random vector e is Gaussian, [11, 12].

For FIR and OE models $\mathcal{A}$ is the identity matrix. The matrix $\mathcal{A}_m$ consists of selected columns of $\mathcal{A}$. Hence, $\det(\mathcal{A}_m^T \mathcal{A}_m) = 1$ for those models. Consequently, (10) is then equivalent to

$$\min_{\theta, y_m} \|\mathcal{A}y - \mathcal{B}u\|_2^2. \tag{11}$$

which is the same criterion as in (7). Several authors have suggested to minimize this criterion also for other model structures than the ones for which it is the correct criterion, e.g. [30]. It is further analyzed in [24, 35, 32], and it has been found to be suboptimal for ARX models and optimal for FIR models. This method is also implemented in `misdata` in Matlab's System Identification toolbox, [20, Chapter 14.2], where it is correctly mentioned that this is an approximate method. In [32] an approximate EM method with similarities to the the method above is proposed to reduce the problems with sub-optimality. It is demonstrated to be computationally less expensive than the EM method for AR models, and it provide a better estimate than when minimizing (11). However, it does not provide the true maximum likelihood estimate as does the EM method and the method we propose.

4 Identification when also input data is missing

In some applications also input data is missing. Missing input data is not as common as missing output data but two possible ways to handle such cases are presented below to make the treatment of missing data complete. In the first approach the missing input data is considered to be deterministic and in the second approach it is considered to be stochastic. The method implemented in `misdata` can also handle missing inputs.

4.1 Deterministic approach

One way of handling missing input data is to treat it as parameters and minimize the criterion

$$\min_{\theta, y_m, u_m} \| \det(\mathcal{A}_m^T \mathcal{A}_m)^{\frac{1}{2n_o}} (\mathcal{A}y - \mathcal{B}u) \|_2^2 \tag{12}$$

where u_m is the missing input data. However, it should be noted that missing input data can only be considered being parameters in a true sense when the number of missing input data points do not tend to infinity when the total number of data points tend to infinity, [18]. If this is not the case the covariance matrix of the estimated input data does not tend to zero as the number of observed data points tends to infinity. Hence, the parameter estimate is not a maximum likelihood estimate even if the random vector e is Gaussian.

4.2 Stochastic approach with model for the input signal

Another way to handle missing input data is to take a similar approach as was done in [14] for ARX models and augment the model (1) with a model for the input signal. The result is a model on the form

$$\begin{bmatrix} \mathcal{A}_1 & -\mathcal{B}_1 \\ 0_n & \mathcal{A}_2 \end{bmatrix} \begin{bmatrix} y \\ u \end{bmatrix} = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \tag{13}$$

where e_1 has covariance matrix $\lambda_1 I_n$ and e_2 has covariance matrix $\lambda_2 I_n$. The matrices $\mathcal{A}_1$ and $\mathcal{B}_1$ can represent any of the models listed in Table 1 that include an input signal and $\mathcal{A}_2$ can represent an AR model, an MA model or an ARMA model. The model (13) is a multivariate time series with y and u as output signals. To make this model fit in the framework used earlier the rows in (13) that describe the input model are scaled by $\beta = \sqrt{\frac{\lambda_1}{\lambda_2}}$. The result is the model

$$\underbrace{\begin{bmatrix} \mathcal{A}_1 & -\mathcal{B}_1 \\ 0_n & \beta\mathcal{A}_2 \end{bmatrix}}_{\mathcal{A}} \underbrace{\begin{bmatrix} y \\ u \end{bmatrix}}_{z} = \underbrace{\begin{bmatrix} e_1 \\ \beta e_2 \end{bmatrix}}_{e}, \tag{14}$$

which is on the form (1) with $\mathcal{B} = 0$. The random vector e has zero mean uncorrelated entries and covariance matrix $\lambda_1 I_{2n}$. Thus, the criterion to minimize is on the form (9). For the model (14) $\det(\mathcal{A}^T\mathcal{A}) = \beta^{2n}$ and hence (9) can be written as

$$\min_{\theta, \beta, z_m} \left\| \left(\frac{\sqrt{\det(\mathcal{A}_m^T \mathcal{A}_m)}}{\beta^n} \right)^{\frac{1}{n_o}} \mathcal{A}z \right\|_2^2. \tag{15}$$

Solving this separable nonlinear least squares problem yields the maximum likelihood estimate of the parameters if the random vector e is Gaussian, [11, 12].

The advantage of the stochastic approach over the deterministic one is that, if the input is indeed well modeled by an AR, MA or ARMA model, the variance of the estimated parameters in $\mathcal{A}_1$ and $\mathcal{B}_1$ can be lower than when missing input data is treated as parameters. The advantage of the deterministic approach is that it can be used even when the input signal cannot be well modeled by a time series.

5 Solving the two separable nonlinear least squares problems

The two optimization problems to solve are (12) and (15) as (10) is only a special case of (12). First it is recapitulated how to solve a separable nonlinear least squares problem and then the formulas needed for solving the specific problems (12) and (15) are given.

5.1 Separable nonlinear least squares with approximate gradient

For a separable nonlinear least squares problem

$$\min_{\theta, x} \|Gx + h\|_2^2 \tag{16}$$

where G and h are nonlinear in θ it is possible to do as is suggested in [7] and first optimize over x and then substitute the optimal x, which is

$$x(\theta) = -G^\dagger h,$$

into (16) and solve the problem

$$\min_\theta \|Gx(\theta)+h\|_2^2 = \min_\theta \|(I-GG^\dagger)h\|_2^2 = \min_\theta \|P_G^\perp h\|_2^2.$$

The solution can be found by a standard nonlinear least squares solver supplied with a function that computes the residual

$$r = Gx(\theta) + h = P_G^{\perp} h \tag{17}$$

and the gradient of the residual defined by

$$\frac{\partial r}{\partial \theta_k} = \frac{\partial P_G^{\perp}}{\partial \theta_k} h + P_G^{\perp} \frac{\partial h}{\partial \theta_k}.$$

However, it has been observed in many studies that the solution is obtained faster if an approximate gradient

$$g = P_G^{\perp} \left(\frac{\partial G}{\partial \theta_k} x(\theta) + \frac{\partial h}{\partial \theta_k} \right), \tag{18}$$

due to Kaufman, [17], is supplied to the nonlinear least squares solver. As this agrees with the experiences from [11, 12], where parameters of ARMAX models subject to missing data were estimated, it is the approach used also in this article. Using this approximate gradient often reduces the computational time with 25% or more. Furthermore, the error between the real gradient and the approximate one is

$$\frac{\partial r}{\partial \theta_k} - g = \frac{\partial P_G^{\perp}}{\partial \theta_k} r$$

which is small close to the optimum if $\|r\|_2$ is small.

5.2 Solving the problem without input model

Similarly to what is done in Section 3, let T_m be the matrix that picks out the missing data in y, let T_o be the matrix that picks out the observed data in y, let R_m be the matrix that picks out the missing data in u and let R_o be the matrix that pics out the observed data in u. This results in

$$\mathcal{A}_m = \mathcal{A} T_m^T;\ \mathcal{A}_o = \mathcal{A} T_o^T;\ \mathcal{B}_m = \mathcal{B} R_m^T;\ \mathcal{B}_o = \mathcal{B} R_o^T$$

and

$$y_m = T_m y;\ y_o = T_o y;\ u_m = R_m u;\ u_o = R_o u.$$

Define γ_1 as

$$\gamma_1 = \det(\mathcal{A}_m^T \mathcal{A}_m)^{\frac{1}{2n_o}}. \tag{19}$$

The derivatives of γ_1 with respect to the parameters are

$$\frac{\partial \gamma_1}{\partial \theta_k} = \frac{\gamma_1}{n_o} \mathrm{trace}\left(\mathcal{A}_m^{\dagger} \frac{\partial \mathcal{A}_m}{\partial \theta_k} \right). \tag{20}$$

For the separable nonlinear least squares problem (12) the matrices and vectors in (17) are

$$G = \gamma_1 \begin{bmatrix} \mathcal{A}_m & -\mathcal{B}_m \end{bmatrix};\ h = \gamma_1 (\mathcal{A}_o y_o - \mathcal{B}_o u_o) \tag{21}$$

$$P_G^{\perp} = P^{\perp}_{\begin{bmatrix} \mathcal{A}_m & -\mathcal{B}_m \end{bmatrix}} = I_n - \begin{bmatrix} \mathcal{A}_m & -\mathcal{B}_m \end{bmatrix} \begin{bmatrix} \mathcal{A}_m & -\mathcal{B}_m \end{bmatrix}^{\dagger} \tag{22}$$

$$x(\theta) = \begin{bmatrix} y_m(\theta) \\ u_m(\theta) \end{bmatrix} = - \begin{bmatrix} \mathcal{A}_m & -\mathcal{B}_m \end{bmatrix}^{\dagger} (\mathcal{A}_o y_o - \mathcal{B}_o u_o) \tag{23}$$

and by defining

$$y(\theta) = T_m^T y_m(\theta) + T_o^T y_o;\ u(\theta) = R_m^T u_m(\theta) + R_o^T u_o$$

the residual can be computed as

$$r = Gx(\theta) + h = \gamma_1 (\mathcal{A} y(\theta) - \mathcal{B} u(\theta)), \tag{24}$$

and the approximate gradient of the residual is

$$g_{\theta_k} = P^{\perp}_{\begin{bmatrix} \mathcal{A}_m & -\mathcal{B}_m \end{bmatrix}} \left[\frac{\partial \gamma_1}{\partial \theta_k} (\mathcal{A} y - \mathcal{B} u) + \gamma_1 \left(\frac{\partial \mathcal{A}}{\partial \theta_k} y - \frac{\partial \mathcal{B}}{\partial \theta_k} u \right) \right]. \tag{25}$$

All necessary derivatives of $\mathcal{A}$ and $\mathcal{B}$ with respect to the parameters are given in (27)-(36). When θ has been estimated an estimate of the variance λ can be computed as

$$\lambda(\hat{\theta}) = \frac{\|\mathcal{A}(\hat{\theta}) y(\hat{\theta}) - \mathcal{B}(\hat{\theta}) u(\hat{\theta})\|_2^2}{\mathrm{trace}\left(P^{\perp}_{\begin{bmatrix} \mathcal{A}_m & -\mathcal{B}_m \end{bmatrix}} \right)}. \tag{26}$$

This algorithm has similarities with the one presented in [35]. Output and input data was estimated in the same way. However, the minimization with respect to θ was done using a more or less ad hoc bias compensation scheme in [35]. Moreover, the algorithm was only applicable to ARX models. We end this subsection by noting that $x(\theta)$ in (23) can be equivalently computed using a fixed interval smoother.

5.3 Solving the problem with input model

If γ_2 is defined as $\gamma_2 = \gamma_1/\beta^{\frac{n}{n_o}}$, the expressions in (19) and (20) can be reused. The derivatives of γ_2 with respect to the parameters are

$$\frac{\partial \gamma_2}{\partial \theta_k} = \frac{1}{\beta^{\frac{n}{n_o}}} \frac{\partial \gamma_1}{\partial \theta_k}; \ \frac{\partial \gamma_2}{\partial \beta} = -\frac{n\gamma_2}{n_o \beta}$$

For the separable nonlinear least squares problem (15) the matrices and vectors in (17) are

$$\begin{aligned} G = \gamma_2 \mathcal{A}_m; \ h = \gamma_2 \mathcal{A}_o z_o \\ P_G^\perp = P_{\mathcal{A}_m}^\perp = I_{2n} - \mathcal{A}_m \mathcal{A}_m^\dagger \\ x(\theta, \beta) = z_m(\theta, \beta) = -\mathcal{A}_m^\dagger \mathcal{A}_o z_o \end{aligned}$$

and by defining

$$z(\theta, \beta) = T_m^T z_m(\theta, \beta) + T_o^T z_o$$

the residual can be computed as

$$r = Gx(\theta, \beta) + h = \gamma_2 \mathcal{A} z(\theta, \beta),$$

and the approximate gradient of the residual is

$$\begin{aligned} g_{\theta_k} = P_{\mathcal{A}_m}^\perp \left(\frac{\partial \gamma_2}{\partial \theta_k} \mathcal{A} z(\theta, \beta) + \gamma_2 \frac{\partial \mathcal{A}}{\partial \theta_k} z(\theta, \beta) \right) \\ g_\beta = P_{\mathcal{A}_m}^\perp \left(\frac{\partial \gamma_2}{\partial \beta} \mathcal{A} z(\theta, \beta) + \gamma_2 \frac{\partial \mathcal{A}}{\partial \beta} z(\theta, \beta) \right). \end{aligned}$$

All necessary derivatives of the blocks in $\mathcal{A}$, with respect to the parameters, except

$$\frac{\partial \mathcal{A}}{\partial \beta} = \begin{bmatrix} 0_n & 0_n \\ 0_n & \mathcal{A}_2 \end{bmatrix}$$

are given in (27)-(36). When θ and β are estimated, estimates of the variances λ_1 and λ_2 can be computed as

$$\lambda_1(\hat{\theta}, \hat{\beta}) = \frac{\|\mathcal{A}(\hat{\theta}, \hat{\beta}) z(\hat{\theta}, \hat{\beta})\|_2^2}{n_o}; \ \lambda_2(\hat{\theta}, \hat{\beta}) = \frac{\lambda_1(\hat{\theta}, \hat{\beta})}{\hat{\beta}^2}.$$

We end this subsection by noting that $x(\theta, \beta)$ can be equivalently computed using a fixed interval smoother.

6 Computational efficiency

The computational complexity of the proposed algorithm is higher than for similar algorithms for the case of no missing data. For the general case described in [11, 12] the flop count per iteration for computing the residual and its gradients is linear in the number of parameters q, quadratic in the underlying number of data n, and cubical in the number of missing data n_m. For the case of system identification of dynamical models it is possible to speed up the implementation, and we show that the total flop count is linear in q and n and cubical in n_m, which is a significant improvement over the general case.

We will go through the computations step by step for the case without input model. The computations are not very much different for the case with input model, and the asymptotic expressions for the computational cost are the same.

First the matrices $\mathcal{A}_m = \mathcal{A} T_m^T$ and $\mathcal{B}_m = \mathcal{B} R_m^T$ in (5.2,5.2) have to be formed. These computations are equivalent to linear filtering, and can hence be implemented efficiently. The flop count is in the order of nq for each column in T_m^T and R_m^T. This flop count is easily obtained in practice by using sparse matrices or making a custom made implementation. In case the degrees of the polynomials in the models are significantly larger than $\log n$ it could pay of to use DFT computations for these filtrations, see e.g. [8]. We have not seen the need for this nor for any custom made implementation—sparse linear algebra techniques are sufficient. The overall flop count is hence in the order of $n_m n q$. We define $\mathcal{M}_m = \begin{bmatrix} \mathcal{A}_m & -\mathcal{B}_m \end{bmatrix}$.

The next steps to an efficient implementation is to compute QR-factorizations such that $\mathcal{A}_m E_1 = Q_1 R_1$ and $\mathcal{M}_m E_2 = Q_2 R_2$, where E_1 and E_2 are permutation matrices. Here the matrices Q_1 and Q_2 are the first columns of orthogonal matrices. The matrix R_1 is upper triangular, square and invertible since $\mathcal{A}_m$ has full column rank. However, the column rank of $\mathcal{M}_m$ is not necessarily full, unless we only have missing outputs. Hence we may write $R_2 = \begin{bmatrix} R_{21} & R_{22} \end{bmatrix}$, where R_{21} is upper triangular, square and invertible, see e.g. [8]. The cost for computing the QR factorizations are in the order of nmr where n is the number of rows, m the number of columns and r is the rank

of the matrix to be factorized. As upper bounds we may take $r = n_m$ and $m = n_m$, and hence we get nn_m^2.

We should now compute $e_o = \mathcal{A}_o y_o - \mathcal{B}_o u_o = \mathcal{A}(T_o^T y_o) - \mathcal{B}(R_m^T u_o)$ in (23) which are just linear filtrations, which we have already discussed. Hence the cost is in the order of nq.

To compute $x(\theta)$ in (23) we should solve a least squares problem, and the solution is given by $-E_2 R_{21}^{-1} Q_2^T e_o$. The cost of forming $Q_2 e_o$ is in the order of rn and solving the liner system of equations costs r^2 flops. Hence an upper bound of the cost is in the order of $nn_m + n_m^2$.

In the expression for γ_1 in (19) the quantity $\det(\mathcal{A}_m^T \mathcal{A}_m^T) = \det(R_1^T R_1)$ needs to be computed. It is easy to see that this can be done by squaring the elements in R_1, and for each column computing the sum of the elements, and then finally multiplying these sums with one another. The cost for this is in the order of the number of elements in R_1 which is upper bounded by n_m^2.

The residual r in (24) is given by $\gamma_1 e$, where $e = \mathcal{M}_m x(\theta) + e_o$. The flop count for this is in the order of nn_m. If we are interested in λ in (26) it can be efficiently computed as $e^T e/(n - \Sigma_{i=1}^n \Sigma_{j=1}^r Q_{2_{i,j}}^2)$ which has a flop count of $n + nr$, which is upper bounded by $n + nn_m$.

We will now start computing the gradient of the residual in (25). To this end we first form $y = T_m^T y_m(\theta) + T_o^T y_o$ and $y = R_m^T u_m(\theta) + R_o^T u_o$ which can be done in the order of zero flops. Given these quantities we should compute

$$\frac{\partial \mathcal{A}}{\partial \theta_k} y - \frac{\partial \mathcal{B}}{\partial \theta_k} u$$

As mentioned in Section 2 this is just filtrations, and notice that many of them are obtained by performing shifts. Hence the total cost is in the order of nq flops.

In the expression for the gradient of γ_1 in (20) we have $\frac{\partial \mathcal{A}_m}{\partial \theta_k}$. This can be obtained by doing similar filtrations as in the previous paragraph but one for each column in T_m^T. Hence the total cost for this is in the order of $nn_m q$ flops.

The flop count for computing the matrix within the trace operator in the expression for the partial derivatives for γ_1 is, since

$$\mathcal{A}_m^\dagger \frac{\partial \mathcal{A}_m}{\partial \theta_k} = E_1 R_1^{-1} Q_1^T \frac{\partial \mathcal{A}_m}{\partial \theta_k}$$

in the order of $qn_m(n_n^2 + n_m n)$. This is the most costly computation.

It remains to form the expression that should be projected in order to obtain the gradient g_{θ_k}, i.e.

$$\xi = \frac{\partial \gamma_1}{\partial \theta_k}(\mathcal{A}y - \mathcal{B}u) + \gamma_1 \left(\frac{\partial \mathcal{A}}{\partial \theta_k} y - \frac{\partial \mathcal{B}}{\partial \theta_k} u \right)$$

The cost for this is just nq. Then it holds that $g_{\theta_k} = \xi - Q_2(Q_2^T \xi)$, for which the flop count is bounded by the order of $n_m n$. We will report numerical experiments in relation to this analysis below.

7 Numerical examples

All numerical examples have been solved using Matlab's nonlinear least squares solver `lsqnonlin` with the options `Algorithm = 'trust-region-reflective'`, `TolFun = 1e-15` and `TolX = 1e-6/sqrt(np)`, where `np` is the number of estimated parameters. This code implements the methods described in [2, 3].

7.1 Example 1

The purpose of this example is to show that minimizing (11) does indeed yield biased parameter estimates but minimizing (10) does not. The input to the ARX system

$$y(k) = \frac{0.7}{1 + 0.7q^{-1}} u(k) + \frac{1}{1 + 0.7q^{-1}} e(k)$$

where q^{-1} is the time-shift operator, was a sequence of random variables equal to ± 1 with equal probability. The noise was Gaussian distributed with variance 0.5. The number of generated input sequences and noise sequences was 1000. This system has two parameters, $a_1 = 0.7$ and $b_1 = 0.7$. Data was generated from time $k = 1$ to time $k = 1000$. The percentage of missing output data was 40%. Data was missing at random time instants. Initial guesses for the parameters were zero. The results are shown in Figure 1 and

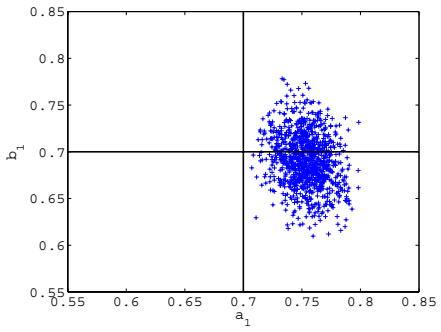

Figure 1: Estimates of the parameters in Example 1 using (11).

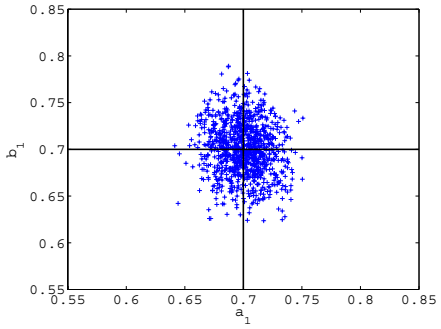

Figure 2: Estimates of the parameters in Example 1 using (12).

Figure 2. The values of the sample standard deviations are similar but the values of the sample means are not. A clear bias is seen when the Euclidean norm of the prediction error vector is minimized. Not a single one of the 1000 estimates is to the left of or on the line $a_1 = 0.7$.

7.2 Example 2

The purpose of this example is to show that it can pay off, in terms of accuracy of the estimated parameters, to introduce an input model if the input signal can be well modeled using a time series model. The input to the ARX system

$$y(k) = \frac{0.7}{1 + 0.7q^{-1}}u(k) + \frac{1}{1 + 0.7q^{-1}}e_1(k)$$

was generated by the AR process

$$u(k) = \frac{1}{1 + 0.2q^{-1} - 0.64q^{-2}}e_2(k).$$

The variance of e_1 and the variance of e_2 were 0.5. The number of generated input sequences and noise sequences were 1000. Data was generated from time $k = 1$ to time $k = 500$. The percentage of missing output data was 20% and the percentage of missing input data was 20%. Data was missing at random time instants. The parameters of the model where initialized by perturbing them such that the zeros of the polynomials defining the model was perturbed 10% of the distance to the origin in a random direction. The results are shown in Figure 3 and Figure 4. The sample standard deviation decreases a little bit

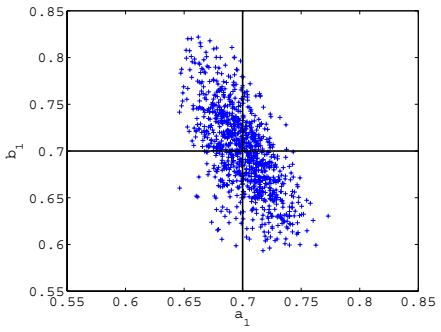

Figure 3: Estimates of the parameters in Example 2 using (12).

for a_1 and decreases significantly for b_1. It should be noted that it does not pay off as much to introduce an input model when the percentage of missing inputs is small.

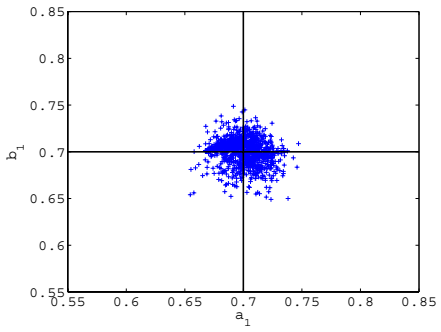

Figure 4: Estimates of the parameters in Example 2 using (15).

Table 2: Sample mean and sample standard deviation of the parameter estimates in Example 3.

Parameter	mean	standard deviation
a_1	1.6889	0.0302
a_2	1.1815	0.0434
a_3	0.3424	0.0196
b_1	2.0016	0.0404
b_2	-1.7341	0.0725
b_3	0.5052	0.0846
c_1	-0.0720	0.0739
c_2	-0.1239	0.0724
c_3	0.0811	0.0682

7.3 Example 3

In this example the input to the ARMAX system

$$y(k) = \frac{2q^{-1} - 1.7321q^{-2} + 0.5q^{-3}}{1 + 1.6899q^{-1} + 1.1830q^{-1} + 0.3430q^{-3}}u(k) + \frac{1 - 0.0657q^{-1} - 0.1228q^{-2} + 0.0800q^{-3}}{1 + 1.6899q^{-1} + 1.1830q^{-1} + 0.3430q^{-3}}e(k)$$

was a sequence of random variables equal to ± 1 with equal probability. The noise was Gaussian distributed with variance 0.5. The number of generated input sequences and noise sequences was 1000. Data was generated from time $k = 1$ to time $k = 500$. The percentage of missing output data was 25%. Data was missing at random time instants. The parameters of the model where initialized by perturbing them such that the zeros of the polynomials defining the model was perturbed 10% of the distance to the origin in a random direction. The results are shown in Table 2. It is seen that the model is estimated with very high accuracy.

7.4 Example 4

In this example we have considered 3 different ARMAX models with total number of parameters, excluding λ, equal to 3, 6 and 8. We have considered data lengths n equal to 200, 500 and 1000. The number of missing output data have been 20, 50 and 100. All 27 combinations of the above data has been investigated, and 20 identification experiments have been conducted for each of them. We report in Table 3 the mean and the standard deviation of the time per iteration and of the number of iterations of the nonlinear least squares solver. All examples have been initialized with a zero parameter vector. The experiments do not confirm the asymptotic estimates of computational complexity. There can be many reasons for this. Flop counts are not always very relevant on modern computer architectures. Prototype implementations in Matlab do not show the same behavior as does codes written in C or similar languages. Moreover, it might be that the experiments we have conducted are not in the regime where the asymptotic results are valid. However, the experiments indicate that it is possible to solve many problems of relevant size in reasonable time using a prototype implementation in Matlab. It should also be mentioned that for a few cases the solver did not converge. This is not surprising since the initialization of the parameters have been the zero vector.

8 Conclusions

In this paper a treatment of many common linear SISO models for system identification when data is missing has been presented. Missing input data can be considered as deterministic or stochastic. The stochastic point of view results in a lower variance of the parameter estimates when it is applicable. On the other hand the deterministic point of view works

Table 3: Table showing the computational time per iteration in seconds together with the number of iterations

.

Data length		200				500				1000			
Parameters	Missing data	Time	S.d.	Iter.	S.d.	Time	S.d.	Iter.	S.d.	Time	S.d.	Iter.	S.d.
3	20	0.0456	0.0082	10.80	1.6733	0.4899	0.0388	9.85	1.4244	3.9459	1.2217	8.60	0.8826
	50	0.0485	0.0032	11.50	2.6656	0.5460	0.020	10.05	2.2821	3.1415	1.1987	9.55	1.8771
	100	0.0646	0.0097	13.80	3.6361	0.7641	0.0762	11.05	2.6848	4.1493	0.2840	9.95	1.9324
6	20	0.0396	0.0048	23.25	6.7033	0.3195	0.0081	21.15	5.3240	1.8898	0.1649	20.85	8.3746
	50	0.0459	0.0012	20.95	4.0972	0.3606	0.0193	19.65	4.5800	2.1164	0.3272	20.25	7.5455
	100	0.0672	0.0027	16.80	3.8196	0.4019	0.0142	19.35	3.8289	2.5991	0.5029	19.20	6.4039
8	20	0.0411	0.0054	23.70	4.5895	0.3274	0.0065	22.70	2.3864	1.9343	0.1365	22.65	3.7173
	50	0.0537	0.0013	21.10	3.2911	0.4108	0.0161	22.35	3.6168	2.3455	0.2077	21.95	3.2521
	100	0.7010	0.0027	19.30	3.3261	0.4954	0.0317	21.55	3.2359	2.8353	0.2352	21.90	3.3230

also when the input signal cannot be well modeled by a time series.

Two easy to implement algorithms were tested on some numerical examples. The first one yields maximum likelihood estimates of the parameters when only output data is missing and the noise is Gaussian. The other yields maximum likelihood estimates of the parameters also when input data is missing and the noise is Gaussian.

When output data is missing it is not the Euclidean norm of the prediction error vector that should be minimized but rather the Euclidean norm of the prediction error vector scaled by a certain function of the covariance matrix of the observed output data. Neglecting to choose the right criterion will result in biased parameter estimates except for OE and FIR models.

It has been discussed how to implement the algorithms in an efficient way. The key is to use sparse linear algebra techniques.

It is possible to extend the results to multi-variable systems. The structure of the matrices depends on how the signals are partitioned. One possibility is to have blocks with lower triangular Toeplitz matrices. Actually the results are possible to extend to any model that is linear in the noise, the inputs and the outputs.

A shortcoming of the proposed method is that it is sensitive to the choice of initial values as is the case also for maximum likelihood estimation using gradient methods for the non-missing data case. We have tried to use `misdata` to initialize our algorithm but failed. The state-of-the-art technique to initialize maximum likelihood algorithms for the non-missing data case is to obtain initial values from a subspace method. We believe that the nuclear norm based methods mentioned in the introduction could be useful for this purpose, and this will be a topic for future research. Because of what has been said above we do not see our algorithm as a competitor to other methods for identification when data is missing, but as a complement, which should be used as a final step to obtain accurate estimates of models.

References

[1] C. F. Ansley and R. Kohn. Exact Likelihood of vector autoregressive-moving average with missing or aggregated data. *Biometrica*, 70(1):275–278, 1983.

[2] T. F. Coleman and Y. Li. On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to bounds. *Mathematical Programming*, 67(2):189–224, 1994.

[3] T. F. Coleman and Y. Li. An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6:418–445, 1996.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977.

[5] T. Ding, M. Sznaier, and O. Camps. A rank minimization approach to fast dynamic event detection and track matching in video sequences. In *Proceedings of the 46th IEEE conference on decision and control*, 2007.

[6] W. Dunsmuir and P. Robinson. Estimation of time series models in the presence of missing data. *Journal of the American Statistical Association*, 76(375):560–568, 1981.

[7] G. Golub and V. Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on Numerical Analysis*, 10(2):413–432, 1973.

[8] G. H. Golub and Ch. F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, Maryland, 1996. Third Edition.

[9] G. Goodwin and G. J. Adams. Multi-rate techniques in non-zero-order hold identification. In *Preprints of the 10th IFAC Symposium on System Identification*, volume 3, pages 125–130, Copenhagen, Denmark, 1994.

[10] C. Grossmann, C. N. Jones, and M. Morari. System identification via nuclear norm regularization for simulated bed processes from incomplete data sets. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 4692–4697, 2009.

[11] A. Hansson and R. Wallin. Maximum likelihood estimation of Gaussian models with missing data – Eight equivalent formulations. Technical Report LiTH-ISY-R-3013, Linköping Univerity, Linköping, Sweden, 2011.

[12] A. Hansson and R. Wallin. Maximum likelihood estimation of Gaussian models with missing data – Eight equivalent formulations. *Automatica*, 2012. Accepted for publication.

[13] A. C. Harvey and C. R. McKenzie. *In Ed. E. Parzen, Time Series Analysis of irregularly Observed Data*, chapter Missing observations in dynamic econometric models: A partial synthesis, pages 108–133. Springer Verlag, New York, New York, USA, 1984.

[14] A. J. Isaksson. Identification of ARX models subject to missing data. *IEEE Transactions on Automatic Control*, 38(5):813–819, 1993.

[15] A. J. Isaksson and V. Kaul. Survey of missing data identification techniques. In *Proceedings of Control Systems 92*, pages 95–99, 1992.

[16] R. H. Jones. Maximum likelihood fitting of ARMA models to time series with missing observations. *Technometrics*, 22:389–395, 1980.

[17] L. Kaufman. A variable projection method for solving separable nonlinear least squares problems. *BIT*, 15:49–57, 1975.

[18] J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data*. Prentice Hall, 1993.

[19] Z. Liu, A. Hansson, and L. Vandenberghe. Nuclear norm system identification with missing inputs and outputs. *Systems & Control Letters*, 62:605–612, 2013.

[20] L. Ljung. *System Identification*. Prentice Hall, Upper Saddle River, New Jersey, USA, 2nd edition, 1999.

[21] I. Markovsky. Data modeling using the nuclear norm heuristic. Technical Report 21936, ECS, University of Southampton, Southampton, 2011.

[22] P. B. McGiffin and D. N. Murthy. Parameter estimation for autoregressive systems with missing observations. *International Journal of Systems Science*, 11(9):1021–1034, 1980.

[23] P. B. McGiffin and D. N. Murthy. Parameter estimation for autoregressive systems with missing observations-Part II. *International Journal of Systems Science*, 12(6):657–663, 1981.

[24] R. B. Miller and O. Ferreiro. *In Ed. E. Parzen, Time Series Analysis of irregularly Observed Data*, chapter A strategy to complete a time series with missing observations, pages 251–275. Springer Verlag, New York, New York, USA, 1984.

[25] J. Penzer and B. Shea. The exact likelihood of an autoregressive-moving average model with incomplete data. *Biometrika*, 84(4):919–928, 1997.

[26] G. Pillonetto and A. Chiuso. A Bayesian learning approach to linear system identification with missing data. In *Proceedings of the joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pages 4698–4703, Shanghai, China, 2009.

[27] R. Pintelon and J. Schoukens. Frequency domain system identification with missing data. *IEEE Transactions on Automatic control*, 45(2):364–369, 2000.

[28] Y. Rosen and B. Porat. Optimal ARMA parameter estimation based on the sample covariances for data with missing observation. *IEEE Transactions on Information Theory*, 35(12):342–349, 1989.

[29] R. Sanchis, A. Sala, and P. Albertos. Scarce data operating conditions: Process model identification. In *Preprints of the 11th IFAC Symposium on System Identification*, volume 2, pages 463–468, Kitakyushu, Japan, 1997.

[30] J. D. Sargan and E. G. Drettakis. Missing data in autoregressive model. *Inernational Economic Review*, 15(1):39–58, 1974.

[31] T. Söderström and P. Stoica. *System identification*. Prentice Hall, Upper Saddle River, New Jersey, USA, 1989.

[32] P. Stoica, L. Xu, and J. Li. A new type of parameter estimation algorithm for missing data problems. *Statistics & Probability Letters*, 75:219–229, 2005.

[33] M. Tanaka. Identification of nonlinear systems with missing data using stochastic neural network. In *Proceedings of 35th IEEE Conference on Decision and Control*, volume 1, pages 933–4, 1996.

[34] M. Tanaka and T. Katayama. Robust identification and smoothing for linear system with outliers and missing data. In *Preprints 11th IFAC World Congress*, pages 160–165, Tallinn, Estonia, USSR, 1990.

[35] R. Wallin, A. J. Isaksson, and L. Ljung. An iterative method for identification of ARX models subject to missing data. In *Proceedings of the 39th IEEE Conference on Decision and Control*, 2000.

[36] E. Weinstein, M. Feder, and A. V. Oppenheim. Sequential algorithms for parameter estimation based on the kullback-liebler information measure. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-38:1652–1654, September 1990.

Appendix

The partial derivatives of the model matrices are:

$$\frac{\partial \mathcal{A}_i}{\partial a_{ik}} = A_i^{-1}\mathcal{A}_i S_n^k, \qquad k = 1, 2, \ldots, n_{ai} \tag{27}$$

$$\frac{\partial \mathcal{A}_1}{\partial b_k} = 0, \qquad k = 1, 2, \ldots, n_b \tag{28}$$

$$\frac{\partial \mathcal{A}_i}{\partial c_{ik}} = -C_i^{-1}\mathcal{A}_i S_n^k, \qquad k = 1, 2, \ldots, n_{ci} \tag{29}$$

$$\frac{\partial \mathcal{A}_i}{\partial d_{ik}} = D_i^{-1}\mathcal{A}_i S_n^k, \qquad k = 1, 2, \ldots, n_{di} \tag{30}$$

$$\frac{\partial \mathcal{A}_1}{\partial f_k} = 0, \qquad k = 1, 2, \ldots, n_f \tag{31}$$

$$\frac{\partial \mathcal{B}_1}{\partial a_{1k}} = 0, \qquad k = 1, 2, \ldots, n_{a1} \tag{32}$$

$$\frac{\partial \mathcal{B}_1}{\partial b_k} = B^{-1}\mathcal{B}_1 S_n^{k-1}, \qquad k = 1, 2, \ldots, n_b \tag{33}$$

$$\frac{\partial \mathcal{B}_1}{\partial c_{1k}} = -C_1^{-1}\mathcal{B}_1 S_n^k, \qquad k = 1, 2, \ldots, n_{c1} \tag{34}$$

$$\frac{\partial \mathcal{B}_1}{\partial d_{1k}} = D_1^{-1}\mathcal{B}_1 S_n^k, \qquad k = 1, 2, \ldots, n_{d1} \tag{35}$$

$$\frac{\partial \mathcal{B}_1}{\partial f_k} = -F^{-1}\mathcal{B}_1 S_n^k, \qquad k = 1, 2, \ldots, n_f. \tag{36}$$