Affinely parametrized state-space models
Ways to maximize the Likelihood Function

Wills, Adrian; Yu, Chengpu; Ljung, Lennart; Verhaegen, Michel

DOI
10.1016/j.ifacol.2018.09.170

Publication date
2018

Document Version
Final published version

Published in
IFAC-PapersOnLine

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 51-15 (2018) 718–723

Affinely Parametrized State-space Models: Ways to Maximize the Likelihood Function

Adrian Wills, Chengpu Yu, Lennart Ljung, and Michel Verhaegen

Abstract: Using Maximum Likelihood (or Prediction Error) methods to identify linear state space model is a prime technique. The likelihood function is a nonconvex function and care must be exercised in the numerical maximization. Here the focus will be on affine parameterizations which allow some special techniques and algorithms. Three approaches to formulate and perform the maximization are described in this contribution: (1) The standard and well known Gauss-Newton iterative search, (2) a scheme based on the EM (expectation-maximization) technique, which becomes especially simple in the affine parameterization case, and (3) a new approach based on lifting the problem to a higher dimension in the parameter space and introducing rank constraints.

Keywords: Parameterized state-space model, maximum-likelihood estimation, expectation-maximization algorithm, difference-of-convex optimization.

1. INTRODUCTION

The identification of parametric state-space models using observed input and output data is a fundamental identification problem which has been intensively investigated in the last few decades Ljung [1999], Verhaegen and Verdult [2007]. This contribution deals with the case that the discrete time state space matrices are affine in the parameters but otherwise have an arbitrary structure. This corresponds to common and important applications, e.g. networks, compartment models, physically parameterized grey box models etc.

Among existing identification methods, prediction error methods (PEM) and maximum likelihood estimates (MLE) can handle such parameterizations, but require reasonable initial parameter estimates. Subspace methods, like MOESP, Verhaegen and Dewilde [1992], and N4SID, Overschee and Moor [1994], cannot handle models with arbitrary structure.

We shall therefore focus on MLE for affine parameterization. Due to the non-convex nature of the likelihood function, special care is required for how to approach its maximization. After a definition of the problem (Section 2) and a formulation of the likelihood function (and the negative log-likelihood function, the NLLF) V in Section 3, we turn to the question of how to minimize V.

The standard technique is applying iterative local, Gauss-Newton search. That is reviewed in Section 4.

It is also possible to to apply the general Expectation-Maximization (EM) technique, Dempster et al. [1977] for MLE to the state-space identification problem. This has been done for black-box parameterizations in Wills et al. [2010]. It is attractive to apply EM also to the case of affine, structured paramerizations, since the E-step becomes a linear regression. The details of this are given in Section 5.

A third technique is based on "lifting": Extend the ML problem by several more parameters to make the criterion much simpler (quadratic) and introduce a number of constraints that ensure that the model structure still is enforced. This leads to a formulation that resembles the sub-space approach. A similar technique was applied to the Hankel matrix of the impulse response obtained from a subspace, black box estimate in Yu et al. [2017] and Yu et al. [2018]. The approach applied to the input-output data is described in Section 6.

2. PROBLEM STATEMENT

We consider a parameterized discrete-time (DT) linear state-space model as follows

$$\begin{aligned} x(k+1) &= A(\theta)x(k) + B(\theta)u(k) \\ y(k) &= C(\theta)x(k) + D(\theta)u(k) + w(k), \end{aligned} \tag{1}$$

where $u(k) \in \mathbb{R}^m, x(k) \in \mathbb{R}^n, y(k) \in \mathbb{R}^p$ and $w(k) \in \mathbb{R}^p$ are system input, state, output, and measurement noise, respectively; $\theta \in \mathbb{R}^d$ is the parameter vector; k is the time index. We assume that the parameterized system matrices are affine with respect to $\theta = [\theta_1, \cdots, \theta_d]^T$, i.e.,

[1] A. Wills is with the School of Electrical Engineering and Computer Science, University of Newcastle, Australia (adrian.wills@newcastle.edu.au)
[2] C. Yu is with the school iof Automation, Beijing Institute of Technology (yuchengpu@bit.edu.cn)
[3] L. Ljung is with the Division of Automatic Control, Department of Electrical Engineering, Linkoping University, Sweden (ljung@isy.liu.se)
[4] M. Verhaegen is with the Delft Center for Systems and Control, Delft University, Delft 2628CD, Netherlands, (m.verhaegen@tudelft.nl)
[5] The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement no. 339681.

10.1016/j.ifacol.2018.09.170

$$A(\theta) = A_0 + \sum_{i=1}^{d} A_i\theta_i, \quad B(\theta) = B_0 + \sum_{i=1}^{d} B_i\theta_i,$$
$$C(\theta) = C_0 + \sum_{i=1}^{d} C_i\theta_i, \quad D(\theta) = D_0 + \sum_{i=1}^{d} D_i\theta_i \tag{2}$$

where the coefficient matrices A_i, B_i, C_i and D_i are known. Note that the above structured representation allows for describing dependencies between the entries of the matrices, and the known coefficient matrices can represent a general system basis which may have low-rank or sparse properties. This type of affine model parameterization is common for LPV (linear parameter varying) systems. With gray-box modeling in discrete time one typically also arrives at an affine parameterization like (2) with the discrete-time physical parameters contained in θ.

3. ML ESTIMATION METHOD

Suppose observed random data $\mathcal{Z}$ has the probability density function $p_{\mathcal{Z}}(z,\theta)$ that depends on an unknown parameter θ. Then the *likelihood function* (LF) for estimating θ from $\mathcal{Z}$ for an actual observation z of $\mathcal{Z}$ is

$$L(\theta, z) = p_{\mathcal{Z}}(z,\theta). \tag{3}$$

The θ that makes the actual observations as likely as possible,

$$\hat{\theta}^{ML} = \arg\max_{\theta} L(z,\theta) \tag{4}$$

is the *Maximum Likelihood Estimate (MLE)*. Since log is a an increasing function, it is customary to instead minimize the negative logarithm of $L(z,\theta)$ (Negative Log Likelihood Function, NLLF):

$$V(z,\theta) = -\log L(z,\theta). \tag{5}$$

To identify the parameter vector θ^* of the discrete-time model (1) using the IO data $z = \{u(k), y(k)\}_{k=0}^{N-1}$, the negative log-likelihood function (NLLF) $V(z,\theta)$ takes the form (see, e.g. Ljung [1999], Sect 7.4)

$$\begin{aligned} V(z,\theta,x(0)) = \;& \frac{1}{N}\sum_{k=0}^{N-1} \|y(k) - \hat{y}(k|\theta)\|^2 \\ s.t. \quad & x(k+1,\theta) = A(\theta)x(k,\theta) + B(\theta)u(k) \\ & \hat{y}(k|\theta) = C(\theta)x(k,\theta) + D(\theta)u(k) \\ & \text{for} \quad k = 0,\cdots,N-1. \end{aligned} \tag{6}$$

This expression assumes that the disturbances $w(k)$ in (1) are white and Gaussian with known covariance I. The expression has also been normalized and stripped from non-essential constants.

Apart from the parameter vector θ, the initial state $x(0)$ is also a variable to be estimated. The expression (6) is quadratic in $x(0)$ for given θ, so it is immediate to find the minimizer $\hat{x}(0,\theta)$ for each θ. Hence $x(0)$ can be directly eliminated from the problem (6).

The performance of the PEM/ML method mainly relies on the selection of the initial parameter estimate. It is shown in Ljung and Parrilo [2003] that the chances to reach the global minimum of (6) from random starting points may be very slim for problems of realistic sizes.

The NLLF is defined by (6). For any given value of θ and $x(0)$ it is straightforward to compute $V(z,\theta,x(0))$, and any other calculation method for the likelihood method must give the same result. The difficulty does not lie there. But the problem is that even though the model parameterization is simple (linear in θ according to (2)), the expression V becomes a complicated function (very high order polynomial) of θ. Note that from (6),

$$x(k,\theta) = A(\theta)^k x(0) + \sum_{j=1}^{k-1} A(\theta)^{k-j} B(\theta) u(j) \tag{7}$$

As a result, V may have an unsmooth surface, with several local extremal points. The minimization of V by local search may thus show difficulties.

We shall now proceed to look into several ways to formulate how to maximize the LF.

4. FINDING THE MLE BY LOCAL SEARCH

The gradient-based optimization algorithms such as Gauss-Newton method [Ljung, 1999, Section 10.2] and gradient projection method [Verhaegen and Verdult, 2007, Chapter 7] can be used to solve (6). In these methods, the crucial step is to compute the predicted output $\hat{y}(k,\theta)$ and its derivative $\frac{\partial \hat{y}(k,\theta)}{\partial \theta}$ at a given point $\theta = \hat{\theta}^i$. Given the parameter estimate $\hat{\theta}^i$, the predicted system output can be obtained by simulating the following system:

$$\begin{aligned} \hat{x}(k+1,\hat{\theta}^i) &= A(\hat{\theta}^i)\hat{x}(k,\hat{\theta}^i) + B(\hat{\theta}^i)u(k) \\ \hat{y}(k|\hat{\theta}^i) &= C(\hat{\theta}^i)\hat{x}(k,\hat{\theta}^i) + D(\hat{\theta}^i)u(k). \end{aligned} \tag{8}$$

Denote $X_j(k,\theta) = \partial\hat{x}(k,\theta)/\partial\theta_j$ for $j = 1,\cdots,l$ with θ_j being the j-th component of θ. Then, the derivative $\frac{\partial \hat{y}(k,\theta)}{\partial \theta_j}$ can be computed by simulating the following systems:

$$\begin{aligned} X_j(k+1,\hat{\theta}^i) &= A(\hat{\theta}^i)X_j(k,\hat{\theta}^i) + \left.\frac{\partial A(\theta)}{\partial\theta_j}\right|_{\theta=\hat{\theta}^i} \hat{x}(k,\hat{\theta}^i) \\ &\quad + \left.\frac{\partial B(\theta)}{\partial\theta_j}\right|_{\theta=\hat{\theta}^i} u(k) \\ \Psi^i(k) = \left.\frac{\partial \hat{y}(k|\theta)}{\partial\theta_j}\right|_{\theta=\hat{\theta}^i} &= C(\hat{\theta}^i)X_j(k,\hat{\theta}^i) + \left.\frac{\partial C(\theta)}{\partial\theta_j}\right|_{\theta=\hat{\theta}^i} \hat{x}(k,\hat{\theta}^i) \\ &\quad + \left.\frac{\partial D(\theta)}{\partial\theta_j}\right|_{\theta=\hat{\theta}^i} u(k), \end{aligned} \tag{9}$$

where $\hat{x}(k,\hat{\theta}^i)$ and $u(k)$ are system inputs, and the matrices $A(\hat{\theta}^i)$, $\left.\frac{\partial A(\theta)}{\partial\theta_j}\right|_{\theta=\hat{\theta}^i}$, $\left.\frac{\partial B(\theta)}{\partial\theta_j}\right|_{\theta=\hat{\theta}^i}$, $C(\hat{\theta}^i)$, $\left.\frac{\partial C(\theta)}{\partial\theta_j}\right|_{\theta=\hat{\theta}^i}$ are fixed at the point $\theta = \hat{\theta}^i$.

From the gradient ($d|p$-matrix) Ψ^i, and the current prediction error $\varepsilon^i(k) = y(k) - \hat{y}(k|\hat{\theta}^i)$ the gradient of the NLLF function V in (6) can be formed as

$$G^i = V\prime(\hat{\theta}^i) = \sum_{k=0}^{N-1} \Psi^i(k)\varepsilon^i(k) \tag{10}$$

as well as the approximation of the second derivative matrix (the Hessian) of V:

$$H^i = \sum_{k=0}^{N-1} \Psi^i(k)[\Psi^i(k)]^T \tag{11}$$

This gives the algorithm

Algorithm 1 Gauss-Newton algorithm for minimizing the NLLF

1) Choose an initial parameter vector $\hat{\theta}^0$. Set $i = 0$
2) From the current parameter estimate $\hat{\theta}^i$ compute the gradient and Hessian G^i and H^i in (10,11)
3) Do the update step $\theta^{i+1} = \theta^i + \mu[H^i]^{-1}G^i$ where μ is adjusted so that the criterion V is decreased
4) Stop if no improvement is achieved, otherwise return to step 2).

Several variants of this family of algorithms exist, essentially corresponding to different ways to form $[H^i]^{-1}G^i$.

The performance of the local search method relies on the selection of the initial parameter estimate. It is shown in Ljung and Parrilo [2003] that the chances to reach the global minimum of (6) from random starting points may be very slim for problems of realistic sizes.

5. FINDING THE MLE BY EXPECTATION-MAXIMIZATION METHOD

An alternative to direct gradient-based optimization of the ML objective outlined above is the Expectation-Maximisation (EM) algorithm. In a similar manner to gradient based methods, this approach also approximates the log-likelihood cost about a current parameter estimate $\hat{\theta}_i$, but different to gradient-based methods the local surrogate model is not formed by a Taylor series expansion.

Instead, the EM method creates a model of the likelihood by first creating a joint likelihood between the actual data and the so-called "missing data". The purpose of designing this joint likelihood is to render the problem more easily solvable if the missing data were actually available. Since it is not, the local model comes as a result of marginalising the missing data relative to its best estimate based on the actual measured data. This is called the Expectation step (E-step). Again, similar to gradient-based search this local model is then maximised as a surrogate for the actual log-likelihood. This is called the Maximisation step (M-step).

In situations where data is literally missing, either by error or measurement censoring, then the choice of missing data can be obvious Isaksson [1993], Goodwin and Feuer [1999].

For state-space systems, it is the "desired" state-sequence, as opposed to missing data, that would render the ML estimation more tractable if it were actually available. Indeed, within the automatic control community, the missing data is almost always chosen as the full state sequence Shumway [1982], Gibson and Ninness [2005], Gibson et al. [2005], Wills et al. [2009], Gopaluni [2008], Goodwin and Agüero [2005], Ghaharamani and Roweis [1999], Schön et al. [2011], Wills et al. [2013].

Unfortunately, for the current affine model structure with no state noise, this creates a difficulty since there is a deterministic relationship between states and measurements. Therefore, rather than choose the entire state sequence as missing, in what follows it is the initial state that is determined to be missing (or desired). The EM method then proceeds by alternating between estimating this missing state (E-step), and, maximising the joint log-likelihood over the system parameters (M-step).

More specifically, following similar arguments to those in Wills et al. [2010], we treat only the initial state $x(0)$ as the "missing data", since all the other states can determined exactly according the model (1). Under this assumption, we may define a "complete data" likelihood via

$$L(\theta, z, x(0)) = p_\theta(z, x(0)), \tag{12}$$

which is related to the likelihood $L(\theta, z)$ in (3), according to

$$p_\theta(z) = \frac{p_\theta(z, x(0))}{p_\theta(x(0) \mid z)}. \tag{13}$$

This allows the following expression for the log-likelihood

$$\log\, p_\theta(z) = \log p_\theta(z, x(0)) - \log p_\theta(x(0) \mid z) \tag{14}$$

Define $\hat{\theta}^i$ as the current estimate of $\hat{\theta}^{ML}$, then we may take the conditional expected value of (14) (the so-called E-step) to arrive at

$$\begin{aligned}\log\, p_\theta(z) = &\underbrace{\int \log p_\theta(z, x(0)) p_{\hat{\theta}^i}(x(0) \mid z) dx(0)}_{\triangleq \mathcal{Q}(\theta, \hat{\theta}^i)} \\ &- \underbrace{\int \log p_\theta(x(0) \mid z) p_{\hat{\theta}^i}(x(0) \mid z) dx(0)}_{\triangleq \mathcal{V}(\theta, \hat{\theta}^i)},\end{aligned} \tag{15}$$

Therefore, the difference between the log-likelihood at $\hat{\theta}^i$ and the log-likelihood at an arbitrary value of θ is given by (see e.g. Wills et al. [2010])

$$\begin{aligned}\log\, p_\theta(z) - \log\, p_{\hat{\theta}^i}(z) = &\left(\mathcal{Q}(\theta, \hat{\theta}^i) - \mathcal{Q}(\hat{\theta}^i, \hat{\theta}^i)\right) \\ &+ \left(\mathcal{V}(\hat{\theta}^i, \hat{\theta}^i) - \mathcal{V}(\theta, \hat{\theta}^i)\right).\end{aligned} \tag{16}$$

It has been established elsewhere (see e.g. Wills et al. [2010]) that

$$\mathcal{V}(\hat{\theta}^i, \hat{\theta}^i) - \mathcal{V}(\theta, \hat{\theta}^i) \geq 0. \tag{17}$$

As a result, if we can find $\hat{\theta}^{i+1}$ (the M-step) such that $\mathcal{Q}(\hat{\theta}^{i+1}, \hat{\theta}^i) > \mathcal{Q}(\hat{\theta}^i, \hat{\theta}^i)$, then necessarily via (16) and (17) $\log\, p_{\hat{\theta}^{i+1}}(z) > \log\, p_{\hat{\theta}^i}(z)$. This observation leads to the EM algorithm, which alternates between forming $\mathcal{Q}(\theta, \hat{\theta}^i)$ using $\hat{\theta}^i$ and then maximising $\mathcal{Q}(\theta, \hat{\theta}^i)$ with respect to θ to obtain a new better estimate $\hat{\theta}^{i+1}$.

With regard to the innovations form of the model structure (1), and with the choice of "missing data" $x(0)$, the function $\mathcal{Q}(\theta, \hat{\theta}^i)$ is given as (ignoring unimportant constants - see Wills et al. [2010])

$$\begin{aligned}&\mathcal{Q}(\theta, \hat{\theta}^i) = -\log\det P_0 - N \log\det R \\ &- \mathrm{Tr}\left\{P_0^{-1}\left((\hat{x}_{0|N} - \mu)(\hat{x}_{0|N} - \mu)^T + P_{0|N}\right)\right\} \\ &- \mathrm{Tr}\left\{R^{-1}\sum_{k=0}^{N-1} \varepsilon_k \varepsilon_k^T\right\} - \mathrm{Tr}\left\{R^{-1}\sum_{k=0}^{N-1} C P_k C^T\right\}\end{aligned} \tag{18}$$

where we have further assumed that

$$w(k) \sim \mathcal{N}(0\,,\, R), \qquad x(0) \sim \mathcal{N}(\mu\,,\, P_0), \tag{19}$$

and

$$\hat{x}_{0|N} \triangleq \mathrm{E}_{\hat{\theta}^i}\left\{x_0 \mid z\right\} \tag{20a}$$

$$P_{0|N} \triangleq \mathrm{Cov}_{\hat{\theta}^i}\left\{x_0 \mid z\right\} \tag{20b}$$

$$\varepsilon_k \triangleq y_k - \hat{y}_{k|k-1} \tag{20c}$$

$$\hat{y}_{k|k-1} = \mathrm{E}_{\hat{\theta}^i}\left\{y_k \mid z_{0:k-1}\right\} \tag{20d}$$

$$P_k \triangleq \mathrm{Cov}_{\hat{\theta}^i}\left\{x_k \mid z_{0:k-1}\right\} \tag{20e}$$

In the above, we have used the notation $z_{0:k-1}$ to denote all the data from time 0 until time $k-1$, i.e. $\{z_0, \ldots, z_{k-1}\}$.

The required terms (20a) and (20b) can be obtained by a Kalman smoother (see e.g. Chapter 10 in Kailath et al. [2000]). The terms (20c)–(20e) may be computed by employing standard Kalman Filter recursions (see e.g. Chapter 9 in Kailath et al. [2000]).

The above constitutes the E-step, where $\mathcal{Q}(\theta, \hat{\theta}^i)$ is computed using the current estimate $\hat{\theta}^i$. Considering the M-step, it is necessary to maximise $\mathcal{Q}(\theta, \hat{\theta}^i)$ over θ to deliver the next iterate $\hat{\theta}^{i+1}$. Towards this end, note that we may split the parameter vector θ in to two parts

$$\theta^T = [\eta^T, \beta^T]^T, \tag{21}$$

where η parameterizes $\{\mu, P_0, R\}$, and β parameterizes $\{A(\beta), B(\beta), C(\beta), D(\beta)\}$. In this paper we are concerned with the case where β parameterizes the associated system matrices according to a known structure, but we also assume that no such structural constraints are imposed on $\{\mu, P_0, R\}$, aside from requiring that P_0 and R are positive definite and symmetric.

With this separation of θ, we note that (18) can be maximised with respect to μ by

$$\mu = \hat{x}_{0|N}. \tag{22}$$

Further, by substituting this expression into (18), the terms involving P_0 become

$$-\log\det P_0 - \operatorname{Tr}\left\{P_0^{-1} P_{0|N}\right\}, \tag{23}$$

which is maximised by

$$P_0 = P_{0|N} \tag{24}$$

Again, by analogous argument

$$R = \frac{1}{N}\sum_{k=0}^{N-1} \varepsilon_k \varepsilon_k^T + C P_k C^T \tag{25}$$

is also a stationary point of (18). Substituting (22), (24) and (25) into (18) delivers a "reduced" form $\widetilde{\mathcal{Q}}(\beta, \hat{\theta}^i)$ that depends only on β as follows

$$\widetilde{\mathcal{Q}}(\beta, \hat{\theta}^i) = -\log\det\left(\frac{1}{N}\sum_{k=0}^{N-1} \varepsilon_k \varepsilon_k^T + C P_k C^T\right). \tag{26}$$

In general, it is not possible to maximise $\widetilde{\mathcal{Q}}(\beta, \hat{\theta}^i)$ in closed form. Therefore, we again employ a gradient-based search procedure (similar to that used above in Algorithm 1) in order to compute $\hat{\beta}^{i+1}$ that maximises $\widetilde{\mathcal{Q}}(\hat{\beta}^{i+1}, \hat{\theta}^i)$. In order to implement this, it is first necessary to develop an expression for the gradient of $\widetilde{\mathcal{Q}}(\beta, \hat{\theta}^i)$ with respect to β. To that end, an expression for this gradient can be straightforwardly derived by repeated application of the Chain rule to deliver

$$\begin{aligned}\frac{\partial \widetilde{\mathcal{Q}}(\beta)}{\partial \beta_i} = &-2\sum_{k=0}^{N-1} \varepsilon_k^T R(\beta)^{-1} \frac{\partial \varepsilon_k}{\partial \beta_i} \\ &- \sum_{k=0}^{N-1} \operatorname{Tr}\left\{R(\beta)^{-1}\frac{\partial C P_k C^T}{\partial \beta_i}\right\},\end{aligned} \tag{27}$$

where $R(\beta)$ is given by

$$R(\beta) \triangleq \frac{1}{N}\sum_{k=0}^{N-1} \varepsilon_k \varepsilon_k^T + C P_k C^T \tag{28}$$

and

$$\frac{\partial \varepsilon_k}{\partial \beta_i} = -\frac{\partial C}{\partial \beta_i}\hat{x}_k - C\frac{\partial \hat{x}_k}{\partial \beta_i} - \frac{\partial B}{\partial \beta_i}, \tag{29a}$$

$$\frac{\partial \hat{x}_{k+1}}{\partial \beta_i} = \frac{\partial A}{\partial \beta_i}\hat{x}_k + A\frac{\partial \hat{x}_k}{\partial \beta_i} + \frac{\partial B}{\partial \beta_i}u_k, \tag{29b}$$

$$\frac{\partial C P_k C^T}{\partial \beta_i} = \frac{\partial C}{\partial \beta_i}P_k C^T + C\frac{\partial P_k}{\partial \beta_i}C^T + C P_k \frac{\partial C^T}{\partial \beta_i}, \tag{29c}$$

$$\begin{aligned}\frac{\partial P_{k+1}}{\partial \beta_i} = &\frac{\partial A}{\partial \beta_i}P_k A^T + A\frac{\partial P_k}{\partial \beta_i}A^T + A P_k \frac{\partial A^T}{\partial \beta_i}, \\ \frac{\partial \hat{x}_0}{\partial \beta_i} = 0, \quad &\frac{\partial P_0}{\partial \beta_i} = 0.\end{aligned} \tag{29d}$$

Combining these E and M steps results in the following EM algorithm for identifying structured state-space models.

Algorithm 2 EM for structured state-space models

1) Choose an initial parameter estimate $\hat{\theta}^0$. Set $i = 0$
2) **Expectation (E) step:**
   Based on $\hat{\theta}^i$ and its associated A, B, C, D, R, μ, P_0 system parameters, run a Kalman smoother to obtain $\hat{x}_{0|N}$ and $P_{0|N}$.
3) **Maximisation (M) step:**
   Set $\mu \leftarrow \hat{x}_{0|N}$ and $P_0 \leftarrow P_{0|N}$.
   Use a gradient-based search algorithm to compute $\hat{\beta}^{i+1} = \arg\min_\beta \widetilde{\mathcal{Q}}(\beta, \hat{\theta}^i)$.
   Set $R \leftarrow R(\hat{\beta}^{i+1})$ and $\hat{\theta}^{i+1} = \{\mu, P_o, R, \hat{\beta}^{i+1}\}$.
4) If not converged, update $i \leftarrow i+1$ and return to step 2).

6. A LIFTING TECHNIQUE TO MINIMIZE THE NLLF

We will use vector and matrix notation for (6). Let

$$Y = [y(0), y(1), \ldots, y(N-1)] \quad (p|N \text{ matrix}) \tag{30a}$$

$$\mathbf{u} = [u(0), u(1), \ldots, u(N)] \quad (m|N+1 \text{ matrix}) \tag{30b}$$

$$\hat{\mathbf{x}}(\theta) = [x(0), x(1,\theta), \ldots, x(N-1,\theta)] \quad (n|N \text{ matrix}) \tag{30c}$$

Then the criterion V in (6) can be written

$$V(z, \theta, x(0)) = \|Y - C(\theta)\hat{\mathbf{x}}(\theta) - D(\theta)\mathbf{u}\|_F^2 \tag{31}$$

where F denotes the Frobenius norm.

A common way to handle the complex surfaces such as V is the expand the dimensionality of the problem to a simpler structure and then projecting back on the smaller dimension. In this case, we may expand the parameter dimension by introducing several new variables, so that the minimization criterion can be made simple, quadratic, in terms of the larger parameter vector. At the same time, the extra introduced variables must be constrained so that the system dynamics, in terms of (7) is preserved. It is desirable that these constraints are simple, linear or bilinear in terms of the new variables. This is a version of the *lifting technique*, frequently used in function minimization, e.g. Balas et al. [1993]. That is what we now set out to do:

Introduce two new $n|N+1$ matrices:

$$\mathbf{x} = [x(0), x_1, \ldots, x_N] \tag{32a}$$

$$\bar{\mathbf{x}} = [\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_{N+1}] \tag{32b}$$

and subject them to the following linear and bilinear constraints

$$\bar{\mathbf{x}}(:, 0:N-1) = \mathbf{x}(:, 1:N) \tag{33a}$$

$$\bar{\mathbf{x}} = A(\theta)\mathbf{x} + B(\theta)\mathbf{u} \tag{33b}$$

It is easy to see that these two constraints force

$$\mathbf{x} = \hat{\mathbf{x}}(\theta) \tag{34}$$

defined in (30).

Furthermore, introduce two new variables $O(p|n$ matrix) and $M(p|m$ matrix), and finally $\Psi(p|N$ matrix), together with the bilinear and linear constraints:

$$\Psi = O\mathbf{x} \tag{35a}$$
$$O = C(\theta) \tag{35b}$$
$$M = D(\theta) \tag{35c}$$

Now collect the new extended parameters

$$\Theta = \{\theta, M, O, \Psi, \mathbf{x}, \bar{\mathbf{x}}\} \tag{36}$$

and consider the quadratic minimization problem

$$\min_{\Theta} \|Y - \Psi - M\mathbf{u}\|_F^2 \tag{37}$$

subject to the linear and bilinear constraints on Θ (33a) and (35).

We see that the constraints force $\Psi = C(\theta)\hat{\mathbf{x}}(\theta)$ and $M = D(\theta)$.

Consequently, the minimization of the likelihood function (31) = (6) is the same as solving the quadratic minimization problem (37) with the indicated linear and bilinear constraints.

These constraints can be summarized as

$$\text{rank} Z = n \tag{38}$$

where

$$Z = \begin{bmatrix} \Psi & O \\ \mathbf{x} & I_n \\ \bar{\mathbf{x}} - B(\theta)\mathbf{u} & A(\theta) \end{bmatrix} \tag{39a}$$
$$\bar{\mathbf{x}}(:, 0:N-1) = \mathbf{x}(:, 1:N) \tag{39b}$$
$$O = C(\theta) \tag{39c}$$
$$M = D(\theta) \tag{39d}$$

To deal with the rank constraint, let $f_n(Z)$, for any real matrix Z, be the sum of the largest n singular values of Z:

$$f_n(Z) = \sum_{i=1}^{n} \sigma_i(Z). \tag{40}$$

Note that $f_n(\cdot)$ is a Ky Fan n-norm Bhatia [2013]. Then the rank constraint can be written

$$\|Z\|_* - f_n(Z) = 0, \tag{41}$$

This constraint can be rewritten in "linearized version" for an iterative solution scheme by utilizing the SVD of Z at the previous iterate j, Z^j:

$$\|Z\|_* - \text{tr}\left(U_1^{j,T} Z V_1^j\right) = 0, \tag{42}$$

with U_1 and V_1 as the left and right SVD matrices for the n largest singular values of Z^j. Treating (42) as a constraint in en epigraph form we the minimize

$$\min_{\Theta, t} \|Y - \Psi - M\mathbf{u}\|_F^2 + t \tag{43a}$$
$$\text{subject to } \|Z\|_* - \text{tr}\left(U_1^{j,T} Z V_1^j\right) \leq t \tag{43b}$$

Note that, due to the affine parametrization, Z is linear in Θ. It is known that the nuclear norm is convex in the matrix elements, so $\|Z\|_*$ is convex in Θ. The criterion (43a) is quadratic in Θ, t, so for given U_1, V_1, the constrained minimization (43) is a convex problem.

The algorithm for minimizing the NLLF V will now be given by

Algorithm 3 Sequential convex programming method for minimizing the NLLF

1) Set $U_1^0 = 0$ and $V_1^0 = 0$.
2) Repeat
   2-1): Obtain the estimates Z^{j+1} and θ by solving (43), (39).
   2-2): Compute the matrices U_1^{j+1} and V_1^{j+1}, which are the left and right singular vectors of Z^{j+1} corresponding to the n largest singular values.
3) until $\frac{\|\theta^{j+1} - \theta^j\|_2}{\|\theta^j\|_2} \leq \varepsilon$ with ε a small value.

7. CONCLUSIONS

Three ways to compute the Maximum Likehood Estimate have been treated. In addition to the standard and well-known Gauss-Newton family of algorithms (Algorithm 1) that can be applied to any parameterization, we have described what an EM algorithm looks like in the case of affine parameterizations, (Algorithm 2) as well as a subspace inspired algorithm based on lifting (Algorithm 3). The two latter algoritms are specifically taylored to the importnat special case of affine model parameterizations.

In all three cases the minimization task has been converted to a sequence of convex minimization problems, from different starting points: The GN approach is based on local, quadratic approximation of the NLLF at the current estimate so it solves a sequence of quadratic problems. The EM approach also utilizes this approximation in the M step. Algorithm 3 treats the non-convex rank constraint (38) by sequential linearization in (43b)

As all iterative algorithms, the treated numerical solutions all need an initial parameter estimate $\hat{\theta}^0$. In the absense of physical or other insights this has to done by a random choice. (Step 1 in algorithms 1 and 2). This can lead to bad convergence properties, as pointed out several times in the contributions. Algorithm 3 offers a simple and non-random initialisation. The latter algorithm has interesting potential, but at present it is challenging to obtain an effective implementation.

REFERENCES

E. Balas, S. Ceria, and G. Cornuejols. A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Math. Programming*, 58:295–323, 1993.

Rajendra Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithms. *J. Royal Statistical Society, ser. B*, 39(1):1–38, 1977.

Z. Ghaharamani and S. T. Roweis. Learning nonlinear dynamical systems using an EM algorithm. In *Advances in Neural Information Processing Systems*, volume 11, pages 599–605. MIT Press, 1999.

S.H. Gibson and B. Ninness. Robust maximum-likelihood estimation of multivariable dynamic systems. *Automatica*, 41(10):1667–1682, 2005.

Stuart Gibson, Adrian Wills, and Brett Ninness. Maximum-likelihood parameter estimation of bilinear systems. *IEEE Trans. Automat. Control*, 50(10):1581–1596, 2005. ISSN 0018-9286.

G. C. Goodwin and J. C. Agüero. Approximate EM algorithms for parameter and state estimation in nonlinear stochastic models. In *Proceedings of the 44th*

IEEE conference on decision and control (CDC) and the European Control Conference (ECC), pages 368–373, Seville, Spain, December 2005.

G.C. Goodwin and A. Feuer. Estimation with missing data. *Mathematical and Computer Modelling of Dynamical Systems*, 5(3):220–244, 1999.

R. B. Gopaluni. A particle filter approach to identification of nonlinear processes under missing observations. *The Canadian Journal of Chemical Engineering*, 86(6):1081–1092, December 2008.

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

Thomas Kailath, Ali H. Sayed, and Babak Hassibi. *Linear Estimation*. Prentice Hall, 2000.

L. Ljung. *System Identification: Theory for the User*. Pearson Education, 1999. ISBN 9780132440530. URL `https://books.google.nl/books?id=fYSrk4wDKPsC`.

Lennart Ljung and Pablo Parrilo. Initialization of physical parameter estimates. In *Proceedings of IFAC Symposium on System Identification*, 2003.

P. Van Overschee and B. De Moor. N4SID: subspace algorithms for the identification of combined deterministic-stochastic systems . *Automatica, (Special Issue)*, 30(1): 75–93, 1994.

T.B. Schön, A.G. Wills, and B. Ninness. System identification of nonlinear state-space models. *Automatica*, 37 (1):39–49, jan 2011.

R.H. Shumway. An approach to time series smoothing and forecasting using the EM algorithm. *Journal of Time Series Analysis*, 3(4):253–264, 1982.

M. Verhaegen and P. Dewilde. The output-error state-space model identification class of algorithms. *Int Journal of Control*, 56(5):1187–1210, 1992.

M. Verhaegen and V. Verdult. *Filtering and System Identification: A Least Squares Approach*. Cambridge University Press, 2007. ISBN 9781139465021. URL `https://books.google.nl/books?id=6Ne76uYOlVwC`.

Adrian Wills, Thomas B Schön, and Brett Ninness. Estimating state-space models in innovations form using the expectation maximisation algorithm. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 5524–5529. IEEE, 2010.

Adrian Wills, Thomas B. Schön, Lennart Ljung, and Brett Ninness. Identification of hammerstein-wiener models. *Automatica*, 49(1):70–81, 2013.

A.G. Wills, B. Ninness, and S.H. Gibson. Maximum likelihood estimation of state space models from frequency domain data. *IEEE Transactions on Automatic Control*, 54(1):19–33, 2009.

Chengpu Yu, L. Ljung, and M. Verhaegen. Gray box identification using difference of convex programming. In *Proc. IFAC World Congress*, Toulouse, France, 2017. Elsevier.

Chengpu Yu, L. Ljung, and M. Verhaegen. Identification of structured state-space models. *Automatica*, 2018.