Article:
Nayek, R. orcid.org/0000-0003-4277-8382, Fuentes, R., Worden, K. et al. (1 more author) (2021) On spike-and-slab priors for Bayesian equation discovery of nonlinear dynamical systems via sparse linear regression. Mechanical Systems and Signal Processing, 161. 107986. ISSN 0888-3270

https://doi.org/10.1016/j.ymssp.2021.107986

On spike-and-slab priors for Bayesian equation discovery of nonlinear dynamical systems via sparse linear regression

R. Nayek, R. Fuentes, K. Worden, E.J. Cross

Dynamics Research Group, Department of Mechanical Engineering, University of Sheffield
Mapping Street, Sheffield S1 3JD, UK

Abstract

This paper presents the use of spike-and-slab (SS) priors for discovering governing differential equations of motion of nonlinear structural dynamic systems. The problem of discovering governing equations is cast as that of selecting *relevant* variables from a predetermined dictionary of basis functions and solved via sparse Bayesian linear regression. The SS priors, which belong to a class of discrete-mixture priors and are known for their strong sparsifying (or *shrinkage*) properties, are employed to induce sparse solutions and select relevant variables. Three different variants of SS priors are explored for performing Bayesian equation discovery. As the posteriors with SS priors are analytically intractable, a Markov chain Monte Carlo (MCMC)-based Gibbs sampler is employed for drawing posterior samples of the model parameters; the posterior samples are used for basis function selection and parameter estimation in equation discovery. The proposed 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 damping. The results demonstrate the effectiveness of the SS priors in identifying the presence and type of nonlinearity in the system. Additionally, comparisons with the Sparse Bayesian (SBL) – that uses a Student's-t prior – indicate that the SS priors can achieve better model selection consistency, reduce false discoveries, and derive models that have superior predictive accuracy. Finally, the Silverbox experimental benchmark is used to validate the proposed methodology.

Keywords: Equation discovery, nonlinear system identification, sparse Bayesian learning, spike-and-slab priors, model selection, Bayesian variable selection

1. Introduction

Spurred on by the rapid increase in computational power and growing rates of data collection, recent years have seen a booming interest in discovering governing differential equations of motion of nonlinear dynamical systems from time-series data [1–18]. Governing differential equations of motion are ordinary or partial differential equations that characterise system behaviour and provide an understanding of the physics of the underlying phenomena. Historically, such equations have been derived based on first principles and some prior knowledge of the nature of the system; once known, they can be used for further analysis, prediction and control of the system. In structural dynamics, the governing equations of motion can often

be represented as a state-space model (SSM) of the form [1],

$$\dot{\boldsymbol{x}} = \mathcal{M}(\boldsymbol{x}) + \boldsymbol{u} \tag{1}$$

where $\boldsymbol{x}$ is the state vector of system responses, $\dot{\boldsymbol{x}}$ is the time derivative of the state vector, $\mathcal{M}$ is the function of states $\boldsymbol{x}$ embedding the equation of motion of the structure, and $\boldsymbol{u}$ is the vector of external input forces that are assumed to enter linearly in Eq. (1). This form in equation is quite common in structural dynamics, although a more general form of Eq. (1) would replace $\mathcal{M}$ with a function of both the state and the input vectors. Due to the ubiquitous presence of nonlinearity in modern structural systems, the structural equations of motion, represented by the model $\mathcal{M}$, typically includes a number of nonlinear terms in $\boldsymbol{x}$. However, in most situations, the true form of $\mathcal{M}$ is unknown, and hence, there arises a need to recover the model $\mathcal{M}$, i.e. the underlying equations of motion. Formally, the task of recovering $\mathcal{M}$ involves solving two sub-problems: *model selection*, which aims to identify a suitable form of $\mathcal{M}$, and *parameter estimation*, which determines the unknown parameters of the chosen form of $\mathcal{M}$. Individually, both these problems have received significant attention in the remit of structural dynamics as well as in the broader context of nonlinear system identification, and the interested reader can find excellent review papers [19, 20] on nonlinear system identification.

When the goal is to discover a parametric form of $\mathcal{M}$, the identified model needs to satisfy two essential attributes: (a) good *prediction power*, by the virtue of which it is able to predict future observations effectively without suffering from over-fitting, and (b) *interpretability*, so that the model includes only a few features (or predictors) that exhibit the strongest effect, thus providing a better understanding of the underlying process. Typically, when prediction is the only aim, the actual choice of the features is of less interest, as long as the fit to the data is good. However, when the aim is also to understand the physical phenomena generating the responses – which is the case here – there is a need to search for the real but unknown relationship between the responses and the features. For a good understanding of the relationship, it is important to select only the relevant features i.e., the features that matter.

Traditional model selection procedures work by postulating a small set of *interpretable* models – chosen based on expert intuition and domain knowledge. The "best" model is selected as the one that achieves a desired balance of model complexity and goodness-of-fit, judged by some information-theoretic criteria such as AIC [21], BIC [22], etc. Nonetheless, these traditional procedures can become prohibitive when prior knowledge is limited and the number of candidate models is large (in the order of hundreds or greater). With the rapid advancement in data-driven modelling in the last two decades, there has been an emergence of alternative frameworks of model selection that rely less on expert knowledge and more on data. An early effort towards data-driven modelling for equation discovery was symbolic regression [1, 2], which searches through a library (or *dictionary*) of simple and interpretable basis functions to identify the parametric form of the governing equations of a nonlinear dynamical system. While this approach 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 [3], the model discovery process was reformulated in terms of sparse linear regression, which

[1] Note the explicit time dependence of the states $\boldsymbol{x}(t)$ and inputs $\boldsymbol{u}(t)$ has been suppressed to simplify notation.

makes the basis function selection process amenable to solution using efficient sparsity-promoting algorithms, thus providing a computationally-cheaper alternative. Since then, the sparse regression approach for data-driven equation discovery of differential equations has been further developed in many studies. Examples include sparse identification of biological networks with rational basis functions [4], model selection using an integral formulation of the differential equation to reduce noise effects [5], model selection for dynamical selection combining sparse regression and information criteria [6], extension of sparse identification to nonlinear systems with control [7], discovery of coordinates for sparse representation of governing equations [8], extracting structured differential equations with under-sampled data [9], sparse learning of stochastic dynamical equations [10], model selection for nonlinear dynamical systems with switching behaviour [11], recovery of differential equations from short impulse response time-series data [12], identification of parametric partial differential equations [13–15]. There are also studies that proposed *black-box* approaches using deep neural networks [16–18] for equation discovery of differential equations; however, they are mostly useful for forecasting and do not provide explicit equations for interpretation.

The work presented in this paper adopts the sparse-regression-based parametric-equation-discovery approach for recovering the governing equation of motion of a structural dynamic system. In this approach, it is assumed that the function $\mathcal{M}$ consists of only a few terms, making it sparse in the space of possible functions. The assumption is generally true for many systems of engineering interest, as their governing physics is often simple and interpretable. To recover $\mathcal{M}$, the idea is to first express $\mathcal{M}$ as a weighted linear combination of a large number of simple intepretable basis functions (or basis functions), and then apply algorithms to select a *relevant* subset of variables that best explains the measurements. As such, in this approach, the model selection problem is turned into a basis function selection problem.

To elaborate on the procedure, consider the example of a Single Degree-of-Freedom (SDOF) oscillator with equation of motion of the form,

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

where m, c, k are the mass, damping, and stiffness, g is an arbitrary nonlinear function of displacement q and velocity $\dot{q}$; $\ddot{q}$ is the acceleration, and u is the input forcing function. An SSM for this system can be written as,

$$\dot{x}_1 = x_2 \tag{3}$$

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

with $x_1 = q$ and $x_2 = \dot{q}$. Eq. (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 Eq. (4), a large dictionary of basis functions $f_1(x_1, x_2), f_2(x_1, x_2), \ldots, f_l(x_1, x_2)$ is constructed, containing several functional forms such as polynomial terms, trigonometric terms, etc. The left-hand side, which represents acceleration $\dot{x}_2$, is then expressed on the right as a weighted linear combination of the basis functions of the dictionary,

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

where $\{\theta_1, \theta_2, \cdots, \theta_l, \theta_{l+1}\}$ are the associated weights. Note that the input is also added to the dictionary to identify its corresponding weight. Given noisy time-series measurements $\{x_{1,j}, x_{2,j}, \dot{x}_{2,j}, u_j\}_{j=1}^{N}$, where j

in the subscript indicates time point t_j, the above problem reduces to a linear regression problem,

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

which can be written in a compact matrix-vector notation as,

$$\boldsymbol{y} = \mathbf{D}\boldsymbol{\theta} + \boldsymbol{\epsilon} \tag{7}$$

Here, $\boldsymbol{y} \in \mathbb{R}^{N\times 1}$ is a vector of observations of acceleration, $\mathbf{D} \in \mathbb{R}^{N\times P}$ is a dictionary[2] matrix composed using states (i.e. displacement and velocity) and input force, $\boldsymbol{\theta} \in \mathbb{R}^{P\times 1}$ is the vector of basis weights and $\boldsymbol{\epsilon} \in \mathbb{R}^{N\times 1}$ is the residual error vector taking into account model inadequacies and measurement errors. The task is now to select which basis functions from the dictionary are to be included in the final estimated model $\hat{\mathcal{M}}$. As only a few basis functions from the dictionary are assumed to contribute actively to the governing dynamics, the solution of $\boldsymbol{\theta}$ would be sparse, i.e. should have only a few weights that are significantly different than zero; hence, it is reasonable to seek sparse solutions of $\boldsymbol{\theta}$ in the above linear regression problem, as illustrated in Figure 1.

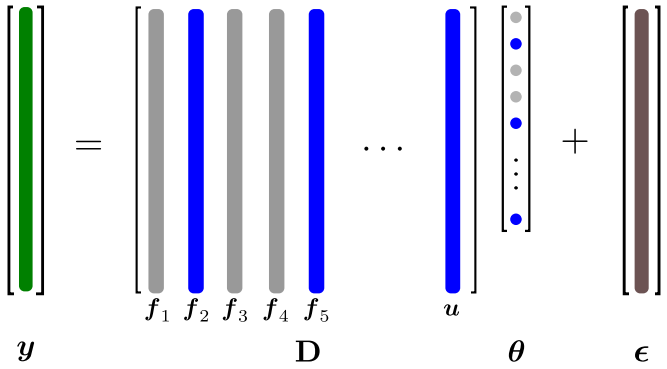


Figure 1: Sparse linear regression for selection of relevant basis functions (shown in blue) in parametric equation discovery approach.

Classical penalisation methods [23], including lasso, ridge, and elastic net penalties, can offer sparse solutions to the linear regression problem. They are deterministic approaches that employ constrained optimisation schemes to achieve sparse solutions, in that they add a convex penalty function to the usual least-squares objective and shrink the small weights to zero while leaving out a few large weights. Another popular deterministic method seeking sparse solutions to the linear regression problem is the sequential threshold least-squares algorithm, which iteratively solves the least-squares problem while zeroing out the small weights in successive iterations. This algorithm has been used in many works on equation discovery of nonlinear dynamical systems, including [3]. However, a common drawback of the deterministic approaches is

[2]The number of columns in the dictionary has been redefined as $P = l + 1$

that the results are sensitive to the choice of a regularisation parameter, and its tuning is required externally by cross-validation.

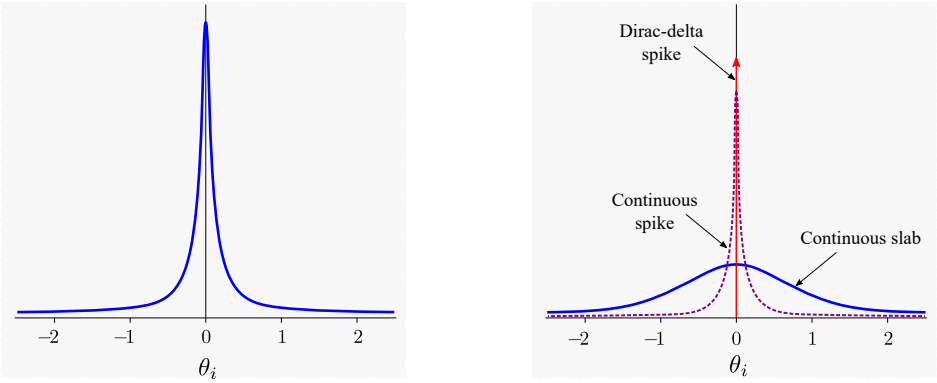
In this work, a Bayesian approach [24, 25] is adopted over a deterministic penalisation or thresholding approach, to solve the sparse linear regression problem. Apart from the usual advantage of uncertainty quantification, the a Bayesian framework offers three additional advantages: (a) it allows for natural penalisation through prior distributions, (b) the penalty parameter is simultaneously estimated with other model parameters and does not require determination through cross-validation, and (c) Bayesian techniques using Markov Chain Monte Carlo (MCMC) sampling facilitate a more straightforward implementation of non-convex penalty functions, unlike classical approaches which use convex penalty functions to achieve a unique minimum.

In a Bayesian approach, sparsity is induced by placing sparsity-promoting (or *shrinkage*) priors on the weights. These priors tend to shrink small weights to zero while allowing a few large weights to escape shrinkage. The densities of these 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. This structure of the priors tends to produce a selective shrinkage of the weights of the linear model, i.e. the posterior distributions of most weights are shrunk towards zero while a small set of weights have a large probability of being significantly different from zero [26]. Examples of such priors include: Laplace [27], Student's-t [24], Horseshoe [28], and spike-and-slab [29–33]. An overview of various shrinkage priors used in sparse Bayesian linear regression can be found in [34, 35].

The use of sparse Bayesian framework in data-driven equation discovery has been explored only very recently, and the current state of research in this direction has been quite limited. A handful of research that exists has mostly focussed on obtaining sparse solutions via SBL which is a particular implementation of the Student's-t prior [24]. Unlike common Bayesian algorithms that use MCMC-based random sampling, the SBL performs a marginal likelihood optimisation to yield parameter posteriors. The SBL was used in [36, 37] for equation discovery of nonlinear structural dynamic systems. A magnitude-based weight-thresholding was combined with SBL in [14] for discovery of governing partial differential equations. Recently, [38] extended the hybrid algorithm with a subsampling approach to discover governing equations of nonlinear systems in the presence of outliers. Apart from equation discovery, SBL has also been developed and used in different applications of structural health monitoring [39, 40], structural damage detection [41–43], input force localisation and estimation [44], to name a few.

The critical challenge in the equation discovery approach is to learn the correct set of basis functions from the dictionary $\mathbf{D}$. Although the SBL can provide quick results, it is based on the Student's-t prior that has less selective shrinkage capabilities compared to priors such as the SS prior. Figure 2 provides a visual illustration of the densities of the Student's-t and SS priors. The Student's-t prior is not as peaked around zero, hence it allows some weights – which should truly be zero – to take non-zero values. In an equation discovery setting, this issue may lead to more terms being selected than is true and may hinder the interpretability of the learned model. On the other hand, an SS prior comprises a small (or a point) mass at zero (the *spike*) for small weights, and a diffused density (the *slab*) for the large weights. The spike is capable of shrinking the small coefficients towards zero; hence the SS prior can induce stronger selective shrinkage of the coefficients compared to the Student's-t prior. Previously, the authors proposed the use of

SS priors in equation discovery of nonlinear systems [45], on which the current work builds.



(a) Student's-t (b) Spike-and-slab

Figure 2: Probability density functions of (a) the Student's-t prior, and (b) the spike-and-slab (SS) prior with either a Dirac-delta spike (displayed by an arrow pointing upwards) or a narrow continuous spike (displayed by dotted line).

This paper explores the performance of three different variants of SS priors in Bayesian equation discovery, and compares the results with those from the SBL. The case studies considered here are restricted to SDOF structural dynamic systems, and this is deemed sufficient to introduce and discuss the main aspects of the proposed approach. The layout of the paper is as follows: Section 2 introduces the model of the three SS prior variants used in this study, derives the MCMC procedure for sampling the model parameters, and outlines the methodology for Bayesian basis function selection using SS priors. Section 3 presents numerical demonstrations of equation discovery for four SDOF oscillators that are of interest in nonlinear structural dynamics: a linear oscillator, a Duffing oscillator with cubic nonlinearity, an oscillator with quadratic viscous damping and one with Coulomb damping. Next, the proposed approach is applied to the Silverbox experimental benchmark in Section 4. Finally, Section 5 provides a critical discussion on the results obtained with the SS priors, and Section 6 summarises the conclusions of the paper.

2. Bayesian basis function selection with spike-and-slab priors

The idea of basis function selection is to identify, out of P basis functions in the dictionary $\mathbf{D}$, the influential variables that have significant effect in explaining $\boldsymbol{y}$. Each combination of variables corresponds to a different model, and so basis function selection amounts to selecting a model from among 2^P possible models. In a Bayesian framework, the idea of distinguishing large effects from small effects is realised by imposing prior distributions on the weight vector $\boldsymbol{\theta}$, such that they have a probability mass concentrated around zero and the rest over a large range of the weight space. In this sense, the SS prior – featuring a mixture of two distributions, one with a spike at zero and the other with a diffused density over a wide range of possible values – conforms to a conceptual ideal and is often considered as the gold standard in Bayesian basis function selection [34].

The first SS prior proposed for Bayesian basis function selection had a spike defined by a Dirac-delta function at zero and a slab given by a uniform distribution [29]. Later on, the Dirac spike was replaced with a zero-mean Gaussian distribution with a small (but fixed) variance, and the uniform slab by another Gaussian distribution with a large variance [30]. In [33], the spike and slab distributions were considered

Variant	Name	Spike distribution	Slab distribution
1	CSS	Independent Student's-t	Independent Student's-t
2	DSS-i	Dirac-delta	Independent Student's-t
3	DSS-g	Dirac-delta	Correlated Student's-t

Table 1: Variants of spike-and-slab priors considered in this study.

Gaussian but with bimodal priors on their variances. For a review of Bayesian basis function selection strategies using SS priors, the reader is directed to [46].

In this paper, three different variants of SS priors, are used for basis function selection, as enumerated in Table 1. The first variant uses a mixture of two continuous zero-mean Student's-t distributions with different (a small and a large) variances for the spike and the slab [33, 47], and is referred to as the continuous spike-and-slab (CSS) prior. The next two variants feature a mixture of a discontinuous Dirac-delta spike distribution and a continuous Student's-t slab distribution, both centered at zero; they are jointly referred to as the discontinuous spike-and-slab, in short DSS, priors. The two DSS prior variants differ in their slab distributions, in that, one follows an independent Student's-t and the other follows a correlated Student's-t (with the correlation fashioned as Zellner's g-prior [48]), and are namely distinguished by their respective suffixes, DSS-i and DSS-g. The dictionary will contain many correlated variables (as will be seen later); as such, it is useful to find if a DSS prior that accounts for the correlation among the variables performs better than its independent counterpart.

To address the two components of the SS priors, a latent indicator variable is introduced for each weight θ_i. The latent indicator variable indicates the classification of a weight to one of the two components: the indicator variable takes a value one if the weight is assigned to the slab component of the prior, and zero otherwise. Since the posteriors using SS priors are analytically intractable, an MCMC-based Gibbs sampling scheme is employed to estimate the posterior probabilities of weights and indicator variables for all three SS prior variants. Basis function selection is then based on the posterior probability of the indicator variable which is estimated by counting the frequency of ones. The details of the SS prior models and the procedure of posterior computation and basis function selection follow next.

2.1. Model specification

For basis selection with SS priors, the linear regression problem in Eq. (7) is considered as part of a larger hierarchical model. Treating the residual error $\boldsymbol{\epsilon}$ as a vector of i.i.d. Gaussian noise variables with variance σ^2, the likelihood function can be written as,

$$\boldsymbol{y} \mid \boldsymbol{\theta}, \sigma^2 \sim \mathcal{N}\left(\mathbf{D}\boldsymbol{\theta}, \sigma^2 \mathbf{I}_N\right) \tag{8}$$

where $\mathcal{N}$ stands for a Gaussian distribution. To specify a two-component SS prior, a vector of latent indicator variables $\boldsymbol{z} = [z_1, \ldots, z_P]^T$ is introduced, where z_i takes a value 0 when θ_i belongs to the spike and takes a value 1 when θ_i falls in the slab. Also, denote by $\boldsymbol{\theta}_r \in \mathbb{R}^{r\times 1}$ the vector comprising those components

of $\boldsymbol{\theta}$ for which $z_i = 1$. Then, the SS prior can be written as,

$$p\left(\boldsymbol{\theta} \mid \boldsymbol{z}\right) = p_{\text{slab}}\left(\boldsymbol{\theta}_r\right) \prod_{i:z_i=0} p_{\text{spike}}\left(\theta_i\right) \tag{9}$$

where p_{spike} and p_{slab} denote the univariate spike and the multivariate slab distributions, respectively. The DSS priors considered in this study have the following forms of spike and slab distributions:

$$\text{DSS}: \; p_{\text{spike}}\left(\theta_i\right) = \delta_0, \;\; \text{and} \;\; p_{\text{slab}}\left(\boldsymbol{\theta}_r\right) = \mathcal{N}\left(\mathbf{0}, \sigma^2 v_s \mathbf{A}_{0,r}\right), \;\; \mathbf{A}_{0,r} = \begin{cases} \mathbf{I}_r & \text{for DSS-i} \\ N\left(\mathbf{D}_r^T \mathbf{D}_r\right)^{-1} & \text{for DSS-g} \end{cases} \tag{10a}$$

$$\text{CSS}: \; p_{\text{spike}}\left(\theta_i\right) = \mathcal{N}\left(0, \sigma^2 v_1 v_{s_i}\right) \;\; \text{and} \;\; p_{\text{slab}}\left(\boldsymbol{\theta}_r\right) = \prod_{i:z_i=1} \mathcal{N}\left(0, \sigma^2 v_0 v_{s_i}\right) \tag{10b}$$

Note that $\mathbf{D}_r$ in Eq. (10a) is a dictionary matrix that includes only the columns of $\mathbf{D}$ for which $z_i = 1$. Similarly, $\boldsymbol{\theta}_r \in R^{r \times 1}$ is the set of components of $\boldsymbol{\theta}$ for which the corresponding values of z_i equals 1 i.e. the components that belong to slab. The following points are to be noted:

- The spike distributions are considered independent of the slab distributions. The spike distribution in DSS is modelled by a Dirac-delta function at zero, denoted by δ_0, whereas that in CSS is modelled by a zero-mean small-variance continuous distribution.
- The slab distributions for all three variants have their mean centred at zero and their (co-)variances proportional to the product of measurement noise variance σ^2 and slab variance v_s. The inclusion of the measurement noise in the prior allows it to scale naturally with the scale (i.e. the measurement units) of the outcome $\boldsymbol{y}$.
- It is natural to assume weight-specific slab variances that can modify the strength of each individual prior [47]. While the CSS priors feature weight-specific slab variances v_{s_i}, a common slab variance v_s is assumed for all weights in the DSS priors. A common slab variance is found to yield better results for DSS priors.
- The difference in the variances of the spike and slab distributions in the CSS case is facilitated by the use of constants v_0 and v_1 such that $v_0 \ll v_1$, leading to a narrow spike and a wide slab.
- The covariance of the slab distribution of the DSS-g prior includes an additional scaling by the Fischer information matrix $N\left(\mathbf{D}_r^T \mathbf{D}_r\right)^{-1}$, which accounts for the correlation among the basis functions; in contrast, the DSS-i prior uses an independent slab distribution over each component of $\boldsymbol{\theta}$.

The marginal Student's-t distributions for the respective slabs (and spikes in case of CSS priors) given the noise variance are achieved by imposing an inverse-Gamma prior on the slab variance v_s (or equivalently on each v_{s_i} for CSS priors),

$$v_s \sim \mathcal{IG}\left(a_v, b_v\right) \tag{11}$$

The transformation of the Gaussian to Student's-t prior on $\boldsymbol{\theta}$ via the inverse-Gamma prior on v_s is because of the scale mixture property of Gaussians [49]. One could have alternatively used an exponential prior on v_s to obtain a marginal Laplace prior on $\boldsymbol{\theta}$, or simply treated v_s as a constant to impose a Gaussian prior.

The motivation for modelling the slab using Student's-t distributions lies in being able to provide a fair comparison with the SBL, which also uses a Student's-t prior.

In SS priors, each latent indicator variable z_i is assigned an independent Bernoulli prior, controlled by a common hyperparameter p_0,

$$z_i \mid p_0 \sim \text{Bern}(p_0) \tag{12}$$

Eq. (12) implies that the selection of a basis function from the dictionary $\mathbf{D}$ is independent of the inclusion of any other basis functions in $\mathbf{D}$. The hyperparameter p_0 in Eq. (12) represents the fraction of the total basis functions in $\mathbf{D}$ that are *a priori* expected to be selected in the final model; it can be assigned a fixed value. For example, $p_0 = \frac{1}{2}$ implies that each basis function in $\mathbf{D}$ has equal chance of being selected and reflects the prior belief that the model should include approximately half of the basis functions in $\mathbf{D}$. However, here p_0 is allowed to be adaptively refined by the data via a Beta prior,

$$p_0 \sim \text{Beta}(a_p, b_p) \tag{13}$$

Finally, the measurement noise variance σ^2 is assigned an inverse-Gamma prior,

$$\sigma^2 \sim \mathcal{IG}\left(a_\sigma, b_\sigma\right) \tag{14}$$

Note that a_v, b_v, a_p, b_p, a_σ, b_σ appearing in Eqs. (11), (13) and (14) are deterministic hyperparameters, controlling the shape of the respective hyper-priors. The complete hierarchical SS model for linear regression is illustrated in Figure 3.

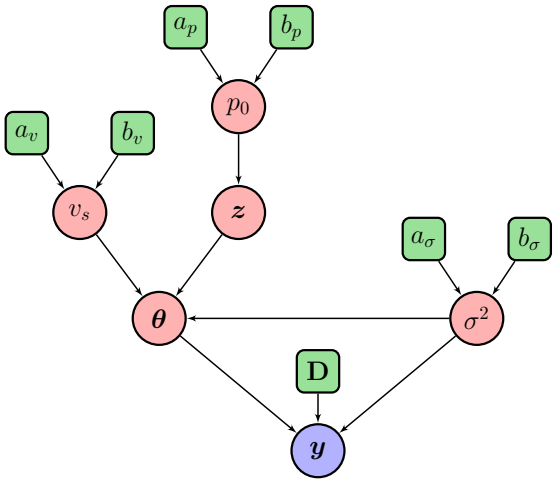


Figure 3: Graphical structure of the hierarchical spike-and-slab model for linear regression; the variables in circles represent random variables, while those in squares represent deterministic parameters. Note, in case of DSS priors, the slab variance v_s is a scalar, while that for CSS priors would be a vector of weight-specific slab variances.

2.2. Posterior computation

Once the hierarchical form of the SS priors is specified, the next part entails extracting the information relevant to basis function selection from the posteriors of $\boldsymbol{z}$, $\boldsymbol{\theta}$ and σ^2. The joint posterior of $p\left(\boldsymbol{\theta}, \boldsymbol{z}, v_s, \sigma^2, p_0 \mid \boldsymbol{y}\right)$ can be computed using Bayes' theorem in the form,

$$p\left(\boldsymbol{\theta}, \boldsymbol{z}, v_s, \sigma^2, p_0 \mid \boldsymbol{y}\right) = \frac{p\left(\boldsymbol{y} \mid \boldsymbol{\theta}, \sigma^2\right) p\left(\boldsymbol{\theta} \mid \boldsymbol{z}, v_s, \sigma^2\right) p\left(\boldsymbol{z} \mid p_0\right) p\left(v_s\right) p\left(\sigma^2\right) p\left(p_0\right)}{p\left(\boldsymbol{y}\right)} \tag{15}$$

where $p\left(\boldsymbol{y} \mid \boldsymbol{\theta}, \sigma^2\right)$ is the likelihood, $p\left(\boldsymbol{\theta} \mid \boldsymbol{z}, v_s, \sigma^2\right)$ is the prior over weights, $p\left(\boldsymbol{z} \mid p_0\right)$ is the prior over latent indicator variables, $p\left(v_s\right)$ is the prior over slab variance, $p\left(\sigma^2\right)$ is the prior over measurement noise, $p\left(p_0\right)$ is the prior over selection probability p_0, and $p\left(\boldsymbol{y}\right)$ is the normalising constant. Exact Bayesian inference is difficult with SS priors, and often MCMC techniques are employed to sample from the posteriors [31, 50]. In this case, a Gibbs sampler [51] is used to draw samples from the posterior. Gibbs sampling needs knowledge of the full conditional distributions which can be derived analytically with the use of conjugate priors. It should be mentioned that the sampling schemes for the DSS and CSS priors differ slightly as a result of the need to integrate out the Dirac-delta function in the case of DSS priors. The Gibbs sampling scheme for the CSS prior has been adopted from [47], and details of the sampling steps are provided in Appendix A. Below, the Gibbs sampling steps for the parameters $\boldsymbol{\theta}$, $\boldsymbol{z}$, p_0, v_s and σ^2 of the DSS priors are provided. The derivation of the conditional densities are provided in a supplementary document accompanying this paper.

(a) The components of $\boldsymbol{\theta}$ that correspond to $z_i = 0$ (i.e. belong to the Dirac-delta spike) are set to zero. The rest of the components belonging to the slab, represented by $\boldsymbol{\theta}_r$, are sampled as follows,

$$\boldsymbol{\theta}_r \mid \boldsymbol{y}, v_s, \sigma^2 \sim \mathcal{N}\left(\boldsymbol{\mu}, \sigma^2 \boldsymbol{\Sigma}\right) \tag{16}$$

where $\boldsymbol{\Sigma} = \left(\mathbf{D}_r^T \mathbf{D}_r + v_s^{-1} \mathbf{A}_{0,r}^{-1}\right)^{-1}$ and $\boldsymbol{\mu} = \boldsymbol{\Sigma} \mathbf{D}_r^T \boldsymbol{y}$.

(b) σ^2 is sampled from an inverse Gamma distribution as follows,

$$\sigma^2 \mid \boldsymbol{y}, \boldsymbol{z}, v_s \sim \mathcal{IG}\left(a_\sigma + \frac{N}{2},\ b_\sigma + \frac{1}{2}\left(\boldsymbol{y}^T \boldsymbol{y} - \boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}\right)\right) \tag{17}$$

(c) v_s is sampled from an inverse Gamma distribution as follows,

$$v_s \mid \boldsymbol{\theta}, \boldsymbol{z}, \sigma^2 \sim \mathcal{IG}\left(a_v + \frac{s_z}{2},\ b_v + \frac{1}{2\sigma^2} \boldsymbol{\theta}_r^T \mathbf{A}_{0,r}^{-1} \boldsymbol{\theta}_r\right) \tag{18}$$

where $s_z = \sum_{i=1}^{P} z_i$.

(d) p_0 is sampled from a Beta distribution as follows,

$$p_0 \mid \boldsymbol{z} \sim \text{Beta}\left(a_p + s_z,\ b_p + P - s_z\right) \tag{19}$$

(e) The conditional distribution of $\boldsymbol{z}$ is expressed componentwise. The odds of $z_i = 1$ to $z_i = 0$ are computed, given the values of other $\boldsymbol{z}$ components, denoted here as $\boldsymbol{z}_{-i}$. The components of $\boldsymbol{z}$ are sampled (in a random order) as follows,

$$z_i \mid \boldsymbol{y}, v_s, p_0 \sim \text{Bern}(\xi_i), \text{ with } \xi_i = \frac{p_0}{p_0 + \frac{p(\boldsymbol{y} \mid z_i = 0, \boldsymbol{z}_{-i}, v_s)}{p(\boldsymbol{y} \mid z_i = 1, \boldsymbol{z}_{-i}, v_s)}(1 - p_0)} \tag{20}$$

In the above sampling step, the marginal likelihood $p\left(\boldsymbol{y} \mid \boldsymbol{z}, v_s\right)$ after integrating out $\boldsymbol{\theta}$ and σ^2 is obtained as,

$$p\left(\boldsymbol{y} \mid \boldsymbol{z}, v_s\right) = \frac{\Gamma(a_\sigma + 0.5N)}{(2\pi)^{N/2} (v_s)^{s_z/2}} \frac{(b_\sigma)^{a_\sigma}}{\Gamma(a_\sigma)} \frac{\left(\det\left(\mathbf{A}_{0,r}^{-1}\right)\right)^{1/2} \left(\det\left(\left(\mathbf{D}_r^T \mathbf{D}_r + v_s^{-1} \mathbf{A}_{0,r}^{-1}\right)^{-1}\right)\right)^{1/2}}{\left(b_\sigma + 0.5 \boldsymbol{y}^T \left(\mathbf{I}_N - \mathbf{D}_r \left(\mathbf{D}_r^T \mathbf{D}_r + v_s^{-1} \mathbf{A}_{0,r}^{-1}\right)^{-1} \mathbf{D}_r^T\right) \boldsymbol{y}\right)^{(a_\sigma + 0.5N)}} \tag{21}$$

where $\Gamma(\cdot)$ denotes the Gamma function and $\det(\cdot)$ denotes the determinant operator. However, when all z_is are zero, the marginal likelihood reduces to a much simpler form

$$p\left(\boldsymbol{y} \mid \boldsymbol{z}, v_s\right) = \frac{\Gamma(a_\sigma + 0.5N)}{(2\pi)^{N/2}} \frac{(b_\sigma)^{a_\sigma}}{\Gamma(a_\sigma)} \frac{1}{(b_\sigma + 0.5\boldsymbol{y}^T\boldsymbol{y})^{(a_\sigma + 0.5N)}} \tag{22}$$

By repeated successive sampling using Eqs. (16) to (20), the following Markov chain is produced,

$$\boldsymbol{\theta}^{(0)}, {\sigma^2}^{(0)}, v_s^{(0)}, p_0^{(0)}, \boldsymbol{z}^{(0)}, \ldots, \boldsymbol{\theta}^{(l)}, {\sigma^2}^{(l)}, v_s^{(l)}, p_0^{(l)}, \boldsymbol{z}^{(l)}, \ldots \tag{23}$$

which embeds the Markov chains for $\boldsymbol{z}$, $\boldsymbol{\theta}$ and σ^2. The first few samples of the chain are discarded as burn-in, and the remaining J samples are used for basis function selection, as described next.

2.3. Basis function selection and posterior prediction

As mentioned previously, there are 2^P models possible with P basis functions in the dictionary, where a model is indexed by which of the z_is equal one and which equal zero. For example, the model with zero basis functions has $\boldsymbol{z} = \boldsymbol{0}$, whereas the model that includes all basis functions has $\boldsymbol{z} = \boldsymbol{1}$. Finding the model with highest posterior probability is often challenging when P is large, as one would probably need more than 2^P samples to explore the entire space of models. In this work, the marginal posterior inclusion probabilities (PIP), $p\left(z_i = 1 \mid \boldsymbol{y}\right)$, are used to select those basis functions whose corresponding probability is more than a fixed probability threshold. The PIPs are approximated using J Gibbs samples, as follows,

$$p\left(z_i = 1 \mid \boldsymbol{y}\right) \approx \frac{1}{J} \sum_{j=1}^{J} \mathbb{I}\left(z_i^{(j)} = 1\right) \tag{24}$$

where $\mathbb{I}\left(\cdot\right)$ stands for an indicator function. Specifically, one selects the i^{th} basis function from $\mathbf{D}$ and includes it in the final model if,

$$p\left(z_i = 1 \mid \boldsymbol{y}\right) > 0.5 \tag{25}$$

The higher the posterior mean of the indicator variable, the higher is evidence that the parameter θ_i might be different from zero and therefore the corresponding basis function will have an impact on $\boldsymbol{y}$. The above criterion implies the selection of basis functions that appear in at least half of the visited models. The final estimated model $\hat{\mathcal{M}}$ so obtained corresponds to the median probability model [52], and is computationally advantageous since estimating this model often requires fewer Gibbs iterations than are required for the highest probability model. The model $\hat{\mathcal{M}}$ is chosen as the discovered governing equations of motion in this study. Post basis function selection, the estimated mean and covariance of the parameter vector $\boldsymbol{\theta}$, denoted by $\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}$, respectively, will feature non-zero components only at indices corresponding to those of the selected basis functions.

Subsequently, predictions with the estimated model $\hat{\mathcal{M}}$ can be performed using $\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}$ via the expressions,

$$\boldsymbol{\mu}_{\boldsymbol{y}^*} = \mathbf{D}^* \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}} \tag{26}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{y}^*} = \mathbf{D}^* \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}} {\mathbf{D}^*}^T + \hat{\mu}_{\sigma^2} \mathbf{I}_{N^*} \tag{27}$$

where $\mathbf{D}^* \in \mathbb{R}^{N^* \times P}$ is the test dictionary, defined at a set of N^* previously-unseen test data points, $\boldsymbol{\mu}_{\boldsymbol{y}^*} \in \mathbb{R}^{N^* \times 1}$ is the predicted mean of the target test vector, $\boldsymbol{\Sigma}_{\boldsymbol{y}^*} \in \mathbb{R}^{N^* \times N^*}$ is the predicted covariance of

the target test vector, and $\hat{\mu}_{\sigma^2} \in \mathbb{R}$ is the mean of the measurement noise variance estimated using J Gibbs samples of σ^2.

3. Numerical studies

In this section, the performance of the proposed sparse Bayesian algorithms in discovering governing equations is investigated. SDOF oscillators of the form expressed by Eq. (4) containing the nonlinear term $g(x_1, x_2)$ are considered, where x_1 and x_2 represent the displacement and velocity states of the oscillator. Different forms of the nonlinearity $g(x_1, x_2)$ lead to different systems of engineering interest. Four different cases of nonlinearities $g(x_1, x_2)$ are considered in this study, as enumerated in Table 2.

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	$c_2 x_2 \lvert x_2 \rvert$	$c_2 = 2$
4	Coulomb friction damping	$c_F \mathrm{sgn}\,(x_2)$	$c_F = 1$

Table 2: Simulation cases.

The first system is a linear system, used here to verify if the proposed method is capable of ruling out the existence of any nonlinearities in the dynamical system. The second system is a Duffing oscillator, with a cubic displacement nonlinearity $g(x_1, x_2) = k_3 x_1^3$; it can be used to represent many physical systems and has been widely used in a large number of studies in nonlinear system identification [19]. In structural systems, the nonlinearity can be used to represent hardening geometric nonlinearity arising as a result of large displacements; as the displacement increases, the nonlinear restoring force becomes greater than that expected from the linear term alone. The third system includes a quadratic viscous damping nonlinearity $g(x_1, x_2) = c_2 x_2 |x_2|$, where $|\cdot|$ denotes the absolute value. This type of damping occurs in fluid flows through orifices or around a slender member. The former situation is common in automotive dampers, whereas the latter occurs in fluid loading of offshore structures [53]. The fourth system includes a Coulomb friction damping nonlinearity $g(x_1, x_2) = c_F \mathrm{sgn}\,(x_2)$, where $\mathrm{sgn}\,(\cdot)$ denotes the signum function. This type of nonlinearity is encountered in situations that involve interfacial motion or sliding [54], such as dry sliding occurring in bolted joints. The four SDOF systems are simulated using the following parameters:

- The parameters of the linear system are taken as: $m = 1$, $c = 2$, and $k = 1000$.
- 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 2.
- The systems are excited using a bandlimited – passband $[0, 100]$Hz – Gaussian excitation with zero mean and standard deviation of 50.
- 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 1000Hz.

- The acceleration $\dot{x}_2$ is obtained using Eq. (4).

Before commencing equation discovery, one requires the knowledge of the time-series data of displacement, velocity, acceleration and input force signals from forced vibration testing of the system; the acceleration data is used as the measurement vector $\boldsymbol{y}$ whereas the displacement, velocity, and input force data are used in composing the basis functions of the dictionary $\mathbf{D}$ (see Eq. (6)). It is assumed that noisy measurements of all input and outputs, i.e., displacement x_1, velocity x_2, acceleration $\dot{x}_2$, and input force u are available, and the noisy signals are used to compose the dictionary $\mathbf{D}$ and the target measurement vector $\boldsymbol{y}$. The noise in the measurements 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.

In this work, the dictionary $\mathbf{D}$ is constructed with 36 basis functions, where each basis function represents a certain function of the states x_1 and x_2:

$$\mathbf{D} = \left\{P^1(\boldsymbol{x}), \ldots, P^6(\boldsymbol{x}), \operatorname{sgn}(\boldsymbol{x}), |\boldsymbol{x}|, \boldsymbol{x} \otimes |\boldsymbol{x}|, u\right\} \tag{28}$$

Here, $P^\gamma(\boldsymbol{x})$ denotes the set of terms in the polynomial expansion of the sum of state vectors $(x_1+x_2)^\gamma$. The dictionary consists of basis functions that are terms from polynomial orders up to $\gamma = 6$ and certain other terms. The term $\operatorname{sgn}(\boldsymbol{x})$ represents the signum functions of states, i.e., $\operatorname{sgn}(x_1)$ and $\operatorname{sgn}(x_2)$. Similarly, $|\boldsymbol{x}|$ denotes the absolute functions of states, i.e., $|x_1|$ and $|x_2|$. The tensor product term $\boldsymbol{x} \otimes |\boldsymbol{x}|$ represents the set of functions: $x_1|x_1|$, $x_1|x_2|$, $x_2|x_1|$ and $x_2|x_2|$. Note that the total number of models that can be formed by combinatorial selection of all 36 basis functions in the dictionary is 2^{36}, and grows exponentially as the number of basis functions increases.

An issue with the constructed dictionary in Eq. (28) is that it is often ill-conditioned. This happens due to a combined effect of (a) the large scale difference among the basis functions and (b) the presence of strong linear correlation between certain basis functions. Appropriate scaling of the columns can help 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 normalised (i.e. they are centered and scaled to have zero mean and unit standard deviation). Additionally, the training measurement data are detrended to have zero mean; as such there is no need to include a constant intercept term in the dictionary. Put formally, the training dictionary and the target vector $(\mathbf{D}_s, \boldsymbol{y}_s)$ input to the Bayesian inference algorithm have the forms,

$$\begin{aligned} \mathbf{D}^s &= (\mathbf{D} - \mathbf{1}\boldsymbol{\mu}_\mathbf{D})\, \boldsymbol{S}_\mathbf{D}^{-1} \\ \boldsymbol{y}^s &= \boldsymbol{y} - \mathbf{1}\mu_{\boldsymbol{y}} \end{aligned} \tag{29}$$

where $\mathbf{1}$ denotes a column vector of ones, $\boldsymbol{\mu}_\mathbf{D}$ is a row vector of the column-wise means of $\mathbf{D}$, $\boldsymbol{S}_\mathbf{D}$ is a diagonal matrix of the column-wise standard deviations of $\mathbf{D}$, and $\mu_{\boldsymbol{y}}$ is the mean of the training target measurement vector $\boldsymbol{y}$. Note that this modification implies that, post Bayesian inference, the estimated mean and covariance of the scaled coefficients $\boldsymbol{\theta}^s$, 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,

$$\begin{aligned} \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}} &= \boldsymbol{S}_\mathbf{D}^{-1} \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}^s} \\ \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}} &= \boldsymbol{S}_\mathbf{D}^{-1} \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}^s} \boldsymbol{S}_\mathbf{D}^{-1} \end{aligned} \tag{30}$$

For Bayesian inference with the SS priors, the Gibbs sampler is commenced with the following initial values of the hyperparameters: $p_0^{(0)} = 0.1$, $v_s^{(0)} = 10$, and $\sigma^{2(0)}$ is set equal to the residual variance from

ordinary least-squares regression. Additionally, for the CSS prior, each component of the vector of weight-specific slab variances is initialised to the value of $v_s^{(0)}$, and the two variance-scaling constants are set to $v_0 = \frac{1}{N}$ and $v_1 = 100v_0$, respectively. Note that both v_0 and v_1 are made to depend on the sample size to ensure model selection consistency [47]. To facilitate faster convergence of the Gibbs sampler to a good solution, the initial vector of binary latent variables $\boldsymbol{z}^{(0)}$ is computed by starting off with $z_1, \ldots, z_P$ set to zero and then activating the components of $\boldsymbol{z}$ that reduce the mean-squared error on the (training) data, until an integer number ($\approx p_0^{(0)} P$) of components of $\boldsymbol{z}$ are equal to one. Given all the other parameters, the initial value of $\boldsymbol{\theta}^{(0)}$ is obtained by sampling from Eq. (16). The deterministic prior parameters are set to the following values: $a_p = 0.1$, $b_p = 1$ are chosen for the Beta prior on p_0 to promote selection of sparse models, $a_v = 0.5$, $b_v = 0.5$ for inverse-Gamma prior on slab variance, and $a_\sigma = 10^{-4}$, $b_\sigma = 10^{-4}$ are chosen for a non-informative prior on measurement noise. Four Markov chains are used for Gibbs sampling with 5000 samples in each chain. The first 1000 samples of each chain are discarded as burn-in, and the remaining $4000 \times 4 = 16000$ samples are used for posterior computation. To ensure variability across the chains, each of them is initialised with randomly perturbed values of the aforementioned initial hyperparameters. The multivariate potential scale reduction factor $\hat{R}$ [55], which estimates the potential decrease in the between-chain variance with respect to the within-chain variance, is applied to assess the convergence of the generated samples of $\boldsymbol{\theta}$; a value of $\hat{R} < 1.1$ is adopted to decide if convergence has been reached.

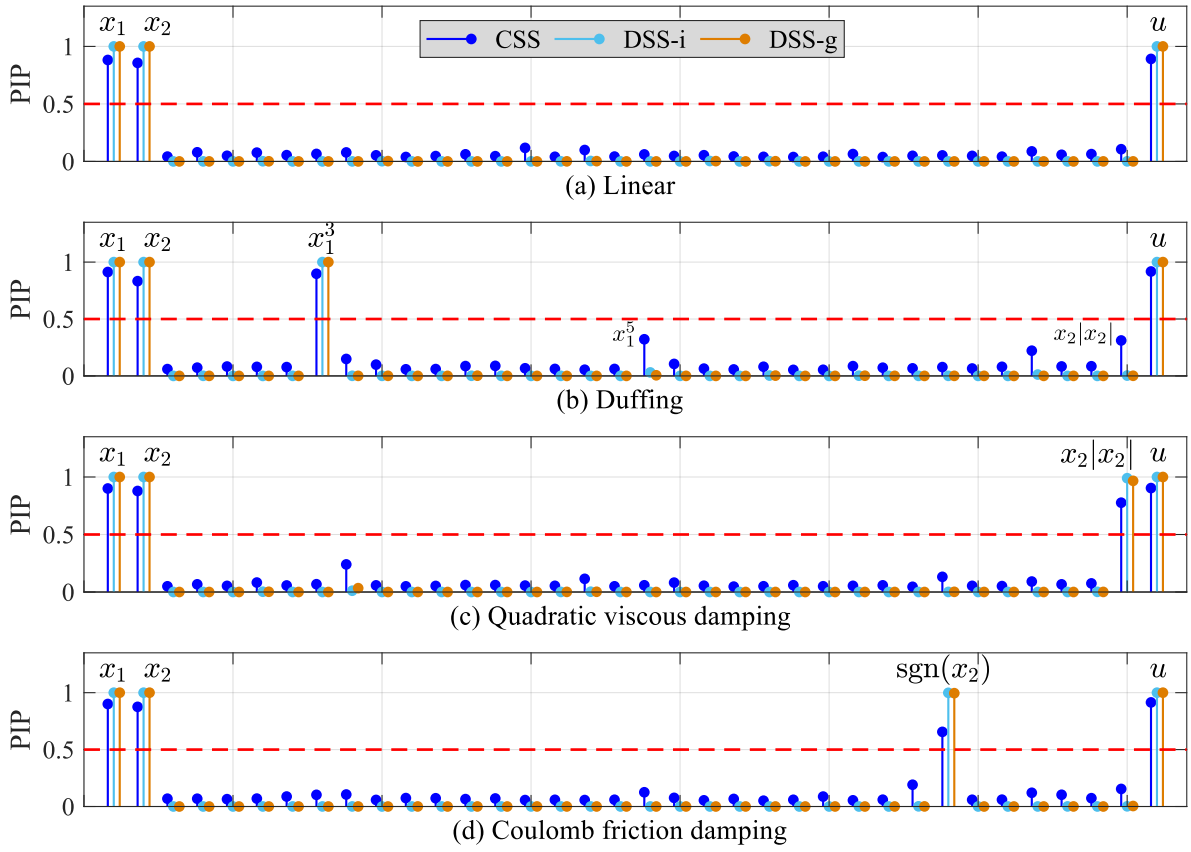


Figure 4: Basis function selection based on marginal posterior inclusion probability (PIP), $p(z_i = 1 \mid \boldsymbol{y})$. The horizontal axes represent the collection of 36 basis functions; the functions having marginal PIP > 0.5 are included in the final estimated model.

Figure 4 demonstrates the procedure of basis function selection for the four systems, based on the marginal PIP, $p(z_i = 1 \mid \boldsymbol{y})$, $i = 1, \ldots, 36$. When $p(z_i = 1 \mid \boldsymbol{y}) = 1$, it implies that the i^{th} basis function had been selected in all Gibbs posterior samples, while $p(z_i = 1 \mid \boldsymbol{y}) = 0$ implies the i^{th} basis function has never been selected. As mentioned in Section 2.3, only those basis functions are included in the final estimated model whose corresponding marginal PIPs are greater than the set threshold of 0.5 (shown by the dotted line in red in Figure 4). It can be seen that the estimated models for all the four systems are able to select the true basis functions out of the pool of 36 basis functions. For DSS priors, the computed marginal PIPs corresponding to the true basis functions are close to one, which indicates a strong selection probability. However, the selection probabilities with CSS priors are not as strong; they exhibit smaller PIPs for the relevant variables, compared to those from DSS priors. The weaker selection probability with CSS priors is apparent in the Duffing oscillator case, where the true relevant variables x_1 and x_1^3 draw marginal PIPs of around 0.9, while a irrelevant variable x_1^5 receives a marginal PIP of 0.3. Similarly, x_2 is selected with PIP of 0.8 whereas $x_2\,|x_2|$ gets discarded with a PIP of 0.3. Although this behaviour occurs more frequently with CSS priors, it can also happen with DSS priors, especially in situations when there are strong correlations between certain basis functions causing the Bayesian algorithm to be confused as to which of the set of correlated basis functions should be selected.

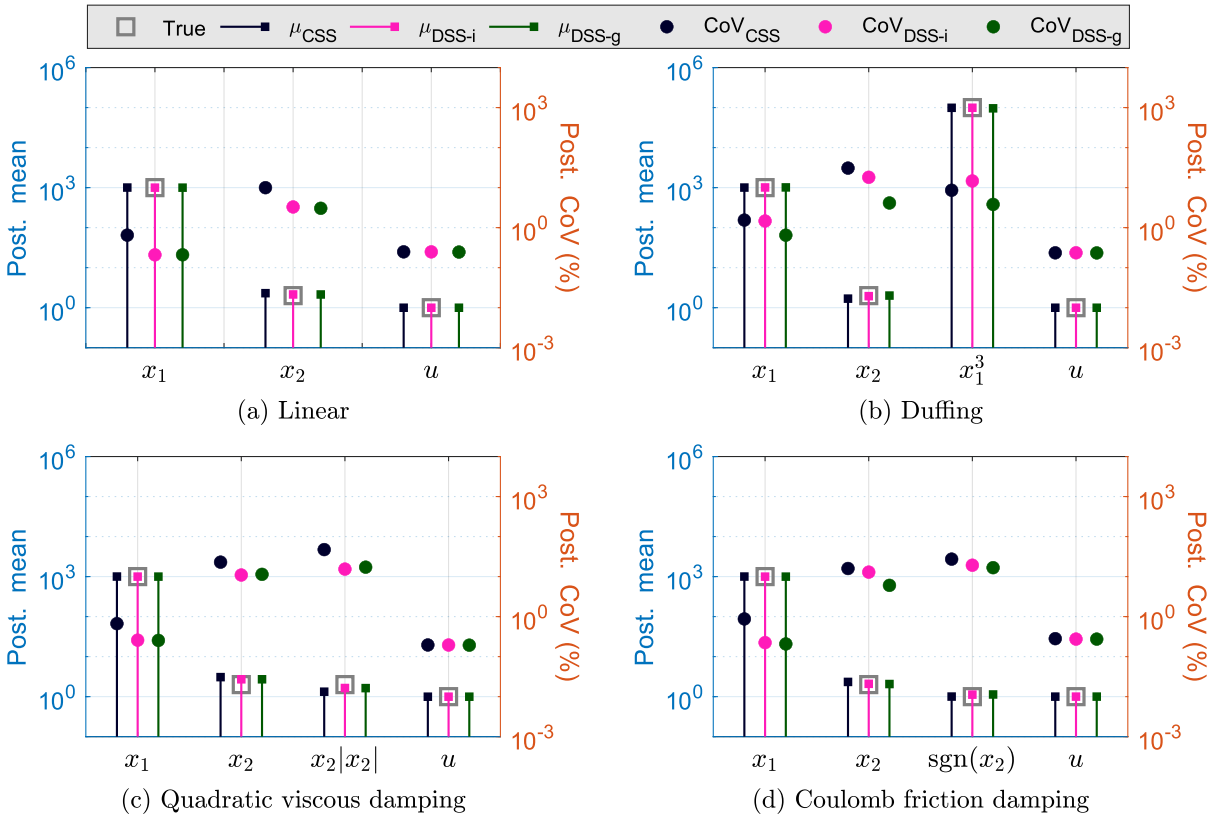


Figure 5: Estimates of parameters of the selected basis functions. The left vertical axes represents absolute posterior means and the right axes illustrated the associated coefficient of variations (CoVs) of the estimated parameters inferred using CSS, DSS-i and DSS-g priors.

Figure 5 plots the absolute posterior means and coefficient of variations (CoVs) of the parameters that correspond to the selected basis functions in Figure 4; the CoVs are expressed as percentage ratios of

the means to the standard deviations. The mean values of the parameters are found to agree very well with the corresponding true values, and the posterior CoVs are quite small for the parameters associated with variables u and x_1. Higher COVs are seen for parameter estimates associated with variables that are functions of x_2. The greater uncertainty associated with variables x_2 and functions of x_2 is because of two factors: (a) small values of the parameters associated with these variables and (b) presence of many other correlated variables in the dictionary, which, coupled together, confuses the Bayesian learner. It is also noted that the posterior standard deviations of the parameters inferred with CSS priors are comparatively larger than those inferred with the DSS priors. For illustration, the pairwise joint posteriors of the parameters for the Duffing oscillator case are plotted in Figure 6. Clearly, the posterior samples with CSS priors can be seen to spread over a larger parameter space in the plots. This behavior with CSS priors is caused by the less restrictive continuous spike distribution, which occasionally allows the irrelevant variables to take non-zero weights and biases the weights of the relevant parameters, thereby inducing a greater spread in the weights (or parameters) of the relevant variables. The posterior mean values of the parameters inferred with CSS priors are, however, found to agree well with the true values of the parameters. It should also be mentioned that the posteriors obtained with SS priors can be multi-modal, as seen from Figure 6.

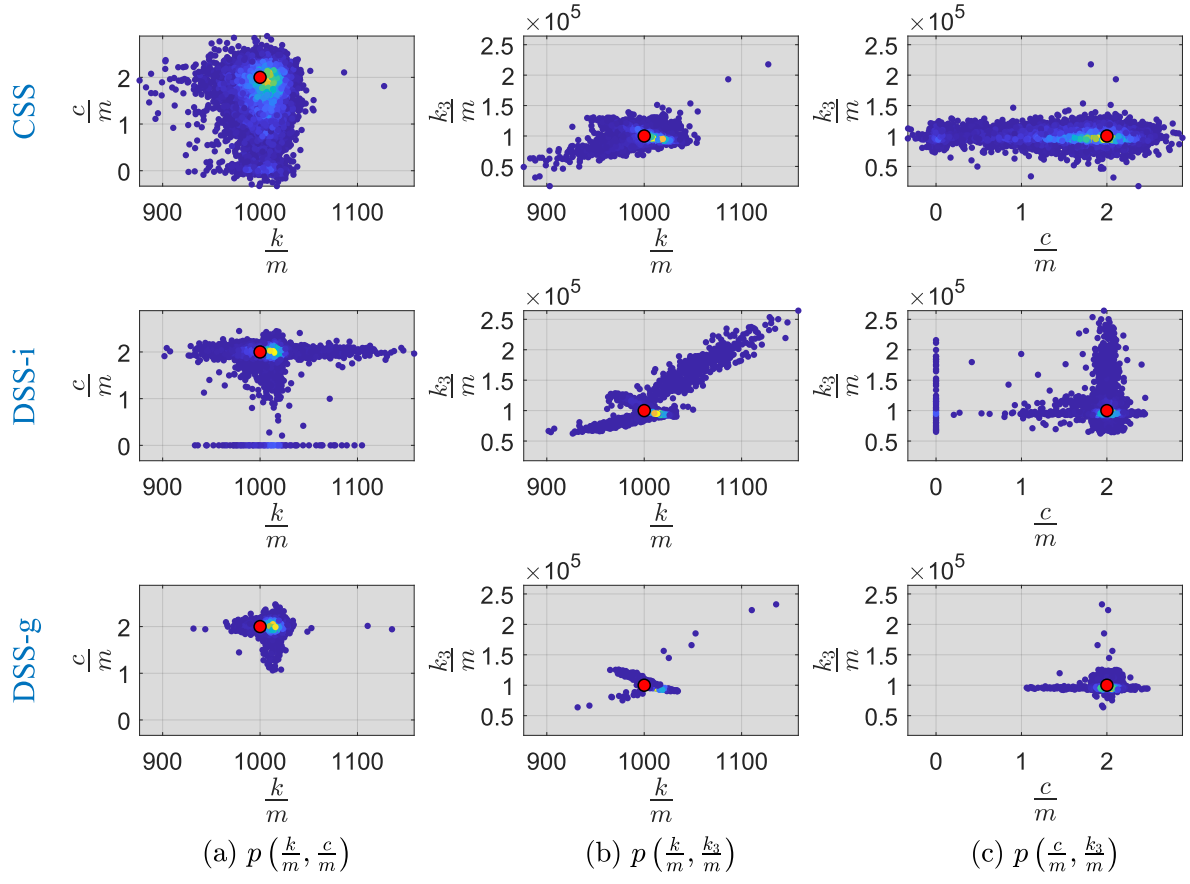


Figure 6: Pairwise joint posteriors of the parameters $\frac{k}{m}$, $\frac{c}{m}$, and $\frac{k_3}{m}$ corresponding to the selected basis functions x_1, x_2, and x_1^3, for the Duffing oscillator case. The red circles indicate the true values of parameters.

Equation discovery performance comparison using Monte Carlo simulations

In this section, Monte Carlo simulations are used to assess the equation discovery performance of the proposed MCMC algorithms with SS priors – hereafter collectively referred to as the MCMC-SS algorithms. Furthermore, the popular SBL algorithm [24, 56] using a Student's-t prior is implemented for the sake of comparison of equation discovery results. The freely available *SparseBayes* software [57] is used for implementing the SBL.

1000 different realisations for each of the four systems, as summarised in Table 2, 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:

- Weight estimation error, $e_\theta = \frac{\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_2}{\|\boldsymbol{\theta}\|_2}$, where $\hat{\boldsymbol{\theta}}$ is the estimate of the true weight vector $\boldsymbol{\theta}$. In the case of SS priors, $\hat{\boldsymbol{\theta}}$ is obtained as the mean estimate of the posterior sample, whereas in the case of SBL, it is obtained as the maximum *a posteriori* estimate. Similarly, one can also define a scaled weight estimation error, $e_{\theta^s} = \frac{\|\boldsymbol{S}_\mathbf{D}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})\|_2}{\|\boldsymbol{S}_\mathbf{D}\boldsymbol{\theta}\|_2}$.

- Test set prediction error, $e_p = \frac{\|\boldsymbol{y}^* - \mathbf{D}^*\hat{\boldsymbol{\theta}}\|_2}{\|\boldsymbol{y}^*\|_2} \times 100$, where $\boldsymbol{y}^*$ is the test set of responses, $\mathbf{D}^*$ is the unscaled test dictionary, and $\hat{\boldsymbol{\theta}}$ is the estimate of the unscaled 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 functions selected to the total number of basis functions selected in the estimated model. A good basis function selection algorithm should output a model with fewer false discoveries and result in a low FDR.

- Exact model selection indicator, denoted by $\hat{\mathcal{M}} = \mathcal{M}$, is an indicator variable that takes value 1 when the estimated model $\hat{\mathcal{M}}$ has the exact same basis functions as the true model $\mathcal{M}$, and is zero otherwise.

- Superset model selection indicator, denoted by $\hat{\mathcal{M}} \supset \mathcal{M}$, is an indicator variable that takes value 1 when the estimated model $\hat{\mathcal{M}}$ includes all the basis functions 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 3.

Table 3 shows that all the proposed MCMC-SS algorithms outperform the SBL in all metrics of performance. It may be noted that the weight errors, given by e_θ, appear quite high for both SBL and MCMC-SS algorithms, however, their median and mode values are comparatively quite small. The high mean values are caused by the long tails of the error distributions, as exemplified in Figure 7 for the Duffing oscillator case. In general, the MCMC-SS algorithms yield quite low levels of parameter estimation errors and false discoveries compared to the SBL. The SBL includes a lot of false discoveries, which is a major deterrent in equation discovery, as selecting the correct set of basis functions is crucial for drawing scientific conclusions based on the estimated model. Moreover, the models estimated by MCMC-SS surpass those by SBL in terms

Type	Alg.	e_{θ^s}	e_θ	e_p	FDR	$\hat{\mathcal{M}} = \mathcal{M}$	$\hat{\mathcal{M}} \supset \mathcal{M}$
Linear	SBL	0.010	502.261	0.097	0.576	0.005	0.999
	MCMC-CSS	0.006	6.735	0.074	0.002	0.993	0.995
	MCMC-DSS-i	0.004	6.479	0.073	0.002	0.993	1.000
	MCMC-DSS-g	**0.004**	**1.513**	**0.071**	**0.001**	**0.997**	**1.000**
Duffing	SBL	0.070	47.377	0.091	0.560	0.001	0.976
	MCMC-CSS	0.028	4.101	0.080	0.051	0.778	0.972
	MCMC-DSS-i	0.030	5.130	0.079	0.040	0.842	0.977
	MCMC-DSS-g	**0.023**	**3.761**	**0.077**	**0.026**	**0.898**	**0.977**
Quadratic damping	SBL	0.017	1546.542	0.073	0.497	0.003	**0.931**
	MCMC-CSS	**0.017**	0.005	0.072	**0.006**	**0.876**	0.886
	MCMC-DSS-i	0.018	**0.004**	**0.072**	0.031	0.850	0.855
	MCMC-DSS-g	0.020	0.004	0.072	0.031	0.845	0.847
Coulomb damping	SBL	0.013	1034.780	0.092	0.496	0.003	**0.993**
	MCMC-CSS	0.010	1.476	0.071	**0.002**	0.769	0.775
	MCMC-DSS-i	0.011	0.004	0.071	0.018	**0.835**	0.838
	MCMC-DSS-g	**0.009**	**0.004**	**0.070**	0.010	0.838	0.840

Table 3: Comparison of results from SBL, MCMC-CSS, MCMC-DSS-i and MCMC-DSS-g, averaged over 1000 realisations. Small values of e^s_θ, e_θ, e_p, FDR are better, whereas average values of $\hat{\mathcal{M}} = \mathcal{M}$ and $\hat{\mathcal{M}} \supset \mathcal{M}$ closer to one are better; bold numbers highlight the best performing metric.

of predictive accuracy as well. Among the three variants of MCMC-SS algorithm, the DSS priors yield quite similar results and often tend to perform slightly better than the CSS prior. The SBL, however, is found to do better in superset model selection rate for the cases of quadratic viscous damping and Coulomb friction damping. In those cases, the SBL is able to include all the relevant variables in the estimated model more often than the MCMC-SS algorithms. The weaker sparsity-promoting property of the SBL allows it more often include all the relevant variables but with many other false discoveries.

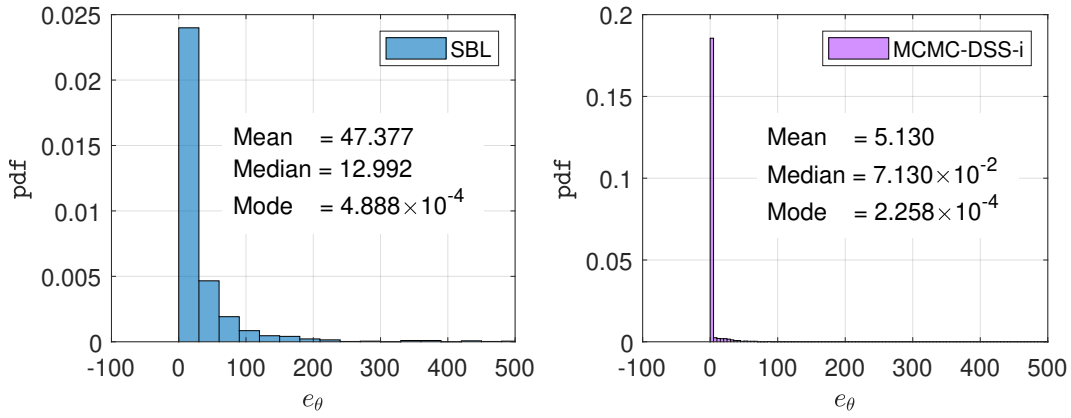


Figure 7: Histograms of weight errors, e_θ, for the Duffing oscillator case, using 1000 Monte Carlo samples.

Overall, the MCMC-SS algorithms show very strong model selection consistency; they are able to select the true models more often and show extremely low rates of false discoveries – an important requirement for intepretability of discovered equations. It can be inferred that the SBL (using the Student's-t prior) very rarely finds the exact true model and will likely include many false discoveries. That being said, the SBL is remarkably fast compared to the MCMC-SS algorithms. A comparison of the average runtimes of the four sparse Bayesian learning algorithms are provided in Table 4; the algorithms are run on a 64-bit Windows

10 PC with Intel Xeon E5-2698v4 CPU @ 2.20GHz. The SBL is undoubtedly the cheapest in terms of computational time, while all the MCMC-SS algorithms are orders of magnitude more expensive than the SBL. Note that the MCMC algorithms could be more time-consuming if the number of sampling iterations were increased. Between the three SS prior variants, the MCMC algorithm implemented with CSS priors is much cheaper than the DSS priors; the increased computational time for the DSS priors is due to the calculation of the marginal likelihood in Eq. (21), needed for integrating out the Dirac-delta function.

SBL	MCMC-CSS	MCMC-DSS-i	MCMC-DSS-g
0.03s	3.89s	36.22s	34.59s

Table 4: Average computational runtimes of SBL and MCMC-SS (run with single chain for 5000 sampling iterations).

4. An experimental application on Silverbox benchmark

This section presents an application of SS priors for Bayesian equation discovery of the Silverbox benchmark [58, 59]. The Silverbox is an electrical circuit resembling a Duffing oscillator with a moving mass m, a viscous damping c and a nonlinear spring $k(q)$. The circuit is designed to relate the displacement $q(t)$ (the output) to the force $u(t)$ (the input) by the following differential equation,

$$m\ddot{q}(t) + c\dot{q}(t) + \underbrace{\left(a + bq^2(t)\right)}_{k(q(t))} q(t) = u(t) \tag{31}$$

The input-output data for the Silverbox benchmark consist of force and displacement measurements. The data here were supplied as part of the *Nonlinear System Identification Benchmarks* workshop held at VUB Brussels and Eindhoven University over the last few years. More details on the experiment and benchmark data can be found in [58, 59].

The `Schroeder80mV.mat` dataset of the Silverbox benchmark has been used here for the application. A section of the input-output data consisting of 10400 samples (between 0.14s and 17.2s), as shown in Figure 8, was used for training the sparse Bayesian algorithms; the input data comprised a random-phase multi-sine excitation containing 1342 odd harmonics of a base frequency $8192/f_{\text{samp}}$Hz and the sampling rate f_{samp} in the experiments was 610.35Hz. Both the displacement and force signals were detrended before use. As the displacement and force signals were the only data measured, numerical differentiation was employed to obtain the velocity and acceleration signals from the displacement data. This case, therefore, serves as a measurement scenario where only one output response is observed.

The results of parameter estimation from MCMC-SS and SBL algorithms are presented in Table 5. The MCMC-SS algorithms selected five basis functions which include the correct linear stiffness term x_1, the viscous damping x_2, the nonlinear cubic stiffness x_1^3, the input force u, and a spurious term $3x_1x_2^2$. A comparison of the posterior means ($\hat{\mu}_{\boldsymbol{\theta}}$) of the weights from the MCMC-SS algorithms reveals that the cubic term x_1^3 is the dominant term followed by x_1, u and x_2, while the weight associated with the spurious term $3x_1x_2^2$ is quite small. As observed previously, the SBL selected many more basis functions apart from the set of basis functions deemed relevant by the MCMC-SS algorithms. However, most of the spurious terms are associated with large standard deviations compared to their mean values, and may be disregarded based

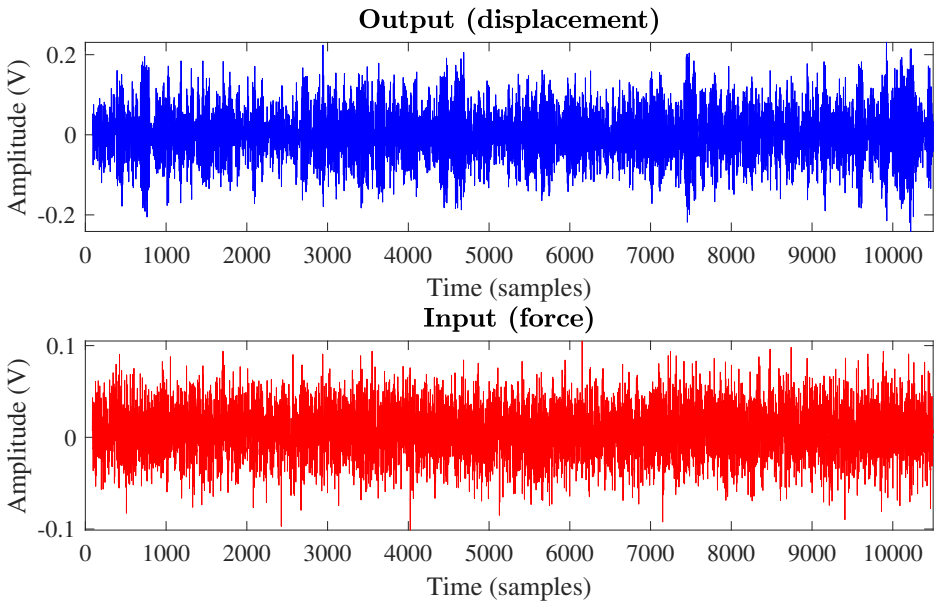


Figure 8: The portion of input-output measurement data from Silverbox that was used in training the sparse Bayesian algorithms; the input consists of random-phase multi-sine excitation.

on the degree of uncertainty. For example, the spurious term x_1^2 selected by SBL has a posterior standard deviation comparable to its posterior mean, and can be ignored.

To assess the predictive power of the discovered models, an independent input-output dataset – where the input excitation is a chirp signal going from high to low frequencies – was used for testing. The test input-output signals correspond to a set of 10500 force-displacement samples from the `Schroeder80mV.mat` dataset, and the 'true' test acceleration was obtained by numerical differentiation of the measured displacement data. To illustrate the prediction performance, the result from the MCMC-DSS-i algorithm is used as a representative, and the predicted mean and confidence intervals (CIs) of the test acceleration signal from the algorithm are plotted alongside the 'true' test acceleration signal in Figure 9 for a subset of 500 samples. The prediction results show a good match around resonance (bottom-right subplot of Figure 9), which occurs at lower frequencies of the down-chirp input. However, some discrepancies are seen at higher input frequencies (bottom-left subplot); nonetheless, the 'true' values are always captured by the predicted CIs. It was found that the test set prediction results were very similar for all the four sparse Bayesian algorithms, with the prediction errors being 0.1197 for SBL and 0.1192 for all three MCMC-SS algorithms. Note that, despite the inclusion of many spurious basis functions in the model found by SBL, the prediction error of SBL model appears to be not much different from that of the MCMC-SS models. It was found that prediction errors were largely dominated by the errors in the input time-series (which is selected in all models), while most of the chosen state-dependent basis functions, be they correct or spurious, played a dominant effect only near the region of resonance. As such, the prediction errors from SBL and MCMC-SS algorithms are largely similar except in a small region of resonance, where the MCMC-SS algorithms show a smaller prediction error than the SBL. Figure 10 illustrates this event by comparing the time-histories of test-set prediction errors from SBL and MCMC-DSS-i algorithms. A rectangular box surrounding the neighbourhood of resonance shows the prediction error of MCMC-DSS-i is lower than that from SBL. Due to this localised small difference,

Relevant variables	Estimated mean and standard derivations of unscaled weights ($\hat{\mu}_{\boldsymbol{\theta}} \pm \hat{\sigma}_{\boldsymbol{\theta}}$)			
	SBL	MCMC-CSS	MCMC-DSS-i	MCMC-DSS-g
$-x_1$	$(14.99 \pm 0.09) \times 10^4$	$(15.04 \pm 0.10) \times 10^4$	$(15.10 \pm 0.04) \times 10^4$	$(15.10 \pm 0.03) \times 10^4$
$-x_2$	26.29 ± 0.77	25.70 ± 1.24	25.32 ± 0.46	25.31 ± 0.45
$-2x_1x_2$	1.93 ± 3.85	0	0	0
$-x_1^2$	-1326.04 ± 1831.99	0	0	0
$-3x_1x_2^2$	0.94 ± 0.17	0.73 ± 0.22	0.69 ± 0.11	0.70 ± 0.08
$-3x_1^2x_2$	-34.45 ± 24.56	0	0	0
$-x_1^3$	$(35.67 \pm 7.33) \times 10^4$	$(38.47 \pm 8.97) \times 10^4$	$(40.62 \pm 2.85) \times 10^4$	$(40.69 \pm 1.94) \times 10^4$
$-10x_1^3x_2^2$	-2.57 ± 2.26	0	0	0
$-6x_1^5x_2$	-2484.65 ± 3006.48	0	0	0
$-\text{sgn}(x_2)$	-17.75 ± 17.76	0	0	0
$-x_1 \lvert x_1 \rvert$	$(15.12 \pm 16.55) \times 10^3$	0	0	0
u	$(10.17 \pm 0.04) \times 10^4$	$(10.17 \pm 0.04) \times 10^4$	$(10.17 \pm 0.04) \times 10^4$	$(10.17 \pm 0.04) \times 10^4$

Table 5: basis function selection and parameter estimation results for the Silverbox nonlinear benchmark using SBL and MCMC-SS algorithms; leftmost column enumerates the set of basis functions deemed relevant by at least one of the four sparse Bayesian algorithms, and the rest of columns show the posterior means and standard deviations of the estimated unscaled weights.

the results of prediction from SBL and MCMC-SS algorithms are very similar for the Silverbox benchmark.

In summary, Bayesian equation discovery performed on the Silverbox benchmark, showed that MCMC-SS algorithms are able to select fewer relevant basis functions compared to the SBL, while furnishing a comparable prediction error. The MCMC-SS algorithms selected a model with five basis functions, while the SBL included many more basis functions in addition to the ones selected by MCMC-SS algorithms. For test set model predictions, both the SBL and the MCMC-SS algorithms were found to closely match the true test set responses near resonance while there were greater discrepancies in predictions away from the region of resonance. Overall, the models inferred with MCMC-SS algorithms can be considered better with significantly lower number of basis functions and more interpretability.

5. Discussion

From the standpoint of equation discovery, the results of Bayesian basis function selection and parameter estimation using the MCMC-SS algorithms are quite encouraging. The estimated models (or governing equations) are not only more interpretable but also superior in prediction, compared to the SBL. However, unlike SBL, the MCMC-SS algorithms are based on random sampling and are orders of magnitude more expensive than the SBL. For faster implementation of Bayesian inference with SS priors, one may consider employing alternative methods such as expectation maximisation [60], variational Bayes [61–63], expectation

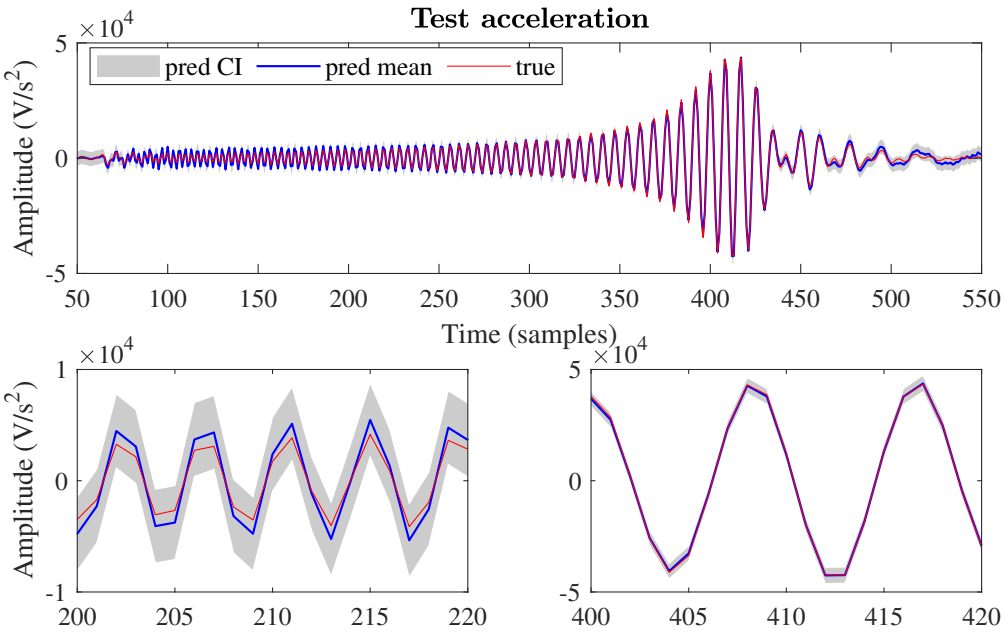


Figure 9: Plot of test set predictions of the Silverbox nonlinear benchmark using the MCMC-DSS-i algorithm (shown for a subset of 500 samples); testing with down-chirp input excitation. The top figure along with the "zoomed-in" bottom figures show the predicted mean and 3σ confidence interval (CI) of the test acceleration plotted alongside with 'true' acceleration. Predictions are better at lower input frequencies (bottom-right figure) than at higher input frequencies (bottom-left figure).

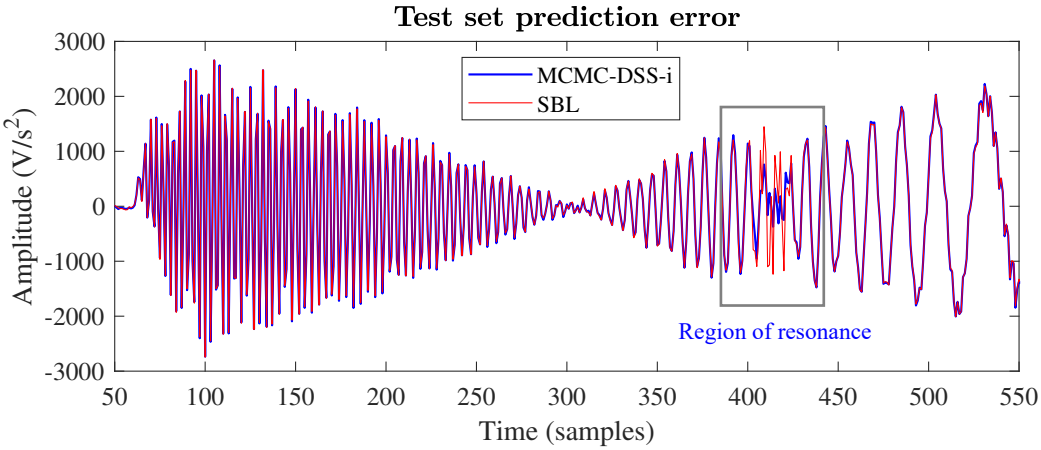


Figure 10: Plot of test set prediction errors from the SBL and the MCMC-DSS-i algorithms on the Silverbox nonlinear benchmark (shown for a subset of 500 samples). Prediction errors are largely similar except near resonance (marked by a rectangular box).

propagation [64], etc.

The dictionary of candidate basis functions plays a significant role in practical implementation of equation discovery algorithms. The success in discovering correct equations greatly hinges on whether or not the true basis functions are included in the dictionary. Absence of the true basis functions in the dictionary will lead to discoveries of terms that are strongly correlated with the true basis functions. For example, if $\sin(x)$ is a true variable that is not included in the dictionary, one will end up selecting correlated basis

functions such as $x, x^3, x^5, \ldots$ from a polynomial-based dictionary. Ideally, the dictionary should consist of as many diverse bases as possible, not just polynomial bases. In practice, however, putting more basis functions in the dictionary increases the correlation between basis functions and causes ill-conditioning of the dictionary matrix. The DSS-g prior was introduced to account for the correlation structure of the bases in the dictionary; its use was found to result in reduced uncertainty in the parameter estimates, however, the mean values of the parameter estimates were not found to be significantly different from that of the DSS-i prior. It was also found that a severely ill-conditioned dictionary can cause poor mixing (or even local entrapment) of the Markov chains and can induce incorrect selection of basis functions from a set of correlated basis functions. Parallel-tempering algorithms [65] can aid in improving the mixing of chains; however, they would significantly increase the computational burden. A reasonable approach would be to assess the dictionary for strongly-correlated basis functions prior to Bayesian inference, and if possible, eliminate some of them after careful deliberation. For example, basis functions x and $\sin(x)$ are highly correlated for small values of x, and one may choose to exclude $\sin(x)$ from the dictionary to prevent ill-conditioning, as has been done here. In the experience of the authors, the SBL is more robust in handling ill-conditioned dictionaries than MCMC-SS algorithms.

It should also be mentioned that the accuracy of the equation discovery approach can be greatly affected by errors in the state variables $\boldsymbol{x}$, present either in the form of measurement noise or state estimation errors. Since the basis functions are dependent on the states $\boldsymbol{x}$, even moderate amounts of errors in $\boldsymbol{x}$ can nonlinearly corrupt the constructed bases in the dictionary, and will eventually result in discovering incorrect equations. In the numerical study in Section 3, measurements of all three response variables, i.e., displacement, velocity and acceleration were assumed, and a relatively small noise was used to corrupt the measurements. While all three variables can be measured for a small system in a laboratory setup, it would be prohibitive to do so in practice, especially for large structural systems. A truly pragmatic approach would be to measure acceleration – the most commonly measured quantity in structural testing – and estimate the displacement and velocity from it. However, a naive numerical integration of the acceleration to obtain displacement or velocity may not work well, as even small amounts of noise in the displacement or velocity could deteriorate the equation discovery results. Recently, a promising optimisation framework was proposed in [66], that leverages automatic differentiation and sparse regression to simultaneously separate the noise signal from the measured data as well as recover the governing differential equations via numerical time-stepping constraints. This approach could be combined with a sparse Bayesian learning framework to make the equation discovery procedure more robust. Future efforts will look at developing robust approaches of equation discovery using acceleration measurements.

6. Conclusions

This paper presents a novel application of SS priors in Bayesian equation discovery of structural dynamic systems, which aims at discovering the governing ordinary differential equations of motion of a structural system from measured input-output data. The equation discovery procedure is tantamount to a simultaneous model selection and parameter estimation problem in system identification. Using a dictionary of nonlinear bases variables composed using the measured data, the problem of Bayesian model selection is turned

into a Bayesian basis function selection problem and solved via sparse linear regression, thus bypassing a combinatorially large search through all possible candidate models. The SS priors are well-known to possess superior sparsity-enforcing properties compared to the Laplace or Student's-t priors, owing to their two-component mixture distributions of a narrow spike and a comparably flat slab. As such, their use in basis function selection has the potential to derive more parsimonious and interpretable equations of motion. In this paper, three different variants of SS priors – namely the CSS, DSS-i and DSS-g – are employed as prior distributions, and MCMC-based Gibbs sampling algorithms are derived to select the relevant variables and estimate associated parameters.

Using a series of numerical simulations, it has been demonstrated that the proposed MCMC-SS algorithms correctly identify the presence and type of various nonlinearities such as a cubic stiffness, a quadratic viscous damping, and a Coulomb friction damping. Furthermore, using Monte Carlo simulations, the performance of MCMC-SS has been compared to SBL, which uses the Student's-t prior. It is found that MCMC-SS algorithms display stronger model selection consistency than SBL. Additionally, the predictive accuracy of the models selected by MCMC-SS is found to be highly competitive to that of the models estimated by SBL.

7. Acknowledgements

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 through grant number EP/S001565/1 is also gratefully acknowledged.

Appendix A. Gibbs sampling scheme for CSS prior

The Gibbs sampling steps for the CSS prior are adopted from the BASAD algorithm [47]. Note that, as mentioned in Section 2.1, the CSS priors involve a vector of slab variances, that is, a slab variance v_{s_i} is associated with each weight θ_i. The steps for the parameters $\boldsymbol{\theta}$, $\boldsymbol{z}$, p_0, v_{s_i} and σ^2 are as follows:

(a) $\boldsymbol{\theta}$ is sampled from a Gaussian distribution as follows,

$$\boldsymbol{\theta} \mid \boldsymbol{y}, \boldsymbol{z}, v_s, \sigma^2 \sim \mathcal{N}\left(\boldsymbol{\mu}, \sigma^2 \boldsymbol{\Sigma}\right) \tag{A.1}$$

where $\boldsymbol{\Sigma} = \left(\mathbf{D}^T\mathbf{D} + \mathbf{V}^{-1}\right)^{-1}$, $\boldsymbol{\mu} = \boldsymbol{\Sigma}\mathbf{D}^T\boldsymbol{y}$, and $\mathbf{V}$ is a diagonal matrix with elements $\mathbf{V}_{i,i} = v_{s_i}\left(v_0(1-z_i) + v_1 z_i\right)$, $i = 1, \ldots, P$.

(b) σ^2 is sampled from an inverse Gamma distribution as follows,

$$\sigma^2 \mid \boldsymbol{y}, \boldsymbol{\theta}, v_{s_i} \sim \mathcal{IG}\left(a_\sigma + \frac{N}{2} + \frac{P}{2},\ b_\sigma + \frac{1}{2}\left(\left(\boldsymbol{y} - \mathbf{D}\boldsymbol{\theta}\right)^T\left(\boldsymbol{y} - \mathbf{D}\boldsymbol{\theta}\right) + \boldsymbol{\theta}^T\mathbf{V}^{-1}\boldsymbol{\theta}\right)\right) \tag{A.2}$$

(c) The vector of weight-specific slab variances is sampled componentwise. The i^{th} component, v_{s_i}, is sampled from an inverse Gamma distribution as follows,

$$v_{s_i} \mid \boldsymbol{\theta}, \boldsymbol{z}, \sigma^2 \sim \mathcal{IG}\left(a_v + \frac{1}{2},\ b_v + \frac{0.5\theta_i^2}{2\sigma^2\left(v_0(1-z_i) + v_1 z_i\right)}\right) \tag{A.3}$$

(d) p_0 is sampled from a Beta distribution as follows,

$$p_0 \mid \boldsymbol{z} \sim \text{Beta}\left(a_p + \sum_{i=1}^{P} z_i,\ b_p + P - \sum_{i=1}^{P} z_i\right) \tag{A.4}$$

(e) The conditional distribution of $\boldsymbol{z}$ is expressed componentwise. The odds of $z_i = 1$ to $z_i = 0$ are computed. The components of $\boldsymbol{z}$ are sampled as follows,

$$z_i \mid \theta_i, v_{s_i}, \sigma^2, p_0 \sim \text{Bern}(\xi_i),\ \text{with } \xi_i = \frac{p_0}{p_0 + \frac{p\left(\theta_i \mid z_i=0, v_{s_i}, \sigma^2\right)}{p\left(\theta_i \mid z_i=1, v_{s_i}, \sigma^2\right)}(1-p_0)} \tag{A.5}$$

In the above sampling step, the probabilities $p\left(\theta_i \mid z_i = 0, v_{s_i}, \sigma^2\right)$ and $p\left(\theta_i \mid z_i = 1, v_{s_i}, \sigma^2\right)$ can be computed by evaluating the Gaussian densities over θ_i, as follows:

$$p\left(\theta_i \mid z_i = 0, v_{s_i}, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2 v_0 v_{s_i}}} \exp\left(-\frac{\theta_i^2}{2\sigma^2 v_0 v_{s_i}}\right)$$
$$p\left(\theta_i \mid z_i = 1, v_{s_i}, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2 v_1 v_{s_i}}} \exp\left(-\frac{\theta_i^2}{2\sigma^2 v_1 v_{s_i}}\right)$$

References

[1] J. Bongard, H. Lipson, Automated reverse engineering of nonlinear dynamical systems, Proceedings of the National Academy of Sciences 104 (24) (2007) 9943–9948.

[2] M. Schmidt, H. Lipson, Distilling free-form natural laws from experimental data, Science 324 (5923) (2009) 81–85.

[3] S. L. Brunton, J. L. Proctor, J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proceedings of the National Academy of Sciences 113 (15) (2016) 3932–3937.

[4] N. M. Mangan, S. L. Brunton, J. L. Proctor, J. N. Kutz, Inferring biological networks by sparse identification of nonlinear dynamics, IEEE Transactions on Molecular, Biological and Multi-Scale Communications 2 (1) (2016) 52–63.

[5] H. Schaeffer, S. G. McCalla, Sparse model selection via integral terms, Physical Review E 96 (2) (2017) 023302.

[6] N. M. Mangan, J. N. Kutz, S. L. Brunton, J. L. Proctor, Model selection for dynamical systems via sparse regression and information criteria, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 473 (2204) (2017) 20170009.

[7] E. Kaiser, J. N. Kutz, S. L. Brunton, Sparse identification of nonlinear dynamics for model predictive control in the low-data limit, Proceedings of the Royal Society A 474 (2219) (2018) 20180335.

[8] K. Champion, B. Lusch, J. N. Kutz, S. L. Brunton, Data-driven discovery of coordinates and governing equations, Proceedings of the National Academy of Sciences 116 (45) (2019) 22445–22451.

[9] H. Schaeffer, G. Tran, R. Ward, L. Zhang, Extracting structured dynamical systems using sparse optimization with very few samples, Multiscale Modeling & Simulation 18 (4) (2020) 1435–1461.

[10] L. Boninsegna, F. Nüske, C. Clementi, Sparse learning of stochastic dynamical equations, The Journal of Chemical Physics 148 (24) (2018) 241723.

[11] N. M. Mangan, T. Askham, S. L. Brunton, J. N. Kutz, J. L. Proctor, Model selection for hybrid dynamical systems via sparse regression, Proceedings of the Royal Society A 475 (2223) (2019) 20180534.

[12] M. Stender, S. Oberst, N. Hoffmann, Recovery of differential equations from impulse response time series data for model identification and feature extraction, Vibration 2 (1) (2019) 25–46.

[13] S. H. Rudy, S. L. Brunton, J. L. Proctor, J. N. Kutz, Data-driven discovery of partial differential equations, Science Advances 3 (4) (2017) e1602614.

[14] S. Zhang, G. Lin, Robust data-driven discovery of governing physical laws with error bars, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 474 (2217) (2018) 20180305.

[15] Z. Chen, Y. Liu, H. Sun, Deep learning of physical laws from scarce data, arXiv preprint arXiv:2005.03448.

[16] S. H. Rudy, J. N. Kutz, S. L. Brunton, Deep learning of dynamics and signal-noise decomposition with time-stepping constraints, Journal of Computational Physics 396 (2019) 483–506.

[17] M. Raissi, P. Perdikaris, G. E. Karniadakis, Multistep neural networks for data-driven discovery of nonlinear dynamical systems, arXiv preprint arXiv:1801.01236.

[18] M. Raissi, Deep hidden physics models: Deep learning of nonlinear partial differential equations, The Journal of Machine Learning Research 19 (1) (2018) 932–955.

[19] G. Kerschen, K. Worden, A. F. Vakakis, J.-C. Golinval, Past, present and future of nonlinear system identification in structural dynamics, Mechanical Systems and Signal Processing 20 (3) (2006) 505–592.

[20] J.-P. Noël, G. Kerschen, Nonlinear system identification in structural dynamics: 10 more years of progress, Mechanical Systems and Signal Processing 83 (2017) 2–35.

[21] H. Akaike, A new look at the statistical model identification, IEEE Transactions on Automatic Control 19 (6) (1974) 716–723.

[22] G. E. Schwarz, Estimating the dimension of a model, The Annals of Statistics 6 (2) (1978) 461–464.

[23] T. Hastie, R. Tibshirani, M. Wainwright, Statistical Learning with Sparsity: the Lasso and Generalizations, CRC press, 2015.

[24] M. E. Tipping, Sparse Bayesian learning and the relevance vector machine, Journal of Machine Learning Research 1 (2001) 211–244.

[25] D. P. Wipf, B. D. Rao, Sparse Bayesian learning for basis selection, IEEE Transactions on Signal processing 52 (8) (2004) 2153–2164.

[26] M. W. Seeger, H. Nickisch, R. Pohmann, B. Schölkopf, Optimization of k-space trajectories for compressed sensing by Bayesian experimental design, Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine 63 (1) (2010) 116–126.

[27] M. W. Seeger, Bayesian inference and optimal design for the sparse linear model, Journal of Machine Learning Research 9 (2008) 759–813.

[28] C. M. Carvalho, N. G. Polson, J. G. Scott, Handling sparsity via the horseshoe, in: Artificial Intelligence and Statistics, 2009, pp. 73–80.

[29] T. J. Mitchell, J. J. Beauchamp, Bayesian variable selection in linear regression, Journal of the American Statistical Association 83 (404) (1988) 1023–1032.

[30] E. I. George, R. E. McCulloch, Variable selection via Gibbs sampling, Journal of the American Statistical Association 88 (423) (1993) 881–889.

[31] J. Geweke, Variable selection and model comparison in regression, In Bayesian Statistics 5: Proceedings of the Fifth Valencia International Meeting (1996) 609–620.

[32] E. I. George, R. E. McCulloch, Approaches for Bayesian variable selection, Statistica Sinica (1997) 339–373.

[33] H. Ishwaran, J. S. Rao, Spike and slab variable selection: Frequentist and Bayesian strategies, The Annals of Statistics 33 (2) (2005) 730–773.

[34] N. G. Polson, J. G. Scott, Shrink globally, act locally: Sparse Bayesian regularization and prediction, Bayesian statistics 9 (501-538) (2010) 105.

[35] S. van Erp, D. L. Oberski, J. Mulder, Shrinkage priors for Bayesian penalized regression, Journal of Mathematical Psychology 89 (2019) 31–50.

[36] R. Fuentes, N. Dervilis, K. Worden, E. J. Cross, Efficient parameter identification and model selection in nonlinear dynamical systems via sparse Bayesian learning, in: Journal of Physics: Conference Series, Vol. 1264, IOP Publishing, 2019, p. 012050.

[37] R. Fuentes, R. Nayek, P. Gardner, N. Dervilis, T. Rogers, K. Worden, E. J. Cross, Equation discovery for nonlinear dynamical systems: A bayesian viewpoint, Mechanical Systems and Signal Processing 154 (2021) 107528.

[38] S. Zhang, G. Lin, Robust data-driven discovery of governing physical laws using a new subsampling-based sparse Bayesian method to tackle four challenges (large noise, outliers, data integration, and extrapolation), arXiv preprint arXiv:1907.07788.

[39] H.-Q. Mu, K.-V. Yuen, Modal frequency-environmental condition relation development using long-term structural health monitoring measurement: Uncertainty quantification, sparse feature selection and multivariate prediction, Measurement 130 (2018) 384–397.

[40] Y. Huang, J. Yu, J. L. Beck, H. Zhu, H. Li, Novel sparseness-inducing dual Kalman filter and its application to tracking time-varying spatially-sparse structural stiffness changes and inputs, Computer Methods in Applied Mechanics and Engineering 372 (2020) 113411.

[41] Y. Huang, J. L. Beck, H. Li, Bayesian system identification based on hierarchical sparse Bayesian learning and Gibbs sampling with application to structural damage assessment, Computer Methods in Applied Mechanics and Engineering 318 (2017) 382–411.

[42] Z. Chen, R. Zhang, J. Zheng, H. Sun, Sparse bayesian learning for structural damage identification, Mechanical Systems and Signal Processing 140 (2020) 106689.

[43] Y.-Q. Ni, Q.-H. Zhang, A Bayesian machine learning approach for online detection of railway wheel defects using track-side monitoring, Structural Health Monitoring (2021) 1475921720921772.

[44] W. Feng, Q. Li, Q. Lu, Force localization and reconstruction based on a novel sparse Kalman filter, Mechanical Systems and Signal Processing 144 (2020) 106890.

[45] R. Nayek, K. Worden, E. J. Cross, R. Fuentes, A sparse Bayesian approach to model structure selection and parameter estimation of dynamical systems using spike-and-slab priors, in: Proceedings of the 8th International Conference on Noise and Vibration Engineering (ISMA2020), 2020.

[46] R. B. O'Hara, M. J. Sillanpää, A review of Bayesian variable selection methods: what, how and which, Bayesian analysis 4 (1) (2009) 85–117.

[47] N. N. Narisetty, X. He, Bayesian variable selection with shrinking and diffusing priors, The Annals of Statistics 42 (2) (2014) 789–817.

[48] F. Liang, R. Paulo, G. Molina, M. A. Clyde, J. O. Berger, Mixtures of g-priors for Bayesian variable selection, Journal of the American Statistical Association 103 (481) (2008) 410–423.

[49] D. F. Andrews, C. L. Mallows, Scale mixtures of normal distributions, Journal of the Royal Statistical Society: Series B (Methodological) 36 (1) (1974) 99–102.

[50] G. Malsiner-Walli, H. Wagner, Comparing spike and slab priors for Bayesian variable selection, Austrian Journal of Statistics 40 (4) (2011) 241–264.

[51] G. Casella, E. I. George, Explaining the Gibbs sampler, The American Statistician 46 (3) (1992) 167–174.

[52] M. M. Barbieri, J. O. Berger, Optimal predictive model selection, The Annals of Statistics 32 (3) (2004) 870–897.

[53] K. Worden, G. R. Tomlinson, Nonlinearity in Structural Dynamics: Detection, Identification and Modelling, CRC Press, 2019.

[54] P. R. Dahl, Solid friction damping of mechanical vibrations, AIAA Journal 14 (12) (1976) 1675–1682.

[55] S. P. Brooks, A. Gelman, General methods for monitoring convergence of iterative simulations, Journal of computational and graphical statistics 7 (4) (1998) 434–455.

[56] M. E. Tipping, A. C. Faul, Fast marginal likelihood maximisation for sparse Bayesian models, in: Proceedings of the Ninth AISTATS Conference, 2003, pp. 1–13.

[57] SparseBayes Software v2.0, http://www.miketipping.com/downloads.htm, Last Accessed: 2020-11-10.

[58] T. Wigren, J. Schoukens, Three free data sets for development and benchmarking in nonlinear system identification, in: 2013 European control conference (ECC), IEEE, 2013, pp. 2933–2938.

[59] T. Wigren, J. Schoukens, Data for benchmarking in nonlinear system identification, Technical Reports from the department of Information Technology 6 (2013) 2013–006.

[60] V. Ročková, E. I. George, EMVS: The EM approach to Bayesian variable selection, Journal of the American Statistical Association 109 (506) (2014) 828–846.

[61] M. K. Titsias, M. Lázaro-Gredilla, Spike and slab variational inference for multi-task and multiple kernel learning, in: Advances in Neural Information Processing Systems, 2011, pp. 2339–2347.

[62] P. Carbonetto, M. Stephens, Scalable variational inference for Bayesian variable selection in regression, and its accuracy in genetic association studies, Bayesian Analysis 7 (1) (2012) 73–108.

[63] J. T. Ormerod, C. You, S. Müller, A variational Bayes approach to variable selection, Electronic Journal of Statistics 11 (2) (2017) 3549–3594.

[64] J. M. Hernández-Lobato, D. Hernández-Lobato, A. Suárez, Expectation propagation in linear regression models with spike-and-slab priors, Machine Learning 99 (3) (2015) 437–487.

[65] D. J. Earl, M. W. Deem, Parallel tempering: Theory, applications, and new perspectives, Physical Chemistry Chemical Physics 7 (23) (2005) 3910–3916.

[66] K. Kaheman, S. L. Brunton, J. N. Kutz, Automatic differentiation to simultaneously identify nonlinear dynamics and extract noise probability distributions from data, arXiv preprint arXiv:2009.08810.