Parameter Estimation in Sparse Linear-Gaussian State-Space Models via Reversible Jump Markov Chain Monte Carlo

Benjamin Cox *and* Víctor Elvira
School of Mathematics
University of Edinburgh
Edinburgh, United Kingdom
benjamin.cox@ed.ac.uk, victor.elvira@ed.ac.uk

***Abstract*—State-space models are ubiquitous for modelling complex systems that evolve over time. In such models, key parameters are usually unknown and must be estimated. In particular, linear systems are parametrised by the transition matrix that encodes the dependencies among state dimensions. Due to physical and computational constraints, it is desirable to estimate this matrix by promoting sparsity in its components, in such a way that the interactions between elements of the state-space are reduced. In this work, we propose a novel Bayesian methodology to estimate model parameters and promote sparsity. The method, called SpaRJ, is based on reversible jump Markov chain Monte Carlo, and allows for the exploration of the space of sparse transition matrices in an efficient manner via the adaptation of the implicit model dimension. The methodology has strong theoretical guarantees and exhibits good performance in numerical examples.**

***Index Terms*—Bayesian methods, graphical inference, Kalman filtering, parameter estimation, sparsity detection, state-space modelling.**

I. Introduction

State-space modelling allows for a flexible description and statistical analysis of dynamic systems. These models incorporate both a hidden state that evolves over time, and a sequence of observations that are linked to the hidden states [1]. The state dynamics are often considered to be Markovian.

In some cases, the state-space model (SSM) is perfectly described, with the inferential goal of estimating the sequence of underlying hidden states. In the Bayesian paradigm, this estimation is done in a probabilistic manner, by providing a sequence of posterior distributions [2]. For a given time-step, if only past and present observations are used in the estimation procedure, this is known as a filtering problem. If all dynamics are linear and the noises are Gaussian, we can exactly compute the filtering distributions [3]. This linear-Gaussian state-space model (LGSSM) provides a relatively simple framework, which is useful when modelling linear systems, or those where linearisation is reasonable. Where this is not appropriate, extensions to non-linear Gaussian dynamics, such as the extended Kalman filter [4] and the unscented Kalman filter [5], are typically used. For even more generic SSMs, such as those with non-Gaussian noises, there exist particle methods that allow for the approximation of the filtering distribution via a number of Monte Carlo samples [6], [7], [8]. Note that, however, all of these methods assume the model parameters to be known.

In practice, the model parameters are often unknown, and must be estimated. Obtaining estimates of these parameters is in general a difficult task [2], [7], [9]. Also note that complex dynamic systems are often composed of many interacting simple units, with each of those units directly affected by only a subset of the rest [10]. Therefore, when estimating the parameters in such models, it is desirable to obtain sparse estimates, which admit several advantages [11]. First, by promoting sparsity, we improve the quality of the inference, since it translates into reducing the dimension of the parameter space. Second, the sparsity also gives an interpretation of the connectivity of the state variables in the line of graphical modelling methods [12], which are used in diverse applications such as biology [13], social networks [14], and neuroscience [15]. Third, sparse estimates allow us to recover this 'interacting blocks' structure.

In this work, we propose the *sparse reversible jump* (SpaRJ) method, a novel Markov chain Monte Carlo (MCMC) algorithm for estimating the transition matrix of a LGSSM. The proposed method allows for the inclusion of prior knowledge of the transition matrix, and in particular, we aim at recovering a sparse matrix. From a Bayesian perspective, we obtain true sparse samples from a posterior distribution of this matrix, unlike in traditional approaches. This is achieved by implementing a reversible jump MCMC (RJMCMC) [16] that runs a Kalman filter for each proposed sample, thus allowing to not only obtain true sparse samples but to also explore different levels of sparsity in the transition matrix. The strong theoretical guarantees of RJMCMC are inherited in the proposed SpaRJ algorithm. In addition, SpaRJ exhibits excellent performance in challenging numerical experiments of increasing dimension.

The rest of this work is structured as follows. In Section II we present the problem and background. In Section III we present our method for sparse Bayesian inference in LGSSMs. In Section IV we provide numerical results for our method.

B.C. acknowledges support from the *Natural Environment Research Council* of the UK through a SENSE CDT studentship (NE/T00939X/1). V.E. acknowledges support from the *Agence Nationale de la Recherche* of France under PISCES (ANR-17-CE40-0031-01) project.

II. Background

A. State-Space Modelling and Filtering

Let us consider the additive linear-Gaussian (LG) state-space model (SSM) given by

$$\begin{aligned} \mathbf{x}_t &= \mathbf{A}\mathbf{x}_{t-1} + \mathbf{q}_t, \\ \mathbf{y}_t &= \mathbf{H}\mathbf{x}_t + \mathbf{r}_t, \end{aligned} \tag{1}$$

for $t = 1, \ldots, T$, where $\mathbf{x}_t \in \mathbb{R}^{d_x}$ is the hidden state with associated observation $\mathbf{y}_t \in \mathbb{R}^{d_y}$ at time t, $\mathbf{A} \in \mathbb{R}^{d_x \times d_x}$ is the transition matrix, $\mathbf{H} \in \mathbb{R}^{d_y \times d_x}$ is the observation matrix, $\mathbf{q}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ is the state noise, and $\mathbf{r}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ is the observation noise. The state prior is $\mathbf{x}_0 \sim \mathcal{N}(\bar{\mathbf{x}}_0, \mathbf{P}_0)$, with $\bar{\mathbf{x}}_0$ and $\mathbf{P}_0$ typically assumed to be known.

A common task in state-space modelling is the estimation of the distribution of $\mathbf{x}_t$ conditional on $\mathbf{y}_{1:t}$, $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. In the case of the LGSSM, this distribution can be obtained using the Kalman filter [2], [3]. From these equations, we obtain the conditional likelihood of the observations, which allows us to estimate the model parameters where they are unknown [2]. In LGSSMs, $\mathbf{H}$ and $\mathbf{R}$ are often assumed to be known parameters of the observation instrument, but $\mathbf{Q}$ and $\mathbf{A}$ are often unknown. For the purposes of this work, we assume that all parameters except $\mathbf{A}$ are known, or are suitably estimated.

B. Parameter Estimation in State-Space Models

The estimation of the parameters of a state-space model is in general a difficult task, as the likelihoods and dependency structures involved are, in general, quite complex [2], [17]. Many methods are used, which we can broadly classify as point estimation methods and distributional methods. Our method is a Bayesian distributional method, which gives several advantages.

Distributional methods estimate the probability density of the parameters given the data, i.e., $p(\mathbf{A}|\mathbf{y}_{1:T})$ in our case, often through the generation of Monte Carlo samples. These samples can be used to compute estimates of various statistics of the parameters, such as mean and (co)variance. Markov chain Monte Carlo (MCMC) methods are often used to obtain these samples [17]. MCMC comprises a class of sampling methods that construct a Markov chain that converges in distribution to the desired distribution at equilibrium [18]. The elements of this chain are taken to be Monte Carlo samples from the desired distribution.

C. Sparse Modelling

Sparsity is a desirable property when fitting and designing statistical models. This is due to sparsity being ingrained in the model selection process for nearly all models. For example, a linear regression in which terms are excluded based on them being insignificant is also a sparse model. Several estimation methods allow for the systematic detection of sparsity [19], [20], permitting the simplification of complex models and giving insight into the behaviour of the system [21], [22].

There are several well studied approaches to estimate models with sparsity. A common approach is to estimate the model parameters under a sparsity inducing penalty, with the classic example of such a penalty being the LASSO [19]. This approach is prevalent, with several extensions to Bayesian modelling in the form of sparsity inducing priors [20].

Sparsely estimating the state transition matrix $\mathbf{A}$ of a LGSSM allows for the resulting estimate to be interpreted as a weighted directional graph of the between-step connectivity of the hidden states [11]. This can be used further to recover Granger-causal relationships between the state variables [23], or to determine an optimal linear combination of the state variables for dimensionality reduction.

D. Reversible Jump Markov Chain Monte Carlo

Reversible jump Markov chain Monte Carlo (RJMCMC) [16] was first proposed as a method for Bayesian model selection, and has since seen use in fields such as ecology [24], Gaussian mixture modelling [25], and hidden Markov modelling [26].

RJMCMC is primarily used to explore and sample variable dimension models, where the model M, and hence the dimension of the associated parameter space Θ, is unknown. Let $\Theta^{(i)}$ be the parameter space associated with model $M^{(i)}$, and $\theta^{(i)} \in \Theta^{(i)}$ an associated realisation of the model parameters. The probability of jumping from model $M^{(i)}$ to $M^{(j)}$ must be calculable a priori, and is denoted by $\pi_{i,j}$. Let $M^{(1)}$ be the current model, and $M^{(2)}$ be a candidate model that we could potentially jump to. When moving to a lower or equally dimensional space, i.e., $\dim(\Theta^{(1)}) \geq \dim(\Theta^{(2)})$, the parameters can be mapped deterministically via

$$\theta^{(2)} = T_{1,2}(\theta^{(1)}),$$

where $T_{1,2}$ is a bijection [18]. If moving to a higher dimensional space, i.e., $\dim(\Theta^{(1)}) < \dim(\Theta^{(2)})$, then no bijection between $\Theta^{(1)}$ and $\Theta^{(2)}$ exists. Reversible jump MCMC addresses this by introducing a stochastic component [16]; the spaces $\Theta^{(1)}$ and $\Theta^{(2)}$ are augmented with simulated draws from selected distributions such that

$$(\theta^{(2)}, u_2) = T_{1,2}(\theta^{(1)}, u_1), \quad u_1 \sim g_{1,2}(\cdot),\ u_2 \sim g_{2,1}(\cdot),$$

where $T_{1,2}$ is a bijection and $g_{i,j}(\cdot)$ are distributions chosen a priori.

These mappings and stochastic draws alter the properties of the Markov chain, and if not corrected the chain will not exhibit detailed balance. In order to correct for this the acceptance ratio for the jump from model $M^{(1)}$ to model $M^{(2)}$ is modified, and is given by

$$\alpha^{(1,2)} = \left| \frac{\partial T_{1,2}(\theta^{(1)}, u_1)}{\partial(\theta^{(1)}, u_1)} \right| \frac{g_{2,1}(u_2)}{g_{1,2}(u_1)} \frac{\pi_{2,1}}{\pi_{1,2}} \frac{p_2(\theta^{(2)})}{p_1(\theta^{(1)})}, \tag{2}$$

where $p_i(\theta^{(i)})$ is the density associated with model $M^{(i)}$ evaluated at $\theta^{(i)}$. We can incorporate the $\pi_{i,j}$ ratio into the prior, as it is calculable a priori. The jump from model 1 to model 2 is accepted with probability $\min(\alpha^{(1,2)}, 1)$, and is otherwise rejected. On rejection, the current value of the chain is kept, as standard for a Metropolis type algorithm.

E. Model Definitions and Notation

We denote by M_n the model selected by the algorithm at iteration n. This model is uniquely defined by $\mathcal{M}_n$, the list of indices of dense elements in $\mathbf{A}_n$. The number of elements of $\mathcal{M}_n$, denoted by $|\mathcal{M}_n|$, is therefore the number of dense elements at iteration n, or the number of dense elements of $\mathbf{A}_n$.

Table I
NOTATION REFERENCE

Notation	Meaning
M_n	Model at iteration n
$\mathcal{M}_n$	Set of pairs of indices of dense elements in M_n
$\lvert\mathcal{M}_n\rvert$	Number of dense elements at iteration n

III. THE SPARJ ALGORITHM

We now present the SpaRJ algorithm, a novel RJMCMC method that sparsely estimates the posterior of the transition matrix of a linear-Gaussian state-space model of the form in Eq. (1). The method is systematically described below.

A. Algorithmic Description of SpaRJ

The algorithm is initialised at the fully dense model M_0. We obtain the initial log-likelihood l_0 by running a Kalman filter with $\mathbf{A}_0$, yielding $l_0 = \log(p(\mathbf{y}_{1:T}|\mathbf{A}_0))$.

The method iterates N times to obtain N samples $\{\mathbf{A}_n\}_{n=1}^N$. Each iteration is split into three steps: model proposal (Step 1), parameter proposal (Step 2), and accept/reject (Step 3), after which the next iteration begins.

Step 1: Propose M^*. At iteration n, the previous model M_{n-1} is retained with probability (w.p.) π_0. If the previous model is not retained (w.p. $1-\pi_0$), then we choose the proposed model M^* to be sparser w.p. π_{-1}, and denser otherwise. We find that $\pi_0 = 0.8$ gives an acceptance rate close to the optimal for Random Walk Metropolis-Hastings (RWMH) of 0.234 [27]. In general, we recommend using $\pi_{-1} = 0.5$ so that the walk over the model space is unbiased.

Step 2: Propose $\mathbf{A}^*$. The method to propose $\mathbf{A}^*$ is determined by M^*. If the proposed model M^* is the same as the previous model M_{n-1}, then A^* is sampled from the conditional posterior $p(\mathbf{A}|M^*)$. Our preferred sampling method is a RWMH sampler, chosen for simplicity and extensibility. In addition, the RWMH sampler only requires a single run of the Kalman filter per iteration, the evaluation of which is the most computationally expensive component of the algorithm. We perturb the parameters using an element-wise Laplace random walk, which admits the following element-wise distribution for the parameter proposal:

$$(a_n)_{ij} \sim \begin{cases} \text{Laplace}((a_{n-1})_{ij}, \sigma), & (i,j) \in \mathcal{M}_n, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

If the proposed model M^* is different from the previous model M_{n-1}, then the parameters θ_{n-1} are identically mapped to θ^*, augmented with stochastic draws if the dimension of the parameter space increases, and with selected elements removed if the dimension decreases. This identity mapping turns the Jacobian term in Eq. (2) equal to 1. We draw from a Laplace$(0, \sigma)$ distribution.

Step 3: Metropolis accept-reject. Once the model and parameter values have been proposed, a Metropolis-Hastings acceptance step is performed. This accept-reject is independent of any that may have taken place in the previous step. We run a Kalman filter with $\mathbf{A}^*$ and extract the log-likelihood of the proposed sample, $l^* = \log(p(\mathbf{y}_{1:T}|\mathbf{A}^*))$. We then construct the log-acceptance ratio, $\log(\alpha) = l^* - l_{n-1} + \lambda(\|\mathbf{A}_{n-1}\|_1 - \|\mathbf{A}^*\|_1) + c$, where the c term is the distributional term from Eq. (2).

Algorithm 1 SpaRJ algorithm

Input: $\mathbf{y}_{1:T}, \mathbf{A}_0, \pi_0, \pi_{-1}, N, \lambda, \sigma, \mathbf{P}_0, \mathbf{Q}, \mathbf{R}, \mathbf{H}, \bar{\mathbf{x}}_0$
Output: Set of N samples $\{\mathbf{A}_n, l_n, M_n\}_{n=1}^N$
Initialisation
Initialise M_0 as fully dense
Evaluate Kalman filter with $\mathbf{A}_0$, obtaining $l_0 := \log(\mathrm{p}(\mathbf{y}_{1:T}|\mathbf{A}_0))$
for $n = 1, \ldots, N$ **do**
  Set $c := 0$.
  Step 1: Propose model
  Retain w.p. π_0
  if *Retain* **then**
    Set $M^* := M_{n-1}$
  else
    Step 1.1: Determine jump direction
    if $\mathbf{A}_{n-1}$ Densest **OR** Sparsest **then**
      Jump sparser or denser respectively
    else
      Jump sparser with probability π_{-1}, otherwise jump denser
    end if
    Step 1.2: Perform jump
    if *Jump sparser* **then** (Step 1.2-s)
      Select $e \in \mathcal{M}_{n-1}$
      Set M^* such that $\mathcal{M}^* = \mathcal{M}_{n-1} \setminus e$
    else (Step 1.2-d)
      Select $e \in \mathcal{M}_0 \setminus \mathcal{M}_{n-1}$
      Set M^* such that $\mathcal{M}^* = \mathcal{M}_{n-1} \cup e$
    end if
  end if
  Step 2: Propose $\mathbf{A}^*$
  if *Retain* **then**
    Step 2.1: Sample posterior
    Propose $\mathbf{A}^*$ using Eq. (3)
  else
    Step 2.2: Map parameters
    if *Jump sparser* **then** (Step 2.2-s)
      Set $\mathbf{A}^*$ to $\mathbf{A}_{n-1}$ with element a_e set to 0.
      Set $c := \log(\text{Laplace}(a_e; 0, \sigma))$.
    else (Step 2.2-d)
      Draw $u \sim \text{Laplace}(0, \sigma)$.
      Set $\mathbf{A}^*$ to $\mathbf{A}_{n-1}$ with element a_e set to u.
      Set $c := -\log(\text{Laplace}(u; 0, \sigma))$.
    end if
  end if
  Step 3: MH accept-reject
  Evaluate Kalman filter with $\mathbf{A} := \mathbf{A}^*$, obtaining $l^* := \log(p(\mathbf{y}_{1:T}|\mathbf{A}^*))$
  $\log(\alpha) := l^* - l_{n-1} + \lambda(\|\mathbf{A}_{n-1}\|_1 - \|\mathbf{A}^*\|_1) + c$
  Accept with probability α
  if *Accept* **then**
    Set $M_n := M^*, \mathbf{A}_n := \mathbf{A}^*, l_n := \log(p(\mathbf{y}_{1:T}|\mathbf{A}^*))$
  else
    Set $M_n := M_{n-1}, \mathbf{A}_n := \mathbf{A}_{n-1}, l_n := l_{n-1}$
  end if
end for

B. Incorporating Prior Knowledge

The hyper-parameters that must be chosen are: π_0, the probability of retaining the current model, π_{-1}, the probability of proposing a sparser model given the current model is not retained, λ, the penalty parameter, and σ, the regeneration scale.

We use a LASSO penalty, which is equivalent to a Laplace prior [20]. We incorporate our prior belief in the value and sparsity of $\mathbf{A}$ via this penalty. This penalty could be changed without modification to the wider algorithm, and is not necessary

for the recovery of sparsity, which is due to the model jumping. For the LASSO penalty we find $\lambda \in [0.1, 1]$ works well.

Prior knowledge is also incorporated in the selection of the π_0 and π_{-1} hyper-parameters. π_{-1} controls the bias of the sampler, directing it to sparser models when $\pi_{-1} > 0.5$ and denser models when $\pi_{-1} < 0.5$. Therefore if the matrix is believed to be very sparse π_{-1} can be chosen to reflect this. If there is no prior knowledge of the sparsity we recommend $\pi_{-1} := 0.5$. The π_0 parameter determines the average time spent under an accepted model. A larger value leads to better within-model behaviour, as on average more samples will be obtained, but requires more iterations to explore sparsity, as the model is switching less often. We recommend $\pi_0 := 0.8$ as we find this yields an average within-model acceptance rate close to the optimum for RWMH of 0.234 [27].

C. Inherited Guarantees

As our algorithm is a RJMCMC method, which themselves fall under Metropolis-Hastings methods [28], we inherit a number of theoretical guarantees. First, every within-model sampling density converges to the true conditional within-model distribution. This follows from use of a symmetric RWMH sampler for the within-model sampling, which is well studied and known to exhibit the required properties of ergodicity and detailed balance.

Second, our overall sampling Markov chain exhibits detailed balance. This follows from the design of the RJMCMC transition kernel [16]. This property means that the limiting distribution of the Markov chain is exactly the desired distribution. In addition, the model sampling chain is ergodic, as all elements of the model space can be reached from all other elements in a finite number of moves. This is composed with the ergodicity of the within-model sampler to obtain the ergodicity, and hence distributional correctness, of the overall sampling chain.

D. Discussion

Whilst our method inherits asymptotic guarantees from RJMCMC, these properties do not provide a guarantee of convergence in finite time. We find that $15,000$ iterations yields good results in moderately-high dimensional cases, specifically $d_x < 50$. Our method is also relatively inexpensive computationally, requiring minimal additional computation when compared to a standard MCMC method applied to state-space models. This small amount of overhead, occurring as a result of drawing from and evaluating the density of explicitly known distributions, allow for us to explore the sparsity of a LGSSM in a Bayesian framework. This capability is, to our knowledge, new within the domain of state-space modelling. This method hence allows for inference that was not previously possible, such as the probabilistic quantification of sparsity in a Bayesian manner, as well as the distribution of sparse elements.

IV. Numerical Study

We generate observations following Eq. (1), with $d_x = d_y$, $\mathbf{H} = \mathbf{Id}_{d_x}$, $\bar{\mathbf{x}}_0 = \mathbf{1}_{d_x}$, $T = 100$ The state covariance matrix $\mathbf{Q}$ is specified per study. In these experiments we run SpaRJ for $N = 15,000$ iterations, discarding the first $5,000$ as burn-in. $\mathbf{A}_0$ is generated using an EM scheme, itself initialised with a random element-wise standard normal matrix. We set $\pi_0 = 0.8$ and $\pi_{-1} = 0.5$ in all cases.

We compare the proposed method with GraphEM [11], an algorithm with similar goals based on proximal optimization. We compare the metrics of precision, recall, specificity, and F1 score, as these are common metrics for sparsity detection as a subset of classification. We take an element to be sparse under our method by majority vote of the samples. In all cases we average the metrics over 200 independent runs of each algorithm.

A. Known Isotropic State Covariance

We generate the $\mathbf{A}$ matrix for each system by drawing from an element-wise standard normal distribution, insert some sparsity in the matrix, and then dividing by the magnitude of the maximal singular value, which ensures the stability of the system in each run. We test the performance of the method with isotropic covariance matrices $\mathbf{Q}$ and $\mathbf{R}$.

Dimension 3 matrix. We generate $\mathbf{A}$ for dimension $d_x = 3$, with sparsity in one element per row and per column. We set $\mathbf{Q}_{(\text{true})} = \mathbf{R}_{(\text{true})} = \mathbf{Id}_{d_x}, \mathbf{P}_{(\text{true})} = 10^{-8}\mathbf{Id}_{d_x}$, and $\lambda = 1$.

Dimension 6 block diagonal matrix. We generate $\mathbf{A}$ for dimension $d_x = 6$ as a block diagonal matrix with three 2×2 blocks. We set $\mathbf{Q}_{(\text{true})} = \mathbf{R}_{(\text{true})} = 10^{-2}\mathbf{Id}_{d_x}$, $\mathbf{P}_{(\text{true})} = 10^{-8}\mathbf{Id}_{d_x}$, and $\lambda = 0.367 \approx \exp(-1)$.

Dimension 12 block diagonal matrix. We generate $\mathbf{A}$ for dimension $d_x = 12$ as a block diagonal matrix with six 2×2 blocks. We set $\mathbf{Q}_{(\text{true})} = \mathbf{R}_{(\text{true})} = 10^{-2}\mathbf{Id}_{d_x}$, $\mathbf{P}_{(\text{true})} = 10^{-8}\mathbf{Id}_{d_x}$, and $\lambda = 0.367$.

Table II
SPARSITY STATISTICS OVER VARIABLE SYSTEMS.

transition matrix structure	method	spec.	recall	prec.	F1
3×3 matrix	GraphEM	0.86	0.98	0.79	0.88
	SpaRJ	**0.98**	**0.99**	**0.99**	**0.99**
6×6 block diagonal	GraphEM	0.83	0.90	0.91	0.91
	SpaRJ	**0.88**	**0.96**	**0.94**	**0.95**
12×12 block diagonal	GraphEM	**0.85**	0.77	**0.96**	0.85
	SpaRJ	0.83	**0.89**	0.91	**0.90**

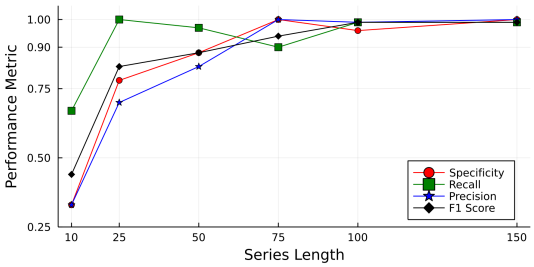

Figure 1. Sparsity statistics over variable series length T for the 3×3 system. Note the rapid convergence to nearly optimal metrics.

For the 3×3 system we also demonstrate our method for different values of $T \in [10, 150]$. In Figure 1, we show averaged metrics over 100 independent runs for both methods. We see that

the longer the series the better the overall performance, showing numerically the convergence to the true distribution.

Table II shows a good performance of SpaRJ and its capability to extract the sparsity structure in all examples. Figure 1 also exhibits the desired increase in performance as more observations are collected.

B. Unknown Anisotropic State Covariance

In many scenarios, the true value of the state covariance $\mathbf{Q}$ is unknown. We generate anisotropic covariance matrices $\mathbf{Q}_{(\mathrm{true})}$ following [29], with Uniform$(0.5, 1.5)$ eigenvalues.

We run an EM algorithm to estimate $\mathbf{A}$ and $\mathbf{Q}$ by $\mathbf{A}^{\mathrm{EM}}$ and $\mathbf{Q}^{\mathrm{EM}}$, respectively. We initialise the EM algorithm at the identity, and then compute the MAP estimates, alternately estimating $\mathbf{A}$ and $\mathbf{Q}$ until $\|\mathbf{A}_n^{\mathrm{EM}} - \mathbf{A}_{n-1}^{\mathrm{EM}}\|_2 + \|\mathbf{Q}_n^{\mathrm{EM}} - \mathbf{Q}_{n-1}^{\mathrm{EM}}\|_2 \leq 10^{-3}$ is satisfied. We set $\mathbf{A}_0 = \mathbf{A}^{\mathrm{EM}}$. We set $\mathbf{Q} = \mathbf{Q}^{\mathrm{EM}}$, and perform inference on $\mathbf{A}$ under this value of $\mathbf{Q}$. Other than $\mathbf{Q}$, all parameters are set as previously.

Table III
SPARSITY STATISTICS OVER VARIABLE SYSTEMS WITH ESTIMATED ANISOTROPIC COVARIANCES.

transition matrix structure	method	spec.	recall	prec.	F1
3×3 matrix	GraphEM	0.75	0.72	0.62	0.65
	SpaRJ	**0.87**	**0.98**	**0.80**	**0.89**
6×6 block diagonal	GraphEM	0.68	0.38	0.70	0.49
	SpaRJ	**0.75**	**0.53**	**0.81**	**0.63**
12×12 block diagonal	GraphEM	**0.76**	0.34	0.88	0.49
	SpaRJ	0.6	**0.53**	**0.88**	**0.65**

We see that our method performs well under these challenging conditions, consistently outperforming existing methods. The performance degrades significantly in this scenario, but our method retains the edge in statistical performance. The performance degradation could be lessened by using better estimates of $\mathbf{Q}$; this is intended to show a worst-case scenario.

V. CONCLUSION

In this work, we have proposed the SpaRJ algorithm, a new Markov-chain Monte Carlo method for obtaining sparse samples of the transition matrix encoding hidden state relationships in linear-Gaussian state-space models. Using reversible jump MCMC along with a structured model space allows for exploration and encapsulation of sparsity in the resulting samples. The algorithm retains strong theoretical guarantees, shows great potential for extension, and numerical results evidence that the method performs well in challenging scenarios.

REFERENCES

[1] J. Durbin and S. J. Koopman, *Time series analysis by state space methods*. Oxford university press, 2012.

[2] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge University Press, 2013, no. 3.

[3] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[4] G. A. Einicke and L. B. White, "Robust extended Kalman filtering," *IEEE Transactions on Signal Processing*, vol. 47, no. 9, pp. 2596–2599, 1999.

[5] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*. IEEE, 2000, pp. 153–158.

[6] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle filtering," *IEEE signal processing magazine*, vol. 20, no. 5, pp. 19–38, 2003.

[7] A. Doucet, A. M. Johansen *et al.*, "A tutorial on particle filtering and smoothing: Fifteen years later," *Handbook of nonlinear filtering*, vol. 12, no. 656-704, p. 3, 2009.

[8] V. Elvira, L. Martino, M. F. Bugallo, and P. M. Djuric, "Elucidating the auxiliary particle filter via multiple importance sampling [lecture notes]," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 145–152, 2019.

[9] S. Sharma, A. Majumdar, V. Elvira, and E. Chouzenoux, "Blind kalman filtering for short-term load forecasting," *IEEE Transactions on Power Systems*, vol. 35, no. 6, pp. 4916–4919, 2020.

[10] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[11] E. Chouzenoux and V. Elvira, "GraphEM: EM algorithm for blind Kalman filtering under graphical sparsity constraints," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 5840–5844.

[12] M. Eichler, "Graphical modelling of multivariate time series," *Probability Theory and Related Fields*, vol. 153, no. 1, pp. 233–268, 2012.

[13] D. Luengo, G. Rios-Munoz, V. Elvira, C. Sanchez, and A. Artes-Rodriguez, "Hierarchical algorithms for causality retrieval in atrial fibrillation intracavitary electrograms," *IEEE Journal of Biomedical and Health Informatics*, vol. 12, no. 1, pp. 143–155, Jan. 2019.

[14] C. Ravazzi, R. Tempo, and F. Dabbene, "Learning influence structure in sparse social networks," *IEEE Transactions on Control of Network Systems*, vol. PP, pp. 1–1, 12 2017.

[15] J. Richiardi, S. Achard, B. Horst, , and D. V. D. Ville, "Machine learning with brain graphs," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 58–70, 2013.

[16] P. J. Green, "Reversible jump Markov chain Monte Carlo computation and bayesian model determination," *Biometrika*, vol. 82, no. 4, pp. 711–732, 1995.

[17] C. Andrieu, A. Doucet, and R. Holenstein, "Particle Markov chain Monte Carlo methods," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 3, pp. 269–342, 2010.

[18] C. Robert and G. Casella, *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[19] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[20] T. Park and G. Casella, "The Bayesian lasso," *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 681–686, 2008.

[21] A. J. Smola and B. Schölkopf, "Sparse greedy matrix approximation for machine learning." Morgan Kaufmann, 2000, pp. 911–918.

[22] L. El Ghaoui, V. Pham, G.-C. Li, V.-A. Duong, A. Srivastava, and K. Bhaduri, "Understanding large text corpora via sparse machine learning," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 6, no. 3, pp. 221–242, 2013. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sam.11187

[23] J. Runge, S. Bathiany, E. Bollt, G. Camps-Valls, D. Coumou, E. Deyle, C. Glymour, M. Kretschmer, M. D. Mahecha, J. Muñoz-Marí *et al.*, "Inferring causation from time series in earth system sciences," *Nature communications*, vol. 10, no. 1, pp. 1–13, 2019.

[24] M. Pagel and A. Meade, "Bayesian analysis of correlated evolution of discrete characters by reversible-jump markov chain monte carlo," *The American Naturalist*, vol. 167, no. 6, pp. 808–825, 2006.

[25] S. Richardson and P. J. Green, "On bayesian analysis of mixtures with an unknown number of components (with discussion)," *Journal of the Royal Statistical Society: series B (statistical methodology)*, vol. 59, no. 4, pp. 731–792, 1997.

[26] O. Cappé, C. P. Robert, and T. Rydén, "Reversible jump, birth-and-death and more general continuous time Markov chain Monte Carlo samplers," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 65, no. 3, pp. 679–700, 2003.

[27] A. Gelman, W. R. Gilks, and G. O. Roberts, "Weak convergence and optimal scaling of random walk Metropolis algorithms," *The Annals of Applied Probability*, vol. 7, no. 1, pp. 110 – 120, 1997. [Online]. Available: https://doi.org/10.1214/aoap/1034625254

[28] L. Martino and V. Elvira, "Metropolis sampling," *arXiv preprint arXiv:1704.04629*, 2017.

[29] F. Mezzadri, "How to generate random matrices from the classical compact groups," *Notices of the American Mathematical Society*, vol. 54, no. 5, pp. 592–604, 2007.