Automatica 135 (2022) 109949

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Parameter estimation for Jump Markov Linear Systems[☆]

Mark P. Balenzuela [*], Adrian G. Wills, Christopher Renton, Brett Ninness

School of Engineering, The University of Newcastle, Callaghan, NSW 2308, Australia

ARTICLE INFO

Article history:
Received 29 June 2020
Received in revised form 26 May 2021
Accepted 19 August 2021
Available online 20 October 2021

Keywords:
Jump Markov systems
Nonlinear state estimation
Optimisation
Expectation maximisation
Identification

ABSTRACT

Jump Markov linear systems (JMLS) are a useful model class for capturing abrupt changes in system behaviour that are temporally random, such as when a fault occurs. In many situations, accurate knowledge of the model is not readily available and can be difficult to obtain based on first principles. This paper presents a method for learning parameter values of this model class based on available input–output data using the maximum-likelihood framework. In particular, the expectation–maximisation method is detailed for this model class with attention given to a deterministic and numerically stable implementation. The presented algorithm is compared to state-of-the-art methods on several simulation examples with favourable results.

1. Introduction

In many situations of practical importance, there exists important underlying dynamics that are known to abruptly change, which will be referred to here as “switched systems”. Examples abound in areas of econometrics (Kim, 1994), telecommunications (Logothetis & Krishnamurthy, 1999), target tracking (Mazor, Averbuch, Bar-Shalom, & Dayan, 1998), fault detection and isolation (FDI) (Blackmore, Gil, Chung, & Williams, 2007; Hashimoto, Kawashima, Nakagami, & Oba, 2001), robotics (Gil & Williams, 2009) and medical applications (Ghahramani & Hinton, 2000). In these situations, for the purposes of control, fault detection, trajectory prediction, decision making and channel equalisation (amongst others), it is vital to be able to both model all the possible dynamics as well as estimating when the dynamics have changed, and to which new mode of operation they have changed to.

Unfortunately, it is often the case that it is difficult to obtain the required dynamic models from first principles considerations, and in addition the time instants when the system changes between modes are not known a-priori. This latter difficulty, together with the importance of the problem of switched system modelling has been recognised by many other authors who have taken an approach of deriving algorithms to estimate both the dynamics and the switching times on the basis of observed system behaviour.

The literature is quite broad, but to focus on approaches most closely related to the work to follow here, there has been strong interest in the application of Sequential Monte Carlo (SMC) methods that treat the problem as a general nonlinear state–space estimation challenge, which while effective can be computationally expensive (Ashley & Andersson, 2014; Schön, Wills, & Ninness, 2011). This computational burden has been shown to be reducible by using Rao-Blackwellization (Blackmore et al., 2007; Svensson, Schön, & Lindsten, 2014), which exploits any conditionally linearity in the dynamic model, and for the Gaussian case, well-known prediction, smoothing and estimation methods exist (Kalman, 1960; Rauch, Striebel, & Tung, 1965).

More recently, stochastic approximation (SA) (Andrieu, Moulines, & Priouret, 2005; Delyon, Lavielle, Moulines, et al., 1999) ideas have been incorporated within the SMC-based framework by Lindsten (2013) and Svensson et al. (2014) to make more efficient use of finite computational resources, improve bias and reduce variance. As an alternative to these SMC approaches, a variational approximation algorithm has been developed in Ghahramani and Hinton (2000).

In addition, switched system models can fall within the class of hybrid systems (Paoletti, Juloski, Ferrari-Trecate, & Vidal, 2007) for which alternative estimation approaches exist. One such approach partitions the data into disjoint segments (Paoletti et al., 2007; Yildirim, Singh, & Doucet, 2013), and performs identification for each segment. Accurately segmenting the data is a major challenge with these approaches (Chen, Bako, & Lecoeuche, 2011; Paoletti et al., 2007).

This paper examines a maximum likelihood approach where a “Jump Markov Linear System” (JMLS) description (Doucet, Gordon, & Krishnamurthy, 2001) is used to capture underlying switching dynamics. This results in a non-convex optimisation problem which will be addressed here using the well known

[☆] The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Antonio Vicino under the direction of Editor Torsten Söderström.

[*] Corresponding author.

E-mail addresses: mark.balenzuela@uon.edu.au (M.P. Balenzuela), adrian.wills@newcastle.edu.au (A.G. Wills), christopher.renton@newcastle.edu.au (C. Renton), brett.ninness@newcastle.edu.au (B. Ninness).

https://doi.org/10.1016/j.automatica.2021.109949
0005-1098/© 2021 Elsevier Ltd. All rights reserved.

"Expectation–Maximisation" (EM) method (Dempster, Laird, & Rubin, 1977).

As the name suggests, this involves a two stage approach where by an expectation (E) step, involving the computation of smoothed quantities delivers a cost function that approximates the underlying likelihood function (L), but is simpler to maximise (M-step) than the likelihood L itself. A series of E and M steps, which continually increase the likelihood approximation at each iteration, also increases the true likelihood L and provides an identification solution.

Indeed, this same approach has been employed in many of the works cited above, all of which have dealt with the problem of complexity in the E-step, which grows exponentially according to m^N, where m is the number of possible modes, and N is the observed data length.

This paper also addresses this exponential complexity and makes the following new contributions relative to existing work:

(1) A novel E-step solution based on a Two-Filter joint smoothing algorithm is developed. This smoothing algorithm employs a merging method to address the above-mentioned exponential complexity. The resulting E-step is a deterministic quantification (as opposed, for example, to SMC methods stochastically approximate the E-step calculation) that can be tuned for accuracy or speed;
(2) An M-step solution that provides parameter updates in closed-form. This includes calculation of a certain a cross-covariance term $\mathbf{S}_\ell$, which has not been estimated and employed in previous work;
(3) Numerically stable implementations of both the Expectation and Maximisation steps are developed via carefully constructed square-root formulations, which are essential in practical deployment of EM-based methods.

Supplementary MATLAB code for this paper is available at: https://bitbucket.org/M_P_Balenzuela/jmls_em/src/master/

2. Problem formulation

In this paper, we address the modelling of systems which can jump between a finite number of linear dynamic system modes using a discrete-time jump-Markov-linear-system description (JMLS) (Doucet et al., 2001), which can be expressed as

$$\underbrace{\begin{bmatrix} y_k \\ x_{k+1} \end{bmatrix}}_{\triangleq \zeta_k} = \underbrace{\begin{bmatrix} \mathbf{C}_{z_k} & \mathbf{D}_{z_k} \\ \mathbf{A}_{z_k} & \mathbf{B}_{z_k} \end{bmatrix}}_{\triangleq \boldsymbol{\Gamma}_{z_k}} \underbrace{\begin{bmatrix} x_k \\ u_k \end{bmatrix}}_{\triangleq \xi_k} + \underbrace{\begin{bmatrix} e_k \\ v_k \end{bmatrix}}_{w_k}, \tag{1a}$$

where $x_k \in \mathbb{R}^{n_x}$ is the system state, $y_k \in \mathbb{R}^{n_y}$ is the system output, $u_k \in \mathbb{R}^{n_u}$ is the system input, $z_k \in \{1, \ldots, m\}$ is a discrete variable that is often called the *model index* that indexes the active parameter set, $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$, $\mathbf{C} \in \mathbb{R}^{n_y \times n_x}$, and $\mathbf{D} \in \mathbb{R}^{n_y \times n_u}$ are general matrices, and the noise terms v_k and e_k originate from the Gaussian white noise process

$$w_k \sim \mathcal{N}_{w_k}\left(0\ ,\ \boldsymbol{\Pi}_{z_k}\right), \quad \boldsymbol{\Pi}_{z_k} = \begin{bmatrix} \mathbf{R}_{z_k} & \mathbf{S}^T_{z_k} \\ \mathbf{S}_{z_k} & \mathbf{Q}_{z_k} \end{bmatrix}, \tag{1b}$$

where $\mathbf{Q} \in \mathbb{R}^{n_x \times n_x}$, and $\mathbf{R}^{n_y \times n_y}$ are symmetric positive definite matrices constructing the symmetric positive definite matrix $\boldsymbol{\Pi}$, and

$$\mathcal{N}_x(\mu, \mathbf{P}) = \det(2\pi\mathbf{P})^{-\frac{1}{2}} e^{-\frac{1}{2}(x-\mu)^T\mathbf{P}^{-1}(x-\mu)} \tag{2}$$

denotes a multivariate Normal distribution. The subscript x on $\mathcal{N}_x$ indicates the variable of the distribution, which will be very important in this paper in order to track where certain variables enter the distribution. The system matrices and noise covariances are allowed to randomly jump or switch values as a function of the model index z_k. A switch event is captured by allowing z_k to transition to z_{k+1} stochastically with the probability of transitioning from the jth model at time-index k to the ith model at time-index $k+1$ given by

$$\mathbb{P}(z_{k+1} = i | z_k = j) = \mathbf{T}_{i,j}, \quad \sum_{i=1}^{m} \mathbf{T}_{i,j} = 1 \quad \forall j. \tag{3}$$

The collection of model parameters, which fully describe the above JMLS class, can be conveniently collected into a parameter object θ, defined as

$$\theta \triangleq \left\{\mathbf{T}, \{\boldsymbol{\Gamma}_i, \boldsymbol{\Pi}_i\}_{i=1}^{m}\right\}. \tag{4}$$

Problem. Presented with known state dimension n_x, known number of system modes $m > 0$, and a data record of N outputs and inputs

$$\mathbf{y} \triangleq y_{1:N} = \{y_1, \ldots, y_N\}, \quad \mathbf{u} \triangleq u_{1:N} = \{u_1, \ldots, u_N\},$$

respectively, the primary focus of this paper is to obtain an estimate of θ, denoted as $\widehat{\theta}$, by solving the maximum-likelihood problem

$$\widehat{\theta} = \arg\max_{\theta} \ln p_\theta(y_{1:N}). \tag{5}$$

The log-likelihood $\ln p_\theta(y_{1:N})$ can be further expressed using conditional probability as

$$\begin{aligned} \ln p_\theta(y_{1:N}) &= \ln p_\theta(y_1) + \sum_{k=2}^{N} \ln p_\theta(y_k \mid y_{1:k-1}) \\ &= \ln \int \sum_{z_1} p_\theta(y_1 \mid x_1, z_1)\, p_\theta(x_1, z_1)\, \mathrm{d}x_1 \\ &\quad + \sum_{k=2}^{N} \ln \int \sum_{z_k} p_\theta(y_k \mid x_k, z_k)\, p_\theta(x_k, z_k \mid y_{1:k-1})\, \mathrm{d}x_k, \end{aligned}$$

where the second equality is obtained using the state space model (1)–(3).

This decomposes the likelihood calculation into a state prediction density and a measurement likelihood function. In the linear case, these quantities and associated expectation can be computed using Kalman filtering methods (Anderson & Moore, 2005; Gibson & Ninness, 2005). In more general non-linear cases, methods based on Monte Carlo approximation of the integral have been explored (Doucet, Godsill, & Andrieu, 2000; Schön et al., 2011). In our case, the specific structure of the state depending on z_k leads to a situation where the expectation involves a sum over a number of terms growing exponentially in data length N.

This implies difficulty in the computation of the likelihood. Furthermore, even when computed, the likelihood $p_\theta(y_{1:N})$ is non-convex which makes the optimisation problem (5) even more challenging. This paper will address both these issues in an expectation maximisation framework.

3. An expectation–maximisation approach

While, for reasons just mentioned, the estimation problem (5) is very challenging, it is interesting to note that if the unknown state and model sequences

$$\mathbf{x} \triangleq x_{1:N+1}, \qquad \mathbf{z} \triangleq z_{1:N+1}. \tag{6}$$

were available then the *full-data* log-likelihood incorporating this knowledge in addtion to $\mathbf{y}$ can be expressed as

$$\ln p_\theta\left(\mathbf{x}, \mathbf{z}, \mathbf{y}\right) = \ln p(x_1, z_1) + \sum_{k=1}^{N} \ln \mathbf{T}_{z_{k+1}, z_k} + \sum_{k=1}^{N} \ln \mathcal{N}_{\zeta_k}\left(\boldsymbol{\Gamma}_{z_k} \xi_k,\ \boldsymbol{\Pi}_{z_k}\right), \tag{7}$$

whose maximum over θ can be expressed in closed-form. This suggests a possible approach of *estimating* the quantities in (6), substituting these estimates in (7) and using the result as an approximation to the solution of the identification problem (5).

This is the essence of the Expectation–Maximisation (EM) method (Dempster et al., 1977), which was developed to solve a wide range of maximum-likelihood estimation problems of the type (5). The EM algorithm has been previously employed for closely related classes (Blackmore et al., 2007; Gil & Williams, 2009), and to more general Jump-Markov nonlinear state–space models (Ashley & Andersson, 2014).

In order to describe the contributions of the current paper relative to these and other existing works, it is important to provide some level of detail for the EM approach. To that end, the EM method is an iterative method, where the essential steps and mechanism for progress towards a maximum-likelihood estimate are summarised as follows. A fundamental connection between the likelihood $p_\theta(\mathbf{y})$ and the full-data likelihood $p_\theta(\mathbf{x}, \mathbf{z}, \mathbf{y})$ is provided by conditional probability via

$$p_\theta(\mathbf{x}, \mathbf{z}, \mathbf{y}) = p_\theta(\mathbf{x}, \mathbf{z} \mid \mathbf{y})\, p_\theta(\mathbf{y}). \tag{8}$$

Taking the logarithm of both sides and rearranging provides

$$\ln p_\theta(\mathbf{y}) = \ln p_\theta(\mathbf{x}, \mathbf{z}, \mathbf{y}) - \ln p_\theta(\mathbf{x}, \mathbf{z} \mid \mathbf{y}). \tag{9}$$

The key step is to approximate $\ln p_\theta(\mathbf{y})$ using the above relation, where the approximation estimates the *missing-data* $\mathbf{x}$ and $\mathbf{z}$ in the right-hand-side expression by their conditional expected values based on the available data $\mathbf{y}$. In particular, at iteration i, given a parameter estimate θ_i we may take expectations of both sides relative to the *data-dependent conditional distribution* $p_{\theta_i}(\mathbf{x}, \mathbf{z} \mid \mathbf{y})$, which we abbreviate using the following notation

$$\mathbb{E}_{\theta_i}\left\{\cdot\right\} \triangleq \int \sum_{\mathbf{z}} \{\cdot\} p_{\theta_i}(\mathbf{x}, \mathbf{z} \mid \mathbf{y})\, d\mathbf{x}, \tag{10}$$

to reveal (the so-called Expectation or E-step)

$$\begin{aligned} \ln p_\theta(\mathbf{y}) &= \mathbb{E}_{\theta_i}\left\{\ln p_\theta(\mathbf{y})\right\} \\ &= \underbrace{\mathbb{E}_{\theta_i}\left\{\ln p_\theta(\mathbf{x}, \mathbf{z}, \mathbf{y})\right\}}_{\mathcal{Q}(\theta,\theta_i)} - \underbrace{\mathbb{E}_{\theta_i}\left\{\ln p_\theta(\mathbf{x}, \mathbf{z} \mid \mathbf{y})\right\}}_{\mathcal{V}(\theta,\theta_i)}, \end{aligned} \tag{11}$$

where the first equality holds since $p_\theta(\mathbf{y})$ does not depend on $\mathbf{x}$ and $\mathbf{z}$ and probability distributions have unit mass, and the second equality comes directly from (9) and (10). Importantly, if we consider the log-likelihood difference resulting from θ and θ_i then

$$\begin{aligned} \ln p_\theta(\mathbf{y}) - \ln p_{\theta_i}(\mathbf{y}) &= \mathcal{Q}(\theta, \theta_i) - \mathcal{Q}(\theta_i, \theta_i) \\ &\quad + \mathcal{V}(\theta_i, \theta_i) - \mathcal{V}(\theta, \theta_i). \end{aligned}$$

The difference $\mathcal{V}(\theta_i, \theta_i) - \mathcal{V}(\theta, \theta_i)$ is known as a Kullback–Leibler divergence and has the important property of being non-negative. This reveals the primary mechanism for progress in the EM method, namely, in order to increase $\ln p_\theta(\mathbf{y})$ it is sufficient to search for θ_{i+1} such that (called the Maximisation or M-step)

$$\mathcal{Q}(\theta_{i+1}, \theta_i) \geq \mathcal{Q}(\theta_i, \theta_i) \implies \ln p_{\theta_{i+1}}(\mathbf{y}) \geq \ln p_{\theta_i}(\mathbf{y}).$$

This paper will detail both the Expectation and Maximisation steps for the JMLS class mentioned above. However, it is important to discuss a fundamental difficulty in applying EM, and more generally, find a maximum-likelihood solution for this class of systems. The above expectations involve a sum over all possible index values for all time, that is a sum over m^N possible combinations, which becomes computationally intractable for even modest data lengths when $m > 1$.

This exponential growth has motivated several researchers to consider approximations of $\mathcal{Q}(\theta, \theta_i)$ instead (Ashley & Andersson, 2014; Blackmore et al., 2007).

4. EM algorithm for JMLS

The EM algorithm iterates the following steps:

(1) **E-step:** Compute $\mathcal{Q}(\theta, \theta_i)$;
(2) **M-step:** Compute $\theta_{i+1} = \arg\max_\theta \mathcal{Q}(\theta, \theta_i)$.

In the following subsections, we detail each of the above steps, starting with the Maximisation step in Section 4.1, which will highlight the necessary objects required from the Expectation step, which is detailed in Section 4.2.

4.1. The maximisation step

In this section, expressions for the maximiser of $\mathcal{Q}(\theta, \theta_i)$ are provided. Towards this, recall from (11) that

$$\mathcal{Q}(\theta, \theta_i) = \mathbb{E}_{\theta_i}\left\{\ln p_\theta(\mathbf{x}, \mathbf{z}, \mathbf{y})\right\}, \tag{12}$$

where the conditional expectation is defined in (10) and depends on the data $\mathbf{y} = y_{1:N}$. Using (7), the function $\mathcal{Q}(\theta, \theta_i)$ can be further expressed as

$$\mathcal{Q}(\theta, \theta_i) = \mathcal{Q}_1(\theta_i) + \mathcal{Q}_2(\theta, \theta_i) + \mathcal{Q}_3(\theta, \theta_i), \tag{13}$$

where $\mathcal{Q}_1(\theta_i) = \mathbb{E}_{\theta_i}\left\{\ln p(x_1, z_1)\right\}$ and (recall ξ_k and ζ_k definitions from (1))

$$\mathcal{Q}_2(\theta, \theta_i) = \mathbb{E}_{\theta_i}\left\{\sum_{k=1}^{N} \ln \mathbf{T}_{z_{k+1}, z_k}\right\}, \tag{14a}$$

$$\mathcal{Q}_3(\theta, \theta_i) = \mathbb{E}_{\theta_i}\left\{\sum_{k=1}^{N} \ln \mathcal{N}_{\zeta_k}\left(\boldsymbol{\Gamma}_{z_k} \xi_k,\ \boldsymbol{\Pi}_{z_k}\right)\right\}. \tag{14b}$$

By the Principle of Minimum Cross-Entropy (Kullback, 1959), it follows that $\mathcal{Q}_1(\theta_i)$ is maximised when $p(x_1, z_1) = p(x_1, z_1 | \mathbf{y})$. In order to obtain the maximiser of $\mathcal{Q}_2(\theta, \theta_i) + \mathcal{Q}_3(\theta, \theta_i)$ over θ, it is convenient to express these functions so that the parameters for each model ℓ, namely $\{\mathbf{T}_{1:m,\ell}, \boldsymbol{\Gamma}_\ell, \boldsymbol{\Pi}_\ell\}$, are grouped together, which is provided by the following lemma.

Lemma 1. *The functions $\mathcal{Q}_2$ and $\mathcal{Q}_3$ can be expressed as*

$$\mathcal{Q}_2(\theta, \theta_i) = \sum_{j=1}^{m} \sum_{\ell=1}^{m} \alpha_{j,\ell} \ln \mathbf{T}_{j,\ell},$$

$$\mathcal{Q}_3(\theta, \theta_i) = c - \frac{1}{2} \sum_{\ell=1}^{m} \beta_\ell \ln |\boldsymbol{\Pi}_\ell| - \operatorname{tr}\left\{\boldsymbol{\Pi}_\ell^{-1} \mathcal{M}(\boldsymbol{\Gamma}_\ell)\right\},$$

where c is a constant that does not depend on θ and

$$\mathcal{M}(\boldsymbol{\Gamma}_\ell) = \boldsymbol{\Phi}_\ell - \boldsymbol{\Gamma}_\ell \boldsymbol{\Psi}_\ell^T - \boldsymbol{\Psi}_\ell \boldsymbol{\Gamma}_\ell^T + \boldsymbol{\Gamma}_\ell \boldsymbol{\Sigma}_\ell \boldsymbol{\Gamma}_\ell^T. \tag{15}$$

The remaining terms are defined via the expectations

$$\beta_\ell = \mathbb{E}_{\theta_i}\left\{\sum_{k=1}^{N} \delta_{\ell, z_k}\right\}, \tag{16a}$$

$$\alpha_{j,\ell} = \mathbb{E}_{\theta_i}\left\{\sum_{k=1}^{N} \delta_{j, z_{k+1}} \delta_{\ell, z_k}\right\}, \tag{16b}$$

$$\boldsymbol{\Phi}_\ell = \mathbb{E}_{\theta_i}\left\{\sum_{k=1}^{N} \delta_{\ell,z_k}\zeta_k\zeta_k^T\right\}, \tag{16c}$$

$$\boldsymbol{\Psi}_\ell = \mathbb{E}_{\theta_i}\left\{\sum_{k=1}^{N} \delta_{\ell,z_k}\zeta_k\xi_k^T\right\}, \tag{16d}$$

$$\boldsymbol{\Sigma}_\ell = \mathbb{E}_{\theta_i}\left\{\sum_{k=1}^{N} \delta_{\ell,z_k}\xi_k\xi_k^T\right\}. \tag{16e}$$

Proof. See Appendix B. □

With $\mathcal{Q}(\theta, \theta_i)$ expressed in this form, the following Lemma then provides expressions for the unique maximiser of $\mathcal{Q}(\theta, \theta_i)$ over θ.

Lemma 2 (*M-step*). *Assuming that $\boldsymbol{\Sigma}_\ell \succ 0$ and $\sum_{j=1}^{m} \alpha_{j,\ell} > 0$ for $\ell = 1, \ldots, m$, then the maximiser of $\mathcal{Q}_2(\theta, \theta_i)$ subject to (3), and the maximiser of $\mathcal{Q}_3(\theta, \theta_i)$ subject to (1) are given by*

$$\mathbf{T}_{j,\ell} = \frac{\alpha_{j,\ell}}{\sum_{j=1}^{m} \alpha_{j,\ell}}, \tag{17a}$$

$$\boldsymbol{\Gamma}_\ell = \boldsymbol{\Psi}_\ell \boldsymbol{\Sigma}_\ell^{-1}, \tag{17b}$$

$$\boldsymbol{\Pi}_\ell = \frac{1}{\beta_\ell}\left(\boldsymbol{\Phi}_\ell - \boldsymbol{\Psi}_\ell \boldsymbol{\Sigma}_\ell^{-1} \boldsymbol{\Psi}_\ell^T\right). \tag{17c}$$

Proof. See Appendix C. □

4.2. The expectation step

In order to implement the above maximisation step we require the computation of $\alpha_{j,\ell}$, β_ℓ, $\boldsymbol{\Phi}_\ell$, $\boldsymbol{\Psi}_\ell$ and $\boldsymbol{\Sigma}_\ell$ from (16), which are all defined as expectations. It turns out that, in general, it is intractable to compute these expectations exactly. Therefore, this section details a method for approximating these terms that allows a trade-off between computational speed and accuracy.

Towards this, we will first establish that the required terms (16) can all be expressed as expectations with respect to the joint smoothed distributions,

$$p(x_k, x_{k+1}, z_k, z_{k+1} \mid \mathbf{y}) \quad \forall k = 1, \ldots, N. \tag{18}$$

We then show that the joint distributions (18) can be approximated with a Gaussian mixture with M_k components

$$p(\chi_k, \eta_k \mid \mathbf{y}) \approx \sum_{j=1}^{M_k} w_k^{j,\eta_k} \mathcal{N}_{\chi_k}\left(\mu_k^{j,\eta_k}, \mathbf{P}_k^{j,\eta_k}\right), \tag{19}$$

where for convenience we introduce the following

$$\chi_k \triangleq \begin{bmatrix} x_k^T & x_{k+1}^T \end{bmatrix}^T, \qquad \eta_k \triangleq (z_{k+1}, z_k). \tag{20}$$

Although this approximation can be made exact, it is well known that this requires exponential growth in the number of components M_k as a function of data length N (Alspach & Sorenson, 1972). This leads to exponential growth in computational complexity and rapidly becomes intractable.

To combat this exponential growth, we extend our previous work in Balenzuela, Wills, Renton, and Ninness (2020) that uses a two-filter approach and a merging strategy to limit the number of components M_k to manageable levels. When compared to other JMLS smoothing schemes, such as the interacting multiple model (IMM) smoother (Helmick, Blair, & Hoffman, 1995), this approach is capable of providing smoothed distributions with either a restriction on the number of components M_k being removed, or with reduced assumptions. In either case, this approach is capable of generating smoothed distributions with improved accuracy, and therefore provides a better approximation of $\mathcal{Q}(\theta, \theta_i)$. Importantly, even the improved JMLS smoother cannot generate (18) exactly due to the exponential number of computations required, and we use the approximation (19). Nonetheless, this approximation has the major benefit of providing a direct trade-off between computational complexity and accuracy. Additionally, a small extension is required to produce (19) as the previous work does not target the joint smoothing distributions (18).

With the approximation (19) for the joint smoothing distributions (18) in place, we then approximate the required terms (16) as combinations of the weights $w_k^{j,\eta_k} \geq 0$, means $\mu_k^{j,\eta_k} \in \mathbb{R}^{2n_x}$ and covariances $\mathbf{P}_k^{j,\eta_k} \in \mathbb{R}^{2n_x \times 2n_x}$.

Towards this end, the following Lemma establishes that (16) can be expressed in terms of the joint distributions (18). Note that these expressions are exact.

Lemma 3. *The expectations in (16) can be stated as (recall ξ_k and ζ_k definitions from (1))*

$$\beta_\ell = \sum_{k=1}^{N} p_{\theta_i}(z_k = \ell \mid \mathbf{y}), \tag{21a}$$

$$\alpha_{j,\ell} = \sum_{k=1}^{N} p_{\theta_i}(z_{k+1} = j, z_k = \ell \mid \mathbf{y}), \tag{21b}$$

$$\boldsymbol{\Phi}_\ell = \sum_{k=1}^{N} \int \zeta_k\zeta_k^T p_{\theta_i}(x_{k+1}, z_k = \ell \mid \mathbf{y})\, dx_{k+1}, \tag{21c}$$

$$\boldsymbol{\Psi}_\ell = \sum_{k=1}^{N} \int \zeta_k\xi_k^T p_{\theta_i}(\chi_k, z_k = \ell \mid \mathbf{y})\, d\chi_k, \tag{21d}$$

$$\boldsymbol{\Sigma}_\ell = \sum_{k=1}^{N} \int \xi_k\xi_k^T p_{\theta_i}(x_k, z_k = \ell \mid \mathbf{y})\, dx_k. \tag{21e}$$

Proof. See Appendix D. □

From the above Lemma, it can be observed that only the joint smoothing distributions (18) and marginals of it are required for computing the terms (16). The smoother solution provided in Balenzuela et al. (2020) is almost directly applicable, save for two important differences

(1) it requires that cross-covariance $\mathbf{S}_{z_k} = 0$, and
(2) it targets the marginal $p_{\theta_i}(x_k, z_k \mid \mathbf{y})$.

In what follows we will address these two differences by presenting explicit formulas for the required joint distributions (18).

To address the first issue, we note that the system can be transformed to an equivalent state–space model (Kailath, Sayed, & Hassibi, 2000), where the cross-covariance term $\mathbf{S} = 0$ via

$$\begin{aligned} \bar{\mathbf{A}}_{z_k} &= \mathbf{A}_{z_k} - \mathbf{C}_{z_k}, \quad \bar{\mathbf{Q}}_{z_k} = \mathbf{Q}_{z_k} - \mathbf{S}_{z_k}\mathbf{R}_{z_k}^{-1}\mathbf{S}_{z_k}^T, \\ \bar{\mathbf{B}}_{z_k} &= \begin{bmatrix} \mathbf{B}_{z_k} - \mathbf{S}_{z_k}\mathbf{R}_{z_k}^{-1}\mathbf{D}_{z_k} & \mathbf{S}_{z_k}\mathbf{R}_{z_k}^{-1} \end{bmatrix}, \\ \bar{\mathbf{D}}_{z_k} &= \begin{bmatrix} \mathbf{D}_{z_k} & \mathbf{0} \end{bmatrix}, \quad \bar{u}_k = \begin{bmatrix} u_k^T & y_k^T \end{bmatrix}^T, \end{aligned} \tag{22}$$

and $\bar{\mathbf{C}}_{z_k} = \mathbf{C}_{z_k}$ and $\bar{\mathbf{R}}_{z_k} = \mathbf{R}_{z_k}$ are unchanged.

To address the second issue of targeting the joint distribution (18), we will extend the two-filter solution in Balenzuela et al. (2020), which only targets the marginal $p_{\theta_i}(x_k, z_k \mid \mathbf{y})$. Importantly, this two-filter approach relies on a combination of forward filtered and backward information likelihood terms, which for completeness have been summarised in Appendix A.

Conveniently, we can reuse these forward filtered and backward information likelihood expressions in forming the joint smoothed distributions (18). In particular, the forward filtered expression is given by the following approximation

$$p(x_k, z_k \mid y_{1:t}) \approx \sum_{i=1}^{M_k^{\mathrm{f}}} w_{k|k}^{i,z_k} \mathcal{N}_{x_k}\left(\mu_{k|k}^{i,z_k}, \mathbf{P}_{k|k}^{i,z_k}\right), \tag{23}$$

and backward information likelihood by

$$p(y_{k:N} | x_k, z_k) \approx \sum_{\ell=1}^{M_k^{\mathrm{c}}} \mathcal{L}_{x_k}\left(r_k^{\ell,z_k}, s_k^{\ell,z_k}, \mathbf{L}_k^{\ell,z_k}\right), \tag{24}$$

where $\mathcal{L}$ is a likelihood function defined as

$$\mathcal{L}_x(r, s, \mathbf{L}) \triangleq e^{-\frac{1}{2}\left(r + 2x^T s + x^T \mathbf{L} x\right)}. \tag{25}$$

Expressions for the sufficient statistics $w_{k|k}^{i,z_k}$, $\mu_{k|k}^{i,z_k}$, $\mathbf{P}_{k|k}^{i,z_k}$ and r_k^{ℓ,z_k}, s_k^{ℓ,z_k}, and $\mathbf{L}_k^{\ell,z_k}$ are detailed in Appendix A.

It is important to note that the expressions (23) and (24) can both be made exact, but this requires an exponential in data-length number of terms M_k^{f} and M_k^{c}, respectively. This is impractical in general, and therefore, the strategy used in Balenzuela et al. (2020) is to limit the maximum number of allowed terms by merging similar components to provide the approximations (23) and (24). Increasing this limit on the number of terms produces more accurate solutions, but at the expense of increased computational complexity.

In the following Lemma we provide an expression for the required joint distributions (18) based on these available statistics. Importantly, the number of components M_k used in the joint smoother description (19) is given by the product $M_k = M_k^{\mathrm{f}} M_{k+1}^{\mathrm{c}}$, so that no further reduction or approximation is introduced.

Lemma 4. *The joint distribution $p_{\theta_i}(\chi_k, \eta_k \mid \mathbf{y})$ can be expressed as the hybrid Gaussian mixture (19), where for $i = 1, \ldots, M_k^f$ and $\ell = 1, \ldots, M_{k+1}^c$, let $M_k = M_k^f M_{k+1}^c$ and define $j = M_k^f(\ell - 1) + i$, then compute*

$$w_k^{j,\eta_k} = \frac{e^{\frac{1}{2}\gamma_k^{j,\eta_k}}}{\sum_{z_k=1}^m \sum_{z_{k+1}=1}^m \sum_{h=1}^{M_k} e^{\frac{1}{2}\gamma_k^{j,\eta_k}}}, \tag{26a}$$

$$\mu_k^{j,\eta_k} = \mathbf{P}_k^{j,\eta_k}\left(\left[\boldsymbol{\Delta}_k^{i,z_k}\right]^{-1}\sigma_k^{i,z_k} - \begin{bmatrix}\mathbf{0}\\ s_{k+1}^{\ell,z_{k+1}}\end{bmatrix}\right), \tag{26b}$$

$$\mathbf{P}_k^{j,\eta_k} = \left(\left[\boldsymbol{\Delta}_k^{i,z_k}\right]^{-1} + \begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{L}_{k+1}^{\ell,z_{k+1}}\end{bmatrix}\right)^{-1}, \tag{26c}$$

where

$$\begin{aligned}\gamma_k^{j,\eta_k} = {} & \left[\mu_k^{j,\eta_k}\right]^T \left[\mathbf{P}_k^{j,\eta_k}\right]^{-1} \mu_k^{j,\eta_k} + \ln\left|\mathbf{P}_k^{j,\eta_k}\right| \\ & - \left[\sigma_k^{i,z_k}\right]^T \left[\boldsymbol{\Delta}_k^{i,z_k}\right]^{-1} \sigma_k^{i,z_k} - \ln\left|\boldsymbol{\Delta}_k^{i,z_k}\right| \\ & - r_{k+1}^{\ell,z_{k+1}} + 2\ln w_{k|k}^{i,z_k} + 2\ln \mathbf{T}_{z_{k+1},z_k},\end{aligned} \tag{26d}$$

$$\sigma_k^{i,z_k} = \begin{bmatrix}\mu_{k|k}^{i,z_k}\\ \bar{\mathbf{A}}_{z_k}\mu_{k|k}^{i,z_k} + \bar{\mathbf{B}}_{z_k}\bar{u}_k\end{bmatrix}, \tag{26e}$$

$$\boldsymbol{\Delta}_k^{i,z_k} = \begin{bmatrix}\mathbf{P}_{k|k}^{i,z_k} & \mathbf{P}_{k|k}^{i,z_k}\bar{\mathbf{A}}_{z_k}^T\\ \bar{\mathbf{A}}_{z_k}\mathbf{P}_{k|k}^{i,z_k} & \bar{\mathbf{A}}_{z_k}\mathbf{P}_{k|k}^{i,z_k}\bar{\mathbf{A}}_{z_k}^T + \bar{\mathbf{Q}}_{z_k}\end{bmatrix}. \tag{26f}$$

Proof. See Appendix E. □

With this Lemma in place, we can now provide expressions for (16) in terms of w_k^{j,η_k}, μ_k^{j,η_k} and $\mathbf{P}_k^{j,\eta_k}$ in the following Lemma.

Lemma 5. *The expectations in (16) can be stated using terms of w_k^{j,η_k}, μ_k^{j,η_k} and $\mathbf{P}_k^{j,\eta_k}$ from Lemma 4 via*

$$\alpha_{j,\ell} = \sum_{k=1}^{N}\sum_{i=1}^{M_k} w_k^{i,(j,\ell)}, \qquad \beta_\ell = \sum_{j=1}^{m}\alpha_{j,\ell}, \tag{27a}$$

$$\begin{aligned}\boldsymbol{\Phi}_\ell = {} & \sum_{k=1}^{N}\sum_{i=1}^{M_k}\sum_{j=1}^{m} w_k^{i,(j,\ell)} \\ & \times \begin{bmatrix} y_k y_k^T & y_k[\underline{\mu}_k^{i,(j,\ell)}]^T \\ \underline{\mu}_k^{i,(j,\ell)} y_k^T & \mathbf{G}_k^{i,(j,\ell)} + \underline{\mu}_k^{i,(j,\ell)}[\underline{\mu}_k^{i,(j,\ell)}]^T \end{bmatrix},\end{aligned} \tag{27b}$$

$$\begin{aligned}\boldsymbol{\Psi}_\ell = {} & \sum_{k=1}^{N}\sum_{i=1}^{M_k}\sum_{j=1}^{m} w_k^{i,(j,\ell)} \\ & \times \begin{bmatrix} y_k[\bar{\mu}_k^{i,(j,\ell)}]^T & y_k u_k^T \\ \mathbf{F}_k^{i,(j,\ell)} + \underline{\mu}_k^{i,(j,\ell)}[\bar{\mu}_k^{i,(j,\ell)}]^T & \underline{\mu}_k^{i,(j,\ell)} u_k^T \end{bmatrix},\end{aligned} \tag{27c}$$

$$\begin{aligned}\boldsymbol{\Sigma}_\ell = {} & \sum_{k=1}^{N}\sum_{i=1}^{M_k}\sum_{j=1}^{m} w_k^{i,(j,\ell)} \\ & \times \begin{bmatrix} \mathbf{E}_k^{i,(j,\ell)} + \bar{\mu}_k^{i,(j,\ell)}[\bar{\mu}_k^{i,(j,\ell)}]^T & \bar{\mu}_k^{i,(j,\ell)} u_k^T \\ u_k[\bar{\mu}_k^{i,(j,\ell)}]^T & u_k u_k^T \end{bmatrix},\end{aligned} \tag{27d}$$

where $\bar{\mu}_k^{i,(j,\ell)} \in \mathbb{R}^{n_x}$ and $\underline{\mu}_k^{i,(j,\ell)} \in \mathbb{R}^{n_x}$, $\mathbf{E}_k^{i,(j,\ell)} \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{F}_k^{i,(j,\ell)} \in \mathbb{R}^{n_x \times n_x}$ and $\mathbf{G}_k^{i,(j,\ell)} \in \mathbb{R}^{n_x \times n_x}$ are conformal partitions of $\mu_k^{i,(j,\ell)}$ and $\mathbf{P}_k^{i,(j,\ell)}$ according to

$$\mu_k^{i,(j,\ell)} = \begin{bmatrix}\bar{\mu}_k^{i,(j,\ell)}\\ \underline{\mu}_k^{i,(j,\ell)}\end{bmatrix}, \quad \mathbf{P}_k^{i,(j,\ell)} = \begin{bmatrix}\mathbf{E}_k^{i,(j,\ell)} & [\mathbf{F}_k^{i,(j,\ell)}]^T\\ \mathbf{F}_k^{i,(j,\ell)} & \mathbf{G}_k^{i,(j,\ell)}\end{bmatrix}. \tag{28}$$

Proof. See Appendix F. □

Using these resulting quantities from this Lemma, we can then directly apply Lemma 2 to compute the new parameter estimate, concluding an iteration of the proposed EM algorithm.

4.3. Final algorithm description

Under mild conditions, it is well known that the EM algorithm will converge to local maxima of the log-likelihood function (Balakrishnan, Hwang, Jang, & Tomlin, 2004). In the current application to jump Markov linear systems, it is also well known that EM may converge to a local maximum where some models are effectively disabled by near zeros in elements of the state transition matrix $\mathbf{T}$ (Logothetis & Krishnamurthy, 1999).

Avoiding such local maxima has been addressed in several ways. One approach uses a heuristic where zeros in the transition matrix are replaced with a small value ϵ to keep these hypotheses alive (Jilkov & Li, 2004). Another approach employs a restart method, where the algorithm is re-initialised using a random sample from a given distribution (Gil & Williams, 2009).

An alternative approach, employed for the proposed algorithm, is to hold the transition model constant until the log-likelihood increment falls below a user-defined tolerance. Thereafter, the transition matrix will be estimated. This approach tends to avoid transition probabilities rapidly converging to zero. In addition, the system matrices $\mathbf{A}_i$, $\mathbf{B}_i$, $\mathbf{C}_i$, $\mathbf{D}_i$ and $\mathbf{Q}_i$, $\mathbf{S}_i$, $\mathbf{R}_i$ for each model are initialised so that $\mathbf{A}_i \neq \mathbf{A}_j$ for $i \neq j$, and similarly for the remaining matrices.

This, in combination with the previous discussion, can be summarised by Algorithm 1.

Algorithm 1 Algorithm Overview

```
Require: Input data u, output data y, prior distribution $p(x_1, z_1)$, and initial parameter estimate $\theta_1$, a maximum number of iterations $i_{\max} > 0$, the tolerance $\epsilon_{\text{init}} > 0$ and a boolean flag $i_{\text{flag}} = 0$.
for $i = 1$ to $i_{\max}$ do
    Compute smoothed distributions via Lemma 4.
    Update $\boldsymbol{\Gamma}_\ell$ via (17b) and $\boldsymbol{\Pi}_\ell$ via (17c).
    Set $\bar{\theta} = \left\{\mathbf{T}, \{\boldsymbol{\Gamma}_i, \boldsymbol{\Pi}_i\}_{i=1}^m\right\}$
    if $|\ln p_{\theta_i}(\mathbf{y}) - \ln p_{\bar{\theta}}(\mathbf{y})| < \epsilon_{\text{init}}$ then
        Set $i_{\text{flag}} = 1$.
    end if
    if $i_{\text{flag}} = 1$ then
        Update $\mathbf{T}$ via (17a).
    end if
    Update $\theta_{i+1} = \left\{\mathbf{T}, \{\boldsymbol{\Gamma}_i, \boldsymbol{\Pi}_i\}_{i=1}^m\right\}$.
end for
```

5. A numerically stable implementation

In this section we detail a robust numerical implementation of the proposed algorithm, which is important when applying the algorithm to realistic problems as naïve implementation often results in loss of symmetry and/or positive semi-definiteness of covariance matrices. Therefore, the presentation here relies on using a so-called square-root implementation, where a general matrix $A = A^{T/2}A^{1/2}$, $A^{T/2} = (A^{1/2})^T$, and $A^{1/2}$ is a square-root factor.

Towards this end, the following Lemma details the robust equations responsible for generating the expectation sufficient statistics and calculating the optimal set of parameters.

Lemma 6. *The maximisation step can be achieved using*

$$\boldsymbol{\Pi}_\ell^{1/2} = \frac{1}{\sqrt{\beta_\ell}}\boldsymbol{\mathcal{R}}_{3,\ell}, \qquad \boldsymbol{\Gamma}_\ell = (\boldsymbol{\mathcal{R}}_{1,\ell}^{-1}\boldsymbol{\mathcal{R}}_{2,\ell})^T, \tag{29}$$

where $\boldsymbol{\mathcal{R}}$ is obtained from the QR-factorisation

$$\boldsymbol{\mathcal{Q}}_\ell \begin{bmatrix} \boldsymbol{\mathcal{R}}_{1,\ell} & \boldsymbol{\mathcal{R}}_{2,\ell} \\ \mathbf{0} & \boldsymbol{\mathcal{R}}_{3,\ell} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Lambda}_1^1(1, \ell) \\ \vdots \\ \boldsymbol{\Lambda}_N^{M^s}(m, \ell) \end{bmatrix}, \tag{30}$$

where $\boldsymbol{\Lambda}_k^i(j, \ell)$ are defined by

$$\boldsymbol{\Lambda}_k^i(j, \ell) \triangleq \sqrt{w_k^{i,(j,\ell)}} \begin{bmatrix} [\bar{\mu}_k^{i,(j,\ell)}]^T & u_k^T & y_k^T & [\underline{\mu}_k^{i,(j,\ell)}]^T \\ \mathbf{H}_{1,k}^{i,(j,\ell)} & \mathbf{0} & \mathbf{0} & \mathbf{H}_{2,k}^{i,(j,\ell)} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H}_{3,k}^{i,(j,\ell)} \end{bmatrix}, \tag{31}$$

with $\bar{\mu}_k^{i,(j,\ell)}$ and $\underline{\mu}_k^{i,(j,\ell)}$ defined in (28), and

$$\begin{bmatrix} \mathbf{H}_{1,k}^{i,(j,\ell)} & \mathbf{H}_{2,k}^{i,(j,\ell)} \\ \mathbf{0} & \mathbf{H}_{3,k}^{i,(j,\ell)} \end{bmatrix} = [\mathbf{P}_k^{i,(j,\ell)}]^{1/2}. \tag{32}$$

Proof. See Appendix G. □

The above Lemma relies on the availability of the square-root factor $[\mathbf{P}_k^{i,(j,\ell)}]^{1/2}$. To this end, the following Lemma details this and associated terms in the joint smoothing equations.

Lemma 7. *A square-root factor of the joint-smoothed covariance matrix can be obtained via*

$$\left[\mathbf{P}_k^{j,\eta_k}\right]^{1/2} = \boldsymbol{\mathcal{R}}_3, \tag{33a}$$

where $\boldsymbol{\mathcal{R}}_3$ is obtained from the QR-factorisation

$$\boldsymbol{\mathcal{Q}} \begin{bmatrix} \boldsymbol{\mathcal{R}}_1 & \boldsymbol{\mathcal{R}}_2 \\ \mathbf{0} & \boldsymbol{\mathcal{R}}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{J}_k^{j,\eta_k} & [\boldsymbol{\Delta}_k^{i,z_k}]^{1/2} \end{bmatrix}, \tag{33b}$$

$$[\boldsymbol{\Delta}_k^{i,z_k}]^{1/2} = \begin{bmatrix} [\mathbf{P}_{k|k}^{i,z_k}]^{1/2} & [\mathbf{P}_{k|k}^{i,z_k}]^{1/2}\bar{\mathbf{A}}_{z_k}^T \\ \mathbf{0} & \bar{\mathbf{Q}}_{z_k}^{1/2} \end{bmatrix}, \tag{33c}$$

$$\mathbf{J}_k^{j,\eta_k} = \begin{bmatrix} \mathbf{0} & [\mathbf{P}_{k|k}^{i,z_k}]^{1/2}\bar{\mathbf{A}}_{z_k}^T[\mathbf{L}_{k+1}^{\ell,z_{k+1}}]^{T/2} \\ \mathbf{0} & \bar{\mathbf{Q}}_{z_k}^{1/2}[\mathbf{L}_{k+1}^{\ell,z_{k+1}}]^{T/2} \end{bmatrix}. \tag{33d}$$

Proof. See Appendix H. □

Eqs. (33a) and (33c) replace (26c) and (26f) respectively, in Lemma 4, with the remainder of the equations being identical. Note that when computing (26d), the log-determinant terms should be calculated using $[\boldsymbol{\Delta}_k^{i,z_k}]^{1/2}$ and $[\mathbf{P}_k^{j,\eta_k}]^{1/2}$ directly. The forward filter square-root factors $[\mathbf{P}_{k|k}^{i,z_k}]^{1/2}$ and backward likelihood square-root factors $[\mathbf{L}_{k+1}^{\ell,z_{k+1}}]^{1/2}$ can be computed using standard square-root implementations.

6. Simulations

Here we provide simulation results from identifying various jump Markov linear systems to demonstrate the effectiveness of the proposed solution. These simulations were ran in a MATLAB 2018a environment using a 64 bit Windows 10 PC with a 2.9 GHz i7-7820HK processor and 32 GB of installed RAM.

6.1. Example 1: First order system

In this example we consider a JMLS system used in Barber (2006), Doucet et al. (2001), Helmick et al. (1995), Kim (1994) and Svensson et al. (2014) with the form

$$x_k = \mathbf{A}_{z_k}x_{k-1} + \mathbf{B}_{z_k}u_k + v_{k-1}, \tag{34a}$$

$$y_k = \mathbf{C}_{z_k}x_k + \mathbf{D}_{z_k}u_k + e_k, \tag{34b}$$

$$v_{k-1} \sim \mathcal{N}_{v_{k-1}}(0, \mathbf{Q}_{z_k}), \tag{34c}$$

$$e_k \sim \mathcal{N}_{e_k}(0, \mathbf{R}_{z_k}), \tag{34d}$$

where the proposed algorithm requires a minor modification for this model convention, as cross-covariance $\mathbf{S}_{z_k}$ is not allowed, and a different time index is attached to the discrete random variable and input in the process model.

In this first example, we compare the following state-of-the-art EM methods:

(1) Particle Smoothed Expectation Maximisation from Schön et al. (2011), denoted here as PSEM;
(2) Particle Stochastic Approximation Expectation–Maximisation from Lindsten (2013), denoted here as PSAEM;
(3) Rao-Blackwellized Particle Stochastic Approximation Expectation–Maximisation from Svensson et al. (2014), denoted here as RB-PSAEM;
(4) The approach proposed in this paper, denoted as Proposed EM.

A single state system was chosen in order to maintain practical computation times for the PSEM and PSAEM algorithms. The three-mode system used for this example was parameterised by

$$\mathbf{A}_1 = 0.9, \mathbf{B}_1 = 0.1, \mathbf{C}_1 = 0.9, \mathbf{D}_1 = 0,$$
$$\mathbf{Q}_1 = 0.045, \mathbf{R}_1 = 0.002,$$
$$\mathbf{A}_2 = 0.65, \mathbf{B}_2 = -0.32, \mathbf{C}_2 = 1, \mathbf{D}_2 = 0,$$
$$\mathbf{Q}_2 = 0.002, \mathbf{R}_2 = 0.005,$$

$$\mathbf{A}_3 = 0.51, \mathbf{B}_3 = 0.2, \mathbf{C}_3 = 1.2, \mathbf{D}_3 = 0,$$
$$\mathbf{Q}_3 = 0.02, \mathbf{R}_3 = 0.009,$$
$$\mathbf{T} = \begin{bmatrix} 0.6 & 0.35 & 0.1 \\ 0.3 & 0.6 & 0.4 \\ 0.1 & 0.05 & 0.5 \end{bmatrix}. \tag{35}$$

The system was simulated for $N = 7000$ time steps, with $u_k \sim \mathcal{N}_{u_k}(0, 1)$, and $x_0 = 0$. The initial guess given to the EM algorithms were

$$\hat{\mathbf{A}}_\ell = 1.2\mathbf{A}_\ell, \quad \hat{\mathbf{B}}_\ell = 0.8\mathbf{B}_\ell, \quad \hat{\mathbf{C}}_\ell = 0.5\mathbf{C}_\ell, \quad \hat{\mathbf{D}}_\ell = 0,$$
$$\hat{\mathbf{Q}}_\ell = 0.8\mathbf{Q}_\ell, \quad \hat{\mathbf{R}}_\ell = 1.5\mathbf{R}_\ell, \quad \hat{\mathbf{T}} = \frac{1}{3}\mathbf{1}_3, \tag{36}$$

where $\mathbf{1}_n$ denotes a $n \times n$ ones matrix, and a hat on the parameter denotes an initial guess. Additionally, the prior used for each method was $p(x_0, z_0) = \frac{1}{3}\mathcal{N}_{x_0}(0, 1)$.

As each of the methods use different approximations, with different approximation settings, the methods can only be compared fairly on computational time. Because of this, the above algorithms were each allowed 500 EM iterations, with the above settings chosen such that the EM algorithms took approximately 55 s to complete an iteration. In particular, computation time was balanced with the following choices:

- PSEM with 75 particles and 34 trajectories;
- PSAEM with 118 particles;
- RB-PSAEM with 40 trajectories;
- The proposed EM solution with 3 components per discrete mode with initial flag $i_{\text{flag}} = 1$ in Algorithm 1, in order to maintain parity across all compared methods.

Fig. 1 shows the approximate log-likelihood of the parameter estimate at each iteration for each of the methods. The log-likelihood was computed by running a JMLS filter with a KLR merging strategy, where each discrete mode within the filter was allowed to store a weighted Gaussian mixture with 5 components. This increase in components was used to improve the accuracy of the estimated log-likelihood.

The Bode magnitude response of each of the modes identified from each method is shown within Fig. 2. While strong conclusions cannot be made with one example, it is reassuring that the proposed method appears to provide the most accurate system estimate.

6.2. Example 2: Second order system

Encouraged by the results from Example 1, in this example we consider the case of identifying a system with a higher state dimension. The computational cost of using PSAEM and PSEM is prohibitive for this example, and they have therefore not been considered. Hence, this example will consider the RB-PSAEM and proposed method only. As with Example 1, the proposed algorithm required a minor modification.

The target system can be described by

$$\mathbf{T} = \begin{bmatrix} 0.6 & 0.5 \\ 0.4 & 0.5 \end{bmatrix}, \tag{37a}$$
$$H_1(q) = \frac{0.7406q + 0.004861}{q^2 + 0.6178q + 0.4385}, \tag{37b}$$
$$H_2(q) = \frac{-1.461q + 1.98}{q^2 - 1.189q + 0.2715}, \tag{37c}$$

where q is the forward shift operator that satisfies $qx_k = x_{k+1}$. This system was simulated for $N = 7000$ time steps, with $u_k \sim \mathcal{N}_{u_k}(0, 1)$, and $x_0 = 0$.

The prior used for both methods was $p(x_0, z_0) = \frac{1}{2}\mathcal{N}_{x_0}(0, \mathbf{I}_2)$, with an initial system guess of another randomly generated system and $\hat{\mathbf{T}} = \frac{1}{2}\mathbf{1}_2$. The RB-PSAEM algorithm was allowed 25 trajectories and the proposed algorithm was allowed 3 components per discrete state with initial flag $i_{\text{flag}} = 1$ in Algorithm 1, in order to maintain parity across all compared methods. These algorithms each ran for 500 EM iterations, with the RB-PSAEM method taking longer than the proposed method. The approximated log-likelihood of the estimated set of parameters per iteration is shown in Fig. 3 for both approaches.

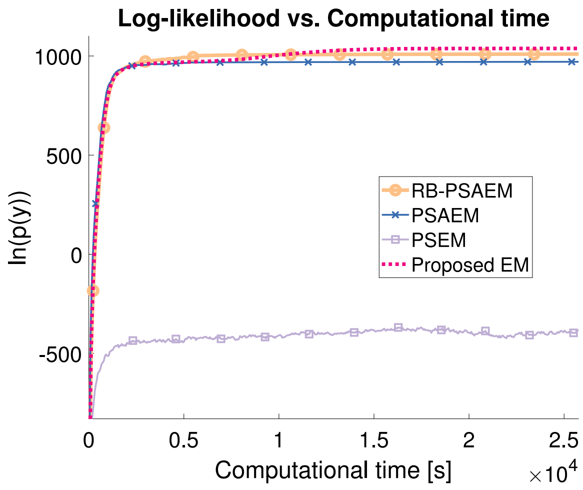


Fig. 1. Approximated log-likelihood of the estimated parameter set at each iteration of the EM algorithms for Example 1. The alternate RB-PSAEM method (solid yellow with circles), PSAEM method (solid blue with crosses), PSEM method (solid light purple with squares), can be compared to the proposed method, shown in dotted red.

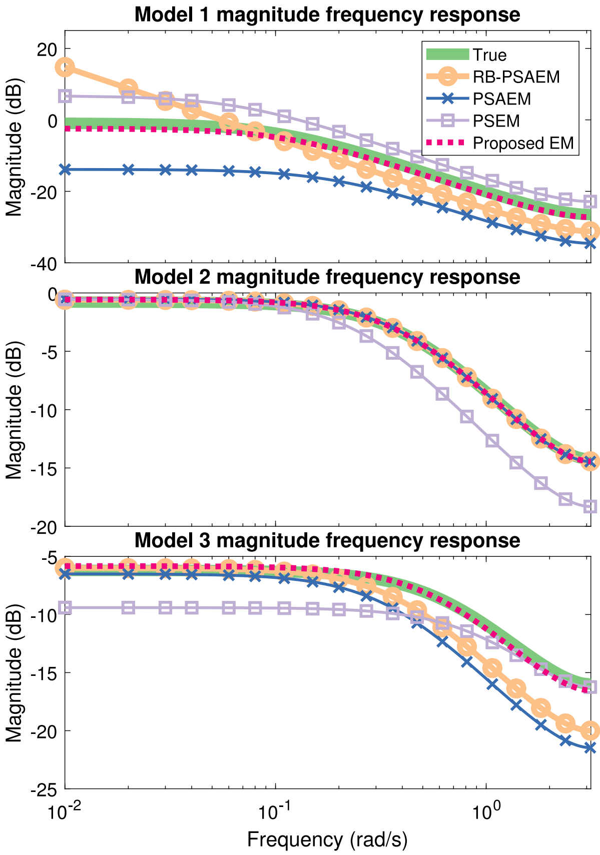


Fig. 2. Bode magnitude response for Example 1.

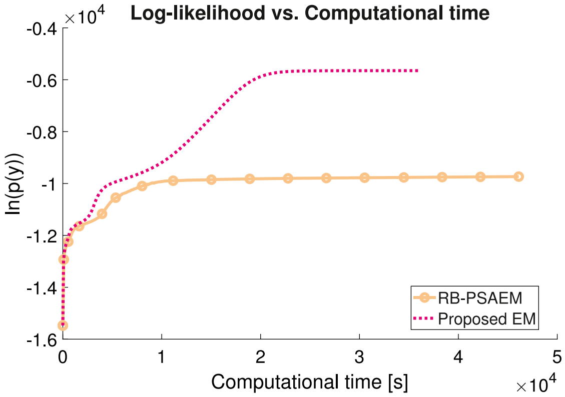


Fig. 3. Approximated log-likelihood of the estimated parameter set at each iteration of the EM algorithms for Example 2. The proposed method is shown in dotted red, where the alternate RB-PSAEM method is shown in solid yellow with circles.

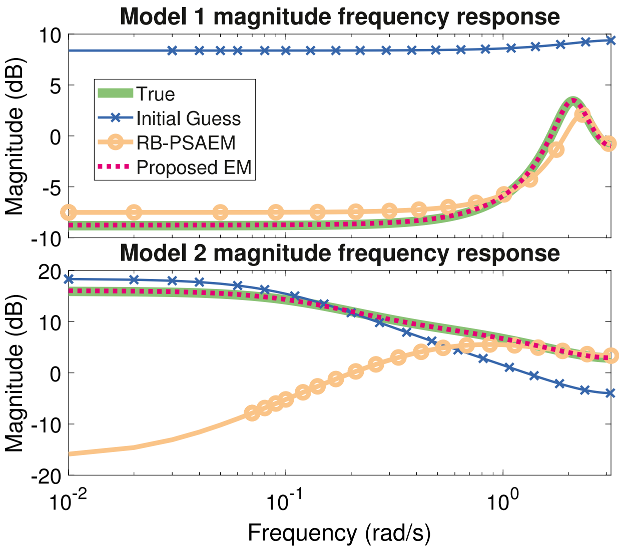


Fig. 4. Bode magnitude response for Example 2.

The frequency response of the identified system, true system and initial system guess are shown within the Bode magnitude plots in Fig. 4. Again, for this example, the proposed method appears to provide more accurate system estimates.

Following this encouraging result, the example was rerun 100 times, each with the initial parameters varied independently and randomly (using a uniform distribution) between ±20% of the true values. This produced Fig. 5, where the shaded regions show the variation of the likelihood for each method. As shown by this figure, the proposed method consistently produced a more likely set of parameters in less time. The proposed method also appears to have far less variation than the alternate stochastic RB-PSAEM method.

6.3. Example 3: Robustness to noise realisation

In this example, the two-mode two-state system from Section 6.2 was used to generate 25 different datasets, each with a length of $N = 250$ time steps, and each with different noise realisations. On each of these datasets, the proposed method and the alternate RB-PSAEM method was allowed 500 EM iterations, with both methods taking about 1400 s to complete. Fig. 6 shows the frequency response of the identified systems from each run, the initial guess common to each run, and the true system.

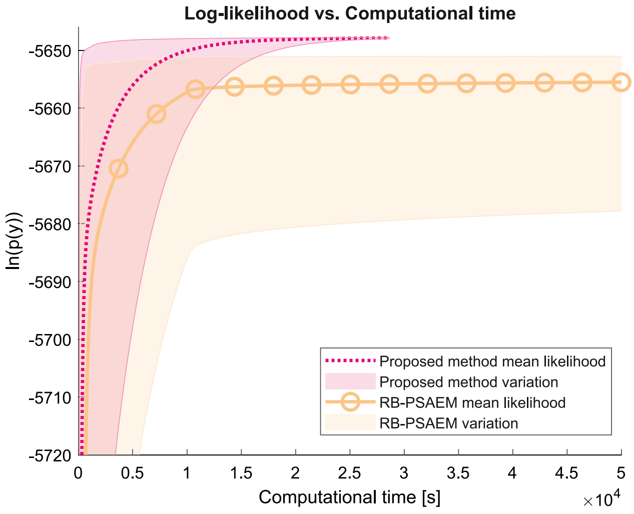


Fig. 5. Approximated log-likelihood for 100 runs of Example 2. The proposed method mean is shown in dotted red, where the alternate RB-PSAEM method mean is shown in solid yellow with circles. The shaded regions are bounded by the maximum and minimum likelihood for each method at that iteration.

In the majority of runs, the proposed method has accurately captured the system dynamics, whereas the RB-PSAEM method had not yet converged for many of these runs. In the few cases, where the proposed method performed poorly, the RB-PSAEM method also appears to have suffered, which is believed to be due to the short dataset and limited EM iterations allowed.

6.4. Example 4: Fifth order system

In this example, we consider the dynamic JMLS system operating according to (1). As alternative methods do not use this convention, and instead operate according to (34), this example cannot compare to them. The distinction in these conventions is both important and subtle. As the convention used in this example is suitable for robotic systems, this example tests the possible application of the proposed algorithm to a practical robotic system. To demonstrate this practicality, this final simulation considers a 5th order single input single output (SISO) system with two modes, both of which were randomly generated using the `drss` function in MATLAB.

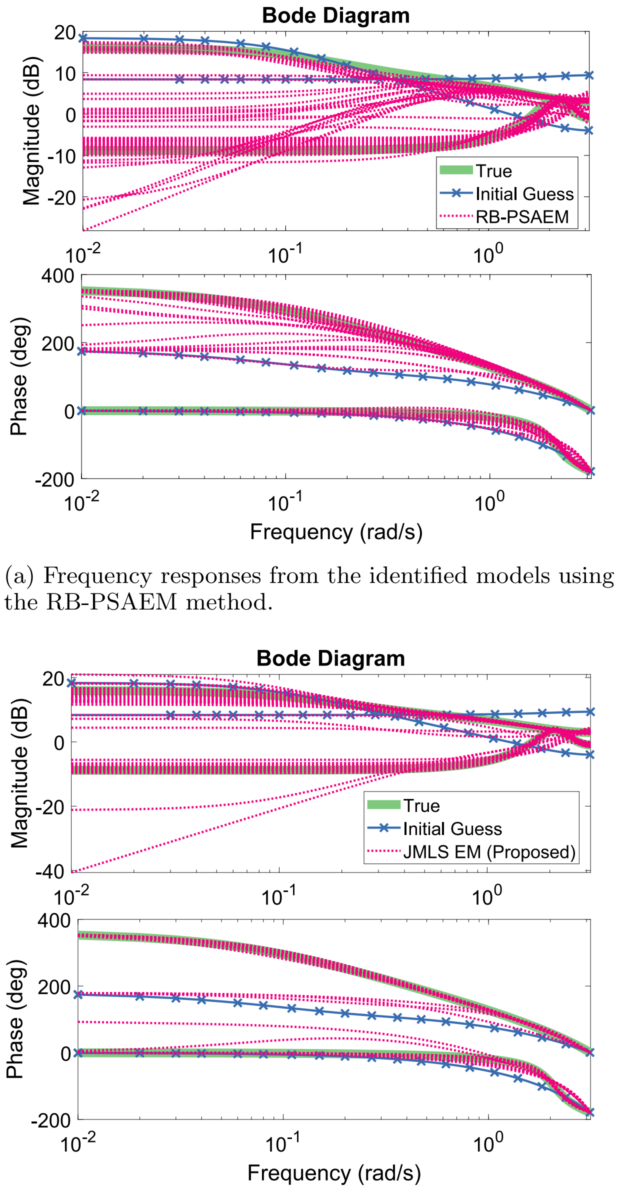
The system was simulated for $N = 2000$ time steps, with $x_0 = 0$ and input chosen as Gaussian white noise with unity variance. The proposed EM method was initialised by another set of random 5th order models, a prior of $p(x_0, z_0) = \frac{1}{2}\mathcal{N}(x_0|0, 3\mathbf{I}_5)$, and the initial state transition matrix $\hat{\mathbf{T}} = \frac{1}{2}\mathbf{1}_2$. The KL reduction stage within the proposed algorithm was allowed to keep six components per discrete state. The tolerance was selected as $\epsilon_{\text{init}} = 0.03$ in Algorithm 1. Finally, convergence to a solution took 1418 iterations of the proposed EM algorithm.

The frequency response from the identified modes, initial guess and true system are shown within Fig. 7. The true and identified transition matrix were

$$\mathbf{T}_{\text{true}} = \begin{bmatrix} 0.7 & 0.35 \\ 0.3 & 0.65 \end{bmatrix}, \text{ and } \mathbf{T}_{\text{identified}} = \begin{bmatrix} 0.6922 & 0.3814 \\ 0.3078 & 0.6186 \end{bmatrix},$$

respectively.

The proposed method has therefore identified the target system with reasonable accuracy, vastly improving on the initial system guess. While for many of the examples in this paper, we



(a) Frequency responses from the identified models using the RB-PSAEM method.

(b) Frequency responses from the identified models using the proposed method.

Fig. 6. Frequency response of the models from Example 3, where the truth (solid green) is shown along with the initial guess (solid blue with crosses), and the identified models (dotted red).

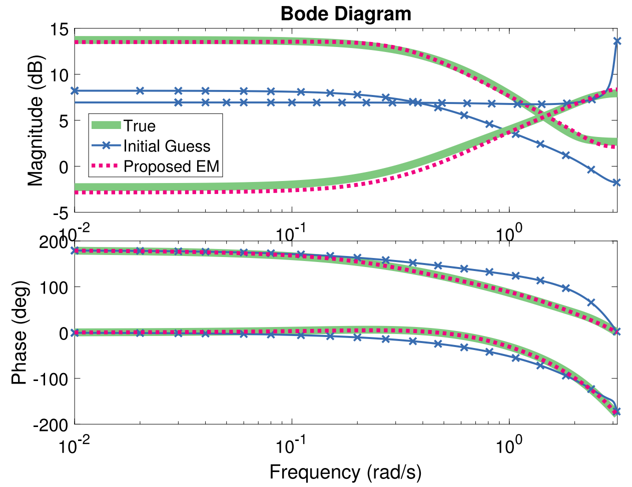


Fig. 7. Frequency response for the two 5th order dynamic models from Example 4. The true system is shown in solid green, where the blue trend with crosses marks the initial guess, and finally the identified system from the proposed method is shown in dotted magenta.

have been required to compare against the convention (34) to avoid modifying competing algorithms, this result indicates that the proposed algorithm can suitably identify systems which operated according to (1), which is more suitable for the operation of robotic systems.

7. Conclusion

Identification of jump Markov linear systems is a computationally expensive problem, scaling exponentially with the number of time steps. To combat this growth, the proposed EM approach in this paper uses approximations within the expectation step that rely on merging of Gaussian components to produce a manageable number of components. In principle, the proposed method can be made exact without relying on asymptotic arguments commonly required in SMC and stochastic approximation methods.

In developing the proposed EM approach, it was necessary to construct a new joint smoothing algorithm. We also provide numerically robust square-root implementations for computing the sufficient statistics of these smoothed distributions. This often overlooked detail is essential in providing practical algorithms that are computationally tractable for realistic problem dimensions.

After providing simulation examples demonstrating that the proposed EM approach performs well against state-of-the-art alternative methods, the proposed algorithm was then applied to an example that operated according to the dynamic convention (1) for JMLS systems. The results show the successful identification of the target system with reasonable accuracy. This convention is more suitable for robotic systems, and highlights the practicality of the proposed algorithm.

Acknowledgements

We would like to thank Dr. Andreas Svensson for kindly providing MATLAB code for the alternate methods.

Appendix A. Forward filter and backward information filter

Assuming the system (1)–(3) the initial distribution $p(x_1, z_1) = \sum_{i=1}^{M_1^{\mathrm{p}}} w_{1|0}^{i,z_1} \mathcal{N}_{x_1}\left(\mu_{1|0}^{i,z_1}, \mathbf{P}_{1|0}^{i,z_1}\right)$, then the sufficient statistics for the filtered distribution (23) for $k = 1, \dots, N$, $M_k^{\mathrm{f}} = M_k^{\mathrm{p}}$, $i = 1, \dots, M_k^{\mathrm{p}}$ and $z_k = 1, \dots, m$, are

$$w_{k|k}^{i,z_k} = \frac{\tilde{w}_{k|k}^{i,z_k}}{\sum_{z_k=1}^{m} \sum_{i=1}^{M_k^{\mathrm{p}}} \tilde{w}_{k|k}^{i,z_k}}, \tag{A.1a}$$

$$\tilde{w}_{k|k}^{i,z_k} = w_{k|k-1}^{i,z_k} \mathcal{N}_{y_k}\left(\mathbf{C}_{z_k}\mu_{k|k-1}^{i,z_k} + \mathbf{D}_{z_k}u_k, \, \boldsymbol{\Sigma}_k^{i,z_k}\right), \tag{A.1b}$$

$$\mu_{k|k}^{i,z_k} = \mu_{k|k-1}^{i,z_k} + \mathbf{K}_k^{i,z_k}[y_k - \mathbf{C}_{z_k}\mu_{k|k-1}^{i,z_k} - \mathbf{D}_{z_k}u_k], \tag{A.1c}$$

$$\mathbf{P}_{k|k}^{i,z_k} = \mathbf{P}_{k|k-1}^{i,z_k} - \mathbf{K}_k^{i,z_k}\mathbf{C}_{z_k}\mathbf{P}_{k|k-1}^{i,z_k}, \tag{A.1d}$$

$$\mathbf{K}_k^{i,z_k} = \mathbf{P}_{k|k-1}^{i,z_k}\mathbf{C}_{z_k}^T(\boldsymbol{\Sigma}_k^{i,z_k})^{-1}, \tag{A.1e}$$

$$\boldsymbol{\Sigma}_k^{i,z_k} = \mathbf{C}_{z_k}\mathbf{P}_{k|k-1}^{i,z_k}\mathbf{C}_{z_k}^T + \mathbf{R}_{z_k}. \tag{A.1f}$$

Then for $M^{\mathrm{p}}_{k+1} = m\,M^{\mathrm{f}}_k$ and for each $i = 1,\ldots,M^{\mathrm{f}}_k$, $\ell = 1,\ldots,m$ and $z_{k+1} = 1,\ldots,m$, define $j \triangleq M^{\mathrm{f}}_k(\ell-1)+i$, and compute

$$w^{j,z_{k+1}}_{k+1|k} = T_{z_{k+1},\ell}\, w^{i,\ell}_{k|k}, \tag{A.1g}$$

$$\mu^{j,z_{k+1}}_{k+1|k} = \mathbf{A}_\ell \mu^{i,\ell}_{k|k} + \mathbf{B}_\ell u_k, \tag{A.1h}$$

$$\mathbf{P}^{j,z_{k+1}}_{k+1|k} = \mathbf{A}_\ell \mathbf{P}^{i,\ell}_{k|k}\mathbf{A}^T_\ell + \mathbf{Q}_\ell. \tag{A.1i}$$

The statistics for the backward information filter (24) start with $M^{\mathrm{c}}_N = 1$ at time $k = N$ with

$$r^{1,z_N}_N = (\mathbf{D}_{z_N}u_N - y_N)^T\mathbf{R}^{-1}_{z_N}(\mathbf{D}_{z_N}u_N - y_N) + \ln|2\pi\mathbf{R}_{z_N}|,$$

$$s^{1,z_N}_N = \mathbf{C}^T_{z_N}\mathbf{R}^{-1}_{z_N}(\mathbf{D}_{z_N}u_N - y_N), \tag{A.2}$$

$$\mathbf{L}^{1,z_N}_N = \mathbf{C}^T_{z_N}\mathbf{R}^{-1}_{z_N}\mathbf{C}_{z_N}. \tag{A.3}$$

Then for $k = N-1,\ldots,1$, it follows that for $M^{\mathrm{b}}_k = mM^{\mathrm{c}}_{k+1}$ and for each $\ell = 1,\ldots,m$, $i = 1,\ldots,M^{\mathrm{b}}_{k+1}$ and $z_k = 1,\ldots,m$, define $j = M^{\mathrm{b}}_{k+1}(\ell-1)+i$ and compute

$$\bar{\mathbf{L}}^{j,z_k}_k = \mathbf{A}^T_{z_k}\,\boldsymbol{\Phi}^{j,z_k}_k\mathbf{A}_{z_k}, \tag{A.4a}$$

$$\bar{s}^{j,z_k}_k = \mathbf{A}^T_{z_k}\left[\boldsymbol{\Phi}^{j,z_k}_k\mathbf{B}_{z_k}u_k + (\boldsymbol{\Gamma}^{j,z_k}_k)^T s^{i,\ell}_{k+1}\right], \tag{A.4b}$$

$$\begin{aligned}\bar{r}^{j,z_k}_k &= r^{i,\ell}_{k+1} - \ln|\boldsymbol{\Gamma}^{j,z_k}_k| - 2\ln T_{\ell,z_k}\\ &\quad + \begin{bmatrix} s^{i,\ell}_{k+1}\\ \mathbf{B}_{z_k}u_k\end{bmatrix}^T \begin{bmatrix}\boldsymbol{\Psi}^{j,z_k}_k & \boldsymbol{\Gamma}^{j,z_k}_k\\ (\boldsymbol{\Gamma}^{j,z_k}_k)^T & \boldsymbol{\Phi}^{j,z_k}_k\end{bmatrix}\begin{bmatrix} s^{i,\ell}_{k+1}\\ \mathbf{B}_{z_k}u_k\end{bmatrix},\end{aligned} \tag{A.4c}$$

$$\boldsymbol{\Gamma}^{j,z_k}_k = \mathbf{I} - \mathbf{Q}_{z_k}\boldsymbol{\Phi}^{j,z_k}_k, \tag{A.4d}$$

$$\boldsymbol{\Psi}^{j,z_k}_k = \mathbf{Q}_{z_k}\boldsymbol{\Phi}^{j,z_k}_k\mathbf{Q}_{z_k} - \mathbf{Q}_{z_k}, \tag{A.4e}$$

$$\boldsymbol{\Phi}^{j,z_k}_k = \left(\mathbf{I} + \mathbf{L}^{i,\ell}_{k+1}\mathbf{Q}_{z_k}\right)^{-1}\mathbf{L}^{i,\ell}_{k+1}. \tag{A.4f}$$

Let $M^{\mathrm{c}}_k = M^{\mathrm{b}}_k$, then for each $j = 1,\ldots,M^{\mathrm{b}}_k$ and $z_k = 1,\ldots,m$ compute the following

$$\mathbf{L}^{j,z_k}_k = \bar{\mathbf{L}}^{j,z_k}_k + \mathbf{C}^T_{z_k}\mathbf{R}^{-1}_{z_k}\mathbf{C}_{z_k}, \tag{A.5a}$$

$$s^{j,z_k}_k = \bar{s}^{j,z_k}_k + \mathbf{C}^T_{z_k}\mathbf{R}^{-1}_{z_k}(\mathbf{D}_{z_k}u_k - y_k), \tag{A.5b}$$

$$\begin{aligned}r^{j,z_k}_k &= \bar{r}^{j,z_k}_k + \ln|2\pi\mathbf{R}_{z_k}|\\ &\quad + (\mathbf{D}_{z_k}u_k - y_k)^T\mathbf{R}^{-1}_{z_k}(\mathbf{D}_{z_k}u_k - y_k).\end{aligned} \tag{A.5c}$$

In the above, for the forward and backward filters, the number of terms grows exponentially since $M^f_{k+1} = mM^f_k$ and $M^c_{k-1} = mM^c_k$. Therefore, in order to limit this growth, we employ a merging strategy that replaces similar pairs of Gaussian components with a single Gaussian. The details of these steps are both important, and not easily replicated here, so we refer to Balenzuela et al. (2020) for further details.

Appendix B. Proof of Lemma 1

By employing the Kronecker Delta indicator function between two integers i, j defined as $\delta_{i,j} = 1$ if $i = j$ and $\delta_{i,j} = 0$ if $i \neq j$, then the lumped functions can then be expressed as

$$\mathcal{Q}_2(\theta,\theta_i) = \mathbb{E}_{\theta_i}\left\{\sum_{j=1}^m\sum_{\ell=1}^m\sum_{k=1}^N \delta_{j,z_{k+1}}\delta_{\ell,z_k}\ln\mathbf{T}_{j,\ell}\right\},$$

$$\mathcal{Q}_3(\theta,\theta_i) = \mathbb{E}_{\theta_i}\left\{\sum_{\ell=1}^m\sum_{k=1}^N \delta_{\ell,z_k}\ln\mathcal{N}_{\zeta_k}\left(\boldsymbol{\Gamma}_\ell\xi_k,\ \boldsymbol{\Pi}_\ell\right)\right\}.$$

Considering $\mathcal{Q}_2$, using linearity of the expectation operator and that $\ln\mathbf{T}_{j,\ell}$ is deterministic, reveals

$$\mathcal{Q}_2(\theta,\theta_i) = \sum_{j=1}^m\sum_{\ell=1}^m \underbrace{\mathbb{E}_{\theta_i}\left\{\sum_{k=1}^N\delta_{j,z_{k+1}}\delta_{\ell,z_k}\right\}}_{=\alpha_{j,\ell}}\ln\mathbf{T}_{j,\ell}.$$

Considering $\mathcal{Q}_3$, again by linearity and definition (2)

$$\begin{aligned}&\mathcal{Q}_3(\theta,\theta_i)\\ &= \mathbb{E}_{\theta_i}\left\{\sum_{\ell=1}^m\sum_{k=1}^N\delta_{\ell,z_k}\left(-\frac{n_x+n_y}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{\Pi}_\ell|\right.\right.\\ &\qquad\left.\left. - \frac{1}{2}\left[\zeta_k - \boldsymbol{\Gamma}_\ell\xi_k\right]^T\boldsymbol{\Pi}^{-1}_\ell\left[\zeta_k - \boldsymbol{\Gamma}_\ell\xi_k\right]\right)\right\}.\end{aligned}$$

Noting that the trace operator is invariant to cyclic permutations of its argument, then for any vectors v and w of the same dimension it follows that $v^Tw = \operatorname{tr}\left\{v^Tw\right\} = \operatorname{tr}\left\{vw^T\right\}$. Therefore

$$\begin{aligned}&\left[\zeta_k - \boldsymbol{\Gamma}_\ell\xi_k\right]^T\boldsymbol{\Pi}^{-1}_\ell\left[\zeta_k - \boldsymbol{\Gamma}_\ell\xi_k\right]\\ &= \operatorname{tr}\left\{\boldsymbol{\Pi}^{-1}_\ell\left[\zeta_k - \boldsymbol{\Gamma}_\ell\xi_k\right]\left[\zeta_k - \boldsymbol{\Gamma}_\ell\xi_k\right]^T\right\}\\ &= \operatorname{tr}\left\{\boldsymbol{\Pi}^{-1}_\ell\left[\zeta_k\zeta_k^T - \boldsymbol{\Gamma}_\ell\xi_k\zeta_k^T - \zeta_k\xi_k^T\boldsymbol{\Gamma}^T_\ell + \boldsymbol{\Gamma}_\ell\xi_k\xi_k^T\boldsymbol{\Gamma}^T_\ell\right]\right\}.\end{aligned}$$

Noting that $\boldsymbol{\Gamma}_\ell$ and $\boldsymbol{\Pi}_\ell$ are both deterministic, then using the linearity of $\mathbb{E}_{\theta_i}\{\cdot\}$ reveals

$$\begin{aligned}\mathcal{Q}_3(\theta,\theta_i) &= \underbrace{\mathbb{E}_{\theta_i}\left\{-\sum_{\ell=1}^m\sum_{k=1}^N\delta_{\ell,z_k}\frac{n_x+n_y}{2}\ln(2\pi)\right\}}_{=c}\\ &\quad - \frac{1}{2}\sum_{\ell=1}^m\underbrace{\mathbb{E}_{\theta_i}\left\{\sum_{k=1}^N\delta_{\ell,z_k}\right\}}_{=\beta_\ell}\ln|\boldsymbol{\Pi}_\ell|\\ &\quad - \frac{1}{2}\sum_{\ell=1}^m\operatorname{tr}\left\{\boldsymbol{\Pi}^{-1}_\ell\left[\underbrace{\sum_{k=1}^N\delta_{\ell,z_k}\zeta_k\zeta_k^T}_{=\boldsymbol{\Phi}_\ell} - \underbrace{\sum_{k=1}^N\delta_{\ell,z_k}\zeta_k\xi_k^T}_{=\boldsymbol{\Psi}_\ell}\boldsymbol{\Gamma}^T_\ell\right.\right.\\ &\qquad\left.\left.\boldsymbol{\Gamma}_\ell\underbrace{\sum_{k=1}^N\delta_{\ell,z_k}\xi_k\zeta_k^T}_{=\boldsymbol{\Psi}^T_\ell} + \boldsymbol{\Gamma}_\ell\underbrace{\sum_{k=1}^N\delta_{\ell,z_k}\xi_k\xi_k^T}_{=\boldsymbol{\Sigma}_\ell}\boldsymbol{\Gamma}^T_\ell\right]\right\}.\end{aligned}$$ □

Appendix C. Proof of Lemma 2

The expressions for $\boldsymbol{\Gamma}_\ell$ and $\boldsymbol{\Pi}_\ell$ follow the same arguments used in Lemma 3.3 in Gibson and Ninness (2005), and is well defined whenever $\boldsymbol{\Sigma}_\ell \succ 0$ as assumed. For $\mathbf{T}_{j,\ell}$, recall from (3) that it must satisfy the constraint that $\sum_{j=1}^M\mathbf{T}_{j,\ell} = 1$. Therefore, first order necessary conditions for optimality require that the Lagrangian (with Lagrange multiplier $\lambda \in \mathbb{R}$)

$$\mathscr{L}(\theta,\lambda) \triangleq \sum_{j=1}^m\sum_{\ell=1}^m\alpha_{j,\ell}\ln\mathbf{T}_{j,\ell} + \lambda\left(1 - \sum_{j=1}^m\mathbf{T}_{j,\ell}\right) \tag{C.1}$$

must satisfy

$$\frac{\partial\mathscr{L}(\theta,\lambda)}{\partial\mathbf{T}_{j,\ell}} = 0 \implies \mathbf{T}_{j,\ell} = \frac{\alpha_{j,\ell}}{\lambda}, \tag{C.2}$$

$$\frac{\partial\mathscr{L}(\theta,\lambda)}{\partial\lambda} = 0 \overset{\text{via (C.2)}}{\implies} \lambda = \sum_{j=1}\alpha_{j,\ell}. \tag{C.3}$$

Substitution of (C.3) into (C.2) reveals the expression for $\mathbf{T}_{j,\ell}$ in (17a) as claimed. This is well defined when $\sum_{j=1}^M\alpha_{j,\ell} > 0$ as assumed. □

Appendix D. Proof of Lemma 3

By definition (16b)

$$\alpha_{j,\ell} = \int \sum_{\mathbf{z}} \sum_{k=1}^{N} \delta_{j,z_{k+1}} \delta_{\ell,z_k}\, p_{\theta_i}(\mathbf{x}, \mathbf{z} \mid \mathbf{y})\mathrm{d}\mathbf{x}$$

$$= \sum_{k=1}^{N} p_{\theta_i}(z_{k+1} = j,\ z_k = \ell \mid \mathbf{y}), \tag{D.1}$$

where the second equality follows directly since $\mathbf{x}$ does not appear in the integrand and the remaining sum over $\mathbf{z}$ is non-zero only when $z_{k+1} = j$ and $z_k = \ell$, hence the expression (21b). The expression for β_ℓ in (21a) follows immediately from the definition (16a). Further, by definition (16e)

$$\boldsymbol{\Sigma}_\ell = \int \sum_{\mathbf{z}} \sum_{k=1}^{N} \delta_{\ell,z_k} \xi_k \xi_k^T p_{\theta_i}(\mathbf{x}, \mathbf{z} \mid \mathbf{y})\mathrm{d}\mathbf{x}$$

$$= \sum_{k=1}^{N} \int \sum_{\mathbf{z}} \delta_{\ell,z_k} \xi_k \xi_k^T p_{\theta_i}(x_k, \mathbf{z} \mid \mathbf{y})\mathrm{d}x_k$$

$$= \sum_{k=1}^{N} \int \xi_k \xi_k^T p_{\theta_i}(x_k, z_k = \ell \mid \mathbf{y})\mathrm{d}x_k, \tag{D.2}$$

where the second equality follows since only x_k appears in the integrand, and the third equality follows since the remaining sum over $\mathbf{z}$ is non-zero only when $z_k = \ell$, hence the expression (21e). Analogous arguments can be used to prove the expressions for (21c) and (21d). □

Appendix E. Proof of Lemma 4

Standard Two-Filter formulations (see e.g. Kailath et al., 2000) reveal that the required joint smoothed distribution can be expressed as

$$p(\chi_k, \eta_k|y_{1:N}) = \frac{p(y_{k+1:N}|\chi_k, \eta_k)p(z_{k+1}|z_k)p(x_{k+1}|x_k, z_k)p(x_k, z_k|y_{1:k})}{p(y_{k+1:N}|y_{1:k})}. \tag{E.1}$$

By definitions (1), (3) and (23), and conjugate products of Normal distributions, it follows immediately that

$$p(z_{k+1}|z_k)p(x_{k+1}|x_k, z_k)p(x_k, z_k|y_{1:k})$$

$$= \sum_{i=1}^{M_k^{\mathrm{f}}} \mathbf{T}_{\eta_k} w_{k|k}^{i,z_k}\, \mathcal{N}_{x_{k+1}}(\bar{\mathbf{A}}_{z_k} x_k + \bar{\mathbf{B}}_{z_k} u_k, \bar{\mathbf{Q}}_{z_k}) \cdot \mathcal{N}_{x_k}\left(\mu_{k|k}^{i,z_k}, \mathbf{P}_{k|k}^{i,z_k}\right)$$

$$= \sum_{i=1}^{M_k^{\mathrm{f}}} w_{k+1|k}^{i,\eta_k} \mathcal{N}_{\chi_k}\left(\mu_{k:k+1|k}^{i,z_k}, \boldsymbol{\Delta}_k^{i,z_k}\right), \tag{E.2}$$

where

$$w_{k+1|k}^{i,\eta_k} = \mathbf{T}_{\eta_k} w_{k|k}^{i,z_k},$$

$$\mu_{k:k+1|k}^{i,z_k} = \begin{bmatrix} \mu_{k|k}^{i,z_k} \\ \bar{\mathbf{A}}_{z_k} \mu_{k|k}^{i,z_k} + \bar{\mathbf{B}}_{z_k} u_k \end{bmatrix}, \tag{E.3}$$

$$\boldsymbol{\Delta}_k^{i,z_k} = \begin{bmatrix} \mathbf{P}_{k|k}^{i,z_k} & \mathbf{P}_{k|k}^{i,z_k} \bar{\mathbf{A}}_{z_k}^T \\ \bar{\mathbf{A}}_{z_k} \mathbf{P}_{k|k}^{i,z_k} & \bar{\mathbf{A}}_{z_k} \mathbf{P}_{k|k}^{i,z_k} \bar{\mathbf{A}}_{z_k}^T + \bar{\mathbf{Q}}_{z_k} \end{bmatrix}.$$

Notice that the remaining numerator component in (E.1) is not effected by x_k and z_k (by conditional independence) and so we can express the required term

$$p(y_{k+1:N} \mid \chi_k, \eta_k) = \sum_{\ell=1}^{M_{k+1}^{\mathrm{c}}} \mathcal{L}\left(\chi_k|\bar{r}_k^{\ell,z_{k+1}},\ \bar{s}_k^{\ell,z_{k+1}},\ \bar{\mathbf{L}}_k^{\ell,z_{k+1}}\right), \tag{E.4}$$

where $\bar{r}_k^{\ell,z_{k+1}} \triangleq r_{k+1}^{\ell,z_{k+1}}$ and

$$\bar{\mathbf{L}}_k^{\ell,z_{k+1}} \triangleq \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_{k+1}^{\ell,z_{k+1}} \end{bmatrix},\ \bar{s}_k^{\ell,z_{k+1}} \triangleq \begin{bmatrix} \mathbf{0} \\ s_{k+1}^{\ell,z_{k+1}} \end{bmatrix}. \tag{E.5}$$

We can now form the numerator of (E.1) as

$$p(y_{k+1:N}|y_{1:k})p(\chi_k, \eta_k|y_{1:N})$$

$$= \sum_{i=1}^{M_k^{\mathrm{f}}} w_{k+1|k}^{i,\eta_k} \mathcal{N}_{\chi_k}\left(\mu_{k:k+1|k}^{i,z_k}, \boldsymbol{\Delta}^{i,z_k}\right) \cdot \sum_{\ell=1}^{M_{k+1}^{\mathrm{c}}} \mathcal{L}_{\chi_k}\left(\bar{r}_k^{\ell,z_{k+1}},\ \bar{s}_k^{\ell,z_{k+1}},\ \bar{\mathbf{L}}_k^{\ell,z_{k+1}}\right). \tag{E.6}$$

By combining the sums into a single index j, Lemma 8 can be applied to give

$$p(y_{k+1:N}|y_{1:k})p(\chi_k, \eta_k|y_{1:N})$$

$$= \sum_{j=1}^{M_k} \tilde{w}_k^{j,\eta_k} \mathcal{N}_{\chi_k}\left(\mu_k^{j,\eta_k}, \mathbf{P}_k^{j,\eta_k}\right), \tag{E.7}$$

where $M_k = M_k^{\mathrm{f}} \cdot M_{k+1}^{\mathrm{c}}$, and $\mathbf{P}_k^{j,\eta_k}$, μ_k^{j,η_k} are provided by (26c) and (26b), respectively, and where

$$\tilde{w}_{k:k+1|N}^{j}(\eta_k) = e^{\frac{1}{2}\gamma^{j,\eta_k}}, \tag{E.8}$$

with γ^{j,η_k} defined in (26d). Finally the normalising constant is computed as

$$p(y_{k+1:N}|y_{1:k}) = \sum_{\eta_k} \int \sum_{j=1}^{M_k} \tilde{w}_k^{j,\eta_k} \mathcal{N}_{\chi_k}\left(\mu_k^{j,\eta_k}, \mathbf{P}_k^{j,\eta_k}\right) \mathrm{d}\chi_k$$

$$= \sum_{\eta_k} \sum_{j=1}^{M_k} \tilde{w}_k^{j,\eta_k}. \tag{E.9}$$

Substituting this into (E.7) and defining

$$w_k^{j,\eta_k} = \frac{\tilde{w}_k^{j,\eta_k}}{\sum_{\eta_k} \sum_{j=1}^{M_k} \tilde{w}_k^{j,\eta_k}} \tag{E.10}$$

yields all the required expressions for $p(\chi_k, \eta_k|y_{1:N})$. □

Lemma 8. *Let the likelihood $\mathcal{L}(x|r, s, \mathbf{L})$, Normal PDF $\mathcal{N}(x|\mu, \mathbf{P})$ and weight $w \geq 0$ be given. Then*

$$\bar{w}\mathcal{N}(x|\bar{\mu}, \bar{\mathbf{P}}) = w\mathcal{N}(x|\mu, \mathbf{P})\mathcal{L}(x|r, s, \mathbf{L}), \tag{E.11}$$

where $\bar{\mathbf{P}} = (\mathbf{P}^{-1} + \mathbf{L})^{-1}$, $\bar{\mu} = \bar{\mathbf{P}}(\mathbf{P}^{-1}\mu - s)$, $\bar{w} = \frac{w|2\pi\bar{\mathbf{P}}|^{\frac{1}{2}} e^{\frac{1}{2}\beta}}{|2\pi\mathbf{P}|^{\frac{1}{2}}}$, $\beta = \bar{\mu}^T\bar{\mathbf{P}}^{-1}\bar{\mu} - \mu^T\mathbf{P}^{-1}\mu - r$.

Proof. Expanding the exponents and collecting terms gives

$$w\mathcal{N}(x|\mu, \mathbf{P})\mathcal{L}(x|r, s, \mathbf{L}) = \frac{we^{-\frac{1}{2}(\mu^T\mathbf{P}^{-1}\mu + r)}}{|2\pi\mathbf{P}|^{\frac{1}{2}}} e^{-\frac{1}{2}f(x)}, \tag{E.12}$$

$$f(x) = x^T(\mathbf{P}^{-1} + \mathbf{L})x - 2x^T(\mathbf{P}^{-1}\mu - s). \tag{E.13}$$

Using the expressions for $\bar{\mathbf{P}}$ and $\bar{\mu}$ affords the simplification $f(x) = (x - \bar{\mu})^T\bar{\mathbf{P}}^{-1}(x - \bar{\mu}) - \bar{\mu}^T\bar{\mathbf{P}}^{-1}\bar{\mu}$. Therefore, using the expression for β reveals

$$w\mathcal{N}(x|\mu, \mathbf{P})\mathcal{L}(x|r, s, \mathbf{L}) = \underbrace{\frac{w|2\pi\bar{\mathbf{P}}|^{\frac{1}{2}}e^{\frac{1}{2}\beta}}{|2\pi\mathbf{P}|^{\frac{1}{2}}}}_{\bar{w}} \cdot \underbrace{|2\pi\bar{\mathbf{P}}|^{-\frac{1}{2}}e^{-\frac{1}{2}(x-\bar{\mu})^T\bar{\mathbf{P}}^{-1}(x-\bar{\mu})}}_{\mathcal{N}(x|\bar{\mu},\bar{\mathbf{P}})}. \quad \square$$

Appendix F. Proof of Lemma 5

Substituting

$$p_{\theta_i}(\chi_k, \eta_k|\mathbf{y}) = \sum_{j=1}^{M_k^S} w_k^{j,\eta_k}\mathcal{N}_{\chi_k}(\mu_k^{j,\eta_k}, \mathbf{P}_k^{j,\eta_k}), \tag{F.1}$$

into the expressions (21a)–(21e), and employing standard results for expectations of $\xi_k\xi_k^T$, $\zeta_k\xi_k^T$ and $\zeta_k\zeta_k^T$ (see Lemma 3.2 in Gibson & Ninness, 2005) provides the expressions in Lemma 5 (noting the partition used in (28)). □

Appendix G. Proof of Lemma 6

It can be shown by use of Lemma 5 and expansion of $\boldsymbol{\Lambda}_k^i(j, \ell)^T \boldsymbol{\Lambda}_k^i(j, \ell)$, defined in (31) that

$$\begin{bmatrix} \boldsymbol{\Sigma}_\ell & \boldsymbol{\Psi}_\ell^T \\ \boldsymbol{\Psi}_\ell & \boldsymbol{\Phi}_\ell \end{bmatrix} = \sum_{k=1}^{N}\sum_{i=1}^{M_k}\sum_{j=1}^{m} \boldsymbol{\Lambda}_k^i(j, \ell)^T \boldsymbol{\Lambda}_k^i(j, \ell)$$
$$= \begin{bmatrix} \boldsymbol{\mathcal{R}}_{1,\ell}^T\boldsymbol{\mathcal{R}}_{1,\ell} & \boldsymbol{\mathcal{R}}_{1,\ell}^T\boldsymbol{\mathcal{R}}_{2,\ell} \\ \boldsymbol{\mathcal{R}}_{2,\ell}^T\boldsymbol{\mathcal{R}}_{1,\ell} & \boldsymbol{\mathcal{R}}_{2,\ell}^T\boldsymbol{\mathcal{R}}_{2,\ell} + \boldsymbol{\mathcal{R}}_{3,\ell}^T\boldsymbol{\mathcal{R}}_{3,\ell} \end{bmatrix}, \tag{G.1}$$

and therefore $\boldsymbol{\Sigma}_\ell = \boldsymbol{\mathcal{R}}_{1,\ell}^T\boldsymbol{\mathcal{R}}_{1,\ell}$, and $\boldsymbol{\Psi}_\ell = \boldsymbol{\mathcal{R}}_{2,\ell}^T\boldsymbol{\mathcal{R}}_{1,\ell}$, and $\boldsymbol{\Phi}_\ell = \boldsymbol{\mathcal{R}}_{2,\ell}^T\boldsymbol{\mathcal{R}}_{2,\ell} + \boldsymbol{\mathcal{R}}_{3,\ell}^T\boldsymbol{\mathcal{R}}_{3,\ell}$. Therefore, as claimed

$$\boldsymbol{\Gamma}_\ell = \boldsymbol{\Psi}_\ell\boldsymbol{\Sigma}_\ell^{-1} = (\boldsymbol{\mathcal{R}}_{1,\ell}^{-1}\boldsymbol{\mathcal{R}}_{2,\ell})^T. \tag{G.2}$$

Further,

$$\begin{aligned} \beta_\ell\boldsymbol{\Pi}_\ell &= \boldsymbol{\Phi}_\ell - \boldsymbol{\Psi}_\ell\boldsymbol{\Sigma}_\ell^{-1}\boldsymbol{\Psi}_\ell^T \\ &= \boldsymbol{\mathcal{R}}_{2,\ell}^T\boldsymbol{\mathcal{R}}_{2,\ell} + \boldsymbol{\mathcal{R}}_{3,\ell}^T\boldsymbol{\mathcal{R}}_{3,\ell} - \boldsymbol{\mathcal{R}}_{2,\ell}^T\boldsymbol{\mathcal{R}}_{2,\ell} \\ &= \boldsymbol{\mathcal{R}}_{3,\ell}^T\boldsymbol{\mathcal{R}}_{3,\ell}. \end{aligned} \tag{G.3}$$

Therefore $\boldsymbol{\Pi}_\ell^{1/2} = \frac{1}{\sqrt{\beta_\ell}}\boldsymbol{\mathcal{R}}_{3,\ell}$ as claimed. □

Appendix H. Proof of Lemma 7

From the proof of Lemma 4, we know the joint-covariance is constructed using

$$\mathbf{P} = \left(\boldsymbol{\Delta}^{-1} + \bar{\mathbf{L}}\right)^{-1}, \tag{H.1}$$

where for readability we omit the indices and function arguments. Continuing by applying the Woodbury matrix identity yields

$$\mathbf{P} = \boldsymbol{\Delta} - \boldsymbol{\Delta}\mathbf{L}^{1/2}(\mathbf{I} + \mathbf{L}^{T/2}\boldsymbol{\Delta}\mathbf{L}^{T/2})^{-1}\mathbf{L}^{1/2}\boldsymbol{\Delta}. \tag{H.2}$$

Importantly, a square-root factor of $\boldsymbol{\Delta}$ is given by,

$$\boldsymbol{\Delta}^{1/2} = \begin{bmatrix} \mathbf{P}_{k|k}^{1/2} & \mathbf{P}_{k|k}^{1/2}\mathbf{A}^T \\ \mathbf{0} & \mathbf{Q}^{1/2} \end{bmatrix}, \tag{H.3}$$

which can be proved by expansion. The QR decomposition

$$\mathcal{Q}\begin{bmatrix} \boldsymbol{\mathcal{R}}_1 & \boldsymbol{\mathcal{R}}_2 \\ 0 & \boldsymbol{\mathcal{R}}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \boldsymbol{\Delta}^{1/2}\mathbf{L}^{T/2} & \boldsymbol{\Delta}^{1/2} \end{bmatrix}, \tag{H.4}$$

which affords the relations

$$\boldsymbol{\mathcal{R}}_1^T\boldsymbol{\mathcal{R}}_1 = \mathbf{I} + \mathbf{L}^{1/2}\boldsymbol{\Delta}\mathbf{L}^{T/2}, \tag{H.5}$$

and

$$\begin{aligned} \boldsymbol{\mathcal{R}}_3^T\boldsymbol{\mathcal{R}}_3 &= \boldsymbol{\Delta} - \boldsymbol{\Delta}\mathbf{L}^{T/2}(\boldsymbol{\mathcal{R}}_1^T\boldsymbol{\mathcal{R}}_1)^{-1}\mathbf{L}^{1/2}\boldsymbol{\Delta} \\ &= \boldsymbol{\Delta} - \boldsymbol{\Delta}\mathbf{L}^{1/2}(\mathbf{I} + \mathbf{L}^{T/2}\boldsymbol{\Delta}\mathbf{L}^{T/2})^{-1}\mathbf{L}^{1/2}\boldsymbol{\Delta} \\ &= \mathbf{P}. \end{aligned} \tag{H.6}$$

Hence $\boldsymbol{\mathcal{R}}_3$ is a square-root factor for $\mathbf{P}$ as required. □

References

Alspach, Daniel, & Sorenson, Harold (1972). Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Transactions on Automatic Control*, *17*(4), 439–448.

Anderson, Brian D. O., & Moore, John B. (2005). *Optimal filtering*. Courier Corporation.

Andrieu, Christophe, Moulines, Éric, & Priouret, Pierre (2005). Stability of stochastic approximation under verifiable conditions. *SIAM Journal on Control and Optimization*, *44*(1), 283–312.

Ashley, Trevor T., & Andersson, Sean B. (2014). A sequential monte carlo framework for the system identification of jump Markov state space models. In *2014 American control conference* (pp. 1144–1149). IEEE.

Balakrishnan, Hamsa, Hwang, Inseok, Jang, Jung Soon, & Tomlin, Claire J (2004). Inference methods for autonomous stochastic linear hybrid systems. In *International workshop on hybrid systems: computation and control* (pp. 64–79). Springer.

Balenzuela, Mark P., Wills, Adrian G., Renton, Christopher, & Ninness, Brett (2020). A new smoothing algorithm for jump Markov linear systems. To appear in *Automatica*. arXiv preprint arXiv:2004.08561.

Barber, David (2006). Expectation correction for smoothed inference in switching linear dynamical systems. *Journal of Machine Learning Research*, *7*(Nov), 2515–2540.

Blackmore, Lars, Gil, Stephanie, Chung, Seung, & Williams, Brian (2007). Model learning for switching linear systems with autonomous mode transitions. In *2007 46th IEEE conference on decision and control* (pp. 4648–4655). IEEE.

Chen, Dulin, Bako, Laurent, & Lecoeuche, Stéphane (2011). A recursive sparse learning method: Application to jump Markov linear systems. *IFAC Proceedings Volumes*, *44*(1), 3198–3203.

Delyon, Bernard, Lavielle, Marc, Moulines, Eric, et al. (1999). Convergence of a stochastic approximation version of the EM algorithm. *The Annals of Statistics*, *27*(1), 94–128.

Dempster, Arthur P., Laird, Nan M., & Rubin, Donald B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, 1–38.

Doucet, Arnaud, Godsill, Simon, & Andrieu, Christophe (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, *10*(3), 197–208.

Doucet, Arnaud, Gordon, Neil J., & Krishnamurthy, Vikram (2001). Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, *49*(3), 613–624.

Ghahramani, Zoubin, & Hinton, Geoffrey E. (2000). Variational learning for switching state-space models. *Neural Computation*, *12*(4), 831–864.

Gibson, Stuart, & Ninness, Brett (2005). Robust maximum-likelihood estimation of multivariable dynamic systems. *Automatica*, *41*(10), 1667–1682.

Gil, Stephanie, & Williams, Brian (2009). Beyond local optimality: An improved approach to hybrid model learning. In *Proceedings of the 48th IEEE conference on decision and control held jointly with 2009 28th Chinese control conference* (pp. 3938–3945). IEEE.

Hashimoto, Masafumi, Kawashima, Hiroyuki, Nakagami, Takashi, & Oba, Fuminori (2001). Sensor fault detection and identification in dead-reckoning system of mobile robot: Interacting multiple model approach. In *Intelligent robots and systems, 2001. Proceedings. 2001 IEEE/RSJ international conference on*: *Vol. 3*, (pp. 1321–1326). IEEE.

Helmick, Ronald E., Blair, W. Dale, & Hoffman, Scott A. (1995). Fixed-interval smoothing for Markovian switching systems. *IEEE Transactions on Information Theory*, *41*(6), 1845–1855.

Jilkov, Vesselin P., & Li, X. Rong (2004). Online Bayesian estimation of transition probabilities for Markovian jump systems. *IEEE Transactions on Signal Processing*, *52*(6), 1620–1630.

Kailath, Thomas, Sayed, Ali H., & Hassibi, Babak (2000). *Linear estimation*: *EPFL-BOOK-233814*, Prentice Hall.

Kalman, Rudolph Emil (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, *82*(1), 35–45.

Kim, Chang-Jin (1994). Dynamic linear models with Markov-switching. *Journal of Econometrics*, *60*(1–2), 1–22.

Kullback, S. (1959). *Information theory and statistics*. New York: Wiley.

Lindsten, Fredrik (2013). An efficient stochastic approximation EM algorithm using conditional particle filters. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 6274–6278). IEEE.

Logothetis, Andrew, & Krishnamurthy, Vikram (1999). Expectation maximization algorithms for MAP estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, *47*(8), 2139–2156.

Mazor, Efim, Averbuch, Amir, Bar-Shalom, Yakov, & Dayan, Joshua (1998). Interacting multiple model methods in target tracking: A survey. *IEEE Transactions on Aerospace and Electronic Systems*, *34*(1), 103–123.

Paoletti, Simone, Juloski, Aleksandar Lj, Ferrari-Trecate, Giancarlo, & Vidal, René (2007). Identification of hybrid systems a tutorial. *European Journal of Control*, *13*(2–3), 242–260.

Rauch, Herbert E., Striebel, C. T., & Tung, F. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, *3*(8), 1445–1450.

Schön, Thomas B., Wills, Adrian, & Ninness, Brett (2011). System identification of nonlinear state-space models. *Automatica*, *47*(1), 39–49.

Svensson, Andreas, Schön, Thomas B., & Lindsten, Fredrik (2014). Identification of jump Markov linear models using particle filters. In *53rd IEEE conference on decision and control* (pp. 6504–6509). IEEE.

Yildirim, Sinan, Singh, Sumeetpal S., & Doucet, Arnaud (2013). An online expectation–maximization algorithm for changepoint models. *Journal of Computational and Graphical Statistics*, *22*(4), 906–926.

Mark P. Balenzuela began showing interest in Mechatronic systems from a very young age. In 2016, he graduated with Bachelor degrees in Mechanical and Mechatronics Engineering from the University of Newcastle, Callaghan, NSW, Australia, and was awarded first class honours and a faculty medal for academic excellence. Afterwards, he began a Ph.D. candidature with the Mechatronics Engineering department at the same university, and was awarded a Ph.D. in 2021. His research interests include system estimation and identification, fault diagnosis, machine learning, and automation. Mark is now working as part of a cutting edge R&D team in industry.

Adrian G. Wills received the B.E. and Ph.D. degrees from The University of Newcastle, Australia, Callaghan, NSW, Australia, in May 1999 and May 2003, respectively. Since then, he has held postdoctoral research positions with Newcastle and spent three years working in industry. His research has been in the areas of system identification and estimation. In July 2015, he returned to The University of Newcastle to lead the mechatronics engineering program.

Christopher Renton received the B.E. degree in mechatronics and the Ph.D. degree from The University of Newcastle, Callaghan, NSW, Australia, in 2008 and 2014, respectively. Since then, he gained several years of industrial experience as an R&D Engineer in the automation industry and as the CTO of a technology startup in the area of autonomous aerial vehicles. His research has been in the areas of Bayesian machine vision, unified estimation and control, and infinite dimensional mapping and planning.

Brett Ninness was Pro Vice-Chancellor of the Faculty of Engineering and Built Environment at the University of Newcastle Australia 2013–2020 and now is a Professor in the Mechatronics group of that University. His research interests are in the areas of dynamic system modelling, system identification, and stochastic signal processing, in which he has authored over 100 papers.

He has served on the editorial boards of Automatica, IEEE Transactions on Automatic Control and as Editor in Chief for IET Control Theory and Applications.

Professor Ninness was a member of the Australian Research Council College of Experts and has served as chair of international committees, including the International Federation of Automatic Control (IFAC) Technical Committee on Modelling, Identification and Signal Processing, and the Institute of Electrical and Electronic Engineers (IEEE) Technical Committee on System Identification and Adaptive Control.