Automatica 140 (2022) 110218

Contents lists available at ScienceDirect

Automatica

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

Brief paper

A new smoothing algorithm for jump Markov linear systems☆

Mark Peter Balenzuela *, Adrian G. Wills, Christopher Renton, Brett Ninness

Faculty of Engineering and Built Environment, The University of Newcastle, Callaghan, NSW 2308, Australia

A R T I C L E I N F O

Article history:
Received 18 October 2019
Received in revised form 24 April 2021
Accepted 23 July 2021
Available online 14 March 2022

Keywords:
Bayesian estimation
Smoothing
Jump Markov linear systems (JMLS)
Switched linear dynamical systems (SLDS)
Interacting multiple models (IMM)

A B S T R A C T

This paper presents a method for calculating the smoothed state distribution for Jump Markov Linear Systems. More specifically, the paper details a novel two-filter smoother that provides closed-form expressions for the smoothed hybrid state distribution. This distribution can be expressed as a Gaussian mixture with a known, but exponentially increasing, number of Gaussian components as the time index increases. This is accompanied by exponential growth in memory and computational requirements, which rapidly becomes intractable. To ameliorate this, we limit the number of allowed mixture terms by employing a Gaussian likelihood mixture reduction strategy, which results in a computationally tractable, but approximate smoothed distribution. The approximation error can be balanced against computational complexity in order to provide an accurate and practical smoothing algorithm that compares favourably to existing state-of-the-art approaches.

1. Introduction

Abrupt and unexpected changes in system behaviour can often lead to highly undesirable outcomes. For example, mechanical failure of aircraft flight-control surfaces can have devastating consequences if not detected and compensated for Costa, Fragoso, and Marques (2006). This particular example of change is caused by a system failure or fault, but more generally there are many other possible causes of abrupt change including environmental influences, modified operating conditions, and reconfiguration of system networks. These types of changes and their potential impact on system performance have been observed in a wide range of applications including econometrics (Kim, 1994), telecommunications (Logothetis & Krishnamurthy, 1999), target tracking (Mazor, Averbuch, Bar-Shalom, & Dayan, 1998), and fault detection and isolation (FDI) (Hashimoto, Kawashima, Nakagami, & Oba, 2001), to name but a few.

Mitigating the potential impact of these abrupt changes relies on timely and reliable detection of such events, which is the primary aim of this paper. From a control perspective, modelling the possibility of these events within a dynamic system structure has received significant attention for several decades now (Costa et al., 2006). System models that cater for these abrupt changes are often afforded the epithets of either *jump* or *switched* to indicate that the system can rapidly change behaviour. Within this broad class of systems are the particular class of interest in this paper, namely discrete-time jump Markov linear systems (JMLS), or as sometimes called, switched linear dynamical systems (SLDS) (Barber, 2006). The primary reason for restricting our attention to this subclass of systems is that they are relatively simple, and yet offer enough flexibility to model the types of real-world phenomena mentioned above.

In order to make this discussion more concrete, the JMLS class we are concerned with in this paper can be expressed as

$$x_{k+1} = \mathbf{A}_k(z_k)x_k + \mathbf{B}_k(z_k)u_k + v_k, \tag{1a}$$

$$y_k = \mathbf{C}_k(z_k)x_k + \mathbf{D}_k(z_k)u_k + e_k, \tag{1b}$$

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, $z_k \in \{1, \dots, m_k\}$ is a discrete random variable that is often called the *model index*, and the noise terms v_k and e_k originate from the Gaussian white noise process

$$v_k \sim \mathcal{N}(0, \mathbf{Q}_k(z_k)), \quad e_k \sim \mathcal{N}(0, \mathbf{R}_k(z_k)). \tag{1c}$$

We have adopted the standard notation $\mathcal{N}(x|\mu, \mathbf{P})$ to denote the multivariate Normal distribution, where x is the variable (omitted in some cases for brevity), μ is the mean, and $\mathbf{P}$ is the covariance of the distribution. The system matrices $\{\mathbf{A}_k, \mathbf{B}_k, \mathbf{C}_k, \mathbf{D}_k, \mathbf{Q}_k, \mathbf{R}_k\}$ are allowed to randomly jump or switch values for each time-index k as a function of the model index z_k, and it is assumed that both the covariance matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$ are symmetric positive definite. The switch event is captured by allowing z_k to transition to z_{k+1} stochastically with the probability of transitioning from

☆ 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.2022.110218
0005-1098/

the jth model at time-index k to the ith model at time-index $k+1$ is expressed as

$$\mathbb{P}(z_{k+1} = i | z_k = j) = T_k(i|j), \tag{2a}$$

$$0 \le T_k(i|j) \le 1 \quad \forall i, j, \qquad \sum_{i=1}^{m_{k+1}} T_k(i|j) = 1 \quad \forall j. \tag{2b}$$

This transition probability mass function (PMF) encodes the type of stochastic switching exhibited by the system. With this as background, we introduce the main focus of the paper.

Problem statement: This paper is directed towards calculating the following joint continuous–discrete smoothed distribution

$$p(x_k, z_k | y_{1:N}), \tag{3}$$

based on $N > 0$ measurements of the input $u_k \in \mathbb{R}^{n_u}$ and output $y_k \in \mathbb{R}^{n_y}$, which are collected to form

$$y_{1:N} = \{y_1, \dots, y_N\}, \quad u_{1:N} = \{u_1, \dots, u_N\} \tag{4}$$

The inherent difficulty in smoothing for the JMLS class may be elucidated by previewing the closed-form expression for the smoothed distribution $p(x_k, z_k \mid y_{1:N})$ (see Section 2), which can be expressed as an indexed Gaussian mixture distribution (see e.g. Everitt, 2014) as follows

$$p(x_k, z_k | y_{1:N}) = \sum_{j=1}^{M_k^s} w_{k|N}^j(z_k) \mathcal{N}\left(x_k | \mu_{k|N}^j(z_k), \mathbf{P}_{k|N}^j(z_k)\right),$$

where $w_{k|N}^j(z_k)$ are non-negative mixture weights. Importantly, the number of components M_k^s is given by $M_k^s = \prod_{k=1}^N m_k$. Therefore, the number of terms that must be computed becomes impractical, even for small data lengths N and a modest number of models m_k, which is well known (Barber, 2006; Bergman & Doucet, 2000; Blom & Bar-Shalom, 1988; Helmick, Blair, & Hoffman, 1995; Kim, 1994).

Perhaps not surprisingly then, the main approaches to solving this smoothing problem employ some form of approximation. One broad category of methods use linear estimation theory while maintaining a practical number of components M_k^s in the mixture. The latter is achieved by using mixture reduction strategies (see e.g. Runnalls, 2007). A second broad category use the so-called Rao–Blackwellized method, that exploits the conditionally linear model structure and describes the model index trajectories $z_{1:N}$ using particle methods (Whiteley, Andrieu, & Doucet, 2010). For the remainder of this paper we will focus on the first group of methods.

Further categorisation includes smoothers based on *forward–backward*, or, *two-filter* formulations (see e.g. Doucet & Johansen, 2011). Forward–backward smoothers include the second order generalised pseudo-Bayesian (GPB2) method (Kim, 1994) and the expectation correction augmented SLDS (aSLDS EC) smoother (Barber, 2012). The GPB2 approach reduces the filtering and smoothing distributions to a single Gaussian component for each model index z_k, whereas the aSLDS-EC smoother allows a more general Gaussian-mixture for each model index. Both smoothers can employ standard Rauch–Tung–Striebel (RTS) corrections, but are forced to reduce the forward prediction distribution to a unimodal Gaussian in order to prevent division by a Gaussian mixture (Barber, 2012; Kim, 1994).

Two-filter formulations avoid the unimodal forward prediction approximation, but are not without challenges, principally that the backward filter also suffers from exponential growth in the number of terms. Reduction in this case requires some careful treatment (Balenzuela et al., 2018; Helmick et al., 1995; Kitagawa, 1994; Rahmathullah, Svensson, & Svensson, 2014). Importantly, traditional reduction methods are not generally applicable as detailed in Rahmathullah et al. (2014). The original two-filter interacting multiple model (IMM) smoother (Helmick et al., 1995) suggests an alternative approximation that employs pseudo-inverses. With this as background, the main contributions of the paper are stated below.

The contributions of this paper are therefore:

(1) Provide exact closed-form expressions for the smoothed distribution for jump Markov linear systems based on a two-filter formulation.
(2) As is well-known (Alspach & Sorenson, 1972; Rahmathullah et al., 2014), (1) involves an exponentially increasing number of mixture components, and as such we also provide a new backward filter likelihood reduction method. This method merges likelihood components in a manner that respects the system model and maintains zeroth, first and second order properties of the reduced components.

The resulting two-filter algorithm is both accurate and computationally tractable and compares favourably with state-of-the-art methods.

The remainder of the paper is organised as follows. In Section 2 we provide an exact solution to the smoothing problem. Section 3 presents a practical algorithm where the number of modes are moderated to manageable levels. Section 4 provides simulations results that compare the new algorithm with existing approaches and Section 5 provides some concluding remarks.

2. The exact solution

To solve the smoothing problem (3) we will employ the well known two-filter approach (Fraser, 1967; Mayne, 1966) that provides a smoothed state distribution according to

$$p(x_k, z_k | y_{1:N}) = \frac{p(y_{k+1:N} | x_k, z_k) p(x_k, z_k | y_{1:k})}{p(y_{k+1:N} | y_{1:k})}, \tag{5}$$

which is a direct application of Bayes' rule. This calculation requires the backwards filter (BF) likelihood $p(y_{k+1:N}|x_k, z_k)$, which contains all of the information from future measurements about the hybrid state, and is calculated recursively using

$$p(y_{k:N}|x_k, z_k) = p(y_k|x_k, z_k) p(y_{k+1:N}|x_k, z_k), \tag{6a}$$

$$p(y_{k:N}|x_{k-1}, z_{k-1}) \tag{6b}$$

$$= \sum_{z_k=1}^{m_k} \int p(y_{k:N}|x_k, z_k) p(x_k, z_k | x_{k-1}, z_{k-1})\, dx_k.$$

In the following Section 2.1 we derive expressions for recursively calculating the statistics of the backward filter likelihood $p(y_{k+1:N}|x_k, z_k)$, and then show how these objects can be used to generate the smoothed distribution $p(x_k, z_k|y_{1:N})$ in Section 2.2.

2.1. Backwards information filter

In this section we detail the novel implementation of the backwards information filter (BIF), which is a backwards filter using so-called information form statistics. We will make extensive use of these statistics in order to express the likelihood, which can be defined as

$$\mathcal{L}\,(x \,|\, r,\ s,\ \mathbf{L}) \triangleq e^{-\frac{1}{2}\left(r + 2x^T s + x^T \mathbf{L} x\right)}. \tag{7}$$

This likelihood is parameterised by the symmetric positive semidefinite information matrix $\mathbf{L} \in \mathbb{R}^{n_x \times n_x}$, information vector $s \in \mathbb{R}^{n_x}$, and information scalar $r \in \mathbb{R}$.

The utility of the information formulation is that it naturally caters for cases where the information matrix $\mathbf{L}$ is not invertible. This is important for capturing the sufficient statistics of backwards filter, since they are not guaranteed to be integrable over x_k (see e.g. Kitagawa, 1994), and therefore do not always have the same form as a Gaussian mixture distribution describing a probability density over x_k.

At the final time sample N, expressions for the likelihood can be straightforwardly obtained as

$$p(y_N|x_N, z_N) = \mathcal{L}\left(x_N \mid \bar{r}_N(z_N),\ \bar{s}_N(z_N),\ \bar{\mathbf{L}}_N(z_N)\right), \tag{8}$$

where

$$\bar{r}_N(z_N) = \zeta_N^T(z_N)\mathbf{R}_N^{-1}(z_N)\zeta_N(z_N) + \ln|2\pi\mathbf{R}_N(z_N)|, \tag{9a}$$

$$\bar{s}_N(z_N) = \mathbf{C}_N^T(z_N)\mathbf{R}_N^{-1}(z_N)\zeta_N(z_N), \tag{9b}$$

$$\bar{\mathbf{L}}_N(z_N) = \mathbf{C}_N^T(z_N)\mathbf{R}_N^{-1}(z_N)\mathbf{C}_N(z_N), \tag{9c}$$

$$\zeta_N(z_N) = \mathbf{D}_N(z_N)u_N - y_N. \tag{9d}$$

Recursing backwards from this initial likelihood to provide the remaining likelihoods is captured by Lemma 1 to follow. Similar equations for calculating information matrix $\mathbf{L}$ and information vector s can be found in existing literature, albeit for a slightly different model class (e.g. see Bergman & Doucet, 2000; Fox, Sudderth, Jordan, & Willsky, 2011; Helmick et al., 1995). Importantly, this existing work does not use or provide recursions for the information scalar r. This is vital to expressing the exact form for the two-filter smoother and is essential to likelihood reduction, which is treated in Section 3. Since the information scalar r intimately depends on $\mathbf{L}$ and s, the following lemma presents all required recursions.

Lemma 1. *Under the model class (1)–(2) and given $p(y_N|x_N, z_N)$, then for $k = N-1, \ldots, 1$, it follows that the backwards-propagated and corrected likelihoods are given by*

$$p(y_{k+1:N}|x_k, z_k) = \sum_{j=1}^{M_k^b} \mathcal{L}\left(x_k \mid r_k^j(z_k),\ s_k^j(z_k),\ \mathbf{L}_k^j(z_k)\right),$$

$$p(y_{k:N}|x_k, z_k) = \sum_{j=1}^{M_k^c} \mathcal{L}\left(x_k \mid \bar{r}_k^j(z_k),\ \bar{s}_k^j(z_k),\ \bar{\mathbf{L}}_k^j(z_k)\right),$$

respectively, where $M_k^b = m_{k+1}\cdot M_{k+1}^c$ and for each $\ell = 1, \ldots, m_{k+1}$, $i = 1, \ldots, M_{k+1}^b$ and $z_k = 1, \ldots, m_k$,

$$\mathbf{L}_k^j(z_k) = \mathbf{A}_k^T(z_k)\boldsymbol{\Phi}_k^j(z_k)\mathbf{A}_k(z_k), \tag{11a}$$

$$s_k^j(z_k) = \mathbf{A}_k^T(z_k)\left[\boldsymbol{\Phi}_k^j(z_k)\mathbf{B}_k(z_k)u_k + (\boldsymbol{\Gamma}_k^j(z_k))^T\bar{s}_{k+1}^i(\ell)\right], \tag{11b}$$

$$\begin{aligned} r_k^j(z_k) &= \bar{r}_{k+1}^i(\ell) - \ln|\boldsymbol{\Gamma}_k^j(z_k)| - 2\ln T_k(\ell|z_k) \\ &\quad + \begin{bmatrix}\bar{s}_{k+1}^i(\ell) \\ \mathbf{B}_k(z_k)u_k\end{bmatrix}^T \begin{bmatrix}\boldsymbol{\Psi}_k^j(z_k) & \boldsymbol{\Gamma}_k^j(z_k) \\ (\boldsymbol{\Gamma}_k^j(z_k))^T & \boldsymbol{\Phi}_k^j(z_k)\end{bmatrix} \begin{bmatrix}\bar{s}_{k+1}^i(\ell) \\ \mathbf{B}_k(z_k)u_k\end{bmatrix}, \end{aligned} \tag{11c}$$

$$\boldsymbol{\Gamma}_k^j(z_k) = \mathbf{I} - \mathbf{Q}_k(z_k)\boldsymbol{\Phi}_k^j(z_k), \tag{11d}$$

$$\boldsymbol{\Psi}_k^j(z_k) = \mathbf{Q}_k(z_k)\boldsymbol{\Phi}_k^j(z_k)\mathbf{Q}_k(z_k) - \mathbf{Q}_k(z_k), \tag{11e}$$

$$\boldsymbol{\Phi}_k^j(z_k) = \left(\mathbf{I} + \bar{\mathbf{L}}_{k+1}^i(\ell)\mathbf{Q}_k(z_k)\right)^{-1}\bar{\mathbf{L}}_{k+1}^i(\ell), \tag{11f}$$

$$j = M_{k+1}^b\cdot(\ell-1) + i, \tag{11g}$$

where $\mathbf{I}$ is the identity matrix and $M_k^c = M_k^b$, then for each $q = 1, \ldots, M_k^b$ and $z_k = 1, \ldots, m_k$,

$$\bar{\mathbf{L}}_k^q(z_k) = \mathbf{L}_k^q(z_k) + \mathbf{C}_k^T(z_k)\mathbf{R}_k^{-1}(z_k)\mathbf{C}_k(z_k), \tag{12a}$$

$$\bar{s}_k^q(z_k) = s_k^q(z_k) + \mathbf{C}_k^T(z_k)\mathbf{R}_k^{-1}(z_k)\zeta_k(z_k), \tag{12b}$$

$$\begin{aligned}\bar{r}_k^q(z_k) &= r_k^q(z_k) + \ln|2\pi\mathbf{R}_k(z_k)| \\ &\quad + \zeta_k^T(z_k)\mathbf{R}_k^{-1}(z_k)\zeta_k(z_k),\end{aligned} \tag{12c}$$

$$\zeta_k(z_k) = \mathbf{D}_k(z_k)u_k - y_k. \tag{12d}$$

Proof. See Appendix B.1.

2.2. Two-filter smoother for JMLS

In order to generate the smoothed distribution using the two-filter method, we also require the forward filtered distribution

$$p(x_k, z_k|y_{1:k}) = \sum_{i=1}^{M_k^f} w_{k|k}^i(z_k)\mathcal{N}(x_k|\mu_{k|k}^i(z_k), \mathbf{P}_{k|k}^i(z_k)).$$

Generation of this distribution is well covered within the literature (Barber, 2012, Chapter 25). It is important to note that some care is required to ensure the correct time index is used on the discrete state z_k and input u_k. Due to the tedious nature of deriving these formulas, we provide a set of expressions using the current notation in Appendix A.

In Lemma 2, we provide the equations for combining the forward filter distribution with the backward filter likelihood to generate the smoothed distribution.

Lemma 2. *The smoothed state distribution can be expressed for each $k = 1, \ldots, N-1$ as*

$$p(x_k, z_k|y_{1:N}) = \sum_{j=1}^{M_k^s} w_{k|N}^j(z_k)\mathcal{N}\left(x_k \mid \mu_{k|N}^j(z_k), \mathbf{P}_{k|N}^j(z_k)\right),$$

where $M_k^s = M_k^b\cdot M_k^f$ and for each $\ell = 1, \ldots, M_k^b$, $i = 1, \ldots, M_k^f$ and $z_k = 1, \ldots, m_k$,

$$w_{k|N}^j(z_k) = \frac{\tilde{w}_{k|N}^j(z_k)}{\sum_{a=1}^{m_k}\sum_{p=1}^{m_{k|N}}\tilde{w}_{k|N}^p(a)}, \tag{13a}$$

$$\tilde{w}_{k|N}^j(z_k) = \frac{w_{k|k}^i(z_k)\sqrt{|\mathbf{P}_{k|N}^j(z_k)|}}{\sqrt{|\mathbf{P}_{k|k}^i(z_k)|}}e^{\frac{1}{2}\beta^j}, \tag{13b}$$

$$\begin{aligned}\beta^j &= \left(\mu_{k|N}^j(z_k)\right)^T\left(\mathbf{P}_{k|N}^j(z_k)\right)^{-1}\mu_{k|N}^j(z_k) \\ &\quad - \left(\mu_{k|k}^i(z_k)\right)^T\left(\mathbf{P}_{k|k}^i(z_k)\right)^{-1}\mu_{k|k}^i(z_k) - r_k^\ell(z_k),\end{aligned} \tag{13c}$$

$$\mu_{k|N}^j(z_k) = \mathbf{P}_{k|N}^j(z_k)\left(\left[\mathbf{P}_{k|k}^i(z_k)\right)^{-1}\mu_{k|k}^i(z_k) - s_k^\ell(z_k)\right],$$

$$\mathbf{P}_{k|N}^j(z_k) = \left(\mathbf{L}_k^\ell(z_k) + \left(\mathbf{P}_{k|k}^i(z_k)\right)^{-1}\right)^{-1}, \tag{13d}$$

$$j = M_k^f\cdot(\ell-1) + i. \tag{13e}$$

Proof. See Appendix B.2.

The number of terms in both the forward and backward filter recursions grows exponentially. This implies that the above solution is not practical for computation, except for cases where the number of observations N is small, or for example, where a fixed-lag smoother is required with small lag-length. Otherwise, we are forced to maintain a practical number of terms in these filters. Maintaining a practical number of likelihood components in the backwards filter is discussed in the following section.

3. A practical algorithm

In this section we provide suitable approximations to reduce the number of components in the backward filter. This ultimately leads to a computationally tractable algorithm that is profiled against existing approaches in Section 4. To achieve this, we here employ a novel likelihood reduction strategy, which relies on a tuning parameters, namely the maximum number backwards information components M^c. This parameter provides a mechanism for trading accuracy (large M^c) against computational speed (small M^c) of the backward filter.

It is important to mention that other strategies have been proposed for a less general class of systems (Kitagawa, 1994), called Gaussian mixture models (GMMs). These suggestions include the use of additional prior information in the backward filter, which forces integrability, but ultimately degrades the estimate. A further suggestion in Kitagawa (1994) involves a batch calculation of the backward filter. In a similar manner, Rahmathullah et al. (2014) augment observations and perform a reduction in dimension when (or if) integrable likelihoods are formed, but also prune unlikely model sequences based off the probability of smoothed offspring. Here we take a fundamentally different approach, described as follows.

3.1. Likelihood reduction

Similar to the forward filter, the backward filter requires an approximation to be made to prevent the computational complexity growing exponentially with each iteration. Likelihood mixture reduction presents additional challenges compared to probability distributions. This fundamental difficulty with merging components from the backward filter stems from the possibility of the likelihood functions having a constant value over a subspace of the state-space (Rahmathullah et al., 2014). Because of this, known function approximators for density functions cannot be applied straightforwardly.

Therefore, consider the reduction of two likelihood components $\mathcal{L}\,(x\,|\,r_i,\,s_i,\,\mathbf{L}_i)$ and $\mathcal{L}\left(x\,|\,r_j,\,s_j,\,\mathbf{L}_j\right)$ to a single likelihood $\mathcal{L}\left(x\,|\,r_{ij},\,s_{ij},\,\mathbf{L}_{ij}\right)$, where $\{\mathbf{L}_{ij}, s_{ij}, r_{ij}\}$ is to be determined from the merge operation. We will restrict the merge operation to (i, j) pairs that satisfy a range-space condition that

$$\mathcal{R}(\mathbf{L}_i) = \mathcal{R}(\mathbf{L}_j), \qquad s_i, s_j \in \mathcal{R}(\mathbf{L}_i), \tag{14}$$

where $\mathcal{R}(\mathbf{A}) \subseteq \mathbb{R}^m$ is the range of a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ and is given by $\mathcal{R}(\mathbf{A}) = \{\mathbf{A}x | x \in \mathbb{R}^n\}$. Since the two input likelihood modes must satisfy a range-space condition, we will further enforce that the output likelihood mode should also satisfy the same range-space condition, otherwise the replacement likelihood component offers new information that was not present in the original modes. That is

$$\mathcal{R}(\mathbf{L}_{ij}) = \mathcal{R}(\mathbf{L}_i) = \mathcal{R}(\mathbf{L}_j), \qquad s_{ij} \in \mathcal{R}(\mathbf{L}_{ij}). \tag{15}$$

The information matrix $\mathbf{L}_i$ is positive semi-definite and symmetric, so that it affords a singular value decomposition (SVD)

$$\mathbf{L}_i = \begin{bmatrix}\mathbf{U} & \mathbf{Z}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Sigma}_i & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{U}^T\\ \mathbf{Z}^T\end{bmatrix}, \tag{16}$$

where $\mathbf{0}$ is the zeros matrix, $\boldsymbol{\Sigma}_i \in \mathbb{R}^{d\times d}$ is a diagonal matrix whose diagonal entries are the non-negative singular values, the columns of $\mathbf{U} \in \mathbb{R}^{n\times d}$ provide an orthonormal basis for the range-space of $\mathbf{L}_i$ and where the columns of $\mathbf{Z} \in \mathbb{R}^{n\times n-d}$ provide an orthonormal basis for the null-space of $\mathbf{L}_i$. Further, by assumption $\mathcal{R}(\mathbf{L}_i) = \mathcal{R}(\mathbf{L}_j) = \mathcal{R}(\mathbf{L}_{ij})$ and therefore

$$\mathbf{L}_j = \mathbf{U}\boldsymbol{\Sigma}_j\mathbf{U}^T, \qquad \boldsymbol{\Sigma}_j = \boldsymbol{\Sigma}_j^T \succ 0, \tag{17a}$$

$$\mathbf{L}_{ij} = \mathbf{U}\boldsymbol{\Sigma}_{ij}\mathbf{U}^T, \qquad \boldsymbol{\Sigma}_{ij} = \boldsymbol{\Sigma}_{ij}^T \succ 0, \tag{17b}$$

$$s_i = \mathbf{U}\eta_i, \qquad \text{for some} \quad \eta_j \in \mathbb{R}^d, \tag{17c}$$

$$s_j = \mathbf{U}\eta_j, \qquad \text{for some} \quad \eta_i \in \mathbb{R}^d, \tag{17d}$$

$$s_{ij} = \mathbf{U}\eta_{ij}, \qquad \text{for some} \quad \eta_{ij} \in \mathbb{R}^d. \tag{17e}$$

Note that $\boldsymbol{\Sigma}_j$ and $\boldsymbol{\Sigma}_{ij}$ are not necessarily diagonal, and $\boldsymbol{\Sigma}_j$ can be conveniently computed via $\mathbf{U}^T\mathbf{L}_j\mathbf{U}$. Additionally, η_i and η_j can be computed using $\mathbf{U}^Ts_i$, and $\mathbf{U}^Ts_j$ respectively.

It follows immediately from the fundamental theorem of linear algebra that

$$x \in \mathcal{R}(\mathbf{L}_i) \iff x = \mathbf{U}\tilde{x}, \quad \text{for some} \quad \tilde{x} \in \mathbb{R}^d. \tag{18}$$

Therefore, the following equalities hold (by substitution) for any $\tilde{x} \in \mathbb{R}^d$

$$\mathcal{L}(x|r_i, s_i, \mathbf{L}_i) = \mathcal{L}(\tilde{x}|r_i, \eta_i, \boldsymbol{\Sigma}_i), \tag{19a}$$

$$\mathcal{L}(x|r_j, s_j, \mathbf{L}_j) = \mathcal{L}(\tilde{x}|r_j, \eta_j, \boldsymbol{\Sigma}_j), \tag{19b}$$

$$\mathcal{L}(x|r_{ij}, s_{ij}, \mathbf{L}_{ij}) = \mathcal{L}(\tilde{x}|r_{ij}, \eta_{ij}, \boldsymbol{\Sigma}_{ij}). \tag{19c}$$

Since $\boldsymbol{\Sigma}_i$, $\boldsymbol{\Sigma}_j$ and $\boldsymbol{\Sigma}_{ij}$ are all full rank, then each exponential term can be expressed as a scaled multivariate Normal distribution according to

$$\mathcal{L}(\tilde{x}|r, \eta, \boldsymbol{\Sigma}) = \alpha\, \mathcal{N}\left(\tilde{x}\,|\,\boldsymbol{\Sigma}^{-1}\eta, \boldsymbol{\Sigma}^{-1}\right), \tag{20}$$

$$\alpha = e^{-\frac{1}{2}\left(r - \eta^T\boldsymbol{\Sigma}^{-1}\eta - \ln|2\pi\boldsymbol{\Sigma}^{-1}|\right)}. \tag{21}$$

Importantly, we can then employ standard GM reduction methods on mixture

$$f(\tilde{x}) = \sum_{\ell}\bar{\alpha}_\ell\mathcal{N}(\tilde{x}|\boldsymbol{\Sigma}_\ell^{-1}\eta_\ell, \boldsymbol{\Sigma}_\ell^{-1}), \quad \bar{\alpha}_\ell = \frac{\alpha_\ell}{\sum_\ell \alpha_\ell}. \tag{22}$$

One possibility is to use Kullback–Leibler reduction (KLR) (Runnalls, 2007) to reduce this mixture, where a bound is placed on the approximation error defined by the Kullback–Leibler divergence (KLD) between the original and reduced mixture after a merge of any two components. It should be noted that KLD is not a symmetric measure, however the approach presented in Runnalls (2007) produces a symmetric upper bound. Also note that alternatives, such as Wasserstein metrics may be used (see e.g. Lee, Halder, & Bhattacharya, 2015).

For KLR, the equations to be implemented for merging an $(i{-}j)$ likelihood pair are

$$r_{ij} = \eta_{ij}^T\boldsymbol{\Sigma}_{ij}^{-1}\eta_{ij} - 2\ln\left(\alpha_i + \alpha_j\right) + \ln|2\pi\boldsymbol{\Sigma}_{ij}^{-1}|, \tag{23a}$$

$$\eta_{ij} = \boldsymbol{\Sigma}_{ij}\left(v_i\mu_i + v_j\mu_j\right), \tag{23b}$$

$$\boldsymbol{\Sigma}_{ij}^{-1} = v_i\boldsymbol{\Sigma}_i^{-1} + v_j\boldsymbol{\Sigma}_j^{-1} + v_iv_j\left(\mu_i - \mu_j\right)\left(\mu_i - \mu_j\right)^T,$$

$$\mu_i = \boldsymbol{\Sigma}_i^{-1}\eta_i, \quad \mu_j = \boldsymbol{\Sigma}_j^{-1}\eta_j, \tag{23c}$$

$$v_i = \frac{\alpha_i}{\alpha_i + \alpha_j}, \quad v_j = \frac{\alpha_j}{\alpha_i + \alpha_j}, \tag{23d}$$

where the $i - j$ pair chosen when multiple likelihoods share a common range-space has the lowest upper bound of the relative KLD given by

$$\bar{B}(i, j) = (\alpha_i + \alpha_j)\ln|\boldsymbol{\Sigma}_{ij}^{-1}| + \alpha_i\ln|\boldsymbol{\Sigma}_i| + \alpha_j\ln|\boldsymbol{\Sigma}_j|. \tag{24}$$

Using the above procedure, we can merge likelihood components within a common range-space until a desired maximum number of components M^c is achieved. Following this reduction process, the generated components can be transformed back into the original state-space using

$$s = \mathbf{U}\eta, \qquad \mathbf{L} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^T, \tag{25}$$

where $\mathbf{U}$ is common to each component sharing the range-space. This approach outlined by (14)–(25) is repeated for each range-space likelihood modes occupy.

For illustration, Fig. 1 shows the scenario where two likelihood components have a common range-space. These likelihood components contain no information about state x_2 and have been superimposed to generate the surface mesh. Using the method provided, a transformation was made to describe these likelihood

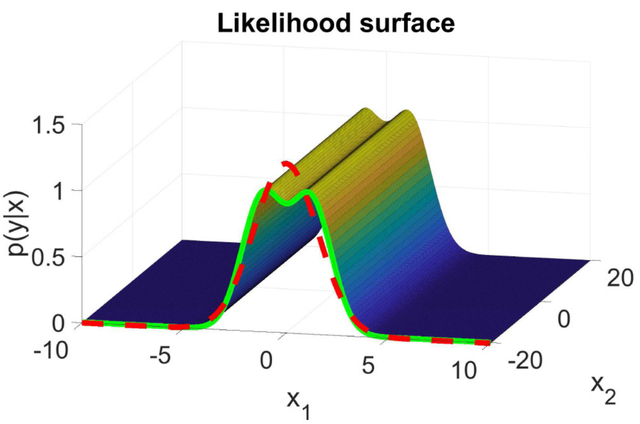

Fig. 1. A likelihood surface with two components comprising $p(y|x)$ is shown, which can also be described in a reduced space shown in solid green. After component reduction, the likelihood function is approximated in the reduced space by the dashed red line, before being transformed back into the full state-space.

components in terms of the common range-space before super imposing them to produce the solid green line. Next the KLR method was used to combine these components in the range-space, resulting in the approximation shown in dotted red. Following this, the approximated likelihood can be transformed back into the original 2D space. Note that very seldom does the observed space align with the basis for the states, but this case is automatically handled by use of the SVD.

3.2. Algorithm overview

The following Algorithm description is provided to clarify the overall operation of the proposed JMLS smoother.

Algorithm 1 The Two-filter JMLS Smoother

Require: The JMLS system (1)–(2), data $y_{1:N}$ and $u_{1:N}$, the prior $p(x_1, z_1)$, and a maximum number of forward filter components M^f, and a maximum number of backward information filter components M^c.

for $k = 1, \ldots, N$ **do**
  Calculate $p(x_k, z_k|y_{1:k})$ via Appendix A.
  Perform reduction of $p(x_k, z_k|y_{1:k})$ so that it has no more than M^f components, e.g. via (Runnalls, 2007).
  Calculate $p(x_{k+1}, z_{k+1}|y_{1:k})$ via Appendix A.
end for
Initialise the backwards information filter likelihood with $p(y_N|x_N, z_N)$ via (9).
for $k = N-1, \ldots, 1$ **do**
  Calculate $p(y_{k+1:N}|x_k, z_k)$ via Lemma 1.
  Calculate $p(y_{k:N}|x_k, z_k)$ using Lemma 1.
  Perform likelihood reduction using Section 3.1 so that no more than M^c components is in the mixture.
end for
for $k = 1, \ldots, N-1$ **do**
  Calculate $p(x_k, z_k|y_{1:N})$ using Lemma 2.
end for

The above (highly parellelisable) proposed solution for JMLS smoothing has the computational complexity of $\mathcal{O}\left((N-1)m^2M^c(n_x^3 + n_y^3 + n_x^2 n_y + n_x n_u + n_y n_u)\right)$ for the BIF and $\mathcal{O}\left((N-1)(mM^cM^f n_x^3)\right)$ for the smoother.

As the true smoothed distribution has a growing number of components, choosing a finite number of allowed forward and backward filter components (M^f and M^c, respectively) will certainly impact accuracy, but in a manner that is extremely difficult to predict in general. In essence, larger values for M^f and M^c improve accuracy, while smaller values reduce computational complexity. Therefore, we adopt the pragmatic approach that available computational resources often stipulate a limit on the number of allowed components and use this to set the values for M^f and M^c. Ideally, the number of components stored after reduction would be sufficiently large that increasing it would not (within reason) alter the distribution.

4. Simulations

Here we provide the results from smoothing three different systems to demonstrate the effectiveness and versatility of the proposed solution summarised in Algorithm 1.

4.1. Example 1 — A jump Markov linear system

In this example we consider a JMLS system in the following form as used in Barber (2006), Doucet, Gordon, and Krishnamurthy (2001), Helmick et al. (1995) and Kim (1994),

$$x_k = \mathbf{A}(z_k)x_{k-1} + \mathbf{B}(z_k)u_k + v_{k-1}, \tag{26a}$$

$$y_k = \mathbf{C}(z_k)x_k + \mathbf{D}(z_k)u_k + e_k, \tag{26b}$$

$$v_{k-1} \sim \mathcal{N}(v_{k-1} \mid 0, \mathbf{Q}(z_k)), \tag{26c}$$

$$e_k \sim \mathcal{N}(e_k \mid 0, \mathbf{R}(z_k)), \tag{26d}$$

this form differs from the system described in (1), as the above system uses a different time index for the switching parameter in the prediction step. To accommodate this, we modify proposed backward filter such that the model switches before the prediction step, and not after. Since some of the alternate algorithms do not explain in detail how to perform likelihood reduction in the case of non-integrable likelihood functions, we use a single state system to circumvent this problem in order to compare their performance to the proposed method.

The parameters used for data generation and smoothing of this system were $u_k = 1$ for all k and

$$\begin{aligned}
&\mathbf{A}(1) = 0.9,\ \mathbf{B}(1) = 0.1,\ \mathbf{C}(1) = 0.9,\ \mathbf{D}(1) = 0.05,\\
&\mathbf{A}(2) = 0.9,\ \mathbf{B}(2) = 0.12,\ \mathbf{C}(2) = 0.85,\ \mathbf{D}(2) = 0.05,\\
&\mathbf{Q}(1) = 0.45,\ \mathbf{R}(1) = 0.5,\\
&\mathbf{Q}(2) = 0.01,\ \mathbf{R}(2) = 1.5,\ \mathbf{T} = \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}.
\end{aligned} \tag{27}$$

The system was simulated for $N = 15$ time steps to produce input–output data. A range of smoothers were run on this data including the proposed method (abbreviated as JMLSS) and other smothers including the IMM, GPB2, and aSLDS EC smoother. A particle smoother (PS) solution was also implemented to provide a ground truth estimate, however it took orders of magnitude longer to run. This experiment repeated 250 times with different datasets, where the distributions were used to calculate a mean KLD error over each timeseries for each of the methods, which in turn was used to generate the boxplot in Fig. 2. This shows the upper bound outliers of the proposed JMLS smoother (JMLSS) are below the lower bounds of the alternate approaches. This indicates that the proposed method consistently outperformed the alternative approaches on each of these 250 runs.

Fig. 3 provides further insight into the level of accuracy of the distributions from the proposed method. In particular, this figure shows the proposed method consistently producing distributions with very little error. Alternative approaches such as the IMM smoother, are shown to produce some distributions with low error, but appear to be inconsistent at doing so when compared to the proposed JMLSS.

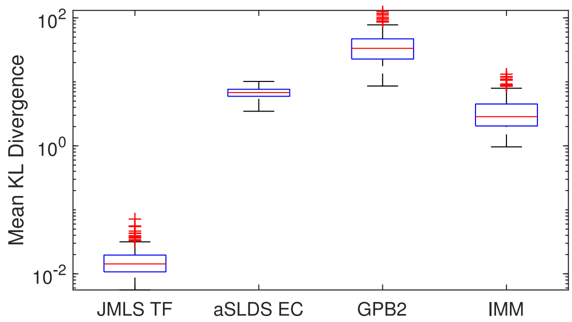

Fig. 2. Boxplot (McGill, Tukey, & Larsen, 1978) of KLD over 250 random smoothed datasets for Example 1.

Table 1
Computation time (s) for Example 1.

PS	JMLSS	aSLDS EC	GPB2	IMM
16598	0.39	0.45	0.02	0.04

Table 1 records the computation time for each method, and it should be noted that all methods were implemented fairly in native Matlab code. It may be concluded that the computation time associated with the JMLSS is an order of magnitude more than the fastest method, but around two orders of magnitude better in terms of accuracy.

4.2. Example 2 — Dynamic JMLS system

In this example we consider a two-state problem, which takes full advantage of the proposed likelihood reduction strategy, as it is a multi-state problem which presents with non-integrable likelihood components in the BIF. The example considers the dynamical model of a mass–spring–damper (MSD) system with a position sensor

$$m\ddot{x}(t) + b\dot{x}(t) + kx(t) = F(t), \qquad y(t) = x(t), \tag{28}$$

where $x(t)$ is the mass position, $F(t)$ is the applied external force, m is the system mass, b is the damping coefficient and k is the spring constant. We further assume that the system can use one of two available sets of parameters, indicated by a superscript, at any timestep,

$$\begin{aligned} &m^1 = 8 \text{ [kg]},\ b^1 = 12 \text{ [Ns/m]},\ k^1 = 10 \text{ [N/m]}, \\ &m^2 = 8 \text{ [kg]},\ b^2 = 0 \text{ [Ns/m]},\ k^2 = 0 \text{ [N/m]}. \end{aligned} \tag{29}$$

The second parameter set represents a possible fault scenario, where the spring and damper have become disconnected from the mass, the transition matrix for this example was chosen to reflect a 1% chance of permanent failure

$$\mathbf{T}_{1,1} = 0.99, \quad \mathbf{T}_{1,2} = 0, \quad \mathbf{T}_{2,1} = 0.01, \quad \mathbf{T}_{2,2} = 1,$$

Using an approach similar to that in Ljung and Wills (2010) and Wills, Schön, Lindsten, and Ninness (2012), the models were discretised and converted into the model class (1) using a sample rate of 100 Hz and simulated analogue to digital converter (ADC) sample time of 0.1 ms. The system was assumed to be driven by $F[k] = 2000 \sin(k/(20\pi))$, then smoothed using the proposed method and a particle smoother (PS) to provide a ground truth. As other alternative smoothers without modification do not support the system form (1) required for this discretisation, they do not appear in this experiment.

The resulting marginalised distributions from this experiment are shown in Fig. 4. Note that the system could easily be smoothed for higher dimensional systems or a larger number of timesteps using the proposed computationally inexpensive closed-form solution, but was set to a second order system with $N = 10$ time steps due to the large computational expense of the particle smoother. It is also possible to increase number of models to accommodate a variety of other fault scenarios, such as sensor failure.

From Fig. 4, the proposed method has been demonstrated to produce accurate smoothed distributions for a multivariate JMLS problem, when operating according to the dynamic class.

4.3. Example 3 — A 5th order dynamic jump Markov linear system

In this example, we very briefly consider a randomly generated 5th order, 2 mode dynamic JMLS system, where each of the linear models is generated using MATLAB's `drss` function. This generated system was simulated for $N = 2000$ time steps with an input $u_k \sim \mathcal{N}(0, 1)$, before filtering and the proposed smoothing algorithm was executed on the dataset. The produced hybrid smoothed distribution was then used to produce estimates of the continuous state at the start of the simulation ($k = 1$). The produced smoothed distributions and the true value of the state is shown in Fig. 5. Note that due to the curse of dimensionality, particle smoothing methods are impractical for this example and are therefore not provided. Additionally, as this system is operating according the dynamic convention, the alternative methods are unable to be used for comparison.

5. Conclusion

We have developed a new smoothing algorithm for jump-Markov-linear-systems using a two-filter approach. This contribution has two components. Firstly, we have developed an algorithm that implements the two-filter smoothing formulas *exactly*, therefore relaxing assumptions imposed by competing methods. Secondly, we have developed an approximation method that allows likelihood reduction that, contrary to existing methods, does not require likelihoods to have a Gaussian form in order to perform merging. The implication is that the new method presents the user with a mechanism to choose between computational cost and accuracy, making it suitable for both real-time and post processing applications.

Compared to alternatives, the proposed approach is very well suited to applications where likelihood modes in the BIF are non-integrable over x_k. This can be encountered for a number of reasons, and is a common occurrence in the first few iterations of the BIF. This property allows the JMLS estimator to handle models with only partial observability of the system state, which is useful for applications in fault diagnosis.

The proposed method has been demonstrated to reliably produce accurate distributions, offering improvements over the compared alternative methods, whilst remaining comparable in terms of computational cost.

Appendix A. Forwards filter

The forward filter and predicted distributions are well known (Barber, 2012). They are included here for completeness using the notation and system description (1)–(2) (which is slightly different to existing descriptions). Assuming that the initial

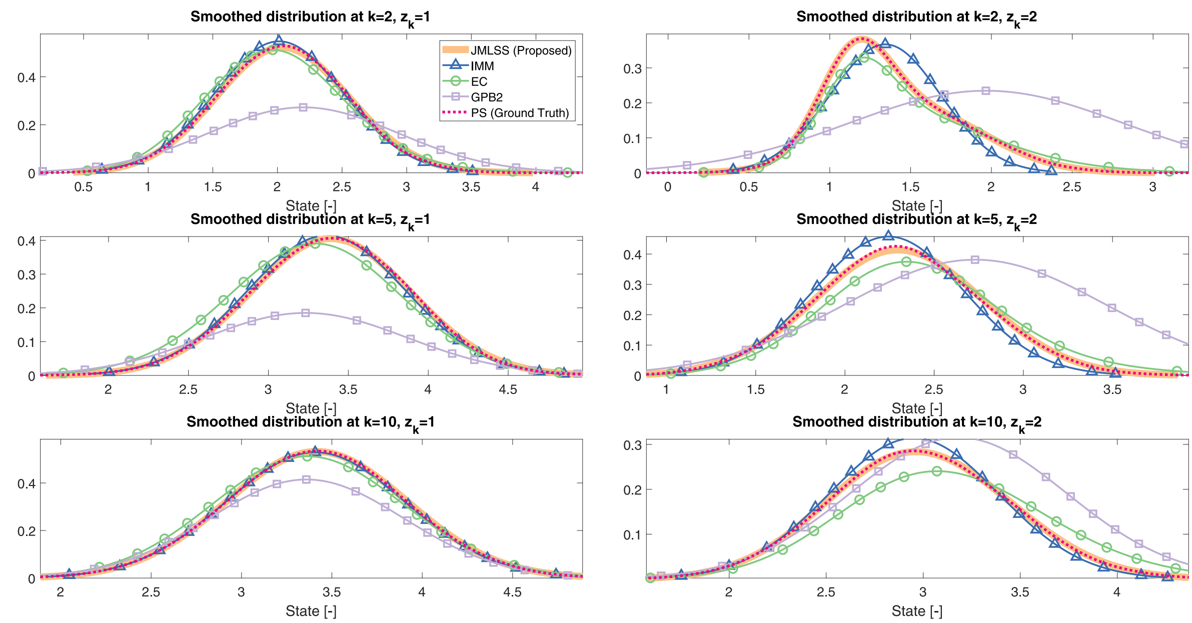

Fig. 3. Generated smoothed distributions for the JMLS system from Example 1.

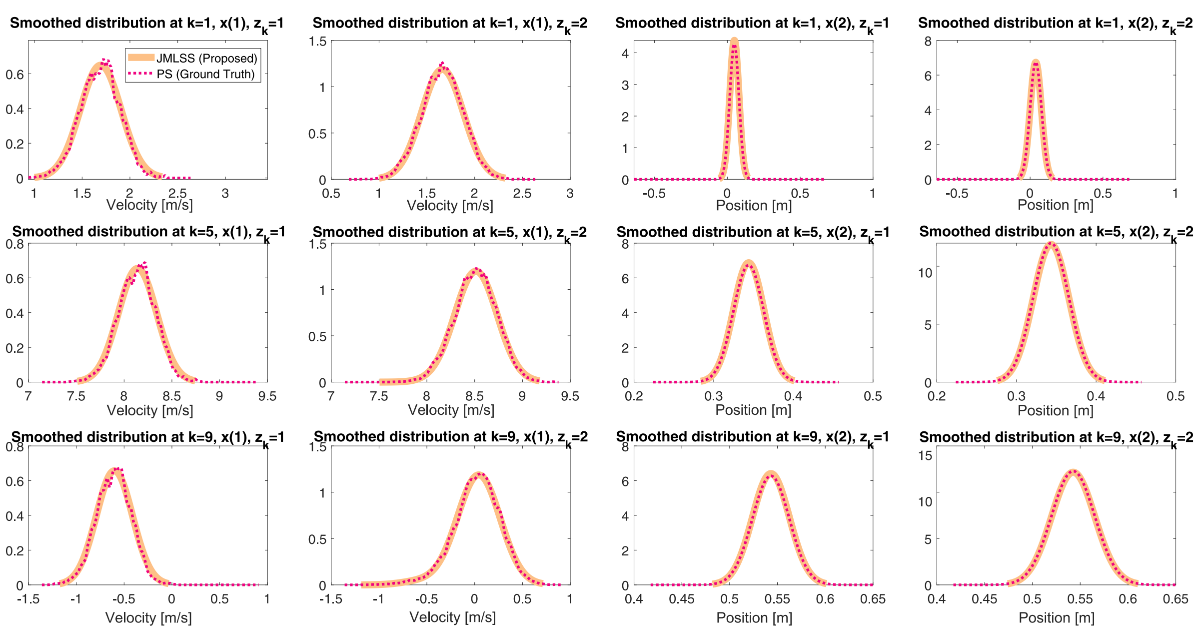

Fig. 4. Generated smoothed distributions for the dynamic JMLS system from Example 2.

distribution is $p(x_1, z_1) = \sum_{i=1}^{M_1^{\mathrm{p}}} w^i_{1|0}(z_1)\mathcal{N}\left(x_1 \middle| \mu^i_{1|0}(z_1), \mathbf{P}^i_{1|0}(z_1)\right)$, then the predicted and filtered distributions are given for $k = 1, \ldots, N$ by

$$p(x_k, z_k|y_{1:k}) = \sum_{i=1}^{M_k^{\mathrm{f}}} w^i_{k|k}(z_k)\mathcal{N}\left(x_k \middle| \mu^i_{k|k}(z_k), \mathbf{P}^i_{k|k}(z_k)\right),$$

$$p(x_{k+1}, z_{k+1}|y_{1:k}) = \sum_{j=1}^{M_{k+1}^{\mathrm{p}}} w^j_{k+1|k}(z_{k+1})\mathcal{N}\left(x_{k+1} \middle| \mu^j_{k+1|k}(z_{k+1}), \mathbf{P}^j_{k+1|k}(z_{k+1})\right),$$

respectively, where $M_k^{\mathrm{f}} = M_k^{\mathrm{p}}$ and for each $i = 1, \ldots, M_k^{\mathrm{p}}$ and $z_k = 1, \ldots, m_k$,

$$w^i_{k|k}(z_k) = \frac{\tilde{w}^i_{k|k}(z_k)}{\sum_{z_k=1}^{m_k} \sum_{i=1}^{M_k^{\mathrm{p}}} \tilde{w}^i_{k|k}(z_k)}, \tag{A.1a}$$

$$\tilde{w}^i_{k|k}(z_k) = w^i_{k|k-1}(z_k) \cdot \mathcal{N}\left(y_k | \eta^i_k(z_k), \boldsymbol{\Xi}^i_k(z_k)\right), \tag{A.1b}$$

$$\mu^i_{k|k}(z_k) = \mu^i_{k|k-1}(z_k) + \mathbf{K}^i_k(z_k)[y_k - \eta^i_k(z_k)], \tag{A.1c}$$

$$\eta^i_k(z_k) = \mathbf{C}_k(z_k)\mu^i_{k|k-1}(z_k) + \mathbf{D}_k(z_k)u_k, \tag{A.1d}$$

$$\mathbf{P}^i_{k|k}(z_k) = \mathbf{P}^i_{k|k-1}(z_k) - \mathbf{K}^i_k(z_k)\mathbf{C}_k(z_k)\mathbf{P}^i_{k|k-1}(z_k), \tag{A.1e}$$

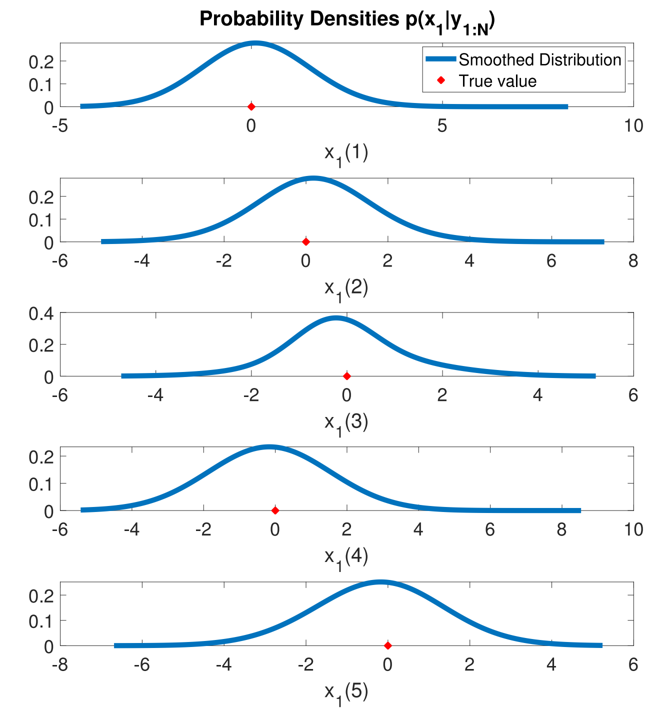

Fig. 5. Generated smoothed distributions for the dynamic JMLS system from Example 3.

$$\mathbf{K}^i_k(z_k) = \mathbf{P}^i_{k|k-1}(z_k)\mathbf{C}^T_k(z_k)(\varXi^i_k(z_k))^{-1}, \tag{A.1f}$$

$$\varXi^i_k(z_k) = \mathbf{C}_k(z_k)\mathbf{P}^i_{k|k-1}(z_k)\mathbf{C}^T_k(z_k) + \mathbf{R}_k(z_k), \tag{A.1g}$$

and $M^{\mathrm{p}}_{k+1} = m_k \cdot M^{\mathrm{f}}_k$, then for each $i = 1, \ldots, M^{\mathrm{f}}_k$, $\ell = 1, \ldots, m_k$ and $z_{k+1} = 1, \ldots, m_{k+1}$,

$$w^j_{k+1|k}(z_{k+1}) = T_k(z_{k+1}|\ell) \cdot w^i_{k|k}(\ell), \tag{A.2a}$$

$$\mu^j_{k+1|k}(z_{k+1}) = \mathbf{A}_k(\ell)\mu^i_{k|k}(\ell) + \mathbf{B}_k(\ell)u_k, \tag{A.2b}$$

$$\mathbf{P}^j_{k+1|k}(z_{k+1}) = \mathbf{A}_k(\ell)\mathbf{P}^i_{k|k}(\ell)\mathbf{A}^T_k(\ell) + \mathbf{Q}_k(\ell), \tag{A.2c}$$

$$j = M^{\mathrm{f}}_k \cdot (\ell - 1) + i. \tag{A.2d}$$

Appendix B. Lemma proofs

B.1. Proof of Lemma 1

Let $\xi \triangleq \{x_k, z_k\}$ and note that $p(y_N \mid \xi_N)$ is provided in (8). According to (6), then $p(y_N \mid \xi_{N-1})$ is given by

$$p(y_N|\xi_{N-1}) = \sum_{z_N=1}^{m_N} \int p(y_N|\xi_N)\, p(\xi_N|\xi_{N-1})\, dx_N \tag{B.1}$$

By direct substitution of the process model $p(\xi_N \mid \xi_{N-1})$ from (1)–(2) and direct substitution of $p(y_N|\xi_N)$ from (8), it follows immediately that (B.1) can be expressed as

$$\begin{aligned} p(y_N|\xi_{N-1}) &= \sum_{z_N=1}^{m_N} \int T_{N-1}(z_N|z_{N-1}) \\ &\cdot \mathcal{L}\left(x_N \mid \bar{r}_N(z_N),\ \bar{s}_N(z_N),\ \bar{\mathbf{L}}_N(z_N)\right) \\ &\cdot \mathcal{N}(x_N|\mathbf{A}_{N-1}(z_{N-1})x_{N-1} + b_{N-1}(z_{N-1}), \mathbf{Q}_{N-1}(z_{N-1}))\, dx_N \end{aligned}$$

Therefore, we can employ Proposition 3 with $\tau \triangleq T_{N-1}(z_N|z_{N-1})$ to arrive at

$$\begin{aligned} p(y_N|\xi_{N-1}) \\ = \sum_{j=1}^{m_N} \mathcal{L}(x_{N-1}|r^j_{N-1}(z_{N-1}), s^j_{N-1}(z_{N-1}), \mathbf{L}^j_{N-1}(z_{N-1})), \end{aligned}$$

where $r^j_{N-1}(z_{N-1}) \triangleq r_{N-1}(z_{N-1}, j)$, $s^j_{N-1}(z_{N-1}) \triangleq s_{N-1}(z_{N-1}, j)$ and $\mathbf{L}^j_{N-1}(z_{N-1}) \triangleq \mathbf{L}_{N-1}(z_{N-1}, j)$ result from Proposition 3 and are provided in (11) with $k = N-1$. From (6) and direct substitution of the above expression for $p(y_N|\xi_{N-1})$ and direct substitution of the likelihood model from (1) we obtain $p(y_{N-1:N} \mid \xi_{N-1})$ as

$$\begin{aligned} p(y_{N-1:N} \mid \xi_{N-1}) &= p(y_N \mid \xi_{N-1})\, p(y_{N-1} \mid \xi_{N-1}), \\ &= \sum_{j=1}^{m_N} \mathcal{L}(x_{N-1}|r^j_{N-1}(z_{N-1}), s^j_{N-1}(z_{N-1}), \mathbf{L}^j_{N-1}(z_{N-1})) \\ &\cdot \mathcal{N}(y_{N-1}|\mathbf{C}_{N-1}(z_{N-1})x_{N-1} + d_{N-1}(z_{N-1}), \mathbf{R}_{N-1}(z_{N-1})) \end{aligned}$$

Direct application of Proposition 4 leads to

$$\begin{aligned} p(y_{N-1:N} \mid \xi_{N-1}) \\ = \sum_{j=1}^{m_N} \mathcal{L}(x_{N-1}|\bar{r}^j_{N-1}(z_{N-1}), \bar{s}^j_{N-1}(z_{N-1}), \bar{\mathbf{L}}^j_{N-1}(z_{N-1})) \end{aligned}$$

with $\bar{r}^j_{N-1}(z_{N-1})$, $\bar{s}^j_{N-1}(z_{N-1})$ and $\bar{\mathbf{L}}^j_{N-1}(z_{N-1})$ defined in (12) with $k = N-1$. According to the recursion (6), we can repeat the above steps to arrive at the expressions for $p(y_{k+1:N} \mid \xi_k)$ and $p(y_{k:N}\xi_k)$ given by Lemma 1. □

B.2. Proof of Lemma 2

Let $\xi_k = (x_k, z_k)$ and recall from (5) that

$$p(\xi_k|y_{1:N}) = \frac{p(y_{k+1:N}|\xi_k)p(\xi_k|y_{1:k})}{\sum_{z_k=1}^{m_k} \int p(y_{k+1:N}|\xi_k)p(\xi_k|y_{1:k})\, dx_k}. \tag{B.2}$$

Concerning the numerator

$$\begin{aligned} p(y_{k+1:N}|\xi_k)&p(\xi_k|y_{1:k}) \\ &= \sum_{\ell=1}^{M^{\mathrm{b}}_k} \mathcal{L}(x_k|\bar{r}^{\ell}_k(z_k), \bar{s}^{\ell}_k(z_k), \bar{\mathbf{L}}^{\ell}_k(z_k)) \\ &\cdot \sum_{i=1}^{M^{\mathrm{f}}_k} w^i_{k|k}(z_k)\mathcal{N}(x_k|\mu^i_{k|k}(z_k), \mathbf{P}^i_{k|k}(z_k)), \\ &= \sum_{\ell=1}^{M^{\mathrm{b}}_k} \sum_{i=1}^{M^{\mathrm{f}}_k} \tilde{w}^{(i,\ell)}_{k|N}(z_k)\mathcal{N}(x_k|\mu^{(i,\ell)}_{k|N}(z_k), \mathbf{P}^{(i,\ell)}_{k|N}(z_k)), \\ &= \sum_{j=1}^{M^{\mathrm{s}}_k} \tilde{w}^j_{k|N}(z_k)\mathcal{N}(x_k|\mu^j_{k|N}(z_k), \mathbf{P}^j_{k|N}(z_k)), \end{aligned} \tag{B.3}$$

where the second-last equality comes from applying Proposition 5 and the last equality is a result of collapsing the double sum over $\{i, \ell\}$ into single sum over j with $M^{\mathrm{s}}_k \triangleq M^{\mathrm{f}}_k M^{\mathrm{b}}_k$. Returning to the denominator of (B.2), direct substitution of (B.3) and exploiting unit area under the Normal PDF yields

$$p(\xi_k|y_{1:N}) = \sum_{j=1}^{M^{\mathrm{s}}_k} w^j_{k|N}(z_k)\mathcal{N}(x_k|\mu^j_{k|N}(z_k), \mathbf{P}^j_{k|N}(z_k)),$$

$$w^j_{k|N}(z_k) = \frac{\tilde{w}^j_{k|N}(z_k)}{\sum_{z_k=1}^{m_k}\sum_{j=1}^{M^{\mathrm{s}}_k} \tilde{w}^j_{k|N}(z_k)}. \quad \square$$

Appendix C. Required propositions

Proposition 3. *Let the likelihood $\mathcal{L}(\hat{x}|\bar{r},\bar{s},\bar{\mathbf{L}})$ and Normal distribution $\mathcal{N}(\hat{x}|\mathbf{A}x+b,\mathbf{Q})$ and a positive scalar τ be given. Then*

$$\mathcal{L}(x|r,s,\mathbf{L}) = \int \tau\mathcal{N}(\hat{x}|\mathbf{A}x+b,\mathbf{Q})\mathcal{L}(\hat{x}|\bar{r},\bar{s},\bar{\mathbf{L}})\,d\hat{x}, \tag{C.1}$$

where $\mathbf{L} = \mathbf{A}^T\boldsymbol{\Phi}\mathbf{A}$, $\boldsymbol{\Phi} = \left(\mathbf{I}+\bar{\mathbf{L}}\mathbf{Q}\right)^{-1}\bar{\mathbf{L}}$, $s = \mathbf{A}^T(\boldsymbol{\Phi}b + \boldsymbol{\Gamma}^T\bar{s})$, $\boldsymbol{\Psi} = \mathbf{Q}\boldsymbol{\Phi}\mathbf{Q}-\mathbf{Q}$, $\boldsymbol{\Gamma} = \mathbf{I}-\mathbf{Q}\boldsymbol{\Phi}$, *and* $r = \bar{r} - \ln|\boldsymbol{\Gamma}| + \bar{s}^T\boldsymbol{\Psi}\bar{s} + 2\bar{s}^T\boldsymbol{\Gamma}b + b^T\boldsymbol{\Phi}b + \ln\tau$.

Proof. Let $\mathbf{M} \triangleq \bar{\mathbf{L}}+\mathbf{Q}^{-1}$, $\mu \triangleq Ax+b$ and $\eta \triangleq \mathbf{M}^{-1}(\bar{s}-\mathbf{Q}^{-1}\mu)$. Then $\mathcal{N}(\hat{x}|\mathbf{A}x+b,\mathbf{Q})\mathcal{L}(\hat{x}|\bar{r},\bar{s},\bar{\mathbf{L}})$ can be written as

$$\mathcal{N}(\hat{x}|\mathbf{A}x+b,\mathbf{Q})\mathcal{L}(\hat{x}|\bar{r},\bar{s},\bar{\mathbf{L}}) = |2\pi Q|^{-\frac{1}{2}}e^{-\frac{1}{2}f(\hat{x},x)}, \tag{C.2}$$

$$f(\hat{x},x) = (\hat{x}+\eta)^T\mathbf{M}(\hat{x}+\eta) - \eta^T\mathbf{M}\eta + \bar{r} + \mu^T\mathbf{Q}^{-1}\mu.$$

Noting that μ and η are not functions of $\hat{x}$ allows

$$\mathcal{L}(x|r,s,\mathbf{L}) = g(x)\underbrace{\int |2\pi M|^{\frac{1}{2}}e^{-\frac{1}{2}(\hat{x}+\eta)^T\mathbf{M}(\hat{x}+\eta)}\,d\hat{x}}_{=1}, \tag{C.3}$$

$$g(x) = |2\pi(Q-M^{-1})|^{-\frac{1}{2}}e^{-\frac{1}{2}(\bar{r}-\eta^T\mathbf{M}\eta+\mu^T\mathbf{Q}^{-1}\mu)}. \tag{C.4}$$

The unit area holds since the integrand is a Normal PDF. Refactoring $g(x)$ to conform with $\mathcal{L}(x|r,s,\mathbf{L})$ can be achieved using the Woodbury matrix identity $\boldsymbol{\Phi} = \left(\mathbf{I}+\bar{\mathbf{L}}\mathbf{Q}\right)^{-1}\bar{\mathbf{L}} = \mathbf{Q}^{-1} - \mathbf{Q}^{-1}\mathbf{M}^{-1}\mathbf{Q}^{-1}$ and the remaining expressions for $\boldsymbol{\Psi}$, $\boldsymbol{\Gamma}$, $\mathbf{L}$, s and r are straightforwardly determined via linear algebra. □

Proposition 4. *Let the likelihood $\mathcal{L}(x|r,s,\mathbf{L})$ and the Normal PDF $\mathcal{N}(y|\mathbf{C}x+d,\mathbf{R})$ be given. Then*

$$\mathcal{L}(x|\bar{r},\bar{s},\bar{\mathbf{L}}) = \mathcal{N}(y|\mathbf{C}x+d,\mathbf{R})\mathcal{L}(x|r,s,\mathbf{L}), \tag{C.5}$$

where $\bar{\mathbf{L}} = \mathbf{L}+\mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}$, $\bar{s} = s+\mathbf{C}^T\mathbf{R}^{-1}\zeta$, $\bar{r} = r+\zeta^T\mathbf{R}^{-1}\zeta+\ln|2\pi\mathbf{R}|$, *and* $\zeta = d-y$.

Proof. Expanding the exponents and collecting terms gives

$$\mathcal{N}(y|\mathbf{C}x+d,\mathbf{R})\mathcal{L}(x|r,s,\mathbf{L}) = e^{-\frac{1}{2}f(x)}, \tag{C.6}$$

$$f(x) = x^T\underbrace{(\mathbf{L}+\mathbf{C}^T\mathbf{R}^{-1}\mathbf{C})}_{\bar{\mathbf{L}}}x + 2x^T\underbrace{(s+\mathbf{C}^T\mathbf{R}^{-1}\zeta)}_{\bar{s}} + \underbrace{r+\zeta^T\mathbf{R}^{-1}\zeta+\ln|2\pi\mathbf{R}|}_{\bar{r}}. \quad \square$$

Proposition 5. *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*

$$\bar{w}\mathcal{N}(x|\bar{\mu},\bar{\mathbf{P}}) = w\mathcal{N}(x|\mu,\mathbf{P})\mathcal{L}(x|r,s,\mathbf{L}), \tag{C.7}$$

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

$$w\mathcal{N}(x|\mu,\mathbf{P})\mathcal{L}(x|r,s,\mathbf{L}) = \frac{we^{-\frac{1}{2}(\mu^T\mathbf{P}^{-1}\mu+r)}}{|2\pi\mathbf{P}|^{\frac{1}{2}}}e^{-\frac{1}{2}f(x)}, \tag{C.8}$$

$$f(x) = x^T(\mathbf{P}^{-1}+\mathbf{L})x - 2x^T(\mathbf{P}^{-1}\mu - s). \tag{C.9}$$

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

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

References

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

Balenzuela, Mark P., Dahlin, Johan, Bartlett, Nathan, Wills, Adrian G., Renton, Christopher, & Ninness, Brett (2018). Accurate Gaussian mixture model smoothing using a two-filter approach. In *Decision and control, 2018 57th IEEE conference on*. IEEE.

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

Barber, David (2012). *Bayesian reasoning and machine learning*. Cambridge University Press.

Bergman, Niclas, & Doucet, Arnaud (2000). Markov chain Monte Carlo data association for target tracking. In *Acoustics, speech, and signal processing, 2000. ICASSP'00. Proceedings. 2000 IEEE international conference on, Vol. 2* (pp. II705–II708). IEEE.

Blom, Henk A. P., & Bar-Shalom, Yaakov (1988). The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, *33*(8), 780–783.

Costa, Oswaldo Luiz Valle, Fragoso, Marcelo Dutra, & Marques, Ricardo Paulino (2006). *Discrete-time Markov jump linear systems*. Springer Science & Business Media.

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.

Doucet, Arnaud, & Johansen, Adam M. (2011). A tutorial on particle filtering and smoothing: Fifteen years later.

Everitt, Brian S. (2014). Finite mixture distributions. *Wiley StatsRef: Statistics Reference Online*.

Fox, Emily, Sudderth, Erik B., Jordan, Michael I., & Willsky, Alan S. (2011). Bayesian nonparametric inference of switching dynamic linear models. *IEEE Transactions on Signal Processing*, *59*(4), 1569–1585.

Fraser, Donald Charles (1967). *A new technique for the optimal smoothing of data* (Ph.D. thesis), Massachusetts Institute of Technology.

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.

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

Kitagawa, Genshiro (1994). The two-filter formula for smoothing and an implementation of the Gaussian-sum smoother. *Annals of the Institute of Statistical Mathematics*, *46*(4), 605–623.

Lee, Kooktae, Halder, Abhishek, & Bhattacharya, Raktim (2015). Performance and robustness analysis of stochastic jump linear systems using wasserstein metric. *Automatica*, *51*, 341–347.

Ljung, Lennart, & Wills, Adrian (2010). Issues in sampling and estimating continuous-time models with stochastic disturbances. *Automatica*, *46*(5), 925–931.

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.

Mayne, David Q. (1966). A solution of the smoothing problem for linear dynamic systems. *Automatica*, *4*(2), 73–92.

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.

McGill, Robert, Tukey, John W., & Larsen, Wayne A. (1978). Variations of box plots. *The American Statistician*, *32*, 12–16.

Rahmathullah, Abu Sajana, Svensson, Lennart, & Svensson, Daniel (2014). Two-filter Gaussian mixture smoothing with posterior pruning. In *Information fusion (FUSION), 2014 17th international conference on* (pp. 1–8). IEEE.

Runnalls, Andrew R. (2007). Kullback-Leibler approach to Gaussian mixture reduction. *IEEE Transactions on Aerospace and Electronic Systems*, *43*(3).

Whiteley, Nick, Andrieu, Christophe, & Doucet, Arnaud (2010). Efficient Bayesian inference for switching state-space models using discrete particle Markov chain Monte Carlo methods. arXiv preprint arXiv:1011.2437.

Wills, Adrian, Schön, Thomas B., Lindsten, Fredrik, & Ninness, Brett (2012). Estimation of linear systems using a Gibbs sampler. *IFAC Proceedings Volumes*, *45*(16), 203–208.

Mark Peter 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.

Professor 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.