A Factor Graph Approach to Parameter Identification for Affine LPV Systems via Expectation Maximization

C. Hoffmann, *Member, IEEE*, A. Isler, and P. Rostalski, *Member, IEEE*

Abstract— **In this paper, a factor graph representation of the expectation maximization (EM)-based estimation technique is employed and developed into a method for affine linear parameter-varying (LPV) system identification. Factor graphs are a graphical model for representing factorizable functions. As a probabilistic model, they allow to model conditional independence statements in probabilistic graphical networks and thus exploit the structure inherent to a system for solving inference problems. A combination of the Kalman Filter/Smoother and EM in the factor graph representation is known to yield parameter estimation techniques that rely on standard message passing algorithms for linear systems. It is shown how the factor graph can be extended, in order to introduce certain parameter-varying matrices that allow EM-based system identification approaches for LPV systems. The approach is tested on a bilinear system and shows accurate estimation results despite completely random initializations.**

I. Introduction

Factor graphs are a graphical model for factorizable functions. In conjunction with a sum-product algorithm, they are used in research related to fields such as signal processing, machine learning and statistics, see [8] for further references. Their potential to form a unified language for representing algorithms stems from the fact that many algorithms in these fields can be interpreted as special cases of the sum-product and max-product algorithms—a.k.a. belief propagation—using message passing on factor graphs. In this paper Forney-style factor graphs (FFGs) and the notation of [8] are adopted.

The modularity of the factor graph approach promises to develop new algorithms by a "mix-and-match" of algorithms already understood and represented as factor graphs. The purpose of this paper is therefore to contribute to the growing literature of factor graph-based graphical interpretations of techniques by devising an EM-based system parameter identification algorithm for nonlinear systems in linear fractional transformation (LFT)-LPV form. To this extent, the present paper is inspired by the maximum-likelihood approach to parameter estimation of bilinear/LPV systems proposed in [5], [15]. The present paper showcases the possibilities and further justifies the claim that factor graphs are a tool well suited for intuitive mix-and-match-style combinations of algorithms [8] and models for the development of new algorithms in a unified modeling language.

LPV systems [16] are constructed from linear systems which depend on time-varying parameters. They are capable of representing many nonlinear and time-varying systems via the notion of quasi-LPV systems in which the parameters are functions of states, inputs and/or outputs. The formulation of LPV systems via linear fractional representations (LFRs) involves the separation of constant state space system matrices and a parameter-varying matrix in a feedback structure. This framework can model rational parameter-dependency and powerful synthesis frameworks exist, e. g., [12], [17], that result in LPV controllers guaranteeing stability and some degree of performance, often in terms of the $\mathcal{L}_2$-gain of the closed-loop system.

In the proposed approach, LFT-LPV models in state-space representation are estimated directly, which avoids the introduction of so-called "dynamic dependence" [14] in the scheduling variables that occurs when input/output models are converted into state-space. In this paper, a parameter-dependency is assumed and including the identification of basis scheduling functions within a factor graph framework is subject to future research. Advantages of EM-based LPV identification lie in a linear scaling of computational complexity with the number of data points, combined with the relatively strong robustness against far-away initializations of the EM algorithm [15]. For an overview about LPV identification techniques, consult [14].

Outline: Section II provides preliminaries on notation and EM via factor graphs. Section III proposes an extended factorization in the factor graph for EM-based identification to solve affine LPV parameter estimation problems. Section IV presents selected estimation results, while Section V continues with an outlook and conclusion.

II. Preliminaries

Notation: Following the notation of [1], sum-product messages in FFGs are denoted $\overrightarrow{\mu}$ and $\overleftarrow{\mu}$ for messages in and against the direction of the edge, respectively. Capital letters are used for unknown variables and lower case letters for particular values.

Parameterizations of random Gaussian distributed variables $D_k \in \mathbb{R}^{n_d}$ are denoted $\mathcal{N}(\mathrm{m}_D, \mathrm{V}_D)$, meaning their probability density functions $\mathcal{N}_D \triangleq \mathcal{N}(\mathrm{m}_D, \mathrm{V}_D)$ are pa-

C. Hoffmann and P. Rostalski are with the Institute of Electrical Engineering in Medicine, Universität zu Lübeck, Moislinger Allee 53–55, 23558, Lübeck, e-mail: christian.hoffmann@uni-luebeck.de, philipp.rostalski@uni-luebeck.de. A. Isler is currently with ABB Switzerland Ltd, Austrasse, 5300 Turgi, Switzerland, e-mail: andreas.isler@ch.abb.com. During the time, parts of the results were conceived, A. Isler was with ETH Zurich/UC Berkeley.

rameterized as

$$\mathcal{N}\left(\mathrm{m}_D, \mathrm{V}_D\right) \sim \frac{\left|\mathrm{V}_D\right|^{-\frac{1}{2}}}{\sqrt{(2\pi)^{n_d}}} e^{-\frac{1}{2}(d-\mathrm{m}_D)^\top \mathrm{V}_D^{-1}(d-\mathrm{m}_D)}. \tag{1}$$

Parameters m_D and V_D are the mean and variance. An alternative parameterization $\mathcal{N}\left(\mathrm{W}_D \mathrm{m}_D, \mathrm{W}_D\right)$ can be formulated in terms of information matrix $\mathrm{W}_D = \mathrm{V}_D^{-1}$ and weighted mean $\xi_D = \mathrm{W}_D \mathrm{m}_D$. Corresponding Gaussian messages are consequently denoted $\overrightarrow{\mathrm{m}}_D$, $\overrightarrow{\mathrm{V}}_D$, etc., as Gaussian message passing can be performed in terms of the distribution parameters, see [7], [8] for details. For convenience of the reader, tables for basic Gaussian message passing from [7] are replicated in the Appendix.

The discrete time instant is denoted by k and is often used as an index. For LPV parameter sets the values of a parameter vector δ_k are assumed to be confined to a compact set $\boldsymbol{\delta}$ $\forall k$, denoted in boldface type. Similarly, LFT-based LPV representations [3] involve feedback over a parameter-dependent matrix $\Delta(\delta_k) = \Delta_k \in \boldsymbol{\Delta}$.

A. Factor Graphs

Fig. 1 shows a simple exemplary FFG, in which nodes denote factors and edges denote variables, cf. [8]. The graph shown is a graphical model of the factorization

$$f(x_1, x_2, x_3) = f_2(x_1) f_2(x_2) f_3(x_1, x_2, x_3)$$

Forney-style factor graphs obey the following rules

1) There is a unique node for every factor.
2) There is a unique edge or half-edge for every variable.
3) Node (f) and edge (x) are connected if and only if f is a function of x.

Reusing variables is realized through equality nodes. E.g., if in Fig. 1 $x \overset{\triangle}{=} x_1 = x_2 = x_3$, $f_3(x_1, x_2, x_3) = \delta(x_1 - x_2)\delta(x_1 - x_3)$, where $\delta(\cdot)$ is the Kronecker delta function.

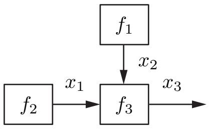


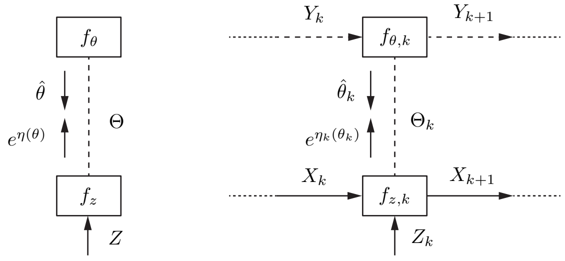
Fig. 1: A simple exemplary FFG.

If functions f_i, $i \in \{1, 2, \ldots, N\}$ of a factor graph (FG) are probability density functions and the edges correspond to random variables X_i, $i \in \{1, 2, \ldots, N\}$, in a cycle-free factor graph, $f_i(x_i)$ can be marginalized by multiplying the messages of the adjacent edge corresponding to X_i in both forward and backward direction (e. g., the summaries of the respective other parts of the graph). This is denoted the sum-product algorithm [8]. The max-product algorithm, on the other hand, can be used to find [8]

$$\hat{f}_l(x_l) \overset{\triangle}{=} \max_{x_i, \forall i \in \{1,2,\ldots,N\} \setminus \{l\}} f(x_1, \ldots, x_N).$$

B. Expectation Maximization via Factor Graphs

Expectation maximization is a particular iterative approach to solve nonlinear regression problems, in order to determine the set of parameters that most likely fit the data [2]. In [1], it has been shown how the EM algorithm for linear system identification can be regarded as message passing on factor graphs.



(a) Messages for EM with factor graphs.

(b) Segmented messages for EM with factor graphs.

Fig. 2: Messages for EM with factor graphs [8].

An FG interpretation of the EM algorithm amounts to defining new message computation rules that interface well with the sum-product or max-product algorithms performed via message passing on other parts of the factor graph. Consider Fig. 2a where $f(z|\theta) = f_z(z|\theta) f_\theta(\theta)$ is a factorized probability density function (PDF) of the observable variables z given the parameters θ. The factors f_z and f_θ may be factorizable (cycle-free) as well

$$\begin{aligned} f_z(z|\theta) &= \prod_k f_{z,k}(z_k, x_k, x_{k+1}|\theta_k), \\ f_\theta(\theta) &= \prod_k f_{\theta,k}(y_k, y_{k+1}|\theta_k), \end{aligned}$$

and may, e. g., correspond to potentially nontrivial factor graphs as in Fig. 2b. With $f(z|\theta) \geqslant 0, \forall (z, \theta)$, the EM algorithm aims to find the maximum-likelihood estimate

$$\hat{\theta} \overset{\triangle}{=} \underset{\theta}{\operatorname{argmax}} f(z|\theta) = \underset{\theta}{\operatorname{argmax}} \ln\left(f(z|\theta)\right). \tag{2}$$

With the hidden variables x, assuming that $f(z|\theta)$ is the marginal yields

$$\hat{\theta} \overset{\triangle}{=} \underset{\theta}{\operatorname{argmax}} \ln\left(\int_{\mathcal{D}_x} f(x, z|\theta)\, \mathrm{d}x\right). \tag{3}$$

The EM algorithm iteratively computes the distribution $p_X(x|z, \hat{\theta}^{(j)})$ of the hidden variables x for fixed current estimate $\hat{\theta}^{(j)}$ and then maximizes the expected value of $f(x, z|\theta)$ over θ assuming $p_X(x|z, \hat{\theta}^{(j)})$ is fixed to obtain $\hat{\theta}^{(j+1)}$.

The two steps of the EM algorithm, between which it is iterated are

$$\begin{aligned} \text{E-Step:} \quad & Q\left(\theta|\hat{\theta}^{(j)}\right) = \mathrm{E}_{p_X}\left[\ln\left(f(X, Z|\theta)\right)\right], \\ \text{M-Step:} \quad & \theta^{(j+1)} = \underset{\theta}{\operatorname{argmax}}\, Q\left(\theta|\hat{\theta}^{(j)}\right). \end{aligned} \tag{4}$$

The term $Q\left(\theta|\hat{\theta}^{(j)}\right)$ constitutes the part of the lower bound to $\ln\left(f(z|\theta)\right)$, denoted $q\left(\theta|\hat{\theta}^{(j)}\right)$, that depends on θ. Therefore maximizing $Q\left(\theta|\hat{\theta}^{(j)}\right)$ maximizes $\ln\left(f(z|\theta)\right)$. Details can be found in [2].

An FG such as shown in Fig. 2b allows for local computations not only w. r. t. f_x and f_y, but also w. r. t. their individual factors. This is explained in more detail in the following.

1) E-Step: Based on a factor graph representation of $f(x, z|\theta)$, the factorization $f(x, z|\theta) = \prod_k f_k(x_k, z|\theta)$, where x_k is a subset of x, can be used to perform the E-Step locally

$$\begin{aligned} Q\left(\theta|\hat{\theta}^{(j)}\right) &= \sum_k \mathrm{E}_{p_X}\left[\ln\left(f_{z,k}(X_k, X_{k+1}, Z_k|\theta_k)\right)\right], \\ &= \sum_k \eta_k(\theta_k). \end{aligned} \tag{5}$$

Applying the exponential function converts the sum in (5) into a product and allows to apply message passing for the calculation of $Q\left(\theta|\hat{\theta}^{(j)}\right)$:

$$e^{Q\left(\theta|\hat{\theta}^{(j)}\right)} = \prod_k e^{\eta_k(\theta_k)}.$$

The $e^{\eta_k(\theta_k)}$ are the "upward" messages, while the "downward" messages are the current estimates $\hat{\theta}_k^{(j)}$ [1].

When parameterizing $e^{\eta_k(\theta_k)}$ as a distribution, for which message passing rules can be formulated, calculating the exponent $Q\left(\theta|\hat{\theta}^{(j)}\right)$ is performed by a forward and backward pass over all k and a marginalization of the resulting forward and backward messages per edge.

2) M-Step: As the function to be maximized is a quadratic sum of squares, raising it to an exponent of the e function renders it a function parameterized like a Gaussian distribution. This parameterization of $Q\left(\theta|\theta^{(j)}\right)$ strongly simplifies the maximization step. In fact, the maximization reduces to the computation of this mode—in case of a Gaussian distribution the mode equals the mean m_θ.

3) Backward EM-Message With Gaussian Message Passing Rules on Multiplier Nodes: Fig. 3 illustrates an isolated factor graph with a EM-message on a multiplier node. The backward messages are calculated in [1]:

$$\overleftarrow{\mathrm{W}}_\theta = \mathrm{W}_I \otimes \left(\mathrm{V}_X + \mathrm{m}_X \mathrm{m}_X^\top\right), \tag{6}$$

$$\overleftarrow{\mathrm{W}}_\theta \overleftarrow{\mathrm{m}}_\theta = \left(\mathrm{W}_I \otimes I_{n_x}\right) \mathrm{cvect}\left(\mathrm{V}_{XY^\top} + \mathrm{m}_X \mathrm{m}_Y^\top - \mathrm{m}_X \mathrm{m}_I^\top\right), \tag{7}$$

In a discrete time segmented factor graph, as shown in Fig. 2b, these backward messages can then be used to perform message passing over the time instants of the part of the factor graph drawn with dashed edges. In a standard factor graph-based interpretation of the EM algorithm, the nodes $f_{\theta,k}$ would be equality nodes to reflect that the parameter is assumed constant, which also allows simple maximization by max-product message passing [1]. However, this graphical approach to the interpretation of such algorithms reveals options to further introduce non-trivial factors $f_{\theta,k}$, e. g., in order to reflect, e. g., linear dynamics in the parameters. An example of a factor graph for EM-based identification of the system matrix of a linear system

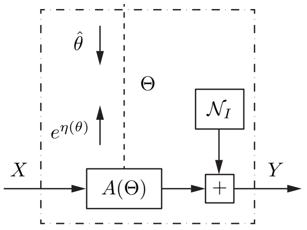


Fig. 3: Factor graph for EM-based identification of matrix in multiplier node [1].

$$G : \begin{bmatrix} x_{k+1} \\ \hdashline y_k \end{bmatrix} = \left[\begin{array}{c:c} A & B_u \\ \hdashline C_y & 0 \end{array}\right] \begin{bmatrix} x_k \\ \hdashline u_k \end{bmatrix}, \tag{8}$$

from [1] is shown in Fig. 4.

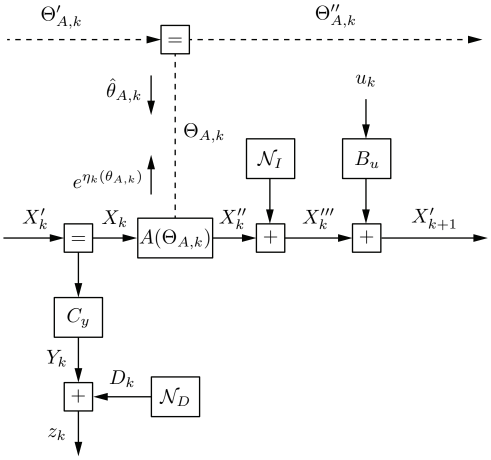


Fig. 4: Factor graph for EM-based identification of the system matrix [1].

III. Expectation Maximization-Based Identification

A. Expectation Maximization-Based Identification of Affine Pure LPV Systems

Consider the discrete-time, LPV system with affine parameter-dependence

$$\mathcal{G}_\Delta : \begin{cases} \begin{bmatrix} x_{k+1} \\ \hdashline y_k \end{bmatrix} = \left[\begin{array}{c:c} \mathcal{A}\left(\Delta_k\right) & \mathcal{B}_u\left(\Delta_k\right) \\ \hdashline \mathcal{C}_y\left(\Delta_k\right) & \mathcal{D}_{yu}\left(\Delta_k\right) \end{array}\right] \begin{bmatrix} x_k \\ \hdashline u_k \end{bmatrix} \\ \qquad \Delta_k \in \boldsymbol{\Delta}, \end{cases} \tag{9}$$

with $x_k \in \mathbb{R}^{n_x}$, $y_k \in \mathbb{R}^{n_y}$, $u_k \in \mathbb{R}^{n_u}$ and where the time-varying matrix Δ_k is confined to a compact set $\boldsymbol{\Delta}$ and may be a function of known states, inputs and outputs. For simplicity, assume that only the system matrix of $\mathcal{G}_\Delta$ admits an affine parameter-dependence on Δ_k. Then an LFR can be written as

$$\mathcal{G}_\Delta : \begin{cases} \left[\begin{array}{c} x_{k+1} \\ \hdashline p_k \\ y_k \end{array}\right] = \left[\begin{array}{c:cc} A & B_\Delta & B_u \\ \hdashline C_\Delta & 0 & 0 \\ C_y & 0 & 0 \end{array}\right] \left[\begin{array}{c} x_k \\ \hdashline q_k \\ u_k \end{array}\right] \\ \qquad q_k = \Delta_k p_k, \qquad \Delta_k \in \mathbf{\Delta}, \end{cases} \tag{10}$$

which means that

$$\mathcal{A}_k = \mathcal{A}\left(\Delta_k\right) = A + B_\Delta \Delta_k C_\Delta = \begin{bmatrix} A & B_\Delta \end{bmatrix} \begin{bmatrix} I_{n_x} \\ \Delta_k C_\Delta \end{bmatrix}.$$

Using the fact that for any real-valued matrices M, N [9]

$$\text{cvect}(MN) = (N^\top \otimes I)\text{cvect}(M), \tag{11}$$

the unknown parameter vector $\theta_A = \text{cvect}\big(\begin{bmatrix} A \; B_\Delta \end{bmatrix}\big)$ enters the vectorized system matrix as follows

$$\begin{aligned} \theta_{\mathcal{A}_k} &= \text{cvect}(\mathcal{A}\left(\Delta_k\right)) && (12) \\ &= \left[\begin{bmatrix} I_{n_x} \\ \Delta_k C_\Delta \end{bmatrix}^\top \otimes I_{n_x}\right] \theta_A = \tilde{\Delta}_k \theta_A. && (13) \end{aligned}$$

This factorization allows to separate the EM-message into one that is related to the LFT coefficient matrices and one that is associated with the time-varying system matrix $\mathcal{A}(\Delta_k)$ by means of a regular multiplication node shown in Fig. 5a. The multiplication is defined with respect to the matrix $\tilde{\Delta}_k$ that is assumed to be a time-varying function of exogeneous parameters, inputs or outputs. Consequently, the upward messages follow the message passing rules

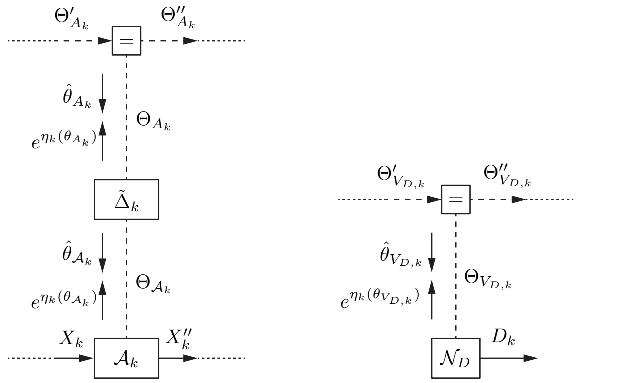
$$\begin{aligned} \overset{\leftarrow}{\mathrm{W}}_{\theta_{A_k}} &= \tilde{\Delta}_k^\top \overset{\leftarrow}{\mathrm{W}}_{\theta_{\mathcal{A}_k}} \tilde{\Delta}_k, \\ \overset{\leftarrow}{\mathrm{W}}_{\theta_{A_k}} \overset{\leftarrow}{\mathrm{m}}_{\theta_{A_k}} &= \tilde{\Delta}_k^\top \overset{\leftarrow}{\mathrm{W}}_{\theta_{\mathcal{A}_k}} \overset{\leftarrow}{\mathrm{m}}_{\theta_{\mathcal{A}_k}}, \end{aligned}$$

while $\overset{\leftarrow}{\mathrm{W}}_{\theta_{\mathcal{A}_k}}$ and $\overset{\leftarrow}{\mathrm{W}}_{\theta_{\mathcal{A}_k}} \overset{\leftarrow}{\mathrm{m}}_{\theta_{\mathcal{A}_k}}$ follow from (6) and (7), respectively. Similarly, additional EM-messages can be introduced for the identification of $\mathcal{C}(\Delta_k)$. Details are omitted for reasons of space.

A dependence of Δ_k on state variables in the vein of quasi-LPV systems [11] would introduce further cycles to the factor graph. However, as exact knowledge of LPV parameters is often assumed in LPV control and system identification—with notable exceptions such as, e. g., [10]—, this does not pose a strong restriction on the proposed algorithm. Instead, it appears as if the factor graph representation reveals a potential future direction of research for quasi-LPV system identification with uncertain scheduling parameters, in that such cycles can be introduced and tackled by appropriate message passing schedules followed by rigorous convergence analysis.

B. Expectation Maximization-Based Gaussian Variance Matrix Identification

In the above EM-based identification scheme, noise variances have to be assumed. If the variance are unknown, it can also be identified using EM. The concept is identical, in that an appropriate EM-message needs to derived. This derivation can be extended from the scalar case considered in [6] or from the diagonal variance case presented in [7] as follows.



(a) Part of factor graph for EM-based identification of an affinely dependent system matrix.

(b) Part of factor graph for EM-based identification of the variance of a Gaussian distribution.

Fig. 5: Messages for EM-based identification of system matrices and noise variances.

With the Gaussian node function $f_k(D_k) \sim \mathcal{N}\left(\mathrm{m}_D, \mathrm{V}_D(\theta_{\mathrm{V}_D})\right)$ and $\mathrm{V}_D(\theta_{\mathrm{V}_D}) = \theta_{\mathrm{V}_D}$ the EM-message $e^{\eta_k(\theta_{\mathrm{V}_D})}$, cf. Fig. 5b, is defined using the distribution function (1) via

$$\begin{aligned} \eta_k(\theta_{\mathrm{V}_D}) &\overset{\triangle}{=} \mathrm{E}\left[\ln\left(f_k(D_k)|\, z, \hat{\theta}_D^{(j)}\right)\right] && (14) \\ &= \text{const.} + \ln\left(|\theta_{\mathrm{V}_D}|^{-\frac{1}{2}}\right) \\ &\quad - \frac{1}{2}\mathrm{E}\left[\mathrm{tr}\left((D_k - \mathrm{m}_D)^\top \theta_{\mathrm{V}_D}^{-1}(D_k - \mathrm{m}_D)\right)\right]. \end{aligned}$$

With further manipulations, it becomes apparent that (14) can be parameterized as an Inverse-Wishart distribution $\mathcal{W}^{-1}(\Xi_{\theta_{\mathrm{V}_D}}, \varrho_{\theta_{\mathrm{V}_D}})$ message with

$$\begin{aligned} \eta_k(\theta_{\mathrm{V}_D}) &\propto \ln\left(|\theta_{\mathrm{V}_D}|^{-\frac{1}{2}(\varrho_{\theta_{\mathrm{V}_D}} + n_d + 1)}\right) - \frac{1}{2}\mathrm{tr}\left(\Xi_{\theta_{\mathrm{V}_D}} \theta_{\mathrm{V}_D}^{-1}\right), \\ \Xi_{\theta_{\mathrm{V}_D}} &= \mathrm{V}_{D_k} + (\mathrm{m}_{D_k} - \mathrm{m}_D)(\mathrm{m}_{D_k} - \mathrm{m}_D)^\top, \\ \varrho_{\theta_{\mathrm{V}_D}} &= -n_D. \end{aligned}$$

In the M-step, the maximization is performed by calculating the mode value

$$\begin{aligned} \mathrm{V}_D\left(\theta_{\mathrm{V}_D}^{(j+1)}\right) = \theta_{\mathrm{V}_D}^{(j+1)} &= \text{mode}\Big(\mathcal{W}^{-1}(\Xi_{\theta_{\mathrm{V}_D}}, \varrho_{\theta_{\mathrm{V}_D}})\Big) \\ &= \frac{\Xi_{\theta_{\mathrm{V}_D}}}{\varrho_{\theta_{\mathrm{V}_D}} + n_D + 1} \end{aligned}$$

C. Implementation Schedule

EM is an iterative algorithm, which—in a factor graph-based interpretation—introduces cycles via the EM edges. These cycles can be dealt with by choosing an appropriate message update schedule. In the case of the parameter identification framework of this paper, this amounts to first computing the a posteriori probability distributions by performing updates and marginalizations on all Kalman Filter/Smoother messages for all k (I.), computing the upward EM-messages (II.) and then performing message passing over the parameter edges (III.). This schedule is illustrated in Fig. 6a. Finally, the maxima of the distributions are evaluated to obtain new iterates. This schedule is outlined in Alg. 1.

Algorithm 1 Message update schedule for factor graph-based EM for parameter identification.

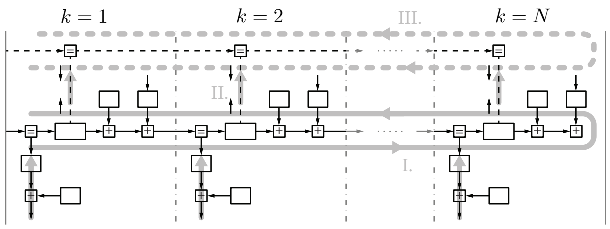
```
while model error > epsilon do
    Kalman Filter/Smoother update: Compute system model messages $\forall k$ based on all $\hat{\theta}_\bullet^{(j)}$ and calculate the marginals on all edges necessary.
    (Local) E-Step: Calculate $e^{\eta_k(\theta_\bullet)}$ messages $\forall k$ based on all $\theta_\bullet^{(j)}$ and calculate the marginals on all edges necessary.
    (Global) E-Step: Pass messages over the $\Theta_{\bullet,k}$ edges $\forall k$ to obtain lower bounds $Q_\bullet$.
    M-Step: Maximize lower bounds $Q_\bullet$ to obtain new iterates $\hat{\theta}_\bullet^{(j+1)}$.
    Set $j = j + 1$
end while
```

The above schedule uses factor graphs to make visible the possibility to localize the E-step and is close to the concept of using EM proposed in [4], [5]. In this regard the EM algorithm's property of guaranteed monotonic increase in likelihood is maintained [1], [13] for constant $\tilde{\Delta}_k$. Alternative schedules could be designed based on the factor graph interpretation, in order to yield new algorithms. Convergence analysis in these cases, however, remains future research.

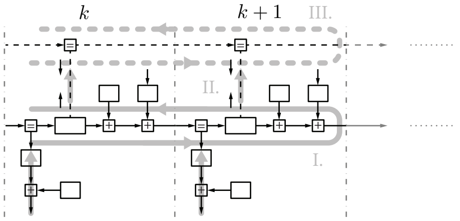
D. Recursive Estimation

The discrete time segmentation of the factor graph quite naturally implies options to design recursive estimation algorithms as illustrated in Fig. 6b. For this purpose, the estimation is first initialized until an acceptable accuracy is observed using a sufficient amount of data. After that, in each recursion the message passing for the EM algorithm over a single future time instant is performed with a predefined number of EM iterations, while the algorithm is properly initialized with prior EM-based estimation and Kalman Filter/Smoother results. An exponential weighting of past input data can be achieved—analog to the exponential weighting in the classical RLS algorithm—by introducing a forgetting factor λ. Each iterate of the current covariance estimate $\mathrm{V}_{\Theta'_{A_k}}$ with which the recursion is initialized is then scaled by λ (resp. $\mathrm{W}_{\Theta'_{A_k}}$ by $1/\lambda$) before new data is processed, see [8] for the corresponding RLS setting.

In addition, the factor graph framework allows for intuition in further associated design choices regarding the inclusion of additional past data in each step, or the frequency with which the recursion should be performed as well as the number of EM iterations.



(a) Non-recursive schedule.



(b) Recursive schedule.

Fig. 6: Illustration of recursive and non-recursive message passing schedules for EM with factor graphs.

IV. Results

A. Bilinear System Identification

EM-based identification of affine LPV systems is tested with the following system

$$\mathcal{G}_\Delta : \left\{ \begin{array}{l} \begin{bmatrix} x_{k+1} \\ \hdashline p_k \\ \hdashline y_k \end{bmatrix} = \left[\begin{array}{cc:cccc:cc} 0.5 & 1.1 & 0.3 & 0.0 & 0.2 & 0.0 & 0.0 & 1.0 \\ 0.2 & 0.3 & 0.0 & 0.1 & 0.0 & 0.2 & -1.0 & 0.0 \\ \hdashline -1 & 0 & \multicolumn{4}{c:}{\multirow{2}{*}{0}} & \multicolumn{2}{c}{\multirow{2}{*}{0}} \\ 0 & -1 & & & & & & \\ \hdashline 1 & 0 & \multicolumn{4}{c:}{\multirow{2}{*}{0}} & \multicolumn{2}{c}{\multirow{2}{*}{0}} \\ 0 & 2 & & & & & & \end{array} \right] \begin{bmatrix} x_k \\ \hdashline q_k \\ \hdashline u_k \end{bmatrix} \\ \qquad q_k = (u_k \otimes I_2)\, p_k, \qquad u_k \in \boldsymbol{\Delta}. \end{array} \right. \tag{15}$$

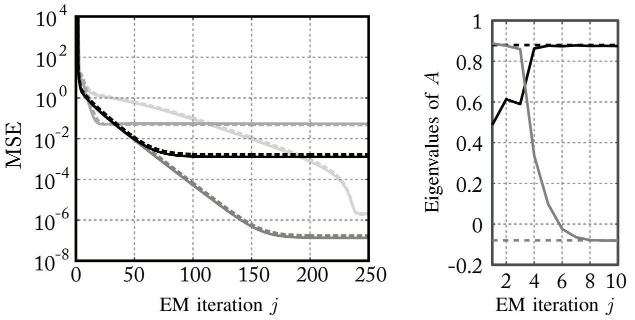
Exemplary results are presented for two cases using the basic model with low and high normalized Gaussian process (I) and measurement (D) noise levels of

(i) $\mathrm{V}_D = 10^{-1} I_{n_y}$, $\mathrm{V}_I = 10^{-1} I_{n_x}$,
(ii) $\mathrm{V}_D = 10^{-3} I_{n_y}$, $\mathrm{V}_I = 10^{-3} I_{n_x}$.

A data set of 1500 time steps is used. Identification is initialized with random matrices A, B_Δ and C, which in the first few iterations may even lead to an initially unstable output behavior. The presented example showcases the property of the EM algorithm to converge quickly from a far off initial parameter set, usually without getting stuck in a local minimum on the way.

Fig. 7a shows the mean squared error (MSE) between simulated outputs of the true and the estimated system both w. r. t. training and validation inputs. It can be observed that lower noise levels lead to decreased MSEs. As in each case the noise sequence of the simulations of the true and estimated systems are chosen to be identical, lower MSE purely reflect improved estimation results. The EM algorithm leads to strong declines in estimation error already within very few iterations, then proceeds with exponential decrease before converging to a static level.

Convergence w. r. t. the eigenvalues of the system matrix A is fast and consistent across noise levels as shown in Fig. 7b, in which case (i) and (ii) are indiscernible. In fact, the convergence w. r. t. the eigenvalues coincides with the transition to the MSE dropping with logarithmic rate between iterations 8 and 65. For completeness Fig. 8 shows a step response of the true and the estimated system for case (i).



(a) MSE between simulated outputs of the true and the estimated system w. r. t. training (——) and validation (- - - -) data. factor graph/EM (i): (——/- - - -); (ii): (——/- - - -); EM [5] (i): (——/- - - -); (ii): (——/- - - -).

(b) Eigenvalues of system matrix A. Estimated (——) and true (- - - -) values.

Fig. 7: Estimation results for LPV system $\mathcal{G}_\Delta$ versus EM iteration j.

In addition, Fig. 7a also includes identification results using a an FG-based implementation of the Kalman Filter/Smoother in conjunction with a non-factor graph-based implementation of the EM algorithm from [5]. Differences in the final obtained accuracy are attributed to implementation issues. The slow convergence in case (ii) using the non-factor graph-based EM again coincides with convergence of the system matrix eigenvalues, of which the second only converges at around iteration 240. Further investigation into the different behaviors is subject to future research, but the comparison already indicates the validity of the factor graph-based approach with good performance.

B. Recursive Bilinear System Identification

Recursive system identification is tested for case (ii) both with system $\mathcal{G}_\Delta$, cf. (15), as well as with a slowly time-varying variation, in which the system matrix is altered as

$$A(k) = \begin{bmatrix} 0.5 \cdot \cos\left(\frac{k\pi}{250}\right) & 1.1 \\ 0.2 & 0.3 \end{bmatrix}.$$

Note that this time-varying nature of the system matrix could as well be represented in an LFR. However, it is supposed that the variation is unknown, e. g., due to an aging process. After initialization with 200 EM iterations, in each recursion 50 EM iterations are performed. Fig. 9a illustrates the tracking of the eigenvalues of the time-varying system matrix $A(k)$, whereas Fig. 9b showcases the MSE between simulated results of both true and estimated model using training and validation data after each 50 EM iterations in each time instant k.

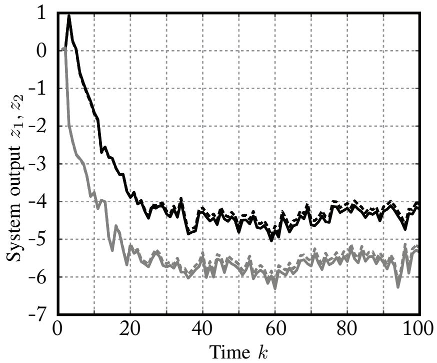
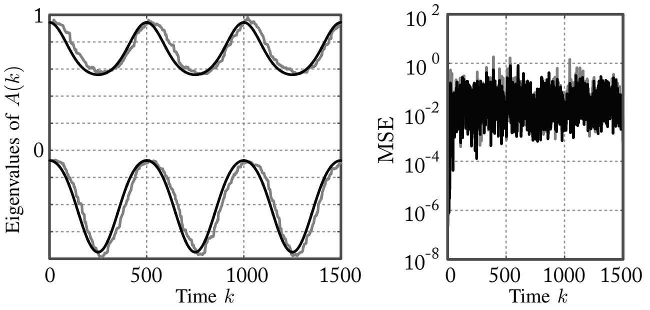


Fig. 8: Simulated outputs of the true (——) and the estimated (- - - -) system w. r. t. step inputs for case (i).
$z_{1,k}$: (——/- - - -); $z_{2,k}$: (——/- - - -)



(a) Eigenvalues of system matrix $A(k)$. True (——) and estimated (——) values.

(b) MSE vs. time instant k between true and the estimated system outputs w. r. t. training (——) and validation (——) inputs after all EM iterations.

Fig. 9: Recursive estimation results for LPV system $\mathcal{G}_\Delta$ with time-varying system matrix versus time instant k.

In each recursion, the information matrix w. r. t. the EM parameter for the system matrix is initialized with a forgetting factor $\lambda = 1.1$.

V. Outlook and Conclusion

The factor graph approach is feasible for representing EM-based identification of affine LPV systems with known dependence on scheduling parameters without uncertainty. The factor graph framework requires relatively simple extensions, in order to allow parameter-dependency and consequently promotes itself as a unifying modeling language for model-based inference problems in systems and control.

In the example, the identification performance appears well suited and reliable. The fast initial convergence from a large region of attraction—typical for EM—is observed, whereas convergence slows down considerably near regions of higher accuracy.

The factor graph-based approach to system identification can be further extended to incorporate rational parameter-dependency using LFT-based LPV models by introducing loops. For this purpose, an investigation into suitable message passing schedules will be necessary. Furthermore, future research can aim at including noisy or even a priori unknown scheduling signals within the factor graph framework.

Future research will also investigate the suitability for identification of real world systems with real data.

Appendix

Tabs. I and II list message passing rules for Gaussian messages. Their derivation is presented in [7].

References

[1] J. Dauwels, A. W. Eckford, S. Korl, and H.-A. Loeliger, "Expectation maximization as message passing - part i: principles and gaussian messages," *ArXiv*, 2009.

[2] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Royal Stat. Soc. Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[3] J. C. Doyle, A. K. Packard, and K. Zhou, "Review of lfts, lmis, and mu," in *Proc. 31st IEEE Conf. Decision Contr.*, 1992.

[4] S. Gibson and B. Ninness, "Robust maximum-likelihood estimation of multivariable dynamic systems," *Automatica*, vol. 41, no. 10, pp. 1667–1682, 2005, ISSN: 00051098.

[5] S. Gibson, A. Wills, and B. Ninness, "Maximum-likelihood parameter estimation of bilinear systems," *IEEE Trans. Automat. Contr.*, vol. 50, no. 10, pp. 1581–1596, 2005, ISSN: 0018-9286.

[6] S. Korl, "A factor graph approach to signal modelling, system identification and filtering," Ph.D. Thesis, ETH Zurich, Zurich, Switzerland, 2005.

[7] H.-A. Loeliger, L. Bruderer, H. Malmberg, F. Wadehn, and N. Zalmai, "On sparsity by nuv-em, gaussian message passing, and kalman smoothing," *ArXiv*, 2016.

TABLE I: Gaussian message passing through an equality-constraint, summation and multiplier node [7].

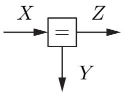
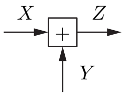
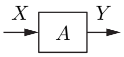
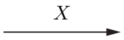
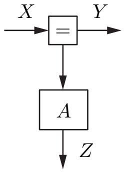
		
$\overrightarrow{\xi}_Z = \overrightarrow{\xi}_X + \overleftarrow{\xi}_Y$ $\overrightarrow{\mathrm{W}}_Z = \overrightarrow{\mathrm{W}}_X + \overleftarrow{\mathrm{W}}_Y$	$\overrightarrow{\mathrm{m}}_Z = \overrightarrow{\mathrm{m}}_X + \overrightarrow{\mathrm{m}}_Y$ $\overrightarrow{\mathrm{V}}_Z = \overrightarrow{\mathrm{V}}_X + \overrightarrow{\mathrm{V}}_Y$	$\overrightarrow{\mathrm{m}}_Y = A\overrightarrow{\mathrm{m}}_X$ $\overrightarrow{\mathrm{V}}_Y = A\overrightarrow{\mathrm{V}}_X A^\top$
$\overleftarrow{\xi}_X = \overleftarrow{\xi}_Y + \overleftarrow{\xi}_Z$ $\overleftarrow{\mathrm{W}}_X = \overleftarrow{\mathrm{W}}_Y + \overleftarrow{\mathrm{W}}_Z$	$\overleftarrow{\mathrm{m}}_X = \overleftarrow{\mathrm{m}}_Z - \overrightarrow{\mathrm{m}}_Y$ $\overleftarrow{\mathrm{V}}_X = \overleftarrow{\mathrm{V}}_Z + \overrightarrow{\mathrm{V}}_Y$	$\overleftarrow{\xi}_X = A^\top \overleftarrow{\xi}_Y$ $\overleftarrow{\mathrm{W}}_X = A^\top \overleftarrow{\mathrm{W}}_Y A$
$\mathrm{m}_X = \mathrm{m}_Y = \mathrm{m}_Z$ $\mathrm{V}_X = \mathrm{V}_Y = \mathrm{V}_Z$	$\mathrm{m}_Z = \mathrm{m}_X + \mathrm{m}_Y$ $\tilde{\mathrm{W}}_X = \tilde{\mathrm{W}}_Y = \tilde{\mathrm{W}}_Z$	$\mathrm{m}_Y = A\mathrm{m}_X$ $\mathrm{V}_Y = A\mathrm{V}_X A^\top$
$\tilde{\xi}_X = \tilde{\xi}_Y + \tilde{\xi}_Z$	$\tilde{\xi}_X = \tilde{\xi}_Y = \tilde{\xi}_Z$	$\tilde{\xi}_X = A^\top \tilde{\xi}_Y$ $\tilde{\mathrm{W}}_X = A^\top \tilde{\mathrm{W}}_Y A$

TABLE II: Gaussian single edge marginals (m, V) and their duals $(\tilde{\xi}, \tilde{\mathrm{W}})$ and message passing through an observation block [7].

	
$\tilde{\xi}_X \overset{\triangle}{=} \tilde{\mathrm{W}}_X(\overrightarrow{\mathrm{m}}_X - \overleftarrow{\mathrm{m}}_X)$ $= \overrightarrow{\xi}_X - \overrightarrow{\mathrm{W}}_X \mathrm{m}_X$ $= \overleftarrow{\mathrm{W}}_X \mathrm{m}_X - \overleftarrow{\xi}_X$	$\overrightarrow{\mathrm{m}}_Z = \overrightarrow{\mathrm{m}}_X + \overrightarrow{\mathrm{V}}_X A^\top G(\overleftarrow{\mathrm{m}}_Y - A\overrightarrow{\mathrm{m}}_X)$ $\overrightarrow{\mathrm{V}}_Z = \overrightarrow{\mathrm{V}}_X - \overrightarrow{\mathrm{V}}_X A^\top G A \overrightarrow{\mathrm{V}}_X$ w/ $G \overset{\triangle}{=} (\overleftarrow{\mathrm{V}}_Y + A\overrightarrow{\mathrm{V}}_X A^\top)^{-1}$
$\tilde{\mathrm{W}}_X \overset{\triangle}{=} (\overrightarrow{\mathrm{V}}_X + \overleftarrow{\mathrm{V}}_X)^{-1}$ $= \overrightarrow{\mathrm{W}}_X \mathrm{V}_X \overleftarrow{\mathrm{W}}_X$ $= \overrightarrow{\mathrm{W}}_X - \overrightarrow{\mathrm{W}}_X \mathrm{V}_X \overrightarrow{\mathrm{W}}_X$ $= \overleftarrow{\mathrm{W}}_X - \overleftarrow{\mathrm{W}}_X \mathrm{V}_X \overleftarrow{\mathrm{W}}_X$	$\tilde{\xi}_X = F^\top \tilde{\xi}_Z + A^\top \overleftarrow{\mathrm{W}}_Y (A\overrightarrow{\mathrm{m}}_Z - \overleftarrow{\mathrm{m}}_Y)$ $= F^\top \tilde{\xi}_Z + A^\top G(A\overrightarrow{\mathrm{m}}_X - \overleftarrow{\mathrm{m}}_Y)$
$\mathrm{m}_X = \mathrm{V}_X(\overrightarrow{\xi}_X + \overleftarrow{\xi}_X)$ $= \overrightarrow{\mathrm{m}}_X - \overrightarrow{\mathrm{V}}_X \tilde{\xi}_X$ $= \overleftarrow{\mathrm{m}}_X + \overleftarrow{\mathrm{V}}_X \tilde{\xi}_X$	$\tilde{\mathrm{W}}_X = F^\top \tilde{\mathrm{W}}_Z F + A^\top \overleftarrow{\mathrm{W}}_Y A F$ $= F^\top \tilde{\mathrm{W}}_Z F + A^\top G A$
$\mathrm{V}_X = (\overrightarrow{\mathrm{W}}_X + \overleftarrow{\mathrm{W}}_X)^{-1}$ $= \overrightarrow{\mathrm{V}}_X \tilde{\mathrm{W}}_X \overleftarrow{\mathrm{V}}_X$ $= \overrightarrow{\mathrm{V}}_X - \overrightarrow{\mathrm{V}}_X \tilde{\mathrm{W}}_X \overrightarrow{\mathrm{V}}_X$ $= \overleftarrow{\mathrm{V}}_X - \overleftarrow{\mathrm{V}}_X \tilde{\mathrm{W}}_X \overleftarrow{\mathrm{V}}_X$	$\tilde{\xi}_X = \tilde{\xi}_Y = \tilde{\xi}_Z$ w/ $F \overset{\triangle}{=} I - \overrightarrow{\mathrm{V}}_Z A^\top \overleftarrow{\mathrm{V}}_Y A$ $= I - \overrightarrow{\mathrm{V}}_X A^\top G A$

[8] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," *Proc. IEEE*, vol. 95, no. 6, pp. 1295–1322, 2007, ISSN: 0018-9219.

[9] H. Neudecker, "Some theorems on matrix differentiation with special reference to kronecker matrix products," vol. 64, no. 327, pp. 953–963, 1969, ISSN: 0162-1459.

[10] D. Piga, P. Cox, R. Toth, and V. Laurain, "Lpv system identification under noise corrupted scheduling and output signal observations," vol. 53, pp. 329–338, 2015, ISSN: 00051098.

[11] W. J. Rugh and J. S. Shamma, "A survey of research on gain-scheduling," *Automatica*, vol. 36, pp. 1401–1425, 2000, ISSN: 00051098.

[12] C. W. Scherer, "Lpv control and full block multipliers," *Automatica*, vol. 37, no. 3, pp. 361–375, 2001, ISSN: 00051098.

[13] P. Stoica and Y. Selen, "Cyclic minimizers, majorization techniques, and the expectation-maximization algorithm: a refresher," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 112–114, 2004, ISSN: 1053-5888.

[14] R. Toth, P. S. C. Heuberger, and Van den Hof, Paul M. J., "Prediction-error identification of lpv systems: present and beyond," in *Control of Linear Parameter Varying Systems with Applications*, J. Mohammadpour and C. W. Scherer, Eds., Boston, MA: Springer US, 2012, pp. 27–58, ISBN: 9781461418337.

[15] A. Wills and B. Ninness, "System identification of linear parameter varying state-space models," in *Linear parameter-*

varying system identification, ser. Advanced series in electrical and computer engineering, P. L. d. Santos, Ed., vol. v. 14, Singapore and London: World Scientific, 2012, pp. 295–315, ISBN: 9789814355445.

[16] F. Wu, "Control of linear parameter varying systems," Ph.D. Thesis, University of Berkeley, California and USA, 1995.

[17] F. Wu and K. Dong, "Gain-scheduling control of lft systems using parameter-dependent lyapunov functions," *Automatica*, vol. 42, no. 1, pp. 39–50, 2006, ISSN: 00051098.