Technical report from Automatic Control at Linköpings universitet

Robust ARX models with automatic order determination and Student's t innovations

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

Division of Automatic Control

E-mail: johan.dahlin@isy.liu.se, lindsten@isy.liu.se, schon@isy.liu.se, adrian.wills@newcastle.edu.au

15th December 2011

Report no.: LiTH-ISY-R-3041

Submitted to 16th IFAC Symposium on System Identification, Brussels, Belgium, 2012

Address:
Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

WWW: http://www.control.isy.liu.se

AUTOMATIC CONTROL
REGLERTEKNIK
LINKÖPINGS UNIVERSITET

Abstract

ARX models is a common class of models of dynamical systems. Here, we consider the case when the innovation process is not well described by Gaussian noise and instead propose to model the driving noise as Student's t distributed. The t distribution is more heavy tailed than the Gaussian distribution, which provides an increased robustness to data anomalies, such as outliers and missing observations. We use a Bayesian setting and design the models to also include an automatic order determination. Basically, this means that we infer knowledge about the posterior distribution of the model order from data. We consider two related models, one with a parametric model order and one with a sparseness prior on the ARX coefficients. We derive Markov chain Monte Carlo samplers to perform inference in these models. Finally, we provide three numerical illustrations with both simulated data and real EEG data to evaluate the proposed methods.

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

Robust ARX models with automatic order determination and Student's t innovations

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: ARX models is a common class of models of dynamical systems. Here, we consider the case when the innovation process is not well described by Gaussian noise and instead propose to model the driving noise as Student's t distributed. The t distribution is more heavy tailed than the Gaussian distribution, which provides an increased robustness to data anomalies, such as outliers and missing observations. We use a Bayesian setting and design the models to also include an automatic order determination. Basically, this means that we infer knowledge about the posterior distribution of the model order from data. We consider two related models, one with a parametric model order and one with a sparseness prior on the ARX coefficients. We derive Markov chain Monte Carlo samplers to perform inference in these models. Finally, we provide three numerical illustrations with both simulated data and real EEG data to evaluate the proposed methods.

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

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

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. In many applications, the excitation noise is assumed to be Gaussian. 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). However, two problems that are often encountered in practice are,

(1) The appropriate order for the model is unknown. It might even be the case that there is no single "best" model order.
(2) The observed data is non-Gaussian in nature. This can for instance be due to the presence of outliers in the observations.

In this work, we propose two Bayesian ARX models and inference algorithms, which address both of these issues. The proposed models differs from a "standard" ARX model on two accounts. First, instead of assuming Gaussian innovations, we model the excitation noise as Student's t distributed. 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 by assuming Student's t distributed innovations in the model, the proposed inference method will be more robust to model errors and outliers in the observations, a property which we illustrate in this work.

Second, to deal with the fact that the "true" model order is unknown, we wish to build some form of automatic order selection into the model. Here, we consider two different alternative models. In the first alternative, we let the model order n be a parameter of the model, which we infer alongside the other unknown parameters. This is done in a Bayesian context, which results 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 $p(\eta \mid D_T)$, where η denotes the unknown model parameters and $D_T \triangleq \{y_{1:T}, u_{1:T}\}$ (here $y_{1:T} \triangleq \{y_1, \ldots y_T\}$) denotes the observed sequence of inputs and outputs. In order to do this we will employ a Markov chain Monte Carlo (MCMC) sampler (see e.g. Robert and Casella [2004]). The idea behind MCMC is to generate a Markov chain which admits the target distribution, i.e. $p(\eta \mid D_T)$, as stationary distribution. Hence, these methods allow us to "indirectly" sample from the target distribution, even when direct sampling is impossible. We can then use the samples from the Markov chain to compute estimates under the posterior parameter distribution.

The inference problem resulting from the use of an ARD prior we solve using standard MCMC algorithms. The case when the model order n is explicitly included in the parameter vector η is more challenging, due to the fact that we are now dealing with a parameter space of varying dimension. Hence, we need to build a Markov chain that moves over spaces of different dimension. This will be done using a so called reversible jump MCMC (RJ-MCMC) algorithm introduced by Green [1995].

The use of RJ-MCMC to estimate the model order and the parameters of an AR model driven by Gaussian noise,

[⋆] 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.

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 application of Student's t distributed innovations, which affect the posterior distributions used in the MCMC sampling. AR processes with Student's t innovations are considered also 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.

2. BAYESIAN ARX MODELS

In this section we present the two proposed Bayesian ARX models. Common to the models is the use of a Student's t distributed excitation noise, as described in Section 2.1. The models differ in how the model order is treated, and 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)

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

Equivalently, we can adopt a latent variable model in which e_t is modelled as zero-mean Gaussian, but with unknown variance. The precision of this Gaussian is given by λz_t, where the latent variable z_t follows a gamma distribution. Hence, a model that is equivalent to (2) is given by

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

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

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

We wish to infer the parameters λ and ν from data and since we propose Bayesian models, we place prior probability densities on these parameters. Similarly to Christmas and Everson [2011], we use (vague) gamma priors according to,

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

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

2.2 Parametric model order

One alternative for automatic order determination is to consider the model order n as a parameter, which we estimate alongside the other parameters of the model. Assume that we can constrain the model order as $n_a \leq n_{\max}$ and $n_b \leq n_{\max}$ (for notational simplicity, we use the same maximum order for both polynomials). We then consider $n_{\max}^2$ different model hypotheses

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

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

$$\varphi_t^n = \begin{pmatrix} -y_{t-1} & \cdots & -y_{t-n_a} & u_{t-1} & \cdots & u_{t-n_b} \end{pmatrix}^\mathsf{T}. \tag{6}$$

Each of these hypotheses is given the same *a priori* probability. Hence, if we let n be a random variable, its prior distribution is given by

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

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

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

with the same precision δ for all n. Finally, we place a gamma prior on δ

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

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 ARX coefficients precision δ, 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

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

for $i = 1, \ldots, m$. The difference between the ARD prior and (8) is that in (11), each coefficient is governed by a different precision δ_i, which are 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

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

3. MARKOV CHAIN MONTE CARLO

Assume that we have observed a sequence of input/output pairs $D_T = \{y_{1:T}, u_{1:T}\}$. We then seek the posterior distribution of the model parameters, $p(\eta \mid D_T)$. This posterior distribution is not available in closed form, and we shall make use of an MCMC sampler to address the inference problem.

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 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 the posterior distributions as proposals, the MH acceptance probability will be exactly 1. 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 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 favor small changes in the model order, but allows for occasional large jumps. The proposal kernel can of course be chosen differently, if we so wish.

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 mean and 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 $a \wedge b := \min(a, b)$. Furthermore, 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 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

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

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 be the precision matrix for the parameter prior, either according to (8) or according to (11), i.e.

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

Since we condition on the latent variables $z_{s+1:T}$ (and the precision parameter λ), 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 precision

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

[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.

described with parametric model order. The ARX coefficients precision δ is a priori gamma distributed according to (9). The likelihood is given by (8) and it follows (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

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

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

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

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

4.4 Latent precision variables

Let us now turn to the parameters defining the excitation noise distribution. We start with the latent precision 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. 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

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

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 points. 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

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

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. We then illustrate how the proposed methods are affected by outliers and missing data in Section 5.2. As a final example, in Section **??** 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 25000 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 then 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 validation. 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 largest model fit [Ljung, 1999, p. 500]. We then use the full estimation data set (300 observations) to reestimate 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.

The average model fit for the validation data, for the 25000 ARX systems are 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.

[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.

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 25000 random ARX models.

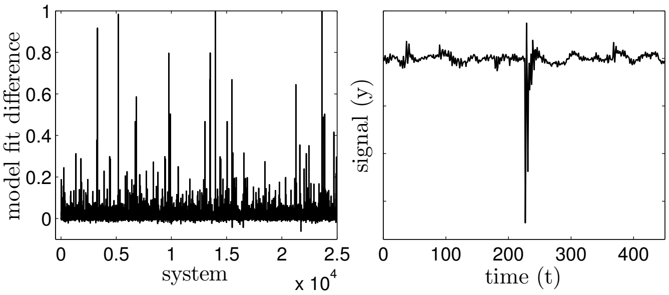

Fig. 1. Left: The difference in model fit between the RJ-MCMC and LS methods. Right: One particular randomly generated ARX model with a large innovation outlier that affects the system output.

In the left part of Figure 1, the differences in model fit between RJ-MCMC and LS for all 25000 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 completely fails 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 corresponding model fit for this system were 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 norm. However, neither of these methods would be able to account for the fact that the model order is unknown.

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 seen 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 Gaussian noise with standard deviation 0.1, to represent missing data due to sensor errors.

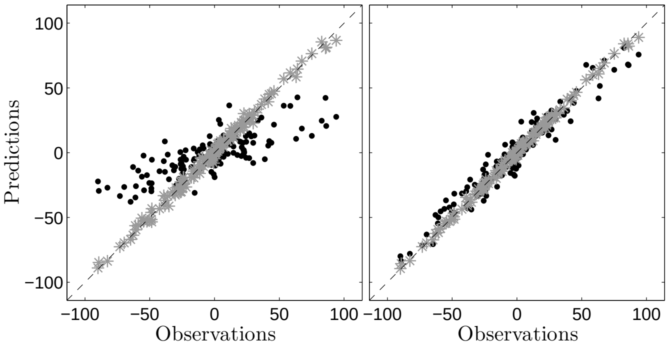

Fig. 2. Predictions vs. observations for data with outliers (left) and data with missing observations (right). The model fit values for the outlier data example are 91.6% for the RJ-MCMC (stars) and 40.2% for LS (dots). The corresponding values for the missing data example are 94.4% and 75.7%.

For each scenario, we generate 1000 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 1000 randomly generated models with added outliers and missing values, respectively. Here, we have not corrupted the validation 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 1000 systems with outliers and missing data, respectively.

In Figure 2, the predicted versus 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.

[3] If an outlier is added to the validation 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.

5.3 Real EEG data

We now present some results from real world EEG data. That EEG data often include large outliers (and therefore deviations from normality) is well-known and therefore this data serves as a good example for practical applications of the proposed methods.

The EEG data[4] shown in Figure 3 is taken from Keirn [1988] and is clearly non-Gaussian. The RJ-MCMC method with Student's t innovations is used to estimate an AR model for this data set. The resulting estimated posterior densities for the model order and the degrees-of-freedom in the innovation distribution are shown in the lower parts of Figure 3.

This illustrates the advantages of the RJ-MCMC compared with traditional methods, as it allows for weighting several different models using the estimated posterior density values. In addition, the estimated posterior density of the DOF of the innovations, is useful for quantifying deviations from normality. This 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 4.1 DOF, we conform that the innovations are clearly not Gaussian. Finally, we conclude that the RJ-MCMC method is useful in estimating AR models with non-Gaussian excitation noise and also returns other useful information not provided by more traditional methods, such as LS.

6. CONCLUSIONS AND FUTURE WORK

We have considered a Bayesian approach to ARX modeling and have proposed two related Bayesian ARX models. To be able to capture non-Gaussian elements in the data and to attain an increased robustness to data anomalies as well as model errors, the innovations are modeled as Student's t distributed. Furthermore, both models contain some mechanism for automatic order selection, based on a parametric order for the first model and an ARD sparseness prior for the second model. To perform inference in these models, we derive two MCMC samplers. For the model with parametric model order we consider reversible jump MCMC (RJ-MCMC) moves, to account for the fact that the dimensionality of the parameter space is changing over iterations. For the model with an ARD prior, we use a more standard MCMC sampler, which we denote ARD-MCMC. Three numerical examples have been presented, providing evidence that the proposed models provide increased robustness to data anomalies, such as outliers and missing data, compared to LS. Furthermore, by evaluating the proposed methods on a large number of randomly generated ARX systems, 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. This was done to provide some confidence in the proposed ideas. The same experiments suggest that a parametric model order is preferable over an ARD prior for ARX models. The gain in average performance for RJ-MCMC over LS, can be traced back to certain systems/data realisations, for which LS gives bad model estimates which deteriorate the average fit. The RJ-MCMC method is more robust against these occasional drops in performance.

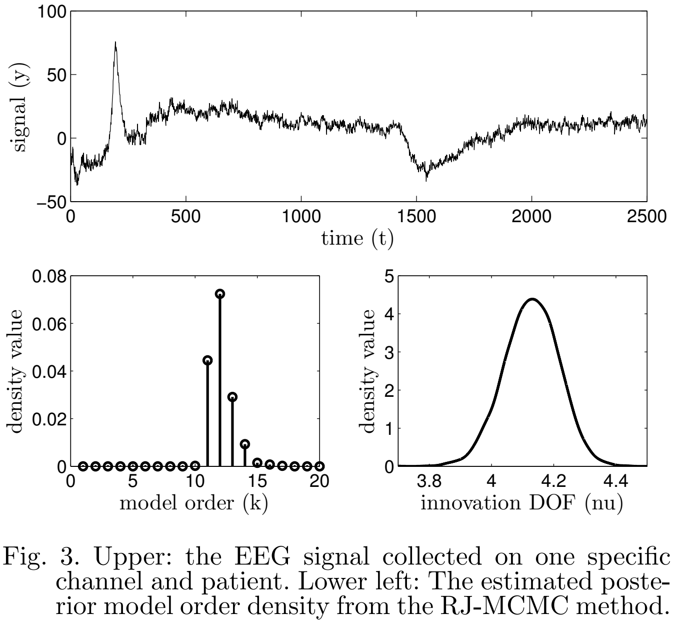

Fig. 3. Upper: the EEG signal collected on one specific channel and patient. Lower left: The estimated posterior model order density from the RJ-MCMC method. Lower right: The estimated posterior degrees-of-freedom density for the Student's t distributed innovations from the RJ-MCMC method.

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 order of an ARX model, 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 other linear models, e.g. OE and ARMAX models. A more far reaching step is to generalize the methods to nonlinear system, possibly starting with nonlinear ARX by using Particle MCMC methods [Andrieu et al., 2010]. It is also interesting to further analyse the differences between the two proposed models and if any other sparseness prior is a better choice than the ARD.

[4] The data is available online at the homepage: `http://www.cs.colostate.edu/eeg/eegSoftware.html`.

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.

Z. A. Keirn. Alternative modes of communication between man and machine. Master's thesis, Purdue University, 1988.

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.

Avdelning, Institution
Division, Department

Division of Automatic Control
Department of Electrical Engineering

Datum
Date

2011-12-15

Språk
Language

□ Svenska/Swedish
☒ Engelska/English
□ ________________

Rapporttyp
Report category

□ Licentiatavhandling
□ Examensarbete
□ C-uppsats
□ D-uppsats
☒ Övrig rapport
□ ________________

URL för elektronisk version

`http://www.control.isy.liu.se`

ISBN

—

ISRN

—

Serietitel och serienummer
Title of series, numbering

ISSN
1400-3902

LiTH-ISY-R-3041

Titel
Title

Robust ARX models with automatic order determination and Student's t innovations

Författare
Author

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

Sammanfattning
Abstract

ARX models is a common class of models of dynamical systems. Here, we consider the case when the innovation process is not well described by Gaussian noise and instead propose to model the driving noise as Student's t distributed. The t distribution is more heavy tailed than the Gaussian distribution, which provides an increased robustness to data anomalies, such as outliers and missing observations. We use a Bayesian setting and design the models to also include an automatic order determination. Basically, this means that we infer knowledge about the posterior distribution of the model order from data. We consider two related models, one with a parametric model order and one with a sparseness prior on the ARX coefficients. We derive Markov chain Monte Carlo samplers to perform inference in these models. Finally, we provide three numerical illustrations with both simulated data and real EEG data to evaluate the proposed methods.

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