Hierarchical Bayesian approaches for robust inference in ARX models

Johan Dahlin, Fredrik Lindsten, Thomas Bo Schön and Adrian George Wills

Linköping University Post Print

N.B.: When citing this work, cite the original article.

Original publication:

Johan Dahlin, Fredrik Lindsten, Thomas Bo Schön and Adrian George Wills, Hierarchical Bayesian approaches for robust inference in ARX models, 2012, The 16th IFAC Symposium on System Identification.
http://dx.doi.org/10.3182/20120711-3-BE-2027.00318

Postprint available at: Linköping University Electronic Press
http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-81258

Hierarchical Bayesian ARX models for robust inference

Johan Dahlin * **Fredrik Lindsten** * **Thomas B. Schön** *
Adrian Wills **

* *Division of Automatic Control, Linköping University, Linköping, Sweden (e-mail: {johan.dahlin,lindsten,schon}@isy.liu.se)*
** *School of EECS, University of Newcastle, Callaghan, NSW 2308, Australia (e-mail: Adrian.Wills@newcastle.edu.au)*

Abstract: Gaussian innovations are the typical choice in most ARX models but using other distributions such as the Student's t could be useful. We demonstrate that this choice of distribution for the innovations provides an increased robustness to data anomalies, such as outliers and missing observations. We consider these models in a Bayesian setting and perform inference using numerical procedures based on Markov Chain Monte Carlo methods. These models include automatic order determination by two alternative methods, based on a parametric model order and a sparseness prior, respectively. The methods and the advantage of our choice of innovations are illustrated in three numerical studies using both simulated data and real EEG data.

Keywords: ARX models, Robust estimation, Bayesian methods, Markov chain Monte Carlo

1. INTRODUCTION

An autoregressive exogenous (ARX) model of orders $n = \{n_a, n_b\}$, is given by

$$y_t + \sum_{i=1}^{n_a} a_i^n y_{t-i} = \sum_{i=1}^{n_b} b_i^n u_{t-i} + e_t, \tag{1}$$

where a_i^n and b_i^n are model coefficients, u_t is a known input signal and e_t is white excitation noise, often assumed to be Gaussian and independent of the input signal. Then, for known model orders n, the maximum likelihood estimate of the unknown ARX coefficients $\theta^n = (a_1^n \cdots a_{n_a}^n \; b_1^n \cdots b_{n_b}^n)$ is given by least squares (LS). In practice, we are often faced with the following problems:

(1) The appropriate order for the model is unknown or no "best" model order may exist.
(2) The observed data is non-Gaussian in nature, e.g. due to outliers.

In this work, we propose two hierarchical Bayesian ARX models and algorithms to make inference in these models, thereby addressing both of the practical issues mentioned above. The proposed models differs from (1) in two aspects: (i) the excitation noise is modelled as Student's t distributed, and (ii) a built-in form of automatic order selection is used.

The t distribution is more heavy-tailed than the Gaussian distribution, which means that the proposed ARX model can capture "jumps" in the internal state of the system (as an effect of occasional large innovations). Furthermore, we believe that this will result in an inference method that is more robust to model errors and outliers in the observations, a property which we illustrate in this work.

We propose two alternative methods to automatically determine the system order n. Firstly, we let the model order n be a parameter of the Bayesian ARX model. The model order is inferred alongside the other unknown parameters, resulting in a posterior probability distribution over model orders. In the second model, we instead use a sparseness prior over the ARX coefficients, known as automatic relevance determination (ARD) [MacKey, 1994, Neal, 1996].

Based on the models introduced above, the resulting identification problem amounts to finding the posterior distribution of the model parameters θ^n and the order n. This is done using Markov Chain Monte Carlo Methods (see e.g. Robert and Casella [2004]), where we are constructing a Markov Chain with the posterior distribution as its stationary distribution. We can thus compute estimates under the posterior parameter distribution by sampling from the constructed Markov Chain.

For the first model, this is a challenging task as the model order is explicitly included in the parameter vector. This is due to the fact that we are now dealing with a parameter space of varying dimension, which thereby require the Markov Chain to do the same. This will be solved using the reversible jump MCMC (RJ-MCMC) algorithm introduced by Green [1995]. The inference problem resulting from the use of an ARD prior is in the other hand solvable using standard MCMC algorithms.

The use of RJ-MCMC to estimate the model order and the parameters of an AR model driven by Gaussian noise, is fairly well studied, see e.g. [Troughton and Godsill, 1998, Godsill, 2001, Brooks et al., 2003]. The present work differs from these contributions, mainly in the use of Student's t distributed innovations. Similar models are also considered by Christmas and Everson [2011], who derive a variational Bayes algorithm for the inference problem. This approach is not based on Monte Carlo sampling, but instead makes use of certain deterministic approximations to overcome the intractable integrals that appear in the expression for the posterior distribution.

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

2. HIERARCHICAL BAYESIAN ARX MODELS

In this section, we present the two proposed hierarchical Bayesian ARX models both using Student's t distributed excitation noise, as described in Section 2.1. The models differ in how the model orders are incorporated. The two alternatives are presented in Sections 2.2 and 2.3, respectively.

2.1 Student's t distributed innovations

We model the excitation noise as Student's t-distributed, with scale λ and ν degrees of freedom (DOF)

$$e_t \sim \mathcal{S}t(0, \lambda, \nu). \tag{2}$$

This can equivalently be seen as a latent variable model in which e_t is modelled as zero-mean Gaussian with unknown variance $(\lambda z_t)^{-1}$ and z_t is a gamma distributed latent variable. Hence, an equivalent model to (2) is given by

$$z_t \sim \mathcal{G}(\nu/2, \nu/2), \tag{3a}$$

$$e_t \sim \mathcal{N}(0, (\lambda z_t)^{-1}), \tag{3b}$$

where $\mathcal{G}(\alpha, \beta)$ is the gamma distribution with shape α and inverse scale β and $\mathcal{N}(\mu, \sigma^2)$ is the Gaussian distribution with mean μ and variance σ^2.

Note that λ and ν are unknowns, we wish to infer these in the proposed Bayesian models. As we do not know much about these parameters, vague (non-informative) gamma priors are used as in Christmas and Everson [2011]

$$p(\lambda) = \mathcal{G}(\lambda; \alpha_\lambda, \beta_\lambda), \tag{4a}$$

$$p(\nu) = \mathcal{G}(\nu; \alpha_\nu, \beta_\nu), \tag{4b}$$

where α and β denote hyperparameters that we define below. Note that these are standard choices resulting from the property of *conjugate priors*. This type of priors used in combination with a suitable likelihood gives an analytical expression for the posterior, see e.g. Bishop [2006] for other examples of conjugate priors.

2.2 Parametric model order

The first automatic order determination alternative is to infer the order n along with the model parameters. Assume that there exists some maximum order such that $n_a, n_b \leq n_{\max}$, resulting in $n_{\max}^2$ different model hypotheses

$$\mathcal{M}_n : \quad y_t = (\varphi_t^n)^\mathsf{T} \theta^n + e_t, \tag{5}$$

for $n = \{1, 1\}, \{1, 2\}, \ldots, \{n_{\max}, n_{\max}\}$, where

$$\varphi_t^n = (-y_{t-1} \ \cdots \ -y_{t-n_a} \ u_{t-1} \ \cdots \ u_{t-n_b})^\mathsf{T}, \tag{6}$$

denotes the known inputs and outputs, θ^n the model coefficients, and e_t the excitation noise that is assumed to be independent of the input signal. We use a uniform prior distribution over these model hypotheses with order n as

$$p(n) = \begin{cases} 1/n_{\max}^2 & \text{if } n_a, n_b \in \{1, \ldots, n_{\max}\}, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Furthermore, we model the coefficients θ^n as random vectors, with prior distributions

$$p(\theta^n \mid n, \delta) = \mathcal{N}(\theta^n; 0, \delta^{-1} I_{n_a+n_b}), \tag{8}$$

with the same variance δ^{-1} for all orders n and where I_n denotes the $n \times n$ identity matrix. Finally, we place the standard conjugate gamma prior on δ as

$$p(\delta) = \mathcal{G}(\delta; \alpha_\delta, \beta_\delta). \tag{9}$$

All put together, the collection of unknowns of the model is given by

$$\eta = \{\theta^n, n, \delta, z_{1:T}, \lambda, \nu\}. \tag{10}$$

The latent variables $z_{1:T}$, as well as the coefficients' variance δ^{-1}, can be seen as nuisance parameters which are not really of interest, but they will simplify the inference.

2.3 Automatic relevance determination

An alternative approach for order determination is to use ARD. Consider a high-order ARX model with fixed orders $n = \{n_{\max}, n_{\max}\}$. Hence, we overparameterise the model and the ARX coefficients θ will be a vector of fixed dimension $m = 2n_{\max}$. To avoid overfitting, we place a sparseness prior, known as ARD, on the ARX coefficients

$$p(\theta_i \mid \delta_i) = \mathcal{N}(\theta_i; 0, \delta_i^{-1}), \tag{11}$$

with the conjugate distribution on the variance

$$p(\delta_i) = \mathcal{G}(\delta_i; \alpha_\delta, \beta_\delta), \tag{12}$$

for $i = 1, \ldots, m$. The difference between the ARD prior and (8) is that in (11), each coefficient is governed by a different variance, which is i.i.d. according to (12). If there is not enough evidence in the data that the ith parameter should be non-zero, this prior will favor a large value for δ_i which means that the ith parameter in effect will be "switched off". Hence, the ARD prior will encourage a sparse solution; see e.g. MacKey [1994], Neal [1996] for further discussion. When using the ARD prior, the collection of unknowns of the model is given by

$$\eta = \{\theta, \delta_{1:m}, z_{1:T}, \lambda, \nu\}, \tag{13}$$

where θ is the parameter vector of the overparameterised model of order $n_{\max}$.

3. MARKOV CHAIN MONTE CARLO

Assume that we have observed a sequence of input/output pairs $D_T = \{u_{1:T}, y_{1:T}\}$. We then seek the posterior distribution of the model parameters, $p(\eta \mid D_T)$, which is not available in closed form. An MCMC sampler is therefore used to approximately sample from the posterior distribution.

The most fundamental MCMC sampler is known as the Metropolis-Hastings (MH) algorithm. In this method, we propose a new value for the state of the Markov chain from some arbitrary chosen proposal kernel. The proposed value is then accepted with a certain probability, otherwise the previous state of the chain is kept.

A special case of the MH algorithm is the Gibbs sampler. In this method, we loop over the different variables of our model, sampling each variable conditioned on the remaining ones. By using these conditional posterior distributions as proposals, the MH acceptance probability will be exactly one. Hence, the Gibbs sampler will always accept its proposed values. As pointed out by Tierney [1994], it is possible to mix different types of proposals. This will be done in the sampling strategies employed in this work, where we use Gibbs moves for some variables and random walk MH moves for other variables.

A generalisation of the MH sampler is the reversible jump MCMC (RJ-MCMC) sampler [Green, 1995], which allows for moves between parameter spaces of different dimensionality. This approach will be used in this work, for the model presented in Section 2.2. The reason is that when the model order n is seen as a parameter, the dimension of the vector θ^n will change between iterations. An RJ-MCMC sampler can be seen as employing standard MH moves, but all variables that are affected by the changed dimensionality must either be accepted or rejected as a group. That is, in our case, we propose new values for

$\{n, \theta^n\}$ as a pair, and either accept or reject both of them (see step (I-1a) below).

For the ARX model with parametric model order, we employ an RJ-MCMC sampler using the following sweep [1],

(I-1) Order and ARX coefficients:
  (a) Draw $\{\theta^{n^\star}, n^\star\} \mid z_{s+1:T}, \lambda, \delta, D_T$.
  (b) Draw $\delta^\star \mid \theta^{n^\star}, n^\star$.

(I-2) Innovation parameters:
  (a) Draw $z^\star_{s+1:T} \mid \theta^{n^\star}, n^\star, \lambda, \nu, D_T$.
  (b) Draw $\lambda^\star \mid \theta^{n^\star}, n^\star, z^\star_{s+1:T}, D_T$.
  (c) Draw $\nu^\star \mid z^\star_{s+1:T}$.

If we instead consider the ARX model with an ARD prior we use the following sweep, denoted ARD-MCMC,

(II-1) ARX coefficients:
  (a) Draw $\theta^\star \mid z_{s+1:T}, \lambda, \delta_{1:m}, D_T$.
  (b) Draw $\delta^\star_{1:m} \mid \theta^\star$.

(II-2) Innovation parameters:
  (a) Draw $z^\star_{s+1:T} \mid \theta^\star, \lambda, \nu, D_T$.
  (b) Draw $\lambda^\star \mid \theta^\star, z^\star_{s+1:T}, D_T$.
  (c) Draw $\nu^\star \mid z^\star_{s+1:T}$.

The difference between the two methods lies in steps (I-1) and (II-1), where the parameters related to the ARX coefficients are sampled. In steps (I-2) and (II-2), we sample the parameters of the excitation noise distribution, and these steps are essentially the same for both samplers.

4. POSTERIORS AND PROPOSAL DISTRIBUTIONS

In this section, we present the posterior and proposal distributions for the model order and other parameters used by the proposed MCMC methods.

4.1 Model order

Sampling the model order and the ARX coefficients in step (I-1a) is done via a reversible jump MH step. We start by proposing a new model order n', according to some chosen proposal kernel $q(n' \mid n)$. In this work, we follow the suggestion by Troughton and Godsill [1998] and use a constrained random walk with discretised Laplace increments with scale parameter ℓ, i.e.

$$q(n'_a \mid n) \propto \exp(-\ell|n'_a - n_a|), \quad \text{if } 1 \leq n'_a \leq n_{\max}, \tag{14}$$

and analogously for n_b. This proposal will favour small changes in the model order, but allows for occasional large jumps.

Once we have sampled the proposed model order n', we generate a set of ARX coefficients from the posterior distribution

$$\theta^{n'} \sim p(\theta^{n'} \mid n', z_{s+1:T}, \lambda, \delta, D_T) = \mathcal{N}(\theta^{n'}; \mu_{\theta^{n'}}, \Sigma_{\theta^{n'}}). \tag{15}$$

The expressions for the mean and the covariance of this Gaussian distribution are provided in the subsequent section. Now, since the proposed coefficients $\theta^{n'}$ are directly connected to the model order n', we apply an MH accept/reject decision to the pair $\{\theta^{n'}, n'\}$. The MH acceptance probability is given by

$$\begin{aligned}\rho_{nn'} &\triangleq 1 \wedge \frac{p(n', \theta^{n'} \mid z_{s+1:T}, \lambda, \delta, D_T)}{p(n, \theta^n \mid z_{s+1:T}, \lambda, \delta, D_T)} \frac{q(n, \theta^n \mid n', \theta^{n'})}{q(n', \theta^{n'} \mid n, \theta^n)} \\ &= 1 \wedge \frac{p(n' \mid z_{s+1:T}, \lambda, \delta, D_T)}{p(n \mid z_{s+1:T}, \lambda, \delta, D_T)} \frac{q(n \mid n')}{q(n' \mid n)},\end{aligned} \tag{16}$$

where $a \wedge b := \min(a, b)$. Since

$$p(n \mid z_{s+1:T}, \lambda, \delta, D_T) \propto p(y_{1:T} \mid n, z_{s+1:T}, \lambda, \delta, u_{1:T}) p(n), \tag{17}$$

where the prior over model orders is flat according to (7), the acceptance probability can be simplified to [Troughton and Godsill, 1998]

$$\rho_{nn'} = 1 \wedge \frac{\delta^{\frac{n'}{2}} |\Sigma_{\theta^{n'}}|^{\frac{1}{2}} \exp\left(\frac{1}{2}\mu^\mathsf{T}_{\theta^{n'}} \Sigma^{-1}_{\theta^{n'}} \mu_{\theta^{n'}}\right)}{\delta^{\frac{n}{2}} |\Sigma_{\theta^n}|^{\frac{1}{2}} \exp\left(\frac{1}{2}\mu^\mathsf{T}_{\theta^n} \Sigma^{-1}_{\theta^n} \mu_{\theta^n}\right)} \frac{q(n \mid n')}{q(n' \mid n)}.$$

Note by (21) that the acceptance probability does not depend on the actual value of $\theta^{n'}$. Hence, we do not have to carry out the sampling according to (15) unless the proposed sample is accepted.

4.2 ARX coefficients

The ARX coefficients are sampled in step (I-1a) and step (II-1a) of the two proposed MCMC samplers, respectively. In both cases, we sample from the posterior distribution over the parameters; see (15). In this section, we adopt the notation used in the RJ-MCMC sampler, but the sampling is completely analogous for the ARD-MCMC sampler. A "stacked" version of the linear regression model (5) is

$$y_{s+1:T} = \Phi^n \theta^n + e_{s+1:T}, \tag{18}$$

where the regression matrix Φ^n is given by

$$\Phi^n = \begin{pmatrix} -y_s & \cdots & -y_{s-n_a} & u_s & \cdots & u_{s-n_b+1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -y_{T-1} & \cdots & -y_{T-n_a} & u_{T-1} & \cdots & u_{T-n_b} \end{pmatrix}. \tag{19}$$

Here, we have take into account that the initial state of the system is not known, and only use observations from time $s+1$ to T in the vector of observations on the left hand side of (18). For the RJ-MCMC sampler $s = \max(n_a, n'_a)$ and for the ARD-MCMC sampler $s = n_{\max}$.

Let Δ^{-1} be the covariance matrix for the parameter prior, either according to (8) or according to (11), i.e.

$$\Delta^{-1} = \begin{cases} \delta I_{n_a+n_b} & \text{for RJ-MCMC,} \\ \operatorname{diag}(\delta_1, \ldots, \delta_m) & \text{for ARD-MCMC.} \end{cases} \tag{20}$$

Since we condition on the latent variables $z_{s+1:T}$ (and the varince parameter λ^{-1}), the noise term in (18) can be viewed as Gaussian according to (3b). It follows that the posterior parameter distribution is Gaussian, as already stated in (15), with mean and covariance given by

$$\mu_{\theta^n} = \Sigma_{\theta^n} (\Phi^n)^\mathsf{T} (\lambda z_{s+1:T} \circ y_{s+1:T}), \tag{21a}$$

$$\Sigma_{\theta^n} = \left((\Phi^n)^\mathsf{T} \operatorname{diag}(\lambda z_{s+1}, \ldots, \lambda z_T) \Phi^n + \Delta\right)^{-1}, \tag{21b}$$

respectively. Here, $\circ$ denotes elementwise multiplication.

4.3 ARX coefficients variance

We now derive the posterior distributions for the ARX coefficients variance(s), sampled in steps (I-1b) and (II-1b) for the two models, respectively.

Consider first the model described with parametric model order. The ARX coefficients variance δ^{-1} is *a priori* gamma distributed according to (9). The likelihood is given by (8) and an analytical expression for the posterior

[1] The reason for why we condition on some variables from time $s+1$ to T, instead of from time 1 to T, is to deal with the unknown initial state of the system. This will be explained in more detail in Section 4.2.

distribution is easily found as the gamma distributed is a conjugate prior. Thereby motivating the standard choice of a gamma distributed prior for the inverse variance in a Gaussian distribution. It follows from standard results (see e.g. Bishop [2006, p. 100]) that

$$p(\delta \mid \theta^n, n) = \mathcal{G}(\delta; \alpha_\delta^{\text{post}}, \beta_\delta^{\text{post}}), \tag{22}$$

with hyperparameters

$$\alpha_\delta^{\text{post}} = \alpha_\delta + \frac{n_a + n_b}{2}, \quad \text{and} \quad \beta_\delta^{\text{post}} = \beta_\delta + \frac{1}{2}(\theta^n)^\mathsf{T}\theta^n. \tag{23}$$

Similarly, for the ARD model, we get from the prior (12) and the likelihood (11), that the posterior distributions for the ARX coefficients variances are given by

$$p(\delta_i \mid \theta_i) = \mathcal{G}(\delta_i; \alpha_{\delta_i}^{\text{post}}, \beta_{\delta_i}^{\text{post}}), \tag{24}$$

with hyperparameters

$$\alpha_{\delta_i}^{\text{post}} = \alpha_\delta + \frac{1}{2}, \qquad \text{and} \qquad \beta_{\delta_i}^{\text{post}} = \beta_\delta + \frac{1}{2}\theta_i^2, \tag{25}$$

for $i = 1, \ldots, m$.

4.4 Latent variance variables

Let us now turn to the parameters defining the excitation noise distribution. We start with the latent variance variables $z_{s+1:T}$. These variables are sampled analogously in steps (I-2a) and (II-2a). The latent variables are *a priori* gamma distributed according to (3a) and since they are i.i.d., we focus on one of them, say z_t. Note that we here once again have chosen a prior distribution conjugate to the likelihood.

The likelihood model for z_t is given by (5), where the model order now is fixed since we condition on n (in the ARD model, the order is always fixed)

$$p(y_t \mid z_t, \theta^n, n, \lambda, \nu, \varphi_t^n) = \mathcal{N}(y_t, (\varphi_t^n)^\mathsf{T}\theta^n, (\lambda z_t)^{-1}). \tag{26}$$

It follows that the posterior is given by

$$p(z_t \mid \theta^n, n, \lambda, \nu, D_T) = \mathcal{G}(z_t; \alpha_z^{\text{post}}, \beta_{z_t}^{\text{post}}), \tag{27}$$

with the hyperparameters

$$\alpha_z^{\text{post}} = \frac{1}{\nu} + \frac{1}{2}, \qquad \text{and} \qquad \beta_{z_t}^{\text{post}} = \frac{\nu}{2} + \frac{\lambda}{2}\epsilon_t^2. \tag{28}$$

Here, the prediction error ϵ_t is given by

$$\epsilon_t = y_t - (\varphi_t^n)^\mathsf{T}\theta^n. \tag{29}$$

We can thus generate $z_{s+1:T}^\star$ by sampling independently from (27) for $t = s+1, \ldots, T$.

4.5 Innovation scale parameter

The innovation scale parameter λ is sampled in steps (I-2b) and (II-2b). This variable follows a model that is very similar to z_t. The difference is that, whereas the individual z_t variables are i.i.d. and only enter the likelihood model (5) for a single t each, we have the same λ for all time instances. The posterior distribution of λ is thus given by

$$p(\lambda \mid \theta^n, n, z_{s+1:T}, D_T) = \mathcal{G}(\lambda; \alpha_\lambda^{\text{post}}, \beta_\lambda^{\text{post}}), \tag{30}$$

with

$$\alpha_\lambda^{\text{post}} = \alpha_\lambda + \frac{T-s}{2}, \tag{31a}$$

$$\beta_\lambda^{\text{post}} = \beta_\lambda + \frac{1}{2}\epsilon_{s+1:T}^\mathsf{T}(z_{s+1:T} \circ \epsilon_{s+1:T}), \tag{31b}$$

where the prediction errors $\epsilon_{s+1:T}$ are given by (29).

4.6 Innovation DOF

The DOF ν, sampled in steps (I-2c) and (II-2c), is *a priori* gamma distributed according to (4b). The likelihood for this variable is given by (3a). It follows that the posterior of ν is given by

$$\begin{aligned} p(\nu \mid z_{s+1:T}) &\propto p(z_{s+1:T} \mid \nu)p(\nu) \\ &= \prod_{t=s+1}^{T} \mathcal{G}(z_t; \nu/2, \nu/2)\mathcal{G}(\nu; \alpha_\nu, \beta_\nu). \end{aligned} \tag{32}$$

Unfortunately, this does not correspond to any standard distribution. To circumvent this, we apply an MH accept/reject step to sample the DOF. Hence, we propose a value according to some proposal kernel $\nu' \sim q(\nu' \mid \nu)$. Here, the proposal is taken as a Gaussian random walk, constrained to the positive real line. The proposed sample is accepted with probability

$$\rho_{\nu\nu'} = 1 \wedge \frac{p(\nu' \mid z_{s+1:T})}{p(\nu \mid z_{s+1:T})} \frac{q(\nu \mid \nu')}{q(\nu' \mid \nu)}, \tag{33}$$

which can be computed using (32).

5. NUMERICAL ILLUSTRATIONS

We now give some numerical results to illustrate the performance of the proposed methods. First, we compare the average performance of the MCMC samplers with least squares (LS) in Section 5.1. These experiments are included mostly to build some confidence in the proposed method. We then illustrate how the proposed methods are affected by outliers and missing data in Section 5.2. As a final example, in Section 5.3 we illustrate the performance of the RJ-MCMC on real EEG data.

5.1 Average model performance

We evaluate the proposed methods by analysing the average identification performance for 25,000 randomly generated ARX systems. These systems are generated by sampling a uniform number of poles and zeros (so that the resulting system is strictly proper) up to some maximum order, here taken as 30. The poles and zeros are generated uniformly over a disc with radius 0.95.

For each system, we generate $T = 450$ observations[2]. The input signal u_t is generated as Gaussian white noise with standard deviation 0.1. The innovations are simulated from a Student's t distribution, $e_t \sim \mathcal{S}t(0, 1, 2)$. The hyperparameters of the model are chosen as $\alpha_\lambda = \beta_\lambda = \alpha_\nu = \beta_\nu = \alpha_\delta = \beta_\delta = 0.1$.

The data is split into three parts with 150 observations each. The first two parts are used for model estimation, and the last part is used for testing the model. For the LS method, we employ cross validation by first estimating models for all possible combinations of model orders n_a and n_b, such that both are less than or equal to $n_{\max} = 30$, on the first batch of data. We then pick the model corresponding to the best model fit [Ljung, 1999, p. 500]. The full estimation data set (300 observations) is then used to re-estimate the model parameters. For the MCMC methods, we use all the estimation data at once, since these methods comprise automatic order determination and no explicit order selection is made.

[2] When simulating the systems, we run the simulations for 900 time steps, out of which the first 450 observations are discarded, to remove the effect of transients.

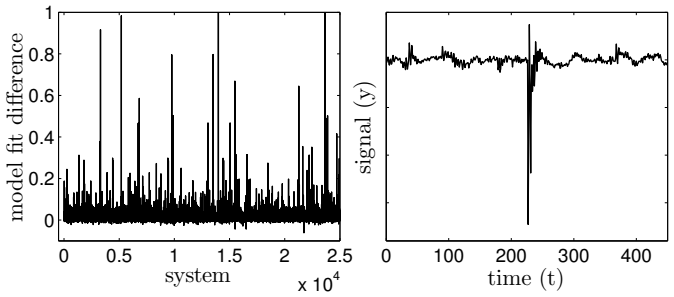

Fig. 1. Left: The difference in model fit between the RJ-MCMC and LS methods. Right: One particular randomly generated ARX model with a large innovation outlier that affects the system output.

The average model fit for the test data, for the 25,000 ARX systems is given in Table 1. We note a slight statistically significant improvement by using the RJ-MCMC method in comparison with the standard LS technique. Also, the RJ-MCMC appear to perform better than the simpler ARD-MCMC method (for this model class). Therefore, we will focus primarily on the former method in the remainder of the numerical illustrations.

Method	mean	CI
LS	77.51	[77.21 77.81]
RJ-MCMC	78.24	[77.95 78.83]
ARD-MCMC	77.73	[77.47 78.06]

Table 1. The average and 95% confidence intervals (CI) for the model fit (in percent) from experiments with $25,000$ random ARX models.

In the left part of Figure 1, the differences in model fit between RJ-MCMC and LS for all 25,000 systems are shown. We note that there are no cases with large negative values, indicating that the RJ-MCMC method performs at least as good as, or better than, LS for the vast majority of these systems. We also note that there are a few cases in which LS is much worse that RJ-MCMC. Hence, the average model fit for LS is deteriorated by the fact that the method fails completely from "time to time". This is not the case for the proposed RJ-MCMC sampler (nor for the ARD-MCMC sampler), which suggests that the proposed method is more robust to variations in the data.

It is interesting to review a typical case with a large difference in model fit between the two methods. Data from such a case is shown in the right part of Figure 1. Here, we see a large jump in the system state. The ARX model with Student's t distributed innovations can, due to the heavy tails of the noise distribution, accommodate for the large output values better than the model with Gaussian noise. The model fit for this system was 46.15% for the RJ-MCMC method and 14.98% for the LS methods.

It is important to note that the use of the LS method is due to its simplicity. For the problem under study the LS method is the maximum likelihood (ML) solution to an ARX model with Gaussian noise and a given model order. The ML problem can of course also be posed for the case where t distributed noise is assumed. Another alternative would be to make use of a prediction error method with a robust norm, such as the Huber or Vapnik norm. A cross validation scheme could also be used to handle the automatic order determination in this setting by an exhaustive search of the model set.

5.2 Robustness to outliers and missing data

We continue by evaluating the proposed models and inference algorithms in the presence of missing data or outliers in the observations. The hypothesis is that, due to the use of Student's t innovations in the model, we should be more robust to such data anomalies than an LS estimate (based on a Gaussian assumption).

In these experiments, the innovations used in the data generation are drawn from a Gaussian distribution with unit variance. We then add outliers or missing observations to the outputs of the systems (i.e. this can be interpreted as an effect of sensor imperfections or measurement noise). This is done by randomly selecting between 1–3 % of the observations in the estimation data, which are modified as described below. In the first set of experiments we add outliers to the selected observations. The size of the outliers are sampled from a uniform distribution $\mathcal{U}(-5y^+, 5y^+)$, with $y^+ = \max|y_t|$. In the second set of experiment, we instead replace the selected observations by zero-mean Gaussian noise with variance 0.01. This is to represent missing data due to sensor errors, resulting in values close to zero compared with the actual observations.

For each scenario, we generate $1,000$ random ARX systems and simulate $T = 450$ observations from each. We then apply the proposed MCMC samplers and LS with cross validation, similarly to the previous sections, but with the modifications described above. Table 2 gives the average results over the $1,000$ randomly generated models with added outliers and missing values, respectively. Here, we have not corrupted the test data by adding outliers or missing observations, not to overshadow the results [3].

The mean results show statistically certain differences between the LS approach and the two proposed methods. We conclude that, in general the proposed MCMC based methods are more robust to data anomalies such as missing observations or outliers.

	Outliers		Missing data	
Method	mean	CI	mean	CI
LS	39.13	[37.86 40.41]	75.20	[74.00 76.40]
RJ-MCMC	70.54	[69.03 72.04]	80.18	[78.74 81.62]
ARD-MCMC	72.46	[71.02 73.91]	81.57	[80.24 82.90]

Table 2. The mean and 95% CIs for the model fit (in percent) from $1,000$ systems with outliers and missing data, respectively.

In Figure 2, the predicted versus the corresponding observed data points are shown for the RJ-MCMC method (stars) and the LS approach (dots), for two of the data batches. It is clearly visible that the LS method is unable to handle the problem with outliers, and the predictions are systematically too small (in absolute value). LS performs better in the situation with missing data, but the variance of the prediction errors is still clearly larger than for the RJ-MCMC method.

5.3 Real EEG data

We now present some results from real world EEG data, which often include large outliers (and therefore deviates from normality). Therefore this data serves as a good example for when the propose methods are useful in a practical setting. The deviations from normality can

[3] If an outlier is added to the test data, the model fit can be extremely low even if there is a good fit for all time points apart from the one where the outlier occurs.

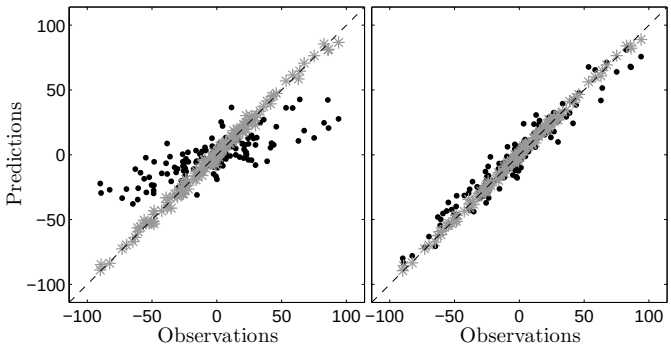

Fig. 2. Predictions vs. observations for data with outliers (left) and data with missing observations (right). The model fit values for the outlier data example are 91.6% for the RJ-MCMC (stars) and 40.2% for LS (dots). The corresponding values for the missing data example are 94.4% and 75.7%.

be seen in Figure 3, by observing the signal and the Q-Q plot, i.e. a comparison between two distributions by plotting their quantiles against each other [Wilk and Gnanadesikan, 1968].

The RJ-MCMC method with Student's t innovations is used to estimate an AR model for this data set. The resulting estimated posterior density for the model order is shown in the lower parts of Figure 3. Knowing this posterior, allows for e.g. weighting several different models together using the estimated density values.

In addition, we can also estimate the posterior density of the DOF of the innovations. This density is useful for quantifying deviations from normality, as the Gaussian distribution is asymptotically recovered from the Student's t distribution with infinite DOF. As the maximum posterior value is attained at approximately 4.0 DOF, this confirm non-Gaussian innovations.

We have thereby illustrated the usefulness of the proposed methods, both for parameter inference but also for estimating useful posterior densities not easily obtainable in the LS framework.

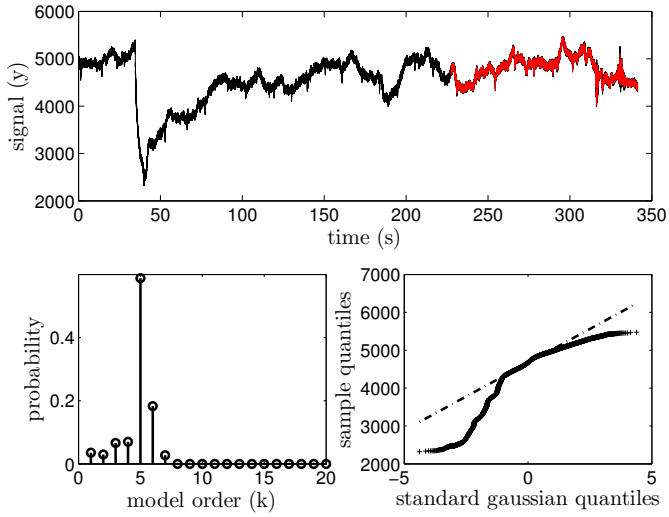

Fig. 3. Upper: the EEG signal collected on one specific channel and patient. Lower left: The estimated posterior model order density from the RJ-MCMC method. Lower right: The Q-Q plot for the data set. The model fit for the results in this figure is 85.6%.

6. CONCLUSIONS AND FUTURE WORK

We have considered hierarchical Bayesian ARX model with Student's t distributed innovations. This was considered to be able to capture non-Gaussian elements in the data and to increase robustness. Furthermore, both models contain a mechanism for automatic order selection. To perform inference in these models, we also derived two MCMC samplers: a reversible jump MCMC (RJ-MCMC) sampler and a standard Gibbs sampler.

Three numerical examples have been presented, providing evidence that the proposed models provide increased robustness to data anomalies, such as outliers and missing data. We have shown that the proposed methods perform on average as good as (ARD-MCMC) or better (RJ-MCMC) than LS with cross validation, when the true system is in the model class. Another benefit with the proposed methods is that they provide a type of information which is not easily attainable using more standard techniques. As an example, this can be the posterior distribution over the model orders, as illustrated in Figure 3.

There are several interesting avenues for future research, and we view the present work as a stepping stone for estimating more complex models. The next step is to generalize the proposed methods to encompass e.g. OE and ARMAX models. A more far reaching step is to generalize the methods to nonlinear systems, possibly by using Particle MCMC methods [Andrieu et al., 2010]. It is also interesting to further analyse the use of sparseness priors in this setting.

ACKNOWLEDGEMENTS

The EEG data was kindly provided by Eline Borch Petersen and Thomas Lunner at Eriksholm Research Centre, Oticon A/S, Denmark.

REFERENCES

C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B*, 72(3):269–342, 2010.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, USA, 2006.

S. P. Brooks, P. Giudici, and G. O. Roberts. Efficient construction of reversible jump Markov chain Monte Carlo proposal distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(1):3–55, February 2003.

J. Christmas and R. Everson. Robust autoregression: Student-t innovations using variational Bayes. *IEEE Transactions on Signal Processing*, 59(1):48–57, 2011.

S. Godsill. On the relationship between Markov chain Monte Carlo methods for model uncertainty. *Journal of Computational and Graphical Statistics*, 10(2):230–248, 2001.

P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrica*, 82(4):711–732, 1995.

L. Ljung. *System identification, Theory for the user*. System sciences series. Prentice Hall, Upper Saddle River, NJ, USA, second edition, 1999.

D. J. C. MacKey. Bayesian non-linear modelling for the prediction competition. *ASHRAE Transactions*, 100(2):1053–1062, 1994.

R. M. Neal. *Bayesian Learning for Neural Networks*. Springer, 1996.

C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

L. Tierney. Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22(4):1701–1728, 1994.

P. T. Troughton and S. J. Godsill. A reversible jump sampler for autoregressive time series. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1998.

M. B. Wilk and R. Gnanadesikan. Probability plotting methods for the analysis of data. *Biometrika*, 55(1):1–17, March 1968.