Estimation of Linear Systems using a Gibbs Sampler ⋆

Adrian Wills * **Thomas B. Schön** ** **Fredrik Lindsten** **
Brett Ninness *

* *School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan, NSW, 2308, Australia (Tel: +61 2 49216028; e-mail: {Adrian.Wills, Brett.Ninness}@newcastle.edu.au).*
** *Division of Automatic Control, Linköping University, SE-581 83 Linköping, Sweden (Tel: +46 13 281373; e-mail: {schon, lindsten}@isy.liu.se).*

Abstract: This paper considers a Bayesian approach to linear system identification. One motivation is the advantage of the minimum mean square error of the associated conditional mean estimate. A further motivation is the error quantifications afforded by the posterior density which are not reliant on asymptotic in data length derivations. To compute these posterior quantities, this paper derives and illustrates a Gibbs sampling approach, which is a randomized algorithm in the family of Markov chain Monte Carlo methods. We provide details on a numerically robust implementation of the Gibbs sampler. In a numerical example, the proposed method is illustrated to give good convergence properties without requiring any user tuning.

Keywords: Parameter Estimation, System Identification, Bayesian statistics, Markov Chain Monte Carlo techniques.

1. INTRODUCTION

This paper is directed at the problem of estimating a parametrized linear state-space model on the basis of observed data. This is a very well studied problem, for which there are several well accepted solution techniques such maximum-likelihood, prediction error and subspace-based estimation methods [8].

While approaches such as these are now widely used because of their effectiveness, their utility is commonly dependent upon the availability of reliable error bounds on the delivered estimate. To provide these, central limit theorems that are asymptotic in the limit as the available data length tends to infinity are assumed to hold approximately for the finite data length available.

This leaves open the question of providing error bounds that are accurate for short data lengths. It also leaves open the question of providing error bounds for implicit functions of the parameters, such as achieved phase margin for a given controller.

Consideration of these and other motivations, such as minimizing the variance of the estimate for a given available finite data length were the motivation for the recent work by Ninness and Henriksen [10] wherein a Bayesian approach was considered. The rationale is that if the posterior distribution of the parameters can be computed, then it can effectively define feasible parameter regions in a manner that is accurate even for short data lengths. Furthermore, if a point estimate is required, the posterior mean, which is known to be the minimum mean-squared-error estimate, may be employed.

Despite these advantages, a Bayesian estimate is not commonly employed, save for special cases such as a linearly parametrized Gaussian situation where a Kalman filter can be used to perform the necessary quantifications. This is likely due to the fact that in general, computing a conditional mean or marginal posterior density requires the evaluation of a multi-dimensional integral.

In the previous work by Ninness and Henriksen [10], this difficulty was addressed by employing randomized algorithms to compute effective approximations to these integrals. More specifically, the Metropolis–Hastings (MH) algorithm (see *e.g.* [12]) was employed to generate realisations from the posterior density of the parameters. Sample averages of these realisations then deliver approximations for the posterior mean or marginal densities.

While these quantifications are approximate, they can be made arbitrarily accurate by running the MH-algorithm for sufficiently many iterations. Furthermore, sample averages or histograms of arbitrary functions (such as achieved phase margin) of the realisations from the posterior provide an effective quantification of the values of these quantities.

Despite these attractive features, the MH-algorithm approach has a significant drawback. A certain "proposal density" needs to be tuned, usually by hand, to deliver realisations that are sufficiently uncorrelated that the sample averages converge at a sufficiently quick rate to the desired expected value.

This paper addresses this weakness by employing a particular variant of the MH-algorithm known as "Gibbs sampling", for which no proposal density tuning is required. This idea has been previously employed in [2] for state-space models for the case of univariate state with no control input. In this paper, we generalise the approach by allowing a fully paramterized multivariable state-space model and allow a more general noise structure and control input.

⋆ This work was supported by: the Australian Research Council through their Discovery Project Program; and CADICS, a Linneaus Center funded by the Swedish Research Council and the project Calibrating Nonlinear Dynamical Models (Contract number: 621-2010-5876) also funded by the Swedish Research Council.

As will be illustrated, this delivers a method for Bayesian estimation of linear systems that does not require user tuning, and enjoys good convergence properties. Unsurprisingly, this benefit comes at a cost. Namely, the nature of the prior density that may be imposed is constrained. However, in Section 6 we discuss a possible modification of the proposed Gibbs sampler, aimed at mitigating this drawback.

2. PROBLEM FORMULATION

Consider a linear time-invariant (LTI) state-space model of the form

$$\underbrace{\begin{bmatrix} x_{t+1} \\ y_t \end{bmatrix}}_{\xi_t} = \underbrace{\begin{bmatrix} A & B \\ C & D \end{bmatrix}}_{\Gamma} \underbrace{\begin{bmatrix} x_t \\ u_t \end{bmatrix}}_{z_t} + \underbrace{\begin{bmatrix} v_t \\ e_t \end{bmatrix}}_{w_t}, \tag{1a}$$

where $x_t \in \mathbf{R}^{n_x}$ denotes the system state, $u_t \in \mathbf{R}^{n_u}$ denotes a known input (excitation) signal and $y_t \in \mathbf{R}^{n_y}$ denotes an observed output. The noise w_t, composed of the process noise $v_t \in \mathbf{R}^{n_x}$ and the measurement noise $e_t \in \mathbf{R}^{n_y}$, is assumed to be a zero mean i.i.d. processes with Gaussian distribution

$$w_t \sim \mathcal{N}(0, \Pi), \qquad \Pi = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}. \tag{1b}$$

Furthermore, let the data records X and Y be defined as

$$X \triangleq \{x_1, \cdots, x_{N+1}\}, \quad Y \triangleq \{y_1, \cdots, y_N\}, \tag{2a}$$

and let the parameters of the model (1) be defined according to

$$\theta^T = \left[\operatorname{vec}\{\Gamma\}^T, \operatorname{vec}\{\Pi\}^T\right] \in \mathbf{R}^{n_\theta}. \tag{3}$$

The aim in this work is to compute a point estimate $\widehat{\theta}$ of θ as the conditional mean

$$\widehat{\theta} \triangleq \mathbf{E}\{\theta \mid Y\} = \int \theta\, p(\theta \mid Y)\, \mathrm{d}\theta. \tag{4}$$

Clearly, this requires knowledge of the posterior density $p(\theta \mid Y)$ and a further goal of this work is to compute posteriors

$$p(f(\theta) \mid Y) \tag{5}$$

of general functions $f : \theta \mapsto \mathbf{C}^n$ of the parameters, such as frequency response, achieved phase margin for a given controller etc.

An attraction of the estimate (4) is its known minimum mean square error property [1]. Despite this attraction, is has not been widely used in a dynamic system identification context, presumably due to the computational difficulty involved in calculating the multivariable integral in (4). This paper will address this difficulty by employing a randomized algorithm from the Markov Chain Monte Carlo class of methods; namely a Gibbs sampler. As will be seen, a dividend of this approach is that the computation of the posterior (5) is straightforward for a very general range of functions f.

3. MARKOV CHAIN MONTE CARLO

The key idea underlying Markov chain Monte Carlo methods is to generate samples from the desired distribution $p(\theta \mid Y)$ by simulating a Markov chain, with this distribution as its stationary distribution.

These samples can then be used to form an appropriate estimate. The basic MCMC sampler is known as the Metropolis-Hastings (MH) algorithm after its inventors Metropolis et al. [9] and Hastings [6]. It allows for completely general specification of the prior density $p(\theta)$, but has a drawback in that it requires specification of a so called "proposal density" which often requires hand tuning in order to obtain satisfactory results.

The paper here make use of a special case of the MH sampler, referred to as the Gibbs sampler, wherein by giving up some flexibility in the specification of the prior $p(\theta)$, the choice of the proposal density to deliver good convergence properties is automatic.

The Gibbs sampler was first introduced by Geman and Geman [4] and later popularised by Gelfand and Smith [3]. The theory of MCMC is by now well developed, including rigorous convergence results, establishing the validity of the approach [12, 13]. A recent introduction of the use of MCMC to solve problems in system identification is provided by Ninness and Henriksen [10].

In order to employ this approach in the linear system identification setting of this paper, it is necessary to first expand the problem to one wherein we see the joint posterior

$$p(\theta, X \mid Y) \tag{6}$$

of both the parameters and the complete state history. This may seem strange and artificial, since it is only the parameter posterior we are interested in. Nevertheless, it is a key step in this paper, because as will be seen, it simplifies the problem of drawing from the involved posterior densities. At the same time, if one simply ignores the realisations of the state history X that are generated, we are left with samples from the sought marginal density $p(\theta \mid Y)$. To be more specific, the Gibbs sampler with target density (6) proceeds by drawing samples from the density of one of the elements θ_i of θ or x_i of X conditioned on all the other variables being known. In other words, at each iteration k of the Gibbs sampler, the following random draws are required

- Sample $\theta_1^k \sim p(\theta_1 \mid \theta_{\setminus 1}, X, Y)$.

  $\vdots$
- Sample $\theta_{n_\theta}^k \sim p(\theta_2 \mid \theta_{\setminus n_\theta}, X, Y)$.
- Sample $x_1^k \sim p(x_1 \mid x_{\setminus 1}, \theta, Y)$.

  $\vdots$
- Sample $x_{N+1}^k \sim p(x_{N+1} \mid x_{\setminus N+1}, \theta, Y)$.

where $x_{\setminus i}$ is used to denote all the elements in x, but x_i, *i.e.* $x_{\setminus i} \triangleq (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N)$, and similarly for θ. Hence, the Gibbs sampler requires full knowledge of all the conditional distributions and the ability to generate samples from these distributions.

The Gibbs sampler as outlined above does not perform well when the variables are strongly dependent on one another, which is the case for the state variables in the dynamical system (1) under consideration. We will reduce this problem by employing a so called *blocked Gibbs sampler*, where several of the variables are grouped together. This simplifies the algorithm to

(1) Given θ^k, generate a sample of the state trajectory according to

$$X^k \sim p(X \mid Y, \theta^k). \tag{7}$$

(2) Then, given X^k generate a sample of θ^{k+1} according to

$$\theta^{k+1} \sim p(\theta \mid X^k, Y). \tag{8}$$

The first step of the algorithm outlined above corresponds to generating a sample from the joint smoothing density of the states in a LTI state-space model. In Section 4 below this is solved by making use of Kalman filter and a so called backward simulator. Generating samples according to (8)

will be facilitated by employing conjugate priors to deliver tractable densities to sample from, which is described in Section 5.

4. SAMPLING THE STATE TRAJECTORY

In order to generate samples from $p(X \mid Y, \theta)$ we could make use of the standard results for Gaussian conditional distributions and draw samples from the resulting normal distribution. This approach is valid for small N, but for increasing N it soon becomes infeasible due to the matrix inversions involved. It is also worth noting that a standard formulation of the Kalman smoother cannot be used here, since it only provide the marginal densities $p(x_t \mid Y, \theta)$ while we require the joint density $p(X \mid Y, \theta) = p(x_1, \cdots, x_{N+1} \mid Y, \theta)$.

Here we solve this by employing a backward simulator, which for the LTI state-space model (1) is developed in Section 4.1. This idea has appeared before in [2]. Here, we generalise this to the full linear state-space model represented by (1).

In order to obtain a numerically reliable implementation we have found it vital to make use of square-root formulas for the forward filter backward simulation steps and these formulas are developed in Section 4.2.

Since the sampling in step (7) of the blocked Gibbs sampler is done for (conditioned on) a fixed parameter value, we shall, for notational simplicity, omit θ from the notation throughout this section.

4.1 Backward simulation

For convenience, let us first define $x_{i:j} \triangleq \{x_i, \cdots, x_j\}$, implying that $X = x_{1:N+1}$, $Y = y_{1:N}$. With this notation in place the conditional density of X given Y is

$$\begin{aligned} p(x_{1:N+1} \mid y_{1:N}) &= p(x_1 \mid x_{2:N+1},\ y_{1:N}) p(x_{2:N+1} \mid y_{1:N}) \\ &= \ldots = p(x_{N+1} \mid y_{1:N}) \prod_{t=1}^{N} p(x_t \mid x_{t+1:N+1},\ y_{1:N}). \end{aligned}$$

This factorisation highlights the fact that we can sample from $p(X \mid Y)$ by using the following strategy; first, draw a sample from the marginal density $p(x_{N+1} \mid y_{1:N})$. Then, conditioned on this sample we draw from the conditional density $p(x_N \mid x_{N+1},\ y_{1:N})$ and continue in this fashion until we reach time $t = 1$. The generated state trajectory is then a sample from the joint density $p(X \mid Y)$.

We now proceed with the details of this backward simulation technique. From the Markov property (which for the state-space model (1) is valid both forward and backward in time) we have that

$$p(x_t | x_{t+1:N+1}, y_{1:N}) = p(x_t | x_{t+1}, y_{1:N}). \tag{9}$$

We can then make use of the fact that the measurements are conditionally independent, given the state, which results in

$$p(x_t | x_{t+1}, y_{1:t}, y_{t+1:N}) = p(x_t | x_{t+1}, y_{1:t}). \tag{10}$$

Put in other words, given the state at time $t+1$, there in no additional information about the state x_t present in the measurements $y_{t+1:N}$. It follows that,

$$p(x_t \mid x_{t+1:N+1},\ y_{1:N}) = p(x_t \mid x_{t+1},\ y_{1:t}). \tag{11}$$

Furthermore, note that

$$\begin{aligned} p(x_t \mid x_{t+1},\ y_{1:t}) &= \frac{p(x_t, x_{t+1} \mid y_{1:t})}{p(x_{t+1} \mid y_{1:t})} \\ &= \frac{p(x_{t+1} \mid x_t) p(x_t \mid y_{1:t})}{p(x_{t+1} \mid y_{1:t})}. \end{aligned} \tag{12}$$

Now since x_{t+1} and $y_{1:t}$ are given, the above denominator is a non-zero number and

$$p(x_t \mid x_{t+1},\ y_{1:t}) \propto p(x_{t+1} \mid x_t) p(x_t \mid y_{1:t}). \tag{13}$$

This provides a simple formula for recursively drawing samples for the joint state sequence X given the measurements Y. In particular, for the linear state-space model with Gaussian noise in (1) we have that

$$-2 \log p(x_{t+1} \mid x_t) \propto \|x_{t+1} - Ax_t - Bu_t\|^2_{Q^{-1}} \tag{14}$$

Further, the distribution $p(x_t \mid y_{1:t})$ is provided by the standard Kalman Filter recursions that deliver

$$-2 \log p(x_t \mid y_{1:t}) \propto \|x_t - \hat{x}_{t|t}\|^2_{P^{-1}_{t|t}} \tag{15}$$

Therefore

$$\begin{aligned} -2 \log p(x_t \mid x_{t+1},\ y_{1:t}) \propto \|x_t - \hat{x}_{t|t}\|^2_{P^{-1}_{t|t}} \\ + \|x_{t+1} - Ax_t - Bu_t\|^2_{Q^{-1}} \end{aligned} \tag{16}$$

From [11] (Section A.2, Equation (A.7)), it follows that

$$-2 \log p(x_t \mid x_{t+1},\ y_{1:t}) \propto \|x_t - \mu_t\|^2_{M_t^{-1}} \tag{17}$$

where

$$M_t = (A^T Q^{-1} A + P^{-1}_{t|t})^{-1} \tag{18a}$$

$$= P_{t|t} - P_{t|t} A^T (A P_{t|t} A^T + Q)^{-1} A P_{t|t}, \tag{18b}$$

$$\mu_t = M_t (A^T Q^{-1} [x_{t+1} - Bu_t] + P^{-1}_{t|t} \hat{x}_t). \tag{18c}$$

In Algorithm 1 below, we provide the steps required in order to draw a sample $X \sim p(X \mid Y,\ \theta)$. For convenience, we will first transform the original system (1) into an equivalent one with $S = 0$. This is necessary in order to employ a square-root implementation of the forward filter equations (see e.g. Chapter 11 in [7]). The resulting equivalent system is given by [7],

$$\begin{bmatrix} x_{t+1} \\ y_t \end{bmatrix} = \begin{bmatrix} \bar{A} & \bar{B} \\ C & D \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix} + \begin{bmatrix} S R^{-1} y_t \\ 0 \end{bmatrix} + \bar{w}_t, \tag{19a}$$

$$\bar{w}_t \sim \mathcal{N}\left(0, \begin{bmatrix} \bar{Q} & 0 \\ 0 & R \end{bmatrix}\right), \tag{19b}$$

where

$$\bar{A} = A - S R^{-1} C, \quad \bar{B} = B - S R^{-1} D, \quad \bar{Q} = Q - S R^{-1} S^T.$$

With this in place, the algorithm can be stated as in Algorithm 1.

4.2 Implementing Algorithm 1 in square-root form

The recursion in Algorithm 1 can be implemented by propagating the square-root of the covariance matrices $P_{t|t}$, $P_{t+1|t}$ and M_t. The benefit of using a square-root implementation is that it improves numerical precision [7], and in addition all the required covariance matrices are ensured to be positive definite by construction. For convenience, we will use the notation that the square-root of a general matrix G is denoted by $G^{1/2}$ so that

$$G^{1/2} G^{T/2} = G. \tag{20}$$

In what follows, we will require the suare-root of R and $\bar{Q}$, denoted as $R^{1/2}$ and $\bar{Q}^{1/2}$, respectively. These quantities

Algorithm 1 Sample from $p(X \mid Y, \theta)$

1: Assume $\hat{x}_{1|0}$ and $P_{1|0}$ are given and run the Kalman filter recursions from $t = 1, \ldots, N$

$$\begin{aligned} e_t &= y_t - C\hat{x}_{t|t-1} - Du_t, \\ \hat{x}_{t|t} &= \hat{x}_{t|t-1} + K_t e_t, \\ \hat{x}_{t+1|t} &= \bar{A}\hat{x}_{t|t} + \bar{B}u_t + SR^{-1}y_t, \\ K_t &= P_{t|t-1}C^T(CP_{t|t-1}C^T + R)^{-1}, \\ P_{t|t} &= P_{t|t-1} - P_{t|t-1}C^T(CP_{t|t-1}C^T + R)^{-1}CP_{t|t-1}, \\ P_{t+1|t} &= \bar{A}P_{t|t}\bar{A}^T + \bar{Q}. \end{aligned}$$

2: Draw $x_{N+1} \sim \mathcal{N}(\hat{x}_{N+1|N},\ P_{N+1|N})$ and iterate the following backward simulation recursion from $t = N, \ldots, 1$

$$\begin{aligned} x_t &\sim \mathcal{N}(\mu_t,\ M_t), \\ M_t &= P_{t|t} - P_{t|t}A^T(AP_{t|t}A^T + Q)^{-1}AP_{t|t}, \\ \mu_t &= \eta_t - L_t A\eta_t, \\ \eta_t &= \hat{x}_t + P_{t|t}A^T Q^{-1}(x_{t+1} - Bu_t), \\ L_t &= P_{t|t}A^T(AP_{t|t}A^T + Q)^{-1}. \end{aligned}$$

may be computed as follows. Let $\mathcal{L}$ be a lower triangular Cholesky factor so that

$$\mathcal{L}\mathcal{L}^T = \begin{bmatrix} \mathcal{L}_{11} & 0 \\ \mathcal{L}_{12} & \mathcal{L}_{22} \end{bmatrix} \begin{bmatrix} \mathcal{L}_{11}^T & \mathcal{L}_{12}^T \\ 0 & \mathcal{L}_{22}^T \end{bmatrix} = \begin{bmatrix} R & S^T \\ S & Q \end{bmatrix} \tag{21}$$

and note that

$$\mathcal{L}_{11}\mathcal{L}_{11}^T = R, \quad \mathcal{L}_{11}\mathcal{L}_{12}^T = S^T, \quad \mathcal{L}_{12}\mathcal{L}_{12}^T + \mathcal{L}_{22}\mathcal{L}_{22}^T = Q. \tag{22a}$$

The last expression can be written as

$$\begin{aligned} \mathcal{L}_{22}\mathcal{L}_{22}^T &= Q - \mathcal{L}_{12}\mathcal{L}_{12}^T = Q - S\mathcal{L}_{11}^{-T}\mathcal{L}_{11}^{-1}S^T \\ &= Q - SR^{-1}S^T, \end{aligned} \tag{23}$$

which is the required quantity, so that $\mathcal{L}_{22}$ is a square-root of $\bar{Q}$, i.e. $\bar{Q}^{1/2} = \mathcal{L}_{22}$. Furthermore, note that the square-root of R is given by (22a), i.e. $R^{1/2} = \mathcal{L}_{11}$, and that

$$SR^{-1} = \mathcal{L}_{12}\mathcal{L}_{11}^{-1}. \tag{24}$$

Therefore, expressions like $SR^{-1}C$ involve a forward substitution $\tilde{C} = \mathcal{L}_{11}^{-1}C$ followed by a matrix multiplication $\mathcal{L}_{12}\tilde{C}$, and no explicit matrix inverse is calculated.

It is assumed that the square-root of $P_{1|0} = P_{1|0}^{1/2}P_{1|0}^{T/2}$ is provided. With this in place, we can employ a $\mathcal{QR}$-decomposition to factor

$$\begin{bmatrix} R^{T/2} & 0 \\ P_{t|t-1}^{T/2}C^T & P_{t|t-1}^{T/2} \end{bmatrix} = \mathcal{Q}^1 \begin{bmatrix} \mathcal{R}_{11}^1 & \mathcal{R}_{12}^1 \\ 0 & \mathcal{R}_{22}^1 \end{bmatrix}, \tag{25}$$

It can be verified that

$$\begin{aligned} &(\mathcal{R}_{22}^1)^T\mathcal{R}_{22}^1 \\ &= P_{t|t-1} - P_{t|t-1}C^T(CP_{t|t-1}C^T + R)^{-1}CP_{t|t-1} \end{aligned} \tag{26}$$

so that $P_{t|t}^{T/2} = \mathcal{R}_{22}^1$ is the required square-root. It can also be verified that $K_t = (\mathcal{R}_{12}^1)^T(\mathcal{R}_{11}^1)^{-T}$. Employing this square-root $P_{t|t}^{T/2}$ in the following $\mathcal{QR}$-decomposition

$$\begin{bmatrix} P_{t|t}^{T/2}\bar{A}^T \\ \bar{Q}^{T/2} \end{bmatrix} = \mathcal{Q}^2 \begin{bmatrix} \mathcal{R}_1^2 \\ 0 \end{bmatrix}, \tag{27}$$

it can be verified that

$$(\mathcal{R}_1^2)^T\mathcal{R}_1^2 = \bar{A}P_{t|t}\bar{A}^T + \bar{Q} \tag{28}$$

and therefore $P_{t+1|t}^{T/2}$ is given by $P_{t+1|t}^{T/2} = \mathcal{R}_1^2$. Using very similar arguments, it can be readily verified that

$$M_t^{T/2} = \mathcal{R}_{22}^3, \qquad L_t = (\mathcal{R}_{12}^3)^T(\mathcal{R}_{11}^3)^{-T}, \tag{29}$$

where again we have used a $\mathcal{QR}$-decomposition to provide

$$\begin{bmatrix} Q^{T/2} & 0 \\ P_{t|t}^{T/2}A^T & P_{t|t}^{T/2} \end{bmatrix} = \mathcal{Q}^3 \begin{bmatrix} \mathcal{R}_{11}^3 & \mathcal{R}_{12}^3 \\ 0 & \mathcal{R}_{22}^3 \end{bmatrix}. \tag{30}$$

Algorithm 2 below summarises the above steps for the square-root implementation.

Algorithm 2 Sample from $p(X \mid Y, \theta)$ in square-root form

1: Compute the Cholesky factor $\mathcal{L}$ from (21) and set

$$\begin{aligned} \bar{A} &= A - \mathcal{L}_{12}\left[\mathcal{L}_{11}^{-1}C\right], \\ \bar{B} &= B - \mathcal{L}_{12}\left[\mathcal{L}_{11}^{-1}D\right], \\ \bar{Q}^{1/2} &= \mathcal{L}_{22}, \\ \bar{R}^{1/2} &= \mathcal{L}_{11}. \end{aligned}$$

2: Assume $\hat{x}_{1|0}$ and $P_{1|0}^{1/2}$ are given and run the Kalman filter recursions from $t = 1, \ldots, N$: For each t, compute $\mathcal{Q}^1\mathcal{R}^1$ from (25) and set

$$P_{t|t}^{T/2} = \mathcal{R}_{22}^1, \quad K_t = (\mathcal{R}_{12}^1)^T(\mathcal{R}_{11}^1)^{-T}$$

then compute $\mathcal{Q}^2\mathcal{R}^2$ from (27) and $\mathcal{Q}^3\mathcal{R}^3$ from (30) and set

$$P_{t+1|t}^{T/2} = \mathcal{R}_1^2, \quad M_t^{T/2} = \mathcal{R}_{22}^3, \quad L_t = (\mathcal{R}_{12}^3)^T(\mathcal{R}_{11}^3)^{-T}.$$

Compute

$$\begin{aligned} e_t &= y_t - C\hat{x}_{t|t-1} - Du_t, \\ \hat{x}_{t|t} &= \hat{x}_{t|t-1} + K_t e_t, \\ \hat{x}_{t+1|t} &= \bar{A}\hat{x}_{t|t} + \bar{B}u_t + SR^{-1}y_t. \end{aligned}$$

3: Draw $x_{N+1} \sim \mathcal{N}(\hat{x}_{N+1|N},\ P_{N+1|N})$ and iterate the following backward simulation recursion for $t = N, \ldots, 1$

$$\begin{aligned} x_t &\sim \mathcal{N}(\mu_t,\ M_t), \\ \mu_t &= \eta_t - L_t A\eta_t, \\ \eta_t &= \hat{x}_t + P_{t|t}A^T Q^{-1}(x_{t+1} - Bu_t). \end{aligned}$$

5. SAMPLING THE PARAMETERS

In this section we discuss how to draw a sample of θ given a sample of the state sequence X and the measurements Y. Note as usual that

$$p(\theta \mid X, Y) \propto p(X, Y \mid \theta)p(\theta). \tag{31}$$

It is assumed in this section that the prior $p(\theta)$ can be adequately described using a Matrix-Normal-Inverse-Wishart (MNIW) distribution [14]. The reason for this assumption is that it is a conjugate prior for the linear Gaussian state-space model, which means that $p(\theta \mid X, Y)$ also has an MNIW distribution, and therefore we can readily obtain samples from it. More specifically, it is assumed that (recall that $\theta = [\mathrm{vec}\{\Gamma\}^T, \mathrm{vec}\{\Pi\}^T]^T$),

$$p(\theta) = p(\Gamma \mid \Pi)p(\Pi), \tag{32}$$

and that Γ conditioned on Π is a matrix-normal distribution, written as $\mathcal{MN}(\Gamma; M, \Pi, V)$ with density

$$p(\Gamma \mid \Pi) = \frac{|V|^{(n+m)/2}}{|2\pi\Pi|^{(n+p)/2}} \exp\left(-\frac{1}{2}\mathrm{tr}\left((\Gamma - M)^T\Pi^{-1}(\Gamma - M)V\right)\right) \tag{33}$$

where M and V^{-1} are the prior mean and row covariance matrices, respectively. Furthermore, it is assumed

that $p(\Pi)$ is distributed according to an inverse Wishart distribution, written as $\mathcal{IW}(\Pi; \ell, \Lambda)$ with density

$$p(\Pi) = \frac{|\Lambda|^{\ell/2}|\Pi|^{-(n+p+\ell+1)/2}}{2^{\ell(n+p)/2}\gamma_{(n+p)}(\ell/2)} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\Pi^{-1}\Lambda\right)\right) \quad (34)$$

where $\gamma_{(n+p)}(\cdot)$ is the multivariate gamma function.

From the above it may be noticed that M, V, Λ and ℓ are all user specified choices for the priors on θ, commonly referred to as hyperparameters. The first two, M, V describe the mean and row accuracy of the matrix Γ, while the second two Λ, ℓ describe a scaling and degree of freedom for the covariance matrix Π.

With a distribution for the prior $p(\theta)$ in place, consider the joint likelihood term $p(X, Y \mid \theta)$ in (31). For the model class in (1) we may write (using Bayes rule)

$$\begin{aligned} p(X, Y \mid \theta) &= p(x_{N+1}, y_N \mid x_N, y_{N-1}, \cdots, y_1, x_1, \theta) \\ &\quad \times p(x_N, y_{N-1}, \cdots, y_1, x_1, \theta), \end{aligned} \quad (35)$$

and since the model is also Markov, then

$$\begin{aligned} &p(x_{N+1}, y_N \mid x_N, y_{N-1}, \cdots, y_1, x_1, \theta) \\ &\qquad = p(x_{N+1}, y_N \mid x_N, \theta), \end{aligned} \quad (36)$$

so that (35) becomes

$$p(X, Y \mid \theta) = p(x_1 \mid \theta) \prod_{t=1}^{N} p(x_{t+1}, y_t \mid x_t, \theta). \quad (37)$$

This latter expression can be re-written using the definitions in (1) and the fact that the initial state does not depend on θ (so that $p(x_1 \mid \theta) = p(x_1)$) to arrive at

$$\begin{aligned} -2\log p(X, Y \mid \theta) &\propto N \log\det\Pi \\ &\quad + \mathrm{tr}\{\Pi^{-1}[\Phi - \Gamma\Psi^T - \Psi\Gamma^T + \Gamma\Sigma\Gamma^T]\}, \end{aligned} \quad (38a)$$

where

$$\Phi = \sum_{t=1}^{N} \begin{bmatrix} x_{t+1} \\ y_t \end{bmatrix} \begin{bmatrix} x_{t+1} \\ y_t \end{bmatrix}^T, \quad \Psi = \sum_{t=1}^{N} \begin{bmatrix} x_{t+1} \\ y_t \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix}^T, \quad (38b)$$

$$\Sigma = \sum_{t=1}^{N} \begin{bmatrix} x_t \\ u_t \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix}^T. \quad (38c)$$

It follows (see *e.g.* [14]) that the posterior distribution is MNIW and given by,

$$p(\Gamma, \Pi \mid X, Y) = p(\Gamma \mid X, Y, \Pi) p(\Pi \mid X, Y),$$

with

$$p(\Gamma \mid X, Y, \Pi) = \mathcal{MN}(\Gamma; \bar{\Psi}\bar{\Sigma}^{-1}, \Pi, \bar{\Sigma}^{-1}), \quad (39a)$$

$$p(\Pi \mid X, Y) = \mathcal{IW}(N + \ell, \Lambda + \bar{\Phi} - \bar{\Psi}\bar{\Sigma}^{-1}\bar{\Psi}^T), \quad (39b)$$

where

$$\bar{\Phi} = \Phi + MVM^T, \quad \bar{\Psi} = \Psi + MV, \quad \bar{\Sigma} = \Sigma + V. \quad (39c)$$

Note that for the purposes of implementation, it is worth recalling that the matrix-normal and the vector-normal distributions are linked by

$$\mathrm{vec}\{\Gamma\} \sim \mathcal{N}(\mathrm{vec}\left\{\Psi\Sigma^{-1}\right\}, \ \bar{\Sigma}^{-1} \otimes \Pi). \quad (40)$$

The above steps are summarised in Algorithm 3, where again, we have been careful to compute the required matrices in a numerically robust manner. The final algorithm is presented in Algorithm 4.

6. DISCUSSION ON PRIORS

So far, we have assumed that the prior knowledge about the system parameters can be described by an MNIW density. This choice is made to enable closed form expressions for the posterior densities used in the Gibbs sampler, which is possible since the MNIW prior is conjugate to the likelihood defined by the model (1). In many cases, the MNIW prior is an adequate choice which does not affect the posterior distribution to any significant extent. Still, in some cases it might be desirable to use some other parameter prior which might represent the *a priori* knowledge about the system in a better way. If this is the case, it is possible to make a simple modification of the proposed Gibbs sampler to account for such an alternative prior.

Algorithm 3 Sample from $p(\theta \mid X, Y)$

1: Compute $\bar{\Phi}$, $\bar{\Psi}$ and $\bar{\Sigma}$ from (39c) and (38b)–(38c).
2: Compute the Cholesky factor $\mathcal{L}$ such that

$$\mathcal{L}\mathcal{L}^T = \begin{bmatrix} \mathcal{L}_{11} & 0 \\ \mathcal{L}_{12} & \mathcal{L}_{22} \end{bmatrix} \begin{bmatrix} \mathcal{L}_{11}^T & \mathcal{L}_{12}^T \\ 0 & \mathcal{L}_{22}^T \end{bmatrix} = \begin{bmatrix} \bar{\Sigma} & \bar{\Psi}^T \\ \bar{\Psi} & \bar{\Phi} \end{bmatrix}$$

and set

$$\Gamma^\star = \bar{\Psi}\bar{\Sigma}^{-1} = \mathcal{L}_{12}\mathcal{L}_{11}^{-1}, \quad (41a)$$

$$\Pi^\star = \bar{\Phi} - \bar{\Psi}\bar{\Sigma}^{-1}\bar{\Psi}^T = \mathcal{L}_{22}\mathcal{L}_{22}^T. \quad (41b)$$

3: Sample Π from

$$\Pi \sim \mathcal{IW}(N + \ell, \Lambda + \Pi^\star). \quad (42)$$

4: Sample Γ from

$$\Gamma \sim \mathcal{MN}(\Gamma; \Gamma^\star, \Pi, \bar{\Sigma}^{-1}). \quad (43)$$

Algorithm 4 Blocked Gibbs Sampler

1: Assume that $\Pi^1 \succ 0$, Γ^1, M, V, Λ and ℓ are given and set $k = 0$. Iterate the following steps until $k \geq k_{\max}$, where $k_{\max} > 0$ is the maximum number of iterations.
2: Sample $X^k \sim p(X \mid Y, \Gamma^k, \Pi^k)$ according to Algorithm 2.
3: Sample $(\Gamma^{k+1}, \Pi^{k+1}) \sim p(\Gamma, \Pi \mid X^k, Y)$ according to Algorithm 3.
4: Update $k \leftarrow k + 1$.

The idea is to use the same mechanism to sample the parameter values in the MCMC method, *i.e.* by running Algorithm 3. Hence, when we sample θ we "pretend" that the parameter prior is MNIW. To compensate for the fact that we use the wrong prior, we complement the sampling with a Metroplis-Hastings (MH) accept/reject decision. In other words, we use Algorithm 3 as a proposal density in an MH sampler.

Consider again the posterior parameter density which can be written,

$$p(\theta \mid X, Y) \propto p(X, Y \mid \theta) p(\theta), \quad (44)$$

where $p(\theta)$ is the "true" parameter prior. We then construct a proposal density according to

$$q(\theta \mid X, Y) \propto p(X, Y \mid \theta) q(\theta), \quad (45)$$

where $q(\theta)$ is an MNIW density. Thus, at iteration k of the MCMC sampler, we can run Algorithm 3 to generate a sample θ' from $q(\theta \mid X, Y)$. With probability,

$$\rho = \frac{p(\theta' \mid X, Y)}{p(\theta^{k-1} \mid X, Y)} \frac{q(\theta^{k-1} \mid X, Y)}{q(\theta' \mid X, Y)} = \frac{p(\theta')}{p(\theta^{k-1})} \frac{q(\theta^{k-1})}{q(\theta')}, \quad (46)$$

we accept the proposed sample and set $\theta^k = \theta'$. If the sample is not accepted, we retain the previous state of the Markov chain and set $\theta^k = \theta^{k-1}$.

The success of this approach depends on how close the artificial MNIW prior is to the "true" prior. If these densities are very different, the acceptance probability in (46) is likely to be too small to obtain a satisfactory mixing of the Markov chain. However, if the priors are fairly similar, the addition of such an accept/reject step is an

easy way to modify the stationary distribution of Markov chain to represent a different prior than MNIW.

7. SIMULATION EXAMPLE

In this section we provide a simulation example that demonstrates the potential of the blocked Gibbs sampler provided in Algorithm 4 by generating the density of the phase margin ϕ for a given controller design. In particular, we generate a data set Y for a given true system and run the Gibbs sampler to generate samples $(\theta, X) \sim p(\theta, X \mid Y)$ based on this data. More specifically, consider the 6'th order true system (A, B, C, D, Q, S, R) with Bode response shown in Figure 1. We simulated an output signal $Y = \{y_1, \cdots, y_N\}$ for $N = 200$ according to

$$x_{t+1} = Ax_t + Bu_t + w_t, \tag{47a}$$
$$y_t = Cx_t + Du_t + e_t, \tag{47b}$$
$$u_t \sim \mathcal{N}(0, 1), \tag{47c}$$
$$\begin{bmatrix} w_t \\ e_t \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} 10^{-3} I_6 & 0 \\ 0 & 10^{-4} \end{bmatrix}\right), \tag{47d}$$
$$x_1 = 0. \tag{47e}$$

Algorithm 4 requires an initial estimate for θ^1, and this is provided by a standard Expectation Maximisation algorithm [5], which delivers the maximum likelihood estimate $(A_1, B_1, C_1, D_1, Q_1, S_1, R_1)$. Additionally, the Gibbs sampler requires the priors $M, V \succ 0$, $\Lambda \succ 0$ and $\ell \geq 1$, which were selected according to

$$M = \begin{bmatrix} A_0 & B_0 \\ C_0 & D_0 \end{bmatrix}, \quad V = I_7, \quad \Lambda = I_7, \quad \ell = 1. \tag{48}$$

As a means of utilising these samples $\{\theta^1, \cdots, \theta^{k_{\max}}\}$, we first consider the conditional mean estimate of the transfer function $G(z) = C(zI - A)^{-1}B + D$, which is shown in Figure 1. Based on this estimate, we designed a controller with nominal phase margin of $\phi = 22$ degrees. The utility of the MCMC approach is that we can compute the density of ϕ for the given data set. This density is shown in Figure 2 and is labelled as the "slow" controller. From this figure it can be seen that there is a small probability (based on the data) that the controller will be unstable. By way of comparison, we designed another controller with a nominal phase margin of $\phi = 14$ degrees, and its density is also shown in Figure 2 and labelled as "fast" controller. It could be argued that this controller is unacceptable based on the large number of cases where the closed loop system is unstable.

8. CONCLUSION

This paper considers the problem of estimating the density $p(\theta, X \mid Y)$ for linear state-space models using a blocked Gibbs sampler. Importantly, this requires the ability to sample from both $p(X \mid Y, \theta)$ and $p(\theta \mid Y, X)$ and we have provided details of these two densities, including their numerically robust square-root implementations. The utility and potential of this approach is profiled on an example where the posterior probability density function of the phase margin of a given controller is examined.

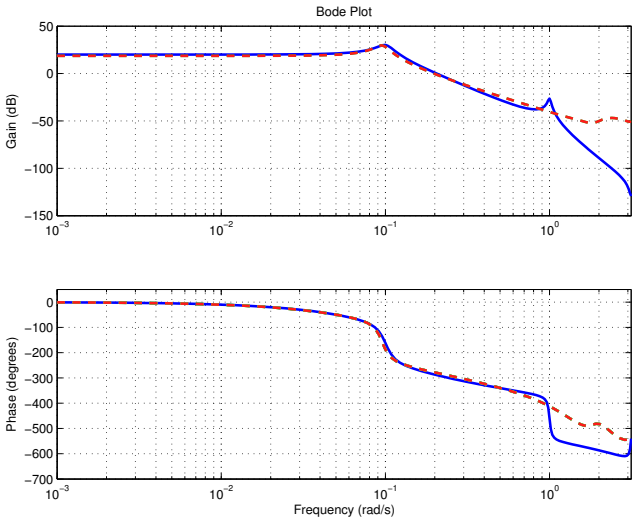

Fig. 1. Bode response for the true system (solid), and conditional mean estimate (dashed).

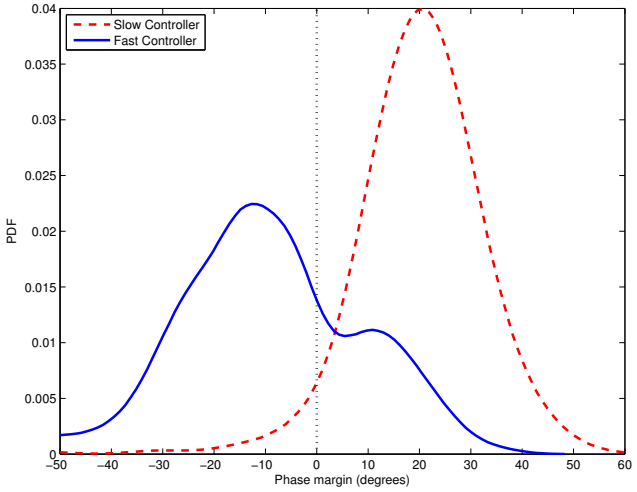

Fig. 2. Probability density function $p(\phi \mid Y)$ of phase margin ϕ for slow (dashed) and fast (solid) controllers. Dotted vertical line shows the zero degrees phase margin point.

REFERENCES

[1] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice-Hall, New Jersey, 1979.

[2] C. K. Carter and R. Kohn. On gibbs sampling for state space models. *Biometrika*, 81(3):541–553, 1994.

[3] A.E. Gelfand and A.F.M Smith. Sampling based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85:398–409, 1990.

[4] S. Geman and D. Geman. Stochastic relaxation, Gibss distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis*, 6:721–741, 1984.

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

[6] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[7] T. Kailath, A.H. Sayed, and B. Hassabi. *Linear Estimation*. Prentice Hall, Upper Saddle River, New Jersey, 2000.

[8] L. Ljung. *System Identification: Theory for the User, (2nd edition)*. Prentice-Hall, Inc., New Jersey, 1999.

[9] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machine. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[10] B. Ninness and S. Henriksen. Bayesian system identification via Markov chain Monte Carlo techniques. *Automatica*, 46(1): 40–51, 2010.

[11] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006. ISBN 0-262-18253-X.

[12] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer texts in statistics. Springer, New York, USA, second edition, 2004.

[13] L. Tierney. Markov chains for exploring posterior distributions. *Ann. Statist.*, 22(4):1701–1762, 1994. ISSN 0090-5364. With discussion and a rejoinder by the author.

[14] M. West and J. Harrison. *Bayesian Forecasting and Dynamic Models*. Springer, New York, 1997.