Bayesian Volterra System Identification Using Reversible Jump MCMC Algorithm

O. Karakuş[a,*], E.E. Kuruoğlu[b], M.A. Altınkaya[a]

[a] *İzmir Institute of Technology (IZTECH), Electrical-Electronics Engineering, İzmir, Turkey*
[b] *ISTI-CNR, via G. Moruzzi 1, 56124, Pisa, Italy*

Abstract

Volterra systems have had significant success in modelling nonlinear systems in various real-world applications. However, it is generally assumed that the nonlinearity degree of the system is known beforehand. In this paper, we contribute to the literature on *Volterra system identification* (VSI) with a numerical Bayesian approach which identifies model coefficients and the nonlinearity degree concurrently. Although this numerical Bayesian method, namely *reversible jump Markov chain Monte Carlo* (RJMCMC) algorithm has been used with success in various model selection problems, our use is in a novel context in the sense that both memory size and nonlinearity degree are estimated. The aforementioned study ensures an anomalous approach to RJMCMC and provides a new understanding on its flexible use which enables trans-structural transitions between different classes of models in addition to transdimensional transitions for which it is classically used. We study the performance of the method both on synthetically generated data and on OFDM communications via QPSK and 16-QAM over a nonlinear channel.

Keywords: Reversible jump MCMC, Volterra system identification, Nonlinearity degree estimation, Nonlinear channel estimation.

1. Introduction

Nonlinear models can be favorable compared to linear ones in many real life phenomena which exhibit nonlinear characteristics. On the other hand, usage of these models is limited because they do not have easy to implement solutions for estimating nonlinear model parameters. *Linear-in-the-parameters* nonlinear models do not share these shortcomings as far as various mathematical methods which are developed for linear models can be applied easily [1].

Volterra models are appealing linear-in-the parameters models for nonlinear modelling for several reasons. Firstly, they are flexible enough to represent various nonlinear systems since continuously differentiable transfer functions

*Corresponding author

Email addresses: oktaykarakus@iyte.edu.tr (O. Karakuş), ercan.kuruoglu@isti.cnr.it (E.E. Kuruoğlu), mustafaaltinkaya@iyte.edu.tr (M.A. Altınkaya)

can be easily approximated by Volterra models with Taylor series expansion. Moreover, various nonlinear differential equations such as Lotka-Volterra, Schrödinger [2], can be rewritten as a Volterra system. Secondly, their inverse are also Volterra type which provides considerable ease in the identification of these systems [3]. This paper is interested in the nonlinearity degree estimation of Volterra models, within the *system identification* (SI) problem.

The application areas of Volterra models cover almost all areas of signal processing, including speech, image, communications, audio, mechanical systems, etc. To name a few: in audio, Volterra model has been used for parametric loudspeaker system identification in [4] and for acoustic echo cancellation in [5]. Nonlinear communication channels in satellite links have been modelled as sparse third order (cubic) Volterra systems which were estimated via adaptive algorithms [6]. Volterra system models were applied also in [7] to coherent optical fiber systems outperforming the adaptive reference methods on equalizing the fiber link channel effects. Volterra systems with complex coefficients have been used in blind identification of single-input-single-output (SISO) communication channels with second order nonlinearity in [8] and for LTI FIR multiple-input-multiple-output (MIMO) systems in [9]. A general approach in the literature, which is shared by the mentioned applications, is to apply Volterra SI (VSI) methodology to the models with predetermined nonlinearity degree and system memory [4–9]. Preknowledge of nonlinearity degree is an unrealistic assumption for most of these applications and estimating the nonlinearity degree of the nonlinear model is of utmost importance.

The Bayesian approach proposed in this paper utilizes *reversible jump Markov chain Monte Carlo* (RJMCMC) algorithm in the VSI problem to estimate the nonlinearity degree, the system memory and model coefficients at the same time. Model space includes linear and nonlinear models with different degrees of nonlinearity while the generally accepted procedure is to use RJMCMC in spaces which include only models of the same class that are spaces containing only linear system parameters or spaces containing only nonlinear system parameters of the same degree of nonlinearity. However, although RJMCMC has been used for transdimensional sampling only, the formulation in the original paper by Green [10] does exclude the potential to explore spaces which include different classes of models, such as linear and nonlinear spaces, or nonlinear spaces with different degrees. Restructuring RJMCMC sampling strategy from that defined by [10] sampling from the spaces of different classes of models, enables applying RJMCMC in more complicated problems such as nonlinear model identification or nonlinear SI.

In this paper, firstly we contribute to the literature with a Bayesian VSI scheme and in contrast to previous works, it provides means to estimate nonlinearity degree in addition to the memory size, hence droping the requirement of preknowledge of nonlinearity degree. This offers greater flexibility in modelling, which can cover a wide spectrum of nonlinear characters observed in the measured data.

Secondly, we have broaden the interpretation of RJMCMC transitions with *trans-structural* transitions beyond trans-dimensionality by performing the original formulation in [10] between different structural models. This also offers a Bayesian test procedure to define the nonlinear relationship between input and output data sets of real life experiments, such as mechanical systems, optical communication systems, biological systems, in terms of Volterra series expansion model structure. In our previous works, this potential was exploited in the estimation of polynomial autoregressive and polynomial moving average models [11, 12].

Furthermore, in addition to the model orders the proposed method also estimates the model coefficients with superior performance both in applications on synthetically generated data sets including a nonlinear communication channel estimation problem. We provide also model selection results obtained by *Akaike information criterion* (AIC) and *Bayesian information criterion* (BIC) as benchmarks to which our method can be compared. Performance comparison for estimating the model coefficients is provided for error measure *normalized mean square error* (NMSE) using *nonlinear least squares* (NLS) estimation.

Rest of the paper is organized as follows: background information for Volterra system models is presented in Section 2. The general RJMCMC procedure and the proposed approach for trans-structural RJMCMC are expressed in Section 3. Construction of RJMCMC for identifying Volterra systems is examined in Section 5. Section 6 exhibits simulation setup, reference methods, simulation results and performance comparison study. Section 7 concludes the paper with a discussion of experimental results.

2. Volterra System Models

A discrete time Volterra model with the output $y(l)$ is given by [13]:

$$y(l) = \mu + \sum_{m=1}^{p} \sum_{\tau_1=1}^{q} \cdots \sum_{\tau_m=\tau_{m-1}}^{q} h^{(m)}_{\tau_1,\ldots,\tau_m} \prod_{j=1}^{m} x(l-\tau_j) \tag{1}$$

where $x(\cdot)$ and $y(\cdot)$ refer to the input and output of the model, respectively and $h^{(m)}_{\tau_1,\ldots,\tau_m}$ denotes the m^{th} order discrete Volterra model coefficients (kernels). The nonlinearity degree is represented by p and q specifies the system memory size. This Volterra model can be represented with the notation: $\mathrm{V}(p,q)$.

Observing (1), Volterra models can be represented in matrix-vector form by using the *linear-in-the-parameters* property:

$$\mathbf{y} = \mathbf{X}\mathbf{h}^{(\mathbf{p},\mathbf{q})} \tag{2}$$

where the $\eta \times 1$ coefficient vector $\mathbf{h}^{(\mathbf{p},\mathbf{q})}$ and $n \times \eta$ data matrix $\mathbf{X}$ are given by:

$$\mathbf{h}^{(p,q)} = \left[\begin{array}{ccccccccc} h_1^{(1)}, & h_2^{(1)}, & \ldots, & h_q^{(1)}, & h_{1,1}^{(2)}, & h_{1,2}^{(2)}, & \ldots, & h_{q,q}^{(2)}, & \ldots, & h_{q,q,\ldots,q}^{(p)} \end{array} \right]^T, \tag{3}$$

$$\mathbf{X} = \begin{bmatrix} x(0) & x(-1) & \ldots & x(1-q) & x^2(0) & x(0)x(-1) & \ldots & x^2(1-q) & \ldots & x^p(1-q) \\ x(1) & x(0) & \ldots & x(2-q) & x^2(1) & x(1)x(0) & \ldots & x^2(2-q) & \ldots & x^p(2-q) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x(n-1) & x(n-2) & \ldots & x(n-q) & x^2(n-1) & x(n-1)x(n-2) & \ldots & x^2(n-q) & \ldots & x^p(n-q) \end{bmatrix}. \tag{4}$$

where n represents the data length. The number of Volterra coefficients has been denoted by η and can be calculated for a $\mathrm{V}(p,q)$ model by:

$$\eta = \binom{p+q}{p} - 1 = \frac{(p+q)!}{p!\,q!} - 1. \tag{5}$$

2.1. Identification of Volterra Systems

Methods used in SI problems provide information about the uncertainties which describe formulations or mathematical expressions about the unknown system in the case of lack of structural (physical) information about the system. SI methods are well established when system to be identified is linear. Most of real life applications, however, have somewhat nonlinear nature, and solution for a nonlinear SI problem can be difficult since the underlying systems may have a large number of possible nonlinearities and the number of possible model structures might be very high [14].

In the literature, adaptive algorithms are very popular in recent years. These methods generally perform Volterra system coefficient estimations based on *nonlinear least mean squares* (NLMS), *least mean pth power* (LMP), *recursive least squares* (RLS) and extended Kalman filters [7, 15–21]. Furthermore, genetic algorithms [22, 23], QR decomposition [24], neuro-fuzzy [25] and neural network [26] architectures have also been used in VSI studies. For all these studies, nonlinearity degree of Volterra model is assumed to be known.

Several studies have used Bayesian methods for system identification problems [27, 28].In addition, *simulated annealing* (SA) [29] and *transitional Markov chain Monte Carlo* (TMCMC) [14] are also used in SI applications of nonlinear dynamical systems.

3. RJMCMC in a New Perspective

RJMCMC was introduced by Peter Green in [10] as a method for transdimensional sampling between spaces of different dimensions. However, the original formulation of Green lends itself to a much wider interpretation than just exploring spaces (”*jumping*” in RJMCMC jargon) of different dimensions. The same formulation can be used to explore spaces of different types such as linear and nonlinear variable spaces. This is more than just exploring spaces of different sizes corresponding to the dimension of the parameter vector.

In the literature, RJMCMC has generally been used in linear model identification problems. Authors of [30] studied model uncertainty problem for *autoregressive* (AR) models by exploring the spaces for different AR orders via partial and full conditional proposals. RJMCMC have been used in *autoregressive integrated moving average* (ARIMA) models by [31] and in *fractional* ARIMA (ARFIMA) models by [32]. In [33–35], RJMCMC has been employed to Bayesian analysis of mixtures of distributions in Gaussian, Poisson and symmetric α-stable, respectively.

RJMCMC has also been employed to the problems of identifying the nonlinear model, *threshold MA* (TMA) in [36]. In addition, [37] employed RJMCMC in restoring nonlinearly distorted AR signals.

3.1. General Methodology

In [10], Green defined RJMCMC as an extended and generalized version of the Metropolis-Hastings (M-H) algorithm [38] and stated that it should therefore have wide applicability in model determination problems.

Assume a transition from a space $x \in \mathcal{X}$ to a state $y \in \mathcal{X}$. The acceptance ratio for the M-H algorithm can be defined by:

$$\min\left\{1, \frac{\pi(y)q(x|y)}{\pi(x)q(y|x)}\right\} \tag{6}$$

where $\pi(\cdot)$ represents the target distribution and $q(y|x)$ refers to proposal distribution from state x to y.

Green’s generalization on M-H algorithm defines the densities $\pi(x)$ and $q(y|x)$ with respect to an arbitrary measure as $\pi(dx)$ and $q(dy|x)$. The transition kernel of the Markov chain has been constructed in two steps, firstly drawing a new candidate state y and then accepting this transition with a probability $\alpha(x, y)$.

Assuming that $\pi(dx)q(dy|x)$ has a density f with respect to a dominating symmetric measure ξ on $\mathcal{X} \times \mathcal{X}$, then the detailed balance for a transition defined above is given by:

$$\alpha(x,y)f(x,y) = \alpha(y,x)f(y,x), \tag{7}$$

$$\alpha(x,y)\pi(dx)q(dy|x) = \alpha(y,x)\pi(dy)q(dx|y). \tag{8}$$

This equation can be solved as in the standard M-H procedure by retaining the detailed balance and making the acceptance probabilities as large as possible if we choose $\alpha(x,y)$ as:

$$\alpha(x,y) = \min\left\{1, \frac{\pi(dy)q(dx|y)}{\pi(dx)q(dy|x)}\right\}. \tag{9}$$

Following [10], when the current state is x with parameter space $\boldsymbol{\theta}$, a move type m is proposed with probability $Pr(x \to y)$, which changes dimension, and takes the state to y with parameter space $\boldsymbol{\theta}^*$.

This transition is required to sample an auxiliary random vector, $\mathbf{u}$ from distribution $q_1(\mathbf{u})$. Another vector $\mathbf{u}'$ will be generated from distribution $q_2(\mathbf{u}')$ to switch back to the state x from y. In order to guarantee the detailed balance, the dimension matching should be satisfied provided by $dim(x) + dim(\mathbf{u}) = dim(y) + dim(\mathbf{u}')$. Then, the resulting acceptance probability $\alpha(x,y)$ of RJMCMC is defined by:

$$\alpha(x,y) = \min\left\{1, \frac{\pi(y)Pr(y \to x)q_2(\mathbf{u}')}{\pi(x)Pr(x \to y)q_1(\mathbf{u})}\left|\frac{\partial(y,\mathbf{u}')}{\partial(x,\mathbf{u})}\right|\right\}, \tag{10}$$

where $\left|\frac{\partial(y,\mathbf{u}')}{\partial(x,\mathbf{u})}\right|$ is the magnitude of the Jacobian which is needed to account for change of variables.

For the constructed RJMCMC structure two types of moves can be defined. Moves of the first type, *between-model* moves, namely *birth* and *death* moves, change dimension up and down respectively. The others, *within-model* moves, which we call as *life* move, update the parameter space by applying a classical M-H algorithm.

3.2. Trans-Structural RJMCMC

The formulation of Green offers deeper interpretation of transition between spaces which is not limited to transdimensional sampling. Thus, exploring spaces with the "same dimensions" but different structures (say trans-structural), or both different dimensions and structures, are possible by applying reversible jump mechanism of Green. Transitions from a linear model to a nonlinear model may be indicated as an example for this. Trans-structural RJMCMC reveals the great potential of RJMCMC within much wider scenarios including transitions from states with the same dimensionality and different structures other than being transdimensional.

Suppose that there is a state space $\mathcal{X} = \bigcup_k \{k\} \times \mathcal{R}^{n_k}$ denotes union of k subspaces which includes models with indicator k, $X_k = \{k\} \times R^{n_k}$ and each can be defined as different types. We mean with different types, for example, linear and nonlinear models or models which are driven with different probability distributions, etc. Now suppose we have two subspaces $\mathcal{X}_1$ and $\mathcal{X}_2$ whose types are different where the dimensions n_1 and n_2 may be equal. Target density π is proper on both subspaces and defined with respect to n_1 and n_2 dimensional Lebesgue measures, respectively. The subspaces $\mathcal{X}_1$ and $\mathcal{X}_2$ have parameters spaces $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ and both have proper densities in $\mathcal{R}^{n_1}$ and $\mathcal{R}^{n_2}$.

Now, define a move type "m", which performs a transition from state $x \in \mathcal{X}$ to state $y \in \mathcal{X}$, with probability p_m and retains the same state with probability $1 - p_m$. This transition will be applied by a transition kernel in two steps as indicated in the previous section. Thus, detailed balance in (8) should be provided. Transitions between models of different structures, contrary to the previous approaches, may include both *birth* of new parameters and *death* of existing parameters at the same time. In addition, number of parameters may be the same for both states. These transitions propose to switch models with different structures, and hence will be named as **switch** moves in trans-structural RJMCMC concept.

Nevertheless, proposing vectors of variables and change-of-variables operations are needed to define parameter vector for candidate state. So, for this type of problems, we define a vector $\mathbf{u}$ of length l_1 for a transition from x to y. Also, we define a vector $\mathbf{u}'$ of length l_2 for the reverse transition from y to x. Both of the vectors $\mathbf{u}$ and $\mathbf{u}'$ are sampled from proper densities q_1 and q_2 with respect to Lebesgue measures in $\mathcal{R}^{l_1}$ and $\mathcal{R}^{l_2}$, respectively.

Following the assumption in [10], the general form of the trans-structural acceptance ratio also necessitates defining the density f (the Radon-Nikodym derivative) of a symmetric measure ξ on $\mathcal{X} \times \mathcal{X}$ which dominates the density $\pi(dx)q(dy|x)$. Now, let the density f be selected for both directions of the transitions as:

$$f(x, y) = \pi(x) q_1(\mathbf{u}) p_m, \tag{11}$$

$$f(y, x) = \pi(y) q_2(\mathbf{u}') p_{m^R} \left| \frac{\partial(y, \mathbf{u}')}{\partial(x, \mathbf{u})} \right| \tag{12}$$

where p_{m^R} is the reverse move probability of m. Then, the acceptance ratio can be easily constructed by (9):

$$\alpha(x, y) = \min \left\{ 1, \frac{\pi(y) p_{m^R} q_2(\mathbf{u}')}{\pi(x) p_m q_1(\mathbf{u})} \left| \frac{\partial(y, \mathbf{u}')}{\partial(x, \mathbf{u})} \right| \right\}, \tag{13}$$

Remark. It can be clearly stated that the acceptance ratio of trans-structural RJMCMC including transitions between the same dimensionality of the models with different structures in (13), has the same form with the one which is derived by Green [10] for the transdimensional transitions in (10). So, Green's formulation can be directly used within much wider implementations. Although RJMCMC has been defined as a model determination tool in transdimensional cases, it will be more meaningful to define RJMCMC as a general model determination tool whether or not the parameter spaces are of different dimensions. As far as the subspaces are of different structures, transitions between them require the reversible jump mechanism. Then, in the acceptance ratio, the cost of these transitions are fulfilled in the Jacobian term, which is required to be calculated due to the change of variables operation.

In order to see that the use of RJMCMC is not limited to transdimensional models, a simple example is given as follows. For this simple example, we consider 2 models each having the same number of parameters, however, one of them is a linear Volterra model say V(1,2), and the other one is nonlinear, say V(2,1). The general expressions of the models from (1), are given below:

$$y(l) = h_1^{(1)} x(l-1) + h_2^{(1)} x(l-2), \tag{14}$$

$$y(l) = h_1^{(1)} x(l-1) + h_{1,1}^{(2)} x^2(l-1). \tag{15}$$

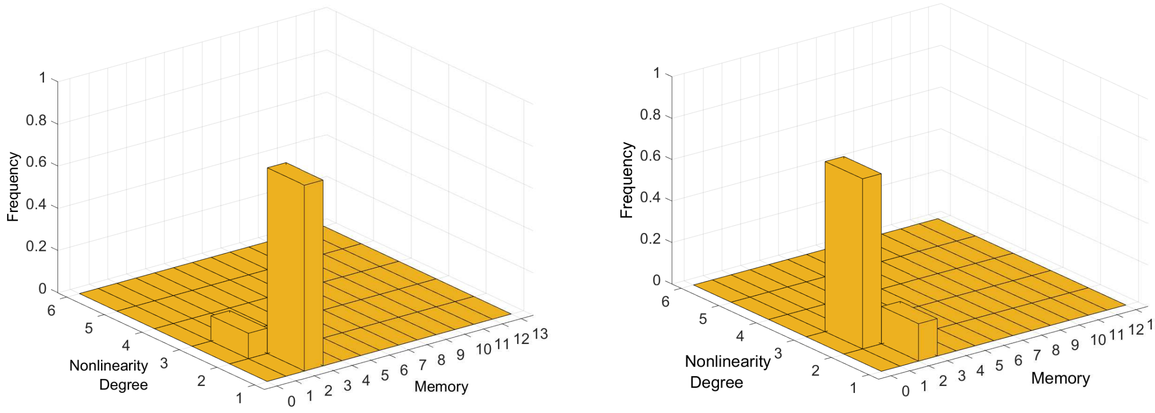
Suppose, we are given the data, $\mathbf{y}$, observed from one of the candidate models and there are two parameter subspaces $\mathcal{X}_k$, which are $\mathcal{X}_1 = \{1\} \times \mathcal{R}^2$ and $\mathcal{X}_2 = \{2\} \times \mathcal{R}^2$. Parameter subspaces will be defined for V(1,2) as $x = (1, h_1^{(1)}, h_2^{(1)}) \in \mathcal{X}_1$ and for V(2,1) as $x' = (2, h_1^{(1)}, h_{1,1}^{(2)}) \in \mathcal{X}_2$. Also, we define a move which switches subspaces with probability p_m and retains the same subspace with probability $1 - p_m$. When it remains in the same subspace, RJMCMC is going to update model coefficients.

When we need to make a transition from V(1,2) to V(2,1), although the parameter dimensions are the same, just one of the model coefficients is common, say $h_1^{(1)}$. The remaining candidate coefficient, say $h_{1,1}^{(2)}$, should be proposed and $h_2^{(1)}$ will be set to 0. For the reverse move from V(2,1) to V(1,2), the mechanism will be the same; $h_2^{(1)}$ will be proposed and $h_{1,1}^{(2)}$ will be set to 0. Coefficient updating mechanism for the moves can be defined as:

$$\text{Move } m \rightarrow \hat{h}_1^{(1)} = h_1^{(1)}, \hat{h}_{1,1}^{(2)} = u, h_2^{(1)} = 0, \tag{16}$$

$$\text{Reverse move } m^R \rightarrow \hat{h}_1^{(1)} = h_1^{(1)}, \hat{h}_2^{(1)} = u', h_{1,1}^{(2)} = 0, \tag{17}$$

where coefficients with hats on them, e.g $\hat{h}_1^{(1)}$ represent the candidate model coefficients, variables $\mathbf{u}$ and $\mathbf{u}'$ have been



(a) V(1,2) (b) V(2,1)

Figure 1: Toy example model estimation histograms - (a) V(1,2) (b) V(2,1).

proposed from the densities q_1 and q_2, respectively which makes Jacobian of the change-of-variables operation unity.

The acceptance ratio from (13) appears as:

$$\alpha(x, y) = \min\left\{1, \frac{\pi(x'|\mathbf{y}) p_{m^R} q_2(\mathbf{u}')}{\pi(x|\mathbf{y}) p_m q_1(\mathbf{u})} \left|\frac{\partial(x', \mathbf{u}')}{\partial(x, \mathbf{u})}\right|\right\}, \tag{18}$$

where $\pi(\cdot|\mathbf{y})$ represents the target distribution of interest given the data $\mathbf{y}$.

A computer simulation has been performed for this problem, and RJMCMC has been constructed to decide the true model given both input and output data of the Volterra models and estimate the model coefficients at the same time. For both of the models, RJMCMC detects true model with %100 performance after 100 realizations. For each model, histograms of model estimates belonging to a single realization are shown in Figure 1.

4. On Convergence and Complexity of (RJ)MCMC Algorithms

The central objective of MCMC sampling is to create a Markov chain with a stationary distribution equal to the target distribution or the posterior for the model parameters. If we run the simulation long enough, the distribution of our samples converges to this stationary distribution. This makes MCMC fundamentally more outstanding than the other sampling algorithms such as importance sampling, etc [39]. In the absence of techniques to select the right run length a priori, the convergence of (RJ)MCMC requires online monitoring of the estimation statistics such as the mean and the autocorrelation. The estimation of optimal run length a priori is still an open problem.

There are some advanced statistical studies [40–43] in the literature which propose methods for monitoring convergence. In particular, Gelman and Rubin in [40, 41] have proposed a way to replicate multiple chains to decide whether or not the algorithm achieves stationarity. Brooks and Guidici in [42] generalize the method of Gelman and Rubin in a two-way *analysis of variance* (ANOVA) based method. In [44], Castelloe and Zimmerman presents a two-way ANOVA based approach as in [42] but they extended the approach from univariate to multivariate cases. A more recent approach which is a specific distance-based diagnostics have been proposed by Sisson and Fan in [45]. This diagnostic is designed for trans-dimensional chains and covers the modelling scenarios like finite-mixture problems and change point analyses.

In addition to the convergence of RJMCMC, another key issue is the computational complexity which is high as in the other sampling algorithms. The computational complexity is directly related with their convergence and is also an open problem. However, there some studies which investigates the computational complexity of these methods. For example, [46] stated that computational complexity of MCMC methods are lower than generic maximum likelihood and extremum estimation methods when log-likelihood or quasi-likelihood are nonconcave or nonsmooth. Moreover, it is also stated that computational time of MCMC algorithms is the polynomial in the dimension of parameter space.

Considering all these, we will deal with the computational gain of RJMCMC in this study. RJMCMC calculates posterior probabilities for models automatically using a hierarchial MCMC sampling scheme hence avoids visiting all candidate models. It uses likelihood and prior and learns from the data in order to visit only plausible model classes. Other sampling methods in the literature such as Nested sampling, *transitional MCMC* (TMCMC), etc. need to enumerate and obtain posteriors for each model class. This superiority of RJMCMC becomes very clear in the presence of a large number of candidate model classes and RJMCMC provides computational gains above the sampling methods which perform exhaustive search on model class space.

5. Implementation of RJMCMC for Volterra systems identification

For the purposes of this study, firstly we define what refers to a linear space and a nonlinear space within the scope of VSI (Please see [47] for axioms of a linear space).

Proposition. 5.1. *It can be easily stated that if a Volterra model, $V(p, q)$ with $p = 1$ and $q \in Q$, namely $V(1, q)$, is a linear space S_q, then S_q satisfies all the axioms of a linear space.*

Proof. Using (1), a $V(1, q)$ model can be expressed as:

$$y(l) = \mu + \sum_{i}^{q} h_i^{(1)} x(l - i). \tag{19}$$

It is seen that model output n-vector $\mathbf{y} = [y(1), y(2), \ldots, y(n)]$ is a linear combination of the input n-vector $\mathbf{x} = [x(1), x(2), \ldots, x(n)]$. Thus, it can be easily shown that a V(1,q) model satisfies all the linear space axioms and is closed under both addition and scalar multiplication. □

Proposition. 5.2. *Assume that a Volterra model, V(p, q) with $p > 1$ is a nonlinear space, $S_{p,q}$ for $p > 1$ and $q \in Q$, then $S_{p,q}$ does not satisfy at least one of the axioms of linear space definition.*

Proof. Assume we define two nonzero processes, $y(l)$ and $z(l)$ from a V(2, 1) model, with model inputs $x(l)$ and $w(l)$, respectively as:

$$y(l) = h_1^{(1)} x(l-1) + h_{1,1}^{(2)} x^2(l-1), \tag{20}$$

$$z(l) = h_1^{(1)} w(l-1) + h_{1,1}^{(2)} w^2(l-1). \tag{21}$$

Let us define new processes, $r(l)$ and $t(l)$, by using the processes above:

$$r(l) = y(l) + z(l), \tag{22}$$

$$t(l) = x(l) + w(l). \tag{23}$$

Thus, for a linear model, process $r(l)$ should be:

$$r(l) = h_1^{(1)} t(l-1) + h_{1,1}^{(2)} t^2(l-1). \tag{24}$$

To show this, start from the definition of $r(l)$:

$$r(l) = y(l) + z(l) \tag{25a}$$

$$r(l) = h_1^{(1)} x(l-1) + h_{1,1}^{(2)} x^2(l-1) + h_1^{(1)} w(l-1) + h_{1,1}^{(2)} w^2(l-1), \tag{25b}$$

$$r(l) = h_1^{(1)} \left(x(l-1) + w(l-1)\right) + h_{1,1}^{(2)} \left(x^2(l-1) + w^2(l-1)\right), \tag{25c}$$

$$r(l) = h_1^{(1)} t(l-1) + 2h_{1,1}^{(2)} x(l-1)w(l-1) - 2h_{1,1}^{(2)} x(l-1)w(l-1) + h_{1,1}^{(2)} \left(x^2(l-1) + w^2(l-1)\right), \tag{25d}$$

$$r(l) = h_1^{(1)} t(l-1) + h_{1,1}^{(2)} \left[x(l-1) + w(l-1)\right]^2 - 2h_{1,1}^{(2)} x(l-1)w(l-1), \tag{25e}$$

$$r(l) = h_1^{(1)} t(l-1) + h_{1,1}^{(2)} t^2(l-1) - 2h_{1,1}^{(2)} x(l-1)w(l-1). \tag{25f}$$

The term $-2h_{1,1}^{(2)} x(l-1)w(l-1)$ in (25f) is nonzero if $h_{1,1}^{(2)}$ is nonzero. Thus, the sequence $r(l)$ does not correspond to a V(2,1) model output and is not closed under addition. It is straightforward that this result can be generalized to all

Volterra models with nonlinearity degree, $p > 1$. Then, a Volterra model, $\mathrm{V}(p, q)$ with $p > 1$ is not a linear space, or equivalently is a nonlinear space, $S_{p,q}$. □

Corollary. 5.3. *Linear and nonlinear Volterra systems can be defined as linear and nonlinear spaces, respectively under the assumption of the propositions 5.1 and 5.2.*

5.1. Defining The Likelihood

The Gaussianity of the output distribution of a Volterra system when the input is normally distributed, is not guaranteed due to the polynomial operations on the input. However, in a previous study [48], it was shown that output distribution of a narrowband Volterra system with white inputs is Gaussian. Following this, a Volterra system whose memory tends to infinity, generates Gaussian outputs due to the summation of a large number of terms following the central limit theorem.

On the other hand, the likelihood is expressed as a measure of how well the estimated model represents the observed data in Bayesian SI studies. For the purposes of this study, we are assuming that the model prediction, $\hat{\mathbf{y}} = [\hat{y}(1), \hat{y}(2), \ldots, \hat{y}(n)]$, and observed system output, $\mathbf{y} = [y(1), y(2), \ldots, y(n)]$ satisfy the prediction error equation [49]:

$$\mathbf{y} = \hat{\mathbf{y}} + \mathbf{e}. \tag{26}$$

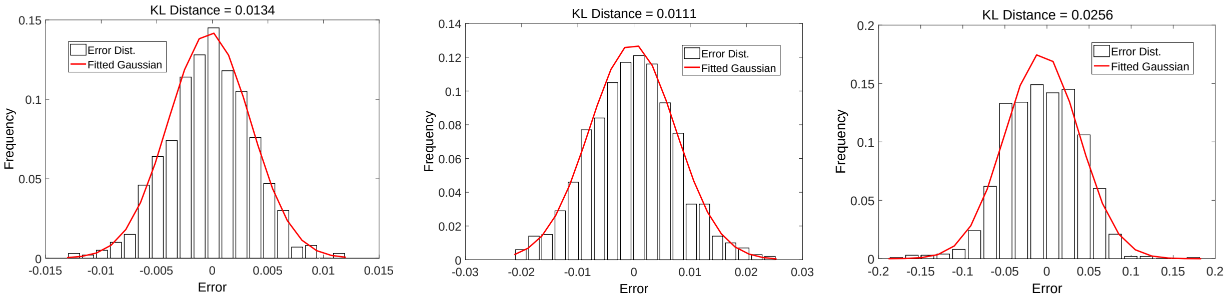
In previous studies [29, 49–51], error-prediction model is assumed to be zero mean Gaussian. In Figure 2, prediction error distributions of three Volterra models which are used for the simulations in this study are depicted. *Kullback-Leibler* (KL) divergence values are calculated with the fitted Gaussian distributions and it has been clearly seen that prediction error distributions for all three Volterra models are Gaussian with a 0.05 significance value of KL divergence.

Thus, the likelihood function can be written simply, using a Gaussian error-prediction model as:

$$f(\mathbf{y}|\theta) = (2\pi\sigma_e^2)^{-n/2} \exp\left(\frac{1}{2\sigma_e^2}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2\right) \tag{27}$$

$$\approx \mathcal{N}(\mathbf{e}|\mathbf{0}, \sigma_e^2\mathbf{I}_n). \tag{28}$$

where $\boldsymbol{\theta}$ is a vector including all the parameters of $\{p, q, \mathbf{h}^{(p,q)}, \sigma_e^2, \sigma_h^2\}$, n is the length of observed data vector $\mathbf{y}$. Also $\mathbf{e} = [e(1), e(2), \ldots e(n)]$ corresponds to the prediction error and σ_e^2 is the error variance.



(a) V(1,10) (b) V(2,5) (c) V(3,3)

Figure 2: Prediction error histograms and fitted Gaussians for models - (a) V(1,10) (b) V(2,5) (c) V(3,3).

5.2. *Hierarchical Bayes Model*

Target distribution of RJMCMC, namely the joint posterior distribution, $f(\boldsymbol{\theta}|\mathbf{x})$ can be decomposed via Bayes theorem for the parameter vector $\boldsymbol{\theta} = \{p, q, \mathbf{h}^{(p,q)}, \sigma_e^2, \sigma_h^2\}$:

$$f(p, q, \mathbf{h}^{(p,q)}, \sigma_e^2, \sigma_h^2|\mathbf{y}) \propto f(\mathbf{y}|p, q, \mathbf{h}^{(p,q)}, \sigma_e^2) f(\mathbf{h}^{(p,q)}|p, q, \sigma_h^2) f(\sigma_h^2) f(\sigma_e^2) f(q) f(p). \tag{29}$$

5.3. *Prior Selection*

In the absence of real prior information, use of noninformative priors is common practice [52]. In the previous studies for time series model determination problems using uniform prior for model order has been a common choice [11, 30, 31]. In addition, as stated in [33], results obtained by uniform priors, can be easily converted to those corresponding to other priors, using the identity:

$$f^*(k, \boldsymbol{\theta}^{(k)}|y) \propto f(k, \boldsymbol{\theta}^{(k)}|y) \frac{f^*(k)}{f(k)} \tag{30}$$

where $f^*(\cdot|y)$ represents the posterior for the prior f^*.

So in this study, we define upper bounds p_{max} and q_{max} for model order values p and q, respectively and assume that the model orders are independent and each model is equally likely. Therefore, uniform priors for the model memory q, and the nonlinearity degree p are used:

$$f(q) = \mathcal{U}(1, q_{\max}) \qquad \text{and} \qquad f(p) = \mathcal{U}(1, p_{\max}). \tag{31}$$

Volterra model coefficients are assumed to be normally distributed a priori and for the variances, σ_e^2 and σ_h^2, we use conjugate priors which are inverse Gamma [31]:

$$f(\mathbf{h}^{(p,q)}|p, q, \sigma_h^2) = \mathcal{N}(\mathbf{h}^{(p,q)}|\mathbf{0}, \sigma_h^2 \mathbf{I}_\eta), \tag{32}$$

$$f(\sigma_h^2) = \mathcal{IG}(\sigma_h^2|\alpha_h, \beta_h), \tag{33}$$

$$f(\sigma_e^2) = \mathcal{IG}(\sigma_e^2|\alpha_e, \beta_e). \tag{34}$$

5.4. Acceptance Ratio and Moves

RJMCMC has three different moves to perform the VSI study . These are, *between-model (switch), within-model (life)* and *update* moves.

5.4.1. Between-Model Move (Switch)

Between-model move corresponds to a move which explores the spaces of different Volterra models at each time it is proposed. Models which are proposed to be switched have different structures and their space dimension can be different or the same.

The acceptance ratio for a **switch** move from (p, q) to (p', q'), is defined as $\alpha_{\text{switch}} = \min\{1, r_{\text{switch}}\}$. Then, r_{switch} is:

$$r_{\text{switch}} = \frac{f(\mathbf{y}|p', q', \mathbf{h}^{(p',q')}, \sigma_e^2)}{f(\mathbf{y}|p, q, \mathbf{h}^{(p,q)}, \sigma_e^2)} \times \frac{f(\mathbf{h}^{(p',q')}|p', q', \sigma_h^2)}{f(\mathbf{h}^{(p,q)}|p, q, \sigma_h^2)} \times \frac{\chi(\mathbf{u}')}{\chi(\mathbf{u})} \times \left|\frac{\partial(\mathbf{h}^{(p',q')}), \mathbf{u}')}{\partial(\mathbf{h}^{(p,q)}, \mathbf{u})}\right|. \tag{35}$$

where $\chi(\cdot)$ will be defined in (45). Model changes are proposed by switch moves and in order to turn back to the previous state after a switch move another switch move should be proposed. Consequently, the reverse move of the switch move is itself. Thus, the ratio p_{m^R}/p_m in (13) is equal to 1 and invisible in (35).

The target joint posterior distribution is proportional to the product of likelihood and priors via Bayes theorem, and hence first two terms in (35) correspond to likelihood and prior ratios, respectively. Proposal ratio is given as the third term and the magnitude of the Jacobian is shown as the fourth term.

5.4.2. Within-Model Move (Life)

RJMCMC not only estimates model orders of a system, but also estimates the coefficients of the model. Hence, the proposed and accepted coefficients in between-model moves, are updated in within-model move, namely the **life** move. A life move will be applied in a case when RJMCMC intends to remain at the same model. Acceptance ratio of the life move is defined as $\alpha_{\text{life}} = \min\{1, r_{\text{life}}\}$. Hence, r_{life} is:

$$r_{\text{life}} = \frac{f(\mathbf{y}|p,q,\widehat{\mathbf{h}}^{(p,q)},\sigma_e^2)}{f(\mathbf{y}|p,q,\mathbf{h}^{(p,q)},\sigma_e^2)} \times \frac{f(\widehat{\mathbf{h}}^{(p,q)}|p,q,\sigma_h^2)}{f(\mathbf{h}^{(p,q)}|p,q,\sigma_h^2)} \times \frac{\psi(\mathbf{h}^{(p,q)}|p,q,\widehat{\mathbf{h}}^{(p,q)})}{\psi(\widehat{\mathbf{h}}^{(p,q)}|p,q,\mathbf{h}^{(p,q)})}). \tag{36}$$

Updating model coefficients includes proposing from the distribution $\psi(\cdot)$:

$$\widehat{\mathbf{h}}^{(p,q)} \sim \psi(\widehat{\mathbf{h}}^{(p,q)}|p,q,\mathbf{h}^{(p,q)}) \tag{37}$$

$$= \mathcal{N}(\widehat{\mathbf{h}}^{(p,q)}|\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n^{-1}), \tag{38}$$

where $\boldsymbol{\mu}_n = \sigma_e^{-2}\boldsymbol{\Sigma}_n^{-1}\mathbf{X}^T\mathbf{y}$, and $\boldsymbol{\Sigma}_n = \sigma_e^{-2}\mathbf{X}^T\mathbf{X} + \sigma_h^{-2}\mathbf{I}_\eta$.

5.4.3. Update Move - Updating Variances

RJMCMC setup for VSI problem includes an error term within most of the definitions. The variance of this error term, σ_e^2 is updated at each iteration via Gibbs Sampling. The full conditional distribution for σ_e^2 is constructed as derived in [30]:

$$f(\sigma_e^2|\mathbf{y},p,q,\mathbf{h}^{(p,q)}) \propto f(\mathbf{y}|p,q,\mathbf{h}^{(p,q)},\sigma_e^2)f(\sigma_e^2) \tag{39}$$

$$\approx \mathcal{N}(\mathbf{e}|\mathbf{0},\sigma_e^2\mathbf{I}_n)\mathcal{IG}(\sigma_e^2|\alpha_e,\beta_e) \tag{40}$$

$$= \mathcal{IG}(\sigma_e^2|\alpha_{en},\beta_{en}), \tag{41}$$

where $\alpha_{en} = \alpha_e + \frac{1}{2}n$ and $\beta_{en} = \beta_e + \frac{1}{2}\mathbf{e}^{\mathrm{T}}\mathbf{e}$.

Similarly, the full conditional distribution for σ_h^2 is obtained as [30]:

$$f(\sigma_h^2|\mathbf{y},p,q,\mathbf{h}^{(p,q)}) \propto f(\mathbf{h}^{(p,q)}|\sigma_h^2)f(\sigma_h^2) \tag{42}$$

$$\approx \mathcal{N}(\sigma_h^2|\mathbf{0},\sigma_h^2\mathbf{I}_\eta)\mathcal{IG}(\sigma_h^2|\alpha_h,\beta_h) \tag{43}$$

$$= \mathcal{IG}(\sigma_h^2|\alpha_{hn},\beta_{hn}), \tag{44}$$

where $\alpha_{hn} = \alpha_h + \frac{1}{2}\eta$ and $\beta_{hn} = \beta_h + \frac{1}{2}(\mathbf{h}^{(p,q)})^{\mathrm{T}}\mathbf{h}^{(p,q)}$ and η has been defined in (5).

5.5. Proposing Candidates

Each RJMCMC iteration requires to select one of the switch or life move firstly with probabilities $\mathrm{P}_{\mathrm{switch}}$ and $\mathrm{P}_{\mathrm{life}}$. Uniform prior is selected for all candidate switchable models (with probability $\mathrm{P}_{\mathrm{switch}}/\rho$ for ρ possible models).

All candidate coefficients should be proposed from the proposal distribution. For instance, in case of a switch move corresponding to a model change from $p = 1$ to $p' = 2$ when $q = 2$, $\lambda = 5 - 2 = 3$ candidate coefficients are needed to be proposed. The λ-vector $\mathbf{u}$ has been proposed from a multivariate Gaussian distribution and $\chi(\mathbf{u})$ is assumed to be:

$$\chi(\mathbf{u}) = \mathcal{N}\left(\mathbf{0}, \left(\frac{\sigma_h^2}{E[|\mathbf{y}|]}\right) I_\lambda\right), \tag{45}$$

where $E[|\mathbf{y}|]$ is the expected value of the absolute value of the data vector $\mathbf{y}$. The variance of the joint distribution is chosen to depend on the data. Different data sets, the magnitudes of which vary in different ranges, constitute distinct limits for proposal distribution $\chi(.)$. This data-dependent *adhoc* choice adds variety for candidate proposals according to the given data.

Moreover, the proposal distribution for candidate coefficients is selected in a way that the candidates will be independent from recent coefficients. Consequently, the change of variables operation is accomplished through an identity function and thus the Jacobian equals to unity.

6. Experimental Analysis

In this section, we study the performance of the proposed VSI algorithm experimentally. The block diagram of the proposed Bayesian VSI procedure has been shown in Figure 3 for a system whose input and output are defined with the vectors $\mathbf{x}$ and $\mathbf{y}$, respectively. Moreover, additive noise sequences for these input and output vectors are $\mathbf{u}$ and $\mathbf{w}$, respectively.

Estimated model order parameter pair $(\widehat{p}, \widehat{q})$ and resulting model coefficient vector $\widehat{\mathbf{h}}^{(\widehat{p},\widehat{q})}$ will be used to generate one-step ahead prediction of the output data, $\widehat{\mathbf{y}}$, by using the Volterra model expression in (1).

6.1. Simulation 1: Synthetically Generated Data

The proposed method has been employed in synthetically generated data sets for this simulation scenario. 3 Volterra models which are V(1, 10), V(2, 5) and V(3, 3) have been implemented. Each model has been given an

Table 1: RJMCMC Algorithm for VSI

input : I/O data set for VSI

Initialize parameters.

for $i \leftarrow 1$ **to** N_{iter} **do**
- **Choose** Life or Switch move with probabilities P_{life}, P_{switch}.
- **if** *Switch is selected* **then**
  - **Sample** candidate model orders, p' and q'.
  - **Perform** *Switch* move in Section 5.4.1 to **calculate** acceptance ratio, $A = \alpha_{\text{switch}}$.
- **else if** *Life is selected* **then**
  - **Candidate** model orders $\rightarrow p' = p^{(i-1)}, q' = q^{(i-1)}$.
  - **Perform** *Life* move in Section 5.4.2 to **calculate** acceptance ratio, $A = \alpha_{\text{life}}$.
- **end**
- **Sample** a random variable $u \sim \mathcal{U}(0, 1)$.
- **if** $u \leq A$ **then**
  - $p^{(i)} = p'$ and $q^{(i)} = q'$.
  - **Update** other parameters with proposed ones.
- **else**
  - $p^{(i)} = p^{(i-1)}$ and $q^{(i)} = q^{(i-1)}$.
  - **Do not change** the values of other parameters.
- **end**
- **Perform** *Update* move in Section 5.4.3 to update variances, σ_e^2 and σ_h^2.

end

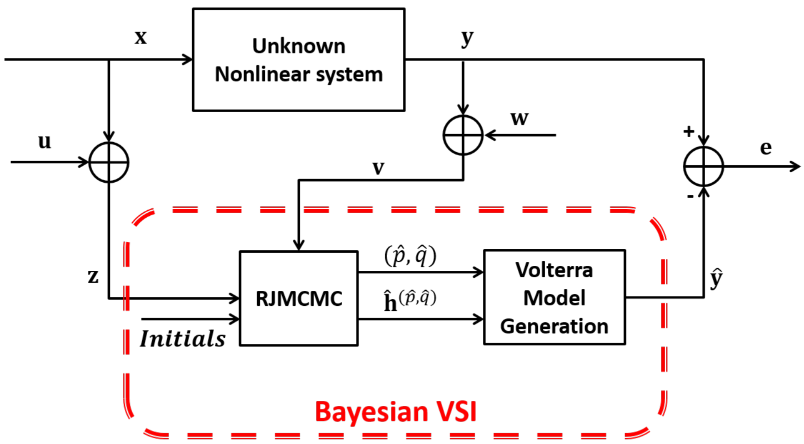


Figure 3: The proposed method VSI block diagram.

input sequence which is a Gaussian process of mean 0 and variance 1 and outputs for each model have been collected. Each data set has a length of $1,000$ samples and mean value, μ, is chosen as 0 for simplicity. Four cases have been employed in order to show the performance of the proposed methods under different conditions (See Table 2).

Table 2: Cases for Simulation 1

	Details
Case 1	Both I/O are noise free
Case 2	Output is corrupted by a white Gaussian noise process of mean 0 and variance 0.1
Case 3	Output is corrupted by a colored Gaussian noise process. The white noise in Case 2 is filtered by an FIR filter, and the output of the filter is used to corrupt the output.
Case 4	Both I/O are corrupted by white Gaussian noise processes of mean 0 and variance 0.1

Table 3: Volterra model coefficients for simulation 1

$V(p,q)$	$\mathbf{h}^{(p,q)} = [\mathbf{h}^{(1)}, \quad \mathbf{h}^{(2)}, \quad \dots, \quad \mathbf{h}^{(p)}]^{\mathrm{T}}$	Calculated SNR(dB) values$^{(*)}$
V(1,10)	$\mathbf{h}^{(1)} = [0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]$	14.13/22.62/10.42/14.19
V(2,5)	$\mathbf{h}^{(1)} = [0.7, 0, 0.2, 0, -0.7]$ $\mathbf{h}^{(2)} = [0, 0.1, 0, 0, -0.25, 0.15, 0, 0.42, 0.02, 0, 0.7, 0, -0.31, 0, 0.28]$	13.52/22.24/10.42/13.58
V(3,3)	$\mathbf{h}^{(1)} = [-0.06, 0.2331, -1.3619]$ $\mathbf{h}^{(2)} = [0, 0.7, 0, 0.3, -0.25, 0.15]$ $\mathbf{h}^{(3)} = [0.5, 0, 0, -0.44, 0.15, -0.25, 0, -0.37, 0, 0.58]$	17.69/26.33/10.44/17.77

$^{(*)}$ Calculated SNR values for each model refer to the values in decibels for Case 2/Case 3/Case 4-Input/Case 4-Output, respectively.

Initial values for hyperparameters of prior distribution of σ_e^2, are selected as $\alpha_e = 1$ and $\beta_e = 1$ and those for σ_h^2, are selected as $\alpha_h = 35$ and $\beta_h = 2$. The initial nonlinearity degree p_0 and system memory q_0 are set to 1 and upper bounds p_{max} and q_{max} are set to 5 and 12, respectively. $\mathbf{h}^{(p_0,q_0)}$ is sampled from the prior distribution in (32). Move probabilities, $\mathrm{P}_{\mathrm{switch}}$ and $\mathrm{P}_{\mathrm{life}}$ are both selected as 0.5.

Model order estimation performance of RJMCMC is compared to two commonly used model order selection methods AIC and BIC. The equations for these are given below:

$$\mathrm{AIC} = 2N + n\log(\mathrm{RSS}/n), \tag{46}$$

$$\mathrm{BIC} = \log(n)N + n\log(\mathrm{RSS}/n), \tag{47}$$

where N is number of parameters for the model, n refers to the data length and RSS corresponds to *the residual sum of squares* which is calculated as:

$$\text{RSS} = \mathbf{y}^{\mathbf{T}}\mathbf{y} - \mathbf{y}^{\mathbf{T}}\mathbf{X}(\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{y}. \tag{48}$$

AIC rewards goodness of fit but penalizes the number of estimated parameters of the model. BIC is more informed then AIC and the penalty term of BIC is more stringent than the penalty term of AIC. Consequently, BIC tends to favor smaller models than AIC.

A similar penalization is also present in RJMCMC whenever model tries to add redundant variables. For example, increasing order by one and setting the additional coefficient to zero does not change the likelihood, but the prior takes a lower value than before, yielding a posterior probability lower than the previous one [53].

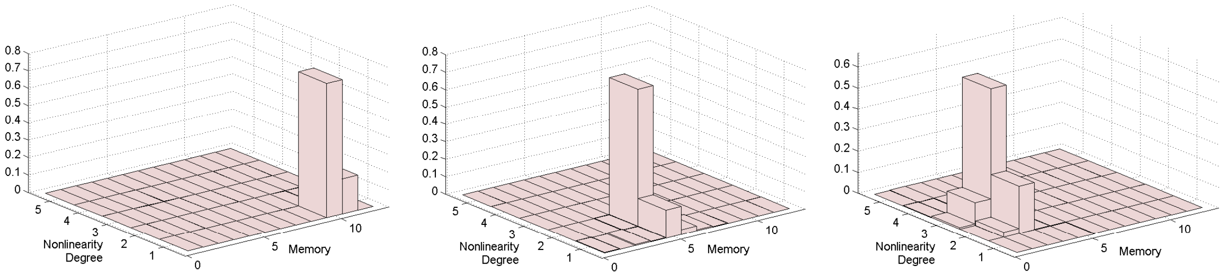
Table 4 shows the model selection performance of RJMCMC and reference methods AIC and BIC after 100 simulations for 100 different data sets from 3 different Volterra models. In each RJMCMC realization the most visited model after burn-in period is taken as the detected model. Examining the correctly detected model order percentages in the Table 4, AIC always falls short of selecting true model order pair as compared to that of RJMCMC and BIC. RJMCMC and BIC achieve generally the same percentages, however, when the model is nonlinear (V(2,5) and V(3,3)), RJMCMC performs better. For case 4, performance of RJMCMC is superior for nonlinear models and its percentage of detection is at least %89, however BIC achieves at most %13 for the same models.

Figure 4 shows the joint posterior density of the model orders, p and q for the simulated models and randomly selected cases in a single example realization. It has been stated that RJMCMC estimates true model order higher than %50 for each example realizations.

Table 4: Percentage of detecting correct model orders

		Case 1	Case 2	Case 3	Case 4
V(1,10)	RJMCMC	**100%**	**100%**	**100%**	**100%**
	AIC	99%	84%	89%	76%
	BIC	**100%**	**100%**	**100%**	**100%**
V(2,5)	RJMCMC	**100%**	99%	**100%**	**93%**
	AIC	93%	68%	85%	0%
	BIC	99%	**100%**	**100%**	11%
V(3,3)	RJMCMC	**100%**	**100%**	**100%**	**89%**
	AIC	98%	83%	93%	0%
	BIC	99%	**100%**	**100%**	13%

Next we compare the success of RJMCMC in estimating model coefficients with NLS estimate which is obtained via the augmented data matrix $\mathbf{X}$. NLS has been given the correct model orders p, and q and performs estimation for model coefficients as:



(a) V(1,10) - Case 2 (b) V(2,5) - Case 3 (c) V(3,3) - Case 4

Figure 4: The joint posterior density of the model orders of (a) - V(1,10), (b) - V(2,5), (c) - V(3,3).

$$\hat{\mathbf{h}}_{\mathrm{NLS}} = (\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{y}, \tag{49}$$

where vector $\mathbf{y}$ is output data and $\mathbf{X}$ is the data matrix which has the form defined in (4).

The performance comparison study has been made on the model coefficient estimation of RJMCMC and NLS methods in terms of error measure, NMSE, which can be defined by:

$$\mathrm{NMSE} = \frac{1}{\eta}\sum_{i=1}^{\eta}\frac{(h_i - \widehat{h_i})^2}{||\mathbf{h}||_2^2}, \tag{50}$$

where $\mathbf{h}$ is the $\eta-$vector of model coefficients, $\widehat{\mathbf{h}}$ is its estimate and $||\mathbf{h}||_2$ is the l_2-norm of $\mathbf{h}$.

Model coefficient estimation performance for all three models and all four cases are shown in Table 5. Examining NMSE values in Table 5 shows that NLS estimation achieves lower error values than RJMCMC for all the cases. Notwithstanding, RJMCMC shows very close performance to the NLS method. Note that the NMSE figures of NLS are hypothetical since they are based on unavailable perfect model order estimates. Consequently, model coefficient estimation performance of RJMCMC appears remarkable because it estimates model orders and coefficients at the same time.

Table 5: Performance comparison of model coefficient estimation in terms of NMSE

		Case 1	Case 2	Case 3	Case 4
V(1,10)	RJMCMC	5.89E-07	2.36E-06	2.47E-06	1.43E-03
	Informed NLS	2.42E-09	8.46E-07	7.86E-07	1.26E-03
V(2,5)	RJMCMC	6.76E-08	2.06E-05	1.12E-07	1.42E-03
	Informed NLS	8.42E-09	1.93E-05	7.73E-08	1.32E-03
V(3,3)	RJMCMC	1.69E-04	1.84E-04	1.74E-04	6.07E-03
	Informed NLS	6.76E-08	2.28E-07	3.90E-08	3.46E-03

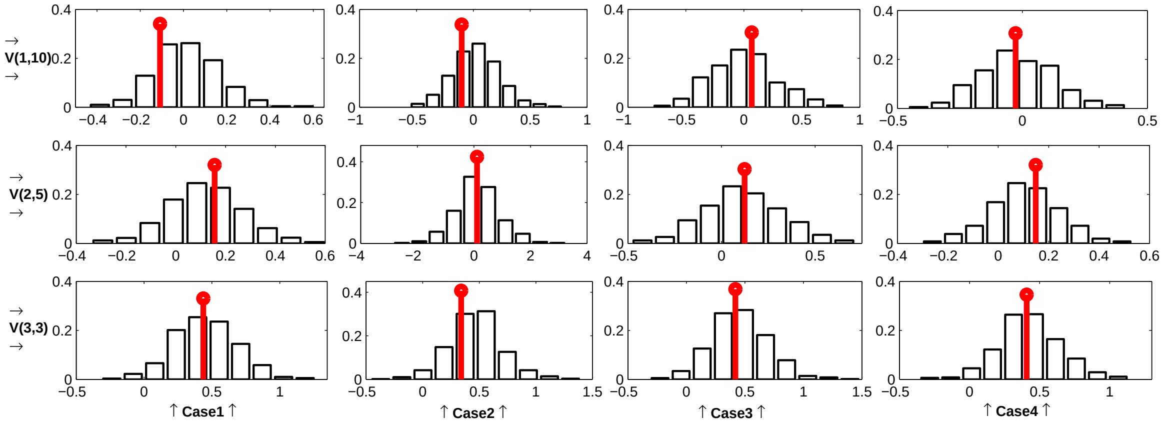


Figure 5: Estimated output histograms for all cases and all models in simulation 1 via RJMCMC. Real data mean values are plotted with a red vertical line. Each row shows the results for simulated models and each column shows the results for simulation cases.

Figure 5 shows estimated output data histogram for each of the twelve synthetically generated Volterra model data. Observing the subplots in Figure 5 depicts that real data means stand in the high probability ranges of estimated data distributions and this reveals the good model estimation performance of the proposed method.

Table 6: RJMCMC Computational Gain.

	Case 1		Case 2		Case 3		Case 4	
	Total*	Avg.**	Total*	Avg.**	Total*	Avg.**	Total*	Avg.**
V(1,10)	16	12.37	18	12.65	16	12.31	17	12.51
V(2,5)	20	10.22	15	9.08	20	10.98	20	13.3
V(3,3)	18	8.11	20	8.06	19	8.5	26	9.79

Each RJMCMC run has performed 30000 iterations, and number of visited Volterra models has been recorded for each run.
* *Numbers at **Total** cells represent the total number of distinct Volterra models visited after 100 RJMCMC runs.*
** *Numbers at **Avg.** cells refer to the average number of Volterra models visited at a single run after 100 RJMCMC runs.*
Model space includes 60 Volterra models.

As stated in the previous sections, RJMCMC is a learning algorithm which avoids performing exhaustive searches, instead performs a model search by using the likelihood, the priors and the data to visit only plausible models. In Table 6, calculations on computational gain of RJMCMC for simulation 1 has been depicted. Examining ”Total” columns shows that higher than 50% of the candidate models (in all the cases these are wrong models) have not been visited and RJMCMC decides ”true model” only visiting a small subset of the model space. Analysing the ”Avg.” columns shows that the search subset is smaller than the total amount and we can state that RJMCMC decides ”true model” by examining at most only one fifth of the model space (at most 12-13 models over 60 possible models). Thus, this exhibits the computational gains of RJMCMC compared with the other model selection methods AIC, BIC or than the sampling algorithms Nested sampling, TMCMC, etc. where all perform exhaustive searchs on model space.

6.2. Simulation 2: Nonlinear Channel Estimation

In communication systems, due to high-power amplifiers at the transmitter side and filtering operations at the receiver side, nonlinear input-output characteristics are frequently observed. Most of these nonlinearities can be approximated via Volterra series. A nonlinear communication channel is expressed in terms of discrete time baseband Volterra model with symmetric coefficients as [16, 54]:

$$y(l) = \sum_{\nu=1}^{\frac{p+1}{2}} \sum_{m_1=1}^{q} \cdots \sum_{m_{2\nu-1}=m_{2\nu-2}}^{q} h^{(2\nu-1)}_{m_1,\dots,m_{2\nu-1}} \prod_{i=1}^{\nu} x(l-m_i) \prod_{j=\nu+1}^{2\nu-1} x^*(l-m_j). \tag{51}$$

where $x(l)$ and $y(l)$ represent the complex input and output envelopes of the system, p is the nonlinearity degree and q is the memory of the channel. The $(2\nu-1)$st-order Volterra coefficient is referred to as $h^{(2\nu-1)}_{m_1,\dots,m_{2\nu-1}}$. Moreover, it has been stated in [55] that powers of even-ordered terms do not contribute to the output. Thus, only odd-ordered terms ($p = 1, 3, \dots$) are taken into account for baseband Volterra representation in (51).

Many modern communication systems such as *asymmetric digital subscriber line* (ADSL) modems, digital video broadcasting and recent mobile communication systems in 4G, utilize OFDM technique. However, due to its high peak-to-average power ratio, OFDM is very vulnerable to nonlinearities [54]. For these reasons, an OFDM communication system which transmits through a nonlinear communication channel has been implemented. The proposed Bayesian VSI model has been employed to estimate this nonlinear channel in terms of Volterra series.

We assume that a baseband Volterra model in (51) represents the unknown nonlinear communication channel with nonlinearity degree of 3 and memory of 2. Uniformly distributed message bits have been modulated via either QPSK or 16-QAM. Modulated symbols have been sent through an OFDM system with 512 sub-carriers. Resulting symbols have been parallel-to-serial converted and transmitted through the nonlinear channel. After adding white Gaussian noise, the transmitted corrupted signal has been received at receiver.

Pilot messages have been employed in order to apply a VSI procedure. Hence, both pilot OFDM output and the corrupted received signal are known at the receiver as input and output of the unknown system, respectively. RJMCMC uses these I/O signals to identify the unknown nonlinear channel. Consequently, proposed method estimates the nonlinearity degree, the system memory and the corresponding channel coefficients. Initial values have been selected as $\alpha_e = 1$, $\beta_e = 1$, $\alpha_h = 50$ and $\beta_h = 2$. The initial system orders are $q_0 = 1$ and $p_0 = 1$. The upper bounds are $q_{max} = 12$ and $p_{max} = 5$. RJMCMC takes all the model orders into account between $p = 1$ and $p = 5$ whether it is odd or even and decides the true Volterra model for the nonlinear channel. For additive noise processes, signal-to-noise ratio (SNR) values between -5 and 25 have been used in order to measure performance of the proposed method

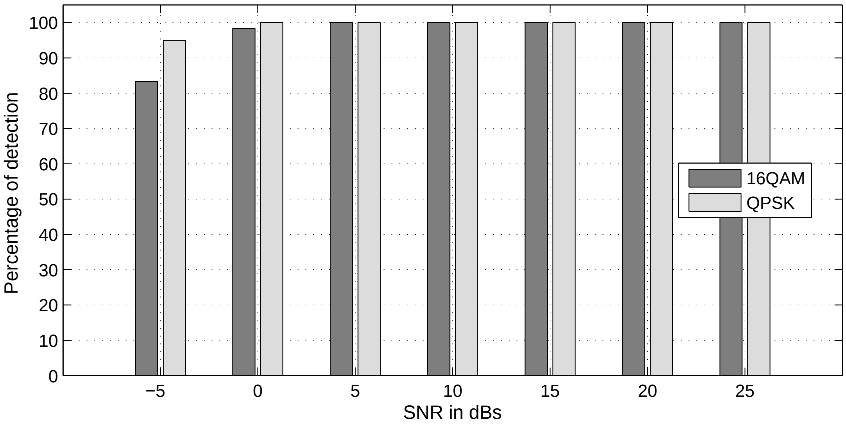


Figure 6: Percentage of correctly estimated model order via RJMCMC for varying SNR. (Nonlinear channel, V(3,2))

under different noisy conditions. Simulations have been repeated for 60 Monte Carlo runs and results are presented as average of these repetitions in order to remove random realization effects. Simulated channel coefficients have been selected as $h_1 = [0.5, \quad 0.3]$ and $h_3 = [-0.7, \quad -0.2, \quad 0.34, \quad -0.27]$ for linear and cubic terms of the baseband Volterra model in (51), respectively.

Percentage of correctly estimated model orders for varying SNR values are shown in Figure 6. Examining Figure 6 it can be clearly stated that, RJMCMC correctly estimates the true nonlinear channel, V(3,2), with full performance by obtaining %100 after 60 RJMCMC runs for both modulations at SNR values higher than 0 dB. Below 5 dB, true channel is correctly estimated at least %80 times of the repetitions.

Figure 7 depicts the NMSE values in logarithmic scale between estimated channel coefficients and true coefficients. Examining Figure 7 shows that at an SNR value of 25 dB RJMCMC estimates the channel with NMSE values of 10^{-4} and 10^{-3} for QPSK and 16-QAM, respectively. For lower SNR values, estimation performances are lower as expected and NMSE values are around 10^{-2} for SNR of 0 dB.

7. Conclusion

In this study, we propose a new perspective on RJMCMC which defines trans-structural transitions between different classes of models such as linear and nonlinear, etc. This provides using RJMCMC in model identification problems the model space of which includes models with dimensions which may differ or not.

Furthermore, this methodology has been employed as a part of a compact VSI method by estimating the nonlinearity degree as well as the system memory and model coefficients. Using the proposed method in VSI problems is advantageous especially when the systems to be identified have varying degrees of nonlinearities and when estimating the nonlinearity degree is crucial. RJMCMC shows remarkable performance on nonlinear channel estimation in an

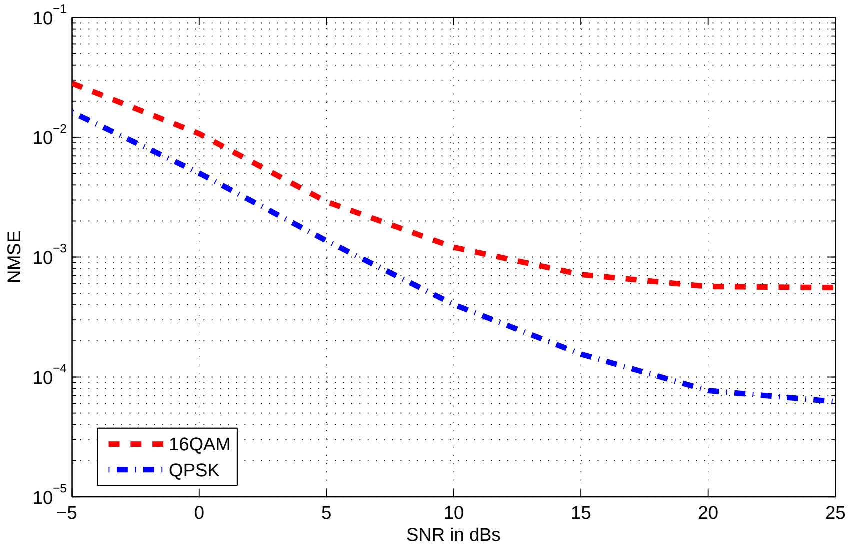


Figure 7: NMSE values for channel coefficients estimation of RJMCMC for QPSK and 16-QAM modulation schemes.

OFDM communication system. Performance results for both QPSK and 16-QAM, are satisfactory for both channel model selection and coefficients estimation studies.

These results demonstrate the potentials of RJMCMC in identifying nonlinear systems and nonlinear communication channels in terms of Volterra models.

References

[1] E. E. Kuruoğlu, Nonlinear least l_p-norm filters for nonlinear autoregressive α-stable processes, Digital Signal Processing 12 (1) (2002) 119–142.

[2] S.-Y. Kang, Volterra type integral equation method for the radial Schrödinger equation: Single channel case, Computers & Mathematics with Applications 48 (10) (2004) 1425–1440.

[3] J.-M. Le Caillec, Spectral inversion of second order Volterra models based on the blind identification of Wiener models, Signal Processing 91 (11) (2011) 2541–2555.

[4] W. Ji, W.-S. Gan, Identification of a parametric loudspeaker system using an adaptive Volterra filter, Applied Acoustics 73 (12) (2012) 1251–1262.

[5] C. Contan, B. S. Kirei, et al., Modified NLMF adaptation of Volterra filters used for nonlinear acoustic echo cancellation, Signal Processing 93 (5) (2013) 1152–1161.

[6] N. Kalouptsidis, G. Mileounis, B. Babadi, V. Tarokh, Adaptive algorithms for sparse system identification, Signal Processing 91 (8) (2011) 1910–1919.

[7] S. Mhatli, B. Nsiri, M. A. Jarajreh, M. Channoufi, R. Attia, Adaptive Volterra equalizer for optical OFDM modem, in: Photonics Prague 2014, International Society for Optics and Photonics, 2015, pp. 94500R–94500R.

[8] G. Mileounis, N. Kalouptsidis, Blind identification of second order Volterra systems with complex random inputs using higher order cumulants, Signal Processing, IEEE Transactions on 57 (10) (2009) 4129–4135.

[9] D. Kotoulas, P. Koukoulas, G.-O. Glentis, Subspace based blind identification of LTI FIR MIMO systems and equalization of finite memory SIMO Volterra systems, Signal Processing 91 (8) (2011) 1941–1950.

[10] P. J. Green, Reversible jump Markov chain Monte Carlo computation and Bayesian model determination, Biometrika 82 (4) (1995) 711–732. doi:10.1093/biomet/82.4.711.

[11] O. Karakus, E. E. Kuruoglu, M. A. Altinkaya, Estimation of the nonlinearity degree for polynomial autoregressive processes with RJMCMC, in: 23rd European Signal Processing Conference (EUSIPCO), IEEE, 2015, pp. 953–957.

[12] O. Karakuş, E. E. Kuruoğlu, M. A. Altınkaya, Bayesian estimation of polynomial moving average models with unknown degree of nonlinearity, in: 24th European Signal Processing Conference (EUSIPCO), IEEE, 2016, pp. 1543–1547.

[13] P. Alper, A consideration of the discrete Volterra series, IEEE Transactions on Automatic Control 10 (3) (1965) 322–327.

[14] P. L. Green, K. Worden, Bayesian and Markov chain Monte Carlo methods for identifying nonlinear systems in the presence of uncertainty, Phil. Trans. R. Soc. A 373 (2051) (2015) 20140405.

[15] K. Shi, P. Shi, Adaptive sparse Volterra system identification with l_0-norm penalty, Signal Processing 91 (10) (2011) 2432–2436.

[16] T. Bouilloc, G. Favier, Nonlinear channel modeling and identification using baseband Volterra–Parafac models, Signal Processing 92 (6) (2012) 1492–1498.

[17] G.-O. Glentis, K. Georgoulakis, K. Angellopoulos, Adaptive channel identification in optical fiber communication systems, in: Transparent Optical Networks (ICTON), 2014 16th International Conference on, IEEE, 2014, pp. 1–4.

[18] L. Lu, H. Zhao, B. Chen, Improved-variable-forgetting-factor recursive algorithm based on the logarithmic cost for volterra system identification, IEEE Transactions on Circuits and Systems II: Express Briefs 63 (6) (2016) 588–592.

[19] N. I. Chaudhary, M. A. Z. Raja, M. S. Aslam, N. Ahmed, Novel generalization of volterra lms algorithm to fractional order with application to system identification, Neural Computing and Applications (2016) 1–18.

[20] R. Claser, V. H. Nascimento, Y. V. Zakharov, A low-complexity rls-dcd algorithm for volterra system identification, in: Signal Processing Conference (EUSIPCO), 2016 24th European, IEEE, 2016, pp. 6–10.

[21] K. Batselier, Z. Chen, N. Wong, A tensor network kalman filter with an application in recursive mimo volterra system identification, arXiv preprint arXiv:1610.05434.

[22] H. M. Abbas, M. M. Bayoumi, Volterra-system identification using adaptive real-coded genetic algorithm, Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 36 (4) (2006) 671–684.

[23] H. Merabti, D. Massicotte, Nonlinear adaptive channel equalization using genetic algorithms, in: New Circuits and Systems Conference (NEWCAS), 2014 IEEE 12th International, IEEE, 2014, pp. 209–212.

[24] M. Shoaib, S. Werner, J. A. Apolinário, Multichannel fast QR-decomposition algorithms: Weight extraction method and its applications, Signal Processing, IEEE Transactions on 58 (1) (2010) 175–188.

[25] H. Loussifi, K. Nouri, N. B. Braiek, A new efficient hybrid intelligent method for nonlinear dynamical systems identification: The Wavelet Kernel Fuzzy Neural Network, Communications in Nonlinear Science and Numerical Simulation 32 (2016) 10–30.

[26] L. Fortuna, A. Rizzo, M. Sinatra, M. Xibilia, Soft analyzers for a sulfur recovery unit, Control Engineering Practice 11 (12) (2003) 1491–1500.

[27] J. L. Beck, Bayesian system identification based on probability logic, Structural Control and Health Monitoring 17 (7) (2010) 825–847.

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

[29] P. L. Green, Bayesian system identification of a nonlinear dynamical system using a novel variant of simulated annealing, Mechanical Systems and Signal Processing 52 (2015) 133–146.

[30] P. T. Troughton, S. J. Godsill, A reversible jump sampler for autoregressive time series, in: Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on, Vol. 4, IEEE, 1998, pp. 2257–2260.

[31] R. S. Ehlers, S. P. Brooks, Bayesian analysis of order uncertainty in ARIMA models, Federal University of Parana, Tech. Rep.

[32] E. Eğri, et al., Bayesian model selection in ARFIMA models, Expert Systems with Applications 37 (12) (2010) 8359–8364.

[33] S. Richardson, P. J. Green, On Bayesian analysis of mixtures with an unknown number of components (with discussion), Journal of the Royal Statistical Society: series B (statistical methodology) 59 (4) (1997) 731–792.

[34] V. Viallefont, S. Richardson, P. J. Green, Bayesian analysis of Poisson mixtures, Journal of nonparametric statistics 14 (1-2) (2002) 181–202.

[35] D. Salas-Gonzalez, E. E. Kuruoglu, D. P. Ruiz, Modelling with mixture of symmetric stable distributions using Gibbs sampling, Signal Processing 90 (3) (2010) 774–783.

[36] Q. Xia, J. Pan, Z. Zhang, J. Liu, A Bayesian nonlinearity test for threshold moving average models, Journal of Time Series Analysis 31 (5) (2010) 329–336.

[37] P. T. Troughton, S. J. Godsill, MCMC methods for restoration of nonlinearly distorted autoregressive signals, Signal Processing 81 (1) (2001) 83–97.

[38] W. Hastings, Monte Carlo samping methods using Markov chains and their applications, Biometrika 57 (1970) 97–109. `doi:10.1093/biomet/57.1.97`.

[39] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Rubin, Bayesian Data Analysis, Second Edition (Chapman & Hall/CRC Texts in Statistical Science), 2nd Edition, Chapman and Hall/CRC, 2003.

[40] A. Gelman, D. B. Rubin, Inference from iterative simulation using multiple sequences, Statistical science (1992) 457–472.

[41] A. Gelman, D. B. Rubin, A single series from the Gibbs sampler provides a false sense of security, Bayesian statistics 4 (1992) 625–631.

[42] S. Brooks, P. Giudici, Markov chain Monte Carlo convergence assessment via two-way analysis of variance, Journal of Computational and Graphical Statistics 9 (2) (2000) 266–285.

[43] J. S. Rosenthal, Minorization conditions and convergence rates for Markov chain Monte Carlo, Journal of the American Statistical Association 90 (430) (1995) 558–566.

[44] J. M. Castelloe, D. L. Zimmerman, Convergence assessment for reversible jump mcmc samplers, Department of Statistics and Actuarial Science, University of Iowa, Technical Report 313.

[45] S. A. Sisson, Y. Fan, A distance-based diagnostic for trans-dimensional markov chains, Statistics and Computing 17 (4) (2007) 357–367.

[46] A. Belloni, V. Chernozhukov, On the computational complexity of mcmc-based estimators in large samples, The Annals of Statistics (2009) 2011–2055.

[47] N. B. Haaser, J. A. Sullivan, Real analysis, Courier Corporation, 1991.

[48] M. A. Ivanov, Distribution of the output reponse of Volterra-type narrowband nonlinear-systems, Telecommunications and Radio Engineering 41 (12) (1987) 99–102.

[49] P. Hadjidoukas, P. Angelikopoulos, D. Rossinelli, D. Alexeev, C. Papadimitriou, P. Koumoutsakos, Bayesian uncertainty quantification and propagation for discrete element simulations of granular materials, Computer Methods in Applied Mechanics and Engineering 282 (2014) 218–238.

[50] E. Simoen, C. Papadimitriou, G. Lombaert, On prediction error correlation in Bayesian model updating, Journal of Sound and Vibration 332 (18) (2013) 4136–4152.

[51] J.-A. Goulet, I. F. Smith, Structural identification with systematic errors and unknown uncertainty dependencies, Computers & structures 128 (2013) 251–258.

[52] P. J. Green, S. Richardson, Modelling heterogeneity with and without the Dirichlet process, Scandinavian journal of statistics (2001) 355–375.

[53] A. Meyer-Gohde, et al., Generalized exogenous processes in DSGE: A Bayesian approach, Tech. rep., Technical report, Humboldt University (2014).

[54] G. Mileounis, P. Koukoulas, N. Kalouptsidis, Input–output identification of nonlinear channels using PSK, QAM and OFDM inputs, Signal Processing 89 (7) (2009) 1359–1369.

[55] S. Benedetto, E. Biglieri, Principles of digital transmission: with wireless applications, Springer Science & Business Media, 1999.