

Equation Discovery Using an Efficient Variational Bayesian Approach with Spike-and-Slab Priors



Rajdip Nayek, Keith Worden, and Elizabeth J. Cross

Abstract A major challenge in the field of nonlinear system identification is the problem of selecting models that are not just good in prediction but also provide insight into the nature of the underlying dynamical system. In this study, a sparse Bayesian equation discovery approach is pursued to address the model selection problem, where it is treated as a Bayesian variable selection problem and solved via sparse linear regression using spike-and-slab priors. The spike-and-slab priors are considered the *gold standard* in Bayesian variable selection; however, Bayesian inference with spike-and-slab priors is not analytically tractable and often Markov chain Monte Carlo techniques are employed, which can be computationally expensive. This study proposes to use a computationally efficient variational Bayes algorithm for facilitating Bayesian equation discovery with spike-and-slab priors. To illustrate its performance, the algorithm has been applied to four systems of engineering interest, which include a baseline linear system, and systems with cubic stiffness, quadratic viscous damping, and Coulomb friction damping. The results of model selection and parameter estimation demonstrate the effectiveness and efficiency of the variational Bayesian inference compared to the conventional Markov-chain-Monte-Carlo-based Bayesian inference.

Keywords Equation discovery · Nonlinear system identification · Spike-and-slab prior · Sparse Bayesian learning · Variational Bayes

1 Introduction

Characterising the behaviour of nonlinear structural dynamical systems plays a key role in shaping the fundamental understanding of the underlying phenomena manifested by such systems. In forward analyses of these systems, generative models are derived from first principles, in the form of governing differential equations of motion, which are then utilised to analyse the behaviour of nonlinear structural dynamical systems and predict the possible future states. These governing differential equations of motion for structural dynamical systems can often be conveniently represented in the state-space form,

$$\dot{\mathbf{x}} = \mathcal{M}(\mathbf{x}) + \mathbf{u} \quad (1)$$

where \mathbf{x} is the state vector of system responses, $\dot{\mathbf{x}}$ is the time derivative of the state vector, \mathbf{u} is the external input force, and $\mathcal{M}(\mathbf{x})$ is the generative model embedding the equation of motion of the structure. When dealing with inverse problems, the form of \mathcal{M} is treated as unknown and one is tasked with positing a suitable generative model of \mathcal{M} that best describes the system dynamics, given some measurement data. This task constitutes the problem of model selection in nonlinear structural system identification. One commonly estimates a *black-box* approximation [1, 2] to \mathcal{M} , if the goal lies in only predicting the future states, given some past measurements. However, when the goal extends beyond prediction and the user aspires to select *interpretable* models—to understand the physics of the observed phenomenon—there arises a need to uncover the full parametric form of an underlying governing equation of motion.

In the pursuit of interpretable dynamical models, standard model selection procedures postulate a small set of candidate models based on expert intuition and use information-theoretic measures to select a best-fit model [3]. However, such

R. Nayek (✉) · K. Worden · E. J. Cross

Dynamics Research Group, Department of Mechanical Engineering, University of Sheffield, Sheffield, UK
e-mail: r.nayek@sheffield.ac.uk; k.worden@sheffield.ac.uk; e.j.cross@sheffield.ac.uk

procedures can become prohibitive when very little expert knowledge is available and the pool of candidate models is large. With the rapid development of data-driven modelling, there has been an emergence of alternative frameworks of model selection for nonlinear dynamical systems that depend more on data and much less on expert knowledge. An early effort in this setting include the data-driven symbolic regression [4, 5] that searches through a library (or *dictionary*) of simple and interpretable functional forms to identify the model structure or the governing equations of a nonlinear dynamical system. While this strategy works well for discovering interpretable physical models, its dependence on evolutionary optimisation for selecting the *relevant* variables from the dictionary makes it computationally expensive and unsuited to large-scale problems. In a more recent study [6], the discovery process was reformulated in terms of sparse linear regression, which makes the variable selection process amenable to solution using efficient sparsity-promoting algorithms, thus providing a computationally cheaper alternative.

This study follows a sparse linear regression approach to equation discovery of nonlinear structural dynamical systems. To describe the approach, consider a Single Degree-of-Freedom (SDOF) oscillator with equation of motion of the form,

$$m\ddot{q} + c\dot{q} + kq + g(q, \dot{q}) = u \quad (2)$$

where m , c , k are the mass, damping, and stiffness, g is an arbitrary nonlinear function of displacement q and velocity \dot{q} , and u is the input forcing function. The first-order state-space formulation for this system is,

$$\dot{x}_1 = x_2 \quad (3)$$

$$\dot{x}_2 = \frac{1}{m} (u - kx_1 - cx_2 - g(x_1, x_2)) \quad (4)$$

with $x_1 = q$ and $x_2 = \dot{q}$. Equation (3) can be ignored as it simply provides the definition of velocity; Eq. (4) captures the governing equation of the structure's motion. To uncover the underlying structure of the right hand side of equation (4), a dictionary of basis functions is constructed, containing several simple and interpretable functional forms of the system states and the input. The right hand side of equation (4) is then expressed as a weighted linear combination of the basis functions of the dictionary as follows:

$$\dot{x}_2 = \theta_1 f_1(x_1, x_2) + \theta_2 f_2(x_1, x_2) + \dots + \theta_l f_l(x_1, x_2) + \theta_{l+1} u \quad (5)$$

where, $\{f_1(x_1, x_2), \dots, f_l(x_1, x_2), u\}$ represent the collection of basis functions/variables and $\{\theta_1, \dots, \theta_l, \theta_{l+1}\}$ correspond to the set of weights. Given noisy observations of time-series measurements of the system $\{x_{1,j}, x_{2,j}, \dot{x}_{2,j}, u_j\}_{j=1}^N$, where j in the subscript indicates time point t_j , the above problem reduces to a linear regression problem,

$$\underbrace{\begin{bmatrix} \dot{x}_{2,1} \\ \dot{x}_{2,2} \\ \vdots \\ \dot{x}_{2,N} \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} f_1(x_{1,1}, x_{2,1}) & f_2(x_{1,1}, x_{2,1}) & \dots & f_l(x_{1,1}, x_{2,1}) & u_1 \\ f_1(x_{1,2}, x_{2,2}) & f_2(x_{1,2}, x_{2,2}) & \dots & f_l(x_{1,2}, x_{2,2}) & u_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ f_1(x_{1,N}, x_{2,N}) & f_2(x_{1,N}, x_{2,N}) & \dots & f_l(x_{1,N}, x_{2,N}) & u_N \end{bmatrix}}_{\mathbf{D}} \underbrace{\begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_l \\ \theta_{l+1} \end{bmatrix}}_{\boldsymbol{\theta}} + \underbrace{\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix}}_{\boldsymbol{\epsilon}} \quad (6)$$

where, \mathbf{y} is the time-series vector of observations of the derivatives of x_2 , \mathbf{D} is the dictionary matrix of basis variables, and $\boldsymbol{\epsilon}$ denotes the vector of residuals, taking into account model inadequacies and measurement errors. The task is now to select which basis variables from the dictionary are to be included in the final model. The equation discovery approach followed here assumes that only a few basis variables from the dictionary would actively contribute to the governing dynamics. As such, the solution of $\boldsymbol{\theta}$ would be *sparse*, i.e. would have only a few non-zero weights; hence, it is reasonable to seek sparse solutions of $\boldsymbol{\theta}$ in the above linear regression problem, as illustrated in Fig. 1.

There exists a variety of classical penalisation methods [7] such as lasso, elastic-net, etc. that promote sparsity by adding a penalty term to the ordinary least-squares objective. Another deterministic sparsity-promoting method is the sequential threshold least-squares, which runs a least-squares algorithm iteratively while eliminating the small weights at each iteration. This method underpins the Sparse Identification of Nonlinear Dynamics (SINDy) algorithm—introduced by Brunton et al. [6] in their pioneering work on equation discovery of nonlinear dynamical systems. Nevertheless, the performance of classical penalisation as well as sequential threshold least-squares often critically depends on a regularisation parameter that needs

$$\begin{array}{c}
 \left[\begin{array}{c} \color{green}{\rule{0.5pt}{1.5cm}} \\ \color{green}{\rule{0.5pt}{1.5cm}} \end{array} \right] = \left[\begin{array}{cccccc} \color{grey}{\rule{0.5pt}{1.5cm}} & \color{blue}{\rule{0.5pt}{1.5cm}} & \color{grey}{\rule{0.5pt}{1.5cm}} & \color{grey}{\rule{0.5pt}{1.5cm}} & \color{grey}{\rule{0.5pt}{1.5cm}} & \color{blue}{\rule{0.5pt}{1.5cm}} \\ \color{grey}{\rule{0.5pt}{1.5cm}} & \color{blue}{\rule{0.5pt}{1.5cm}} & \color{grey}{\rule{0.5pt}{1.5cm}} & \color{grey}{\rule{0.5pt}{1.5cm}} & \color{grey}{\rule{0.5pt}{1.5cm}} & \color{blue}{\rule{0.5pt}{1.5cm}} \\ \color{grey}{\rule{0.5pt}{1.5cm}} & \color{blue}{\rule{0.5pt}{1.5cm}} & \color{grey}{\rule{0.5pt}{1.5cm}} & \color{grey}{\rule{0.5pt}{1.5cm}} & \color{grey}{\rule{0.5pt}{1.5cm}} & \color{blue}{\rule{0.5pt}{1.5cm}} \\ \color{grey}{\rule{0.5pt}{1.5cm}} & \color{blue}{\rule{0.5pt}{1.5cm}} & \color{grey}{\rule{0.5pt}{1.5cm}} & \color{grey}{\rule{0.5pt}{1.5cm}} & \color{grey}{\rule{0.5pt}{1.5cm}} & \color{blue}{\rule{0.5pt}{1.5cm}} \\ \color{grey}{\rule{0.5pt}{1.5cm}} & \color{blue}{\rule{0.5pt}{1.5cm}} & \color{grey}{\rule{0.5pt}{1.5cm}} & \color{grey}{\rule{0.5pt}{1.5cm}} & \color{grey}{\rule{0.5pt}{1.5cm}} & \color{blue}{\rule{0.5pt}{1.5cm}} \end{array} \right] \cdots \left[\begin{array}{c} \color{blue}{\rule{0.5pt}{1.5cm}} \\ \color{blue}{\rule{0.5pt}{1.5cm}} \\ \color{blue}{\rule{0.5pt}{1.5cm}} \\ \vdots \\ \color{blue}{\rule{0.5pt}{1.5cm}} \end{array} \right] + \left[\begin{array}{c} \color{brown}{\rule{0.5pt}{1.5cm}} \\ \color{brown}{\rule{0.5pt}{1.5cm}} \end{array} \right] \\
 \mathbf{y} \qquad \qquad \qquad \mathbf{D} \qquad \qquad \qquad \boldsymbol{\theta} \qquad \qquad \qquad \boldsymbol{\epsilon}
 \end{array}$$

Fig. 1 Sparse linear regression for selection of relevant basis variables (shown in blue)

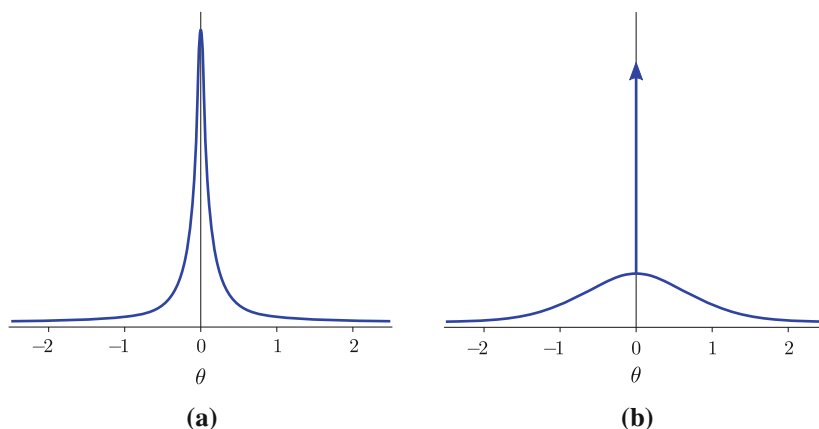


Fig. 2 Probability density functions of (a) the Student's- t prior, and (b) the spike-and-slab prior (with the spike displayed by an arrow pointing upwards)

tuning via cross-validation. A sparse Bayesian framework [8, 9] provides a more appealing alternative in this context; apart from the usual advantage of uncertainty quantification, it offers natural penalisation via sparsity-promoting prior distributions and allows simultaneous estimation of model and regularisation parameters, thereby avoiding the need for cross-validation.

In a Bayesian framework, sparse solutions to Eq. (6) are obtained by placing sparsity-promoting (or *shrinkage*) priors on the unknown weight vector $\boldsymbol{\theta}$. The densities of such priors feature a strong peak at zero and heavy tails: the peak at zero enforces most of the values to be (near) zero while heavy tails allow a few non-zero values. Examples of such priors include Laplace [10], Student's- t [8], Horseshoe [11], and spike-and-slab [12]. Previous studies [13, 14] on sparse Bayesian model discovery approaches employed the Relevance Vector Machine (RVM) [15]—a popular implementation of the Student's- t prior. Despite its remarkable computational efficiency, an issue with the RVM is that it often results in false discoveries [16]. This issue with the RVM arises due to the use of the Student's- t prior, and is undesirable, as false discoveries introduce more complexity and hinder the interpretability of the estimated model.

Compared to a Student's- t prior, a spike-and-slab (SS) prior is capable of producing sparser solutions and reducing false discoveries. An SS prior represents a mixture of two distributions—a point mass at zero (the *spike*) for small weights, and a diffused density (the *slab*) for the large weights—and is considered as the gold standard in Bayesian variable selection (BVS) [17]. It is capable of shrinking the small weights to exactly zero; hence, it has the potential to induce a greater degree of sparsity among the weights compared to a Student's- t prior. Figure 2 provides a visual depiction of the densities of the Student's- t and SS priors.

A disadvantage of the SS prior is that the Bayesian inference can be computationally demanding: the posterior computation with the SS prior is analytically intractable and one typically employs Markov Chain Monte Carlo (MCMC)-based approaches—most commonly Gibbs sampling—to draw samples from the posterior distribution. Employing a Gibbs sampler, it was shown that equation discovery with SS priors leads to more interpretable models [18] compared to the RVM, although the runtime of the Gibbs sampler could be prohibitive for large systems. A faster alternative to MCMC-based approaches is to use a variational Bayesian approach. A few studies [19, 20] have previously proposed Variational Bayes

(VB) algorithms with SS priors to reduce the computational burden in BVS. The implementation in [19] assumed complete independence of the variational distributions of the model parameters, and it led to severe underestimation of the posterior covariance of the model parameters. On the other hand, [20] relaxed the independence assumption to a greater extent and was able to better control the underestimation of posterior covariance.

This paper presents a novel application of VB to Bayesian equation discovery of dynamical systems with SS priors. A VB algorithm is derived for posterior inference with SS priors, and its performance has been compared with an MCMC-based sampling approach. Furthermore, the efficiency of the proposed approach has been compared with two other algorithms: (a) RVM (that uses a Student's- t prior) and (b) the deterministic SINDy algorithm [6]. The rest of the paper is structured as follows: Sect. 2 describes the SS prior model for linear regression, followed by Sect. 3 describing the variational Bayesian approach for BVS. Section 4 presents numerical demonstrations of equation discovery for four SDOF oscillators: a linear oscillator, a Duffing oscillator with cubic nonlinearity, an oscillator with quadratic viscous damping, and one with Coulomb damping. Finally, Sect. 5 summarises the conclusions of the paper.

2 Linear Regression Model with Spike-and-Slab Prior

Consider once again the linear regression problem in Eq. (6), rewritten here in a compact matrix-vector form,

$$\mathbf{y} = \mathbf{D}\boldsymbol{\theta} + \boldsymbol{\epsilon} \quad (7)$$

where, \mathbf{y} is a $N \times 1$ vector of state derivatives (also referred to as the target vector), \mathbf{D} is a $N \times P$ dictionary matrix,¹ $\boldsymbol{\theta}$ is the $P \times 1$ weight vector, and $\boldsymbol{\epsilon}$ is the $N \times 1$ residual error vector. The residual error $\boldsymbol{\epsilon}$ is modelled as a vector of independent Gaussian noise with diagonal covariance matrix $\sigma^2 \mathbf{I}_N$ ($\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_N)$). With a known dictionary \mathbf{D} , the likelihood function can be written as,

$$\mathbf{y} | \boldsymbol{\theta}, \sigma^2 \sim \mathcal{N}(\mathbf{D}\boldsymbol{\theta}, \sigma^2 \mathbf{I}_N) \quad (8)$$

To perform BVS with the SS prior, the linear regression problem is considered as part of a hierarchical model. The key feature of the hierarchical model is that each component of $\boldsymbol{\theta}$ is assigned an independent SS prior, defined as follows:

$$\theta_i | z_i, v_s \sim (1 - z_i)\delta(\theta_i) + z_i \mathcal{N}(0, \sigma^2 v_s) \quad (9)$$

The spike part of the prior is modelled by a Dirac delta function at zero [12], while the slab part is modelled by a continuous zero-mean Gaussian density with a variance $\sigma^2 v_s$. Here v_s is the slab variance and it is multiplied with σ^2 so that the prior naturally scales with the scale of the outcome, that is, the results would not depend on the measurement units of \mathbf{y} . Whether a weight θ_i belongs to the spike or the slab is determined by an indicator variable z_i : $z_i = 0$ implies $\theta_i = 0$, and $z_i = 1$ implies $\theta_i \sim \mathcal{N}(0, v_s)$. In other words, $z_i = 0$ or $z_i = 1$ determines the inclusion or exclusion of the i th basis variable in the model. Furthermore, each indicator variable z_i is assigned an independent Bernoulli prior, controlled by a common parameter p_0 ,

$$z_i | p_0 \sim \text{Bern}(p_0) \quad (10)$$

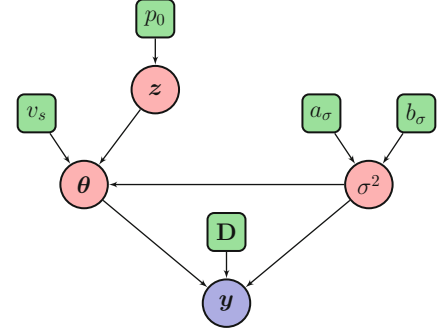
Equation (10) implies that the selection of a basis variable is independent of the inclusion of any other basis variables from the dictionary \mathbf{D} . The parameter $p_0 \in (0, 1)$ represents the probability of $z_i = 1$ and controls the degree of sparsity imposed by the SS prior. Together, the weight vector $\boldsymbol{\theta}$ and the vector of indicator variables \mathbf{z} constitute the main parameters of the SS prior model for linear regression. Finally, inverse-Gamma priors are assigned to the error variance σ^2 ,

$$\sigma^2 \sim \mathcal{IG}(a_\sigma, b_\sigma) \quad (11)$$

Note that v_s , p_0 , a_σ , b_σ appearing in Eqs. (9)–(11) are treated as deterministic parameters, controlling the shape of the respective priors. The hierarchical SS model for linear regression is illustrated in Fig. 3.

¹Note the number of columns in the dictionary has been redefined as $P = l + 1$.

Fig. 3 Graphical structure of the hierarchical spike-and-slab model for linear regression; the variables in circles represent random variables, while those in squares represent deterministic parameters



3 Variational Bayesian Inference for Variable Selection

The information required for BVS using the SS prior is derived from the joint posterior distribution of the model parameters $p(\boldsymbol{\theta}, \mathbf{z}, \sigma^2 | \mathbf{y})$, which can be computed using Bayes' rule in the form,

$$p(\boldsymbol{\theta}, \mathbf{z}, \sigma^2 | \mathbf{y}) = \frac{p(\mathbf{y} | \boldsymbol{\theta}, \sigma^2) p(\boldsymbol{\theta} | \mathbf{z}, \sigma^2) p(\mathbf{z}) p(\sigma^2)}{p(\mathbf{y})} \quad (12)$$

Here $p(\mathbf{y} | \boldsymbol{\theta}, \sigma^2)$ is the likelihood, $p(\boldsymbol{\theta} | \mathbf{z}, \sigma^2) p(\mathbf{z})$ is the SS prior over \mathbf{z} and $\boldsymbol{\theta}$, $p(\sigma^2)$ is the prior over measurement noise variance, and $p(\mathbf{y})$ is the normalising constant. Unfortunately, the posterior in Eq. (12) cannot be computed analytically, and using MCMC-based sampling methods can be computationally expensive. Therefore, in this section, a VB methodology is pursued for approximating the joint posterior distribution $p(\boldsymbol{\theta}, \mathbf{z}, \sigma^2 | \mathbf{y})$ by a simpler distribution. However, the presence of the Dirac delta function in the SS prior makes the derivation of the VB algorithm difficult, and hence the linear regression model with SS prior is reparameterised in a form that is more amenable to VB inference methods. Specifically, the form of the SS prior is rewritten as [20],

$$\begin{aligned} \mathbf{y} | \boldsymbol{\theta}, \mathbf{z}, \sigma^2 &\sim \mathcal{N}(\mathbf{D}\boldsymbol{\Gamma}\boldsymbol{\theta}, \sigma^2 \mathbf{I}_N), \quad \sigma^2 \sim \mathcal{IG}(a_\sigma, b_\sigma), \\ \theta_i &\sim \mathcal{N}(0, \sigma^2 v_s), \quad \text{and } z_i \sim \text{Bern}(p_0), \quad i = 1, \dots, P \end{aligned} \quad (13)$$

where the newly introduced term represents $\boldsymbol{\Gamma} = \text{diag}(z_1, \dots, z_P)$.

3.1 Variational Bayes

In VB inference, a factorised distribution $q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2)$ is chosen from a predetermined family of simple distributions \mathcal{Q} and then the distributional parameters are optimised such that the Kullback–Leibler (KL) divergence between the true posterior $p(\boldsymbol{\theta}, \mathbf{z}, \sigma^2 | \mathbf{y})$ and the optimised variational approximation $q^*(\boldsymbol{\theta}, \mathbf{z}, \sigma^2)$ is a minimum. Put formally,

$$q^*(\boldsymbol{\theta}, \mathbf{z}, \sigma^2) = \arg \min_{q \in \mathcal{Q}} \text{KL}[q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2) || p(\boldsymbol{\theta}, \mathbf{z}, \sigma^2 | \mathbf{y})] = \arg \min_{q \in \mathcal{Q}} \mathbb{E}_{q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2)} \left[\ln \left(\frac{q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2)}{p(\boldsymbol{\theta}, \mathbf{z}, \sigma^2 | \mathbf{y})} \right) \right] \quad (14)$$

where $\mathbb{E}_{q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2)}[\cdot]$ denotes the expectation with respect to the variational distribution $q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2)$. The expansion of the KL divergence term in Eq. (14) leads to an expression for the evidence lower bound (ELBO), which plays a key role in assessing the convergence of VB methods.

$$\text{KL}[q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2) || p(\boldsymbol{\theta}, \mathbf{z}, \sigma^2 | \mathbf{y})] = \mathbb{E}_{q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2)} \left[\ln \left(\frac{q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2)}{p(\boldsymbol{\theta}, \mathbf{z}, \sigma^2 | \mathbf{y})} \right) \right]$$

$$\begin{aligned}
&= \mathbb{E}_{q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2)} \left[\ln q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2) \right] \\
&\quad - \mathbb{E}_{q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2)} \left[\ln \left(p(\mathbf{y} | \boldsymbol{\theta}, \mathbf{z}, \sigma^2) p(\boldsymbol{\theta}, \mathbf{z}, \sigma^2) \right) \right] + \ln p(\mathbf{y}) \\
&= \mathbb{E}_{q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2)} \left[\ln \left(\frac{q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2)}{p(\boldsymbol{\theta}, \mathbf{z}, \sigma^2)} \right) \right] - \mathbb{E}_{q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2)} \left[\ln p(\mathbf{y} | \boldsymbol{\theta}, \mathbf{z}, \sigma^2) \right] + \ln p(\mathbf{y}) \\
&= \underbrace{\text{KL} \left[q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2) \parallel p(\boldsymbol{\theta}, \mathbf{z}, \sigma^2) \right]}_{-\text{ELBO}} - \mathbb{E}_{q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2)} \left[\ln p(\mathbf{y} | \boldsymbol{\theta}, \mathbf{z}, \sigma^2) \right] + \ln p(\mathbf{y}) \\
&= \ln p(\mathbf{y}) - \text{ELBO} \geq 0
\end{aligned} \tag{15}$$

Since the KL divergence is non-negative and $\ln p(\mathbf{y})$ is constant with respect to the variational distribution $q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2)$, the ELBO can be seen as the lower bound to $\ln p(\mathbf{y})$, and hence, minimising $\text{KL} \left[q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2) \parallel p(\boldsymbol{\theta}, \mathbf{z}, \sigma^2 | \mathbf{y}) \right]$ is equivalent to maximising the ELBO, thus,

$$q^*(\boldsymbol{\theta}, \mathbf{z}, \sigma^2) = \arg \max_{q \in \mathcal{Q}} \underbrace{\mathbb{E}_{q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2)} \left[\ln p(\mathbf{y} | \boldsymbol{\theta}, \mathbf{z}, \sigma^2) \right] - \text{KL} \left[q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2) \parallel p(\boldsymbol{\theta}, \mathbf{z}, \sigma^2) \right]}_{\text{ELBO}} \tag{16}$$

In this study, $q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2)$ has been chosen to have the following factorised form [20]:

$$q(\boldsymbol{\theta}, \mathbf{z}, \sigma^2) = q(\boldsymbol{\theta}) q(\sigma^2) \prod_{i=1}^P q(z_i) \tag{17}$$

and the corresponding individual variational distributions are selected as,

$$\begin{aligned}
q(\boldsymbol{\theta}) &= \mathcal{N}(\boldsymbol{\mu}^q, \boldsymbol{\Sigma}^q), \\
q(\sigma^2) &= \mathcal{IG}(a_\sigma^q, b_\sigma^q), \\
q(z_i) &= \text{Bern}(w_i^q), \text{ for } i = 1, \dots, P
\end{aligned} \tag{18}$$

Here, $\{\boldsymbol{\mu}^q, \boldsymbol{\Sigma}^q, a_\sigma^q, b_\sigma^q, w_i^q\}$ represents a set of deterministic variational parameters whose values need to be optimised to draw the approximate variational distribution closer to the true posterior distribution, in the sense of KL divergence (see Eq. (14)). The optimal choice of the set of variational parameters that maximise the ELBO in Eq. (16) satisfies the following relations [21, 22]:

$$\begin{aligned}
q^*(\boldsymbol{\theta}) &\propto \mathbb{E}_{q(\mathbf{z})q(\sigma^2)} \left[\ln p(\mathbf{y}, \boldsymbol{\theta}, \mathbf{z}, \sigma^2) \right] \\
q^*(\mathbf{z}) &\propto \mathbb{E}_{q(\boldsymbol{\theta})q(\sigma^2)} \left[\ln p(\mathbf{y}, \boldsymbol{\theta}, \mathbf{z}, \sigma^2) \right] \\
q^*(\sigma^2) &\propto \mathbb{E}_{q(\boldsymbol{\theta})q(\mathbf{z})} \left[\ln p(\mathbf{y}, \boldsymbol{\theta}, \mathbf{z}, \sigma^2) \right]
\end{aligned} \tag{19}$$

and on solving the above, the expression for the variational parameters can be obtained as

$$\boldsymbol{\Sigma}^q = \left[\tau \left((\mathbf{D}^T \mathbf{D}) \odot \boldsymbol{\Omega} + v_s^{-1} \mathbf{I}_P \right) \right]^{-1}, \tag{20}$$

$$\boldsymbol{\mu}^q = \tau \boldsymbol{\Sigma}^q \mathbf{W}^q \mathbf{D}^T \mathbf{y}, \tag{21}$$

$$a_\sigma^q = a_\sigma + 0.5N + 0.5P, \tag{22}$$

$$b_\sigma^q = b_\sigma + 0.5 \left[\mathbf{y}^T \mathbf{y} - 2\mathbf{y}^T \mathbf{D} \mathbf{W}^q \boldsymbol{\mu}^q + \text{tr} \left\{ \left((\mathbf{D}^T \mathbf{D}) \odot \boldsymbol{\Omega} + v_s^{-1} \mathbf{I}_P \right) \left(\boldsymbol{\mu}^q \boldsymbol{\mu}^{qT} + \boldsymbol{\Sigma}^q \right) \right\} \right], \tag{23}$$

$$\tau = a_\sigma^q / b_\sigma^q, \quad (24)$$

$$\eta_i = \text{logit}(p_0) - 0.5\tau \left((\mu_i^q)^2 + \Sigma_{i,i}^q \right) \mathbf{f}_i^T \mathbf{f}_i + \tau \mathbf{f}_i^T \left[\mathbf{y} \mu_i^q - \mathbf{D}_{-i} \mathbf{W}_{-i}^q \left(\boldsymbol{\mu}_{-i}^q \mu_i^q + \boldsymbol{\Sigma}_{-i,i}^q \right) \right], \quad (25)$$

$$w_i^q = \text{expit}(\eta_i) \quad (26)$$

In the above expressions, $\text{logit}(A) = \ln(A) - \ln(1 - A)$, $\text{expit}(A) = \text{logit}^{-1}(A) = \exp(A)/(1 + \exp(A))$, $\mathbf{w}^q = [w_1^q, \dots, w_P^q]^T$, $\mathbf{W}^q = \text{diag}(\mathbf{w}^q)$, $\boldsymbol{\Omega} = \mathbf{w}^q \mathbf{w}^{qT} + \mathbf{W}^q (\mathbf{I}_P - \mathbf{W}^q)$, and the symbol \odot denotes the element-wise product between two matrices. Additionally, \mathbf{f}_i denotes the i th column of \mathbf{D} , and \mathbf{D}_{-i} represents the dictionary matrix with the i th column removed. As the variational parameters do not have explicit solutions and their update expressions are dependent upon each other, an iterative coordinate-wise updating procedure is followed for optimising them, in which they are first initialised and then cyclically updated conditional on the updates of other parameters.

3.2 Initialisation and Convergence

To implement the VB algorithm, one needs to set the values of the prior parameters $\{v_s, p_0, a_\sigma, b_\sigma\}$ and initialise the variational parameters $\{\mathbf{w}^q, \tau\}$. The prior parameters are set as follows: a slab variance $v_s = 10$, noise prior parameters $a_\sigma = 10^{-4}$, $b_\sigma = 10^{-4}$, and a small probability $p_0 = 0.1$ to favour the selection of simpler models. The VB algorithm is found to be quite sensitive to the initial choice of the variational parameter \mathbf{w}^q which represents the vector of inclusion probabilities of the basis variables. To provide a good initial guess, a grid-search procedure was suggested in [20]; however, such an initialisation procedure is time-consuming and deemed inconvenient here. In this work, \mathbf{w}^q is initialised by setting it to the model diagnostic parameter vector $\boldsymbol{\gamma}$ output by the RVM algorithm. The RVM diagnostic parameter, $\gamma_i \in (0, 1)$, can be interpreted as a probabilistic measure of how important the i th basis variable is in explaining the target vector \mathbf{y} [8]; as such, it relates well with the idea of variable inclusion probabilities. Moreover, a RVM run is cheap, and therefore getting a good initial point for $\mathbf{w}^{(0)}$ takes very little time. Lastly, the initial value of τ , which represents the expected precision of the noise σ^2 , is set to $\tau^{(0)} = 1000$.

The VB algorithm iteratively and monotonically maximises the ELBO and converges to a local maximum of the bound. Starting with the initial variational parameters, the VB iterations are continued until the relative increase in the ELBO between two successive VB iterations is very small, that is, when

$$\text{ELBO}^{(t)} - \text{ELBO}^{(t-1)} < \rho \quad (27)$$

the iterations are terminated. Here, $\rho = 10^{-6}$ is considered. The value of ELBO, at each iteration t , is computed using the simplified expression:

$$\begin{aligned} \text{ELBO}^{(t)} = & 0.5P - 0.5N \ln(2\pi) - 0.5P \ln(v_s) + a_\sigma \ln(b_\sigma) - \ln \Gamma(a_\sigma) + \ln \Gamma(a_\sigma^{(t)}) - a_\sigma^{(t)} \ln \Gamma(b_\sigma^{(t)}) \\ & + 0.5 \ln |\boldsymbol{\Sigma}^{(t)}| + \sum_{i=1}^P \left[w_i^{(t)} \ln \left(\frac{p_0}{w_i^{(t)}} \right) + (1 - w_i^{(t)}) \ln \left(\frac{1 - p_0}{1 - w_i^{(t)}} \right) \right] \end{aligned} \quad (28)$$

where $\Gamma(\cdot)$ is the Gamma function, and $a_\sigma^{(t)}$, $b_\sigma^{(t)}$, $\boldsymbol{\mu}^{(t)}$, $\boldsymbol{\Sigma}^{(t)}$, $\mathbf{w}^{(t)}$ denote the variational parameters at the t th iteration, dropping the ‘ q ’ superscript. Upon convergence, the variational parameters from the final step are denoted by a_σ^* , b_σ^* , $\boldsymbol{\mu}^*$, $\boldsymbol{\Sigma}^*$, \mathbf{w}^* .

3.3 Bayesian Variable Selection

With a total of P basis variables in the dictionary, there are 2^P possible models, where a model is indexed by which of the z_i s equal one and which equal zero. For example, the model with zero basis variables has $\mathbf{z} = \mathbf{0}$, whereas the model that includes all basis variables has $\mathbf{z} = \mathbf{1}$. The *relevant* basis variables to be included in the final model are selected based on the marginal *posterior inclusion probabilities* (PIP), $p(z_i = 1 | \mathbf{y})$. Specifically, the basis variables whose corresponding PIPs

are greater than half, i.e. $p(z_i = 1 | \mathbf{y}) > 0.5$, $i = 1, \dots, P$, are included in the final estimated model. The corresponding model is popularly known as the median probability model and is considered optimal for prediction [23]. In VB inference, the estimated w_i^* s can be interpreted as an approximation to the posterior probability of $p(z_i = 1 | \mathbf{y})$. Therefore, post inference, the set of basis variables which correspond to $w_i^* > 0.5$ are included in the estimated model. Furthermore, the estimated mean and covariance of the vector of weights θ , denoted by $\hat{\boldsymbol{\mu}}_\theta$ and $\hat{\boldsymbol{\Sigma}}_\theta$, are respectively populated with values of $\boldsymbol{\mu}^*$ and $\boldsymbol{\Sigma}^*$ at the indices corresponding to the selected set of basis variables, and the rest of the entries of $\hat{\boldsymbol{\mu}}_\theta$ and $\hat{\boldsymbol{\Sigma}}_\theta$ are set to zero. Thereafter, predictions with the estimated model can be made using the mean and the covariance of the weights, as shown below,

$$\boldsymbol{\mu}_{\mathbf{y}^*} = \mathbf{D}^* \hat{\boldsymbol{\mu}}_\theta \quad (29)$$

$$\mathbf{V}_{\mathbf{y}^*} = \mathbf{D}^* \hat{\boldsymbol{\Sigma}}_\theta \mathbf{D}^{*T} + (a_\sigma^*/b_\sigma^*)^{-1} \mathbf{I}_{N^*} \quad (30)$$

where \mathbf{D}^* is the $N^* \times P$ test dictionary, defined at a set of N^* previously unseen test data points, $\boldsymbol{\mu}_{\mathbf{y}^*}$ is the $N^* \times 1$ predicted mean of the target vector, and $\mathbf{V}_{\mathbf{y}^*}$ is the predicted covariance associated with the target vector.

4 Numerical Studies

This section presents numerical studies for exploring the performance of the SS-prior-based VB inference for model/equation discovery. Four SDOF oscillators of the form expressed by Eq. (4) are considered, each having different forms of the nonlinear term $g(x_1, x_2)$, as enumerated in Table 1.

The four systems are simulated using the following parameters:

- The parameters of the linear system are taken as: $m = 1$, $c = 2$, and $k = 10^3$.
- The three other nonlinear systems use the same values of parameters for the underlying linear part and only differ in the additional nonlinear term $g(x_1, x_2)$. The respective forms and the values of $g(x_1, x_2)$ are provided in Table 1.
- The systems are excited using a band-limited Gaussian excitation with standard deviation of 50 and passband of 0 to 100 Hz.
- The displacement x_1 and velocity x_2 for each system are simulated using a fixed-step fourth-order Runge–Kutta numerical integration scheme, with a sampling rate of 1 kHz.
- The acceleration \dot{x}_2 is obtained using Eq. (4).

Noisy observations of the displacement x_1 , the velocity x_2 , the acceleration \dot{x}_2 , and the input force u are assumed, and the noise is modelled as sequences of zero-mean Gaussian white noise with a standard deviation equal to 5% of the standard deviation of the simulated quantities.

The equation discovery approach commences with a dictionary of candidate basis variables. In this work, the dictionary \mathbf{D} is composed of 36 basis variables, where each basis variable represents a certain function of the noisy measurements x_1 , x_2 ,

$$\mathbf{D} = \left\{ P^1(\mathbf{x}), \dots, P^6(\mathbf{x}), \text{sgn}(\mathbf{x}), |\mathbf{x}|, \mathbf{x} \otimes |\mathbf{x}|, u \right\} \quad (31)$$

where, $P^\gamma(\mathbf{x})$ denotes the polynomial expansion of order γ of the sum of state vectors $(x_1 + x_2)^\gamma$. The dictionary consists of basis variables that are terms from polynomial orders up to $\gamma = 6$ and certain other terms. The term $\text{sgn}(\mathbf{x})$ represents the signum functions of states, i.e., $\text{sgn}(x_1)$ and $\text{sgn}(x_2)$. Similarly, $|\mathbf{x}|$ denotes the absolute functions of states, i.e., $|x_1|$ and $|x_2|$. The tensor product term $\mathbf{x} \otimes |\mathbf{x}|$ represents the following set of functions: $x_1|x_1|$, $x_1|x_2|$, $x_2|x_1|$, and $x_2|x_2|$. Finally, the measured input force u is included directly in the dictionary. Note that the total number of models that can be formed by

Table 1 Simulation cases

System	Name	$g(x_1, x_2)$	
1	Linear	0	
2	Duffing	$k_3 x_1^3$	$k_3 = 10^5$
3	Quadratic viscous damping	$c_2 x_2 x_2 $	$c_2 = 2$
4	Coulomb friction damping	$c_F \text{sgn}(x_2)$	$c_F = 1$

combinatorial selection of all 36 basis variables in the dictionary is 2^{36} , and it grows exponentially as the number of basis variables increases.

The constructed dictionary in Eq. (31) is often ill-conditioned, due to a combined effect of large-scale difference among the basis variables and the presence of strong linear correlation between certain basis variables. Appropriate scaling of the columns (i.e. the basis variables) helps to reduce the difference in scales and improve the conditioning of the dictionary. For the purpose of Bayesian inference, the columns of the training dictionary are standardised (i.e. they are centred and scaled to have zero mean and unit standard deviation) and the training target vector is detrended to have zero mean. Formally put, the transformed pair of dictionary and the target vector $(\mathbf{D}_s, \mathbf{y}_s)$ input to the Bayesian inference algorithm in the training phase has the form,

$$\begin{aligned}\mathbf{D}^s &= (\mathbf{D} - \mathbf{1}\boldsymbol{\mu}_D) \mathbf{S}_D^{-1} \\ \mathbf{y}^s &= \mathbf{y} - \mathbf{1}\mu_y\end{aligned}\quad (32)$$

where, $\mathbf{1}$ denotes a column vector of ones, $\boldsymbol{\mu}_D$ is a row vector of the column-wise means of \mathbf{D} , \mathbf{S}_D is a diagonal matrix of the column-wise standard deviations of \mathbf{D} , and μ_y is the mean of the target training vector \mathbf{y} . Note that this modification implies that, post Bayesian inference, the estimated mean and covariance of the scaled weights $\boldsymbol{\theta}^s$, denoted by $\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}^s}$ and $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}^s}$, have to be transformed back to the original space using the relations: $\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}} = \mathbf{S}_D^{-1} \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}^s}$ and $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}} = \mathbf{S}_D^{-1} \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}^s} \mathbf{S}_D^{-1}$.

As the VB methodology is based on approximating the posterior, it is worthwhile to compare the results of the VB inference to that of the MCMC-based Bayesian inference that can yield arbitrarily accurate posteriors. In this study, the MCMC-based Bayesian inference is conducted with a Gibbs sampler, with sampling steps as follows:

$$\begin{aligned}\boldsymbol{\theta}_r \mid \mathbf{y}, \sigma^2 &\sim \mathcal{N}\left(\mathbf{a}_r, \sigma^2 \mathbf{A}_r\right), \text{ with } \mathbf{A}_r = \left(\mathbf{D}_r^T \mathbf{D}_r + v_s^{-1} \mathbf{I}_r\right)^{-1}, \quad \mathbf{a}_r = \mathbf{A}_r \mathbf{D}_r^T \mathbf{y} \\ \sigma^2 \mid \mathbf{y}, \boldsymbol{\theta}_z &\sim \mathcal{IG}\left(a_\sigma + 0.5N, b_\sigma + 0.5\left(\mathbf{y}^T \mathbf{y} - \mathbf{a}_r^T \mathbf{A}_r^{-1} \mathbf{a}_r\right)\right) \\ z_i \mid \mathbf{y} &\sim \text{Bern}(\xi_i), \text{ with } \xi_i = \frac{p_0}{p_0 + R_i(1 - p_0)}, \quad R_i = \frac{p(\mathbf{y} \mid z_i = 0, \mathbf{z}_{-i})}{p(\mathbf{y} \mid z_i = 1, \mathbf{z}_{-i})}, \quad i = 1, \dots, P\end{aligned}\quad (33)$$

where $\boldsymbol{\theta}_r$ is a $r \times 1$ vector consisting of components of $\boldsymbol{\theta}$ which belong to the slab (i.e. corresponding to $z_i = 1$), \mathbf{D}_r is a $N \times r$ matrix that includes only those columns of \mathbf{D} whose corresponding components of \mathbf{z} are unity, and the calculation of R_i uses the marginal likelihood of \mathbf{y} given \mathbf{z}

$$p(\mathbf{y} \mid \mathbf{z}) = \frac{\Gamma(a_\sigma + 0.5N) (b_\sigma)^{a_\sigma}}{(2\pi)^{N/2} (v_s)^{r/2} \Gamma(a_\sigma)} \frac{\left| \left(\mathbf{D}_r^T \mathbf{D}_r + v_s^{-1} \mathbf{I}_r\right)^{-1} \right|^{1/2}}{\left(b_\sigma + 0.5 \mathbf{y}^T \left(\mathbf{I}_N - \mathbf{D}_r \left(\mathbf{D}_r^T \mathbf{D}_r + v_s^{-1} \mathbf{I}_r\right)^{-1} \mathbf{D}_r^T \right) \mathbf{y} \right)^{(a_\sigma + 0.5N)}}\quad (34)$$

The Gibbs sampler is run with four chains, each chain having a total of 5000 samples and the first 1000 samples are discarded as burn-in. The following values were set for the deterministic prior parameters: $a_\sigma = 10^{-4}$, $b_\sigma = 10^{-4}$, $p_0 = 0.1$, and $v_s = 10$. The measurement noise variance $\sigma^{2(0)}$ for each chain was initialised with slightly perturbed values about a nominal mean value—set equal to the residual variance from an ordinary least-squares regression. To facilitate faster convergence of the Gibbs sampler to a good solution, the initial vector of binary latent variables $\mathbf{z}^{(0)}$ was computed by starting off with z_1, \dots, z_P set to zero and then activating the components of \mathbf{z} that reduce the mean-squared error on the training target vector, until an integer number ($\approx p_0 P$) of components of \mathbf{z} is equal to one. The multivariate potential scale reduction factor \hat{R} [24], which estimates the potential decrease in the between-chain variance with respect to the within-chain variance, was applied to assess the convergence of the generated samples of $\boldsymbol{\theta}$. A value of $\hat{R} < 1.1$ was adopted to decide if convergence had been reached.

Figure 4 demonstrates the procedure of basis variable selection applied to the four SDOF oscillator systems using PIP. In the case of MCMC, the PIP is calculated for the i th basis variable by averaging over the posterior samples of z_i , $i = 1, \dots, 36$. For VB, the PIP for the i th basis variable is approximated by w_i^* . Basis variables with higher values of PIP imply greater relevance, and as mentioned in Sect. 3.3, only those basis variables are selected whose PIPs are greater than 0.5 (shown by a dotted line in red). It can be seen that the estimated models for all the four systems are able to select the true basis variables out of the pool of 36 basis variables. In all cases, the computed PIPs corresponding to the true basis variables are close to one, which indicates very strong selection probability; however, that may not always be the case. For example, in

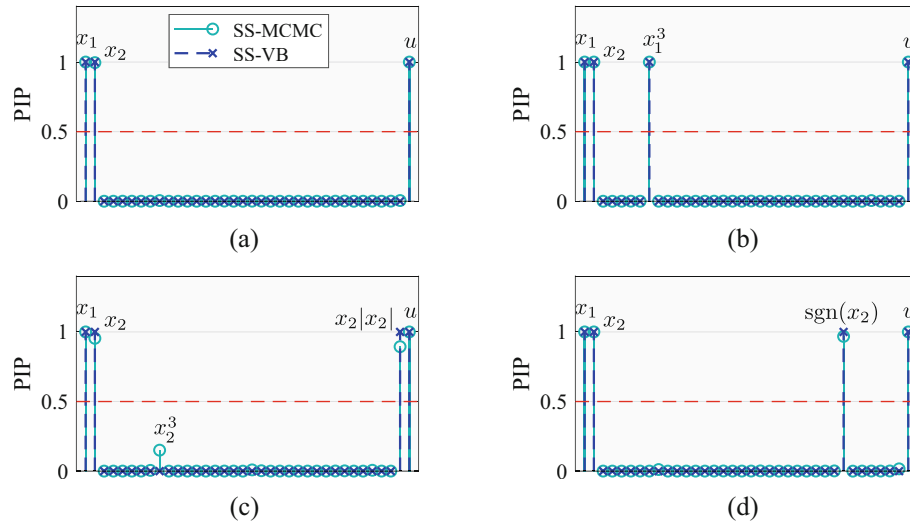


Fig. 4 Illustration of Bayesian variable selection using posterior inclusion probability (PIP), computed by $p(z_i = 1 | \mathbf{y})$ in case of MCMC (SS-MCMC) and approximated by w_i^* in case of VB (SS-VB). (a) Linear. (b) Duffing. (c) Viscous damping. (d) Friction damping

the quadratic viscous damping case (see Fig. 4c), the SS-MCMC algorithm selects the true basis variable $x_2|x_2|$ with a PIP of 0.9, while it discards a correlated basis function x_2^3 with a PIP of 0.15. Such situations can arise when there exist strong correlations between certain basis variables, causing the Bayesian algorithm to get confused as to which among the set of correlated basis variables should be selected.

For the sake of comparison, Fig. 5 plots the pairwise joint posteriors of the model parameters obtained from MCMC samples and those obtained from VB (using means and 95% confidence ellipses). Posteriors using SS priors often tend to be multimodal, as is demonstrated by the scatterplots from MCMC samples. Since VB uses a single approximating distribution, it is impossible to capture the multiple modes of the true posterior. Instead, VB methods approximate the true posterior around its maximum *a posteriori* (MAP) estimate. This feature is clearly indicated in the above plots where the mean of the VB distribution (labelled by \times) is found to coincide with the MCMC MAP estimate (depicted by the regions where the MCMC samples are most concentrated). Alongside the VB means, 95% confidence ellipses—representing the joint 95% confidence bounds—of the model parameters are also plotted (shown by yellow lines). It is noted that the 95% confidence ellipses from VB are always smaller than the support of posterior samples from MCMC, which signify the underestimation of posterior covariances from VB—a well known issue with VB methods [25].

4.1 Performance Comparison Using Monte Carlo Simulations

In this section, Monte Carlo simulations are used to compare the performance of equation discovery by the VB algorithm with that of MCMC. In addition, results from the popular sparse Bayesian RVM algorithm [8] and the deterministic SINDy algorithm [6] are also included for comparison. The RVM is implemented following the algorithm outlined in [15]. As mentioned in Sect. 1, SINDy implements a sequential threshold least-squares to promote sparsity and requires selection of the value of a regularisation parameter via cross-validation. A naive sweep over a sequence of regularisation parameter values was performed, and a value was selected for which the corresponding test set prediction error was a minimum.

Thousand different realisations for each of the four systems, as summarised in Table 1, were considered. The realisations were created by introducing random perturbations of 0.1κ to the nominal values of the parameters c, k, k_3, c_2, c_F , such that the new realisations have parameters $\bar{c} = (1 + 0.1\kappa)c$, $\bar{k} = (1 + 0.1\kappa)k$, and so on. The variable κ was sampled from a standard Gaussian distribution $\mathcal{N}(0, 1)$ for each realisation. Note that the nominal values of parameters are the ones that were used in the previous numerical study. In order to assess the performance, the following performance metrics are defined:

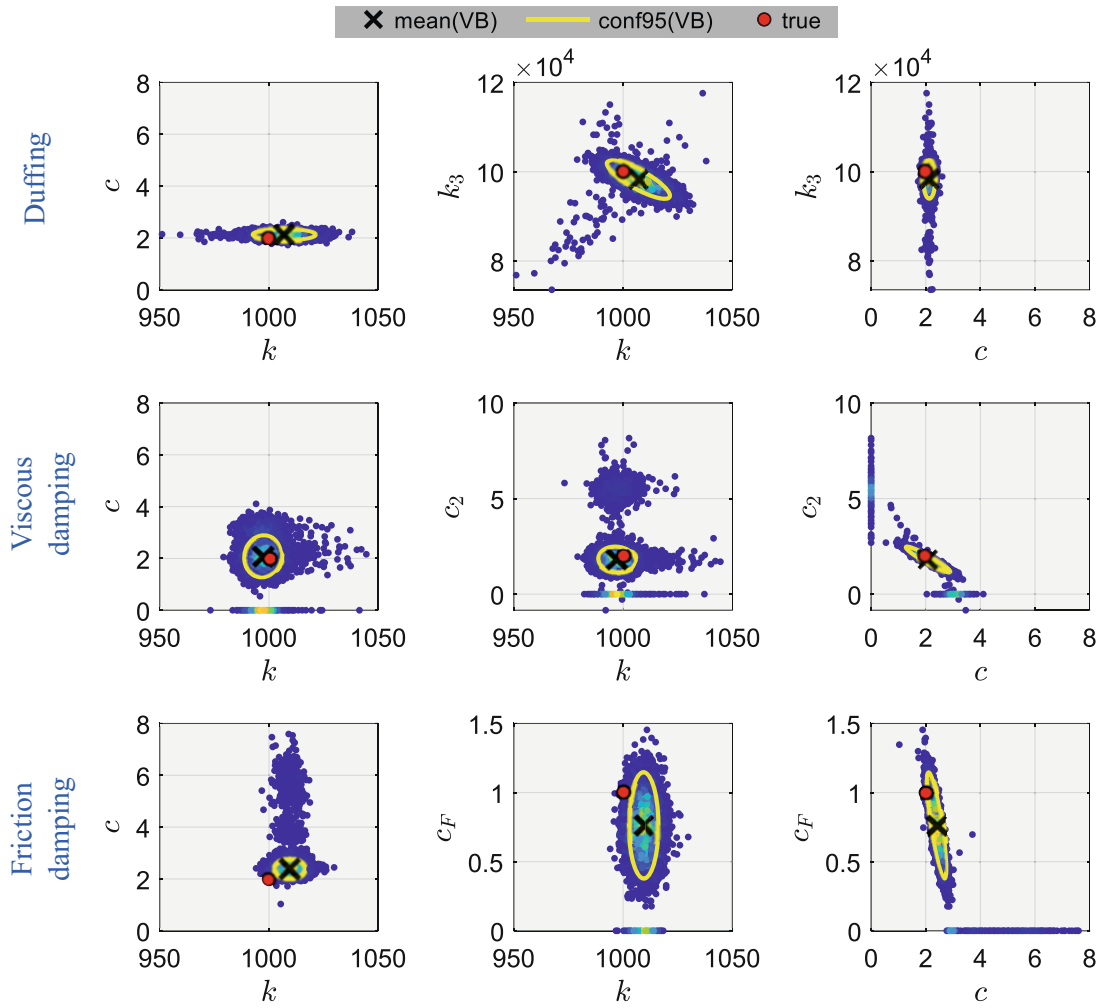


Fig. 5 Comparison of pairwise joint posteriors of the model parameters obtained by MCMC (shown as scatter-plot with posterior samples) and VB (depicted by mean and 95% confidence ellipse) for the three nonlinear oscillators: Duffing, quadratic viscous damping, and Coulomb friction damping; the red circle shows the true parameter values

- Weight estimation error, $e_\theta = \frac{\|\hat{\theta} - \theta\|_2}{\|\theta\|_2}$, where $\hat{\theta}$ is the estimate of the true (unscaled) weight vector θ corresponding to the unscaled dictionary. Similarly, one can also define a scaled weight estimation error, $e_{\theta^s} = \frac{\|S_D(\hat{\theta} - \theta)\|_2}{\|S_D\theta\|_2}$. In the case of SS priors, $\hat{\theta}$ is set to the estimated mean of the weights, whereas in the case of RVM, it is obtained as the MAP estimate.
- Test set prediction error, $e_p = \frac{\|\mathbf{y}^* - \mathbf{D}^*\hat{\theta}\|_2}{\|\mathbf{y}^*\|_2}$, where \mathbf{y}^* is the test set of responses, \mathbf{D}^* is the unscaled test dictionary, and $\hat{\theta}$ is the estimate of weight vector obtained using training data. 2000 data points were used for training and another 2000 data points for testing.
- False discovery rate (FDR), defined as the ratio of the number of false basis variables selected to the total number of basis variables included in the estimated model.
- Exact model selection indicator, $\hat{\mathcal{M}} = \mathcal{M}$, is a variable that takes value one when the estimated model $\hat{\mathcal{M}}$ has the exact same basis variables as the true model \mathcal{M} , and is zero otherwise.
- Superset model selection indicator, $\hat{\mathcal{M}} \supset \mathcal{M}$, is a variable that takes value one when the estimated model $\hat{\mathcal{M}}$ includes all the basis variables present in the true model \mathcal{M} , and is zero otherwise.

The above performance metrics are evaluated for each of the 1000 different realisations for all four systems, and the averages of the results are reported in Table 2.

Table 2 Comparison of results from SINDy, RVM, SS-MCMC, and SS-VB algorithms, averaged over 1000 realisations; bold numbers highlight the best performing metric

Type	Alg.	e_{θ^s}	e_{θ}	e_p	FDR	$\hat{\mathcal{M}} = \mathcal{M}$	$\hat{\mathcal{M}} \supset \mathcal{M}$
Linear	SINDy	0.168	$\approx 3 \times 10^4$	0.302	0.580	0.190	0.778
	RVM	0.010	502.261	0.097	0.576	0.005	0.999
	SS-MCMC	0.004	5.863	0.072	0.002	0.992	1.000
	SS-VB	0.004	0.164	0.071	0.003	0.991	0.997
Duffing	SINDy	0.253	609.134	0.168	0.558	0.113	0.742
	RVM	0.070	47.377	0.091	0.560	0.001	0.976
	SS-MCMC	0.026	4.116	0.078	0.037	0.846	0.977
	SS-VB	0.034	4.631	0.076	0.041	0.845	0.971
Quadratic damping	SINDy	0.147	$\approx 4 \times 10^5$	0.097	0.488	0.106	0.520
	RVM	0.017	1546.542	0.073	0.497	0.003	0.931
	SS-MCMC	0.017	0.004	0.072	0.030	0.859	0.865
	SS-VB	0.012	0.004	0.071	0.019	0.886	0.892
Coulomb damping	SINDy	0.216	$\approx 4 \times 10^5$	0.257	0.592	0.021	0.371
	RVM	0.013	1034.780	0.092	0.496	0.003	0.993
	SS-MCMC	0.011	0.004	0.071	0.018	0.847	0.850
	SS-VB	0.009	0.026	0.070	0.003	0.762	0.765

Table 3 Average computational runtimes for a single run of SINDy, RVM, SS-MCMC (single chain), and SS-VB

SINDy	RVM	SS-MCMC	SS-VB
3.65s	0.03s	34.23s	0.33s

It is noted that for all four systems, the model selection and parameter estimation performance using SS priors are superior to that using RVM and SINDy, with the deterministic SINDy algorithm performing worst among the four algorithms. Overall, it can be inferred that SINDy and the RVM very rarely find the exact true model and will likely include many false discoveries. The occurrence of false discoveries can be regarded a major deterrent in equation discovery, where selecting the correct set of basis variables is crucial for drawing scientific conclusions from the estimated model. Both implementations of SS priors do remarkably well, particularly in reducing the false discovery rate and in increasing the exact model selection rate. Moreover, the results from VB are seen to closely match those from MCMC, even outperforming them on many occasions. Such competitive performance demonstrated by VB makes it efficient compared to MCMC, as the VB takes much less run time than MCMC. The average runtimes of the four algorithms on a PC with 64-bit Windows 10 with 128GB RAM and Intel Xeon E5-2698 (version 4) CPU at 2.20 GHz are enumerated in Table 3. The RVM is the cheapest in terms of computational time, while the SS-MCMC is the most expensive. The SINDy algorithm run with a known regularisation parameter is often faster than the RVM; however, when accounting for the time to find the appropriate regularisation parameter, the SINDy can be much slower than RVM, as is shown in Table 3. The time reported for SS-VB is the combined time of running both the RVM (required for initialising the VB) and the VB algorithm, and it is apparent that the SS-VB is much faster than SS-MCMC, taking only 1/100th of the time used by SS-MCMC.

5 Conclusions

This paper investigates the use of an efficient variational Bayesian approach for performing equation discovery of nonlinear structural dynamic systems. Using a dictionary composed of interpretable functions, the task of Bayesian equation discovery is turned into a BVS problem, and solved using SS priors, which have the potential to derive more parsimonious and interpretable models of the underlying structural dynamics. However, MCMC-based Bayesian inference with SS priors is computationally demanding and can be prohibitive when the size of the dictionary grows and the number of observations is large. Unlike the MCMC-based approaches, the VB methodology approximates the true posterior with a simple distribution; it converts the Bayesian inference into a distribution-fitting optimisation problem and solves the optimisation at a much lower computational cost.

Using a series of numerical simulations, it has been demonstrated that the SS-VB algorithm correctly identifies the presence and type of various nonlinearities such as a cubic stiffness, a quadratic viscous damping, and a Coulomb friction

damping. Most importantly, the SS-VB algorithm yields performance on par with the MCMC-based Gibbs sampler at a much lower computational cost, making the VB approach very efficient in Bayesian equation discovery of nonlinear dynamical systems.

Acknowledgments This work has been funded by the UK Engineering and Physical Sciences Research Council (EPSRC), via the Autonomous Inspection in Manufacturing and Re-manufacturing (AIMaReM) grant EP/N018427/1. Support for K. Worden from the EPSRC via grant reference number EP/J016942/1 and for E.J. Cross via grant number EP/S001565/1 is also gratefully acknowledged.

References

1. Ljung, L.: Nonlinear Black-box modeling in system identification. Linköping University (1995)
2. Kocijan, J., Girard, A., Banko, B., Murray-Smith, R.: Dynamic systems identification with Gaussian processes. *Math. Comput. Model. Dyn. Syst.* **11**(4), 411–424 (2005)
3. Nakamura, T., Judd, K., Mees, A.I., Small, M.: A comparative study of information criteria for model selection. *Int. J. Bifurcation Chaos* **16**(08), 2153–2175 (2006)
4. Bongard, J., Lipson, H.: Automated reverse engineering of nonlinear dynamical systems. *Proc. Natl. Acad. Sci.* **104**(24), 9943–9948 (2007)
5. Schmidt, M., Lipson, H.: Distilling free-form natural laws from experimental data. *Science* **324** (5923), 81–85 (2009)
6. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl. Acad. Sci.* **113**(15), 3932–3937 (2016)
7. Hastie, T., Tibshirani, R., Wainwright, M.: *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, Boca Raton (2015)
8. Tipping, M.E.: Sparse Bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.* **1**, 211–244 (2001)
9. Wipf, D.P., Rao, B.D.: Sparse Bayesian learning for basis selection. *IEEE Trans. Signal Process.* **52**(8), 2153–2164 (2004)
10. Seeger, M.W.: Bayesian inference and optimal design for the sparse linear model. *J. Mach. Learn. Res.* **9**, 759–813 (2008)
11. Carvalho, C.M., Polson, N.G., Scott, J.G.: Handling sparsity via the Horseshoe. In: *Artificial Intelligence and Statistics*, pp. 73–80 (2009)
12. Mitchell, T.J., Beauchamp, J.J.: Bayesian variable selection in linear regression. *J. Am. Stat. Assoc.* **83**(404), 1023–1032 (1988)
13. Zhang, S., Lin, G.: Robust data-driven discovery of governing physical laws with error bars. *Proc. R. Soc. A Math. Phys. Eng. Sci.* **474**(2217), 20180305 (2018)
14. Fuentes, R., Dervilis, N., Worden, K., Cross, E.J.: Efficient parameter identification and model selection in nonlinear dynamical systems via sparse Bayesian learning. *J. Phys. Conf. Ser.* **1264**, 012050 (2019). IOP Publishing
15. Tipping, M.E., Faul, A.C.: Fast marginal likelihood maximisation for sparse Bayesian models. In: *Proceedings of the Ninth AISTATS Conference*, pp. 1–13 (2003)
16. Nayek, R., Worden, K., Cross, E.J., Fuentes, R.: A sparse Bayesian approach to model structure selection and parameter estimation of dynamical systems using spike-and-slab priors. In: *Proceedings of the International Conference on Noise and Vibration Engineering - ISMA2020 and International Conference on Uncertainty in Structural Dynamics - USD2020* (2020)
17. Polson, N.G., Scott, J.G.: Shrink globally, act locally: sparse Bayesian regularization and prediction. *Bayesian Stat.* **9**(501–538), 105 (2010)
18. Nayek, R., Fuentes, R., Worden, K., Cross, E.J.: On spike-and-slab priors for Bayesian equation discovery of nonlinear dynamical systems via sparse linear regression (2020). Preprint, arXiv:2012.01937
19. Carbonetto, P., Stephens, M.: Scalable variational inference for Bayesian variable selection in regression, and its accuracy in genetic association studies. *Bayesian Anal.* **7**(1), 73–108 (2012)
20. Ormerod, J.T., You, C., Müller, S.: A variational Bayes approach to variable selection. *Electron. J. Stat.* **11**(2), 3549–3594 (2017)
21. Bishop, C.M.: *Pattern Recognition and Machine Learning*. Springer, New York (2006)
22. Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: a review for statisticians. *J. Am. Stat. Assoc.* **112**(518), 859–877 (2017)
23. Barbieri, M.M., Berger, J.O.: Optimal predictive model selection. *Ann. Stat.* **32**(3), 870–897 (2004)
24. Brooks, S.P., Gelman, A.: General methods for monitoring convergence of iterative simulations. *J. Comput. Graph. Stat.* **7**(4), 434–455 (1998)
25. Giordano, R., Broderick, T., Jordan, M.I.: Covariances, robustness and variational Bayes. *J. Mach. Learn. Res.* **19**(1), 1981–2029 (2018)