Maximum likelihood estimation of Gaussian models with missing data-Eight equivalent formulations

Anders Hansson and Ragnar Wallin

Original Publication:

Anders Hansson and Ragnar Wallin, Maximum likelihood estimation of Gaussian models with missing data-Eight equivalent formulations, 2012, Automatica, (48), 9, 1955-1962.
http://dx.doi.org/10.1016/j.automatica.2012.05.060

Maximum Likelihood Estimation of Gaussian Models with Missing Data—Eight Equivalent Formulations ⋆

Anders Hansson [a], Ragnar Wallin [a]

[a] *Division of Automatic Control*
Linkoping University
SE–581 83 Linkoping, Sweden

Abstract

In this paper we derive the maximum likelihood problem for missing data from a Gaussian model. We present in total eight different equivalent formulations of the resulting optimization problem, four out of which are nonlinear least squares formulations. Among these formulations are also formulations based on the expectation-maximization algorithm. Expressions for the derivatives needed in order to solve the optimization problems are presented. We also present numerical comparisons for two of the formulations for an ARMAX model.

Key words: Maximum Likelihood Estimation, Missing Data, Expectation Maximization Algorithm, ARMAX Models

1 Introduction

Missing data is common in statistics, and it is important that this is addressed in a correct way in order to not disturb the conclusions drawn from the data, see e.g. [LR93]. In this paper we are interested in estimating parameters in a Gaussian model. A potential application we have in mind is linear models of dynamical systems, see e.g. [Lju99,SS83,VD96,PS01]. A popular method for this is Maximum Likelihood (ML) estimation.

ML estimation with missing data for these type of models have been considered by several authors. The first reference for ARMA models is [Jon80], where the log-likelihood for the problem is derived using a state-space formulation. In [WR86] instead an innovation transformation is used to derive the log-likelihood function. In [Isa93] it is for ARX models suggested that the so-called Expectation Maximization (EM) algorithm could be used, see e.g. [DLR77,Wu83].

We will revisit both the direct log-likelihood approach as well as the EM algorithm and in a setting that is more general. Just as for the case when data are not missing it is possible to reformulate the direct log-likelihood approach as a Nonlinear Least Squares (NLS) problem, [Lju99]. For the case of missing data it is also possible to interpret this reformulation as what is called a separable NLS problem, see [Bjö96]. This makes it possible to use efficient special purpose codes for such problems, [GP03]. We will also see that it is possible to reformulate the EM algorithm as a NLS problem. The main motivation for making reformulations as NLS problems is numerical accuracy and efficiency. Without these reformulations we where not able to solve more than very small-scale problems without running into numerical difficulties.

We will restrict ourselves to the case when the data is missing at random. Loosely speaking this means that the mechanism by which the data is missing is independent of the observed data, see [LR93] for the precise definition. When this is the case there is no need to involve the distribution for how the data is missing in the analysis. This assumption does not exclude missing data patterns that are deterministic.

The remaining part of the paper is organized as follows. At the end of this section notational conventions are given. In Section 2 we revisit the density function for a multivariate Gaussian random vector. In Section 3 we use the Schur complement formula to derive the marginal density function for the observed data. This is used in Section 4 to formulate the ML estimation problem for a

⋆ This paper was not presented at any IFAC meeting. Corresponding author Anders Hansson, Phone: +4613 281681, Fax: +4613 139282

Email addresses: hansson@isy.lu.se (Anders Hansson), ragnarw@isy.liu.se (Ragnar Wallin).

parameterized Gaussian model. The first order partial derivatives of the log-likelihood function with respect to the parameters are derived. Also the Fisher information matrix is recapitulated. Then, in Section 5 two equivalent ML problems are derived together with expressions for the partial derivatives of the objective functions. In Section 6 the EM algorithm is revisited for the Gaussian model, and the partial derivatives needed in order to carry out the optimization is derived. In Section 7 the four equivalent formulations are all reformulated as NLS problems. In order to test the different algorithms presented, an ARMAX model is introduced in Section 8. In Section 9 the numerical implementation of the algorithms are discussed. In Section 10 numerical results are presented for ARMAX models, and finally, in Section 11, some conclusions and directions for future research are given.

1.1 Notation

For two matrices X and Y of compatible dimensions we define their inner product $X \bullet Y = \mathrm{Tr} X^T Y$, where $\mathrm{Tr}(\cdot)$ is the trace operator. With $\mathrm{E}(\cdot)$ we denote the expectation operator with respect to a density function.

2 Gaussian Model

We consider an n-dimensional random vector e with zero mean Gaussian distribution and covariance matrix λI_n, where $\lambda > 0$, and where I_n is the $n \times n$ identity matrix. We will when it is obvious from the context omit the subscript n. In addition to this random vector we are also interested in an n-dimensional random vector x which is related to e via the affine equation

$$\Phi x + \Gamma = e \tag{1}$$

where we assume that Φ is invertible. The random vector x will contain both the observed and the missing data, as described in more detail later on. The matrix Φ and the vector Γ will depend on the parameter vector to be estimated. However, for the time being we suppress this dependence. The covariance matrix for x is

$$\Sigma = \lambda \left(\Phi^T \Phi\right)^{-1}$$

Moreover, the mean μ of x satisfies $\Phi\mu + \Gamma = 0$ and we can express μ as $\mu = -\Phi^{-1}\Gamma$. The density function for x is

$$p(x) = \frac{1}{\sqrt{(2\pi)^n \det(\Sigma)}} \times \exp\left\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right\} \tag{2}$$

We deliberately misuse x as both a random vector and as the argument of the density function for itself. It is straightforward to show that

$$p(x) = \frac{1}{\sqrt{(2\pi)^n \det(\lambda(\Phi^T\Phi)^{-1})}} \times \exp\left\{-\frac{1}{2\lambda}(\Phi x + \Gamma)^T(\Phi x + \Gamma)\right\}$$

3 Missing Data

We would like to separate the vector x in one part x_o that we call observed and one part x_m that we call missing or not observed. We do this with the permutation matrix $T = \begin{bmatrix} T_o \\ T_m \end{bmatrix}$ such that

$$\begin{bmatrix} x_o \\ x_m \end{bmatrix} = Tx = \begin{bmatrix} T_o \\ T_m \end{bmatrix} x$$

We also define

$$\begin{bmatrix} \Phi_o & \Phi_m \end{bmatrix} = \Phi T^T$$

so that we may write (1) as

$$\Phi x + \Gamma = \Phi_o x_o + \Phi_m x_m + \Gamma = e \tag{3}$$

Similarly we define $\begin{bmatrix} \mu_o \\ \mu_m \end{bmatrix} = T\mu$. We also define $\xi = x - \mu$, $\xi_o = x_o - \mu_o$, and $\xi_m = x_m - \mu_m$. Hence we may write the quadratic form in (2) as

$$\xi^T T \Sigma^{-1} T^T \xi$$

We are interested in the density function of x_o, since this would make it possible for us to, in case the mean and covariance of x depend in some way on a parameter vector, perform ML estimation of this parameter vector based on observations of only x_o. To this end we will transform the density function for x using the Schur complement formula, which is the following identity:

$$\begin{bmatrix} X & Y \\ Y^T & Z \end{bmatrix} = \begin{bmatrix} I & 0 \\ Z^{-1}Y^T & I \end{bmatrix}^T \begin{bmatrix} X - YZ^{-1}Y^T & 0 \\ 0 & Z \end{bmatrix} \times \begin{bmatrix} I & 0 \\ Z^{-1}Y^T & I \end{bmatrix} \tag{4}$$

With

$$\begin{bmatrix} X & Y \\ Y^T & Z \end{bmatrix} = T\Phi^T\Phi T^T = \begin{bmatrix} \Phi_o^T\Phi_o & \Phi_o^T\Phi_m \\ \Phi_m^T\Phi_o & \Phi_m^T\Phi_m \end{bmatrix}$$

we obtain the following center matrix in the right hand side of the Schur complement formula

$$\begin{bmatrix} \Phi_o^T P \Phi_o & 0 \\ 0 & Z \end{bmatrix} \tag{5}$$

where $P = I - \Phi_m Z^{-1} \Phi_m^T$ is a projection matrix which projects onto the orthogonal complement of the range space of Φ_m. As a projection matrix it has the following properties: $P^2 = P$, $P\Phi_m = 0$.

We realize from the preceding derivations that we may write the quadratic form in the exponent of the density function p in (2) as

$$\xi^T T \Sigma^{-1} T^T \xi = \xi_o^T \Phi_o^T P \Phi_o \xi_o + \tilde{\xi}_m^T Z \tilde{\xi}_m$$

where $\tilde{\xi}_m$ is defined via

$$\begin{aligned} \hat{\xi}_m &= -Z^{-1} Y^T \xi_o \\ \tilde{\xi}_m &= \xi_m - \hat{\xi}_m \end{aligned} \tag{6}$$

From this we realize that ξ_o and $\tilde{\xi}_m$ are zero mean independent Gaussian random vectors with covariance matrices

$$\begin{aligned} \Sigma_o &= \lambda \left(\Phi_o^T P \Phi_o \right)^{-1} \\ \tilde{\Sigma}_m &= \lambda Z^{-1} \end{aligned} \tag{7}$$

It is straight forward to verify that in the new variables (1) is equivalent to

$$\begin{bmatrix} P\Phi_o & \Phi_m \end{bmatrix} \begin{bmatrix} \xi_o \\ \tilde{\xi}_m \end{bmatrix} = e \tag{8}$$

The use of the Schur complement formula for obtaining the marginal density of x_o is mentioned e.g. in [Cot74].

4 Maximum Likelihood Estimation

We now assume that Φ and Γ are functions of a q-dimensional parameter vector θ. Then all the preceding covariance matrices and mean vectors will also be functions of θ. We are interested in performing ML estimation of θ and λ based on observations of x_o. We obtain the following log-likelihood function for x_o:

$$\begin{aligned} L(\lambda, \theta) = &\frac{1}{2\lambda} (x_o - \mu_o)^T \Phi_o^T P \Phi_o (x_o - \mu_o) \\ &+ \frac{n_o}{2} \log \lambda - \frac{1}{2} \log \det(\Phi_o^T P \Phi_o) \end{aligned} \tag{9}$$

where n_o is the dimension of x_o. From the Schur complement formula in (4) and the center matrix in (5) it follows that $\det \Phi^T \Phi = \det \Phi_o^T P \Phi_o \times \det Z$. Moreover, $P\Phi T^T T \mu = P \Phi_o \mu_o = -P\Gamma$. Hence we may re-write the log-likelihood function as

$$\begin{aligned} L(\lambda, \theta) = &\frac{1}{2\lambda} (\Phi_o x_o + \Gamma)^T P (\Phi_o x_o + \Gamma) + \frac{n_o}{2} \log \lambda \\ &- \frac{1}{2} \log \det(\Phi^T \Phi) + \frac{1}{2} \log \det(Z) \end{aligned}$$

where we have made use of the fact that $P^2 = P$. The advantage of this reformulation is that in the application we have in mind often $\det \Phi^T \Phi = 1$, and then this term disappears. However, also when this is not the case, to remove the explicit dependence on P is advantageous when performing the minimization of the log-likelihood function, since the P-dependence complicates the expressions for the derivatives needed in the optimization algorithms.

4.1 Derivatives

We will now derive the partial derivatives needed in order to optimize the log-likelihood function in (9). We will do these derivations term by term. To this end we write $L = f_1 + f_2 - f_3 + f_4$, where

$$\begin{aligned} f_1 &= \frac{1}{2\lambda} e_o^T P e_o \\ f_2 &= \frac{n_o}{2} \log \lambda \\ f_3 &= \frac{1}{2} \log \det W \\ f_4 &= \frac{1}{2} \log \det Z \end{aligned}$$

where $e_o = \Phi_o x_o + \Gamma$, and where $W = \Phi^T \Phi$. Then it follows that

$$\begin{aligned} \frac{\partial f_1}{\partial \theta_k} &= \frac{1}{\lambda} e_o^T P \frac{\partial e_o}{\partial \theta_k} + \frac{1}{2\lambda} e_o^T \frac{\partial P}{\partial \theta_k} e_o \\ \frac{\partial f_2}{\partial \theta_k} &= 0 \\ \frac{\partial f_3}{\partial \theta_k} &= \frac{1}{2} W^{-1} \bullet \frac{\partial W}{\partial \theta_k} \\ \frac{\partial f_4}{\partial \theta_k} &= \frac{1}{2} Z^{-1} \bullet \frac{\partial Z}{\partial \theta_k} \end{aligned}$$

where

$$\frac{\partial e_o}{\partial \theta_k} = \frac{\partial \Phi_o}{\partial \theta_k} x_o + \frac{\partial \Gamma}{\partial \theta_k}$$

where

$$\begin{aligned} \frac{\partial P}{\partial \theta_k} = \frac{\partial}{\partial \theta_k} \left(I - \Phi_m Z^{-1} \Phi_m^T \right) = &- \frac{\partial \Phi_m}{\partial \theta_k} Z^{-1} \Phi_m^T \\ &- \Phi_m \frac{\partial Z^{-1}}{\partial \theta_k} \Phi_m^T - \Phi_m Z^{-1} \frac{\partial \Phi_m^T}{\partial \theta_k} \end{aligned}$$

and where

$$\begin{aligned}\frac{\partial Z^{-1}}{\partial \theta_k} &= -Z^{-1}\frac{\partial Z}{\partial \theta_k}Z^{-1}\\ \frac{\partial Z}{\partial \theta_k} &= T_m\frac{\partial W}{\partial \theta_k}T_m^T\\ \frac{\partial W}{\partial \theta_k} &= \frac{\partial \Phi^T}{\partial \theta_k}\Phi + \Phi^T\frac{\partial \Phi}{\partial \theta_k}\end{aligned}$$

The partial derivatives with respect to λ are zero except for

$$\begin{aligned}\frac{\partial f_1}{\partial \lambda} &= \frac{-1}{\lambda}f_1\\ \frac{\partial f_2}{\partial \lambda} &= \frac{n_o}{2\lambda}\end{aligned}$$

4.2 Fisher Information Matrix

It is well-known that estimates obtained from ML estimation in most cases are unbiased and have covariance matrix equal to the inverse of the Fisher information matrix $\mathcal{I}$ assuming that the inverse exists. For Gaussian ML-problems there is a closed form expression for the elements of this matrix, [Kay93], and they are given here for short reference

$$\mathcal{I}_{k,l} = \frac{\partial \mu_o^T}{\partial \bar{\theta}_k}\Sigma_o^{-1}\frac{\partial \mu_o}{\partial \bar{\theta}_k} + \frac{1}{2}\left(\Sigma_o^{-1}\frac{\partial \Sigma_o}{\partial \bar{\theta}_k}\right)\bullet\left(\Sigma_o^{-1}\frac{\partial \Sigma_o}{\partial \bar{\theta}_l}\right)$$

where $\bar{\theta}^T = \begin{bmatrix}\lambda & \theta^T\end{bmatrix}^T$. With $U = \Phi_o^T P \Phi_o$ it holds that

$$\begin{aligned}\frac{\partial \mu_o}{\partial \lambda} &= 0\\ \frac{\partial \mu_o}{\partial \theta_k} &= -T_o\Phi^{-1}\left(\frac{\partial \Phi}{\partial \theta_k}\mu + \frac{\partial \Gamma}{\partial \theta_k}\right)\\ \frac{\partial \Sigma_o}{\partial \lambda} &= U^{-1}\\ \frac{\partial \Sigma_o}{\partial \theta_k} &= -\lambda U^{-1}\frac{\partial U}{\partial \theta_k}U^{-1}\\ \frac{\partial U}{\partial \theta_k} &= \frac{\partial \Phi_o^T}{\partial \theta_k}P\Phi_o + \Phi_o^T\frac{\partial P}{\partial \theta_k}\Phi_o + \Phi_o^T P\frac{\partial \Phi_o}{\partial \theta_k}\end{aligned}$$

5 Equivalent ML Problems

We add a term to the log-likelihood function in (9) involving $\tilde{\xi}_m$ and we also optimize over this new variable. The reason for doing this is that we will obtain a simpler optimization problem in terms of gradient expressions at the expense of introducing an extra variable. We hence consider an optimization problem involving the objective function

$$\begin{aligned}&\frac{1}{2\lambda}\tilde{\xi}_m^T\Phi_m^T\Phi_m\tilde{\xi}_m + \frac{1}{2\lambda}(\Phi_o x_o + \Gamma)^T P(\Phi_o x_o + \Gamma)\\ &+ \frac{n_o}{2}\log\lambda - \frac{1}{2}\log\det(\Phi^T\Phi) + \frac{1}{2}\log\det(Z)\end{aligned}$$

where we optimize over $(\lambda, \theta, \tilde{\xi}_m)$. The optimal value of $\tilde{\xi}_m$ is zero, since $\Phi_m^T\Phi_m$ is positive definite for any value of θ. Hence this optimization problem is equivalent to the original ML estimation problem. The first two terms in the objective function sum up to $e^T e/(2\lambda)$ by (8). Hence yet another equivalent optimization problem is to consider the objective function

$$\begin{aligned}l_c(\lambda, \theta, x_m, e) = &\frac{1}{2\lambda}e^T e + \frac{n_o}{2}\log\lambda\\ &- \frac{1}{2}\log\det(\Phi^T\Phi) + \frac{1}{2}\log\det(Z)\end{aligned} \quad (10)$$

and to optimize this objective function over $(\lambda, \theta, x_m, e)$ subject to the constraint (3). Finally we can substitute e in the objective function using (3) in order to obtain an unconstrained problem with variables (λ, θ, x_m) which has objective function

$$\begin{aligned}l(\lambda, \theta, x_m) = &\frac{1}{2\lambda}(\Phi x + \Gamma)^T(\Phi x + \Gamma) + \frac{n_o}{2}\log\lambda\\ &- \frac{1}{2}\log\det(\Phi^T\Phi) + \frac{1}{2}\log\det(Z)\end{aligned} \quad (11)$$

This is also an optimization problem equivalent to the ML estimation problem.

5.1 Derivatives

We realize that the only new term in the objective function in (11) is

$$f_0 = \frac{1}{2\lambda}e^T e$$

where $e = \Phi x + \Gamma$, i.e. $l = f_0 + f_2 - f_3 + f_4$. The first order derivatives are

$$\begin{aligned}\frac{\partial f_0}{\partial \theta_k} &= \frac{1}{\lambda}e^T\frac{\partial e}{\partial \theta_k}\\ \frac{\partial f_0}{\partial x_{m_k}} &= \frac{1}{\lambda}e^T\frac{\partial e}{\partial x_{m_k}}\\ \frac{\partial f_0}{\partial \lambda} &= -\frac{1}{\lambda}f_0\end{aligned}$$

where

$$\begin{aligned}\frac{\partial e}{\partial \theta_k} &= \frac{\partial \Phi}{\partial \theta_k}x + \frac{\partial \Gamma}{\partial \theta_k}\\ \frac{\partial e}{\partial x_{m_k}} &= (\Phi_m)_k\end{aligned}$$

The computation of the derivatives is less demanding for this formulation of the ML estimation problem. However, we have one more optimization vector x_m.

For the case when we keep the constraint (1) and use the objective function in (10) we have almost the same new term in the objective function, i.e. $l_c = f_{0c} + f_2 - f_3 + f_4$, where

$$f_{0c} = \frac{1}{2\lambda} e^T e$$

is now defined to be a function of (λ, e), i.e. it does not depend on θ. The partial derivatives with respect to λ are the same as the ones for f_0. With respect to e we get

$$\frac{\partial f_{0c}}{\partial e} = \frac{1}{\lambda} e$$

The variable θ is only present in f_3 and f_4. All variables except λ are present in the constraint

$$g(\theta, x_m, e) = \Phi x + \Gamma - e = 0 \tag{12}$$

that typically needs to be linearized in optimization algorithms. To this end the first order partial derivatives of g are needed:

$$\begin{aligned} \frac{\partial g}{\partial \theta_k} &= \frac{\partial \Phi}{\partial \theta_k} x + \frac{\partial \Gamma}{\partial \theta_k} \\ \frac{\partial g}{\partial x_{m_k}} &= (\Phi_m)_k \\ \frac{\partial g}{\partial e_k} &= (I)_k \end{aligned}$$

6 Expectation-Maximization Algorithm

Because of the computational complexity of ML estimation when data is missing for the first approach presented above, the EM algorithm has been suggested as a remedy. We will here revisit it for our problem. The idea is to recursively update the parameter vector (λ, θ) based on the previous value of the parameter vector (λ^-, θ^-) by minimizing

$$Q(\lambda, \theta) = \mathrm{E}\left\{ -\log p(x, \lambda, \theta) \mid x_o, \lambda^-, \theta^- \right\} \tag{13}$$

with respect to (λ, θ), where the conditional density of x given x_o based on the previous value of the parameter vector (λ^-, θ^-) is used to evaluate the expectation.

We immediately realize that we may write

$$-\log p(x, \lambda, \theta) = \frac{1}{2\lambda} (\Phi(\theta)x + \Gamma(\theta))^T (\Phi(\theta)x + \Gamma(\theta)) \tag{14}$$

$$+ \frac{n}{2} \log \lambda - \frac{1}{2} \log \det(\Phi^T(\theta)\Phi(\theta)) \tag{15}$$

We write $Q = F_1 + F_2 - F_3$, where

$$F_1 = \mathrm{E}\left\{ \frac{1}{2\lambda} (\Phi(\theta)x + \Gamma(\theta))^T (\Phi(\theta)x + \Gamma(\theta)) \mid x_o, \lambda^-, \theta^- \right\} \tag{16}$$

$$F_2 = \mathrm{E}\left\{ \frac{n}{2} \log \lambda \mid x_o, \lambda^-, \theta^- \right\} = \frac{n}{2} \log \lambda \tag{17}$$

$$F_3 = \mathrm{E}\left\{ \frac{1}{2} \log \det(\Phi^T(\theta)\Phi(\theta)) \mid x_o, \lambda^-, \theta^- \right\} = f_3 \tag{18}$$

It remains to evaluate the expectation for the first term. To this end we remember that by (6) and the definitions following (3) we may write

$$x_m(\theta^-) = \hat{\xi}_m(\theta^-) + \tilde{\xi}_m(\theta^-) + \mu_m(\theta^-)$$

where $\hat{\xi}_m(\theta^-)$ is a function of $\xi_o(\theta^-) = x_o(\theta^-) - \mu_o(\theta^-)$. Hence, with this change of variables from $x_m(\theta^-)$ to $\tilde{\xi}_m(\theta^-)$, for which $\tilde{\xi}_m(\theta^-)$ and $x_o(\theta^-)$ are independent, it follows that

$$\begin{aligned} F_1 = \mathrm{E}\Bigg\{ & \frac{1}{2\lambda} \begin{bmatrix} 0 \\ \tilde{\xi}_m(\theta^-) \end{bmatrix}^T T\Phi^T(\theta)\Phi(\theta)T^T \begin{bmatrix} 0 \\ \tilde{\xi}_m(\theta^-) \end{bmatrix} \Bigg\} \\ & + \frac{1}{2\lambda} \left\{ \Phi(\theta)T^T \begin{bmatrix} x_o \\ \hat{x}_m(\theta^-) \end{bmatrix} + \Gamma(\theta) \right\}^T \\ & \times \left\{ \Phi(\theta)T^T \begin{bmatrix} x_o \\ \hat{x}_m(\theta^-) \end{bmatrix} + \Gamma(\theta) \right\} \\ & = \frac{\lambda^-}{2\lambda} \mathrm{Tr}\left\{ \Phi_m(\theta)^T \Phi_m(\theta) Z(\theta^-)^{-1} \right\} \\ & + \frac{1}{2\lambda} \left\{ \Phi(\theta)T^T \begin{bmatrix} x_o \\ \hat{x}_m(\theta^-) \end{bmatrix} + \Gamma(\theta) \right\}^T \\ & \times \left\{ \Phi(\theta)T^T \begin{bmatrix} x_o \\ \hat{x}_m(\theta^-) \end{bmatrix} + \Gamma(\theta) \right\} \end{aligned}$$

where

$$\begin{aligned} \hat{x}_m(\theta^-) &= \hat{\xi}_m(\theta^-) + \mu_m(\theta^-) \\ &= -Z(\theta^-)^{-1} Y(\theta^-)^T \xi_0(\theta^-) + \mu_m(\theta^-) \\ &= -Z(\theta^-)^{-1} \Phi_m(\theta^-)^T \left(\Phi_o(\theta^-) x_o + \Gamma(\theta^-) \right) \end{aligned}$$

We may now write $F_1 = F_{11} + F_{12}$, where

$$\begin{aligned} F_{11} &= \frac{\lambda^-}{2\lambda} \mathrm{Tr}\left\{ Z(\theta^-)^{-1} Z(\theta) \right\} \\ F_{12} &= \frac{1}{2\lambda} \hat{e}(\theta, \theta^-)^T \hat{e}(\theta, \theta^-) \end{aligned}$$

and where

$$\hat{x}(\theta^-) = \begin{bmatrix} x_o \\ \hat{x}_m(\theta^-) \end{bmatrix}$$
$$\hat{e}(\theta, \theta^-) = \Phi(\theta) T^T \hat{x}(\theta^-) + \Gamma(\theta)$$

6.1 Derivatives

We will in this section not explicitly write out the dependence on (λ, θ). The partial derivatives of the different terms of Q with respect to θ_k are given by

$$\frac{\partial F_{11}}{\partial \theta_k} = \frac{\lambda^-}{2\lambda} Z(\theta^-)^{-1} \bullet \frac{\partial Z}{\partial \theta_k}$$
$$\frac{\partial F_{12}}{\partial \theta_k} = \frac{1}{\lambda} \hat{e}(\theta, \theta^-)^T \frac{\partial \hat{e}(\theta, \theta^-)}{\partial \theta_k}$$
$$\frac{\partial F_2}{\partial \theta_k} = 0$$

where

$$\frac{\partial \hat{e}(\theta, \theta^-)}{\partial \theta_k} = \frac{\partial \Phi}{\partial \theta_k} T^T \hat{x}(\theta^-) + \frac{\partial \Gamma}{\partial \theta_k}$$

With respect to λ we get

$$\frac{\partial F_{11}}{\partial \lambda} = -\frac{1}{\lambda} F_{11}$$
$$\frac{\partial F_{12}}{\partial \lambda} = -\frac{1}{\lambda} F_{12}$$
$$\frac{\partial F_2}{\partial \lambda} = \frac{n}{2\lambda}$$

7 Nonlinear Least Squares Reformulation

In this section we will reformulate the previous four formulations as NLS problems by analytically performing the minimization with respect to λ and eliminating this variable. The reason for this reformulation is that there are very efficient algorithms available for NLS problems, e.g. [Bjö96,NW06].

7.1 Original ML Problem

Using the optimality condition that the gradient of L in (9) with respect to λ is zero results in $\lambda = e_o^T P e_o / n_o$, which after back-substitution results in the following log-likelihood function to be minimized with respect to θ:

$$\frac{n_o}{2} \log \frac{e_o^T P e_o}{n_o} - \frac{1}{2} \log \det W + \frac{1}{2} \log \det Z$$

Writing this as one logarithm and realizing that it is equivalent to minimize the argument of the logarithm, the following NLS problem results as an equivalent optimization problem:

$$\min_\theta \|R(\theta)\|_2^2$$

where $R(\theta) = \gamma(\theta) P e_o$, and $\gamma(\theta) = \frac{1}{\sqrt{n_o}} \left(\frac{\det Z}{\det W} \right)^{\frac{1}{2n_o}}$. Most numerical methods for NLS problems only use first order derivatives, and it is straight forward to verify that

$$\frac{\partial R}{\partial \theta_k} = \frac{\partial \gamma}{\partial \theta_k} P e_o + \gamma \frac{\partial P}{\partial \theta_k} e_o + \gamma P \frac{\partial e_o}{\partial \theta_k}$$

where

$$\sqrt{n_o} \frac{\partial \gamma}{\partial \theta_k} = \frac{1}{2n_o} (\det Z)^{-1+\frac{1}{2n_o}} \frac{\partial \det Z}{\partial \theta_k} (\det W)^{-\frac{1}{2n_o}} - \frac{1}{2n_o} (\det W)^{-1-\frac{1}{2n_o}} \frac{\partial \det W}{\partial \theta_k} (\det Z)^{\frac{1}{2n_o}}$$

and where

$$\frac{\partial \det Z}{\partial \theta_k} = \det Z \times Z^{-1} \bullet \frac{\partial Z}{\partial \theta_k}$$
$$\frac{\partial \det W}{\partial \theta_k} = \det W \times W^{-1} \bullet \frac{\partial W}{\partial \theta_k}$$

7.2 Formulation Containing Missing Data

Proceeding as above it follows that for l in (11) the optimal value of λ is $\lambda = e^T e / n_o$, and that the equivalent NLS problem is

$$\min_{\theta, x_m} \|r(\theta, x_m)\|_2^2$$

where $r(\theta, x_m) = \gamma e$. The first order partial derivatives are

$$\frac{\partial r}{\partial \theta_k} = \frac{\partial \gamma}{\partial \theta_k} e + \gamma \frac{\partial e}{\partial \theta_k}$$
$$\frac{\partial r}{\partial x_{m_k}} = \gamma \frac{\partial e}{\partial x_{m_k}}$$

We remark that r is linear in x_m, and hence this NLS problem is a separable NLS problem, [Bjö96], and it is straight forward to verify that by analytically minimizing with respect to x_m, the NLS problem of the previous subsection is obtained. The minimizing argument is $x_m = -Z^{-1} \Phi_m^T e_o$.

7.3 Constrained Formulation

Proceeding as above for l_c in (10) it follows that the optimal value of λ still is $\lambda = e^T e / n_o$, and that the

equivalent NLS problem now is constrained:

$$\min_{\theta,x_m,e} \|r_c(\theta,e)\|_2^2$$
$$\text{s.t. } g((\theta,x_m,e)=0$$

where $r_c(\theta,e)=\gamma e$, and where g is defined as in (12). The first order partial derivatives are

$$\frac{\partial r_c}{\partial \theta_k}=\frac{\partial \gamma}{\partial \theta_k}e$$
$$\frac{\partial r_c}{\partial e_k}=\gamma$$

7.4 EM Algorithm

Using the optimality condition that the gradient of Q in (13) with respect to λ is zero results in

$$\lambda=\frac{\hat{e}^T\hat{e}}{n}+\frac{\lambda^-\mathrm{Tr}\left(Z(\theta)Z^{-1}(\theta^-)\right)}{n}$$

which after back-substitution results in $\log\lambda$ to be minimized with respect to θ. It is equivalent to minimize λ or to solve the NLS

$$\min_\theta \|r_Q(\theta)\|_2^2$$

where with vec denoting the vectorization operator, see e.g. [Lüt96],

$$\sqrt{n}r_Q(\theta)=\begin{bmatrix}\hat{e}\\ \mathrm{vec}\Xi\end{bmatrix}$$

where $\Xi=\sqrt{\lambda^-}Z^{1/2}(\theta)Z^{-1/2}(\theta^-)$, and where we have used the symmetric square root of a symmetric positive definite matrix. It follows that

$$\sqrt{n}\frac{\partial r_Q}{\partial\theta_k}=\begin{bmatrix}\frac{\partial\hat{e}}{\partial\theta_k}\\ \mathrm{vec}\frac{\partial\Xi}{\partial\theta_k}\end{bmatrix}$$

where $\frac{\partial\Xi}{\partial\theta_k}=\sqrt{\lambda^-}\frac{\partial Z^{1/2}}{\partial\theta_k}Z^{-1/2}(\theta^-)$ and where $\frac{\partial Z^{1/2}}{\partial\theta_k}$ is, since $Z^{1/2}$ does not have any imaginary axis eigenvalues, the unique solution to the algebraic Lyapunov equation

$$\frac{\partial Z^{1/2}}{\partial\theta_k}Z^{1/2}+Z^{1/2}\frac{\partial Z^{1/2}}{\partial\theta_k}=\frac{\partial Z}{\partial\theta_k}$$

The proof follows by applying the chain rule to the identity $Z^{1/2}Z^{1/2}=Z$.

8 ARMAX Model

Consider an ARMAX-model

$$\begin{aligned}&y_k+a_1y_{k-1}+\ldots+a_{n_a}y_{k-n_a}\\&=b_0u_k+b_1u_{k-1}+\ldots+b_{n_b}u_{k-n_b}\\&+e_k+c_1e_{k-1}+\ldots+c_{n_c}e_{k-n_c}\end{aligned}$$

for $k=1,2,\ldots,n$, which assuming that $y_0=y_{-1}=\ldots=y_{-n_a}=0$, $u_0=u_{-1}=\ldots=u_{-n_b}=0$ and $e_0=e_{-1}=\ldots=e_{-n_c}=0$ equivalently can be written as

$$Ay=Bu+Ce$$

where A, B, and C are lower triangular Toeplitz matrices with first columns

$$\begin{bmatrix}1\\a\\0\end{bmatrix};\quad\begin{bmatrix}b\\0\end{bmatrix};\quad\begin{bmatrix}1\\c\\0\end{bmatrix}$$

respectively, where $a^T=\begin{bmatrix}a_1&\cdots&a_{n_a}\end{bmatrix}$, $b^T=\begin{bmatrix}b_0&\cdots&b_{n_b}\end{bmatrix}$, and $c^T=\begin{bmatrix}c_1&\cdots&c_{n_c}\end{bmatrix}$. Furthermore $y^T=\begin{bmatrix}y_1&y_2&\cdots&y_n\end{bmatrix}$, $u^T=\begin{bmatrix}u_1&u_2&\cdots&u_n\end{bmatrix}$, and $e^T=\begin{bmatrix}e_1&e_2&\cdots&e_n\end{bmatrix}$. We assume as before that e is an n-dimensional random vector with Gaussian distribution of zero mean and covariance λI.

We now let $x=y$, $\Phi=C^{-1}A$, $\Gamma=-C^{-1}Bu$ and $\theta^T=\begin{bmatrix}a^T&b^T&c^T\end{bmatrix}$. We notice that Φ is invertible and that $\det\Phi^T\Phi=1$ for all values of θ.

The first order partial derivatives of Φ and Γ with respect to θ are

$$\frac{\partial\Phi}{\partial a_k}=E_k$$
$$\frac{\partial\Phi}{\partial b_k}=0$$
$$\frac{\partial\Phi}{\partial c_k}=-E_k\Phi$$
$$\frac{\partial\Gamma}{\partial a_k}=0$$
$$\frac{\partial\Gamma}{\partial b_k}=-E_ku$$
$$\frac{\partial\Gamma}{\partial c_k}=-E_k\Gamma$$

where $E_k=C^{-1}S_k$, and where S_k is a square shift matrix with zeros except for ones in the kth sub-diagonal.

9 Numerical Implementation

The first four formulations of the ML estimation problem suffer from numerical difficulties, and we will present no results regarding their performance. For the last four formulations, which are all NLS formulations, we have implemented two of them, see below for motivation. The numerical implementation has been carried out in Matlab using its optimization toolbox with the function `lsqnonlin`, which implements a NLS algorithm using first order derivatives only. This is a standard approach in system identification, see e.g. [Lju99]. We have used the trust-region-reflective option. See [Bjö96] and the references therein for more details on different types of NLS algorithms. The following tolerances for termination have been used for the method described in Section 7.1, which we from now on call the variable projection method: `TolFun` $= 10^{-15}$ and `TolX` $= 10^{-5}/\sqrt{q}$.

There are many other codes available than the one in the Matlab optimization toolbox. It should also be mentioned that there is a potential to obtain faster convergence and better numerical behavior by utilizing special purpose codes for separable NLS problems, see e.g. the survey [GP03] and the references therein. Because of this we have used the so-called Kaufman search direction when implementing the variable projection method in Section 7.1. This avoids computing the exact partial derivatives of P, which are the most costly derivatives to compute. According to [GP03] this is the best choice for separable NLS problems, and it also outperforms the formulation in Section 7.2 which does not need to compute the partial derivatives of P. In the appendix we show how the Kaufman search direction can be implemented in any standard NLS solver. It is actually possible to interpret the Kaufman search direction for the method in Section 7.1 as a block coordinate descent method for the method in Section 7.2 where at each iteration a descent step is taken in the θ-direction and an exact minimization is performed in the x_m-direction. This is the reason why the Kaufman search direction is preferable, [Par85]. To summarize, it can be interpreted either as an approximation of the gradient for the method in Section 7.1, see the appendix, or as an efficient block-coordinate method for the method in Section 7.2.

The EM method in Section 7.4 is often proposed as a good method to use when data is missing, but it is in general not very fast unless it is possible to solve the minimization step quickly. For certain problems there are closed form solutions, [Isa93]. In our case this is not true in general. However, it is not necessary to perform the minimization step to very high accuracy. It is often sufficient to just obtain a decrease in Q. These type of algorithms are usually called generalized EM algorithms, e.g. [MK08]. Here we will employ such an algorithm for optimizing Q by running two iterations of `lsqnonlin` for the first five iterations in the EM algorithm, and thereafter running one iteration. The overall termination criteria for the EM method is $\|\bar{\theta} - \bar{\theta}^-\|_2 \leq 10^{-5}/\sqrt{q}$.

We have not considered to implement the constrained formulation of the NLS problem in Section 7.3. This could potentially be a good candidate, especially for the application to ARMAX models in case the constraint is multiplied with C, since then the constraint will become bilinear in the variables. In addition to this the derivatives with respect to the objective function and the constraints are very simple. However, the application of constrained NLS for black box identification in [VMSVD10] does not show that it is advantageous with respect to computational speed for that application. However, it has the advantage that it can address unstable models.

The computational complexity in terms of flop count per iteration for computing the gradients is for all algorithms of the same order. This is the most time-consuming part of an iteration. The flop count is linear in the number of parameters q, quadratic in the underlying number of data n, and cubical in the number of missing data $n_m = n - n_o$. It should me mentioned that for the ARMAX model it is possible to make use of the Toeplitz structure of A, B, and C to decrease the computational time for the partial derivatives of Φ and Γ. Further speedup can be obtained by changing the order in which the computations are performed. This is not within the scope of this work and is left for future research.

10 Examples

In this section we will evaluate the variable projection method in Section 7.1 and the EM method in Section 7.4 on ARMAX examples of different complexity. The data used has been generated from models with θ-parameters defined by the polynomials in the forward shift operator q seen in Table 1. The variance λ was 0.5. The values of n have been $(300, 400, 500)$ and the percentages of missing data have been $(10, 20, 30)$, and their occurrence in time has been picked randomly with equal probability. The input signal u was a sequence of independent random variables equal to ± 1 with equal probability. Hence in total nine different examples have been investigated for the two methods. To initialize the optimization methods the poles and the zeros have been moved 10% of their distance to the origin in a random direction. For each problem 20 different realizations of the random vector e have been considered, and the results presented are estimated means together with estimated standard deviations for computational time in seconds and number of iterations for the different methods.

The results of the experiments are presented in tables 2–4. It is seen that that computational times and number of iterations grow for increasing model order and increasing percentage of missing data for all methods. The variable projection method is always faster than the

Table 1
Models

Model Order	a	b	c
1	$q + 0.7$	$2q$	$q + 0.5$
3	$(q + 0.7)(q^2 + \sqrt{2} \times 0.7q + 0.7^2)$	$2q(q^2 - \sqrt{3} \times 0.5q + 0.5^2)$	$(q + 0.5)()q^2 - \sqrt{2} \times 0.4q + 0.4^2)$
5	$(q + 0.7)(q^2 + \sqrt{2} \times 0.7q + 0.7^2)(q^2 + 0.7^2)$	$2q(q^2 - \sqrt{3} \times 0.5q + 0.5^2)(q^2 + 0.5^2)$	$(q + 0.5)(q^2 - \sqrt{2} \times 0.4q + 0.4^2)(q^2 + \sqrt{2} \times 0.5q + 0.5^2)$

Table 2
Results, 10% missing data

Method		Variable Projection				EM			
Model Order	Data Length	Time	S.d.	Iter.	S.d.	Time	S.d.	Iter.	S.d.
1	300	0.8015	0.1335	5.7	1.160	1.800	0.1597	11.4	0.9661
	400	1.715	0.2368	6.3	1.1595	3.243	0.4603	11.0	1.700
	500	2.377	0.3057	4.8	0.9189	5.557	0.6041	11.0	1.054
3	300	2.257	0.5383	8.0	2.3094	7.650	1.470	21.5	3.206
	400	4.332	0.5336	7.2	1.033	13.41	1.822	19.6	2.459
	500	7.666	1.444	6.4	1.1738	28.30	6.353	21.4	1.838
5	300	7.499	2.494	19.4	7.214	18.54	6.703	31.2	9.004
	400	16.01	5.208	19.5	6.980	31.37	5.851	26.6	3.836
	500	32.58	11.25	22.1	8.900	67.66	28.78	30.5	11.14

EM method. Both methods are sensitive with respect to initialization of the parameter vector. The reason for this is that the optimization problems are non-convex, and hence there are potentially several local minima of the objective functions. However, the often put forward comment that the EM method should be less sensitive with respect to initialization, has not been seen to be true in our experiments.

11 Conclusions

In this paper the maximum likelihood problem for missing data from a Gaussian model has been revisited. We have presented eight different equivalent formulations of the resulting optimization problem, four out of which are nonlinear least squares formulations. Two of the formulations are based on the expectation-maximization algorithm. Expressions for the derivatives needed in order to solve the optimization problems have been presented. We have also presented numerical comparisons for two of the formulations for an ARMAX model. It has been seen that the variable projection method results in the most efficient implementation. In [WH11] more details for applications to dynamic models such as ARX, ARMAX and Box Jenkins are given.

References

[Bjö96] Å. Björk. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, 1996.

[Cot74] R. W. Cottle. Manifestations of the Schur complement. *Linear Algebra and its Applications*, 8:189–211, 1974.

[DLR77] 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.

[GP03] G. Golub and V. Pereyra. Separable nonlinear least squares: the variable projection method and its applications. *Inverse Problems*, 19:R1–R26, 1003.

[Isa93] A. J. Isaksson. Identification of ARX models subject to missing data. *IEEE Transactions on Automatic Control*, 38(5):813–819, 1993.

[Jon80] R. H. Jones. Maximum likelihood fitting of ARMA models to time series with missing observations. *Technometrics*, 22:389–395, 1980.

[Kau75] L. Kaufman. A variable projection method for solving separable nonlinear least squares problems. *BIT*, 15:49–57, 1975.

[Kay93] S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 1993.

[Lju99] L. Ljung. *System Identification*. Prentice Hall, Upper Saddle River, New Jersey, USA, 2nd edition, 1999.

[LR93] J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data*. Prentice Hall, 1993.

[Lüt96] H. Lütkepohl. *Handbook of Matrices*. John Wiley & Sons, Chichester, 1996.

[MK08] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extension*. John Wiley & Sons, New Jersey, 2008.

[NW06] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2006.

[Par85] T. A. Parks. Reducible nonlinear programming problems. Ph. D. dissertation, Rice University, 1985.

[PS01] R. Pintelon and J. Schoukens. *System identification: A frequency domain approach*. IEEE Press, New York, New York, USA, 2001.

[SS83] T. Söderström and P. Stoica. *Instrumental variable methods for system identification*. Number 57 in

Table 3
Results, 20% missing data

Method		Variable Projection				EM			
Model Order	Data Length	Time	S.d.	Iter.	S.d.	Time	S.d.	Iter.	S.d.
1	300	0.8093	0.1427	5.0	1.155	2.941	0.3466	16.4	1.838
	400	1.598	0.1807	5.2	0.9189	4.642	0.5595	13.6	1.578
	500	2.567	0.3166	4.7	0.9487	8.674	1.079	14.8	1.687
3	300	2.545	0.3822	8.4	1.506	13.828	4.398	29.2	5.789
	400	5.068	1.211	8.4	2.547	27.54	7.094	30.4	6.687
	500	9.096	2.887	8.3	3.498	46.37	9.987	28.7	3.466
5	300	8.665	2.583	21.9	6.999	33.79	21.97	42.9	25.98
	400	21.33	7.627	25.4	9.119	60.94	21.19	37.1	11.54
	500	36.71	13.89	26.0	10.47	102.4	36.44	37.1	12.36

Table 4
Results, 30% missing data

Method		Variable Projection				EM			
Model Order	Data Length	Time	S.d.	Iter.	S.d.	Time	S.d.	Iter.	S.d.
1	300	0.9975	0.1463	6.4	1.174	4.582	0.7679	20.4	2.952
	400	2.101	0.5871	6.8	2.251	8.884	1.221	20.4	2.319
	500	3.231	0.9740	5.8	1.135	14.55	3.433	19.0	2.867
3	300	3.059	1.020	10.5	4.223	25.39	8.742	42.9	12.42
	400	6.797	2.216	11.2	3.994	45.72	11.81	39.1	8.504
	500	12.79	8.833	10.0	5.333	109.4	57.86	44.4	18.66
5	300	10.68	3.327	26.8	8.753	64.87	21.78	65.8	19.98
	400	23.0	8.200	26.8	10.90	124.8	56.64	59.5	25.93
	500	38.56	9.302	25.5	6.819	166.6	46.38	45.8	12.18

Lecture notes in control and information sciences. Springer Verlag, New York, New York, USA, 1983.

[VD96] P. Van Overschee and B. DeMoor. *Subspace identification for linear systems: Theory, implementation, applications*. Kluwer, Boston, Massachusetts, USA, 1996.

[VMSVD10] A. Van Mulders, J. Schoukens, M. Volckaert, and M. Diehl. Two nonlinear optimization methods for black box identification compared. *Automatica*, 46:1675–1681, 2010.

[WH11] R. Wallin and A. Hansson. System identification of linear SISO models subject to missing output data and input data. *Automatica*, 2011. submitted for possible publication.

[WR86] M. A. Wincek and G. C. Reinsel. An exact maximum likelihood estimation procedure for regression-arma time series models with possibly nonconsecutive data. *Journal of the Royal Statistical Society. Series B (Methodogical)*, 48(3):303–313, 1986.

[Wu83] C. F. J. Wu. On the converegence properties of the EM algorithm. *Annals of Statistics*, 11:95–103, 1983.

Appendix—Kaufman Search Direction

In this appendix we will revisit the Kaufman search direction. Our derivation is based on [Par85]. Consider a separable NLS problem on the form

$$\min_{x,\alpha} \frac{1}{2}\|F(x,\alpha)\|_2^2 \tag{19}$$

where $F(x,\alpha) = A(\alpha)x - b(\alpha)$. Also consider the reduced problem by explicit minimization above with respect to x:

$$\min_{\alpha} \frac{1}{2}\|f(\alpha)\|_2^2 \tag{20}$$

where $f(\alpha) = F(x(\alpha),\alpha)$ with $x(\alpha) = (A^TA)^{-1}A^Tb$, i.e. $f(\alpha) = -Pb$, where $P = I - A(A^TA)^{-1}A^T$ is a projection matrix.

The exact search direction for (20) is based on the Jacobian of $f(\alpha) = -Pb$ and given by

$$-\frac{\partial P}{\partial \alpha}b - P\frac{\partial b}{\partial \alpha}$$

where $\frac{\partial P}{\partial \alpha}b$ is defined to be the matrix with columns

$\frac{\partial P}{\partial \alpha_k} b$. In this expression

$$\begin{aligned}\frac{\partial P}{\partial \alpha_k} &= -\frac{\partial A}{\partial \alpha_k}(A^T A)^{-1} A^T \\ &+ A(A^T A)^{-1}\left(\frac{\partial A^T}{\partial \alpha_k} A + A^T \frac{\partial A}{\partial \alpha_k}\right)(A^T A)^{-1} A^T \\ &- A(A^T A)^{-1}\frac{\partial A^T}{\partial \alpha_k} \\ &= -P\frac{\partial A}{\partial \alpha_k}(A^T A)^{-1} A^T - A(A^T A)^{-1}\frac{\partial A^T}{\partial \alpha_k} P\end{aligned}$$

From this we conclude that the exact Jacobian is given by

$$P\frac{\partial A}{\partial \alpha}(A^T A)^{-1} A^T b + A(A^T A)^{-1}\frac{\partial A^T}{\partial \alpha} P b - P\frac{\partial b}{\partial \alpha}$$

The second term contains the factor $Pb = -f(\alpha)$, which is small for values of α close to optimum, and this is the term that is omitted in the approximation proposed by Kaufman, [Kau75]. In the original work this approximation was suggested for the Gauss-Newton-Marquardt algorithm. Here we have used it for a trust-region algorithm.

The detailed expression for our separable NLS problem follows from making the following identifications: $A = \gamma\Phi_m$, $b = -\gamma e_o$, $\alpha = \theta$ and $x = x_m$, where the latter is a misuse of notation. It follows that $P = I - \Phi_m Z^{-1}\Phi_m^T$ as before. Moreover

$$\begin{aligned}\frac{\partial A}{\partial \alpha_k} &= \frac{\partial \gamma}{\partial \theta_k}\Phi_m + \gamma\frac{\partial \Phi_m}{\partial \theta_k} \\ \frac{\partial b}{\partial \alpha_k} &= -\frac{\partial \gamma}{\partial \theta_k} e_o - \gamma\frac{\partial e_o}{\partial \theta_k}\end{aligned}$$

In addition to this we make use of that $-(A^T A)^{-1} b$ is the least squares solution of $\min_{x_m} \|\Phi_m x_m + e_o\|_2$. Then the Kaufman Jacobian can be efficiently computed from the above formulas as

$$-P\left(\frac{\partial A}{\partial \alpha} x_m + \frac{\partial b}{\partial \alpha}\right)$$