Mechanical Systems and Signal Processing 154 (2021) 107528

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

journal homepage: www.elsevier.com/locate/ymssp

Equation discovery for nonlinear dynamical systems: A Bayesian viewpoint

R. Fuentes, R. Nayek, P. Gardner, N. Dervilis, T. Rogers, K. Worden, E.J. Cross *

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

ARTICLE INFO

Article history:
Received 5 February 2020
Received in revised form 4 October 2020
Accepted 3 December 2020
Available online 10 January 2021

Keywords:
Equation discovery
Nonlinear system identification
Sparse Bayesian learning
Relevance Vector Machine (RVM)

ABSTRACT

This paper presents a new Bayesian approach to equation discovery – combined structure detection and parameter estimation – for system identification (SI) in nonlinear structural dynamics. The structure detection is accomplished via a sparsity-inducing prior within a Relevance Vector Machine (RVM) framework; the prior ensures that terms making no contribution to the model are driven to zero coefficient values. Motivated by the idea of compressive sensing (CS) and recent results from the machine learning community on sparse linear regression, the paper adopts the use of an over-complete dictionary to represent a large number of candidate terms for the equation describing the system. Unlike other sparse learners, like the Lasso and its derivatives, which are potentially sensitive to hyperparameter selection, the proposed method exploits the principled means of fixing priors and hyperpriors that are available via a hierarchical Bayesian approach. The approach is successfully demonstrated and validated via a number of simulated case studies of common Single-Degree-of-Freedom (SDOF) nonlinear dynamic systems, and on two challenging experimental data sets.

1. Introduction

The task of identifying equations that correctly describe an observed system's dynamics has been of fundamental interest to the scientific and engineering communities for many years. Historically, this task has involved the combination of empirical observations with a great deal of scientific wit. However, modern problems increasingly call for predictive capability (precision and accuracy) of models beyond what can usually be achieved by human inference alone. Recently, increased attention has been drawn to the use of statistical and machine learning models for the problem of *system identification* (SI), which refers to the task of correctly identifying the model forms, and the parameters of such models that best describe the system dynamics. The field was driven forward largely by the electrical engineering and control systems communities [1], where accurate predictive models are central to the design of successful control strategies and algorithms. The field of system identification, however, spans applications within the physical sciences [2], biological systems [3–6], mechanical and mechatronic systems [7–9] and fluid dynamics [10].

As one might expect, terminology differs across the different application domains. Within the mechanical and civil engineering communities, system identification has sometimes been referred to as *model updating* [11], in the statistical community (with its associated application domains), SI is referred to as *model calibration* [12]. The problem is often discussed in

* Corresponding author.
E-mail address: e.j.cross@sheffield.ac.uk (E.J. Cross).

https://doi.org/10.1016/j.ymssp.2020.107528
0888-3270/

terms of two sub-problems: the problem of determining the functional form of the equations of motion is often termed *structure detection*, while the problem of estimating any undetermined parameters within that form is called *parameter estimation* [13]. Another name for the problem of addressing both issues at the same time – and the term of choice in the current paper – is *equation discovery*. In all cases, the methods make use of measured data from the system of interest, and are, therefore *machine learning* methods [14].

The key question under investigation in this paper is of how to accurately and simultaneously recover the correct equations of motion of a dynamical system together with the associated parameters, at the same time. Naturally, combined model selection and parameter estimation is significantly more challenging, with selection of model complexity a particular issue. Bayesian inference has emerged as a powerful tool to address exactly this type of problem; it has been studied in the field of system identification owing to its natural ability to quantify uncertainty in parameter estimates [15,9]. This uncertainty quantification leads directly to the idea of Bayesian model comparison [16,17], where one seeks to compare the quality of fit of different models according to *posterior* probability distributions (after observing evidence) over them. In general, the inference over entire probability distributions that the Bayesian approach allows, brings the advantage that one is able to quantify the uncertainty in the parameter estimates, and potentially propagate this into confidence intervals on predictions [9]. Furthermore, Bayesian estimation methods offer some natural protection from *overfitting*, which is a common problem in engineering contexts where data may be scarce. Bayesian methods generally implement a form of 'Occam's razor' which controls model complexity [16]. The current paper is concerned with methods which amplify the 'Occam' capability in promoting sparse solutions.

This paper provides an approach for equation discovery of parametric models based on Bayesian inference; structure detection will be achieved by selecting terms from a predetermined dictionary. This dictionary could include for example, a constant offset, linear and polynomial terms, as well as trigonometric and discontinuous functions. Upon selection of a dictionary, the learning problem requires the solution of the linear problem,

$$\mathbf{y} = \mathbf{D}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{1}$$

where $\mathbf{y}$ is the target of interest, $\mathbf{D}$ is a matrix with each column corresponding to the inputs transformed by an element of the dictionary, $\boldsymbol{\beta}$, collects the weights corresponding to each dictionary element, and $\boldsymbol{\epsilon}$ is a residual error[1]. This form will be reintroduced with more rigour in the main body of the paper. Naturally, this problem is computationally nontrivial, as the dictionary may contain a very large candidate pool. The problem is one of *subset selection* in combinatorial optimisation terms and is *NP*-hard; further, the multiplying parameter for each term must be estimated.

Classical approaches to system identification make heavy use of techniques derived from linear least-squares regression to solve problems of the type posed in Eq. (1) [1], where the assumed model form is implicit in the design matrix used, and the algorithm returns the vector of parameters $\boldsymbol{\beta}$. However, if a large number M, of candidate basis vectors are included in the design matrix, one tends to encounter two problems. The first issue is that the problem is under-determined without a large number of training samples to ensure a numerically-stable solution. The second, and perhaps more important problem, is that candidate basis vectors that may not really belong to the model will still have a non-zero contribution to the solution. The main adverse effect from this is that the remaining parameters for the terms that *should* be in the model may be biased. In a nonparametric model, bias is not an issue; however, if the terms in the dictionary are considered physically meaningful, they will be assigned non-physical values. Furthermore, the biased model may not generalise well to predictions in unseen circumstances.

This paper presents an approach to solving the type of problem described by Eq. (1), which sets to zero any parameters associated with basis functions that should not be in the model. The main problem is then to determine which basis functions *should* be present; some condition is needed. Various methods have arisen in the statistical learning communities that can deal with this type of problem efficiently. Many methods adopt the principle that one should use the smallest number of terms possible from the dictionary, i.e. M is small and the solution is *sparse*. One key idea is to introduce a penalty term on the traditional least-squares cost function, which leads to an optimisation problem,

$$\text{minimise} : \left\{ \frac{1}{2N} ||\mathbf{y} - \mathbf{D}\boldsymbol{\beta}||_2^2 + \lambda ||\boldsymbol{\beta}||_p \right\} \tag{2}$$

which expresses the penalty over the parameter vector with a general p-norm, weighted by a hyperparameter λ; N is the number of training data points. The value of this norm controls how much the weights will be driven towards smaller values. The ℓ_0 ($p = 0$) norm is the ideal case for sparse learning as it essentially counts the number of terms in the vector; however, this leads to a combinatorially-hard problem and also requires non-convex optimisation to solve, making it impractical. Setting $p = 1$ leads to the well-known *Least absolute shrinkage and selection operator* (Lasso) [18]. This ℓ_1-regularisation has become a popular choice for problems where sparse solutions are sought, as it is effective at shrinking to zero terms that do not contribute; furthermore, it can be solved using convex optimisation..

Sparse linear regression has recently been investigated in the specific context of equation discovery; [19] shows that the Lasso can be used effectively for basis selection in learning parsimonious representations of nonlinear dynamical systems. The main drawback of the approach is that the level of sparsity (number of non-zero components) of the solution is com-

[1] Throughout this paper, vectors will be denoted by boldface symbols, while matrices will be represented by boldface capital letters.

pletely dictated by the hyperparameter λ, in Eq. (2). High (resp. low) values of λ lead to more (resp. less) sparse solutions. This issue means that discovery of the 'correct' equations describing the dynamics depends critically on a 'correct' choice of this hyperparameter. In [19], this issue is side-stepped by manual selection of a threshold that yields a suitable number of basis vectors for each problem. In principle, there are ways of automatically tuning λ that would yield optimal levels of sparsity, *cross-validation* being one of the simplest and most effective [20]; however, other methods exist, such as the Bayesian Lasso [21].

Further to the issue of having to select a sparsity level, another problem with regularised regression from a system identification perspective, concerns the accuracy of the solution. The use of regularisation explicitly implies that the parameter estimates $\boldsymbol{\beta}$ of the model will be biased. As discussed above, this issue does help with overfitting, but will adversely impact the physical interpretation of the parameters.

In [22], a similar problem formulation was presented by evaluating different candidate functional forms in a state space representation and using a combination of symbolic regression and genetic programming to find the functional forms that best matched an observed time series, while also respecting conservation laws. While the general idea was sound, the approach to optimisation lacked a natural balance between complexity and predictive accuracy. A genetic program is not guaranteed to prefer solutions that are simple or parsimonious – this is the problem of *bloat* in genetic programming. This paper builds on some of the ideas of [22,19] in terms of problem formulation, but deviates in terms of solution.

The focus of this paper will be to address these key issues in equation discovery by the use of Bayesian inference. Apart from uncertainty quantification of the estimated parameters, a Bayesian approach offers two additional advantages over deterministic approaches: (a) the prior distributions used in a Bayesian approach naturally allow for penalisation of the parameters and (b) the penalty parameter is simultaneously estimated with other model parameters and does not require any additional step such as cross-validation. The approach to Bayesian inference with sparsity investigated here will be the *Relevance Vector Machine* (RVM) [23], devised originally as a sparse probabilistic alternative to the Support Vector Machine (SVM).

The RVM was originally designed for nonparametric learning using kernels, but has proved generally to be a powerful algorithm for problems of basis selection [24]. The principle behind the RVM is similar to that of ℓ_1-regularised regression; it yields sparse solutions as it relies on a Student-t prior distribution that is sharply peaked around zero (the Lasso can be shown to be a special case of Bayesian inference under the assumption of a Laplace prior [20].) The RVM is arguably a better sparse solver than the Lasso, as it is probabilistic, and there exist efficient algorithms that solve the basis selection problem with the RVM, without resorting to thresholds or tuning parameters. The RVM has been already used in some SI contexts; in [25] it was used to select sparse sets of lags within a polynomial *Nonlinear Autoregressive with eXogenous inputs* (NARX) model. The RVM has also been suggested for the specific task of equation discovery in dynamical systems in [26], on which the current work builds. A similar approach has also been suggested for this task in [27], where a probabilistic approach is taken, but based on ℓ_1-regularisation.

The case studies presented here are restricted to Single-Degree-of-freedom (SDOF) systems, and this is sufficient to introduce and discuss the main aspects of the new approach. Moving to Multi-Degree-of-Freedom (MDOF) systems would only increase the size of the dictionaries somewhat and introduce some additional correlations between terms. As the sizes of the dictionaries used in this paper are already quite substantial, one would not expect a move to MDOF systems to introduce any new qualitative technical difficulties.

The layout of the paper is as follows: Section 2 will outline the key elements of sparse Bayesian computation. Section 3 presents a series of numerical experiments on several systems that are of interest in nonlinear dynamics, and Section 4 will present analysis of two real experimental systems. Section 5 provides a critical discussion of the main results, focussing on the practical limitations of the proposed approach. Finally, Section 6 presents the conclusions of the paper.

2. Sparse Bayesian equation discovery

The introductory section briefly stated that the learning problem reduces to the solution of the matrix Eq. (1). This result is introduced more rigorously here, before continuing to discuss the sparse solver proposed for use here.

2.1. Problem formulation

The problem at hand is system identification of a dynamical system. In order to constrain the problem somewhat, a state-space (first-order) representation of systems will be adopted of the form,

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{u}(t) + \boldsymbol{\epsilon}(t) \tag{3}$$

where $\mathbf{x}(t)$ is the *state vector* of system response variables, $\mathbf{u}(t)$ is an external forcing function, and $\boldsymbol{\epsilon}(t)$ is a residual (error) term which can take into account measurement noise etc. Overdots denote differentiations with respect to time. From this point onwards, the explicit time dependence of $\mathbf{x}$ will generally be omitted for notational simplicity.

The form in Eq. (3) assumes that the forcing enters linearly, which is usually the case in structural dynamics. A completely general form of the equation would replace the RHS with $f(\mathbf{x}(t), \mathbf{u}(t))$. However, as with the move from SDOF to MDOF sys-

tems, accommodating the general form does not require a change in the approach presented here; again, one would only expect an increased size for the dictionary in order to include more candidate terms involving u.

In this paper, the unknown function $\mathbf{f}$ will be approximated by a linear superposition over a set of functions from a pre-specified *dictionary*, so that,

$$\dot{\mathbf{x}}(t) = \boldsymbol{\beta}_1 d_1(\mathbf{x}) + \boldsymbol{\beta}_2 d_2(\mathbf{x}) + \ldots + \boldsymbol{\beta}_M d_M(\mathbf{x}) + \mathbf{u}(t) + \boldsymbol{\epsilon}(t) \tag{4}$$

where the $\boldsymbol{\beta}_i$ are the parameters for estimation and the functions $d_i(\mathbf{x})$ are the entries in the dictionary $\mathbf{D} = \{d_1(\mathbf{x}), \ldots, d_M(\mathbf{x})\}$.

To further simplify the analysis and to explain how the problem can be reduced to a linear regression problem, it will be assumed that the systems of interest can be represented by Single-Degree-of-Freedom (SDOF) oscillators of the form,

$$m\ddot{y} + c\dot{y} + g(y, \dot{y}) = u(t) \tag{5}$$

where $\{m, c\}$ are the usual mass and damping coefficients, and g is an arbitrary nonlinear function of displacement y and velocity $\dot{y}$. The state-space equations for this system are simply,

$$\dot{x}_1 = x_2 \tag{6}$$

$$\dot{x}_2 = \frac{1}{m}(u(t) - kx_1 - cx_2 - g(x_1, x_2)) \tag{7}$$

on identifying x_1 with displacement y. As the first equation is simply the definition of velocity, the equation discovery problem in Eq. (4) has been reduced to the problem of identifying,

$$\dot{x}_2 = \beta_1 d_1(x_1, x_2) + \beta_2 d_2(x_1, x_2) + \ldots + \beta_M d_M(x_1, x_2) + u(t) + \epsilon(t) \tag{8}$$

Simply adding the function $u(t)$ to the dictionary as an element $d_0 = u(t)$, produces,

$$\dot{x}_2 = \beta_0 d_0 + \beta_1 d_1(x_1, x_2) + \beta_2 d_2(x_1, x_2) + \ldots + \beta_M d_M(x_1, x_2) + \epsilon(t) \tag{9}$$

This is now a completely standard linear regression problem, given measurements of the specified variables. A training set of sampled time data will be assumed of the form $T = \{x_{1,i}, x_{2,i}, u_i; i = 1, \ldots, N\}$, and yields the matrix problem stated in Eq. (1), restated here for convenience,

$$\mathbf{y} = \mathbf{D}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{10}$$

with $\mathbf{y} = (\dot{x}_{2,1}, \ldots, \dot{x}_{2,N})^T$, $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_M)^T$, and the i^{th} column of the matrix $\mathbf{D}$ is the vector of values $(d_i(x_{1,1}, x_{2,1}), \ldots, d_i(x_{1,N}, x_{2,N}))^T$. In this case, $\boldsymbol{\epsilon}$ is the vector of model residuals per sample point. Note that this can be adapted if the target of interest is multivariate.

2.2. Inducing sparsity

The problem formulation here requires the use of a sparse solver which selects only the columns of $\mathbf{D}$ in Eq. (1) which make significant contributions to the overall model. The sparse Bayesian learning approach adopted here is specifically designed for this type of ill-posed linear estimation problem [23].

This main idea is illustrated in Fig. 1, which shows the solution to Eq. (1) using sparse Bayesian learning, where the problem of interest has the true nonlinear function $g(y, \dot{y}) = ky + k_3 y^3$ (Duffing's equation), where k and k_3 are the linear and cubic stiffness coefficients respectively. The measured data in this illustration are from a free-decay of the system ($u(t) = 0$) and the dictionary adopted has many more terms than those present in the 'true' equation. The figure shows how each column of the design matrix $\mathbf{D}$ would contribute to the model, and shows the magnitudes of the parameters determined by the sparse solver (which will be described in detail in the next section). Recall that the variables x_1 and x_2 in the dictionary terms correspond to the system displacement and velocity respectively. The solver has estimated only three non-zero parameters, those corresponding to x_1 (displacement – linear stiffness term), x_2 (velocity – linear viscous damping term) and x_1^3 (cubic stiffness term), exactly as required.

Sparse learning is used here to provide a solution to the problem of Eq. (1), that switches off any columns of $\mathbf{D}$ that do not significantly contribute to the observed dynamics. The Bayesian approach adopted here can also derive posterior probability distributions over the model parameters $\boldsymbol{\beta}$ and predictive outputs. The particular algorithm used here is the *Relevance Vector Machine* (RVM) [23], outlined in the next section.

2.2.1. Formulation of the RVM

The presentation here essentially follows that of Tipping [23]. The RVM is required to select only a sufficient and necessary number of columns in $\mathbf{D}$ in Eq. (1) (relevance vectors in this context), that explain the observed data well. The observations of the model are assumed to be corrupted with noise and are modelled by a target vector, $\mathbf{t}$,

$$\mathbf{t} = \mathbf{y} + \boldsymbol{\epsilon} \tag{11}$$

where $\boldsymbol{\epsilon}$ is the residual/noise term.

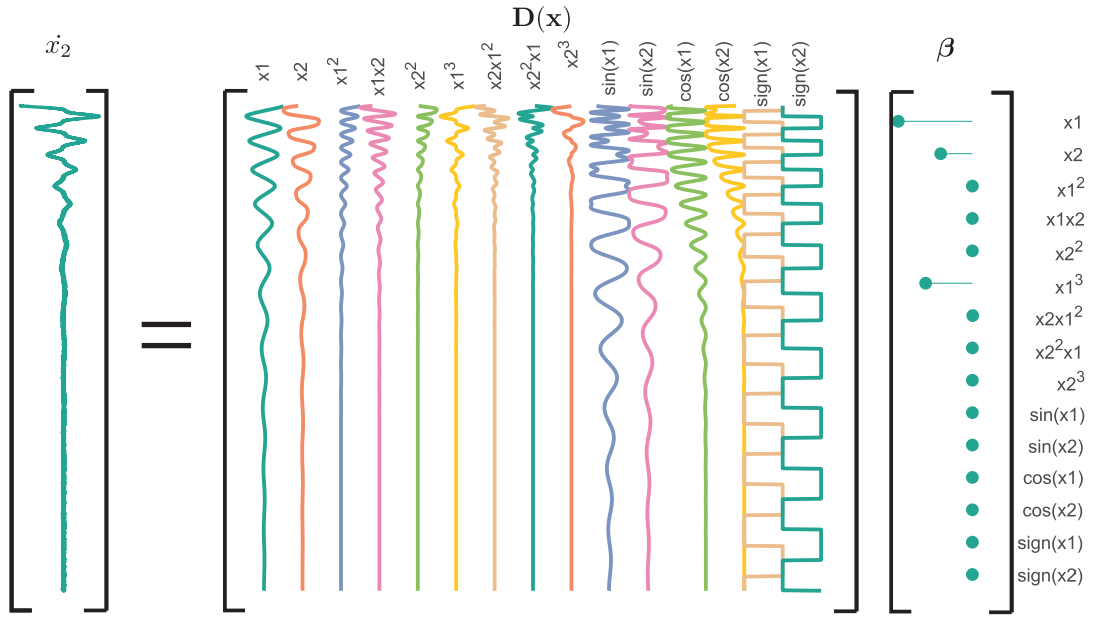

Fig. 1. Illustration of the problem formulation on a free-decaying Duffing oscillator. The second state derivative (acceleration) is given in terms of candidate polynomial, trigonometric and discontinuous functions of $\mathbf{x}$, and a sparse Bayesian solution to $\boldsymbol{\beta}$ is shown, indicating which terms of the dictionary are active (non-zero).

The key ingredient in the formulation of the RVM is the prior distribution of the parameter vector, $p(\boldsymbol{\beta}|\boldsymbol{\alpha})$, (where $\boldsymbol{\alpha}$ is a *hyperparameter*); it is this prior which *enforces sparsity*. The prior is given as a hierarchical Gaussian distribution, which is a *conjugate prior* to a Gaussian distribution and thus yields tractable analysis [14]. The hierarchical prior is,

$$p(\boldsymbol{\beta}|\boldsymbol{\alpha}) = \prod_{i=1}^{M} \mathcal{N}(\beta_i|0, \alpha_i^{-1}) \tag{12}$$

The *hyperparameter* vector $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_M\}$ defines the precision in the prior distribution of the parameters. This hyperparameter vector is what will yield the information about which model terms are significant. However, the values of $\boldsymbol{\alpha}$ are not known *a priori* and must be estimated. In a Bayesian approach the elements in $\boldsymbol{\alpha}$ will need their own *hyperpriors* with hyper-hyperparameters. Furthermore a prior distribution is needed for the variance of ϵ. Assuming a zero-mean Gaussian distribution for the errors, it transpires that it is more convenient to work with the *precision* ρ, which is the reciprocal of the variance σ^2. The overall prescription adopted here is that both $\boldsymbol{\alpha}$ and ρ are Gamma distributed,

$$p(\boldsymbol{\alpha}) = \prod_{i=1}^{M} \Gamma(a)^{-1} b^a \alpha^{a-1} e^{-b\alpha} \tag{13}$$

$$p(\rho) = \Gamma(c)^{-1} d^c \alpha^{c-1} e^{-d\rho} \tag{14}$$

where Γ is the Gamma function and a, b and c, d are the necessary hyper-hyperparameters of the prior and noise precisions respectively. These hyper-hyperparameters control the sparsity of the model; in practice they need to be set such that $p(\boldsymbol{\beta}|\boldsymbol{\alpha})$ becomes peaked around zero, to within numerical precision. For a more detailed description of the role of these hierarchical hyper-priors see [23].

Assuming a Gaussian likelihood function, the posterior distribution over the parameters can be written using Bayes' rule as,

$$p(\boldsymbol{\beta}|\mathbf{t}, \boldsymbol{\alpha}, \sigma^2) = \frac{p(\mathbf{t}|\boldsymbol{\beta}, \sigma^2) p(\boldsymbol{\beta}|\boldsymbol{\alpha})}{p(\mathbf{t}|\boldsymbol{\alpha}, \sigma^2)} \tag{15}$$

Because of the conjugacy, one can use standard Gaussian identities [14], and this yields a posterior Gaussian,

$$p(\boldsymbol{\beta}|\mathbf{t}, \boldsymbol{\alpha}, \sigma^2) = \mathcal{N}(\boldsymbol{\mu}, \Sigma) \tag{16}$$

where the posterior mean and variance are given by,

$$\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\mathbf{D}^{\top}\mathbf{t} \tag{17}$$

$$\boldsymbol{\Sigma} = \mathbf{A} + \boldsymbol{\sigma}^{-2}\mathbf{D}^{\top}\mathbf{D}^{-1} \tag{18}$$

where $\mathbf{A}$ is a diagonal matrix with the elements of $\boldsymbol{\alpha}$ along its diagonal. At the risk of repetition, Eqs. (17) and (18) define the mean and covariance of the *coefficient vector* $\boldsymbol{\beta}$. Both of these equations represent the classic action of a Bayesian algorithm in that prior estimates of the quantities are updated based on observed data, in this case represented by $\mathbf{D}$ and $\mathbf{t}$.

In order to make predictions with this model, one would wish to evaluate the distribution $p(\mathbf{t}_{\star}|\mathbf{t}, \boldsymbol{\alpha}, \sigma^2)$(where $\mathbf{t}_{\star}$ is a set of previously unseen testing data points); this can be shown to be a multivariate Gaussian with mean and covariance [23],

$$\mathbf{y}_{\star} = \mathbf{D}\boldsymbol{\mu} \tag{19}$$

$$\mathbf{V}_{\star} = \sigma^2\mathbf{I} + \mathbf{D}^{\top}\boldsymbol{\Sigma}\mathbf{D} \tag{20}$$

The predictive variance in Eq. (20) is the sum of two terms: the signal noise, σ^2 and the predictive uncertainty arising from the term $\mathbf{D}^{\top}\boldsymbol{\Sigma}\mathbf{D}$.

For sparse Bayesian learning to be effectively realised, one has to optimise the hyperparameter vector $\boldsymbol{\alpha}$ that encodes the sparsity level and determine the parameter σ^2 that estimates the signal noise; this can be achieved using a type-II maximum-likelihood procedure based on the Expectation Maximisation (EM) [28] algorithm. In the original RVM paper [23], the EM steps are clearly described and these lead to efficient pruning of 'irrelevant' vectors. However, in [29] a more efficient version[2] of the EM algorithm is described, and this is the version used in the current work for the hyperparameter optimisation. More details on the hyperparameter optimisation procedure are provided in Appendix A.

As it is with any EM optimisation procedure, the RVM may become stuck in local optima and yield suboptimal solutions. From an equation discovery point of view, a locally optimum solution could correspond to a variable selection scenario where the RVM algorithm incorrectly includes a correlated variable in the model instead of the true variable. Such scenarios can happen when the measurements are highly noisy and/or the dictionary matrix is severely ill-conditioned.

In the case studies following, no issues with local minima were encountered. Multiple runs of the RVM were conducted with different initial values of the measurement noise variance, the results were very similar each time, indicating that a global optimum had been found.

3. Numerical case studies

In order to investigate the proposed approach to equation discovery (i.e. combined model selection and parameter estimation), several numerical experiments are carried out. The class of systems being investigated here is the single degree-of-freedom oscillator of Eq. (5) with general nonlinearity $g(\dot{y}, y)$. As before, the system is converted into a state-space representation with variables $x_1 = y$ (displacement) and $x_2 = \dot{y}$, yielding the regression problem expressed via Eq. (9).

Different forms of the nonlinearity $g(x_1, x_2)$ yield various systems of engineering interest, and five different representative cases are investigated here, as summarised in Table 1. The linear case, although simple, is included in order to establish that the proposed algorithm is capable of ruling out any nonlinearities when necessary. In all cases, the parameters used for the underlying linear model were $m = 1, k = 1 \times 10^4$ and $c = 20$, which places the natural frequency of the underlying linear oscillator at 15.9 Hz.

The second system considered includes a quadratic damping term. This is representative of systems that operate in fluids, where the drag force is non-negligible and contributes significantly to the damping. The third system is a classic Duffing oscillator, with a cubic stiffness nonlinearity $g(x_1, x_2) = k_3 x_1^3$. The Duffing oscillator often appears in studies in SI; this is because of the complex behaviours it can generate, but also because it is representative of geometric nonlinearities found in real systems, such as beams undergoing large displacements. The fourth system of interest here possesses a Coulomb friction nonlinearity; this model assumes that the frictional force is constant (proportional to the normal load), but depends on the direction of the velocity. This nonlinearity is compactly represented by $g(x_1, x_2) = k_c \mathrm{sgn}(x_2)$ (where sgn denotes the *signum* function). This type of system is of general interest in nonlinear SI [7,30,31], as it represents a wide range of practical structures where dry sliding occurs; for example, systems with bolted joints.

The fifth and final system investigated here is the Bouc-Wen model [32], which represents a hysteretic nonlinear restoring force through a third state variable, $g(x_1, x_2) = x_3(x_1, x_2)$. The dynamics of the restoring force are then described by the following nonlinear first-order differential equation,

$$\dot{x}_3 = \begin{cases} -a|x_2|x_3^n - bx_2|x_3^n| + Ax_2 & \text{for } n \text{ odd} \\ -a|x_2|x_3^{n-1}|x_3| - bx_2|x_3^n| + Ax_2 & \text{for } n \text{ even} \end{cases} \tag{21}$$

where the parameters A, a, b, n control the smoothness of the process, and allow a versatile prescription for the hysteresis loop [32]. Note that the restoring force x_3, is not only nonlinear, but also discontinuous, because of the modulus terms. The Bouc-Wen system has proven to be a useful model in the identification and control of a large number of processes with nonlinear restoring forces [33]; thus, identifying the parameters of this model is of fundamental interest in structural

[2] The accompanying MATLAB software can be downloaded from http://www.miketipping.com/downloads.htm.

Table 1
Summary of nonlinearities in simulated systems considered.

System	Name	$g(x_1, x_2)$	
1	Linear	0	
2	Quadratic Damping	$k_2 x_2 \lvert x_2 \rvert$	$k_2 = 2$
3	Duffing	$k_3 x_1^3$	$k_3 = 10^9$
4	Coulomb Friction	$k_c \mathrm{sgn}(x_1)$	$k_c = 1$
5	Bouc-Wen	x_3 (Eq. (21))	$A = 6800, n = 3, a = 1.5, b = -1.5$

dynamics. Previous studies have focussed on both parametric [9,34–37], and nonparametric identification [38]. From the point of view of equation discovery, the Bouc-Wen model presents a stronger challenge, as the nonlinear restoring force x_3 is usually an *unobserved* variable and must be estimated. In this paper, this issue is sidestepped in order to focus on the problem at hand, of identifying the correct terms in the nonlinear differential equation. It is noted however, that this restoring force can sometimes be measured in laboratory environments (see for example [39]). While the focus here is not placed on the identification of the latent forcing term, it should be noted that approaches based on Approximate Bayesian Computation (ABC) [37] and evolutionary approaches [40] have been shown to cope well with the joint model-parameter selection problem in this setting, albeit with a reduced number of candidate terms.

Each of the systems of interest here was simulated using a fourth-order Runge–Kutta numerical integration scheme, with a sample rate of 32768 Hz. This sample rate is significantly higher than the natural frequency of the linear system; this is to properly accommodate harmonics in the data that might otherwise cause aliasing and also to minimise the error in any numerical differentiation and minimise any artefacts of the numerical integration, which might confuse the sparse Bayesian learner. The state vector $\mathbf{x}$, collected from the simulation provides displacement and velocity; the derivatives, $\dot{\mathbf{x}}$, are obtained by numerical differentiation with respect to time. Although the derivatives are also available from the Runge–Kutta scheme, numerical differentiation was used to introduce an element of reality into the exercise here, reflecting the fact that one does not regularly measure all state variables, and recognising that the operation is likely to introduce some high-frequency noise. The simulation variables from the Runge–Kutta scheme were corrupted with white Gaussian noise with a variance of 0.4%, relative to the standard deviation of the observations. Note that this noise propagates through the numerical differentiation, again leading to a higher variance in $\dot{\mathbf{x}}$. The addition of noise is realistic here in terms of the nonlinear nature of the problem. Adding white Gaussian noise to the state variables – which might be safely assumed for the measurement instrumentation – would not be expected to bias any coefficients of linear model terms [41]. However, numerical differentiation introduces coloured noise, weighted towards the high frequencies, and Gaussian noise on state variables will manifest as coloured and correlated noise on nonlinear dictionary terms e.g. $(x_1 + \epsilon)^3 = x_1^3 + 3\epsilon x_1^2 + 3\epsilon^2 x_1 + \epsilon^3$. Furthermore, while ϵ and ϵ^3 would be zero-mean, ϵ^2 would not, and would thus generate a non-zero expectation for an x_1^2 term that could potentially confuse equation discovery. The case studies will show that this latter effect does not appear to be an issue when the state noise level is low.

The equation discovery problem was addressed using the sparse learner approach presented in Section 2. The success of the algorithm depends critically on the dictionary $\mathbf{D}$. In this paper, a dictionary was assembled using candidate functions that include multinomial expansion terms as well as trigonometric functions,

$$\mathbf{D}(\mathbf{x}) = \left\{ u(t), P^1(\mathbf{x}), \ldots, P^n(\mathbf{x}), \sin(\mathbf{x}), \cos(\mathbf{x}), \tan(\mathbf{x}), \mathrm{sgn}(\mathbf{x}) \right\} \tag{22}$$

where $P^n(\mathbf{x})$ is a convenient shorthand here for the terms present in the expansion $(x_1 + x_2)^n$; polynomial orders up to $n = 6$ were used here. For the Bouc-Wen case, the dictionary was augmented with the state x_3 and multinomials containing the terms $|x_1|, |x_2|$ and $|x_3|$. For this case, the force $u(t)$ is also included in the dictionary; to simplify matters, linearity in $u(t)$ is assumed, although this condition could be relaxed in more complicated cases, as in fluid–structure interaction problems for example [42].

The inference over model and system parameters is given in terms of the regression on $\dot{x}_2$, for the first four systems (i.e. $\dot{x}_2$ are the targets), and in terms of the hysteretic restoring force state $\dot{x}_3$ for the Bouc-Wen system.

Appropriate scaling of the observed data $\mathbf{x}$ is important for the success of the procedure. All states and columns of $\mathbf{D}$ have been scaled to unit standard deviation in order to optimise the conditioning of the numerics; this is critical when high powers of the variables are included.

The overall results are presented in Fig. 2 for each of the systems excited with a single sine wave of 10 Hz at an amplitude of 100 N[3] The first column in Fig. 2 shows the phase-space representation in terms of the simulation variables x_1 and x_2. The

[3] A good question here – raised by one of the anonymous reviewers – is whether a sine wave is an effective excitation for system identification. In general, the question of optimal excitation for nonlinear SI is quite nuanced. In linear system identification, the question of optimality is related to whether the excitation is *persistently exciting*. This is a quite subtle technical condition; however, a simple view is provided by Definition 14.2 in [41], which essentially asserts that an input signal is persistently exciting if it has broadband support. Under this definition, a single sine wave is clearly not persistently exciting. However, this is not a problem in the current work, where the objective is simply to demonstrate the effectiveness of the approach to nonlinear identification. The input works here because the nonlinearity produces additional frequency support in the response from the harmonics. Finally, a sine wave can confuse acceleration terms and displacement terms at low excitation (no harmonics); however, this is not a problem in the case studies here as acceleration terms are not included in the dictionary. Persistent excitation is discussed in [41] in the context of *informative experiments*; the results shown here indicate that the synthetic experiments here are sufficiently information that effective identification is possible.

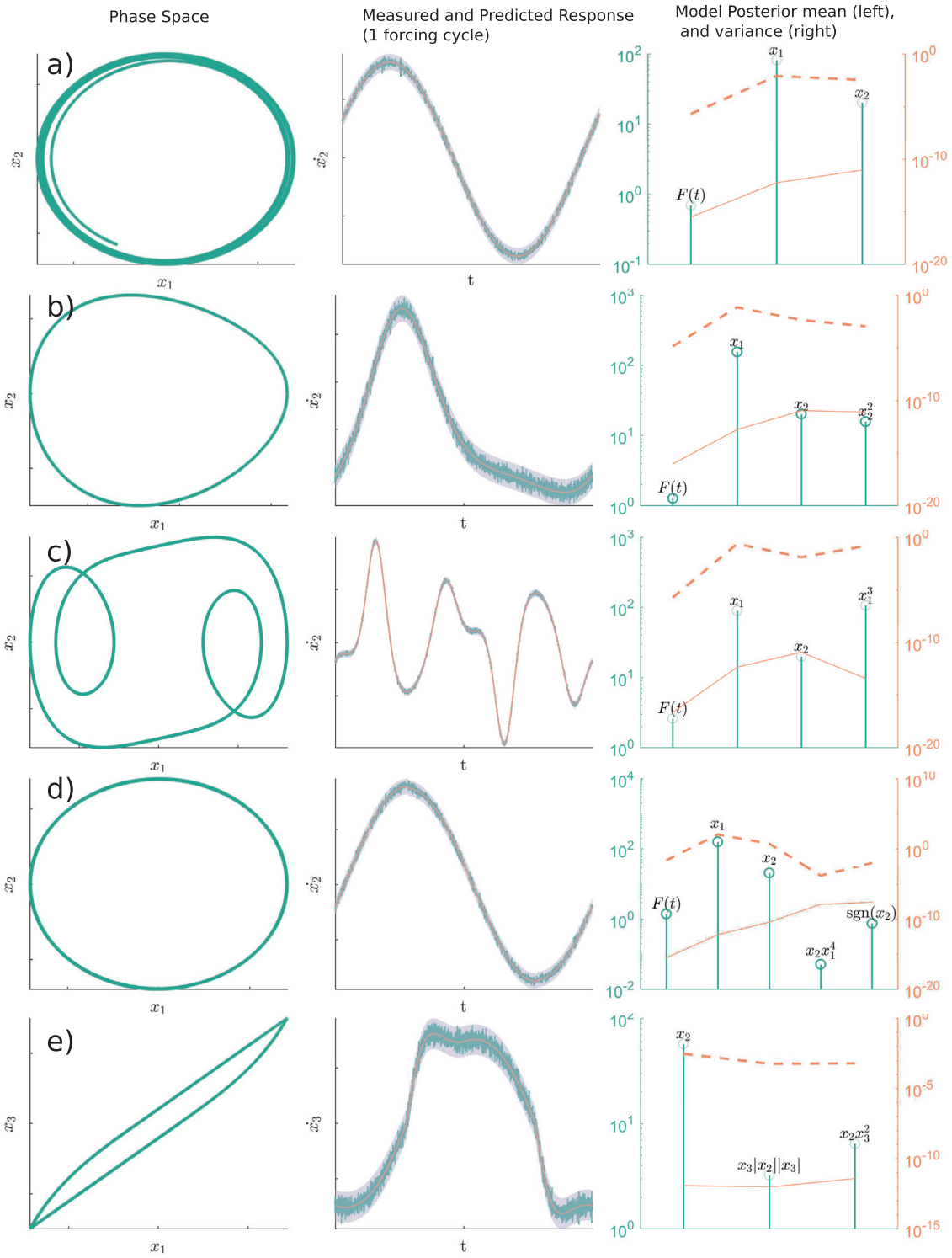

Fig. 2. Sparse Bayesian solution for simulated systems One to Five. The left column shows the system phase portrait; the middle column shows the target variable together with the predictive response, the shaded area illustrates the predictive uncertainty through the 3σ confidence interval; the right column shows the (log absolute) weights (left axis) of the identified non-zero terms of the sparse Bayesian learner, together with prior (solid line) and posterior (dashed line) variances on the right hand axis. Numbers on the axes of plots including standardised variables are not shown.

second column shows the target and predicted responses in the time domain (for one excitation cycle), where the shaded area illustrates the predictive uncertainty via the 3σ confidence interval. The third column shows the absolute values of the coefficient vector $\boldsymbol{\beta}$ resulting from sparse Bayesian inference (on a logarithmic scale). For clarity, only coefficients that yielded non-zero values are shown. The posterior variances and optimised prior variance hyperparameters are shown alongside the coefficient, on the right axis. The posterior variance quantifies the posterior uncertainty around each coefficient vector. The optimised hyper-prior variance, α_i for each term is also called a 'sparsity factor' as this quantifies the degree to which any given column in the dictionary contributes to the solution. If α is low, this means that the solution is concentrated tightly around the parameter, thus deeming it 'relevant'. If α is high, this implies that the vector does not contribute to a sparse solution.

Terms that form part of a likely solution are those that have a low variance in both the posterior and the optimised hyper-prior. These two variances, which result from the Bayesian treatment of the sparse solution, provide one with tools to assess how likely it is that suspected 'spurious' terms are truly part of the dynamics, or have crept in from elsewhere (such as analogue or digital filtering).

The results for the basic linear oscillator (System One) are shown in Fig. 2a. The only non-zero coefficients arising from the sparse Bayesian regression on $\dot{x}_2$ are the driving force $u(t)$ and the linear terms in displacement and velocity, x_1 and x_2. The predicted response time series is not only accurate in terms of its mean, but the predicted uncertainty correctly captures the additive measurement noise.

Fig. 2b shows the solution for System Two, which contains the quadratic damping term. The system is identified correctly, with the addition of a quadratic term in the velocity to the linear terms correctly selected. The change of shape that this nonlinearity introduces in the phase-space is very slight, but is evident in both the time-history and the phase portrait in the asymmetry between the upper and lower parts of the cycle, since the square term has no dependency on the direction of velocity.

The results for the Duffing oscillator are shown in Fig. 2c. The effect of the nonlinearity is much more evident in the time-history and in the phase-space. The cubic term in x_1 is identified correctly and the predictive distribution captures the process well.

Fig. 2d shows the results for System Four, which includes a Coulomb damping (friction) term. This term introduces a small discontinuity at the peaks and troughs of the target $\dot{x}_2$, which are only very subtly visible in the time series response. The solution correctly identifies the presence of the term $\mathrm{sgn}(x_2)$, but also selects one spurious polynomial term; this is likely to be an artefact of the numerical integration scheme used to generate the simulation, which does not allow for the fact that a trajectory might cross the discontinuity in mid-timestep. It was verified that at higher sample rates, this effect is mitigated, and at lower sample rates more spurious terms tended to appear.

Fig. 2e shows the results for the Bouc-Wen hysteresis model – System Five here. As mentioned previously, combinations from the expansion $(x_1 + x_2 + x_3 + |x_1| + |x_2| + |x_3|)^n$ were used to build the candidate vectors in the dictionary for this system. In this case, it was assumed that the restoring force state was 'measured' and therefore available for inclusion in the dictionary. The results for equation discovery of $\dot{x}_3$ are shown in Fig. 2e for (the true) $n = 2$. The sparse Bayesian learner correctly identifies the terms for this model form, and the predictive performance reflects this.

4. Experimental studies

Two experimental case studies were investigated. In both cases, the data were supplied as part of the *Nonlinear System Identification Benchmarks* workshops which have been held at VUB Brussels and Eindhoven University over the last few years.[4]

4.1. The silverbox benchmark

The 'Silverbox' is an electronic circuit which has been designed to simulate the response of a Duffing oscillator. As such, it was created as a second-order linear time-invariant (LTI) system with a static cubic nonlinearity made dynamic via feedback [43]. The training data for the benchmark comprised a response set generated from a multi-sine input, while an independent test set was generated from a chirp input. The reader is referred to [43] for the details of the circuit and benchmark data.

For training of the sparse Bayesian algorithm, a section of data of duration 8*s* was used, consisting of a random-phase multi-sine excitation containing 1342 odd harmonics of a base frequency f_0, where $f_0 = f_s/8192s^{-1}$ and the sample rate f_s in the experiments was 610.35 Hz.

When the sparse Bayesian algorithm was applied to the Silverbox training data, the results were as summarised in Fig. 3. As in the simulated case studies, the leftmost entry in the figure shows the phase trajectory of the data; the centre entry shows the predicted response, together with the 3σ confidence intervals on the predictions; the rightmost entry shows the (log magnitude) posterior mean parameter estimates together with the two variance measures previously discussed.

The results are good, although it is difficult to see from the rather compressed time data; Fig. 4 shows a zoom on the time data. The predictions follow the mean trend of the target and the 'true' target values are enclosed by the confidence intervals as desired. In terms of equation discovery, the dominant term is the forcing term (denoted by *Fe* in the figure), with the linear

[4] www.nonlinearbenchmark.org.

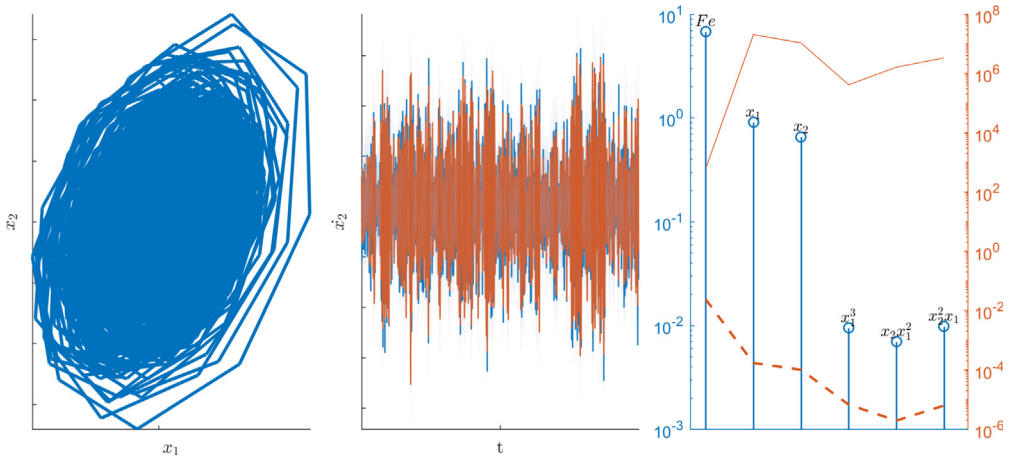

Fig. 3. Results from sparse Bayesian equation discovery on the Silverbox nonlinear benchmark: training set with multi-sine excitation. Measured response is in blue, predicted response is in red; 3σ confidence interval for prediction is represented by the grey region. Numbers on the axes of plots including standardised variables are not shown. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

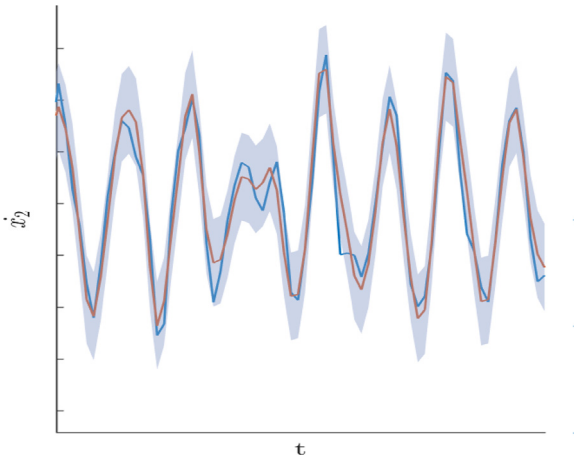

Fig. 4. Zoomed results from central plot in Fig. 3 (training data). Measured response is in blue, predicted response is in red; 3σ confidence interval for prediction is represented by .the grey region.

terms in x_1 and x_2 coming next. The nonlinear terms kept in the model include the correct x_1^3, but also two cross-terms $x_1^2x_2$ and $x_1x_2^2$; the spurious terms are likely included as a result of ancillary circuitry in the Silverbox, the comparatively low sampling rate for the data here may also be a contributing factor.

Although the main question of interest in this work is in discovering the 'correct' equation terms, it is interesting to understand how the complete identification with parameter estimation performs. Indeed, the real challenge in any machine learning problem is to generalise to an independent test set. In this case, the results for the identified model on the chirp test set are shown in Fig. 5, covering a period of 1.63*s*. The chirp is a 'sweep-down' signal; the model fidelity proves to be highest around resonance with some discrepancies at the higher and lower frequencies. However, it is gratifying to note that the 'true' values are always captured by the model confidence intervals. As the RVM has selected more-or-less what is believed to be the correct governing terms in the regression equation, it is likely that the mismatch here arises due to biased parameter estimates.

4.2. Electro-mechanical positioning system

The second experimental case study from the Nonlinear Benchmark workshop was an electro-mechanical positioning system [44] as illustrated in Fig. 6. The system represents a standard configuration for a prismatic joint found in robots and machine tools. The main source of nonlinearity in the system was expected to be friction; the expected equation of motion, as specified in [44] is,

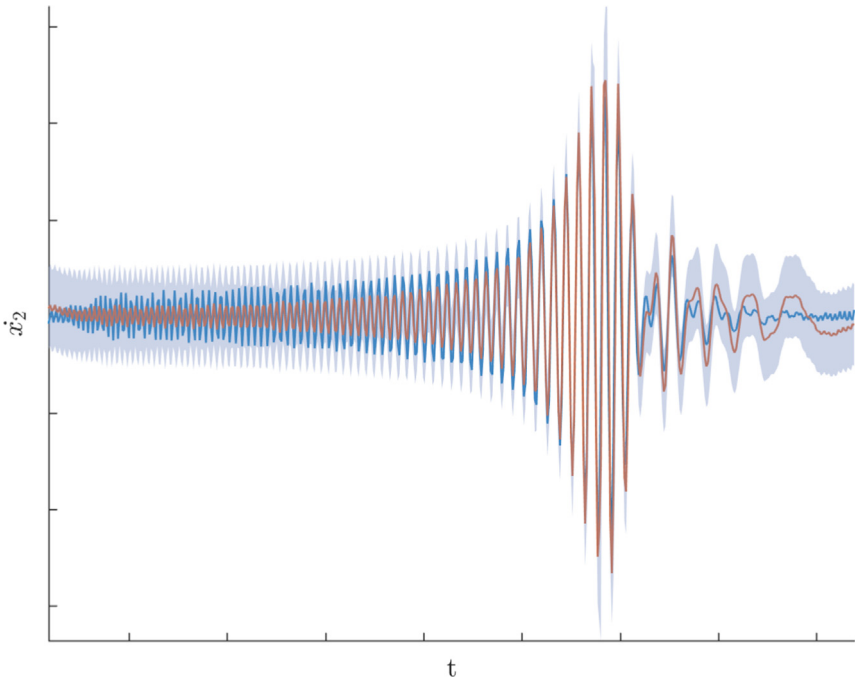

Fig. 5. Results from sparse Bayesian equation discovery on the Silverbox nonlinear benchmark: testing set with 1.6*s* chirp excitation. Measured response is in blue, predicted response is in red; 3σ confidence interval for prediction is represented by the grey region. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

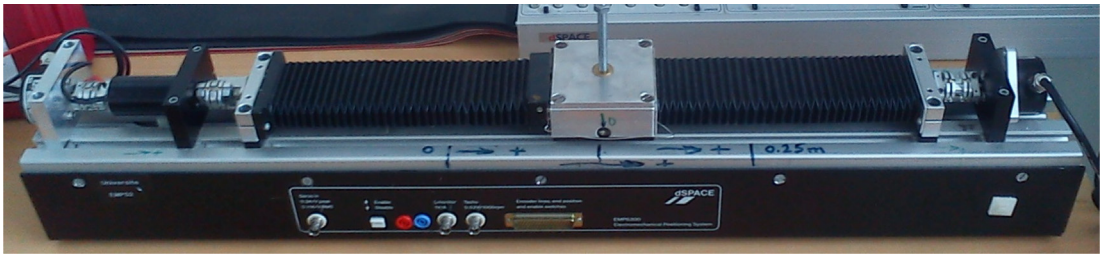

Fig. 6. Electro-mechanical positioning system benchmark [43].

$$m\dot{x_2} + cx_2 + k_c\text{sgn}(x_2) + b = F(t) \tag{23}$$

so that the expected restoring force includes viscous and Coulomb damping, and an offset b.

The sampling frequency was 8192 Hz and the duration of the tests was 25s (approximately, as this was measured by an incremental encoder with resolution of 12500 counts per revolution). Outputs were measured by a dSPACE card. To compute the velocity involved in the feedback control for the system, the motor position was filtered with an FIR filter. For the training data, the system was excited using 'bang-bang' accelerations; these excitations were augmented in the validation set by pulses – for details see [44].

The sparse Bayesian algorithm was applied to the training data, which comprised of a subset of 3.01*s* of the available data, with the results shown in Fig. 7. Fig. 7a shows the usual representation of the parameter estimates together with their variance measures. The response is dominated by the viscous and Coulomb terms as desired; however, there are other (smaller) nonlinear terms in the velocity x_2. The extra terms are not surprising; in the first case, Coulomb friction is never expected to be a perfect description of real friction; in the second case, the signals have been subjected to various signal processing operations, like the aforementioned digital filter. Fig. 7b shows the model predictions together with the $\pm 3\sigma$ confidence intervals; the predictions are excellent, with the confidence intervals clearly capturing the discrepancies due to measurement noise.

An alternative approach to regressing on $\dot{x}_2$ is to regress directly on the force $F(t)$; the results of this exercise are presented in Fig. 8. In this case, the most significant term is the inertial term $\dot{x_2}$; next in significance is $\dot{x}_1 = x_2$, which is, of course, the viscous damping term. As before, the Coulomb term is considered significant and there are other small nonlinear terms in x_2, which are assumed to represent corrections to the Coulomb term for real friction as before. As in the Silverbox case, the sam-

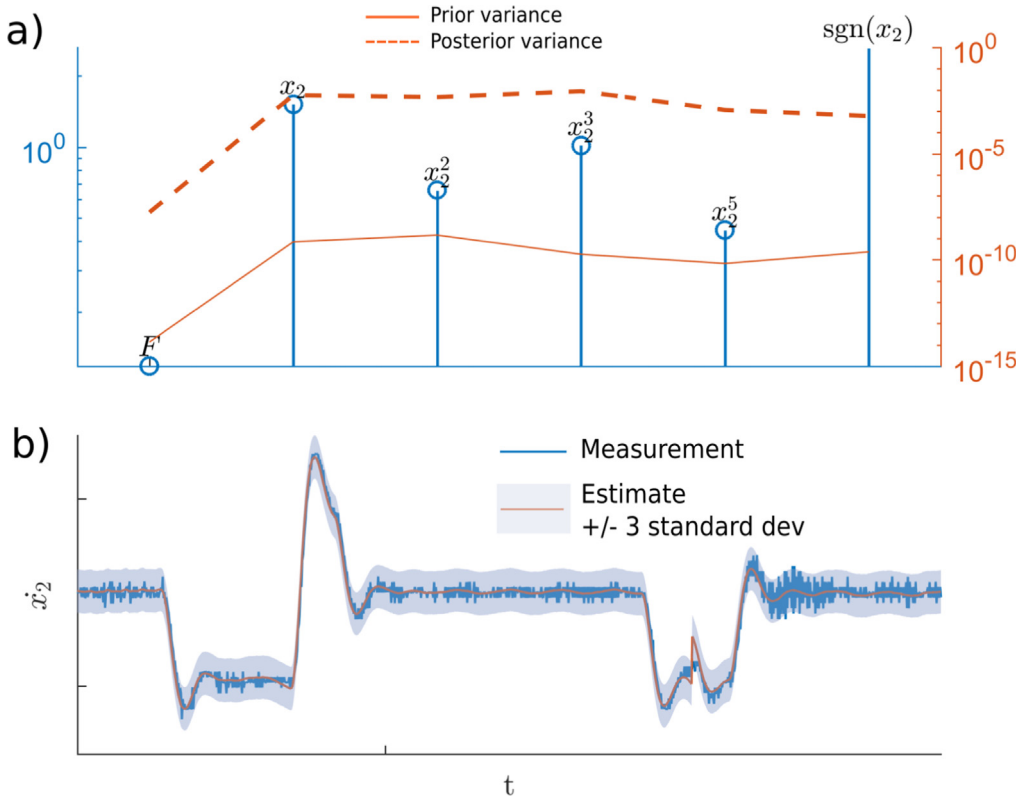

Fig. 7. Results from equation discovery on electro-mechanical positioning system with $\dot{x}_2$ target: a) parameter estimates and variance measures; b) comparison between measured and predicted data – training data.

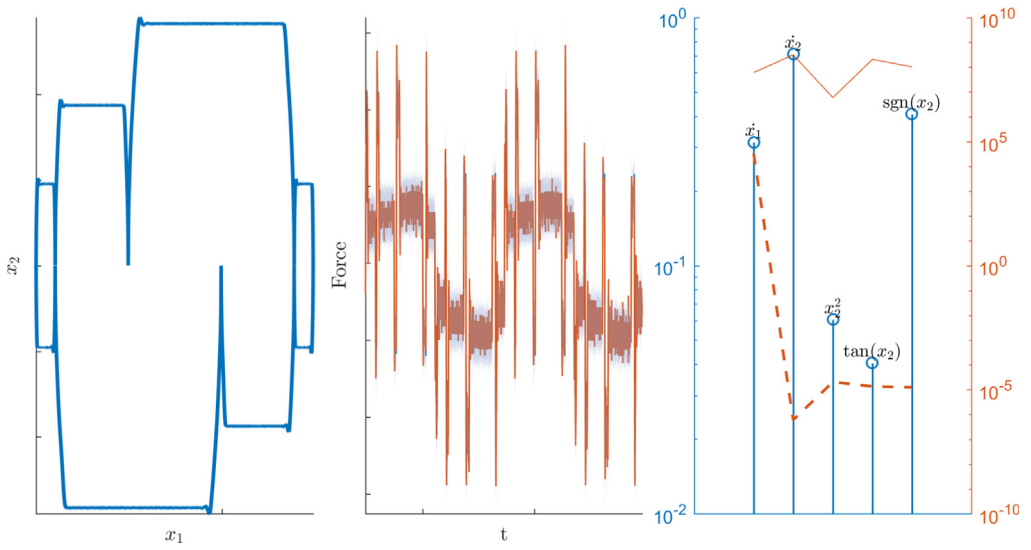

Fig. 8. Results from sparse Bayesian equation discovery on the electro-mechanical positioning system nonlinear benchmark with $F(t)$ target: training set. Measured response is in blue, predicted response is in red; 3σ confidence interval for prediction is represented by the grey region. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

pling is a little too high for one to resolve the detail of the time data comparison, so a zoomed version of the figure is provided in Fig. 9. The zoomed figure shows that the model predictions are excellent, with the confidence intervals appropriately capturing the uncertainty in the output propagated through from noise in the input data.

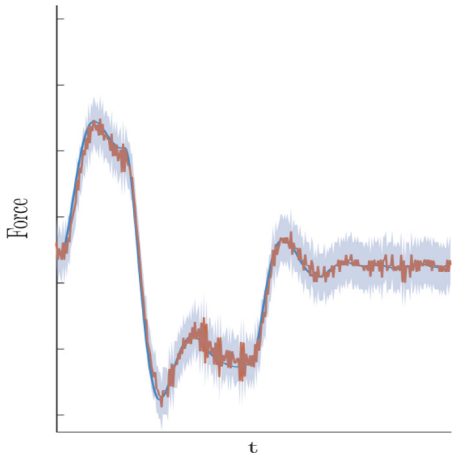

Fig. 9. Zoomed results from Fig. 8. Measured response is in blue, predicted response is in red; 3σ confidence interval for prediction is represented by the grey region. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Something interesting happens when the trained model (for $\dot{x}_2$) is applied to the independent testing set (Fig. 10). As one can see, the confidence intervals are very large. A zoomed plot of the mean prediction is shown in Fig. 11.

The results in Fig. 10 show that the mean predictions track the measured data closely, but shows a small constant offset. The real system is known to have a constant offset in the dynamics, and the suspicion here is that the offset during testing was different to that during training.

Finally, as an objective measure of goodness of fit, the models were evaluated using a normalised mean-square error of the form,

$$NMSE = \frac{100}{N\sigma^2}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{24}$$

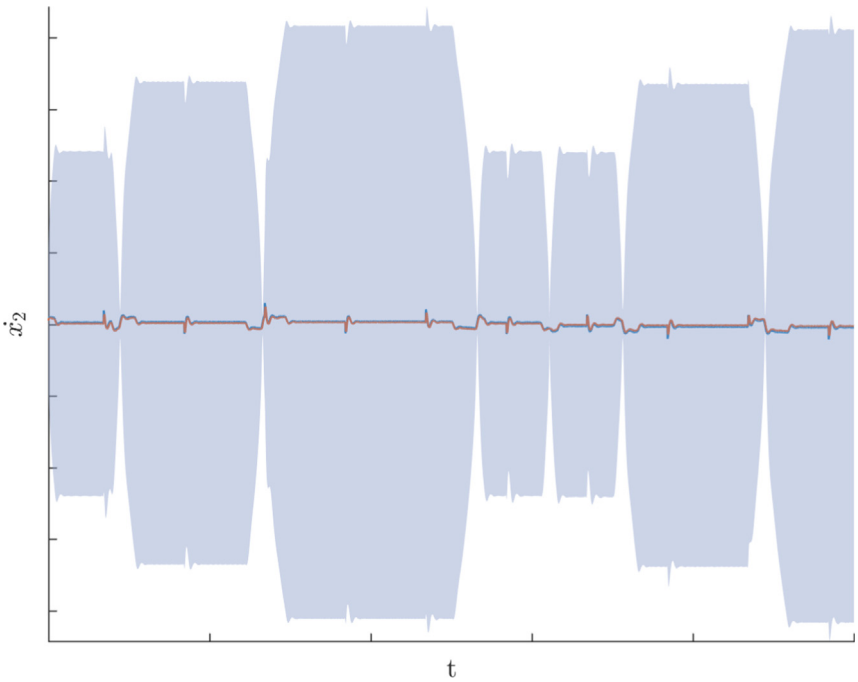

Fig. 10. Application of $\dot{x}_2$ model to independent test set. Measured response is in blue, predicted response is in red; 3σ confidence interval for prediction is represented by the grey region. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

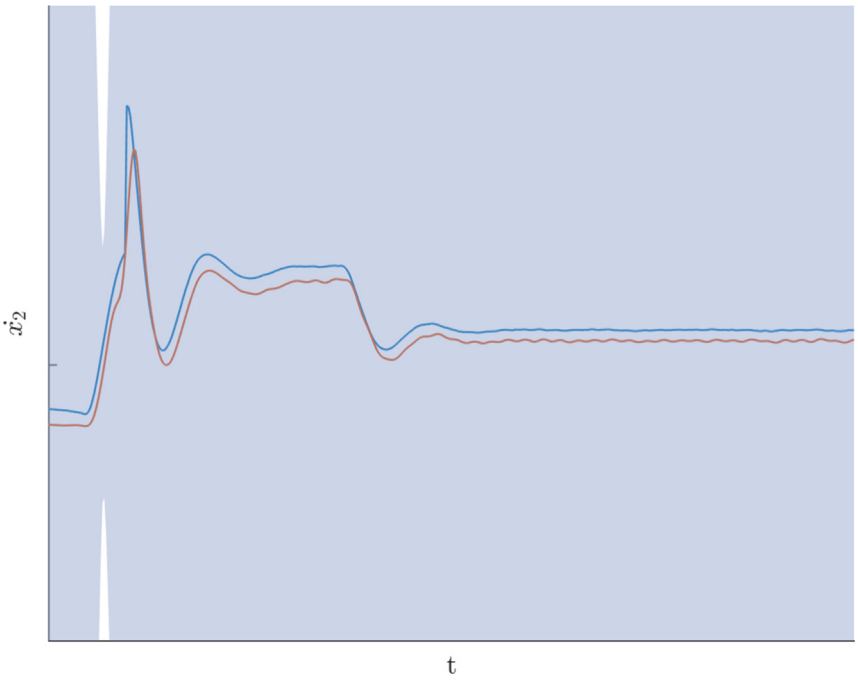

Fig. 11. Application of $\dot{x}_2$ model to independent test set - zoom of Fig. 10. Measured response is in blue, predicted response is in red, the shaded area represents the confidence intervals which exceed the limits of the plot for t.his zoomed view. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where y_i is the sampled variable of interest, and $\hat{y}_i$ is the corresponding model prediction; σ_y^2 is the variance of the measured displacements. This error function has the following useful property; if the mean of the output signal is used as the model i.e. $\hat{y}_i = \overline{y}$ for all i, the error is 100.0 (and can be thought of as a percentage). The errors for the models fitted here are given in Table 2.

5. Discussion

The results of performing equation discovery – combined model selection and parameter identification – with sparse Bayesian learning, as presented here are very encouraging. The algorithm is able to identify individual terms of various dynamical systems which are known to present challenges for SI, and this is a positive result. However, it is important to note that this investigation has been restricted in some respects. In the first case, no simulation results are shown here for different types of forcing, something which will clearly have an influence on the identified system.

While the simulation results that have been presented show the procedure working almost at its best, it is important to highlight that the outcome can depend strongly on the simulation settings, data pre-processing and noise levels. The fact that the algorithm is sensitive to the simulation parameters makes sense given that the simulation algorithm is a dynamical system itself. Also, any amount of digital filtering of the displacements, velocities or accelerations tended to generate a solution with significantly more polynomial terms than those of the pure system of interest. This makes sense, given that most pre-processing tasks could be described as dynamical systems also. This effect may have also made itself evident for the experimental data here, where hardware filters were applied as well as closed-loop control. Another factor that significantly influences the outcome of the identification is the scheme for numerical differentiation required to estimate $\dot{\mathbf{x}}$. Tools are available to perform complex numerical differentiation operations [45]; however, it was found that these would also have an effect on the identified system. In the end, a simple point-difference yielded the most consistent and robust results.

Another interesting discussion point concerns the use of sparsity-inducing priors. The primary drawback of strong sparsity-inducing priors such as the Student-t (RVM) and Laplace (Lasso) is the bias they introduce while attempting to shrink every coefficient towards zero. When learning the dynamics of a system, bias should be avoided if at all possible. There are other types of sparsity-inducing priors that have been shown to induce sparsity while reducing bias, such as the Horseshoe. However, in the authors' view, the current gold standard in Bayesian learning with sparsity is achieved through the use of *spike-and-slab* prior distributions [46]. In this case, the prior is composed of a distribution that is sharply peaked around zero (the spike), and a distribution that defines a broad prior (the slab). This prescription has the effect of attracting only those coefficients that do not have a contribution towards a good fit, to zero. On the other hand, coefficients

Table 2
Relative errors for the training and validation sets.

	Training	Validation
Force	3.83%	20.8%
Acceleration	16.5%	19.53%

that are likely to belong to the model fall in the regime of the slab component, defined by a broad and possibly uninformative prior distribution that minimises parameter bias.

A final remark is that a move to multiple degrees of freedom would be straightforward in principle under this scheme, but this is also outside the scope of this paper and will be addressed in future work.

6. Conclusions

This paper has presented a new approach for equation discovery – combined parameter estimation and model selection – for nonlinear systems using sparse Bayesian learning techniques. The SI problem has been formulated in terms of a first-order (state-space) differential equation that uses a dictionary containing a large number of candidate functional forms that could form part of the solution. The solution to the sparse Bayesian learning problem exploits the use of the Relevance Vector Machine (RVM) here due to its computational tractability and fast implementations.

Sensitivity to hyperparameters has proved to be an issue for many sparse methods. For most non-Bayesian approaches in particular (e.g.. the Lasso), this can be attributed to the use of a single hyperparameter λ to regulate the extent to which the individual model parameters are shrunk towards zero. A consequence of relying on a single hyperparameter is that a large value will indiscriminately shrink all parameter values towards zero, which is undesirable when where there may be some small but relevant parameters. In contrast, the Bayesian approach using the RVM employs predictor-specific hyperparameters, where each predictor is independently controlled by its own hyperparameter. As such, the RVM approach is arguably less sensitive to hyperparameter tuning, which is an added advantage to the method.

The approach has been demonstrated and validated using a series of numerical simulations of nonlinear systems; the results here show that the method correctly identifies the type and presence of several nonlinearities such as: cubic stiffness (Duffing oscillator), quadratic and Coulomb damping, as well as the type and presence of hysteresis in the form of a Bouc-Wen model. Furthermore, the approach has been demonstrated on two experimental benchmarks with considerable success: a nonlinear electrical circuit with known cubic nonlinearity and an electro-mechanical positioning system with Coulomb friction nonlinearity. In both experimental cases the expected terms corresponding to the known dynamics of such systems were recovered, albeit with extra terms owing to the limitations of sampling real-world data as well as analogue and digital pre-processing.

There is clear scope for continuing to explore other strategies for performing SI under a hierarchical Bayesian framework with sparse priors that encourage parsimonious representations of physical systems, as this would allow one to learn representations that are potentially capable of extrapolation, whilst quantifying the uncertainty in such predictions. One clear avenue of further work is that of exploring sparsity-inducing prior forms that minimise the bias induced by the implicit regularisation. One limitation of the work presented here is the authors' own bias towards mechanical systems; one avenue of further research is the application of this framework to other types of natural and engineering systems.

CRediT authorship contribution statement

R. Fuentes: Conceptualization, Methodology, Software, Writing - original draft. **R. Nayek:** Writing - review & editing. **P. Gardner:** Writing - review & editing. **N. Dervilis:** Writing - review & editing. **T. Rogers:** Writing - review & editing. **K. Worden:** Supervision, Writing - original draft, Funding acquisition. **E.J. Cross:** Supervision, Writing - review & editing, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has been supported by the UK Engineering and Physical Sciences Research Council (EPSRC), through the project: *Autonomous Inspection in Manufacturing and Re-manufacturing* (AIMaReM), Grant No. EP/N018427/1. KW also gratefully acknowledges support from the EPSRC through Grant No. EP/J016942/1 and EJC is grateful to EPSRC for support via Grant No.

EP/S001565/1. Finally, the authors would also like to thank Dr. Kartik Chandrasekhar, of the Dynamics Research Group at Sheffield, for useful discussions that led to a better paper.

Appendix A. Marginal likelihood maximisation procedure for RVM

The RVM requires computing the optimal values of the hyperparameters $\boldsymbol{\alpha}$ and σ^2. They are determined by maximising the marginal likelihood function using Type II maximum likelihood. The marginal likelihood function is obtained by integrating out the vector $\boldsymbol{\beta}$ as follows,

$$p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2) = \int p(\mathbf{t}|\boldsymbol{\beta},\sigma^2)\, p(\boldsymbol{\beta}|\boldsymbol{\alpha})\, d\boldsymbol{\beta} \tag{A.1}$$

Since both $p(\mathbf{t}|\boldsymbol{\beta},\sigma^2)$ and $p(\boldsymbol{\beta}|\boldsymbol{\alpha})$ have Gaussian distributions, their convolution result in a Gaussian distribution, and the above integration can be analytically computed to give the log marginal likelihood,

$$\log p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2) = -\frac{1}{2}\left[N\log(2\pi) + \log|\mathbf{C}| + \mathbf{t}^\top\mathbf{C}^{-1}\mathbf{t}\right] \tag{A.2}$$

where,

$$\mathbf{C} = \sigma^2\mathbf{I} + \mathbf{D}\mathbf{A}^{-1}\mathbf{D}^\top \tag{A.3}$$

and $\mathbf{A}$ is the diagonal matrix with the elements of $\boldsymbol{\alpha}$ along its diagonal. Another form of the log marginal likelihood can be defined in terms of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ (defined in Eqs. (17) and (18)) as

$$\log p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2) = -\frac{1}{2}\left[N\log(2\pi\sigma^2) + (\sigma^{-2}\mathbf{t}^\top\mathbf{t} - \boldsymbol{\mu}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}) + \log|\boldsymbol{\Sigma}| - \sum_{i=1}^{M}\log\alpha_i\right] \tag{A.4}$$

The goal is to maximise Eq. (A.2) or equivalently (A.4) with respect to the hyperparameters $\boldsymbol{\alpha}$ and σ^2. One approach is set the derivatives of the log marginal likelihood (Eq. A.4) with respect to the hyperparameters to zero, which gives,

$$\begin{aligned} \tfrac{d}{d\alpha_i}\log p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2) = \;& \tfrac{1}{2\alpha_i} - \tfrac{1}{2}\Sigma_{ii} - \tfrac{1}{2}\mu_i^2 = 0 \\ \Rightarrow \;& \alpha_i^{new} = \tfrac{1-\alpha_i\Sigma_{ii}}{m_i^2} \end{aligned} \tag{A.5}$$

and,

$$\begin{aligned} \tfrac{d}{d\sigma^{-2}}\log p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2) = \;& \tfrac{1}{2}\left[\tfrac{N}{\sigma^{-2}} - \|\mathbf{t}-\mathbf{D}\boldsymbol{\mu}\|^2 - \mathrm{tr}(\boldsymbol{\Sigma}\mathbf{D}^\top\mathbf{D})\right] = 0 \\ \Rightarrow \;& (\sigma^{-2})^{new} = \frac{N - M + \sum_{i=1}^{M}\alpha_i\Sigma_{ii}}{\|\mathbf{t}-\mathbf{D}\boldsymbol{\mu}\|^2} \end{aligned} \tag{A.6}$$

Here, Σ_{ii} is the element in the i^{th} diagonal of $\boldsymbol{\Sigma}$. The hyperparameters α_i and σ^2 which maximise the marginal likelihood are then found iteratively by setting $\boldsymbol{\alpha}$ and σ^2 to initial values, finding values of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ from equations (17) and (18), using these to calculate new estimates for $\boldsymbol{\alpha}$ and σ^2 and repeating this process until a convergence criteria is met.

It is noted that the result in Eq. (A.5) for re-estimating α_i is an implicit function of α_i. Due to this implicit form, [29,47] proposed a second approach to solving the optimisation problem for the RVM, in which the dependence of the marginal likelihood on a particular α_i is made explicit and then the stationary points are determined. To do this, the contribution of α_i in the matrix $\mathbf{C}$ (defined in equation A.3) is explicitly written out as follows,

$$\begin{aligned} \mathbf{C} \;& = \sigma^2\mathbf{I} + \sum_{m\neq i}\alpha_m^{-1}\boldsymbol{d}_m\boldsymbol{d}_m^\top + \alpha_i^{-1}\boldsymbol{d}_i\boldsymbol{d}_i^\top \\ & = \mathbf{C}_{-i} + \alpha_i^{-1}\boldsymbol{d}_i\boldsymbol{d}_i^\top \end{aligned} \tag{A.7}$$

where $\mathbf{C}_{-i}$ represents $\mathbf{C}$ with the contribution of the basis vector i removed. The determinant and inverse of $\mathbf{C}$ can be written as,

$$|\mathbf{C}| = |\mathbf{C}_{-i}|\,|1 + \alpha_i^{-1}\boldsymbol{d}_i^\top\mathbf{C}_{-i}^{-1}\boldsymbol{d}_i| \tag{A.8}$$

$$\mathbf{C}^{-1} = \mathbf{C}_{-i}^{-1} - \frac{\mathbf{C}_{-i}^{-1}\boldsymbol{d}_i\boldsymbol{d}_i^\top\mathbf{C}_{-i}^{-1}}{\alpha_i + \boldsymbol{d}_i^\top\mathbf{C}_{-i}^{-1}\boldsymbol{d}_i} \tag{A.9}$$

using established matrix determinant and inverse identities. Using the above results, the log marginal likelihood function in Eq. (A.2) can be re-written as,

$$\log p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2) = -\frac{1}{2}\Big[N\log(2\pi) + \log|\mathbf{C}_{-i}| + \mathbf{t}^\top\mathbf{C}_{-i}^{-1}\mathbf{t} \tag{A.10}$$

$$- \log\alpha_i + \log\left(\alpha_i + \boldsymbol{d}_i^\top\mathbf{C}_{-i}^{-1}\boldsymbol{d}_i\right) - \frac{\left(\boldsymbol{d}_i^\top\mathbf{C}_{-i}^{-1}\mathbf{t}\right)^2}{\alpha_i + \boldsymbol{d}_i^\top\mathbf{C}_{-i}^{-1}\boldsymbol{d}_i}\Big] \tag{A.11}$$

$$= \log p(\mathbf{t}|\boldsymbol{\alpha}_{-i},\sigma^2) + \underbrace{\frac{1}{2}\left[\log\alpha_i - \log(\alpha_i + s_i) + \frac{q_i^2}{\alpha_i + s_i}\right]}_{\ell(\alpha_i)} \tag{A.12}$$

where $\log p(\mathbf{t}|\boldsymbol{\alpha}_{-i},\sigma^2)$ is simply the log marginal likelihood with the basis function $\boldsymbol{d}_i$ removed and the quantity $\ell(\alpha_i)$ contains all the dependence on α_i. Moreover, the two quantities s_i and q_i are,

$$s_i = \boldsymbol{d}_i^\top\mathbf{C}_{-i}^{-1}\boldsymbol{d}_i \tag{A.13}$$

$$q_i = \boldsymbol{d}_i^\top\mathbf{C}_{-i}^{-1}\mathbf{t} \tag{A.14}$$

The stationary points of the marginal likelihood with respect to α_i occur when the derivative,

$$\frac{d\ell(\alpha_i)}{d\alpha_i} = \frac{\alpha_i^{-1}s_i^2 - (q_i^2 - s_i)}{2(\alpha_i + s_i)^2} \tag{A.15}$$

is set to zero. Since $\alpha_i \geqslant 0$, there are two possible solution cases for α_i,

$$\alpha_i = \frac{s_i^2}{q_i^2 - s_i} \quad \text{if } q_i^2 > s_i \tag{A.16}$$

$$\alpha_i = \infty \quad \text{if} q_i^2 \leqslant s_i \tag{A.17}$$

The relative values of s_i and q_i determine whether a particular basis vector will be included in the model or not. Note that this approach yields a closed-form explicit solution for α_i, given values of the other hyperparameters. The resulting sequential sparse Bayesian learning algorithm used in RVM is summarised below [29]:

1. Initialisation step
   (a) Choose starting value of σ^2.
   (b) Initialise using a single basis vector $\boldsymbol{d}_i$. Choose $\boldsymbol{d}_i$ as the basis vector from $\mathbf{D}$ with the largest normalised projection onto the target vector $\mathbf{t}$, $\|\boldsymbol{d}_i^T\mathbf{t}\|^2/\|\boldsymbol{d}_i\|^2$. The corresponding α_i is set using equation (A.16), as follows:

   $$\alpha_i = \frac{\|\boldsymbol{d}_i\|^2}{\|\boldsymbol{d}_i^T\mathbf{t}\|^2/\|\boldsymbol{d}_i\|^2 - \sigma^2}$$

   All other α_m are notionally set to infinity, so that only one basis vector is included in the model.
2. Compute $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$ (which are scalars initially), along with q_i and s_i for all M basis vectors.
3. Select a candidate basis vector $\boldsymbol{d}_i$ from $\mathbf{D}$.
4. If $q_i^2 > s_i$ and $\alpha_i < \infty$ (that is, $\boldsymbol{d}_i$ is in the model), update α_i using Eq. (A.16).
5. If $q_i^2 > s_i$ and $\alpha_i = \infty$, then add $\boldsymbol{d}_i$ to the model and evaluate α_i using Eq. (A.16).
6. If $q_i^2 \leqslant s_i$ and $\alpha_i < \infty$, then remove $\boldsymbol{d}_i$ to the model and set $\alpha_i = \infty$.
7. Update $\sigma^2 = \frac{\|\mathbf{t}-\mathbf{D}\boldsymbol{\mu}\|^2}{N - M + \sum_{i=1}^{M}\alpha_i\Sigma_{ii}}$ (using equation (A.6)).
8. If the estimates have converged, then terminate, else go to 3.

References

[1] L. Ljung, System identification, in: Wiley Encyclopedia of Electrical and Electronics Engineering, John Wiley and Sons, 2017, pp. 1–19.
[2] M. Raissi, G.E. Karniadakis, Hidden physics models: Machine learning of nonlinear partial differential equations, J. Comput. Phys. (2018).
[3] G.A. Bekey, J.E.W. Beneken, Identification of biological systems: a survey, Automatica 14 (1978) 41–47.
[4] H. Kitano, Systems biology: toward system-level understanding of biological systems, Found. Systems Biol. (2001) 1–36.
[5] K. Hiroaki, Computational systems biology, 2002.
[6] P. Kirk, T. Thorne, Michael M.P.H. S~tumpf, Model selection in systems and synthetic biology, Curr. Opin. Biotechnol. 24 (2013) 767–774.
[7] G. Kerschen, K. Worden, A.F. Vakakis, J.-C. Golinval, Past, present and future of nonlinear system identification in structural dynamics, Mech. Syst. Signal Process. 20 (2006) 505–592.
[8] J. Ching, J.L. Beck, K.A. Porter, Bayesian state and parameter estimation of uncertain dynamical systems, Probab. Eng. Mech. 21 (2006) 81–96.
[9] K. Worden, J.J. Hensman, Parameter estimation and model selection for a class of hysteretic systems using Bayesian inference, Mech. Syst. Signal Process. 32 (2012) 153–169.

[10] T.J. Cowan, A.S. Arena, K.K. Gupta, Accelerating computational fluid dynamics based aeroelastic predictions using system identification, J. Aircraft 38 (2001) 81–87.
[11] J.L. Beck, L.S. Katafygiotis, Updating models and their uncertainties. I: Bayesian statistical framework, ASCE J. Eng. Mech. 124 (1998) 455–461.
[12] M.C. Kennedy, A. O'Hagan, Bayesian calibration of computer models, J. Roy. Soc.– Series B 63 (2001) 425–464.
[13] S.A. Billings, Nonlinear System Identification: NARMAX, Methods in the Time, Frequency, and Spatio-Temporal Domains, Wiley-Blackwell, 2013.
[14] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2013.
[15] J.L. Beck, Bayesian system identification based on probability logic, Struct. Control Health Monitor. 17 (2010) 825–847.
[16] D.J.C. MacKay, Information Theory, Inference and Learning Algorithms, Cambridge University Press, 2005.
[17] A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, D.B. Rubin, Bayesian Data Analysis, third ed., Chapman and Hall, 2014.
[18] R. Tibshirani, Regression selection and shrinkage via the Lasso, J. Roy. Stat. Soc.– Series B 58 (1996) 267–288.
[19] S.L. Brunton, J.L. Proctor, J.N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Nat. Acad. Sci. USA 113 (2016) 3932–3937.
[20] T. Hastie, R. Tibshirani, M. Wainwright, Statistical Learning with Sparsity: The Lasso and Generalizations, Chapman and Hall, 2015.
[21] T. Park, G. Casella, The Bayesian Lasso, J. Am. Stat. Assoc. 103 (2008) 681–686.
[22] M. Schmidt, H. Lipson, Distilling free-form natural laws from experimental data, Science 324 (2009) 81–85.
[23] M. Tipping, Sparse Bayesian learning and the relevance vector machine, J. Machine Learn. Res. 1 (2001) 211–244.
[24] D.P. Wipf, B.D. Rao, Sparse Bayesian learning for basis selection, IEEE Trans. Signal Process. 52 (2004).
[25] W.R. Jacobs, T. Baldacchino, S.R. Anderson, Sparse Bayesian identification of polynomial NARX models, IFAC-PapersOnLine 48 (2015) 172–177.
[26] R. Fuentes, N. Dervilis, K. Worden, E.J. Cross, Efficient parameter identification and model selection in nonlinear dynamical systems via sparse Bayesian learning, in: Proceedings of Recent Advances in Structural Dynamics – (RASD), 2019.
[27] W. Pan, Y. Yuan, J. Goncalves, G.-B. Stan, A sparse Bayesian approach to the identification of nonlinear state-space systems, IEEE Trans. Autom. Control 61 (2016) 182–187.
[28] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. Roy. Stat. Soc.– Series B: Methodol. 39 (1977) 1–38.
[29] M.E. Tipping, A.C. Faul, Fast marginal likelihood maximisation for sparse Bayesian models, in: Proceedings of the 9th AISTATS Conference, 2003, pp. 1–13.
[30] P.R. Dahl, Solid friction damping of mechanical vibrations, AIAA Journal 14 (1976) 1675–1682.
[31] J.W. Liang, B.F. Feeny, Identifying Coulomb and viscous friction from free-vibration decrements, Nonlinear Dyn. 16 (1998) 337–347.
[32] Y.K. Wen, Method of random vibration of hysteretic systems, J. Eng. Mech. Divis. 102 (1976) 249–263.
[33] M. Ismail, F. Ikhouane, J. Rodellar, The hysteresis Bouc-Wen model, a survey, Arch. Comput. Methods Eng. 16 (2009) 161–188.
[34] X. Zhu, X. Lu, Parametric identification of Bouc-Wen model and its application in mild steel damper modeling, Proc. Eng. 14 (2011) 318–324.
[35] A.E. Charalampakis, V.K. Koumousis, Identification of Bouc-Wen hysteretic systems by a hybrid evolutionary algorithm, J. Sound Vib. 314 (2008) 571–585.
[36] A.E. Charalampakis, C.K. Dimou, Identification of Bouc-Wen hysteretic systems using particle swarm optimization, Comput. Struct. 88 (2010) 1197–1205.
[37] A. Ben Abdessalem, N. Dervilis, D.J. Wagg, K. Worden, Model selection and parameter estimation in structural dynamics using approximate Bayesian computation, Mech. Syst. Signal Process. 99 (2018) 306–325.
[38] M.D. Spiridonakos, E.N. Chatzi, Metamodeling of dynamic nonlinear structural systems through polynomial chaos NARX models, Comput. Struct. 157 (2015) 99–113.
[39] K. Chandrasekhar, J.A. Rongong, E.J. Cross, Mechanical behaviour of tangled metal wire devices, Mech. Syst. Signal Process. 118 (2019) 13–29.
[40] K. Worden, R.J. Barthorpe, E.J. Cross, N. Dervilis, G.R. Holmes, G. Manson, T.J. Rogers, On evolutionary system identification with applications to nonlinear benchmarks, Mech. Syst. Signal Process. (2018).
[41] L. Ljung, System Identification: Theory for the User, Prentice Hall, 1987.
[42] T.J. Rogers, K. Worden, G. Manson, U.T. Tygesen, E.J. Cross, A Bayesian filtering approach to operational modal analysis with recovery of forcing signals, in: International Conference on Noise and Vibration Engineering (ISMA), Leuven, Belgium, 2018.
[43] T. Wigren, J. Schoukens, Three free data sets for development and benchmarking in nonlinear system identification, in: Proceedings of European Control Conference (ECC), Zurich, Switzerland, 2013, pp. 2933–2938.
[44] A. Janot, M. Gautier, M. Brunot, Data set and reference models of EMPS, in: 2019 Workshop on Nonlinear System Identification Benchmarks, Eindhoven, The Netherlands, 2019.
[45] K. Worden, Data processing and experiment design for the restoring force surface method. part I: integration and differentiation of measured time data, Mech. Syst. Signal Process. 4 (1990) 321–344.
[46] E.I. George, R.E. McCulloch, Approaches for Bayesian variable selection, Stat. Sinica 7 (1997) 373.
[47] A.C. Faul, M.E. Tipping, Analysis of sparse Bayesian learning, in Advances in Neural Information Processing Systems, 2002, pp. 383–389.