White Rose Research Online

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk
Universities of Leeds, Sheffield and York

Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/128829/

Version: Accepted Version

Article:
Jacobs, W.R., Baldacchino, T., Dodd, T.J. et al. (2018) Sparse Bayesian Nonlinear System Identification using Variational Inference. IEEE Transactions on Automatic Control, 63 (12). pp. 4172-4187. ISSN: 0018-9286

https://doi.org/10.1109/TAC.2018.2813004

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

Sparse Bayesian Nonlinear System Identification using Variational Inference

William R. Jacobs, Tara Baldacchino, Tony Dodd, and Sean R. Anderson

***Abstract*—Bayesian nonlinear system identification for one of the major classes of dynamic model, the nonlinear autoregressive with exogenous input (NARX) model, has not been widely studied to date. Markov chain Monte Carlo (MCMC) methods have been developed, which tend to be accurate but can also be slow to converge. In this contribution, we present a novel, computationally efficient solution to sparse Bayesian identification of the NARX model using variational inference, which is orders of magnitude faster than MCMC methods. A sparsity-inducing hyper-prior is used to solve the structure detection problem. Key results include: 1. successful demonstration of the method on low signal-to-noise ratio signals (down to 2dB); 2. successful benchmarking in terms of speed and accuracy against a number of other algorithms: Bayesian LASSO, reversible jump MCMC, forward regression orthogonalisation, LASSO and simulation error minimisation with pruning; 3. accurate identification of a real world system, an electroactive polymer; and 4. demonstration for the first time of numerically propagating the estimated nonlinear time-domain model parameter uncertainty into the frequency-domain.**

***Keywords*—Bayesian estimation, variational inference, system identification, NARX model.**

I. Introduction

In nonlinear system identification, a popular model class is the nonlinear autoregressive with exogenous inputs (NARX) model [1], [2]. Reasons for this popularity include the compactness of the representation, compared to e.g. Volterra series [3], relative simplicity of estimating parameters due to the linear-in-the-parameters structure [4], and the fact that frequency-domain analysis methods have been developed for the model class, facilitating analysis of nonlinear dynamics [5], [6]. The NARX model is a black-box model, meaning that terms are unknown and must be selected - typically regarded as one of the most challenging problems in nonlinear system identification [7], which is the focus of this paper, in a Bayesian context.

Least-squares and maximum likelihood-type methods have dominated NARX model identification, e.g. by forward regression [4], [8], forward-backward pruning methods [9], [10], the expectation-maximisation (EM) algorithm [11] and sparse estimation [12], [13]. Bayesian model identification, on the other hand, is advantageous for a number of reasons: (i) parameter uncertainty is intrinsically described, useful for analysis, simulation and control design [14]; (ii) overfitting is avoided by natural penalisation of overly complex models [15]; (iii) model uncertainty can be accurately quantified even for data records with relatively few samples [16]; and (iv) prior information can be incorporated where available [17].

There are a range of methods addressing Bayesian nonlinear system identification, which can be divided by model class, e.g. state-space [18], white-box [19], [20], mixture of experts [21], continuous-time [22], Wiener [23] and NARX models [24]–[26]. Bayesian methods for NARX modelling include nonparametric approaches based on Gaussian process regression [24], [25] and sampling methods based on reversible jump Markov chain Monte Carlo (MCMC) [26]. Although MCMC methods tend to be accurate, they are also computationally intensive and can be prohibitively slow, because they tend to rely on large numbers of samples, and may exhibit slow convergence to a stationary distribution [27].

There is a need, therefore, for a computationally efficient algorithm for Bayesian NARX model identification. To address this challenge, we develop a novel method based on variational Bayesian (VB) inference [28], [29]. The algorithm is both simple to implement and computationally efficient, as it is based on closed form updates of relatively low computational complexity. Model term selection is performed using a sparsity inducing hyper-prior, based on a method known as automatic relevance determination (ARD), which is used to iteratively prune redundant terms from the model [15].

Sparse Bayesian learning algorithms using ARD have been developed for solving regression and classification problems with kernel methods [30]–[32], and have inspired alternatives, e.g. applied to signal denoising [33] and pattern recognition problems with correlated errors [34]. However, ARD, as proposed by [31], is not sufficient to perform the term selection problem in a single step for non-trivial problems. A key novelty here is a new algorithm for term selection of NARX models using sparse variational Bayes (SVB) with ARD, where ARD is iteratively applied to a reducing subset of model terms, retaining the subset identified at each iteration, until there is a single term left in the model.

There are a number of advantages to this SVB-NARX modelling algorithm. Firstly, the SVB-NARX algorithm is fully Bayesian, meaning that as in [31], nuisance parameters such as the noise variance are automatically estimated. This contrasts to e.g. [18], where the noise variance in a Bayesian LASSO-type algorithm is assumed known (although could be estimated, e.g. via an EM-algorithm approach [31]). A further advantage of SVB-NARX is that a metric is intrinsically generated (the variational lower bound) that can be used for automated model selection, which trades-off accuracy and model complexity. The final main advantage is that SVB-NARX is a fast algorithm because it can prune terms in batches - as we show in the results, it is much faster when selecting from large term sets

W. Jacobs, T. Baldacchino, T. Dodd, and S. R. Anderson are with the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, S1 3JD UK, e-mail: w.jacobs@sheffield.ac.uk

than forward regression orthogonalisation (FRO) [4], Bayesian LASSO [18], and a Bayesian algorithm based on reversible jump MCMC [26].

Elements of this work have been published in brief conference format [35], and have been extended here in a number of important ways: a more thorough and rigorous theoretical development, expanded benchmarking, and application to real-world data. The real world data is obtained from a dielectric elastomer actuator (DEA), which is a type of actuator used in soft robotics [36]–[38]. Furthermore, we demonstrate, for the first time, frequency-domain uncertainty analysis for nonlinear systems by propagation of parameter uncertainty from the identified Bayesian model of the DEA into the frequency-domain, using Monte Carlo sampling and nonlinear output frequency response functions (NOFRFs) [6].

The paper is organized as follows. Section II defines the Bayesian NARX model with associated priors. Section III gives background on variational Bayesian inference. In Section IV-A the parameter estimation algorithm for NARX models is derived using variational inference. In Section V the structure detection algorithm for NARX models is derived using variational inference combined with ARD. In Section VI numerical examples are given, along with benchmark analysis, and finally an application to a real-world system (a set of DEAs) using experimental data. The paper is summarised in Section VII.

II. System identification within a Bayesian Framework

A. The NARX model

In this section Bayesian inference of linear-in-the-parameters regression models is introduced in the context of the NARX model. Consider a single-input single-output dynamic system as some non-linear function, $f(.)$, of lagged system inputs, u_k, and outputs, y_k, at sample time k,

$$y_k = f(\mathbf{x}_k) + e_k \tag{1}$$

where $\mathbf{x}_k = \left(y_{k-1}, ..., y_{k-n_y}, u_{k-1}, ..., u_{k-n_u}\right)$ and e_k is a zero-mean Normally distributed white noise process with variance τ^{-1}. n_u and n_y are the maximum lags, or dynamic orders, of the input and output respectively. The non-linear function $f(.)$ can be decomposed into a sum of M weighted basis functions, e.g. by polynomials, radial basis functions, wavelets etc., which is a linear-in-the-parameters model,

$$\begin{aligned} f(\mathbf{x}_k) &= \sum_{m=1}^{M} \theta_m \phi_m(\mathbf{x}_k) && (2)\\ &= \Phi_k \boldsymbol{\theta} && (3) \end{aligned}$$

where

$$\begin{aligned} \Phi_k &= [\phi_1(\mathbf{x}_k), \phi_2(\mathbf{x}_k), \ldots, \phi_M(\mathbf{x}_k)], \quad \Phi_k \in \mathbb{R}^{1\times M} \\ \boldsymbol{\theta} &= [\theta_1, \theta_2, \ldots, \theta_M]^T, \quad \boldsymbol{\theta} \in \mathbb{R}^{M\times 1} \end{aligned}$$

and Φ_k is the k'th row of the matrix $\boldsymbol{\Phi}$ such that

$$\boldsymbol{\Phi} = [\Phi_1^T, \Phi_2^T, \ldots, \Phi_N^T]^T, \quad \boldsymbol{\Phi} \in \mathbb{R}^{N\times M}$$

where N is the number of data points. The term selection problem, addressed in this paper, is the selection of the best choice of basis functions, ϕ_m, such that $f(\cdot)$ provides a parsimonious description of the true system behaviour.

B. Bayesian parameter estimation

Bayes' rule allows us to infer the posterior distribution of the set of model parameters, denoted $\boldsymbol{\Theta}$, given the observed data set $\mathbf{y} = [y_1, \ldots, y_N]^T$, then

$$p(\boldsymbol{\Theta}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\Phi}, \boldsymbol{\Theta})p(\boldsymbol{\Theta})}{p(\mathbf{y})}. \tag{4}$$

In the above equation the term $p(\mathbf{y}|\boldsymbol{\Phi}, \boldsymbol{\Theta})$ is referred to as the likelihood function and $p(\boldsymbol{\Theta})$ is the prior distribution of the parameters before observing the data. The denominator, $p(\mathbf{y})$ is named the marginal likelihood and is given as

$$p(\mathbf{y}) = \int p(\mathbf{y}|\boldsymbol{\Phi}, \boldsymbol{\Theta})p(\boldsymbol{\Theta})d\boldsymbol{\Theta}. \tag{5}$$

in order to normalise the posterior distribution.

The evaluation of (5) can be extremely challenging and necessitates the need for approximation methods in all but the most simple of cases [27]. In this paper (5) will be approximated by the use of variational Bayesian techniques.

C. Likelihood function

The likelihood function for the NARX model given in (1) for $\mathbf{y}$ under the assumption of a Normal *i.i.d.* noise sequence is

$$\begin{aligned} p(\mathbf{y}|\boldsymbol{\Phi}, \boldsymbol{\theta}, \tau) &= \prod_{k=1}^{N} p(y_k|\Phi_k, \boldsymbol{\theta}, \tau) && (6)\\ &= \prod_{k=1}^{N} \mathcal{N}(y_k|\Phi_k\boldsymbol{\theta}, \tau^{-1}) && (7)\\ &= \left(\frac{\tau}{2\pi}\right)^{\frac{N}{2}} \exp\left(-\frac{\tau}{2}\sum_{k=1}^{N}(y_k - \Phi_k\boldsymbol{\theta})^2\right) && (8) \end{aligned}$$

where $\mathcal{N}(x|\mu, \sigma)$ denotes the Normal distribution of the variable x with mean μ and variance σ.

D. Priors

The likelihood function given by (8) is a member of the exponential family and so the choice of an exponential prior is required for conjugacy [39].

The mean, $\Phi_k\boldsymbol{\theta}$, and precision, τ, of the likelihood are unknown parameters to be inferred and as such the choice of Normal-Gamma prior distribution is made [40] such that

$$\begin{aligned} p(\boldsymbol{\theta}, \tau|\boldsymbol{\alpha}) &= p(\boldsymbol{\theta}|\tau, \boldsymbol{\alpha})p(\tau) && (9)\\ &= \mathcal{N}(\boldsymbol{\theta}|0, (\tau\mathbf{A})^{-1})\text{Gam}(\tau|a_0, b_0) && (10)\\ &= 2\pi^{-M/2}|\mathbf{A}|^{1/2}\frac{b_0^{a_0}}{\Gamma(a_0)}\tau^{M/2+a_0-1} \\ &\quad \exp\left(-\frac{\tau}{2}(\boldsymbol{\theta}^T\mathbf{A}\boldsymbol{\theta} + 2b_0)\right) && (11) \end{aligned}$$

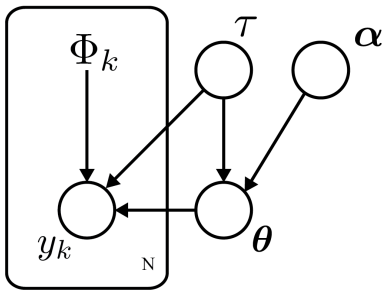

Fig. 1. Probabilistic graphical model of the hierarchical model represented in (14). The plate (box), denoted by the number of data samples N, indicates N *i.i.d* observations. The arrows indicate the direction of conditional dependence.

where the Normal distribution has been further parametrised by $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_M)^T = \text{diag}(\mathbf{A})$. The introduction of $\boldsymbol{\alpha}$ into the model naturally incorporates ARD, this is the basis of the sparse estimation framework that will be discussed later. The variable $\boldsymbol{\alpha}$ is an unknown model parameter and will also be inferred and so requires the introduction of a hyper-prior (prior of a prior), $p(\boldsymbol{\alpha})$. The choice of a conjugate prior is again chosen in order to simplify the later analysis. The hyper-prior is therefore assigned as independent Gamma distributions,

$$p(\boldsymbol{\alpha}) = \prod_{m=1}^{M} \text{Gam}(\alpha_m | c_0, d_0) \tag{12}$$

$$= \prod_{m=1}^{M} \frac{d_0^{c_0}}{\Gamma(c_0)} \alpha_m^{c_0 - 1} \exp(-d_0 \alpha_m) \tag{13}$$

The model parameters $\{\boldsymbol{\theta}, \tau, \boldsymbol{\alpha}\}$ along with hyper-parameters $\{a_0, b_0, c_0, d_0\}$ can be initialised to have broad/uninformative prior distributions so that the inference process is dominated by the influence of the data [17].

The joint distribution over all of the random variables can now be expressed hierarchically as

$$p(\mathbf{y}, \boldsymbol{\Phi}, \boldsymbol{\theta}, \tau, \boldsymbol{\alpha}) = p(\mathbf{y}|\boldsymbol{\Phi}, \boldsymbol{\theta}, \tau) p(\boldsymbol{\theta}|\tau, \boldsymbol{\alpha}) p(\tau) p(\boldsymbol{\alpha}), \tag{14}$$

assuming τ and $\boldsymbol{\alpha}$ independent and noting that $\boldsymbol{\Phi}$ is a function of the observed data and not a random variable. The decomposition can by made more transparent be considering the directed graphical model shown in Figure 1.

E. Prior for Automatic Relevance Determination

ARD has been incorporated into the Bayesian model via the introduction of the hyper-parameter $\boldsymbol{\alpha}$ in (9), where α_m corresponds to the precision (inverse of the variance) of θ_m. α_m therefore controls the magnitude of θ_m, if $\alpha_m^{-1} = 0$ then the precision of θ_m is infinite and in order to maintain a high likelihood $\theta_m = 0$, indicating that the m'th model term is not relevant to the generation of the data. α_m is hence acting as a sparse regularisation term that acts independently on each model weight θ_m. The values of α_m can then be used as a basis for pruning irrelevant basis functions from the model [41], [42].

In the remainder of this paper the value, α_m^{-1}, is named as the m'th ARD value, where small values indicate terms that are not relevant to the generation of the output. This value will be used to drive the structure detection in the SVB-NARX system identification algorithm introduced in Section V.

F. Posterior distribution

The joint posterior distribution over the model parameters can be found by considering (4), with $\boldsymbol{\Theta} = \{\boldsymbol{\theta}, \tau, \boldsymbol{\alpha}\}$, where

$$p(\boldsymbol{\theta}, \tau, \boldsymbol{\alpha}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\Phi}, \boldsymbol{\theta}, \tau) p(\boldsymbol{\theta}|\tau, \boldsymbol{\alpha}) p(\tau) p(\boldsymbol{\alpha})}{p(\mathbf{y})}. \tag{15}$$

The inclusion of the hyper-parameter, $\boldsymbol{\alpha}$, into the model causes the marginal likelihood in the denominator of (15) to become intractable, *i.e.* no direct analytical solution is possible. Many methods exist for approximating the marginal likelihood [43], commonly these techniques are based on random sampling [27]. Here, variational Bayesian inference will be used because the posterior distribution can be approximated in a series of closed form update equations, avoiding the use of computationally expensive sampling methods.

III. Variational Bayesian inference

In many non-trivial cases the evaluation of posterior distributions is infeasible as is the case with the linear regression with ARD introduced in the previous section. The full Bayesian treatment in closed form is only feasible for a limited class of models [39]. Numerical integration techniques can always be used, however, the computational expense is often prohibitive. Variational Bayes provides a method for approximating the posterior distribution. In this section some general results of variational Bayesian inference are discussed followed by its application to the linear regression model given by (2).

A. Variational optimisation of the Bayesian model

In the context of Bayesian inference problems variational calculus can be used as a method for approximating posterior distributions. Consider a fully Bayesian model such that all the model parameters are assigned prior distributions as described in Section II. The aim of the inference problem is to find an approximation to the posterior distribution $p(\boldsymbol{\Theta}|\boldsymbol{\Phi}, \mathbf{y})$ assuming that the marginal distribution, $p(\mathbf{y})$, is intractable. For notational simplicity the conditional dependence on $\boldsymbol{\Phi}$ will be assumed implicit and dropped from the following discussion.

The marginal distribution is defined as

$$p(\mathbf{y}) = \int_{\boldsymbol{\Theta}} p(\mathbf{y}|\boldsymbol{\Theta}) p(\boldsymbol{\Theta}) \, \mathrm{d}\boldsymbol{\Theta}. \tag{16}$$

Introducing any variational distribution, $Q(\boldsymbol{\Theta})$, to approximate $p(\boldsymbol{\Theta}|\mathbf{y})$ produces a lower bound on $p(\mathbf{y})$. This is achieved by first taking the natural logarithm of (16) and introducing $Q(\boldsymbol{\Theta})$ such that

$$\ln p(\mathbf{y}) = \ln \int_{\mathbf{\Theta}} Q(\mathbf{\Theta}) \frac{p(\mathbf{y}|\mathbf{\Theta})p(\mathbf{\Theta})}{Q(\mathbf{\Theta})} \, \mathrm{d}\mathbf{\Theta}. \quad (17)$$

$$\geq \int_{\mathbf{\Theta}} Q(\mathbf{\Theta}) \ln \frac{p(\mathbf{y}|\mathbf{\Theta})p(\mathbf{\Theta})}{Q(\mathbf{\Theta})} \, \mathrm{d}\mathbf{\Theta}. \quad (18)$$

$$= \mathcal{L}[Q(\mathbf{\Theta})] \quad (19)$$

where (18) follows from Jensen's inequality [39]. The variational lower bound (VLB) $\mathcal{L}[Q(\mathbf{\Theta})]$ can be maximised to provide an approximation of $\ln p(\mathbf{y})$. Variational optimisation is employed to perform the variational maximization of $\mathcal{L}[Q(\mathbf{\Theta})]$ with respect to the free distribution $Q(\mathbf{\Theta})$.

By noting that $Q(\mathbf{\Theta})$ is a proper probability distribution and therefore its integral with respect to $\mathbf{\Theta}$ is equal to unity, and rearranging (18) - (19),

$$\mathcal{L}[Q(\mathbf{\Theta})] = \int_{\mathbf{\Theta}} Q(\mathbf{\Theta}) \ln \frac{p(\mathbf{y}|\mathbf{\Theta})p(\mathbf{\Theta})}{Q(\mathbf{\Theta})} \, \mathrm{d}\mathbf{\Theta}. \quad (20)$$

$$= \int_{\mathbf{\Theta}} Q(\mathbf{\Theta}) \ln \frac{p(\mathbf{\Theta}|\mathbf{y})}{Q(\mathbf{\Theta})} + Q(\mathbf{\Theta}) \ln p(\mathbf{y}) \, \mathrm{d}\mathbf{\Theta}$$

$$= \int Q(\mathbf{\Theta}) \ln p(\mathbf{y}, \mathbf{\Theta}) \mathrm{d}\mathbf{\Theta} - \int Q(\mathbf{\Theta}) \ln Q(\mathbf{\Theta}) \mathrm{d}\mathbf{\Theta} \quad (21)$$

$$\ln p(\mathbf{y}) = \mathcal{L}[Q(\mathbf{\Theta})] - \int_{\mathbf{\Theta}} Q(\mathbf{\Theta}) \ln \frac{p(\mathbf{\Theta}|\mathbf{y})}{Q(\mathbf{\Theta})} \, \mathrm{d}\mathbf{\Theta} \quad (22)$$

$$= \mathcal{L}[Q(\mathbf{\Theta})] + KL(Q(\mathbf{\Theta})||p(\mathbf{\Theta}|\mathbf{y})) \quad (23)$$

where

$$KL(Q(\mathbf{\Theta})||p(\mathbf{\Theta}|\mathbf{y})) = -\int_{\mathbf{\Theta}} Q(\mathbf{\Theta}) \ln \frac{p(\mathbf{\Theta}|\mathbf{y})}{Q(\mathbf{\Theta})} \, \mathrm{d}\mathbf{\Theta} \quad (24)$$

is the Kullback-Leibler (KL) divergence from Q to p. A few things of interest can be noted from (22). First, the log marginal distribution, $p(\mathbf{y})$, can be de-constructed into a lower bound on the distribution and a measure of the difference between the true posterior distribution, $p(\mathbf{\Theta}|\mathbf{y})$, and its approximating distribution, $Q(\mathbf{\Theta})$, in the form of the KL divergence. It is then evident that the optimal approximation to the marginal distribution is found by maximising the lower bound, or equivalently minimising the KL divergence. It is also clear that the choice of $Q(\mathbf{\Theta}) = p(\mathbf{\Theta}|\mathbf{y})$ would result in an exact match between the bound and the marginal distribution.

B. Factorised distributions

In order to make the variational optimisation of $\mathcal{L}[Q(\mathbf{\Theta})]$ feasible, the family of possible distributions, $Q(\mathbf{\Theta})$, over which the optimisation is performed, must be restricted. The assumption is made that the variational distribution can be factorised such that

$$Q(\mathbf{\Theta}) = \prod_j q_j(\Theta_j) \quad (25)$$

where each $q_j(\Theta_j)$ are independent, an approximation known as the mean field theory in physics. The function is maximised with respect to each distribution $q_j(\Theta_j)$ separately while holding all others fixed.

Substituting the factorised distribution (25) into (21) and then separating the i'th distribution over which to perform the optimisation gives

$$\mathcal{L}[Q(\mathbf{\Theta}))] = \int \prod_j q_j(\Theta_j) \left(\ln p(\mathbf{y}, \mathbf{\Theta}) - \sum_j \ln q_j(\Theta_j) \right) \mathrm{d}\mathbf{\Theta}$$

$$= \int q_i(\Theta_i) \left(\int \ln p(\mathbf{y}, \mathbf{\Theta}) \prod_{j \neq i} q_j(\Theta_j) \mathrm{d}\Theta_j \right) \mathrm{d}\Theta_i - \int q_i(\Theta_i) \ln q_i(\Theta_i) \mathrm{d}\Theta_i + \text{const} \quad (26)$$

where the terms $\prod_{j \neq i} q_j(\Theta_j) \ln q_j(\Theta_j)$ have been absorbed into a constant term.

In order to find the distribution $q_i(\Theta_i)$ that maximises $\mathcal{L}[Q(\mathbf{\Theta})]$ a variational optimisation is performed with respect to $q_i(\Theta_i)$ such that (full details of the following derivations can be found in [44])

$$\frac{\delta \mathcal{L}[q_i(\Theta_i)]}{\delta q_i(\Theta_i)} = \frac{\partial}{\partial q_i(\Theta_i)} \left(q_i(\Theta_i) \left(\int \ln p(\mathbf{y}, \mathbf{\Theta}) \prod_{j \neq i} q_j(\Theta_j) \mathrm{d}\Theta_j \right) - q_i(\Theta_i) \ln q_i(\Theta_i) \right) \quad (27)$$

$$= \int \ln p(\mathbf{y}, \mathbf{\Theta}) \prod_{j \neq i} q_j(\Theta_j) \mathrm{d}\Theta_j - \ln q_i(\Theta_i) + \text{const.} \quad (28)$$

where $\frac{\delta F[f(x)]}{\delta f(x)}$ denotes the functional derivative of the functional $F[f(x)]$ with respect to the function $f(x)$ and $\frac{\partial f(x)}{\partial x}$ denotes the partial derivative of the function $f(x)$ with respect to the variable x.

The integral that forms the first term on the right hand side of (28) is the expectation of the log joint distribution where the expectation is taken with respect to all of the distributions $q_j(\Theta_j)$ for which $j \neq i$, such that

$$\mathbb{E}_{j \neq i}[\ln(\mathbf{y}, \mathbf{\Theta})] = \int \ln p(\mathbf{y}, \mathbf{\Theta}) \prod_{j \neq i} q_j(\Theta_j) \mathrm{d}\Theta_j \quad (29)$$

where $\mathbb{E}_{j \neq i}$ denotes the expectation with respect to the distributions q over all the variables in the set $\mathbf{\Theta}$ for which $j \neq i$.

Substituting (29) into (28), equating to zero and rearranging for the variational distribution, a general expression for the optimal solution and therefore the update of the ith factor of the variational distribution $q_i^{(t+1)}(\Theta_i)$ is then given by

$$\ln q_i^{(t+1)}(\Theta_i) = \mathbb{E}_{j \neq i}[\ln p(\mathbf{y}, \mathbf{\Theta})] + \text{const.} \quad (30)$$

Performing the update (30) for each factor $q_j(\Theta_j)$ of the variational distribution completes one optimisation step of $\mathcal{L}[Q(\boldsymbol{\Theta})]$. The optimisation is performed iteratively where t indicates the current iteration such that $q_j^{(t+1)}$ is the update using the statistics of the distributions $q_j^{(t)}$ at the previous iteration, see Figure 2.

Equation (30) can also be arrived at using the Kullback-Leibler divergence, as detailed in [39] in Section 10.1.1.

C. Convergence

The lower bound is guaranteed to converge because it is convex with respect to each of the factors of $q_i^{(t+1)}(\Theta_i)$. In order to facilitate the computation of the VLB, (21) can be written more explicitly as

$$\mathcal{L}[Q(\boldsymbol{\Theta})] = \mathbb{E}_{\Theta}[\ln p(\mathbf{y}, \boldsymbol{\Theta})] - \mathbb{E}_{\Theta}[\ln Q(\boldsymbol{\Theta})] \tag{31}$$

which is the form used to monitor convergence of the algorithm.

Although the calculation of the VLB is not required for the inference problem it provides a check that the algorithm, as well as the theory behind it, is functioning correctly as the bound is guaranteed to increase with each update of the variational distribution [44]. In addition, it is important to note that the VLB will be used later as a method for model comparison and therefore must be calculated for that purpose.

IV. Variational inference for NARX models

A. Variational inference of model parameters

The variational Bayesian inference procedure can now be applied to the NARX model defined by (2) following [45]. Applying the approximation defined by (25), the assumption is made that the posterior distribution $p(\boldsymbol{\theta}, \tau, \boldsymbol{\alpha}|\mathbf{y})$ can be approximated by

$$Q(\boldsymbol{\theta}, \tau, \boldsymbol{\alpha}) = q(\boldsymbol{\theta}, \tau) q(\boldsymbol{\alpha}) \tag{32}$$

whereby the VLB can be maximised with respect to each factor. From (18), the VLB is given by

$$\begin{aligned} \mathcal{L}[Q(\boldsymbol{\theta}, \tau, \boldsymbol{\alpha})] &= \iiint Q(\boldsymbol{\theta}, \tau, \boldsymbol{\alpha}) \\ &\quad \ln \frac{p(\mathbf{y}|\boldsymbol{\Phi}, \boldsymbol{\theta}, \tau, \alpha) p(\boldsymbol{\theta}, \tau|\boldsymbol{\alpha}) p(\boldsymbol{\alpha})}{Q(\boldsymbol{\theta}, \tau, \boldsymbol{\alpha})} \mathrm{d}\boldsymbol{\theta} \mathrm{d}\tau \mathrm{d}\boldsymbol{\alpha}. \end{aligned} \tag{33}$$

Using the results of the variational Bayesian inference derived above, the optimisation of the bound is performed via (30) for the distributions $q(\boldsymbol{\theta}, \tau)$ and $q(\boldsymbol{\alpha})$ resulting in a set of closed form update equations.

1) Update for $q(\boldsymbol{\theta}, \tau)$: The variational posterior $q_K(\boldsymbol{\theta}, \tau)$ is found by maximising the VLB, $\mathcal{L}(Q)$, with fixed $q(\boldsymbol{\alpha})$, where the subscript K denotes the updated parameters. Noting that $p(\mathbf{y}, \boldsymbol{\Theta})$ is given by (14) when $\boldsymbol{\Theta} = \{\boldsymbol{\theta}, \tau, \boldsymbol{\alpha}\}$, then from (30) the update equation for $q(\boldsymbol{\theta}, \tau)$ is given as

$$\begin{aligned} \ln q_K(\boldsymbol{\theta}, \tau) &= \ln p(\mathbf{y}|\boldsymbol{\Phi}, \boldsymbol{\theta}, \tau) + \mathbb{E}_{\boldsymbol{\alpha}}[\ln p(\boldsymbol{\theta}, \tau|\boldsymbol{\alpha})] \\ &\quad + \text{const} \end{aligned} \tag{34}$$

$$\begin{aligned} &= \left(\frac{M}{2} + a_0 - 1 + \frac{N}{2}\right) \ln(\tau) \\ &\quad - \frac{\tau}{2} \left(\boldsymbol{\theta}^T \left(\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{A}] + \sum_{k=1}^{N} \Phi_k^T \Phi_k \right) \boldsymbol{\theta} \right. \\ &\quad \left. + \sum_{k=1}^{N} y_k^2 - 2 \sum_{k=1}^{N} y_k \Phi_k \boldsymbol{\theta} + 2b_0 \right) \\ &\quad + \text{const} \end{aligned} \tag{35}$$

where $p(\mathbf{y}|\boldsymbol{\Phi}, \boldsymbol{\theta}, \tau)$ and $p(\boldsymbol{\theta}, \tau|\boldsymbol{\alpha})$ are given by (8) and (11) respectively and all the terms not dependent on $\boldsymbol{\theta}$ and τ have been absorbed into the constant term.

Given that the likelihood function, (8), is in the form of a Normal distribution, the conjugate Normal-Gamma prior, (11), is chosen. Hence, the posterior distribution can be expressed as

$$q_K(\boldsymbol{\theta}, \tau) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}_K, \tau^{-1}\mathbf{V}_K)\text{Gam}(\tau|a_K, b_K), \tag{36}$$

The method of completing the square is used to find the exponent of the Normal distribution in the posterior by equating coefficients of (35) and (36). First, separating out all the coefficients of $-\frac{\tau}{2}\boldsymbol{\theta}^T\boldsymbol{\theta}$, to find $\mathbf{V}_K^{-1}$

$$-\frac{\tau}{2}\boldsymbol{\theta}^T\mathbf{V}_K^{-1}\boldsymbol{\theta} = -\frac{\tau}{2}\boldsymbol{\theta}^T \left(\sum_{k=1}^{N} \Phi_k^T \Phi_k + \mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{A}] \right) \boldsymbol{\theta} \tag{37}$$

$$\mathbf{V}_K^{-1} = \sum_{k=1}^{N} \Phi_k^T \Phi_k + \mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{A}]. \tag{38}$$

Separating out the coefficients of $\boldsymbol{\theta}$ to find $\boldsymbol{\theta}_K$

$$\tau\boldsymbol{\theta}^T\mathbf{V}_K^{-1}\boldsymbol{\theta}_K = \tau\boldsymbol{\theta}^T \sum_{k=1}^{N} \Phi_k^T y_k \tag{39}$$

$$\boldsymbol{\theta}_K = \mathbf{V}_K \sum_{k=1}^{N} \Phi_k^T y_k. \tag{40}$$

Now, (36) can be expressed as

$$\begin{aligned} \ln q_K(\boldsymbol{\theta}, \tau) &= \ln \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}_K, \tau^{-1}\mathbf{V}_K) \\ &\quad - \frac{\tau}{2} \left(\sum_{k=1}^{N} y_k^2 + 2b_0 - \boldsymbol{\theta}_K^T \mathbf{V}_K^{-1} \boldsymbol{\theta}_K \right) \\ &\quad + \left(a_0 - 1 + \frac{N}{2} \right) \ln(\tau) + \text{const}. \end{aligned} \tag{41}$$

where the second and third terms in (41) are to be equated with the terms in the Gamma distribution. Equating coefficients of τ

$$-\tau b_K = -\frac{\tau}{2} \left(\sum_{k=1}^{N} y_k^2 + 2b_0 - \boldsymbol{\theta}_K^T \mathbf{V}_K^{-1} \boldsymbol{\theta}_K \right) \tag{42}$$

$$b_K = b_0 + \frac{1}{2} \left(\sum_{k=1}^{N} y_k^2 - \boldsymbol{\theta}_K^T \mathbf{V}_K^{-1} \boldsymbol{\theta}_K \right). \tag{43}$$

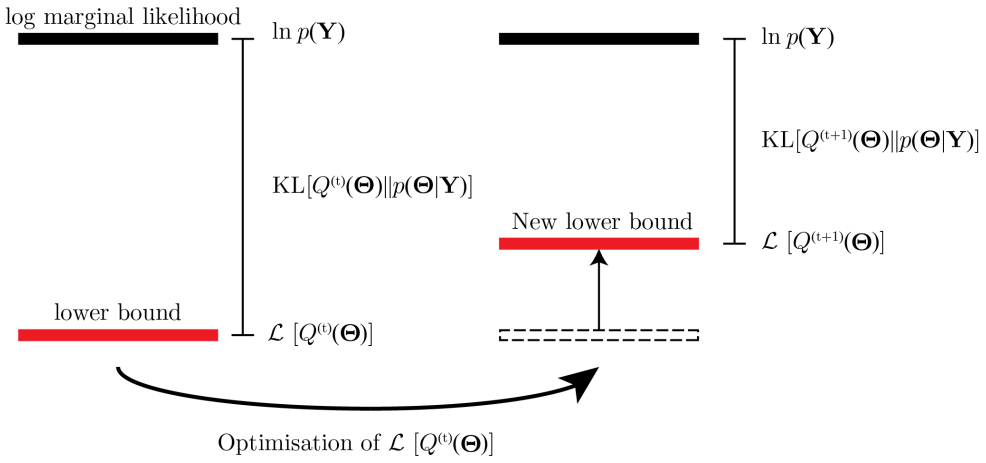

Fig. 2. Variational inference is performed by iteratively updating the VLB via an optimisation step: the diagram illustrates the variational Bayesian update according to (30).

And finally equating coefficients of $\ln \tau$

$$(a_K - 1)\ln\tau = \left(a_0 - 1 + \frac{N}{2}\right)\ln\tau \quad (44)$$

$$a_K = a_0 + \frac{N}{2}. \quad (45)$$

The update of $\ln q(\boldsymbol{\theta}, \tau)$ is by the computation of (38), (40), (43) and (45).

2) Update for $q_K(\boldsymbol{\alpha})$: The variational posterior $q(\boldsymbol{\alpha})$ is found by maximising the VLB, $\mathcal{L}(Q)$, with fixed $q(\boldsymbol{\theta}, \tau)$. Using (30), and equating coefficients of α_m,

$$\ln q_K(\boldsymbol{\alpha}) = \mathbb{E}_{\boldsymbol{\theta},\tau}(\ln p(\boldsymbol{\theta}, \tau|\boldsymbol{\alpha})) + \ln p(\boldsymbol{\alpha}) + \text{const}$$
$$= \sum_{m=1}^{M} \ln \text{Gam}(\alpha_m|c_K, d_{K_m}) \quad (46)$$

where

$$c_K = c_0 + \frac{1}{2} \quad (47)$$

$$d_{K_m} = d_0 + \frac{1}{2}\mathbb{E}_{\boldsymbol{\theta},\tau}[\tau\theta_m^2]. \quad (48)$$

The update of $q(\boldsymbol{\alpha})$ is hence performed by the computation of (47) - (48).

The required expectations are found by considering the standard moments of the relevant distributions [39] such that

$$\mathbb{E}_{\boldsymbol{\theta},\tau}[\tau\theta_m^2] = \theta_{K_m}^2 \frac{a_K}{b_K} + \mathbf{V}_{K_{mm}} \quad (49)$$

and the required expectation for the update of $q(\boldsymbol{\theta}, \tau)$ is given by

$$\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{A}] = \mathbf{A}_K \quad (50)$$

where $\mathbf{A}_K$ is a diagonal matrix with elements

$$\mathbb{E}_{\boldsymbol{\alpha}}[\alpha_m] = \frac{c_K}{d_{K_m}}. \quad (51)$$

The variational approximation, $Q(\boldsymbol{\theta}, \tau, \boldsymbol{\alpha}) = q(\boldsymbol{\theta}, \tau)q(\boldsymbol{\alpha})$, to the true posterior $p(\boldsymbol{\theta}, \tau, \boldsymbol{\alpha}|\mathbf{y})$ is found by updating $q(\boldsymbol{\theta}, \tau)$ and $q(\boldsymbol{\alpha})$ by iterating between the update equations defined by (38), (40), (43) and (45), and (47)-(48) respectively. These updates then form an iterative algorithm that is summarised in Algorithm 1. The variables $b_K, d_{K_m}, \boldsymbol{\theta}_K$ and $\mathbf{V}_K$ must first be initialised: b_K and d_{K_m} are initialised to their respective prior values, $b_K|_{t=0} = b_0$ and $d_K|_{t=0} = d_0$. $\boldsymbol{\theta}_K$ and $\mathbf{V}_K$ are initialised using the least squares estimate, such that

$$\boldsymbol{\theta}_K|_{t=0} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\mathbf{y}, \quad (52)$$

$$\mathbf{V}_K|_{t=0} = \boldsymbol{\Phi}^T\boldsymbol{\Phi}. \quad (53)$$

B. Variational lower bound $\mathcal{L}(Q)$

The VLB is found by considering (31). The dependencies between the parameters in the first term in (31) are easily determined by considering the joint distribution given by (14) and the probabilistic graphical model in Figure 1. Expanding the second term follows from (32) along with (36) and (46). The lower bound is then given by

$$\mathcal{L}[Q(\boldsymbol{\Theta})] = \mathbb{E}_{\boldsymbol{\Theta}}[\ln p(\mathbf{y}, \boldsymbol{\Theta})] - \mathbb{E}_{\boldsymbol{\Theta}}[\ln Q(\boldsymbol{\Theta})] \quad (54)$$

$$= \mathbb{E}_{\boldsymbol{\theta},\tau}[\ln(p(\mathbf{y}|\boldsymbol{\Phi}, \boldsymbol{\theta}, \tau))] + \mathbb{E}_{\boldsymbol{\theta},\tau,\boldsymbol{\alpha}}[\ln p(\boldsymbol{\theta}, \tau|\boldsymbol{\alpha})] + \mathbb{E}_{\boldsymbol{\alpha}}[\ln p(\boldsymbol{\alpha})] - \mathbb{E}_{\boldsymbol{\theta},\tau}[\ln q(\boldsymbol{\theta}, \tau)] - \mathbb{E}_{\boldsymbol{\alpha}}[\ln q(\boldsymbol{\alpha})]. \quad (55)$$

Taking the expectations of Equations (8), (11), (13), (36) and (46), the components of (55) are evaluated as:

$$\mathbb{E}_{\boldsymbol{\theta},\tau} \ln p(\mathbf{y}|\boldsymbol{\Phi}, \boldsymbol{\theta}, \tau) = \frac{N}{2}(\psi(a_K) - \ln b_K - \ln 2\pi) - \frac{1}{2}\sum_{k=1}^{N}\left(\frac{a_K}{b_K}(y_k - \Phi_k\boldsymbol{\theta})^2 + \Phi_k\mathbf{V}_K\Phi_k^T\right) \quad (56)$$

where $\psi(\cdot)$ is the digamma function.

$$\mathbb{E}_{\boldsymbol{\theta},\tau,\boldsymbol{\alpha}} \ln p(\boldsymbol{\theta},\tau|\alpha) = \frac{M}{2}(\psi(a_K) - \ln b_K + \psi(c_K) - \ln 2\pi) - b_0\frac{a_K}{b_K} - \frac{1}{2}\sum_{m=1}^{M}\left(\ln d_{K_m} + \frac{c_K}{d_{K_m}}\left(\frac{a_K}{b_K}\theta_{K_m}^2 + \mathbf{V}_{K_{mm}}\right)\right) - \ln\Gamma(a_0) + a_0\ln(b_0) + (a_0-1)(\psi(a_K) - \ln b_K) \quad (57)$$

$$\mathbb{E}_{\boldsymbol{\alpha}} \ln p(\boldsymbol{\alpha}) = -M(\ln\Gamma(c_0) + c_0 \ln d_0) + \sum_{m=1}^{M}\left((c_0-1)(\psi(c_K) - \ln d_{K_m}) - d_0\frac{c_K}{d_{K_m}}\right) \quad (58)$$

$$\mathbb{E}_{\boldsymbol{\theta},\tau} \ln q_K(\boldsymbol{\theta},\tau) = \frac{M}{2}(\psi(a_K) - \ln b_K - \ln 2\pi - 1) - \frac{1}{2}\ln|\mathbf{V}_K| - \ln\Gamma(a_K) + a_K \ln b_K + (a_K - 1)(\psi(a_K) - \ln b_K) - a_K \quad (59)$$

$$\mathbb{E}_{\boldsymbol{\alpha}} \ln q_K(\boldsymbol{\alpha}) = \sum_{m=1}^{M}((c_K - 1)\psi(c_K) + \ln d_{K_m}) - M(\ln\Gamma(c_K) + c_K). \quad (60)$$

Substituting (56) - (60) into (55) provides an expression for the variational bound as

$$\mathcal{L}(Q) = -\frac{N}{2}\ln 2\pi + \frac{1}{2}\ln|\mathbf{V}_K| - b_0\frac{a_K}{b_K} - \frac{1}{2}\sum_{k=1}^{N}\left(\frac{a_K}{b_K}(y_k - \Phi_k\boldsymbol{\theta}_K)^2 + \Phi_k\mathbf{V}_K\Phi_k^T\right) + \ln\Gamma(a_K) - a_K\ln b_K + a_K - \ln\Gamma(a_0) + a_0\ln b_0 - \sum_{m=1}^{M}(c_K\ln d_{K_m}) + M\left(\frac{1}{2} - \ln\Gamma(c_0) + c_0\ln d_0 + \ln\Gamma(c_K)\right) \quad (61)$$

The variational posterior distribution $Q(\boldsymbol{\theta},\tau,\boldsymbol{\alpha})$ can now be calculated iteratively by computing $q(\boldsymbol{\theta},\tau)$ and $q(\boldsymbol{\alpha})$. At each iteration the VLB, $\mathcal{L}(Q)$ can be computed via (61). The best approximation is found when $\mathcal{L}(Q)$ plateaus such that the condition $\mathcal{L}(Q)_t - \mathcal{L}(Q)_{t-1} <= T_{\mathcal{L}(Q)}$ is met, where $T_{\mathcal{L}(Q)}$ is a predefined threshold.

C. Predictive distribution

Predictions of a new, unseen, data point can be made by calculating a predictive distribution for the model at sample $k+1$. Given the input-output training data $\mathcal{D}$, the task is the evaluation of the distribution $p(y_{k+1}|\mathcal{D})$. The predictive distribution is found by marginalising over the parameters such that

Algorithm 1: VBNARX

procedure VBNARX($\boldsymbol{\Phi}, \mathbf{y}, T_{\mathcal{L}(\mathcal{Q})}, a_0, b_0, c_0, d_0$)
  $t = 0$
  $b_K = b_0$
  $d_K = d_0$
  $a_K = a_0 + \frac{N}{2}$
  $c_K = c_0 + \frac{1}{2}$
  $\boldsymbol{\theta}_K = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\mathbf{y}$
  $\mathbf{V}_K = \boldsymbol{\Phi}^T\boldsymbol{\Phi}$
  while $\mathcal{L}(Q)_t - \mathcal{L}(Q)_{t-1} > T_{\mathcal{L}(Q)}$ **do**
    $t = t + 1$
    Variational E Step:
    $\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{A}] = \mathbf{A}_K$, where $\mathbf{A}_{K,ii} = c_K/d_{Ki}$
    $\mathbb{E}_{\boldsymbol{\theta},\tau}[\tau\theta_m^2] = \theta_{K_m}^2\frac{a_K}{b_K} + \mathbf{V}_{K_{mm}}$
    Variational M Step:
    $\mathbf{V}_K^{-1} = \sum_{k=1}^{N}\Phi_k^T\Phi_k + \mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{A}]$
    $\boldsymbol{\theta}_K = \mathbf{V}_K\sum_{k=1}^{N}\Phi_k^T y_k$
    $b_K = b_0 + \frac{1}{2}\left(\sum_{k=1}^{N} y_k^2 - \boldsymbol{\theta}_K^T\mathbf{V}_K^{-1}\boldsymbol{\theta}_K\right)$
    $d_{K_m} = d_0 + \frac{1}{2}\mathbb{E}_{\boldsymbol{\theta},\tau}[\tau\theta_m^2]$
    Update $\mathcal{L}(Q)_t$ via (61)
  end while
  return $\mathcal{L}(Q)_t, c_K, d_K, \boldsymbol{\Theta}_{\boldsymbol{K}}$
end procedure

Fig. 3. Algorithmic description of the parameter estimation procedure for the NARX model using variational Bayesian inference, termed VBNARX

that

$$\begin{aligned} p(y_{k+1}|\Phi_{k+1},\mathcal{D}) &= \iiint p(\Phi_{k+1}|\Phi_{k+1},\boldsymbol{\theta},\tau)p(\boldsymbol{\theta},\tau,\boldsymbol{\alpha}|\mathcal{D})\mathrm{d}\boldsymbol{\theta}\mathrm{d}\tau\mathrm{d}\boldsymbol{\alpha} \\ &\approx \iiint p(y_{k+1}|\Phi_{k+1},\boldsymbol{\theta},\tau)q(\boldsymbol{\theta},\tau)q(\boldsymbol{\alpha})\mathrm{d}\boldsymbol{\theta}\mathrm{d}\tau\mathrm{d}\boldsymbol{\alpha} \\ &= \iint \mathcal{N}(y_{k+1}|\Phi_{k+1}\boldsymbol{\theta}_K,\tau^{-1})\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}_K,\tau^{-1}\mathbf{V}_K) \\ &\quad \mathrm{Gam}(\tau|a_K,b_K)\mathrm{d}\boldsymbol{\theta}\mathrm{d}\tau \\ &= \int \mathcal{N}(y_{k+1}|\Phi_{k+1}\boldsymbol{\theta}_K,\tau^{-1}(1+\Phi_{k+1}\mathbf{V}_K\Phi_{k+1}^T)) \\ &\quad \mathrm{Gam}(\tau|a_K,b_K)\mathrm{d}\tau \\ &= \mathrm{St}(y_{k+1}|\mu,\lambda,\nu) \end{aligned} \quad (62)$$

where $\mu = \Phi_{k+1}\boldsymbol{\theta}_K$, $\lambda = \frac{a_K}{b_K}(1 + \Phi_{k+1}\mathbf{V}_K\Phi_{k+1}^T)^{-1}$ and $\nu = 2a_K$. The resulting distribution, denoted St, is a Student's t-distribution. The distribution over $\boldsymbol{\alpha}$ does not appear in the third line of the above derivation because it is independent of the other distributions and hence it integrates to unity. In the final step, standard results from convolving conjugate distributions have been used [39]. The mean and variance of the distribution are given by

$$\mathbb{E}[y_{k+1}] = \Phi_k\boldsymbol{\theta}_K, \quad (63)$$

$$\mathrm{Var}[y_{k+1}] = (1 + \Phi_k\mathbf{V}_K\Phi_k^T)\frac{b_K}{(a_K - 1)}. \quad (64)$$

V. Structure detection for NARX models with variational inference and ARD

The variational Bayesian inference procedure provides a method for estimating the posterior distributions of linear in the parameters models, such as those of the NARX form. Through the incorporation of ARD into the procedure, a measure of how relevant each basis function is to the prediction of the data is achieved. Importantly the VLB provides a measure of how good the approximate posterior distribution of the parameters is to the true posterior distribution. For a given data set, the VLB is directly comparable across different models, and hence provides a method for model selection. In this section we take advantage of these features of the variational Bayesian inference in order to develop an algorithm, named the SVB-NARX algorithm, for parsimonious model structure detection.

A. The SVB-NARX algorithm

In overview, the algorithm proceeds by iteratively pruning basis functions from an initial superset of basis function, denoted $\mathcal{M}_0$, until there is a single basis function remaining. At each iteration of the algorithm, a subset of the basis functions that are selected at the previous iteration is chosen. The subset selection is performed by making use of the sparse variational inference procedure, defined in Algorithm 1. The variational inference provides a measure of how relevant each basis function is to the generation of the data. The least relevant terms are not included in the new set. The SVB-NARX algorithm is summarised in Algorithm 2,

The initial superset of basis function, $\mathcal{M}_0$ is defined as

$$\mathcal{M}_0 = \{\phi_m\}_{m=1}^{M}, \tag{65}$$

where ϕ_m is the m'th basis function, corresponding to the m'th column of $\mathbf{\Phi}$, and $M = |\mathcal{M}_0|$ is the total number of basis functions in the current model structure.

At each iteration of the algorithm, denoted by s, the variational Bayesian inference procedure is applied to the model defined by the current set of basis functions, $\mathcal{M}_s$, using Algorithm 1. Upon convergence of Algorithm 1 the VLB is recorded as $\mathcal{L}(Q)^s$, which provides a measure of quality of the model structure defined by $\mathcal{M}_s$

ARD values, defined in Section II-E, associated with each basis function are calculated as

$$ARD^s = \left\{\mathbb{E}_{\boldsymbol{\alpha}}[\alpha_m]^{-1}\right\}_{m=1}^{M} \tag{66}$$

where $\mathbb{E}_{\boldsymbol{\alpha}}[\alpha_m]$ is given by equation (51).

Terms that correspond to ARD values falling below the threshold T^s_{ARD} are pruned from the model, *i.e.* they are not included in the new model structure, $\mathcal{M}_{s+1}$. The threshold is updated at each algorithm iteration as

$$\begin{aligned} \ln T^s_{ARD} &= \min(\ln ARD^s) \\ &+ \frac{\max(\ln ARD^s) - \min(\ln ARD^s)}{r} \end{aligned} \tag{67}$$

with the resolution, r, being a tuning parameter of the algorithm set by the modeller. Consequences of the choice of r are discussed in the following section.

Algorithm 2: SVBNARX

procedure SVBNARX($\mathbf{\Phi}, \mathbf{y}, T_{\mathcal{L}(\mathcal{Q})}, T_{\text{ARD}}, a_0, b_0, c_0, d_0$)
  $\mathcal{M}_0 = \{\phi_m\}_{m=1}^{M}$
  $s = 0$
  while $M > 1$ **do**
    $\mathcal{L}(Q)^s, c_K, d_K, \boldsymbol{\theta}^s_K =$
      VBNARX($\mathcal{M}_s, \mathbf{y}, T_{\mathcal{L}(\mathcal{Q})}, a_0, b_0, c_0, d_0$)
    Calculate $\{ARD^s_m\}_{m=1}^{M}$ via (66)
    Calculate T^s_{ARD} via (67)
    $\mathcal{M}^- = \emptyset$
    for $m = 1 : |\mathcal{M}_s|$ **do**
      if $ARD^s_m \leq T^s_{\text{ARD}}$ **then**
        $\mathcal{M}^- = \mathcal{M}^- \cup \phi_m$
      end if
    end for
    $\mathcal{M}_{s+1} = \mathcal{M}_s \setminus \mathcal{M}^-$
    $M = |\mathcal{M}_{s+1}|$
    $s = s + 1$
  end while
  $\mathcal{M}^* = \mathcal{M}_{s^*}$ where $s^* = \underset{s}{\text{argmax}}\mathcal{L}(Q)^s$
  $\boldsymbol{\theta}^*_K = \boldsymbol{\theta}^{s^*}_K$
  return $\mathcal{M}^*, \boldsymbol{\theta}^*_K$
end procedure

Fig. 4. Sparse Bayesian identification of the NARX model using variational inference and automatic relevance determination

The threshold is dependent on the range of the $\ln(ARD)$ values and removes terms at the lower fraction of this range depending on the value of r. This choice of threshold has the advantage of removing increasingly less terms at each algorithm iteration and hence discriminating more in the pruning as the number of basis functions decreases. $\ln(ARD)$ values are used to calculate the threshold, because the ARD values associated with highly relevant model terms can be very high in comparison to less relevant (but still correct) model terms. $\ln(ARD)$ values will provide greater discrimination between less relevant terms in this case. The iteration ends when $M = 1$ (all but one term have been pruned from the model).

The result of the iterative pruning is a selection of models, $\mathcal{M}_s$ with a diminishing number of basis functions, each of which can directly be compared by its associated VLB, $\mathcal{L}(Q)^s$. The optimal model choice, $\mathcal{M}^*$, is then selected as the model with the greatest maximum lower bound such that

$$\mathcal{M}^* = \mathcal{M}_{s^*}, \quad \text{where} \quad s^* = \max_s \mathcal{L}(Q)^s. \tag{68}$$

The justification for the optimal model being selected as the one that maximises the lower bound is given in Section V-C.

Note that the algorithm could be terminated at the maximum of the VLB, however, it is informative (and computationally inexpensive) to run the algorithm to completion. For a real world problem the peak may not be well defined and the modeller may wish to choose a model with less terms whilst penalising accuracy.

B. Algorithm Initialisation and Choice of Hyper-parameters

The SVB-NARX algorithm requires initialisation of the hyper-parameters associated with the prior distributions, namely a_0, b_0, c_0 and d_0. A standard choice of hyper-parameters are $a_0 = c_0 = 1 \times 10^{-2}$ and $b_0 = d_0 = 1 \times 10^{-4}$, so as to produce uninformative prior distributions [39], [45]. The mean of the Gamma distribution on τ^{-1} at these values is undefined but it has mode $b_0/(a_0 + 1) \approx 1 \times 10^{-4}$. This implies that the most likely variance on $\boldsymbol{\theta}$ will be small *a priori*. The *a priori* variance on τ^{-1} is also undefined at these values, however the variance on τ can be computed as $a_0/b_0^2 = 1 \times 10^6$. It can hence be concluded that although the prior distribution indicates a preference for $\boldsymbol{\theta}$ to take small values, this effect will be minimal on the inference because of the broad distribution. The same reasoning can be applied to the prior distribution on $\boldsymbol{\alpha}$.

The single tuning parameter of the algorithm is the resolution, r, whose value is set in advance by the modeller. It is named resolution because it defines the region of ARD values that are pruned from the model via (67). Increasing the value of r leads to a higher resolution, resulting in less terms being pruned at each iteration, s, and consequently longer computation time. Conversely, reducing the value of r increases the number of terms pruned at each iteration.

It is to be noted that if r is chosen too small then correct model terms may be incorrectly pruned from the model. However, the value of r can always be set high to avoid incorrectly pruning terms, where the only penalty is longer computation time. In addition, to mitigate effects introduced by tuning r it should also be noted that for a given model and data set the VLB is independent of the resolution that produced it. This allows for multiple algorithm runs with varying values of r to produce models that are directly comparable.

C. Algorithm properties

In this section we explain key properties of the algorithm.

1) Model selection by the variational lower bound: In the previous section it is stated that the optimal model choice is taken to be the model that maximises the VLB after it has reached convergence.

In (4) the conditional dependencies on the model $\mathcal{M}_s$ are neglected. Explicitly including the conditional dependencies, (4) can be written

$$p(\boldsymbol{\Theta}|\boldsymbol{\Phi}, \mathcal{M}_s) = \frac{p(\mathbf{y}|\boldsymbol{\Phi}, \boldsymbol{\Theta}, \mathcal{M}_s)p(\boldsymbol{\Theta}|\mathcal{M}_s)}{p(\mathbf{y}|\mathcal{M}_s)} \tag{69}$$

Considering the posterior distribution over the models, $\mathcal{M}_s$, conditional on the data and applying Bayes' theorem the posterior distribution over the s'th model is given by

$$p(\mathcal{M}_s|\mathbf{y}) = \frac{p(\mathbf{y}|\mathcal{M}_s)p(\mathcal{M}_s)}{p(\mathbf{y})}. \tag{70}$$

The first term in the numerator on the right hand side of (70) is the same as the marginal likelihood in (69). Setting equal prior distributions $p(\mathcal{M}_s)$ for each model and noting that the denominator is constant for a given data set, the posterior is proportional to the marginal likelihood in (69)

$$p(\mathcal{M}_s|\mathbf{y}) \ \propto p(\mathbf{y}|\mathcal{M}_s). \tag{71}$$

For a more in depth discussion of the role of the marginal likelihood in Bayesian model selection see [15].

The VLB, $\mathcal{L}(Q)^s$, calculated for each model is an approximation of the marginal likelihood, $p(\mathbf{y}|\mathcal{M}_s)$. Equation (71) therefore provides the justification for using the VLB as a criterion for selecting the final model structure.

2) Computational complexity: The computational complexity of the SVB-NARX algorithm is dominated by the matrix inversion of the result of (38) in order to perform the variational update of the model parameters. The computational complexity of the algorithm is therefore cubic in the number of parameters $O(M^3)$. The pruning step of the algorithm results in a smaller set of model parameters, M, being evaluated at each iteration of the algorithm and hence the time taken for the variational updates to reach convergence decreases significantly as the algorithm progresses.

In practice the computational complexity of SVB-NARX translates to run times that can be an order of magnitude less than methods based on MCMC (see Section VI). The long computation times in MCMC methods are not attributed to the complexity of individual mathematical operations, but rather tends to depend on the random initialisation of samples, and whether the initialisation is close to the stationary distribution of model and parameters, and is then affected by how long the Markov chain takes to converge to that stationary distribution: these aspects are difficult to quantify precisely because they depend on the inherent randomness of the MCMC process. Similar computational comparisons have been made elsewhere in the literature for VB and MCMC methods [46].

VI. Results and Discussion

In this section the SVB-NARX algorithm is demonstrated and benchmarked on a numerical example and then applied to a real system. The benchmark algorithms include a sparse Bayesian LASSO (BL) algorithm [18], a Bayesian algorithm by some of the authors based on reversible jump MCMC (RJMCMC) [26], the FRO algorithm based on orthogonal least squares [4], a standard LASSO algorithm [47], and a simulation based method, SEMP, [9]. The algorithm is then applied to the identification of a real system, a dielectric elastomer actuator used in soft robotics [36], [48], [49].

A. Benchmarking on a known nonlinear system

The SVB-NARX algorithm was demonstrated and benchmarked on the system below, a polynomial NARX model, which has previously been used as a challenging example because a popular algorithm, FRO, fails to select all terms correctly [9], [50],

$$\begin{aligned} y_k = \theta_1 y_{k-2} + \theta_2 y_{k-1} u_{k-1} + \theta_3 u_{k-2}^2 \\ + \theta_4 y_{k-1}^3 + \theta_5 y_{k-2} u_{k-2}^2 + e_k \end{aligned} \tag{72}$$

where $\theta_1 = -0.5$, $\theta_2 = 0.7$, $\theta_3 = 0.6$, $\theta_4 = 0.2$, $\theta_5 = -0.7$, and e_k is a Normally distributed white noise sequence drawn from the distribution $\mathcal{N}(e_k|0, \sigma_e^2)$. The system was simulated for $N = 1000$ data samples at a noise level of $\sigma_e^2 = 0.0004$ generating signals with a signal-to-noise ratio (SNR) ≈20 dB, where SNR=$10\log_{10}\left(\sigma_y^2/\sigma_e^2\right)$, where σ_y^2 is the noise-free output variance. Additional SNRs in the range 2-20 dB were also investigated). The input, u_k, was drawn from a uniform distribution in the range $[-1, 1]$.

To perform the term selection, a superset of basis functions is typically defined. For polynomial models, basis functions in the superset can take the form of any polynomial combination of the elements of $\mathbf{x}_k$ up to a maximum order n_p [2]. In this case $n_p = 3$. In addition the maximum dynamic orders for input and output were set to $n_u = n_y = 4$. This led to an initial superset of $\mathcal{M} = 164$ model terms (for all algorithms). Algorithm initialisation was performed as follows.

SVB-NARX: for SVB-NARX initialisation, prior distibutions were set using hyper-parameter values as discussed in Section V-B. The resolution of the algorithm was tested at values of $r = 10, 100$ and 1000 to illustrate the effect of this tuning parameter.

Bayesian-LASSO: the BL algorithm requires the manual tuning of the parameter λ_{BL}, which is defined as the variance of the noise, σ_e^2. Setting λ_{BL} to the true noise variance was found to result in an incorrect term selection. The parameter was therefore set to $\lambda_{BL} = 0.002$, which gave the correct terms.

RJMCMC: the RJMCMC algorithm hyper-parameters for prior distributions were set as in [26]. In addition the noise variance was manually set to $\sigma_e^2 = 0.1$. The RJMCMC algorithm was executed 100 times for $100,000$ iterations per execution.

FRO: for FRO, the algorithm is usually terminated when the error reduction (ERR) ratio value drops below a threshold: in this case the threshold was set to 0.01.

LASSO: the LASSO algorithm used here was based on the method reported in [47], with regularisation weight set to $\lambda_{LASSO} = 0.0043$, selected from a range of values to minimise the mean squared prediction error (MSPE) from a 10-fold cross-validation test, where MSPE $= \frac{1}{N}\sum_k (y_k - \hat{y}_{\text{mpo},k})^2$, where $\hat{y}_{\text{mpo},k}$ is the k'th element of the model predicted output.

SEMP: the SEMP algorithm is usually terminated when the change in MSPE drops below a threshold, where a decrease of less than 1×10^{-6} was taken as the threshold here.

All algorithms were implemented by the authors in Matlab except BL, where a toolbox by the original authors of [18] at https://github.com/panweihit/BSID was used.

The SVB-NARX algorithm, with $r = 100$, performed well over a range of noise levels, selecting the correct model terms at SNRs of 2-20 dB (see Figure 5, where the dashed black line indicates the correct model structure and corresponds in all instances to the maximum of the variational bound). The true model parameters were within the 95% confidence intervals at all noise levels (this is not shown in Figure 5 in order not to overcrowd the plots on the bottom row). As should be expected, the variance of the parameter distributions increases with increasing noise, see Figure 5.

TABLE I. COMPARISON OF TERM SELECTION AND PARAMETER ESTIMATION FOR THE SYSTEM GIVEN BY (72).

SVB-NARX			
Correct Term?	Basis function	ARD ($\times 10^3$)	Parameter estimate
Yes	y_{k-2}	0.6076	-0.4985± 0.0053
Yes	$y_{k-1}u_{k-1}$	1.2103	0.7035± 0.0064
Yes	u_{k-2}^2	0.8788	0.5995± 0.0031
Yes	y_{k-1}^3	0.1004	0.2026± 0.0031
Yes	$y_{k-2}u_{k-2}^2$	1.2119	-0.7040± 0.0122
BL			
Correct Term?	Basis function		Parameter estimate
Yes	y_{k-2}		-0.4985± 0.0118
Yes	$y_{k-1}u_{k-1}$		0.7035± 0.0142
Yes	u_{k-2}^2		0.5995± 0.0069
Yes	y_{k-1}^3		0.2026± 0.0069
Yes	$y_{k-2}u_{k-2}^2$		-0.7041± 0.0272
RJMCMC			
Correct Term?	Basis function		Parameter estimate
Yes	y_{k-2}		-0.5010± 0.0023
Yes	$y_{k-1}u_{k-1}$		0.6998± 0.0035
Yes	u_{k-2}^2		0.5977± 0.0017
Yes	y_{k-1}^3		0.2018± 0.0008
Yes	$y_{k-2}u_{k-2}^2$		-0.6915± 0.0114
FRO			
Correct Term?	Basis function	ERR	Parameter estimate
No	$y_{k-4}u_{k-2}^2$	0.3582	0.0073
Yes	$y_{k-1}u_{k-1}$	0.1654	0.7037
Yes	u_{k-2}^2	0.1216	0.5980
Yes	y_{k-2}	0.2657	-0.4983
Yes	y_{k-1}^3	0.0563	0.2028
Yes	$y_{k-2}u_{k-2}^2$	0.0296	-0.6992
LASSO			
Correct Term?	Basis function	-	Parameter estimate
Yes	y_{k-2}		-0.4960
Yes	$y_{k-1}u_{k-1}$		0.6923
Yes	u_{k-2}^2		0.5881
Yes	y_{k-1}^3		0.1931
No	$y_{k-1}^2 y_{k-3}$		-0.0043
No	$y_{k-1}y_{k-2}^2$		0.0000
No	$y_{k-1}y_{k-3}^2$		0.0005
No	$y_{k-2}u_{k-1}^2$		-0.0035
Yes	$y_{k-2}u_{k-2}^2$		-0.6789
No	$y_{k-4}u_{k-2}^2$		0.0100
SEMP			
Correct Term?	Basis function	MSPE	Parameter estimate
Yes	u_{k-2}^2	0.1485	0.5995
Yes	y_{k-2}	0.1096	-0.4985
Yes	$y_{k-1}u_{k-1}$	0.0795	0.7035
Yes	y_{k-1}^3	0.0409	0.2027
Yes	$y_{k-2}u_{k-2}^2$	0.0264	-0.7040

In the benchmark study, all algorithms except FRO and LASSO correctly selected the model terms (Table I). The SVB-NARX algorithm correctly identified the model structure at each tested resolution level, $r = 10, 100$ and 1000, demonstrating its insensitivity to this tuning parameter, In Table I ARD (given by (66)) and model parameters are shown for $r = 100$. In addition, the algorithm inferred a distribution over the noise variance with a high degree of accuracy (estimated to be 0.00041, where the true value was 0.0004). For the BL algorithm the distributions over the model parameters had a significantly larger variance than for SVB-NARX or RJMCMC. The FRO algorithm selected an incorrect term at the first iteration (see Table I), which was likely due to the local nature of the search [51]. The LASSO algorithm selected 7 incorrect terms: this problem of LASSO overselecting model terms has been noted elsewhere

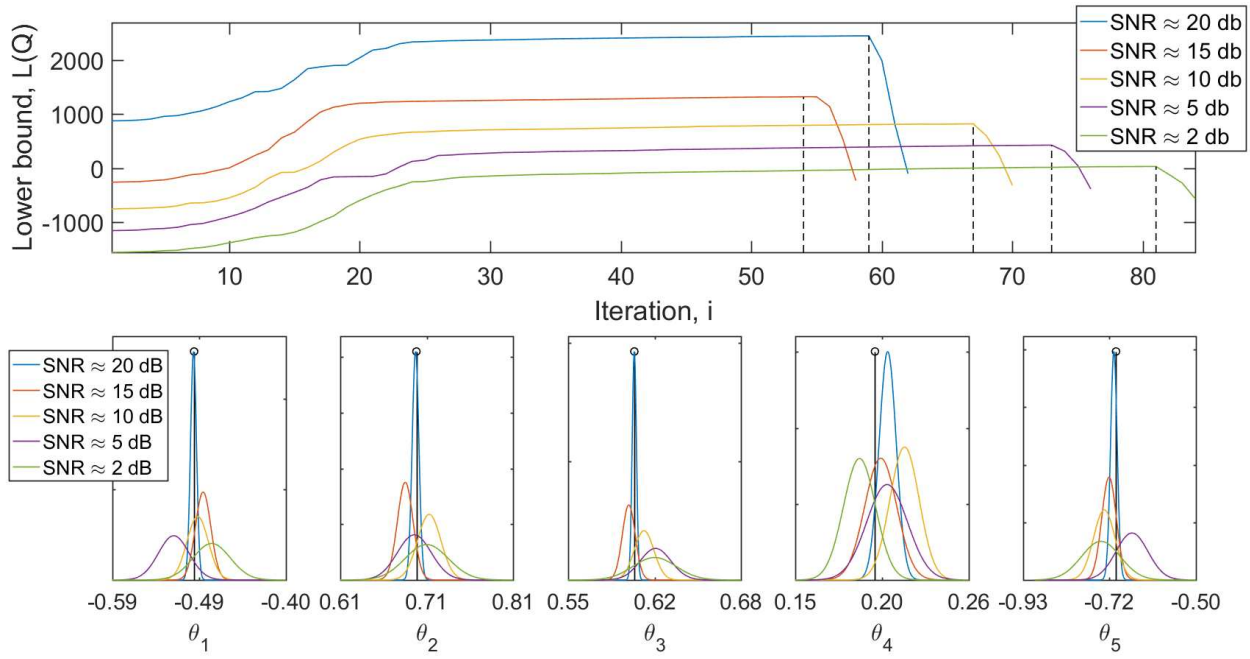

Fig. 5. Results of Numerical example 1. *Top*: SVB-NARX model structure detection at varying signal-to-noise ratios (SNRs), 2-20dB. The VLB is plotted against algorithm iteration number for each noise level. The correct model structure is indicated by the black dotted line and corresponds to the maximum of the VLB for all noise levels, indicating the correct model structure at all SNRs. The bound converges to a smaller value with increasing noise. *Bottom*: Parameter distributions calculated at different noise levels. The true parameter is given by the stem plot. As expected the parameter estimates are less certain at higher noise levels.

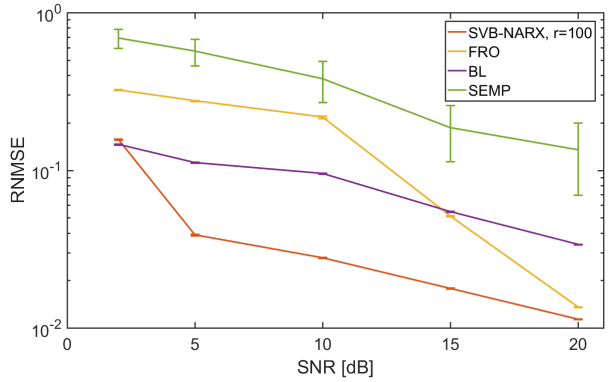

Fig. 6. Comparison of benchmark algorithms at varying levels of SNR using the average root normalised mean square error (RNMSE) of identified model parameters. The RNMSE average was taken over 100 Monte Carlo simulations. Error bars show 2 standard deviations from the mean.

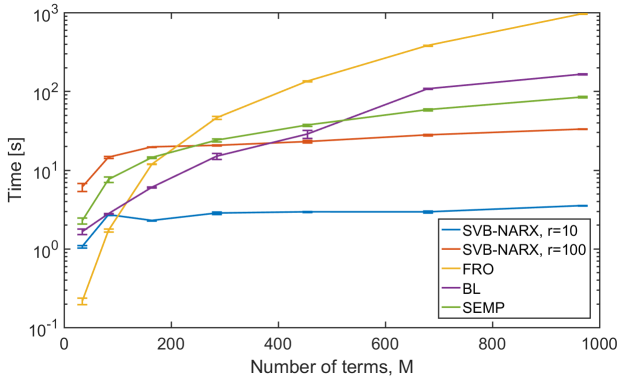

Fig. 7. Average timings of structure detection algorithms over 10 runs. Error bars show 2 standard deviations from the mean.

[13] (the true model structure was not recovered at any value of λ_{LASSO}). Finally, the SEMP algorithm correctly identified the model structure.

The same model was used to investigate the effect of noise on the identification performance for each of the benchmark algorithms (RJMCMC was omitted because of the long computation time required). In total, 100 input-output data sets were generated for six different SNR values in the range 2-20 dB. The initial super-set of model terms was set as described above (164 terms). The root of the normalised mean squared error (RNMSE) of the model parameters was used as a performance index, defined as $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_2 / \|\boldsymbol{\theta}\|_2$, where $\hat{\boldsymbol{\theta}}$ was the estimate of the true parameter $\boldsymbol{\theta}$. Note that the parameters of incorrectly selected terms were included here, in order to incorporate the full identification procedure in the comparison (hence, large errors were typically due to incorrect terms included in the selection procedure). Initialisation of all algorithms was performed as before; the SVB-NARX algorithm was run at a resolution of $r = 100$. SVB-NARX outperformed the other algorithms at all SNR levels except 2dB, for which BL was marginally better, see Figure 6.

To investigate computational efficiency, the benchmark example was repeated varying the size of the initial superset of model terms. This was done by increasing the dynamic orders $n_y = n_u = 2, \ldots, 8$. RJMCMC was omitted because it took orders of magnitude longer than the other algorithms (for 164 terms, it took $\sim$20 seconds per trial of 100,000 iterations but requiring multiple trials, in this case 100, i.e. $\sim$2000 seconds). The SVB-NARX algorithm was run at two resolution levels of $r = 10$ and $r = 100$ to demonstrate the effect of this tuning parameter. Initialisation and term selection for all other algorithms remained the same. Computational times were averaged over 10 trials. For a small initial basis function set ($<$100 terms) all of the tested algorithms were comparable, taking under 10 seconds to complete (Figure 7). Greater differences between algorithms were seen for more model terms: in particular SVB-NARX was fastest for over 500 terms and had a relatively flat trend as the number of terms increased toward 1000 (Figure 7). The greater efficiency of SVB-NARX is due to the pruning procedure removing multiple terms at each iteration, which rapidly decreases the dimensionality of the inference task as the algorithm progresses, leading to significantly decreased computation time at each iteration.

In summary, the benchmark has demonstrated that SVB-NARX has advantages in the automated tuning of the noise variance, as well as increased computational efficiency for large numbers of terms. The automated tuning of the noise variance with SVB-NARX is particularly important because alternatives such as BL and RJMCMC do not select the correct terms for certain manually tuned values of noise variance, and the successful value would be unknown for real world problems.

B. Experimental system identification of a dielectric elastomer actuator

The SVB-NARX algorithm was applied in this section to the identification of a set of six dielectric elastomer actuators (DEAs), which are a type of soft-smart actuator used in robotics [36], [48], [49]. DEAs are known to exhibit nonlinear dynamics, posing challenges for control design [37], [38], [52].

The dataset used in this investigation has already been published and so we only give brief details here, readers are referred to [37] for more information. The input-output data comprised voltage as input, and displacement as output (see Figure 8). The voltage input signal was designed as band-limited white noise (0-1Hz), with amplitude offset to lie in the range 1.5-3.5V, sampled at 50Hz (and then down-sampled for identification to 12.5Hz). We used 160 seconds of data for identification, N=2000 samples, and partitioned the data for identification and validation (comprising N=1000 data samples for each segment). Detailed results are given for one DEA, and then summarised for the remaining five. For comparison, the SEMP, FRO and BL algorithms are also evaluated.

Regarding initialisation of the identification algorithms, the superset of initial model terms was generated with $n_u = n_y = 3$ and $n_p = 3$. The SVB-NARX algorithm was initialised with $r = 1000$, and hyper-parameters were initialised as in Section V-B. The FRO algorithm was terminated when the ERR value fell below 1×10^{-5}. The SEMP algorithm was terminated by thresholding the MSPE. The BL algorithm required the selection of the tuning parameter λ, which was set to $\lambda = 5.5 \times 10^{-4}$, selected by running the algorithm over a range of values of λ and selecting the model with the minimum prediction error over the training data.

TABLE II. MODEL TERMS AND PARAMETERS OF A DEA IDENTIFIED USING SVB-NARX, SEMP, FRO AND BL. PARAMETER VALUES FOR THE SVB-NARX IDENTIFIED MODEL ARE GIVEN WITH THEIR 95% CONFIDENCE INTERVALS.

Terms	SVB-NARX	SEMP	FRO	BL
DC	-	0.0475	0.0518	-
y_{k-1}	0.9056 $\pm$ 0.0226	0.8950	1.0024	0.9619
y_{k-2}	-	-	-0.2471	-
y_{k-3}	0.1977 $\pm$ 0.0293	0.0415	0.1379	0.1487
u_{k-1}	-	0.4115	-	-
u_{k-2}	-	-0.4216	-	-0.0081
u_{k-3}	-	0.0671	-	-
$y_{k-1}y_{k-2}$	-	-0.0770	-	-
$y_{k-1}u_{k-1}$	-	-	3.0050	0.2048
$y_{k-1}u_{k-2}$	-0.5504 $\pm$ 0.0411	-	-5.1093	-0.6826
$y_{k-1}u_{k-3}$	-	-	1.7671	-
y_{k-2}^2	-0.2144 $\pm$ 0.0419	-	-	-0.2286
$y_{k-2}u_{k-1}$	0.5725 $\pm$ 0.0426	-	-2.2368	0.5311
$y_{k-2}u_{k-2}$	-	-	4.4082	-
$y_{k-2}u_{k-3}$	-	-	-1.7435	-
u_{k-1}^2	0.8962 $\pm$ 0.1620	-	-	-
$u_{k-1}u_{k-2}$	-1.3596 $\pm$ 0.3081	-	-	-
$u_{k-2}u_{k-3}$	1.1632 $\pm$ 0.3193	-	-	-
u_{k-3}^2	-0.6669 $\pm$ 0.1730	-	-	-
$y_{k-1}^2y_{k-2}$	-	-	-	-
$y_{k-1}^2u_{k-1}$	0.3082 $\pm$ 0.0497	-	-	-
$y_{k-1}y_{k-2}u_{k-3}$	-0.5610 $\pm$ 0.1472	-	-	-
$y_{k-1}u_{k-1}^2$	-	0.6566	- 0.2914	1.3845
$y_{k-1}u_{k-3}^2$	-	-	-0.2780	-0.1049
$y_{k-1}u_{k-1}u_{k-2}$	-	-	-	-1.3043
$y_{k-1}u_{k-2}^2$	-	-0.7383	-	-
$y_{k-2}^2u_{k-3}$	0.3906 $\pm$ 0.1061	-	-	
$y_{k-2}y_{k-3}u_{k-2}$	-	-	-	0.1182
$y_{k-2}u_{k-2}u_{k-3}$	-	-	-	0.6200
$y_{k-2}u_{k-1}u_{k-3}$	-	0.2786	-	-
$y_{k-2}u_{k-3}^2$	-	-	-	-0.2903
$y_{k-3}u_{k-1}^2$	-	-0.0366	-	-0.2530
u_{k-1}^3	0.2016 $\pm$ 0.0386	-	-	-
$u_{k-1}u_{k-2}u_{k-3}$	-0.3676 $\pm$ 0.1317	-	-	-
$u_{k-2}^2u_{k-3}$	0.1755 $\pm$ 0.1030	-	-	-
Prediction errors on validation dataset	2.47%	2.09%	2.43%	2.39%

Application of each identification algorithm to the training data resulted in a model with accurate model simulations on the validation data, around 2-3% error in each case (Figure 9 and Table II). There was some variation in the size of each model, where the number of terms selected was as follows: SVB-NARX 15 terms; FRO 12 terms; SEMP 11 terms; BL 14 terms (Table II). Hence, SEMP and FRO identified models with fewer terms than for BL and SVB-NARX. However, because the thresholds for SEMP and FRO are selected by hand-tuning, no strong conclusions can be drawn from these numbers. Additional terms do not necessarily indicate that the models identified by SVB-NARX and BL are any more complex: the parameters estimated by SVB-NARX are regularised due to the prior distribution over the parameters, whilst model complexity and over fitting are penalised automatically as part of the inference procedure.

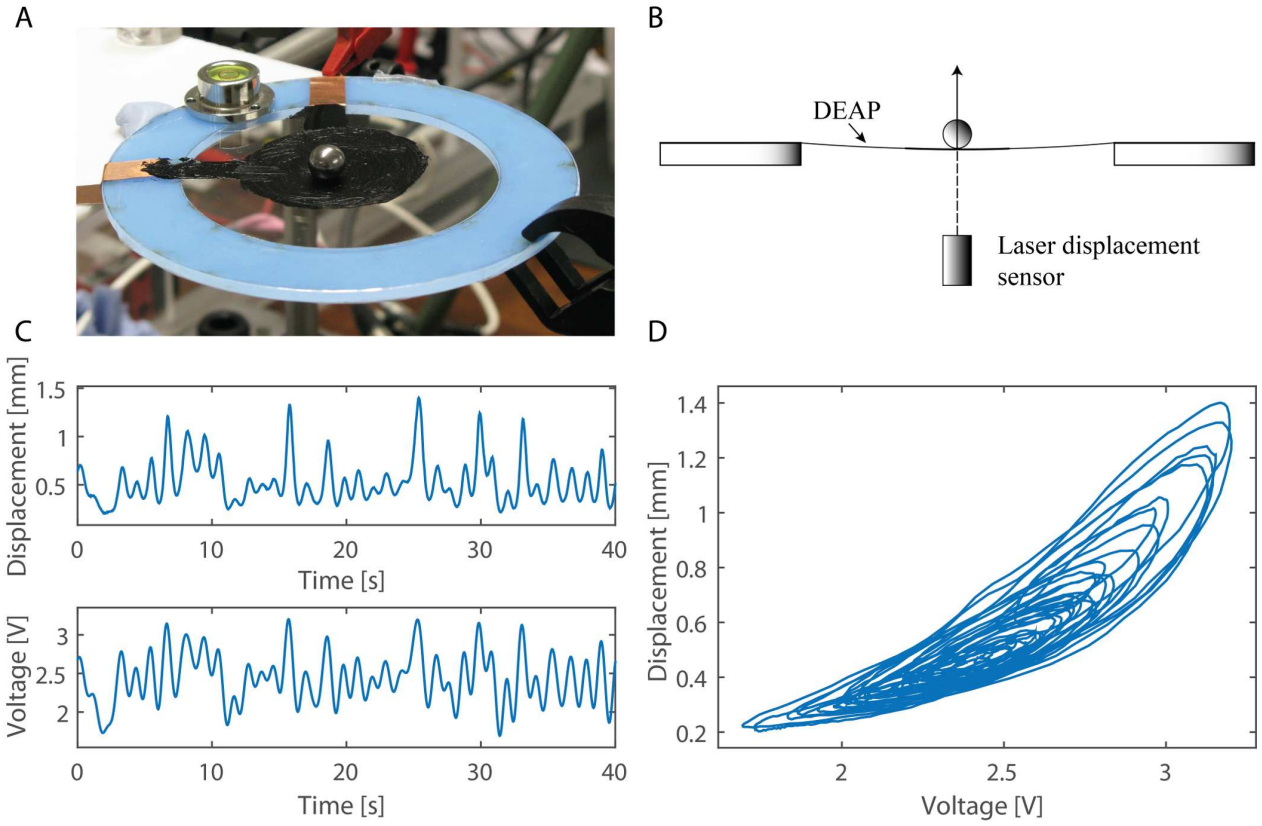

Fig. 8. Experimental setup for the dielectric elastomer actuators (DEAs). A: Photo of the DEA experimental rig: An elastomer film is anchored over a rigid blue plastic frame. Electrodes are painted on with conducting carbon grease (inner black circle), to form the DEA, and connected to an external power supply. A metal ball weights the DEA in the centre. On application of a voltage across the DEA the actuator expands raising the ball in the vertical direction: the resulting vertical displacement is the output measured by a laser displacement sensor underneath. B: Diagram of the DEA experimental rig. C: Typical input (voltage) and output (displacement) data obtained from one DEA experiment (zoomed for clarity to the first 40 seconds). D: Input vs output (voltage-displacement) plot, indicating nonlinear dynamics of the actuator.

A striking feature of the identified NARX models, shown in Table II, is that they consist of different terms, implying that they might describe different dynamics. However, comparison of dynamics based on the model equation is not informative because they are not necessarily unique descriptors [2]. NARX models are preferably compared and analysed in the frequency domain, where the dynamics should be uniquely described [2]. We performed such an analysis here via nonlinear output frequency response functions (NOFRFs) [2], [6].

We briefly explain the NOFRF analysis procedure here. The frequency response of a nonlinear system, $Y(j\omega)$, can be analysed using the sum of n-th order output spectra, $Y(j\omega) = \sum_n Y_n(j\omega)$ [53], which is based on the standard description originating in Volterra series analysis of nonlinear systems, $y(t) = \sum_n y_n(t)$, where $y_n(t)$ is known as the n-th order system output [54]. The NOFRFs allow the reconstruction of the n-th order output spectra as $Y_n(j\omega) = G_n(j\omega)U_n(j\omega)$, where $G_n(j\omega)$ is the NOFRF and $U_n(j\omega)$ is the spectrum of a specific input signal [6]. The procedure is based on: 1. simulating the identified NARX model with multiple level input signals chosen for specific spectral characteristics, 2. taking the fast Fourier transform (FFT) of the inputs and outputs, 3. estimating the NOFRFs, $G_n(j\omega)$, from input-output FFT data, and 4. using the estimated NOFRFs to reconstruct the n-th order output spectra, $Y_n(j\omega)$ [2], [6].

The n-th order output spectra, $Y_n(j\omega)$, can be used to analyse and compare models obtained by nonlinear system identification. This frequency-domain representation of the identified model should be a unique descriptor and therefore much more useful and informative than attempting to compare model equations (parameters and terms) directly.

The frequency domain analysis was performed for all six DEAs using SVB-NARX (and SEMP only, as the best performing alternative model in the sense of minimising prediction error). We found that whilst model equations differed, there was good agreement in the n-th order output spectra, $Y_n(j\omega)$, giving confidence that SVB-NARX and SEMP identified models described the same dynamics (see Figure 10). In addition, we used Monte Carlo sampling (100 samples) of the SVB-NARX model parameters to provide an uncertainty description in the frequency-domain (shaded region in Figure 10). This is the first time, to our knowledge, that this type of uncertainty analysis has been performed and could have important benefits in areas such as systems analysis, control design and fault diagnosis.

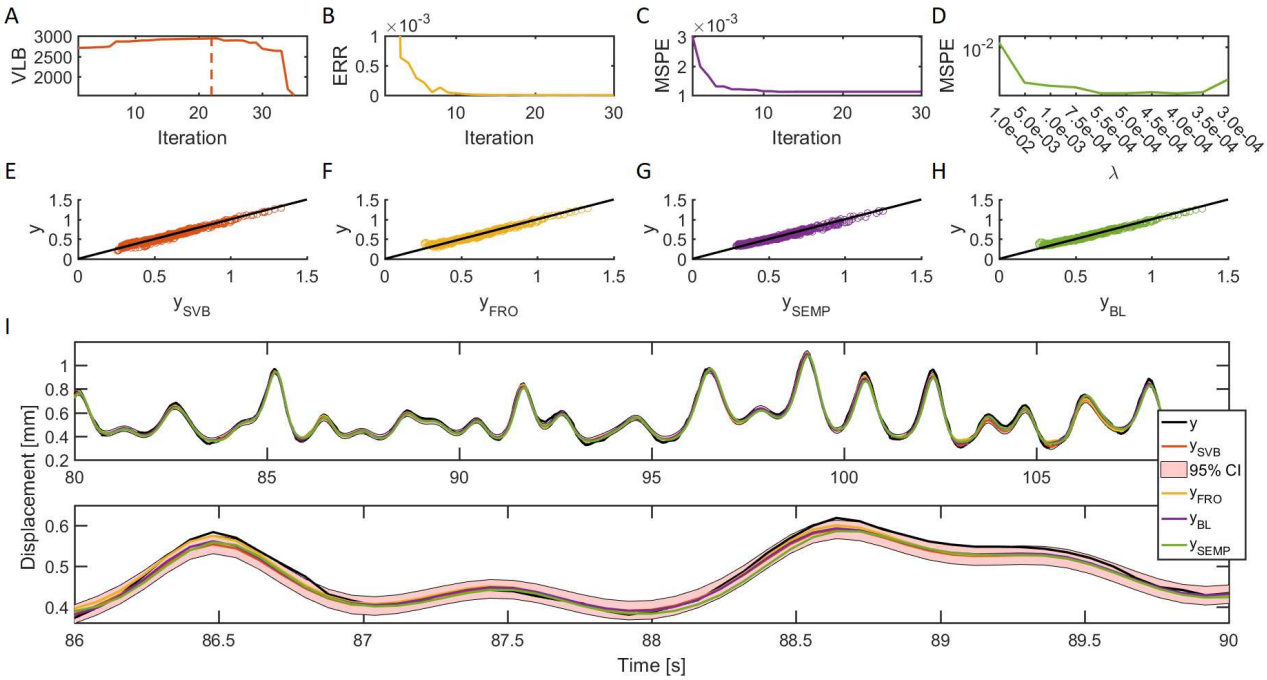

Fig. 9. Identification of the DEA using SVB-NARX, FRO, SEMP and BL. A) The SVB-NARX algorithm selects the optimal model structure at iteration 22 as marked by the dashed line: 15 model terms are selected. B) The ERR for FRO model selection: 12 model terms are selected. C) The MSPE for SEMP model selection: 11 terms are selected. D) The MSPE for the BL algorithm as a function of λ: 14 model terms are selected (corresponding to the minimum MSPE). E-H) The system output versus the model predicted output for the models identified by all four algorithms. I) The model predicted output over validation data for the model identified by SVB-NARX (Red) with 95% confidence intervals (Light red shaded area), FRO (Yellow), BL (Purple) and SEMP (Green) with the measured output (Black).

VII. Summary

In this paper a novel approach to nonlinear system identification of NARX models within a sparse variational Bayesian framework was introduced: The SVB-NARX algorithm. Term selection was driven by the inclusion of a sparsity inducing hyper-prior. We found that the algorithm was relatively fast compared to other nonlinear system identification methods and that it performed successfully even at low SNR levels (down to 2dB). The SVB-NARX algorithm was applied to a real world problem: identification of dielectric elastomer actuators. The algorithm produced an accurate model, and for the first time in nonlinear systems analysis we exploited the Bayesian nature of the SVB-NARX algorithm to numerically propagate the model parameter uncertainty into the nonlinear output frequency response functions.

Acknowledgment

The authors would like to thank the University of Sheffield and the EPRSC for a doctoral training award scholarship to W. Jacobs, which provided funding support for this work. We would like to thank the anonymous reviewers for their helpful comments, which led to a much improved version of this paper.

References

[1] I. J. Leontaritis and S. A. Billings, "Input-output parametric models for non-linear systems Part I: deterministic non-linear systems," *International Journal of Control*, vol. 41, no. 2, pp. 303–328, Feb. 1985.

[2] S. A. Billings, *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. Wiley, 2013.

[3] S. Chen and S. A. Billings, "Representations of non-linear systems: the NARMAX model," *International Journal of Control*, vol. 49, no. 3, pp. 1013–1032, Mar. 1989.

[4] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *International Journal of Control*, vol. 50, no. 5, pp. 1873–1896, Nov. 1989.

[5] S. A. Billings and K. M. Tsang, "Spectral analysis for non-linear systems, Part I: Parametric non-linear spectral analysis," *Mechanical Systems and Signal Processing*, vol. 3, no. 4, pp. 319–339, 1989.

[6] Z. Lang and S. Billings, "Energy transfer properties of non-linear systems in the frequency domain," *International Journal of Control*, vol. 78, no. 5, pp. 345–362, 2005.

[7] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky, "Nonlinear black-box modeling in system identification: a unified overview," *Automatica*, vol. 31, no. 12, pp. 1691–1724, 1995.

[8] K. Li, J.-X. Peng, and G. W. Irwin, "A fast nonlinear model identification method," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1211–1216, Aug 2005.

[9] L. Piroddi and W. Spinelli, "An identification algorithm for polynomial NARX models based on simulation error minimization," *International Journal of Control*, vol. 76, no. 17, pp. 1767–1781, Nov. 2003.

[10] L. Zhang and K. Li, "Forward and backward least angle regression for nonlinear system identification," *Automatica*, vol. 53, pp. 94–102, 2015.

[11] T. Baldacchino, S. R. Anderson, and V. Kadirkamanathan, "Structure detection and parameter estimation for NARX models in a unified EM framework," *Automatica*, vol. 48, no. 5, pp. 857–865, May 2012.

[12] S. L. Kukreja, J. Löfberg, and M. J. Brenner, "A least absolute shrinkage

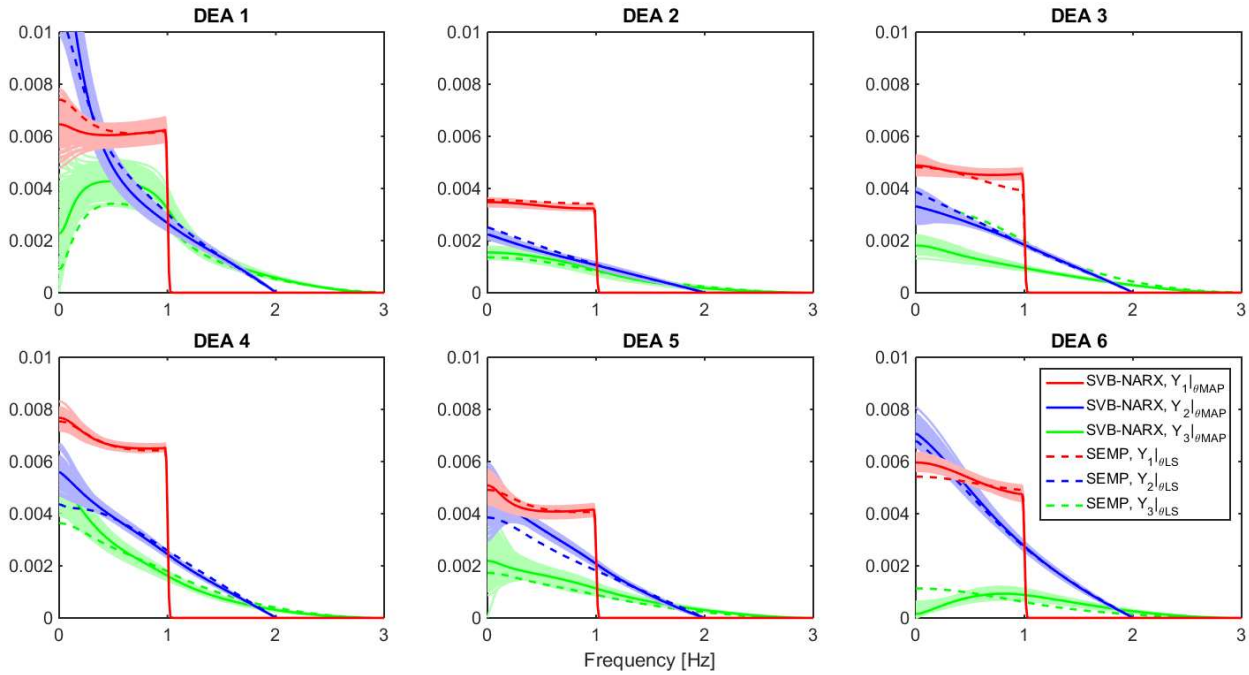

Fig. 10. Analysis of DEA model dynamics in the frequency domain using nonlinear output frequency response functions. The amplitudes of the first three nonlinear output spectra (Y_1, Y_2 and Y_3) are obtained for each DEA and from each modelling algorithm: SVB-NARX and SEMP. *Solid lines*: SVB-NARX, *maximum a posteriori* (MAP) estimate. *Dashed lines*: SEMP. *Shaded regions*: An uncertainty description over each amplitude of the nonlinear output spectra obtained by Monte Carlo sampling from the posterior parameter distribution of the SVB-NARX model and propagated into the frequency domain by evaluating the corresponding nonlinear output spectra (using 100 samples in each case).

and selection operator (lasso) for nonlinear system identification," *IFAC Proceedings Volumes*, vol. 39, no. 1, pp. 814–819, 2006.

[13] M. Bonin, V. Seghezza, and L. Piroddi, "NARX model selection based on simulation error minimisation and lasso," *IET Control Theory & Applications*, vol. 4, no. 7, pp. 1157–1168, 2010.

[14] V. Peterka, "Bayesian system identification," *Automatica*, vol. 17, no. 1, pp. 41–53, 1981.

[15] D. Mackay, "Probable networks and plausible predictions — a review of practical Bayesian methods for supervised neural networks," *Network: Computation in Neural Systems*, vol. 6, no. 3, pp. 469–505, Aug. 1995.

[16] B. Ninness and S. Henriksen, "Bayesian system identification via Markov chain Monte Carlo techniques," *Automatica*, vol. 46, no. 1, pp. 40–51, Jan. 2010.

[17] A. Gelman, *Bayesian Data Analysis*. Boca Raton, Fl: Chapman & Hall/CRC, 2004.

[18] W. Pan, Y. Yuan, J. Gonalves, and G. B. Stan, "A sparse Bayesian approach to the identification of nonlinear state-space systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 1, pp. 182–187, Jan 2016.

[19] P. L. Green and K. Worden, "Bayesian and Markov chain Monte Carlo methods for identifying nonlinear systems in the presence of uncertainty," *Phil. Trans. R. Soc. A*, vol. 373, p. 20140405, 2015.

[20] J. L. Beck, "Bayesian system identification based on probability logic," *Journal of International Association for Structural Control and Monitoring*, vol. 17, no. 7, pp. 825–847, 2010.

[21] T. Baldacchino, E. J. Cross, K. Worden, and J. Rowson, "Variational Bayesian mixture of experts models and sensitivity analysis for nonlinear dynamical systems," *Mechanical Systems and Signal Processing*, vol. 66-67, pp. 178–200, Jan. 2016.

[22] K. Krishnanathan, S. R. Anderson, S. A. Billings, and V. Kadirkamanathan, "Computational system identification of continuous-time nonlinear systems using approximate Bayesian computation," *International Journal of Systems Science*, vol. 47, pp. 3537–3544, 2016.

[23] F. Lindsten, T. B. Schön, and M. I. Jordan, "Bayesian semiparametric Wiener system identification," *Automatica*, vol. 49, no. 7, pp. 2053–2063, 2013.

[24] R. Frigola and C. E. Rasmussen, "Integrated pre-processing for Bayesian nonlinear system identification with Gaussian processes," in *52nd IEEE Conference on Decision and Control*, 2013, pp. 5371–5376.

[25] J. Kocijan, A. Girard, B. Banko, and R. Murray-Smith, "Dynamic systems identification with Gaussian processes," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 11, no. 4, pp. 411–424, 2005.

[26] T. Baldacchino, S. R. Anderson, and V. Kadirkamanathan, "Computational system identification for Bayesian NARMAX modelling," *Automatica*, vol. 49, no. 9, pp. 2641–2651, Sep. 2013.

[27] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, Eds., *Markov Chain Monte Carlo in Practice*. Boca Raton, Fla: Chapman & Hall, 1998.

[28] Z. Ghahramani and M. J. Beal, "Propagation algorithms for variational Bayesian learning," *Advances in Neural Information Processing Systems*, pp. 507–513, 2001.

[29] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine Learning*, vol. 37, no. 2, pp. 183–233, 1999.

[30] C. M. Bishop and M. E. Tipping, "Variational relevance vector machines," in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, 2000, pp. 46–53.

[31] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *The Journal of Machine Learning Research*, vol. 1, pp. 211–244, Sep. 2001.

[32] C. Lu, A. Devos, J. A. Suykens, C. Arús, and S. Van Huffel, "Bagging linear sparse Bayesian learning models for variable selection

in cancer diagnosis," *IEEE Transactions on Information Technology in Biomedicine*, vol. 11, no. 3, pp. 338–347, 2007.

[33] G.-M. Zhang, D. M. Harvey, and D. R. Braden, "Signal denoising and ultrasonic flaw detection via overcomplete and sparse representations," *The Journal of the Acoustical Society of America*, vol. 124, no. 5, pp. 2963–2972, 2008.

[34] H.-Q. Mu and K.-V. Yuen, "Novel sparse Bayesian learning and its application to ground motion pattern recognition," *Journal of Computing in Civil Engineering*, vol. 31, no. 5, 2017.

[35] W. R. Jacobs, T. Baldacchino, and S. R. Anderson, "Sparse Bayesian Identification of Polynomial NARX Models," *IFAC-PapersOnLine*, vol. 48, no. 28, pp. 172–177, 2015.

[36] A. O'Halloran, F. O'Malley, and P. McHugh, "A review on dielectric elastomer actuators, technology, applications, and challenges," *Journal of Applied Physics*, vol. 104, no. 7, p. 071101, 2008.

[37] W. R. Jacobs, E. D. Wilson, T. Assaf, J. Rossiter, T. J. Dodd, J. Porrill, and S. R. Anderson, "Control-focused, nonlinear and time-varying modelling of dielectric elastomer actuators with frequency response analysis," *Smart Materials and Structures*, vol. 24, no. 5, p. 055002, May 2015.

[38] E. D. Wilson, T. Assaf, M. J. Pearson, J. M. Rossiter, P. Dean, S. R. Anderson, and J. Porrill, "Biohybrid control of general linear systems using the adaptive filter model of cerebellum," *Frontiers in Neurorobotics*, vol. 9, no. 5, 2015.

[39] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[40] J. E. Griffin and P. J. Brown, "Inference with Normal-Gamma prior distributions in regression problems," *Bayesian Analysis*, vol. 5, no. 1, pp. 171–188, Mar. 2010.

[41] R. M. Neal, *Bayesian Learning for Neural Networks*. New York: Springer, 1996.

[42] D. P. Wipf and S. S. Nagarajan, "A new view of automatic relevance determination," in *Advances in neural information processing systems*, 2008, pp. 1625–1632.

[43] S. Sun, "A review of deterministic approximate inference techniques for Bayesian machine learning," *Neural Computing and Applications*, vol. 23, no. 7, pp. 2039–2050, 2013.

[44] M. J. Beal, "Variational algorithms for approximate Bayesian inference," Ph.D. dissertation, University of London, 2003.

[45] J. Drugowitsch, "Variational Bayesian inference for linear and logistic regression," *arXiv preprint*, vol. arXiv:1310.5438, 2013.

[46] D. M. Blei and M. I. Jordan, "Variational inference for Dirichlet process mixtures," *Bayesian Analysis*, vol. 1, no. 1, pp. 121–144, 2006.

[47] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, p. 1, 2010.

[48] Y. Bar-Cohen, Ed., *Electroactive polymer (EAP) actuators as artificial muscles: reality, potential, and challenges*, 2nd ed. Bellingham, Wash: SPIE Press, 2004.

[49] P. Brochu and Q. Pei, "Advances in Dielectric Elastomers for Actuators and Artificial Muscles," *Macromolecular Rapid Communications*, vol. 31, no. 1, pp. 10–36, Jan. 2010.

[50] K. Z. Mao and S. A. Billings, "Algorithms for minimal model structure detection in nonlinear dynamic system identification," *International Journal of Control*, vol. 68, no. 2, pp. 311–330, Jan. 1997.

[51] A. Sherstinsky and R. W. Picard, "On the efficiency of the orthogonal least squares training method for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 7, no. 1, pp. 195–200, 1996.

[52] G. Rizzello, D. Naso, A. York, and S. Seelecke, "Modeling, identification, and control of a dielectric electro-active polymer positioning system," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 632–643, 2015.

[53] Z.-Q. Lang and S. A. Billings, "Output frequency characteristics of nonlinear systems," *International Journal of Control*, vol. 64, no. 6, pp. 1049–1067, Aug. 1996.

[54] L. O. Chua and C.-Y. Ng, "Frequency domain analysis of nonlinear systems: general theory," *IEE Journal on Electronic Circuits and Systems*, vol. 3, no. 4, pp. 165–185, 1979.

William R. Jacobs received the M.Phys. degree in physics with mathematics from the University of Sheffield, Sheffield, UK, in 2010 and the Ph.D. degree in nonlinear system identification from the University of Sheffield in 2016.

He is currently working as a Research Associate in the Rolls Royce University Technology Centre, Department of Automatic Control and Systems Engineering at the University of Sheffield.

Tara Baldacchino received the B.Eng. degree in electrical engineering from the Faculty of Engineering, University of Malta, Malta, in 2006 and the M.Sc. degree in control systems and the Ph.D. degree from the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, U.K., in 2008 and 2011, respectively.

From 2011 to 2015 she was a Research Associate in the Dynamics Research Group, Department of Mechanical Engineering, University of Sheffield. She is currently a University Teacher in the Department of Automatic Control and Systems Engineering, University of Sheffield.

Tony Dodd received the B.Eng. degree in aerospace systems engineering from the University of Southampton, Southampton, UK, in 1994 and the Ph.D. degree in machine learning from the Department of Electronics and Computer Science, University of Southampton in 2000.

From 2002 to 2010, he was a Lecturer in the Department of Automatic Control and Systems Engineering at the University of Sheffield, Sheffield, UK, and from 2010 to 2015 was a Senior Lecturer at the University of Sheffield. He is currently a Professor of Autonomous Systems Engineering and Director of the MEng Engineering degree programme at the University of Sheffield.

Sean R. Anderson received the M.Eng. degree in control systems engineering from the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK, in 2001 and the Ph.D. degree in nonlinear system identification and predictive control from the University of Sheffield in 2005.

From 2005 to 2010, he was a Research Associate in the Neural Algorithms Research Group, University of Sheffield. From 2010 to 2012 he was a Research Associate in the Department of Automatic Control and Systems Engineering, University of Sheffield and has been a senior lecturer there since 2015.