Bayesian learning-based Kalman smoothing for linear dynamical systems with unknown sparse inputs

Chakraborty, Rupam Kalyan; Joseph, Geethu; Murthy, Chandra R.

DOI
10.1109/ICASSP48485.2024.10446534

Publication date
2024

Document Version
Final published version

Published in
2024 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2024 - Proceedings

Citation (APA)
Chakraborty, R. K., Joseph, G., & Murthy, C. R. (2024). Bayesian learning-based Kalman smoothing for linear dynamical systems with unknown sparse inputs. In *2024 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2024 - Proceedings* (pp. 13431-13435). (ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings). IEEE. https://doi.org/10.1109/ICASSP48485.2024.10446534

ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10446534

BAYESIAN LEARNING-BASED KALMAN SMOOTHING FOR LINEAR DYNAMICAL SYSTEMS WITH UNKNOWN SPARSE INPUTS

Rupam Kalyan Chakraborty, Geethu Joseph†, and Chandra R. Murthy**

*ECE Department, Indian Institute of Science, Bengaluru, India
†Signal Processing Systems Group, Delft University of Technology, Delft, The Netherlands
Emails: *{rupamkc, cmurthy}@iisc.ac.in, †g.joseph@tudelft.nl

ABSTRACT

We consider the problem of jointly estimating the states and sparse inputs of a linear dynamical system using noisy low-dimensional observations. We exploit the underlying sparsity in the inputs using fictitious sparsity-promoting Gaussian priors with unknown variances (as hyperparameters). We develop two Bayesian learning-based techniques to estimate states and inputs: sparse Bayesian learning and variational Bayesian inference. Through numerical simulations, we illustrate that our algorithms outperform the conventional Kalman filtering based algorithm and other state-of-the-art sparsity-driven algorithms, especially in the low-dimensional measurement regime.

Index Terms— Kalman smoothing, robust filtering, sparse Bayesian learning, variational Bayesian inference, Gaussian prior

1. INTRODUCTION

Sparse control of linear dynamical systems (LDS) has recently gained considerable research interest [1–5]. This new research area deals with the optimum control of an LDS with sparsity constraints on the control inputs, i.e., the number of nonzero entries in the inputs (active input elements) is small compared to their lengths. Such constraints arise in networked control systems [1, 2], opinion dynamics manipulation [6], computer vision [7, 8], and cyber-physical systems [9, 10]. In such systems, an important goal is to jointly estimate the states and sparse inputs of the LDS from its measurements or output. For example, owing to their compact representations, sparse control inputs are desirable in networked control systems where the controller and plant communicate over a bandwidth-limited channel [11]. Similarly, recovery of malicious attacks on cyber-physical systems, modeled as sparse inputs, is crucial in detecting and mitigating the attacks [9, 10].

Motivated by the above applications, this paper focuses on developing state and input recovery algorithms for observable LDSs with sparse control inputs. Specifically, we consider a discrete-time LDS with state transition matrix $\boldsymbol{A}_k \in \mathbb{R}^{n\times n}$, input matrix $\boldsymbol{B}_k \in \mathbb{R}^{n\times m}$, and the measurement matrices $\boldsymbol{C}_k \in \mathbb{R}^{p\times n}$ and $\boldsymbol{D}_k \in \mathbb{R}^{p\times m}$ at discrete time k,

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{B}_k\boldsymbol{u}_k + \boldsymbol{w}_k \tag{1}$$

$$\boldsymbol{y}_k = \boldsymbol{C}_k\boldsymbol{x}_k + \boldsymbol{D}_k\boldsymbol{u}_k + \boldsymbol{v}_k. \tag{2}$$

Here, $\boldsymbol{u}_k \in \mathbb{R}^m$ is the input, $\boldsymbol{x}_k \in \mathbb{R}^n$ is the state, and $\boldsymbol{y}_k \in \mathbb{R}^p$ is the measurement at time k. Also, $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are noise terms. We aim to simultaneously estimate the states and sparse inputs $\{\boldsymbol{x}_k, \boldsymbol{u}_k : \|\boldsymbol{u}_k\|_0 \ll n\}_{k=1}^K$ from the low dimensional measurements $\{\boldsymbol{y}_k\}_{k=1}^K$ with $p < m$, for a given $K > 0$. Here, $\|\cdot\|_0$ denotes the ℓ_0-norm.

Joint recovery of states and input without assuming any specific structure on the inputs or states has been studied extensively, and several algorithms exist in the control literature [12–16]. However, these algorithms do not account for any underlying sparsity structure that may exist in the inputs. Exploiting sparsity can potentially facilitate the recovery of states and inputs with far fewer measurements than conventional approaches. The limited existing works have solved the problem of jointly recovering the state and sparse input sequences as one of ℓ_1 minimization using convex optimization methods [17]. The necessary and sufficient conditions for observability of sparse control inputs and the initial state for a noiseless LDS have also been investigated [18]. However, the ℓ_1 minimization-based methods involve solving for a large dimensional unknown sparse vector obtained by concatenating the state/input vectors and do not exploit the temporal correlation, for example, in the state evolution model. To address the above gaps in the literature, this paper presents new sparsity-driven estimators with better recovery performance and reduced complexity.

The specific contributions of the paper are as follows: We develop an approach called the sparse robust Kalman smoothing (RKS) that imposes sparsity on the estimated control inputs using the exponential family of prior distributions. We present two techniques to solve the resulting estimation problem: sparse Bayesian learning (SBL)-RKS and variational Bayesian (VB)-RKS. In the first technique, inspired by the SBL framework, we rely on the type-II maximum likelihood estimation combined with Kalman smoothing to determine the states and sparse inputs. In the second technique, we develop a VB-based approach to group the prior parameters and unknown states and inputs as unobserved variables, which are inferred via their posterior distributions. We analyze and derive the time and memory complexities of both approaches. Further, we extend the two approaches to the case of jointly sparse control inputs and present a similar analysis.

2. SPARSE ROBUST KALMAN SMOOTHING

In this section, we present algorithms to estimate state and sparse inputs $\{\boldsymbol{x}_k, \boldsymbol{u}_k\}_{k=1}^K$ from $\{\boldsymbol{y}_k\}_{k=1}^K$ in (1) and (2). We assume that noise $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_k)$ and $\boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k)$ are independent, where $\boldsymbol{Q}_k \in \mathbb{R}^{n\times n}$ and $\boldsymbol{R}_k \in \mathbb{R}^{p\times p}$ are positive definite matrices.

We use a Gaussian prior to promote sparsity in the inputs, $\boldsymbol{u}_k \sim \mathcal{N}\left(\boldsymbol{0}\ \mathrm{Diag}(\boldsymbol{\gamma}_k)\right)$, for $k = 1, 2, \ldots, K$ where $\boldsymbol{\gamma}_k \in \mathbb{R}^m$ is the unknown hyperparameter. We learn the unknown hyperparameters, states, and sparse inputs from the measurements using two approaches: SBL-RKS and VB-RKS, which are described next.

2.1. Sparse Bayesian Learning-based RKS

SBL-RKS first computes the estimate $\hat{\boldsymbol{\gamma}}_k^{\mathrm{ML}}$ of the hyperparameter as

$$\hat{\boldsymbol{\gamma}}_k^{\mathrm{ML}} = \underset{\boldsymbol{\gamma} \in \mathbb{R}_+^{m \times 1}}{\arg\max}\, p\left(\{\boldsymbol{y}_k\}_{k=1}^K ; \boldsymbol{\gamma}\right). \tag{3}$$

Using $\hat{\boldsymbol{\gamma}}_k^{\mathrm{ML}}$, we can estimate the states and inputs using a Kalman filtering and smoothing algorithm. For this, we note that (1) and (2) are equivalent to the following system:

$$\boldsymbol{\xi}_{k+1} = \begin{bmatrix} \boldsymbol{A}_k & \boldsymbol{B}_k \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{\xi}_k + \begin{bmatrix} \boldsymbol{w}_k \\ \boldsymbol{z}_k \end{bmatrix} \text{ and } \boldsymbol{y}_k = \begin{bmatrix} \boldsymbol{C}_k & \boldsymbol{D}_k \end{bmatrix} \boldsymbol{\xi}_k + \boldsymbol{v}_k, \tag{4}$$

where $\boldsymbol{z}_k = \boldsymbol{u}_{k+1}$ is an auxiliary variable and we define

$$\boldsymbol{\xi}_k = \begin{bmatrix} \boldsymbol{x}_k^{\mathsf{T}} & \boldsymbol{u}_k^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}. \tag{5}$$

Also, using the noise covariance $\boldsymbol{Q}_k$ and Gaussian prior on the inputs, we have $\begin{bmatrix} \boldsymbol{w}_k^{\mathsf{T}} & \boldsymbol{z}_k^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \sim \mathcal{N}\left(\boldsymbol{0}, \bar{\boldsymbol{Q}}_k\right)$, where

$$\bar{\boldsymbol{Q}}_k = \begin{bmatrix} \boldsymbol{Q}_k & \boldsymbol{0} \\ \boldsymbol{0} & \mathrm{Diag}\left\{\boldsymbol{\gamma}_{k+1}\right\} \end{bmatrix}, \tag{6}$$

for $k = 1, 2, \ldots, K$. Hence, estimating the states and inputs is equivalent to estimating $\{\boldsymbol{\xi}_k\}_{k=1}^K$ using $\{\boldsymbol{y}_k\}_{k=1}^K$ via the standard Kalman filtering and smoothing due to the Gaussian assumptions. Next, we complete the derivation of SBL-RKS by solving (3).

The optimization problem in (3) does not admit a closed-form solution, so we employ the expectation-maximization (EM) algorithm to solve it. The EM algorithm is an iterative method with E and M steps. In the rth iteration, the E-step computes the expected log-likelihood function $\mathcal{Q}^{(r)}$ of $\{\boldsymbol{\gamma}_k\}_{k=1}^K$ with respect to the current distribution of the measurements $\{\boldsymbol{y}_k\}_{k=1}^K$ and the hidden data $\{\boldsymbol{\xi}_k\}_{k=1}^K$, given the previous estimate $\boldsymbol{\gamma}_k^{(r-1)}$ of the hyperparameter $\boldsymbol{\gamma}_k$. The M-step maximizes $\mathcal{Q}^{(r)}$ to obtain the new estimate $\boldsymbol{\gamma}_k^{(r)}$. To compute $\mathcal{Q}^{(r)}$, we use the state space model in (4) to get

$$p\left(\{\boldsymbol{y}_k, \boldsymbol{\xi}_k\}_{k=1}^K ; \{\boldsymbol{\gamma}_k\}_{k=1}^K\right) = \prod_{k=1}^K p(\boldsymbol{y}_k | \boldsymbol{\xi}_k) p\left(\boldsymbol{\xi}_k \mid \boldsymbol{\xi}_{k-1}; \boldsymbol{\gamma}_k\right), \tag{7}$$

where $\boldsymbol{\xi}_0 = \boldsymbol{0}$. Since $\boldsymbol{\xi}_k = \begin{bmatrix} \boldsymbol{x}_k^{\mathsf{T}} & \boldsymbol{u}_k^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$, the E-step is given by

$$\mathcal{Q}^{(r)}\left(\{\boldsymbol{\gamma}_k\}_{k=1}^K\right) = \sum_{k=1}^K \mathbb{E}_{\boldsymbol{\xi}_k, \boldsymbol{\xi}_{k-1} | \{\boldsymbol{y}_k\}_{k=1}^K ; \boldsymbol{\gamma}_k^{(r-1)}} \{\log(p(\boldsymbol{y}_k | \boldsymbol{\xi}_k) \times p\left(\boldsymbol{x}_k \mid \boldsymbol{\xi}_{k-1}\right) p\left(\boldsymbol{u}_k ; \boldsymbol{\gamma}_k\right))\}. \tag{8}$$

From the above relation, the M-step that maximizes $\mathcal{Q}^{(r)}$ with respect to $\{\boldsymbol{\gamma}_k\}_{k=1}^K$ is separable and simplifies to

$$\boldsymbol{\gamma}_k^{(r)} = \underset{\boldsymbol{\gamma}}{\arg\max}\, \mathbb{E}_{\boldsymbol{u}_k | \{\boldsymbol{y}_k\}_{k=1}^K ; \boldsymbol{\gamma}_k^{(r-1)}} \left\{p\left(\boldsymbol{u}_k ; \boldsymbol{\gamma}\right)\right\}. \tag{9}$$

Using the Gaussian assumption on $\boldsymbol{u}_k$, we derive the M-step as

$$\boldsymbol{\gamma}_k^{(r)} = \underset{\boldsymbol{\gamma}: \boldsymbol{\Gamma} = \mathrm{Diag}\{\boldsymbol{\gamma}\}}{\arg\min} \log|\boldsymbol{\Gamma}| + \mathrm{Tr}\left\{\boldsymbol{\Gamma}^{-1}(\hat{\boldsymbol{u}}_{k|K} \hat{\boldsymbol{u}}_{k|K}^{\mathsf{T}} + \boldsymbol{P}_{k|K}^{\boldsymbol{u}})\right\} \tag{10}$$

$$= \mathrm{Diag}\left\{\hat{\boldsymbol{u}}_{k|K} \hat{\boldsymbol{u}}_{k|K}^{\mathsf{T}} + \boldsymbol{P}_{k|K}^{\boldsymbol{u}}\right\}. \tag{11}$$

Here, $\hat{\boldsymbol{u}}_{t|k} \in \mathbb{R}^p$ and $\boldsymbol{P}_{t|k}^{\boldsymbol{u}} \in \mathbb{R}^{p \times p}$ are the mean and covariance of posterior Gaussian distribution of $\boldsymbol{u}_t$ given $\{\boldsymbol{y}_k\}_{k=1}^K$, for any t and k. They can be computed by applying Kalman filtering and smoothing on the modified state space model in (4). The overall SBL-RKS algorithm is summarized in Algorithm 1.

Algorithm 1 RKS with Sparse Bayesian Learning

Inputs: $\{\boldsymbol{y}_k, \boldsymbol{A}_k, \boldsymbol{B}_k, \boldsymbol{C}_k, \boldsymbol{D}_k, \boldsymbol{Q}_k, \boldsymbol{R}_k\}_{k=1}^K$
Parameters: $\epsilon_{\mathrm{thres}}$ and $r_{\max}$
Initialization: $\boldsymbol{\gamma}_k^{(0)} = \boldsymbol{1}$ for $k = 1, 2, \ldots, K$, $r = 1$, $\epsilon = 2\epsilon_{\mathrm{thres}}$

```
 1: $\bar{\boldsymbol{A}}_k = \begin{bmatrix} \boldsymbol{A}_k & \boldsymbol{B}_k \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \in \mathbb{R}^{(n+m)\times(n+m)}$, $\bar{\boldsymbol{C}}_k = \begin{bmatrix} \boldsymbol{C}_k & \boldsymbol{D}_k \end{bmatrix}$
 2: while $\epsilon > \epsilon_{\mathrm{thres}}$ and $r < r_{\max}$ do
 3:     Compute $\bar{\boldsymbol{Q}}_k$ using (6), for $k = 1, 2, \ldots, K$
      #E-Step:
 4:     $\hat{\boldsymbol{\xi}}_{0|0} = \boldsymbol{0}$, $\boldsymbol{P}_{0|0}^{\boldsymbol{\xi}} = \boldsymbol{I}$
      #Filtering:
 5:     for $k = 1, 2, \ldots, K$ do
 6:         $\hat{\boldsymbol{\xi}}_{k|k-1} = \bar{\boldsymbol{A}}_{k-1} \hat{\boldsymbol{\xi}}_{k-1|k-1}$
 7:         $\boldsymbol{P}_{k|k-1}^{\boldsymbol{\xi}} = \bar{\boldsymbol{A}}_{k-1} \boldsymbol{P}_{k-1|k-1}^{\boldsymbol{\xi}} \bar{\boldsymbol{A}}_{k-1}^{\mathsf{T}} + \bar{\boldsymbol{Q}}_{k-1}$
 8:         $\boldsymbol{G}_k = \boldsymbol{P}_{k|k-1}^{\boldsymbol{\xi}} \bar{\boldsymbol{C}}_k^{\mathsf{T}} \left(\boldsymbol{R}_k + \bar{\boldsymbol{C}}_k \boldsymbol{P}_{k|k-1}^{\boldsymbol{\xi}} \bar{\boldsymbol{C}}_k^{\mathsf{T}}\right)^{-1}$
 9:         $\hat{\boldsymbol{\xi}}_{k|k} = \hat{\boldsymbol{\xi}}_{k|k-1} + \boldsymbol{G}_k \left(\boldsymbol{y}_k - \bar{\boldsymbol{C}}_k \hat{\boldsymbol{\xi}}_{k|k-1}\right)$
10:         $\boldsymbol{P}_{k|k}^{\boldsymbol{\xi}} = \left(\boldsymbol{I} - \boldsymbol{G}_k \bar{\boldsymbol{C}}_k\right) \boldsymbol{P}_{k|k-1}^{\boldsymbol{\xi}}$
11:     end for
      #Smoothing:
12:     for $k = K-1, K-2, \ldots, 1$ do
13:         $\boldsymbol{K}_k = \boldsymbol{P}_{k|k}^{\boldsymbol{\xi}} \bar{\boldsymbol{A}}_k^{\mathsf{T}} \left(\boldsymbol{P}_{k+1|k}^{\boldsymbol{\xi}}\right)^{-1}$
14:         $\boldsymbol{P}_{k|K}^{\boldsymbol{\xi}} = \boldsymbol{P}_{k|k}^{\boldsymbol{\xi}} + \boldsymbol{K}_k \left(\boldsymbol{P}_{k+1|K}^{\boldsymbol{\xi}} - \boldsymbol{P}_{k+1|k}^{\boldsymbol{\xi}}\right) \boldsymbol{K}_k^{\mathsf{T}}$
15:         $\hat{\boldsymbol{\xi}}_{k|K} = \hat{\boldsymbol{\xi}}_{k|k} + \boldsymbol{K}_k \left(\hat{\boldsymbol{\xi}}_{k+1|K} - \bar{\boldsymbol{A}}_k \hat{\boldsymbol{\xi}}_{k|k}\right)$
16:         $\boldsymbol{P}_{k+1,k|K}^{\boldsymbol{\xi}} = \begin{bmatrix} \boldsymbol{P}_{k+1|K}^{\boldsymbol{x}} & \boldsymbol{P}_{k+1|K}^{\boldsymbol{xu}} \end{bmatrix}^{\mathsf{T}} \boldsymbol{K}_k^{\mathsf{T}}$
17:         Compute $\hat{\boldsymbol{u}}_{k|K}$ and $\boldsymbol{P}_{k|K}^{\boldsymbol{u}}$ from $\hat{\boldsymbol{\xi}}_{k|K}$ and $\boldsymbol{P}_{k|K}^{\boldsymbol{\xi}}$ using (5)
18:     end for
      #M-step:
19:     $\boldsymbol{\gamma}_k^{(r)} = \mathrm{Diag}\left\{\hat{\boldsymbol{u}}_{k|K} \hat{\boldsymbol{u}}_{k|K}^{\mathsf{T}} + \boldsymbol{P}_{k|K}^{\boldsymbol{u}}\right\}$, for $k = 1, 2, \ldots, K$
20:     $\epsilon = \sum_{k=1}^K \left\|\boldsymbol{\gamma}_k^{(r)} - \boldsymbol{\gamma}_k^{(r-1)}\right\|^2$, $r \leftarrow r + 1$
21: end while
22: Compute $\{\hat{\boldsymbol{x}}_{k|K}, \hat{\boldsymbol{u}}_{k|K}\}_{k=1}^K$ from $\hat{\boldsymbol{\xi}}_{k|K}$ using (5)
```

Ensure: $\{\hat{\boldsymbol{x}}_{k|K}\}_{k=1}^K$ and $\{\hat{\boldsymbol{u}}_{k|K}\}_{k=1}^K$

When the inputs are jointly sparse, we can use a common prior $\boldsymbol{u}_k \sim \mathcal{N}\left(\boldsymbol{0}, \mathrm{Diag}\{\boldsymbol{\gamma}\}\right)$, i.e., $\boldsymbol{\gamma}_k = \boldsymbol{\gamma}$. The resulting algorithm, called multiple measurement vector SBL-RKS (MSBL-RKS), is identical to Algorithm 1 except for Steps 3 and 19, which change to

$$\bar{\boldsymbol{Q}}_k = \begin{bmatrix} \boldsymbol{Q}_k & \boldsymbol{0} \\ \boldsymbol{0} & \mathrm{Diag}\left\{\boldsymbol{\gamma}\right\} \end{bmatrix}, k = 1, 2, \ldots, K \tag{12}$$

$$\boldsymbol{\gamma}^{(r)} = \frac{1}{K} \sum_{k=1}^K \mathrm{Diag}\left\{\hat{\boldsymbol{u}}_{k|K} \hat{\boldsymbol{u}}_{k|K}^{\mathsf{T}} + \boldsymbol{P}_{k|K}^{\boldsymbol{u}}\right\}. \tag{13}$$

2.2. Variational Bayesian-RKS

In the variational Bayesian approach, we employ a two-stage hierarchical prior. Specifically, we assume $\boldsymbol{u}_k \sim \mathcal{N}\left(\boldsymbol{0}\ \mathrm{Diag}(\boldsymbol{\gamma}_k)\right)$, and $\boldsymbol{\beta}_k(i) \sim \mathrm{Gamma}(a, b)$, where the precision hyperparameter

$\boldsymbol{\beta}_k(i) = 1/\boldsymbol{\gamma}_k(i)$. Here, $\mathrm{Gamma}(a, b)$ is the Gamma distribution with shape parameter $a > 0$ and rate parameter $b > 0$. Unlike the two-step estimation in SBL-RKS (hyperparameter estimation followed by the state-input estimation), VB-RKS estimates the unknowns $\mathcal{Z} = \left\{\boldsymbol{X}_1^K, \boldsymbol{U}_1^K, \{\boldsymbol{\beta}_k\}_{k=1}^K\right\}$ jointly as the mean of their posterior distribution. However, the posterior computation is intractable, and we approximate it using factorized distributions,

$$p(\mathcal{Z}|\{\boldsymbol{y}_k\}_{k=1}^K) \approx q(\mathcal{Z}) = \prod_{k=1}^K q_k^{\boldsymbol{x}}(\boldsymbol{x}_k) q_k^{\boldsymbol{u}}(\boldsymbol{u}_k) q_k^{\boldsymbol{\beta}}(\boldsymbol{\beta}_k), \tag{14}$$

where $q_k^{\boldsymbol{x}}(\cdot), q_k^{\boldsymbol{u}}(\cdot)$, and $q_k^{\boldsymbol{\beta}}(\cdot)$ are the marginal distributions of the latent variables $\boldsymbol{x}_k, \boldsymbol{u}_k$, and $\boldsymbol{\beta}_k$, respectively.

The optimal marginal distribution that minimizes the Kullback-Leibler divergence between the true and factorized posteriors is [19]

$$\log q_k^{\boldsymbol{x}}(\boldsymbol{x}_k) \propto \mathbb{E}_{q(\mathcal{Z}\setminus\boldsymbol{x}_k)} \left\{\log p\left(\mathcal{Z}, \{\boldsymbol{y}_k\}_{k=1}^K\right)\right\}, \tag{15}$$

where $\propto$ denotes equivalence up to an additive constant, and the expectation is with respect to all the latent variables except $\boldsymbol{x}_k$. Further,

$$\begin{aligned} p(\mathcal{Z}, \{\boldsymbol{y}_k\}_{k=1}^K) = \prod_{k=1}^K p(\boldsymbol{y}_k|\boldsymbol{x}_k, \boldsymbol{u}_k) p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) \\ \times p(\boldsymbol{u}_k|\boldsymbol{\beta}_k) p(\boldsymbol{\beta}_k), \end{aligned} \tag{16}$$

where $p(\boldsymbol{u}_k|\boldsymbol{\beta}_k)$ is Gaussian and $p(\boldsymbol{\beta}_k)$ is Gamma distributed. Consequently, we arrive at

$$\begin{aligned} \ln q_k^{\boldsymbol{x}}(\boldsymbol{x}_k) \propto & \|\boldsymbol{y}_k - \boldsymbol{C}_k\boldsymbol{x}_k - \boldsymbol{D}_k\langle\boldsymbol{u}_k\rangle\|_{\boldsymbol{R}_k}^2 \\ & + \|\langle\boldsymbol{x}_{k+1}\rangle - \boldsymbol{A}_k\boldsymbol{x}_k - \boldsymbol{B}_k\langle\boldsymbol{u}_k\rangle\|_{\boldsymbol{Q}_k}^2 \\ & + \|\boldsymbol{x}_k - \boldsymbol{A}_{k-1}\langle\boldsymbol{x}_{k-1}\rangle - \boldsymbol{B}_{k-1}\langle\boldsymbol{u}_{k-1}\rangle\|_{\boldsymbol{Q}_{k-1}}^2, \end{aligned} \tag{17}$$

where $\langle\cdot\rangle$ denotes the mean of a random variable following the marginal distribution derived from $q(\cdot)$. Hence, the marginal distribution $q_k^{\boldsymbol{x}}(\boldsymbol{x}_k)$ is Gaussian whose mean is given by

$$\begin{aligned} \langle\boldsymbol{x}_k\rangle = \boldsymbol{P}_k^{\boldsymbol{x}} \Big[& \boldsymbol{C}_k^T\boldsymbol{R}_k^{-1}\boldsymbol{y}_k + \boldsymbol{Q}_{k-1}^{-1}\boldsymbol{B}_{k-1}\langle\boldsymbol{u}_{k-1}\rangle \\ & - \left(\boldsymbol{C}_k^T\boldsymbol{R}_k^{-1}\boldsymbol{D}_k + \boldsymbol{A}_k^T\boldsymbol{Q}_k^{-1}\boldsymbol{B}_k\right)\langle\boldsymbol{u}_k\rangle \\ & + \boldsymbol{Q}_{k-1}^{-1}\boldsymbol{A}_{k-1}\langle\boldsymbol{x}_{k-1}\rangle + \boldsymbol{A}_k^T\boldsymbol{Q}_k^{-1}\langle\boldsymbol{x}_{k+1}\rangle\Big], \end{aligned} \tag{18}$$

where $\boldsymbol{P}_k^{\boldsymbol{x}} = \left(\boldsymbol{C}_k^T\boldsymbol{R}_k^{-1}\boldsymbol{C}_k + \boldsymbol{Q}_{k-1}^{-1} + \boldsymbol{A}_k^T\boldsymbol{Q}_k^{-1}\boldsymbol{A}_k\right)^{-1}$. Similarly, the marginal distribution of $\boldsymbol{u}_k$ is also Gaussian whose mean in

$$\begin{aligned} \langle\boldsymbol{u}_k\rangle = \boldsymbol{P}_k^{\boldsymbol{u}} \Big[& \boldsymbol{D}_k^T\boldsymbol{R}_k^{-1}\boldsymbol{y}_k - \left(\boldsymbol{D}_k^T\boldsymbol{R}_k^{-1}\boldsymbol{C}_k + \boldsymbol{B}_k^T\boldsymbol{Q}_k^{-1}\boldsymbol{A}_k\right) \\ & \times \langle\boldsymbol{x}_k\rangle + \boldsymbol{B}_k^T\boldsymbol{Q}_k^{-1}\langle\boldsymbol{x}_{k+1}\rangle\Big], \end{aligned} \tag{19}$$

where $\boldsymbol{P}_k^{\boldsymbol{u}} = \left(\boldsymbol{D}_k^T\boldsymbol{R}_k^{-1}\boldsymbol{D}_k + \boldsymbol{B}_k^T\boldsymbol{Q}_k^{-1}\boldsymbol{B}_k + \langle\mathrm{diag}\{\boldsymbol{\beta}_k\}\rangle\right)^{-1}$. We use $\boldsymbol{x}_{K+1} = \boldsymbol{0}$ for $k = K$ and $\boldsymbol{x}_0 = \boldsymbol{0}$ for $k = 1$ in (18) and (19). Finally, $q(\boldsymbol{\beta}_k) = \prod_{i=1}^m q(\boldsymbol{\beta}_k(i))$ is Gamma distributed with mean

$$\langle\boldsymbol{\beta}_k(i)\rangle = \frac{a + 0.5}{b + 0.5\langle\boldsymbol{u}_k^2(i)\rangle} = \frac{a + 0.5}{b + 0.5\left[\langle\boldsymbol{u}_k(i)\rangle^2 + \boldsymbol{P}_k^{\boldsymbol{u}}(i, i)\right]}. \tag{20}$$

Using (18), (19), and (20), the marginal distribution parameters are iteratively updated until convergence to obtain the approximate posterior distribution. The pseudocode is summarized in Algorithm 2.

Algorithm 2 Variational Bayesian RKS

```
Require: $\{\boldsymbol{y}_k, \boldsymbol{A}_k, \boldsymbol{B}_k, \boldsymbol{C}_k, \boldsymbol{D}_k, \boldsymbol{Q}_k, \boldsymbol{R}_k\}_{k=1}^K$
Parameters: $r_{\max}$ and $\tilde{r}_{\max}$
Initialization: $\langle\boldsymbol{x}_k\rangle = \boldsymbol{0}$, $\langle\boldsymbol{u}_k\rangle = \boldsymbol{0}$, $\langle\boldsymbol{\beta}_k\rangle = \boldsymbol{1}$ for $k = 1, \ldots, K$
1: for $r = 1, 2, \ldots, r_{\max}$ do
2:     for $\tilde{r} = 1, 2, \ldots, \tilde{r}_{\max}$ do
3:         Compute $\boldsymbol{x}_k^{(r,\tilde{r})} = \langle\boldsymbol{x}_k\rangle$ using (18) for $k = 1, \ldots, K$
4:         Compute $\boldsymbol{u}_k^{(r,\tilde{r})} = \langle\boldsymbol{u}_k\rangle$ using (19) for $k = 1, \ldots, K$
5:     end for
6:     Compute $\boldsymbol{\beta}_k^{(r)} = \langle\boldsymbol{\beta}_k\rangle$ using (20) for $k = 1, \ldots, K$
7: end for
Ensure: $\left\{\boldsymbol{x}_k^{(r,\tilde{r})}\right\}_{k=1}^K$ and $\left\{\boldsymbol{u}_k^{(r,\tilde{r})}\right\}_{k=1}^K$
```

When all the inputs are jointly sparse, similar to SBL-RKS, we use a common prior $\boldsymbol{u}_k \sim \mathcal{N}(\boldsymbol{0}, \mathrm{Diag}\{\boldsymbol{\beta}\})$, i.e., $\boldsymbol{\beta}_k = \boldsymbol{\beta}$ for $k = 1, 2, \ldots, K$. The VB-RKS for joint sparse input recovery, called multiple measurement vector VB-RKS (MVB-RKS), is identical to Algorithm 2 except that (20) in Step 6 changes as

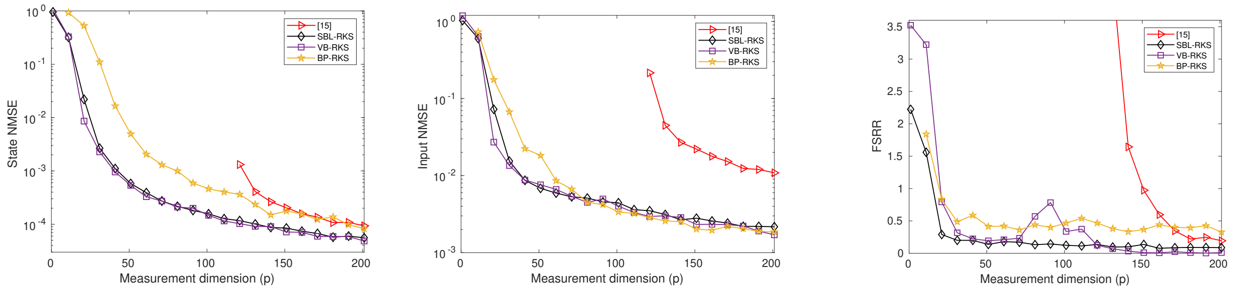
$$\langle\boldsymbol{\beta}(i)\rangle = \frac{a + 0.5}{b + \frac{0.5}{K}\sum_{k=1}^K\langle\boldsymbol{u}_k^2(i)\rangle}. \tag{21}$$

2.3. Complexity Analysis

The SBL-RKS, and VB-RKS are iterative, and each iteration of both the algorithms has time complexity $\mathcal{O}(K(n^3 + m^3 + p^3))$ which scales linearly with K for the versions with and without the joint sparsity assumption. Since the sparsity-driven algorithms consider low-dimensional measurements where $m \geq p$, the time complexity reduces to $\mathcal{O}(K(n^3 + m^3))$. For comparison, we consider the state-of-the-art ℓ_1 minimization-based algorithm, referred to as basis pursuit (BP)-RKS (group BP-RKS for the joint support case). We have derived BP-RKS and group BP-RKS by extending the algorithm in [17] to handle noise and joint support recovery. BP-RKS is a non-iterative algorithm whose complexity scales as $\mathcal{O}(K^{\frac{7}{2}}m^{\frac{3}{2}}p^2 + K(n^3 + p^3))$ due to the convex programming optimization using the interior point method. So, our algorithms have low complexity order when the number of iterations is small. The auxiliary space complexity of SBL-RKS, VB-RKS, and BP-RKS algorithms are $\mathcal{O}(p^2 + K(n^2 + m^2))$, which reduces to $\mathcal{O}(K(n^2 + m^2))$ when $m \geq p$. We omit the details due to space constraints.

3. SIMULATION RESULTS

In this section, we present the empirical results that illustrate the performance of the proposed algorithms. Our setting is as follows. We choose the state dimension $n = 30$, the input dimension $m = 100$, and the number of time steps $K = 30$. The sparsity level of the input is $s = 5$, and the locations of s nonzero entries are chosen uniformly at random from $\{1, 2, \ldots, m\}$. Further, the nonzero entries are drawn independently from a normal distribution $\mathcal{N}(0, \sigma_u^2)$ with $\sigma_u = 5$. For the time-varying support case, we choose different support for each time instant k, and for the jointly sparse case, we use the same support for all values of k. The entries of system matrices, $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$, and the initial state $\boldsymbol{x}_1$ are independently drawn from the standard normal distribution. Also, the process noise covariance $\boldsymbol{Q}$ and the measurement noise covariance $\boldsymbol{R}$ are chosen



(a) NMSE in state estimation

(b) NMSE in input estimation

(c) Percentage of false support recovery

Fig. 1: Performance comparison of our sparse recovery algorithms and RKS as a function of measurement dimension p when the support of control inputs are time-varying, with $n = 30$, $m = 100$, $K = 30$, $s = 5$. The SNR is 20 dB.

Table 1: Run time comparison of the algorithms with $n = 30$, $p = 20$, $m = 100$, $K = 30$, $s = 5$. The SNR is 20 dB.

Support	Algorithm	Runtime	Number of iterations
Time varying	BP-RKS	73.94 s	-
	SBL-RKS	**14.5 s**	$r_{\max}$=30
	VB-RKS	∼5 min	$r_{\max}$=200 $\tilde{r}_{\max}$=200
Jointly sparse	Group BP-RKS	52 s	-
	MSBL-RKS	**13.5 s**	$r_{\max}$=30
	MVB-RKS	∼5 min	$r_{\max}$=200 $\tilde{r}_{\max}$=200

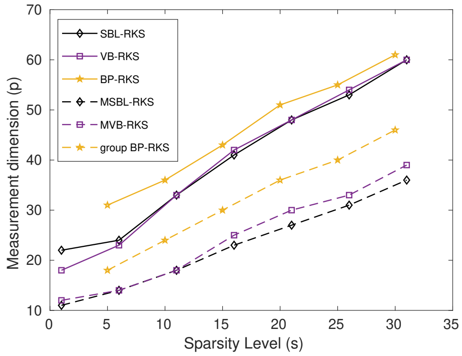


Fig. 2: Phase transition diagram for our sparse recovery algorithms with $n = 30$, $m = 100$, $K = 30$. The SNR is 20 dB.

to be identity matrix and $\sigma_v^2 \boldsymbol{I}$ respectively. Finally, σ_v is computed from the measurement SNR using the relation $\mathrm{SNR} = s\sigma_u^2/\sigma_v^2$.

For this setting, we compare the performance of our algorithms with two benchmark algorithms: (BP)-RKS (group BP-RKS for the joint support case) [17], and the state-of-the-art algorithm in [15], which is a minimum variance unbiased estimator of states and (non-sparse) inputs. We use three metrics for comparison: normalized mean squared error (NMSE) in the state and input estimation, false support recovery rate (FSRR) for input estimation, and run time. Here, FSRR is defined as the ratio of the Hamming distance between the true and estimated supports and the length of the inputs. We define the support as a binary vector with one corresponding to the entries greater than $0.8\sigma_u$ and zeros elsewhere. The results are summarized in Figs. 1 and 2 and Table 1.

From Fig. 1, we infer that the conventional filtering algorithm [15] has poor NMSE performance compared to the sparsity-driven approaches of SBL-RKS, VB-RKS, and BP-RKS. This underscores the importance of exploiting sparsity for accurate state and input estimation. Also, the algorithm in [15] operates in the high-measurement regime as it does not exploit sparsity, and fails in the low-dimensional measurement regime where $p < m$. The NMSE of the filtering algorithm in [15] is comparable to that of the sparsity-driven algorithms only when $p > m$, but the latter outperform [15] even in that regime. We observe similar results for the joint support case, which we omit because of space limitations.

Furthermore, it is evident from Fig. 1 that the SBL-RKS and VB-RKS outperform BP-RKS in terms of NMSE in both state/input estimation and FSRR in the low measurement regime. As we increase the number of measurements, the performances become comparable. This difference is explicitly shown using the phase transition diagram in Fig. 2, which plots the minimum value of the number of measurements p required for 90% recovery accuracy (i.e., successful recovery of the sparse signals in 90% of the random experiments). Here, a sparse vector is said to be successfully recovered if the NMSE between the original signal and the recovered signal is below 0.05. Clearly, the SBL-RKS and VB-RKS algorithms require fewer measurements compared to BP-RKS. The joint sparsity-aware counterpart algorithms follow a similar trend: group BP-RKS requires fewer measurements than the non-joint sparse algorithms; and MVB-RKS and MSBL-RKS outperform group BP-RKS.

From Table 1, we see that SBL-RKS has the best run time. VB-RKS has a similar complexity order as that of SBL-RKS and simple update steps with a shorter run time per iteration. However, it needs a large number of iterations for convergence, resulting in a longer overall run time. Thus, SBL-RKS is a better choice compared to VB-RKS, although both algorithms have a similar recovery performance.

4. CONCLUSION

We studied the joint estimation of the states and sparse inputs of a linear dynamical system. We developed novel algorithms for the joint estimation problem using a Gaussian prior-based hierarchical Bayesian learning approach. We also extended our approaches to the jointly sparse input case and demonstrated utility of exploiting the underlying sparsity. The theoretical guarantees and fundamental limits of sparse recovery algorithms for such linear dynamical systems are interesting directions for future work.

5. REFERENCES

[1] A. Olshevsky, "Minimal controllability problems," *IEEE Trans. Control Network Syst.*, vol. 1, no. 3, pp. 249–258, Sep. 2014.

[2] M. Siami, A. Olshevsky, and A. Jadbabaie, "Deterministic and randomized actuator scheduling with guaranteed performance bounds," *IEEE Trans. Autom. Control*, vol. 66, no. 4, pp. 1686–1701, Jun. 2020.

[3] G. Joseph and C. R. Murthy, "Controllability of linear dynamical systems under input sparsity constraints," *IEEE Trans. Autom. Control*, vol. 66, no. 2, pp. 924–931, Apr. 2020.

[4] G. Joseph, "Controllability of a linear system with nonnegative sparse controls," *IEEE Trans. Autom. Control*, vol. 67, no. 1, pp. 468–473, May 2021.

[5] ——, "Output controllability of a linear dynamical system with sparse controls," *IEEE Trans. Control Netw. Syst.*, vol. 10, no. 1, pp. 147–156, Mar. 2023.

[6] G. Joseph, B. Nettasinghe, V. Krishnamurthy, and P. K. Varshney, "Controllability of network opinion in Erdös-Rényi graphs using sparse control inputs," *SIAM J. Control Optim.*, vol. 59, no. 3, pp. 2321–2345, 2021.

[7] A. Bissacco, A. Chiuso, Y. Ma, and S. Soatto, "Recognition of human gaits," in *Proc. IEEE CVPR*, vol. 2, Dec. 2001, pp. II–II.

[8] M. Raptis, K. Wnuk, and S. Soatto, "Spike train driven dynamical models for human actions," in *Proc. IEEE CVPR*, Jun. 2010, pp. 2077–2084.

[9] A. A. Cárdenas, S. Amin, and S. Sastry, "Research challenges for the security of control systems." *HotSec*, vol. 5, p. 15, Jul. 2008.

[10] J. Slay and M. Miller, "Lessons learned from the Maroochy water breach," in *Proc. Int. J. Crit. Infrastruct. Prot.*, Mar. 2007, pp. 73–82.

[11] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*. Birkhäuser, 2013.

[12] B. Friedland, "Treatment of bias in recursive filtering," *IEEE Trans. Autom. Control*, vol. 14, no. 4, pp. 359–367, Aug. 1969.

[13] P. K. Kitanidis, "Unbiased minimum-variance linear state estimation," *Automatica*, vol. 23, no. 6, pp. 775–778, Nov. 1987.

[14] S. Gillijns and B. De Moor, "Unbiased minimum-variance input and state estimation for linear discrete-time systems," *Automatica*, vol. 43, no. 1, pp. 111–116, Jan. 2007.

[15] ——, "Unbiased minimum-variance input and state estimation for linear discrete-time systems with direct feedthrough," *Automatica*, vol. 43, no. 5, pp. 934–937, May 2007.

[16] G. Joseph and P. K. Varshney, "State estimation of linear systems with sparse inputs and Markov-modulated missing outputs," in *Proc. EUSIPCO*, Aug. 2022, pp. 837–841.

[17] S. Sefati, N. J. Cowan, and R. Vidal, "Linear systems with sparse inputs: Observability and input recovery," in *Proc. ACC*, Jul. 2015, pp. 5251–5257.

[18] K. Poe, E. Mallada, and R. Vidal, "Necessary and sufficient conditions for simultaneous state and input recovery of linear systems with sparse inputs by ℓ_1-minimization," 2023.

[19] D. Nguyen, "An in depth introduction to variational bayes note," *Available at SSRN*, 2023.