Mechanical Systems and Signal Processing 112 (2018) 194–232

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

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

On evolutionary system identification with applications to nonlinear benchmarks

K. Worden *, R.J. Barthorpe, E.J. Cross, N. Dervilis, G.R. Holmes, G. Manson, T.J. Rogers

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

ARTICLE INFO

Article history:
Received 8 April 2017
Received in revised form 23 March 2018
Accepted 2 April 2018

Keywords:
Nonlinear system identification
Evolutionary optimisation
Differential evolution
SADE
JADE
White-, grey-, black-box models

ABSTRACT

This paper presents a record of the participation of the authors in a workshop on nonlinear system identification held in 2016. It provides a summary of a keynote lecture by one of the authors and also gives an account of how the authors developed identification strategies and methods for a number of benchmark nonlinear systems presented as challenges, before and during the workshop. It is argued here that more general frameworks are now emerging for nonlinear system identification, which are capable of addressing substantial ranges of problems. One of these frameworks is based on evolutionary optimisation (EO); it is a framework developed by the authors in previous papers and extended here. As one might expect from the 'no-free-lunch' theorem for optimisation, the methodology is not particularly sensitive to the particular (EO) algorithm used, and a number of different variants are presented in this paper, some used for the first time in system identification problems, which show equal capability. In fact, the EO approach advocated in this paper succeeded in finding the best solutions to two of the three benchmark problems which motivated the workshop. The paper provides considerable discussion on the approaches used and makes a number of suggestions regarding best practice; one of the major new opportunities identified here concerns the application of grey-box models which combine the insight of any prior physical-law based models (white box) with the power of machine learners with universal approximation properties (black box).

1. Introduction

In March of 2016, an interesting meeting on the subject of nonlinear system identification (NLSI) took place at Vrije Universiteit Brussel (VUB) in the Belgian capital. The meeting was interesting for two reasons; in the first case, it was organised with the intention of bringing together experts from the disciplines of electrical engineering, mechanical engineering and machine learning, in order to draw out common elements of best practice for nonlinear system modelling/identification, and also to exploit any potential synergies. The second feature of interest was that the meeting was organised around the discussion of three benchmarks for NLSI, each designed in such a way as to challenge theory and practice in specific ways.

The three benchmark problems were as follows:

* Corresponding author.
E-mail address: k.worden@sheffield.ac.uk (K. Worden).

https://doi.org/10.1016/j.ymssp.2018.04.001
0888-3270/© 2018 The Authors. Published by Elsevier Ltd.

- **A Bouc-Wen Hysteretic System**. The NLSI challenges of this benchmark were associated with the fact that the system of interest had an unmeasurable state in its equations of motion, and the fact that the model form was not linear in the parameters. The system equations were encoded in a Matlab p-file, which allowed participants complete freedom in choosing the form of the excitation used for identification. The data were thus generated by computer simulation, although noise was added to the response in the p-file to give an element of realism.
- **A Wiener-Hammerstein System with Process Noise**. The main challenge associated with this benchmark was that the system was a block-structured system where significant noise was added to an internal state. The system was encoded in an electronic circuit and was thus experimental (at least from an electrical engineering point of view). Although participants did not have the freedom to completely experiment with excitation signals, they were allowed to propose signals, which were then used in a number of measurement campaigns in order to collect data for the benchmark exercise.
- **A Cascaded Tanks System**. This was an experimental liquid level system, in which fluid passed between two tanks. The main challenges were that an unmeasured state was present again, but mainly that the record of data for NLSI was very short. Further challenges arose due to the overflow of the tanks, which introduced a hard saturation nonlinearity and some uncertainty in the form of process noise. Participants were not given any control of the experiment in this particular case.

This paper is a record of the participation of a team of University of Sheffield (UoS) academics and researchers. It comprises a summary of a keynote presentation by one of the authors, followed by detailed descriptions of how the team attempted to solve the benchmark problems. It is argued here that general frameworks are beginning to emerge for NLSI, which are capable of addressing ranges of disparate problems. Two of the main candidates for such a general framework are those based on evolutionary optimisation (EO) and Bayesian inference. In fact, the algorithms applied here were taken from the EO approach developed by the authors over a number of years and extended in order to address the benchmarks. The power of the EO framework is clearly evidenced by the fact that it provided the best solutions to two out of the three benchmark problems at the focus of the workshop. As one might expect from the 'no-free-lunch' theorem for optimisation [1], it would be surprising if a single variant of the EO algorithm stood out as the overall best choice, so the viewpoint here has been to present a number of possibilities (reflecting the slightly different tastes of the authors and illustrating the range). Although the EO approach is favoured in this paper, the Bayesian framework for NLSI is also very powerful and is being pursued by the authors; however, there is simply not room here to compare the two frameworks. If the reader is interested in seeing how modern Bayesian methods can contribute to NLSI they can consult the references in the following section.

One of the main contributions of this paper is to highlight and develop the idea of using *grey-box* models for NLSI. Grey box models combine the insight of a physics-based (white box) model with the explanatory power of machine learners (black box) which have universal approximation/representation properties. In fact, the grey-box model presented here for Benchmark Three also combines the power of the EO and Bayesian approaches by using a Gaussian process NARX (Nonlinear AutoRegressive with eXogeneous inputs) model to capture behaviour missed by the physical model and to thus substantially improve predictions.

It is important to note two facts. The first is that the paper has been formed in order to give an honest account of the identification results, as presented at the workshop; it deliberately does not contain any results which exploit lessons learned *during or after* the meeting, although those lessons have led to a great deal of progress for the participants since. The second fact to note, is that four separate studies are presented here, each carried out by separate subgroups of the UoS team; this means that the studies may reflect slightly different views on the *practice* of NLSI – the authors are all in general agreement about the aims, objectives and importance of the subject. The amount of ground covered here also means that the paper is rather lengthy; this is sadly unavoidable if it is to reflect properly the weight of the work conducted.

The layout of the paper is as follows: the following section summarises one of the workshop keynotes, and discusses the question of whether NLSI can be reduced to a problem in machine learning. The three subsequent sections, outline in turn, how the authors approached the three VUB benchmark problems.

2. Is system identification simply machine learning?

2.1. Introduction

The material of this section was originally the subject of a keynote at the VUB Workshop; as a result, its remit is broader than the material which follows, which presents studies of the specific benchmark exercises. However, the discussion will touch on various issues which surface during the detailed studies and will also attempt to capture aspects of current thinking in terms of NLSI within the Engineering Dynamics community. In order to faithfully cover what was discussed in the keynote, it will be necessary to go over a little previously-published ground; however, this will also help to make the paper more self-contained.

Historically, one could argue that the main developments in the general theory of linear SI have come from the electrical engineering and control communities. This work resulted in a comprehensive and rigorous body of material which is visible through classic texts and monographs like [2,3]. Although general ideas from SI were certainly adopted by the Engineering Dynamics community, the main developments there are associated with a specific method – *modal analysis* [4]. Modal analysis arose naturally as a result of the fact that linear engineering dynamics is concerned with a specific system of second-order differential equations, derived from Newton's second law, and usually expressed in the matrix form,

$$M\ddot{\underline{y}} + C\dot{\underline{y}} + K\underline{y} = \underline{x} \tag{1}$$

where $x(t)$ is the excitation force and $y(t)$ is the displacement response of the system of interest; M, C and K are, respectively, the so-called system *mass*, *damping* and *stiffness* matrices. (Throughout this paper, capitals will denote matrices and underlines will denote vectors; overdots will indicate differentiation with respect to time.) Modal analysis works by using matrix diagonalisation to reduce an N-Degree-of-Freedom system to N Single-Degree-of-Freedom (SDOF) systems; it has proved overwhelmingly successful in the context of linear engineering dynamics.

The main limitations of the linear approaches discussed (and they are *serious* limitations) is that they do not adequately address nonlinearity, nonstationarity and uncertainty. The issue of nonstationarity in SI deserves an article of its own (in fact, a recent special issue of MSSP was dedicated to this matter[1]) and is not considered further here. In terms of *nonlinearity*, a great deal of work has been carried out over the last fifty years in particular, but it is fair to say that no panacea has emerged. Instead, at least as far as structural dynamics is concerned, a *toolbox* philosophy has evolved. Quite a wide variety of approaches to nonlinear SI (NLSI) have been developed, each with its own optimal domain of applicability, as summarised in [5–7]. At the moment, there is the prospect of more generally applicable methods emerging, and this will be discussed in more detail a little later.

In terms of *uncertainty*, there are a number of interesting matters to discuss. It is fair to say that Engineering Dynamics has largely assumed throughout its history that *deterministic* models are appropriate for system modelling and prediction; recent (and not so recent) developments suggest otherwise. For example, the detailed dynamical modelling of biomechanical systems is becoming more commonplace, and this faces the immediate problem that the mechanical properties of tissue vary considerably from individual to individual and even within a single individual; this presents serious issues in terms of the construction of predictive models able to generalise from person to person. Because of the problem of uncertainty, deeper probabilistic reasoning is becoming much more common in the analysis of dynamical problems. (Of course, probability theory is only one of a large range of possible uncertainty theories [8]; however it is by far the most highly developed and pervasive). Many of the lessons learned recently have come from the field of *machine learning*.

In some areas of engineering dynamics, uncertainty has been (at least partially), accommodated in theory and practice for a long time, and it is interesting to note that SI is a good example of this – at least in the linear case. It has long been recognised, that to identify a parametric model from measured data, one has to allow for the fact that noise may be present in any measurements in order that the identified parameters for the model are meaningful. Here, *meaningful* has largely been taken in the past to mean *unbiased*, which is to say that the parameters truly allow modelling of the underlying structure or system of interest, rather than capturing aspects of the noise inherent in the individual set of data used for learning the model. However, quite independently of Bayesian approaches, developments in regularised methods of linear SI have forced a re-evaluation of the meaning and import of bias [9,10]. In general, the inclusion of *noise models* in the linear and nonlinear approaches has often been considered sufficient for the removal of bias; possibly the most principled approach in the nonlinear case is the NARMAX approach pioneered by Billings and co-workers [11]. Despite some progress, one might argue that the probabilistic reasoning that underlies SI was often *hidden* until recently. In particular, despite the fact that many least-squares estimators used for SI are maximum-likelihood estimators under given assumptions, SI users in structural dynamics would often implement algorithms in linear algebra and treat the resulting crisp parameters as constituting 'the model'; the main objective of noise models has been to ensure that there is no systematic bias in the estimated parameters. Furthermore, even if a covariance matrix for the estimates was found, it was usually only used to provide confidence intervals or 'error bars' on the parameters; predictions would still be made using crisp parameter estimates. Such approaches are powerful, but do not fully accommodate the fact that the data may be consistent with a number of different parametric models, and thus only go part way to recognising and accommodating general aspects of uncertainty.

Recent advances in machine learning [12,13] have not only offered more principled and holistic means of addressing the issue of uncertainty, but have also offered the prospect of a more general paradigm for NLSI. In fact, another more general approach has emerged in recent years which is also rooted in the broader discipline of *soft* or *natural* computing [14] – one based on *evolutionary optimisation*. Both of these new developments for NLSI will be discussed in more detail here; however, it is the evolutionary methods which will dominate the detailed investigation of the VUB benchmarks later in the paper. Before discussing the general approaches in detail, it will prove useful to define a little terminology; it will be useful to divide predictive models into two classes: *white* and *black-box* models.

- A *white-box* model here is one where the equations of motion have been derived from the underlying physics of the problem and the model parameters have direct physical meanings; e.g. finite element models or the *lumped-mass* model represented by Eq. (1).
- A *black-box* model is formed by taking a class of models with some universal approximation property and learning the parameters from data; in such a model, like a neural network, the parameters will not generally be physical.

SI or *learning from data* in machine learning terms, is essential to a black-box approach; for the white-box model, parameters may be estimated from data or fixed by physical laws.

[1] Volume 47: issues 1 and 2.

2.2. Evolutionary optimisation

As far as the current authors are concerned, the original motivation for developing optimisation-based methods for NLSI was not so much rooted in uncertainty analysis, but with other technical problems associated with identifying nonlinear systems. These technical problems are most easily discussed with respect to specific systems; as the Bouc-Wen system was the original system of interest [15] and also appears later as one of the VUB benchmarks, it makes good sense to introduce it here.

The general Bouc-Wen (BW) model [16,17] is a nonlinear hysteretic restoring force model, where the total restoring force is composed of a polynomial non-hysteretic and a hysteretic component based on the displacement $y(t)$ and velocity $\dot{y}(t)$ time-histories. The general Single-Degree-of-Freedom (SDOF) hysteretic system described in the terms of Wen [17] is,

$$m\ddot{y} + r(y,\dot{y}) + z(y,\dot{y}) = x(t) \tag{2}$$

where $r(y,\dot{y})$ is the polynomial part of the restoring force and $z(y,\dot{y})$ the hysteretic; m is the mass of the system and $x(t)$ is the excitation force. The polynomial component may be assumed linear if desired (and justified), but is essentially a static nonlinear function of y and $\dot{y}$. In contrast, the hysteretic component is defined via an additional equation of motion [17],

$$\dot{z} = A\dot{y} - \alpha|\dot{y}|z^n - \beta\dot{y}|z^n| \tag{3}$$

for n odd, or,

$$\dot{z} = A\dot{y} - \alpha|\dot{y}|z^{n-1}|z| - \beta\dot{y}z^n \tag{4}$$

for n even.

The parameters α, β and n govern the shape and smoothness of the hysteresis loop (this will be elaborated later). As a system identification problem, this set of equations presents a number of difficulties, foremost are:

- The variables available from measurement will generally be the input x and some form of response: displacement, velocity or acceleration. Even if all the response variables mentioned are available, the state z is not measurable and therefore it is not possible to use Eqs. (3) or (4) directly in a least-squares formulation.
- The parameter n enters the state Eqs. (3) and (4) in a nonlinear way; this means that a linear least-squares approach is not applicable to the estimation of the full parameter set; at the least, some iterative nonlinear least-squares approach is needed as in [18].

Both of these difficulties can be addressed by adopting an (evolutionary) optimisation approach. The SI problem is simply framed as a minimisation problem with the objective/cost function defined as a normalised mean-square error between the 'measured' data and that predicted using a given parameter estimate, i.e.,

$$J(m,c,k,\alpha,\beta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i(m,c,k,\alpha,\beta,n))^2 \tag{5}$$

where the error is framed in terms of displacement response, N is the number of measured points and the caret denotes a quantity predicted by the model. In general, any metric measuring the 'distance' between measured data and predictions can be used; the authors usually use a normalised version of (5),

$$J(m,c,k,\alpha,\beta) = \frac{100}{N\sigma_y^2}\sum_{i=1}^{N}(y_i - \hat{y}_i(m,c,k,\alpha,\beta,n))^2 \tag{6}$$

where σ_y^2 is the variance of the measured displacements. This cost function has the following useful property; if the mean of the output signal is used as the model i.e. $\hat{y}_i = \bar{y}$ for all i, the cost function is 100.0 (and can be thought of as a percentage). This definition of cost could just as easily be used with velocity or acceleration data. The clear advantage of the optimisation approach is that it does not require measurements of the *latent* variable z and is insensitive to whether the model is linear in the parameters.

So far, this type of problem could be approached using any optimisation method e.g. a Gauss-Newton approach was adopted in [18]. However, evolutionary approaches offer various advantages, and in order to show this, it is useful to give a concrete example. As variants on the *Differential Evolution* (DE) algorithm were used to analyse two of the VUB benchmarks, it makes sense to describe the basic DE algorithm here.

The standard DE algorithm of [19] attempts to transform a randomly-generated initial population of parameter vectors into an optimal solution through repeated cycles of evolutionary operations, in this case: *mutation*, *crossover* and *selection*. In order to assess the suitability of a certain solution, a cost or fitness function is needed; the cost function in Eq. (6) is the one used here. Fig. 1 shows a schematic for the DE procedure for evolving between populations. The following process is repeated with each vector within the current population being taken as a *target vector*; each of these vectors has an associated cost taken from Eq. (6). Each target vector is pitted against a *trial vector* in a competition which results in the vector with lowest cost advancing to the next generation.

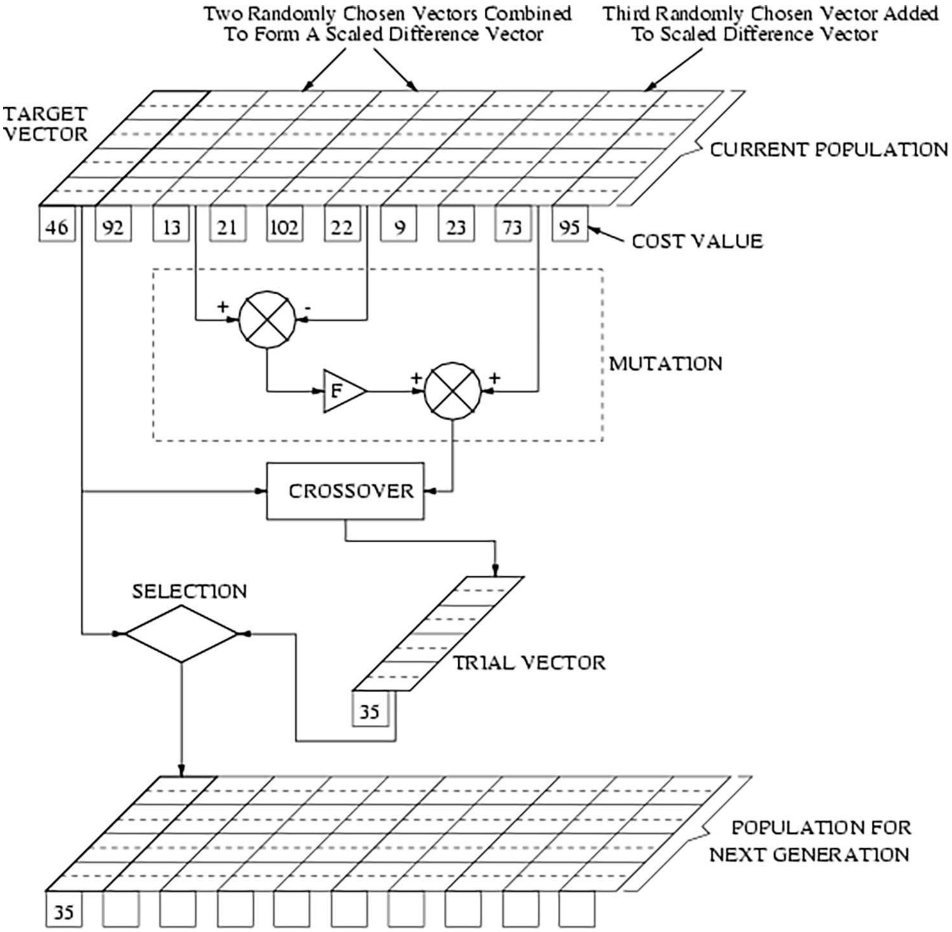

Two Randomly Chosen Vectors Combined To Form A Scaled Difference Vector
Third Randomly Chosen Vector Added To Scaled Difference Vector
TARGET VECTOR
CURRENT POPULATION
46
92
13
21
102
22
9
23
73
95
COST VALUE
+
-
F
+
+
MUTATION
CROSSOVER
SELECTION
TRIAL VECTOR
35
POPULATION FOR NEXT GENERATION
35

Fig. 1. Schematic for the standard DE algorithm.

The mutation procedure used in basic DE proceeds as follows. Two vectors A and B are randomly chosen from the current population to form a vector differential $A - B$. A *mutated* vector is then obtained by adding this differential, multiplied by a scaling factor F, to a further randomly chosen vector C to give the overall expression for the mutated vector: $C + F(A - B)$. The scaling factor, F, is often found to have an optimal value between 0.4 and 1.0.

The *trial vector* is the child of two vectors: the target vector and the mutated vector, and is obtained via a crossover process; in this work uniform crossover is used. Uniform crossover decides which of the two parent vectors contributes to each chromosome of the trial vector by a series of $D - 1$ binomial experiments. Each experiment is mediated by a crossover parameter C_r (where $0 \leqslant C_r \leqslant 1$). If a random number generated from the uniform distribution on $[0, 1]$ is greater than C_r, the trial vector takes its parameter from the target vector, otherwise the parameter comes from the mutated vector. In order to ensure that all trial vectors differ from their associated target vector, even if $C_r = 0$, a single chromosome in the trial vector is randomly chosen to take the corresponding value from the mutated vector.

This process of evolving through the generations is repeated until the population becomes dominated by only a few low cost solutions, any of which would be suitable. Like the vast majority of optimisation algorithms, convergence to the global minimum is not guaranteed; however, one of the benefits of the evolutionary approach is that it is more resistant to finding a local minimum. In fact, this usually proves to be the main benefit. The other advantage of the approach is that the algorithm does not need estimates of the gradients or Hessians of the parameters.

As discussed above, evolutionary optimisation provides a useful framework for NLSI that overcomes a number of technical problems that one encounters in trying to use standard least-squares methods; in fact, some variant or other has been used in order to address each of the benchmark studies discussed later. Where the evolutionary approaches can be found wanting, is in their accommodation of uncertainty. In fact, it is possible to estimate confidence intervals for parameters [20], but this does not amount to a comprehensive treatment of uncertainty, as observed earlier. One aspect of the evolutionary approaches which is almost never exploited, is that they return a population of solutions at every generation – all the information obtained throughout this process could be used to account for uncertainty and could result in more robust predictions.

Generally speaking, the evolutionary approaches are probably best applied for white-box models. This is because the size of the population required is a function of the number of parameters estimated; one usually chooses the number of individuals to be five or ten times the number of model parameters. Larger populations will lead to more computational cost, and if evaluation of the objective function is not fast, the cost may be prohibitive. As white-box models are almost always more parsimonious in terms of parameters, they are singled out for evolutionary modelling. This observation has not stopped various people attempting to use evolutionary algorithms to train neural networks, for example, but the evidence shows clearly that this is not usually a good idea.

Unlike the evolutionary approaches, the other general framework for NLSI which is emerging, is almost entirely motivated by a desire to understand and account for uncertainty, and this will be discussed next.

2.3. Bayesian inference

As supported by recent work in the machine learning community, a more robust approach to parameter estimation, and also *model selection*, can be formulated on the basis of *Bayesian* principles [12,21,13]. It will be shown that, among the potential advantages offered by a Bayesian formulation are:

- The estimation procedure will return parameter distributions rather than point estimates of parameters.
- Predictions can (in principle) be made by integrating over all possible models consistent with the data, weighted by their probabilities.
- Evidence for a given model structure can be computed, leading to a principled means of model selection.

The Bayesian approaches to NLSI/system models can be applied to both white- and black-box models; in fact, methods for black-box models arguably emerged first e.g. Bayesian learning algorithms for Multi-Layer Perceptron (MLP) neural networks [22]. In terms of white-box models, Bayesian methods are not new to structural dynamics, as evidenced by over 20 years of work by Jim Beck and colleagues [23–27]; however, they have by no means been *fully exploited*. More recently, Bayesian ID methods for white-box differential equation models have emerged in the context of *systems biology* [28,29].

In order to discuss the advantages of a Bayesian approach, it is useful to re-state what the problem of SI is, i.e. given measured data from a structure, how does one infer the equations of motion which 'generated' the data. Although the problem can be stated simply, it has a number of technical difficulties and is generally not at all easy to solve. At the base of the issues is the fact that SI is an inverse problem of the second kind and can be extremely ill-posed even if the underlying equations are assumed to be linear in the parameters of interest; the 'solution' may not be unique. Furthermore, if the equations of motion of the system of interest are not linear in the parameters, the difficulties multiply.

Much of the difficulty can be blamed on *uncertainty* again; even if the issue is simply that the measurements or data from a system will almost always be contaminated by some form of *random noise*. For notation, assume data $D = \{(x_i, y_i), i = 1, \dots, N\}$ of sampled inputs x_i and outputs y_i. If there is no noise, then the identification algorithm of choice should give a *deterministic* estimate of any system parameters $\underline{w}$; however, if noise $\epsilon(t)$ is present, $\underline{w}$ will become a random variable conditioned on D. In this context one arguably no longer wishes an *estimate* of $\underline{w}$, but to specify ones belief in its value, expressed through some appropriate uncertainty framework. If the framework of choice is probability theory, the problem becomes one of estimating the probability density function of the parameters $p(\underline{w}|D, \mathcal{M})$, where the density is conditioned on the *training* or estimation data D, but also on the choice of model $\mathcal{M}$. Thus, in the presence of noise, the most one can learn from any data is the probability density function of the parameters; however, in a probabilistic context, *this is everything*.[2]

Even so, the uncertainty in any estimated parameters is by no means the only uncertainty of interest. The usual objective of SI is to provide some sort of predictive model i.e. a mathematical model which can estimate or predict system outputs if a *different* system input is given. If a crisp parameter estimate is all one has, the best one can do is a crisp prediction; however, if a parameter distribution is known, one can potentially do better and form a *predictive distribution*. Suppose the response to a new input sequence $\underline{x}^*$ were desired, one could in principle find the density for the predicted outputs,

$$\underline{y}^* \sim p(\underline{y}^* | \underline{x}^*, \underline{w}, D, \mathcal{M}) \tag{7}$$

and then the mean of this distribution would give 'best' estimates for predictions, and the covariance would allow one to establish confidence intervals. Note that the prediction assumes the presence of a $\underline{w}$. In practice, one might use the mean or the mode of the parameter distribution, but these are still *point estimates*, not reflecting the uncertainty in the parameters. A fully Bayesian prediction strategy would, again, in principle, allow one to *marginalise* over parameter estimates, i.e.,

$$p\left(\underline{y}^* | \underline{x}^*, D, \mathcal{M}\right) = \int p\left(\underline{y}^* | \underline{x}^*, \underline{w}, \mathcal{M}\right) p(\underline{w}|D, \mathcal{M}) d\underline{w} \tag{8}$$

[2] The notation conventions for probabilistic quantities in the paper are as follows. A lower-case p signifies that the quantity is a *density* associated with a continuous random variation; an upper-case P denotes the probability associated with a discrete random variable.

This is a very powerful idea: allowing for a fixed model *structure*, *one makes predictions using an entire set of parameters consistent with the training data*; each point in parameter space weighted according to the likelihood of the given data. In practice, there are problems in implementing the full Bayesian approach due to the difficulty of evaluating the integral in (8).

Another advantage offered by the Bayesian approach is that it can potentially weigh the evidence for competing model forms. Suppose the true model structure must be one of a finite number $\{\mathcal{M}_i, i = 1, \ldots, M\}$; one can imagine computing the probability of observing the data $P(D|\mathcal{M}_i)$ and selecting the model with highest probability. Even more in the Bayesian spirit, one could marginalise over *all possible* model structures e.g. for prediction,

$$p(\underline{y}^*|\underline{x}^*, D) = \sum_{i=1}^{M} p(\underline{y}^*|\underline{x}^*, \mathcal{M}_i, D)P(\mathcal{M}_i|D) \tag{9}$$

Unfortunately, the posterior over models $P(\mathcal{M}_i|D)$ is difficult to compute. If one appeals to Bayes theorem in the form,

$$P(\mathcal{M}_i|D) = \frac{p(D|\mathcal{M}_i)P(\mathcal{M}_i)}{p(D)} \tag{10}$$

and assumes equal priors on models, one arrives at the *Bayes factor*,

$$B_{ij} = \frac{P(\mathcal{M}_i|D)}{P(\mathcal{M}_j|D)} = \frac{p(D|\mathcal{M}_i)}{p(D|\mathcal{M}_j)} \tag{11}$$

which weights the evidence for two models in terms of marginal likelihoods of the data given the models. Sadly, the marginal likelihoods are usually intractable integrals.

In summary, assuming one can overcome some of the computational difficulties involved (e.g. high-dimensional numerical integrals), the Bayesian framework for NLSI is very general indeed. Like the evolutionary approaches, the Bayesian ones have no technical problems with unmeasured states or models which are nonlinear in the parameters. All of this suggests that NLSI has been reduced to the problem of finding appropriate computational algorithms for machine learning; that the problem of NLSI has been *reduced* to one of machine learning. The next part of the discussion here will argue that this is not the case.

2.4. Is NLSI just machine learning?

To recap, it appears that NLSI can be formulated in terms of a machine learning approach; the problems raised so far relate only to difficulties in numerical calculations. The argument here is going to be that NLSI is more than this, and the first argument will be based on going back to the idea of *uncertainty*.

The previous discussion has highlighted the importance of considering uncertainty; however, in the reality of modelling engineering systems and structures, one needs to go a little further and address the issue that there are two main *types* of uncertainty.

Aleatory Uncertainty: is essentially *randomness*. Examples are measurement noise superimposed on data or the behaviour of truly stochastic systems (i.e. Brownian motion). This is uncertainty which cannot be removed – *irreducible* uncertainty.

Epistemic Uncertainty: is essentially *ignorance*. It commonly arises because all of the underlying causes (physics) of a problem are not known. This type of uncertainty can be removed by designing experiments to learn the missing physics – it is *reducible*.

The case will be presented here that even the Bayesian approach discussed earlier is not adequate to address the full issues of uncertainty because it is only really formulated to deal with aleatory uncertainty.[3] In reality, there may be ignorance of the *form* of the model, even of the underlying physical principles of the processes at work.

Dealing with epistemic uncertainty leads one naturally to the idea of a *Grey-Box* model. A grey-box model is one for which only some of the underlying physics is specified i.e. it has a white-box component; one can then attempt to reduce any model error by adding a nonparametric component and learning its behaviour from data. This observation in turn leads to the idea of two types of grey-box models:

- A grey-box model will be said to be of **Type A** if the nonparametric component is a true black-box model.
- A grey-box model will be said to be of **Type B** if the nonparametric component is motivated in some way by physics rather than simple possession of a universal approximation property.

Type B models are arguably the result of physics and creativity and cannot be found by learning from data alone. Two examples will be considered here.

[3] Although it will not be pursued here, one might argue that it is not logical to speak of *irreducible* uncertainty at all. To prove that uncertainty is truly aleatory, one would need to establish that there is *no possible experiment* which could reduce it. This is surely not provable, so the statement that a given uncertainty is aleatory is not falsifiable and thus scientifically meaningless [30].

2.4.1. Friction models

Friction is *dynamically* the resistance to motion produced by interfacial contacts between two bodies in relative motion. The phenomenon has a microstructural origin, and the detailed physics is the subject of the discipline of *tribology*. A true white-box model of friction would be prohibitively costly for most purposes of structural dynamics, so simplified *effective* models are usually assumed. The most simplistic semi-physical representation is via the *Coulomb* model which simply reverses the action of a constant force when the direction of motion reverses; in the context of an SDOF oscillator, one has,

$$m\ddot{y} + F(\dot{y}) + ky = x(t)$$

$$F(\dot{y}) = F_c \mathrm{sgn}(\dot{y}) \tag{12}$$

The Coulomb model is very limited, but is conceptually simple and very convenient for SI. Among the immediate limitations of the model is the fact that it does not distinguish between static and dynamic friction and that it does not account for the hysteresis loops which are observed in reality. In order to accommodate the hysteresis effect, the more sophisticated Dahl model was introduced in 1968; the basic model has the form,

$$m\ddot{y} + \sigma_0 z + ky = x(t)$$

$$\dot{z} = \dot{y}\left(1 - \mathrm{sgn}(\dot{y})\frac{\sigma_0 z}{F_c}\right)\left|1 - \mathrm{sgn}(\dot{y})\frac{\sigma_0 z}{F_c}\right|^{\delta_D} \tag{13}$$

where the z is a state variable interpreted as the elastic deformation of surface asperities of adjacent bodies (note the resemblance to the Bouc-Wen model in this respect; this is a characteristic of hysteresis models) and σ_0 represents a sort of average asperity stiffness and δ_D determines the shape of the hysteresis but, in the literature, is often set to unity. The Dahl model is a Type B grey-box model; the white-box component is simply an SDOF oscillator, while the black-box component is constructed by considering a model of the average movement/displacement of microscopic asperities. The SI problem has become more difficult than the Coulomb case (the model has an unmeasured state and is nonlinear in the parameters) but gives a better representation of the dynamics, respecting better as it does, the underlying physics of the problem.

The Dahl model motivated the construction of a better model – the *LuGre* model [31]. While the Dahl model captures the difference between static and dynamic friction (sometimes called *predisplacement*) and hysteresis, it is unable to account for stick-slip behaviour and the so-called *Stribeck* effect (decrease of friction with velocity over a certain velocity regime; visible in Fig. 2(a)). A simplistic view of the LuGre model is that it adds an effective damping component for the average asperity movement (Fig. 3).

The equations of motion for the LuGre model incorporated into the motion of an SDOF oscillator are,

$$m\ddot{y} + \sigma_0 z + \sigma_1 \dot{z} + \sigma_2 \dot{y} + ky = x(t)$$

$$\dot{z} = \dot{y} - \frac{\sigma_0 |\dot{y}|}{s(\dot{y})} z$$

$$s(\dot{y}) = F_C + (F_S - F_C)\exp\left\{-\left(\frac{\dot{y}}{v_s}\right)^{\delta_{vs}}\right\} \tag{14}$$

where F_S and F_C are the static and Columb friction coefficients respectively, $s(\dot{y})$ is the Stribeck curve, v_s is the Stribeck velocity and δ_{vs} is the Stribeck shape factor ($\delta_{vs} = 2$ is often used in the literature). A parametrised model of the Stribeck curve can be included in the overall identification/learning problem. Note that, although some of the parameters do not have completely clear physical interpretations, others – like F_C and F_S – encode simple, direct physics.

Other friction models have evolved in turn from the LuGre model, including the *Leuven integrated friction model* which also accounts for presliding hysteresis [33]. However, the point has been made for now; Type B grey-box models like the Dahl and LuGre models are based on physical and engineering insight and rely on the introduction of model terms which are very problem specific and capture behaviour in a parsimonious manner, that would be difficult for black-box basis terms to capture in such a small number of terms. While the grey-box structures are more challenging from an NLSI viewpoint, they are addressable using the powerful methods discussed earlier; for a study on the issues associated with estimating grey-box friction models more generally, the reader may consult [34].

2.4.2. Hysteresis models

The second example of a class of Type B grey-box models is provided by hysteretic systems. Setting aside friction, hysteretic or memory-dependent phenomena are observed in many areas of physics and engineering such as electromagnetism, phase transitions and elastoplasticity of solids [35]. As in the case of friction, the exact physics at play is often complex and a simplified effective model structure can be sufficient for the purposes of engineering dynamics. As introduced in Section 2.2, one of the most commonly-used effective hysteresis models is the *Bouc-Wen* (BW) model. As shown earlier, the BW model incorporates an unmeasured system state specified by an additional equation of motion, which allows a versatile

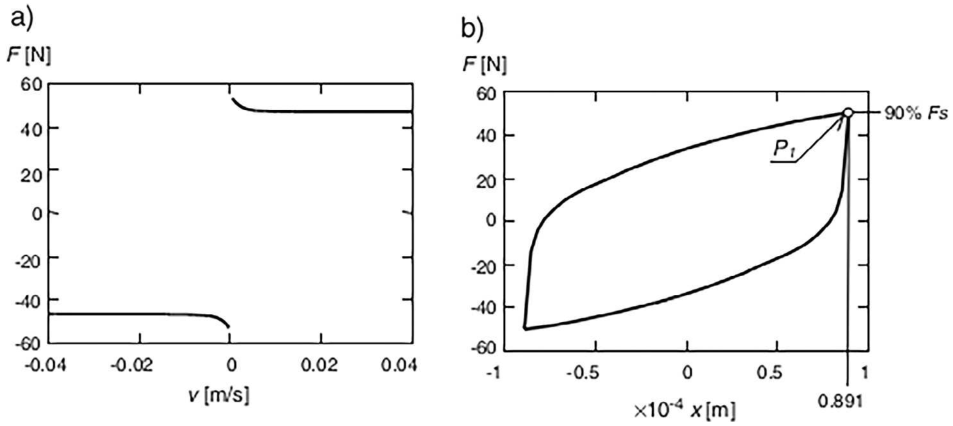


Fig. 2. Results from experimental friction force measurement taken from [32]: (a) shows the Stribeck effect and (b) shows the hysteresis loop.

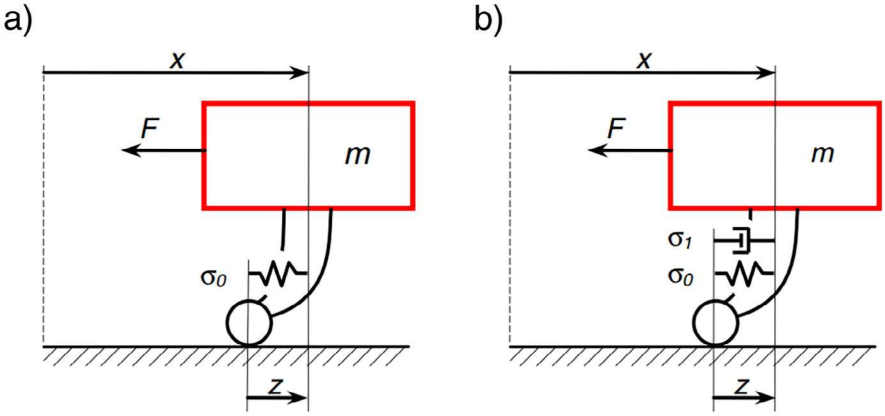


Fig. 3. Graphical representation of variable state z in two friction models: (a) Dahl model, (b) LuGre model (following [32]).

representation of a family of hysteresis loops (Fig. 4). Unlike the friction models, the BW model does not have a direct physical interpretation; however, it has been constructed in order to give the aforementioned versatility.

Sadly, the BW model is not generally versatile enough. Various effects commonly occur in hysteretic systems which cannot be captured by the basic model. One example is given in Fig. 5, which shows pinching of the hysteresis loops for a nailed sheathing connection in a wooden frame [36].

As in the case of the friction models, new terms need to be added in order to capture the missing physics. As before, these are designed as *effective* terms intended to capture the required behaviour without detailed modelling of the microstructural physics which cause it. One of the most successful extensions of the BW model is the *Bouc-Wen-Baber-Noori (BWBN) Model* [37–39]. The BWBN model is designed to capture the pinching effect illustrated earlier, and also to model the strength and stiffness degradation observed in many real hysteretic structural systems. The form of the model is,

$$\dot{z} = \frac{h(z)}{\eta(\epsilon)}\dot{y}\left\{A(\epsilon) - \nu(\epsilon)[\beta \text{sgn}(\dot{y})|z|^{n-1}z + \gamma|z|^{n}]\right\} \tag{15}$$

where $\eta(\epsilon), \nu(\epsilon)$ and $h(z)$ are parameters associated with the strength, stiffness and pinching, degradation effects; $\eta(\epsilon), \nu(\epsilon)$ and $A(\epsilon)$ are increasing functions of the absorbed hysteretic energy ϵ,

$$\eta(\epsilon) = \eta_0 + \delta_\eta \epsilon(t)$$

$$\nu(\epsilon) = \nu_0 + \delta_\nu \epsilon(t)$$

$$A(\epsilon) = A_0 + \delta_A \epsilon(t) \tag{16}$$

The pinching function $h(z)$ is specified as,

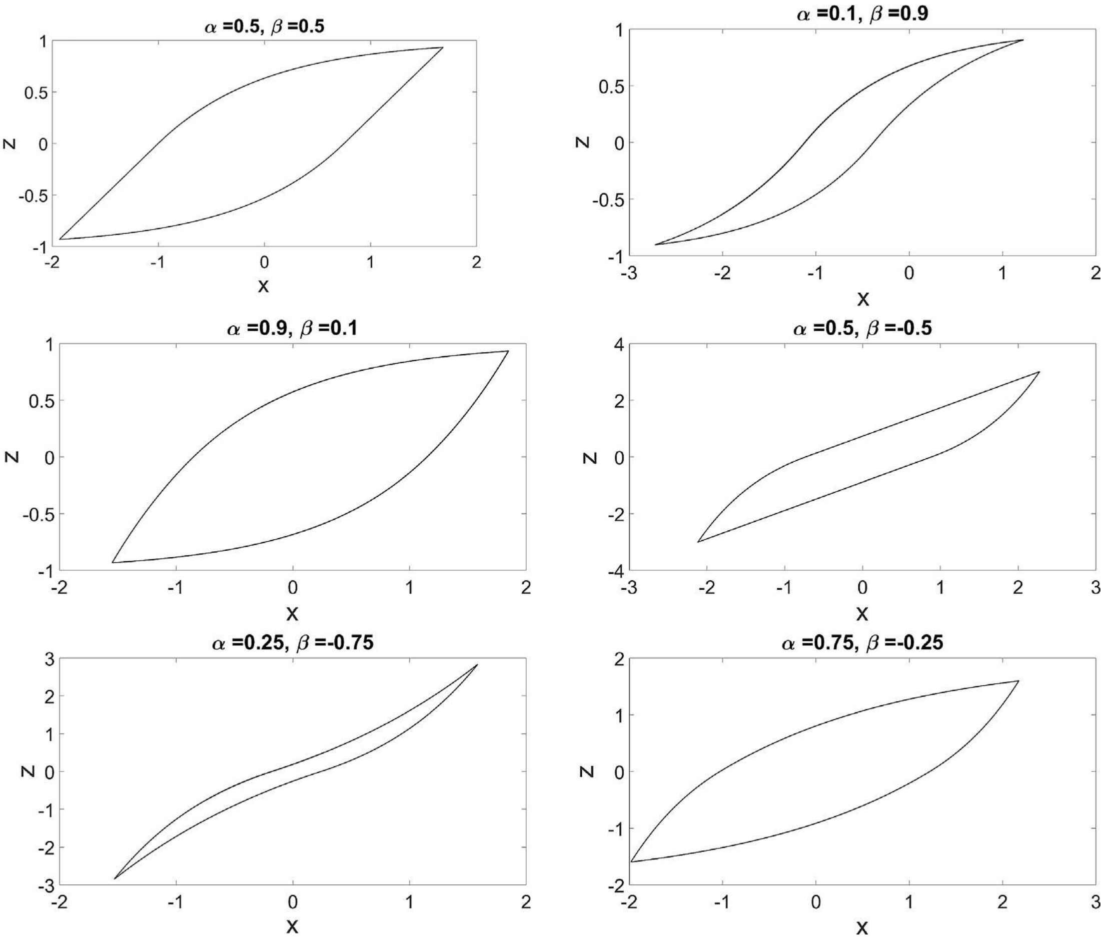


Fig. 4. Samples from the range of hysteresis loops allowed within the BW model (following [17]).

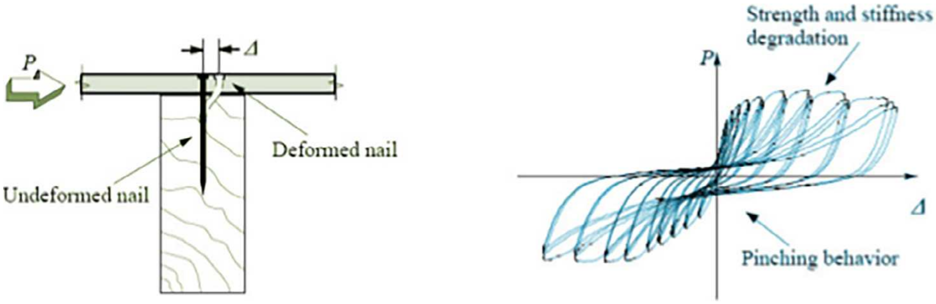


Fig. 5. Illustration of a nailed sheathing connection in a wooden frame and the corresponding pinching hysteresis curve [36].

$$h(z) = 1 - \zeta_1(\epsilon)\exp\left(-\frac{(z\,\mathrm{sgn}(\dot{y}) - qz_u)^2}{\zeta_2(\epsilon)^2}\right) \tag{17}$$

where

$$\zeta_1(\epsilon) = (1 - exp(p\epsilon))\zeta$$

$$\zeta_2(\epsilon) = (\psi_0 + \delta_\psi\epsilon)(\lambda + \zeta_1(\epsilon)) \tag{18}$$

and z_u is the ultimate value of z, specified by,

$$z_u^n = \frac{1}{\nu(\beta + \gamma)} \tag{19}$$

Explaining how the extra terms are motivated is beyond the scope of this paper, the reader should consult the original references. The point here is that the BWBN model is a Type B grey-box model, *informed* by basic physics (like the absorbed energy) but not attempting to capture it in all its microstructural detail. As in the case of the friction models, the result is a parsimonious model which will capture behaviour better than a black-box model with a comparable number of parameters would.

Once one has the model (and some data), machine learning can take over in order to estimate parameters (or to select between candidate models), but getting the model form is another matter. One might argue about whether extraction of a model of this complexity is SI, or whether it is fundamental physics; the current authors would argue that it is SI – it is not intended as an exploration of basic physics, but as a means of providing an effective predictive model.

2.5. Conclusions

The local conclusions for this section are as follows:

- The Bayesian and evolutionary viewpoints on nonlinear SI offer quite general frameworks for the estimation of model parameters. They offer advantages over point parameter estimation (populations in the case of the evolutionary approach, distributions in the case of the Bayesian). In terms of uncertainty analysis, the Bayesian approach is likely to be advantageous even when evolutionary schemes allow estimates of parameter confidences.
- Many of the insights here have come from machine learning work, along with very powerful parameter estimation and model structure detection techniques from the Mechanical and Electrical Engineering communities; this synergy is precisely what the VUB workshop was intended to expose.
- Machine learning is not everything. System identification needs physical insight and expertise in order to overcome the problem of model form uncertainty. This is just as true for grey-box models as white-box models.
- Although it has not been discussed yet, the problems of developing an optimal test or data collection strategy is still not completely possible using automated analysis (this will be discussed in the context of the Benchmark One results).

The paper next moves on to the VUB benchmark problems. Each problem was addressed by subgroups of the overall team. As discussed earlier, each subgroup separately adopted an evolutionary optimisation approach. In two cases, the algorithm adopted was an extension or variant of the Differential Evolution algorithm described earlier; in the other, a type of swarming algorithm motivated by the behaviour of Antarctic krill was adopted. In terms of Benchmark One and the white-box component of Benchmark Three, it would have been a straightforward matter to adopt a Bayesian white-box approach; in fact, some of the current authors have recently applied *Approximate Bayesian Computation* (ABC) techniques to the problem of hysteretic (Bouc-Wen) system identification, with a great deal of success [40,41]. However, applying the methods in order to make a comparison here, would have lengthened the paper considerably. In terms of Benchmark Two, it is probably fair to say that there is no comprehensive Bayesian approach to the identification of Wiener-Hammerstein models agreed upon in the research community; although MCMC methods could clearly be applied if the computational burden did not prohibit it.

3. Benchmark one: a Bouc-Wen system

3.1. Introduction

With this section, begins the study of the VUB benchmarks. In each case, the participants in the workshop were presented with a detailed specification of the problem, including how to access or generate the required data. Only those details which are necessary for understanding the results here will be included in the current paper.

This section is concerned with Benchmark One, which required the system identification of a simulated SDOF Bouc-Wen (BW) hysteresis system. As discussed earlier, the general BW model [16,17] is one of the most commonly-used mathematical models for describing hysteretic behaviour. It was used as a benchmark because of the specific challenges associated with the system possessing an unmeasurable internal variable and the dynamic nature of the nonlinearity. As the current authors had already developed a successful evolutionary approach to BW system identification [15], the opportunity was taken to focus on two methodological issues: benchmarking the identification with a linear model, and choosing appropriate (if not optimal) excitation signals for the generation of training data.

In this case, the detailed specifications of the benchmark can be found in [42]. Restating the equation of motion for the BW oscillator in the form and notation of [42] gives,

$$m_L\ddot{y}(t) + r(y,\dot{y}) + z(y,\dot{y}) = x(t) \tag{20}$$

where m_L is the system mass, $y(t)$ is the displacement and $x(t)$ is the force input. The static nonlinear term, $r(y,\dot{y})$, which only depends upon the current values of displacement and velocity, is assumed to be linear,

$$r(y,\dot{y}) = k_L y + c_L \dot{y} \tag{21}$$

where k_L and c_L are the linear stiffness and linear viscous damping parameters. The history-dependent (hysteretic) nonlinear term, $z(y,\dot{y})$, obeys the first-order differential equation,

$$\dot{z}(y,\dot{y}) = \alpha\dot{y} - \beta\left(\gamma|\dot{y}||z|^{\upsilon-1}z + \delta\dot{y}|z|^{\upsilon}\right) \tag{22}$$

where $\alpha, \beta, \gamma, \delta$ and υ are the Bouc-Wen parameters which control the shape/structure of the hysteresis loop.

For the purposes of the benchmark study, the BW system to be identified was encoded within a Matlab p-file. This gave the user complete freedom over the choice of force input. This freedom was critical to the system identification strategy which will be discussed in the next section. The BW p-file made use of a Newmark numerical integration scheme and detailed guidelines were given regarding upsampling, filtering and decimation for generation of data. The p-file also added a constant level of Gaussian band-limited noise to the output displacement; the force input was assumed to be free of noise. In order to allow comparison of different system identification strategies, two fixed test datasets were provided. One of these datasets was a random phase multisine dataset whilst the other was a sine-sweep dataset. The test data sets were not to be used during the parameter estimation phase of the work but were to allow reporting of a Figure-of-Merit (FoM) for each of the two test datasets, which could be used to compare identification results across participants. The FoM was an un-normalised Root-Mean-Square error, defined by,

$$\text{FoM} = \sqrt{\frac{1}{N_t}\sum_{i=1}^{N_t}(y_i - \hat{y}_i)^2} \tag{23}$$

where N_t is the number of points in the given test set, i is the sample index and the caret denotes quantities estimated by the model.

It is important to note that there can be two important modes of prediction using models generally; although the distinction is most marked when a pure discrete-time model is used rather than a continuous-time one. Suppose the model output at sampling instant i is a function of previous samples of input and output. For the training data, measured inputs *and* outputs are available. This means that one can estimate the current output by substituting measured inputs *and* outputs into the model function. Somewhat confusingly, some sectors of the SI community refer to this mode of estimation as *prediction*. A more stringent test of the model is to start with measured initial conditions and to recursively feed back estimated outputs into the model function in order to generate the next estimate; this mode is called *simulation*. For the benchmarks, the participants were asked to report *simulation* errors, where the distinction was possible. For the continuous-time model of Benchmark One; forward predictions are made by using an initial-value solver, and the mode is naturally *simulation*. Furthermore, in all cases throughout this paper, the final error measures reported are for an independent testing set, unseen throughout the training process.

In terms of *training*, all system parameters were to be identified using alternative training data generated using the supplied p-file.

3.2. System identification strategy and focus of study

As stated above, the focus of the current study will be concerned with presenting the authors' views regarding two important factors, which are sometimes overlooked, related to the success of a nonlinear system identification procedure. The first of these factors is that the choice of forcing input, in terms of both the signal *type* and *amplitude*, plays a critical role in any NLSI procedure. The second factor relates to the idea that it can be difficult to gauge the success of a nonlinear identification procedure without the use of a reference or baseline model. It is held that, before conducting a nonlinear system identification procedure, the best linear system should first be identified.

Although this was not a focal point of the study, the optimisation algorithm employed in this study was partially chosen as it was a good opportunity to showcase a relatively new, robust optimisation technique, which may be unfamiliar to many. The algorithm chosen was the JADE [43] algorithm, a relatively simple, adaptive variant of the better known Differential Evolution (DE) discussed in the introduction here [19]. Whilst the basic DE algorithm is simple and relatively robust, it does require that the hyperparameters, F (in the mutation operation) and C_r (in the crossover operation) be specified before commencing. Poor choices for these hyperparameters may result in lack of convergence or premature convergence. Furthermore, it is often the case that the best values for the hyperparameters may vary during the optimisation process. JADE seeks to address these issues whilst retaining simplicity.

There are a number of key differences between JADE [43] and basic DE [19]. The first occurs during the mutation operation, whereby the current best solution is combined with a differential of two randomly chosen vectors. The scaling factor within the mutation operation is drawn from a probability distribution rather than kept at a fixed value as in DE. The crossover operation is implemented in the same way as in DE, except that the crossover ratio (which decides the probability of a child element being drawn from the original or the mutation matrix) is also drawn from a probability distribution rather than being a fixed constant. The final difference is during the selection phase whereby, in addition to updating the matrix for use in the next generation, the mean values for the hyperparameter distributions are adjusted to take into account previous successful values. Within the selection phase, 'losing' solutions are saved within an archive and may be selected later to

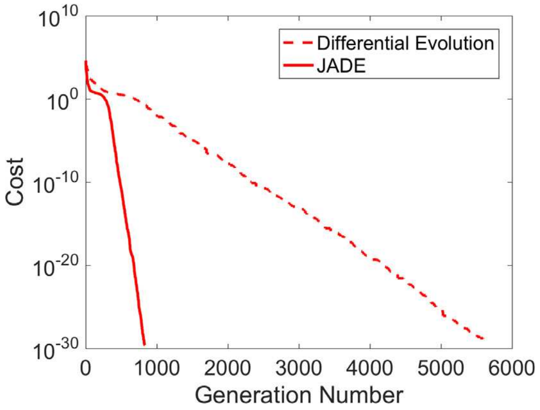


Fig. 6. Plot comparing results of Differential Evolution and JADE optimisation when applied to a 10 dimensional Rosenbrock function.

form the random differential within the mutation operation. The purpose of this archive is to counteract the greedy nature of using the current best solution within the mutation phase, thereby preventing premature convergence. Whilst there is an extra computational cost incurred due to these extra operations in JADE when compared to standard DE, this cost is generally dwarfed by the cost of function evaluations which will be the same in DE and JADE.

In order to verify the algorithm code and to illustrate the potential benefits of JADE compared with basic DE, a standard optimisation benchmark problem, namely a ten-dimensional variant of Rosenbrock's function, was investigated. The results are shown in Fig. 6. Although both the standard DE and JADE identify the best solution to within machine precision, the JADE algorithm only requires around one-sixth of the number of generations. In the authors' experience, this level of saving is typical.

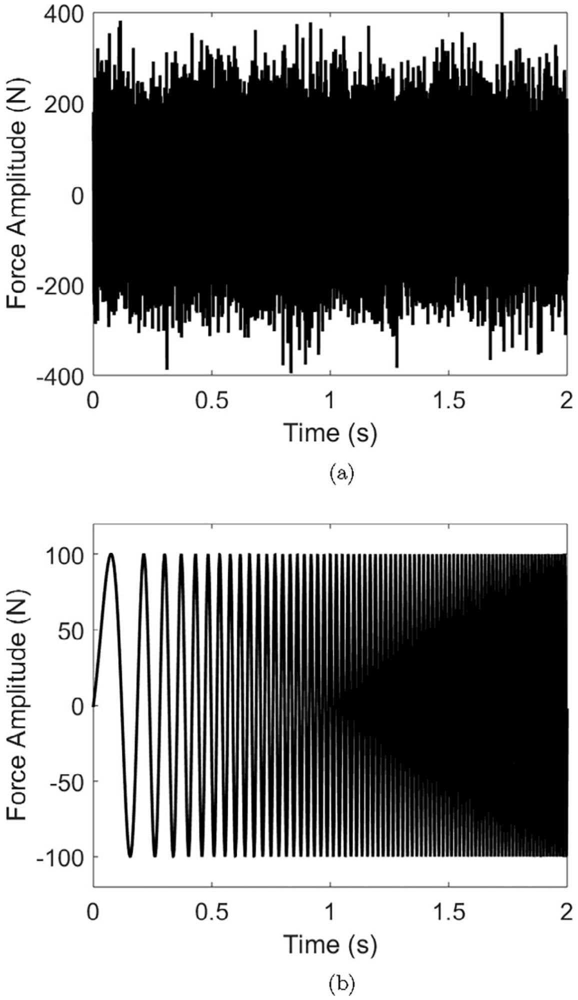
3.3. Results of linear system identification

As stated previously, it is the authors firm opinion that, before conducting a nonlinear system identification, a baseline linear system identification is first required. Apart from serving as a baseline and allowing the practitioner to decide if the extra complexity of a nonlinear ID is justified, the linear SI process is capable of highlighting the point at which a linear system approximation is no longer capable of reproducing the actual system behaviour to within some degree of error. Identification of the point at which the nonlinearity begins to have a significant bearing on the response ensures that the nonlinear SI process is being conducted at an appropriate level of forcing. In the current work, the linear SI will also serve to highlight the influence of the added noise and will help to narrow the search range for the linear parameters within the nonlinear SI.

The aforementioned approach was conducted on the BW system for both a broadband random input and a linear chirp input. For each input type, a number of forcing levels was investigated. For the broadband random input, signal variances of $0.1, 0.5, 1, 5, 10, 50, 100, 500, 1000, 5000, 1 \times 10^4, 5 \times 10^4, 1 \times 10^5, 5 \times 10^5, 1 \times 10^6, 5 \times 10^6$ and 1×10^7 N^2 were considered, whilst the amplitudes chosen for the chirp input were $0.1, 0.5, 1, 5, 10, 50, 100, 500, 1000$ and 5000 N. The forcing levels were chosen to ensure that similar ranges of input signal power were being applied for the random and chirp inputs. Fig. 7 shows an example of each type of input; both each had a duration of 2 s and were sampled at 15 kHz. The frequency of the chirp input varied linearly from 2 Hz to 75 Hz.

The Bouc-Wen system was simulated in Simulink with a 4th-order Runge-Kutta numerical integration scheme. For each of the input types and each of the forcing levels, three runs of the JADE optimisation algorithm were conducted; in each case a population size of 30 was used and the algorithm was run for a total of 100 generations. The parameter bounds for the mass, damping and linear stiffness were set at one order of magnitude below to one order above the parameters given in the problem specification; these bounds are given in Table 1.

On completion of each run, both the lowest FoM (Eq. (23)) and NMSE (Eq. (6)) were stored along with the corresponding estimates for the mass, damping and linear stiffness parameters. Fig. 8 shows the NMSE values for each of the runs for the various forcing levels of both the random and chirp inputs. Fig. 8(a) shows the results from the broadband random input whilst Fig. 8(b) shows the results from the chirp input. There is consistency across the three optimisation runs at each forcing level in that there is limited scatter in the three points for a particular forcing level. The next observation is of the similarity in the overall trend of the results, with both the random and chirp results exhibiting a 'U' shape. This is explained quite easily and, in many ways, is the justification for conducting the linear SI. At low levels of input power, the effect of the constant level of output noise has a large bearing on the error value. At high levels of input power, the increased error is due to the nonlinear effect becoming more pronounced and a linear system being incapable of reproducing the behaviour. Coincidentally, both the random input and chirp input produce the same minimum value of NMSE of 1.6% although this occurs at



(a)

(b)

Fig. 7. Examples of forcing input signals. (a) Shows broadband random input with variance of 10,000 N^2 and (b) shows linear chirp input of magnitude 100 N with frequency varying from 2 Hz to 75 Hz. Both inputs were sampled at one frequency of 15 kHz and signal duration was 2 s.

Table 1
Lower and upper bounds for the linear model parameters for use within the JADE optimisation.

Parameter name	Lower bound	Upper bound
m_L	0.2 kg	20 kg
c_L	1 N/(m/s)	100 N/(m/s)
k_L	1×10^4 N/m	1×10^6 N/m

an input power of 1000 N^2 for the random input and 50 N^2 for the chirp input. It may also be noted that the chirp input results in greater NMSE values than the random input for equivalent power. This information will be revisited in the nonlinear SI section.

Fig. 9 shows the results of the parameter estimation for each of the three JADE runs for each of the forcing levels for the random input. At low forcing levels, the noise effect results in inconsistent predictive capability. At higher forcing levels, the nonlinear effect is absorbed into the linear parameter estimates; in particular, the viscous damping term attempts to absorb the hysteretic effect. At the highest power levels, the viscous damping term reaches its upper bound constraint. Fig. 10 shows the parameter estimates for the three JADE runs for each of the forcing levels for the chirp input. The results show similar behaviour to the random input, but the linear SI loses consistent predictive capability at a lower input power.

The linear system identification has served several purposes. It has shown the levels of input power required for the random and chirp input to result in responses containing a significant nonlinear component, and it also provides the

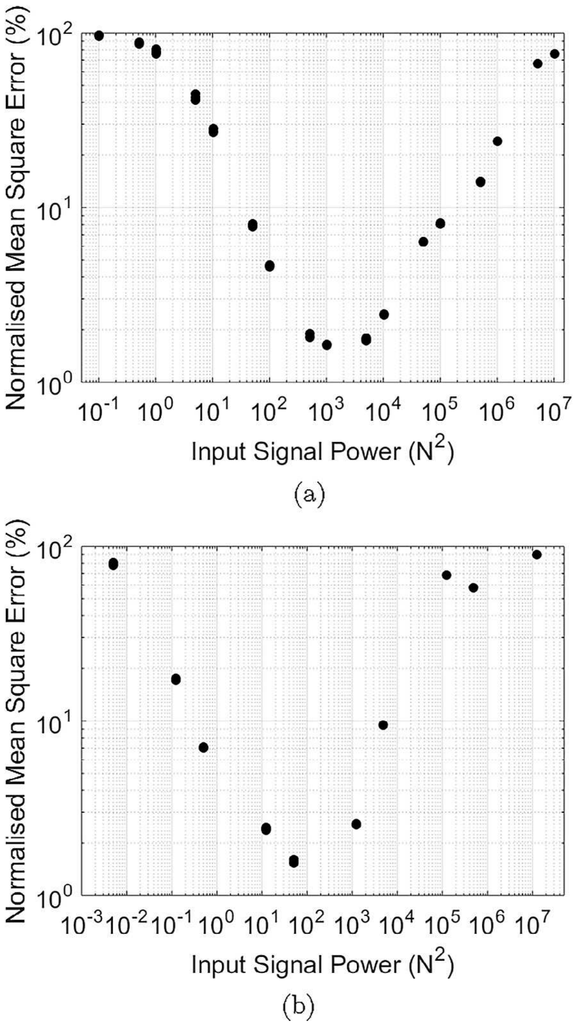


Fig. 8. Plots of NMSE vs. Input Power for linear SI. (a) Shows results of linear identification results using broadband random input and (b) shows linear identification results using chirp input.

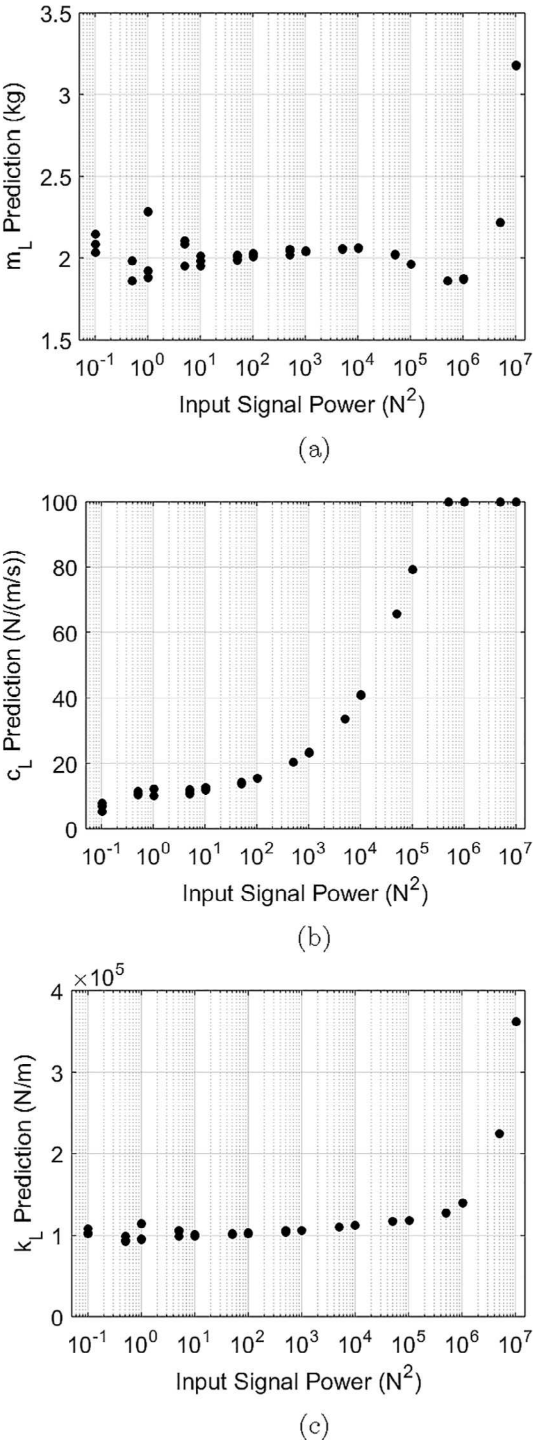
opportunity to reduce the linear parameter range for the nonlinear SI. After the SI, it was decided to alter the permissible mass range to be between 1.5 kg and 2.5 kg and the permissible viscous damping parameter to be between 5 N/(m/s) and 30 N/(m/s). It was also decided that the linear stiffness parameter range should not be reduced from the bounds given in Table 1 due to the linear stiffness effect being divided across the two terms (K_L and α) in the BW system.

3.4. Results of nonlinear system identification

Now that the linear SI has been conducted, the nonlinear SI process may commence, informed by the previous results. The identification followed a similar pattern to the linear identification process. As before, broadband random and chirp inputs were investigated over a range of different forcing levels; however, on this occasion, some of the lower levels of excitation were not investigated. In order to eliminate redundancy in the BW model, the β term was combined with the γ and δ terms. Furthermore, after some preliminary tests, it was decided that the υ term could be fixed to a value of 1.[4] For each forcing level, three runs of the JADE optimisation algorithm were conducted. Because of the greater complexity of the nonlinear identification problem, the population size was set to 70 and the optimisation was run for 200 generations. (A rule of thumb for evolutionary optimisation is that the number of individuals should be set to 5–10 times the number of parameters.) The lower and upper bounds for the parameters of interest are shown in Table 2.

The lowest FoM and NMSE were stored along with the corresponding nonlinear parameter estimates after each run.

[4] In general, this parameter should be determined in a principled manner. It can be included as parameter for estimation, or it can be regarded as a hyperparameter and decided by cross-validation. However, in this particular analysis, preliminary analysis showed that unity was the only credible choice.



(a)

(b)

(c)

Fig. 9. Plots of Parameter Estimates vs. Input Power for linear SI for broadband random input: (a) mass, (b) viscous damping, and (c) linear stiffness.

3.5. Broadband random training input

Fig. 11 shows the NMSE values for each of the JADE runs for the various forcing levels of the random input, superimposed onto the equivalent results from the previously presented linear SI results. The plot clearly shows the forcing levels at which the nonlinear identification provides added value. The improvement only really occurs above an input power of 1000 N^2 (the

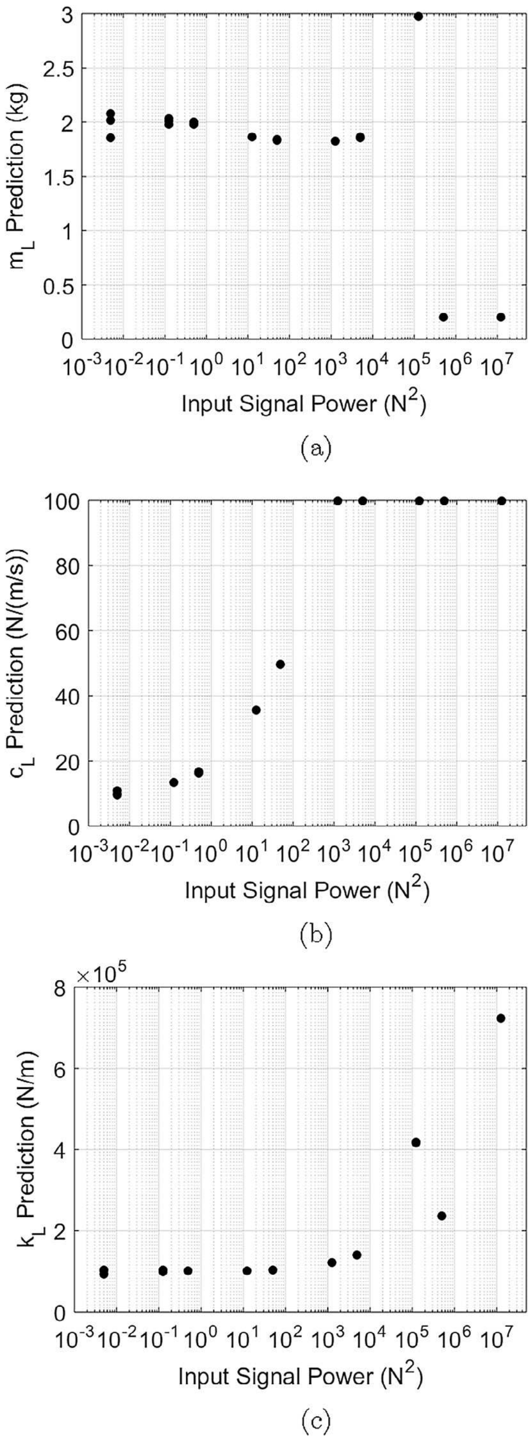


Fig. 10. Plots of Parameter Estimates vs. Input Power for linear SI for chirp input: (a) mass, (b) viscous damping, and (c) linear stiffness.

point at which the linear SI gave the lowest error). As the input power increases, so the NMSE of the nonlinear identification decreases and the gap between the linear and nonlinear SI widens. The lowest NMSE was $7.3 \times 10^{-3}\%$ for one of the runs with an input power of 5×10^6 N^2.

Fig. 12 shows the parameter estimates for each of the runs for the broadband input. Most immediately noticeable, is the significant amount of scatter and inconsistency in the parameter estimates at all but the higher forcing levels; this is likely to

Table 2
Lower and upper bounds for the nonlinear model parameters for use within the JADE optimisation.

Parameter name	Lower bound	Upper bound
m_L	1.5 kg	2.5 kg
c_L	5 N/(m/s)	30 N/(m/s)
k_L	1×10^4 N/m	1×10^6 N/m
α	1×10^4 N/m	1×10^6 N/m
$\beta\gamma$	-2×10^3/m	2×10^3/m
$\beta\delta$	-2×10^3/m	2×10^3/m

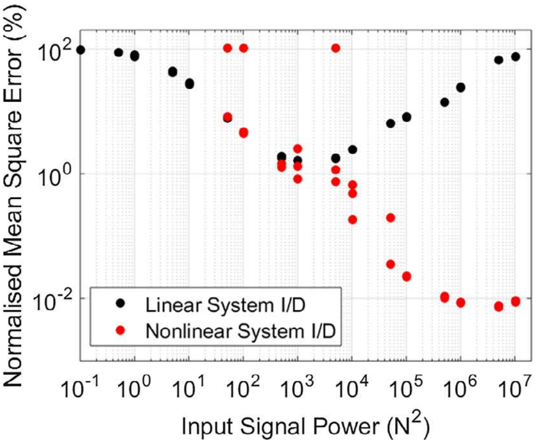


Fig. 11. Plot of Normalised Mean Square Error vs. Input Power for linear and nonlinear SI for broadband random input.

be a result of insensitivity to the nonlinear terms at low forcing. There is not only scatter between the three runs at a particular forcing level, but also a lack of consistency between the estimates at different forcing levels. A lack of variability between the three JADE runs and consistency between forcing levels is only observable for the highest five input powers (1×10^5 N^2 and greater). The only parameter which still lacks some consistency, even at the higher forcing levels is c_L, the linear damping parameter, whose estimates generally increase with increasing input power.

3.6. Chirp training input

Fig. 13 shows the NMSE values for each of the JADE runs for the various forcing levels of the chirp input, superimposed on the equivalent results from the previously-presented linear SI results. The added value provided by the nonlinear identification over linear SI is significantly more pronounced than for the random input and commences at a lower level of input power. At the highest power level, there is an increase in NMSE values and inconsistency between runs – this may be due to a mismatch between the Runge-Kutta (used for model prediction) and Newmark (used for generating training data) numerical integration algorithms. The lowest NMSE obtained was 3.9×10^{-4}% for one of the runs with an input power of 5×10^5 N^2 – this is around 1/20th of the lowest NMSE using random inputs.

Fig. 14 shows the parameter estimates for each of the JADE runs for the chirp input. On this occasion, there is some scatter between runs and inconsistency between forcing levels for the first three or four forcing levels; after that (i.e. when the nonlinearity has significant effect on the response), the estimates become more consistent. At the highest level of forcing, the scatter between the three runs returns. The consistency of these parameter estimates, when viewed in combination with the NMSE plots of Fig. 13, give a large degree of confidence that, by returning a nonlinear model NMSE that shows vast improvement over the corresponding linear model NMSE, the true system has been identified. This should be contrasted with the situation where a nonlinear SI (with no prior linear SI baseline) is conducted using a relatively low level of input forcing and a low value for the NMSE is incorrectly interpreted as meaning the true system has been identified.

Table 3 shows parameter estimates for the chirp input run that resulted in the lowest NMSE of 3.9×10^{-4}%. Two of the three JADE optimisation runs for the 1000 N (equivalent to an Input Power of 5×10^5 N^2) amplitude chirp forcing input returned exactly the same parameter values, whilst the other run returned values extremely close to those shown. Fig. 15 shows the predicted displacement response for the estimate that resulted in the lowest NMSE. As would be expected from

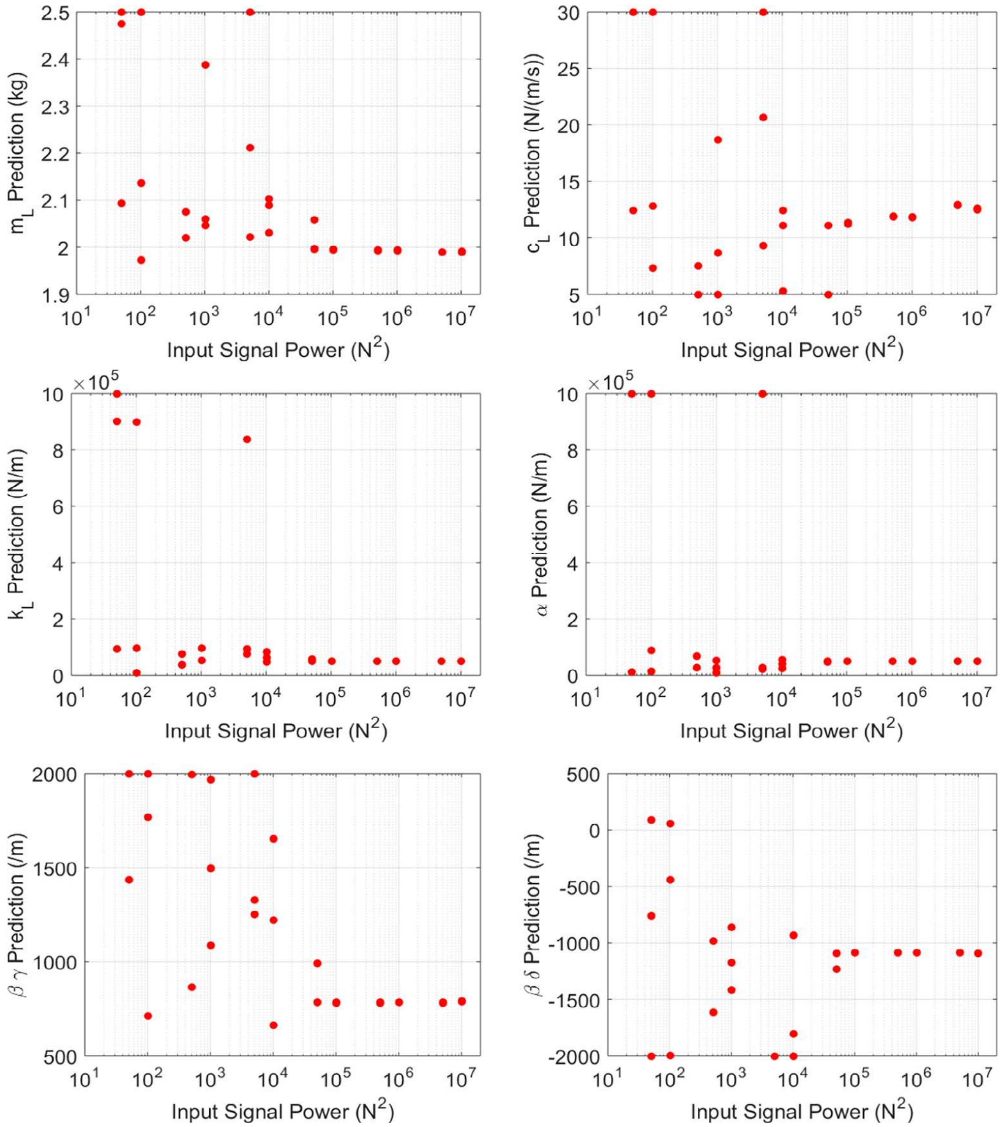


Fig. 12. Plots of Parameter Estimates vs. Input Signal Power for nonlinear SI for broadband random input: Top left is the mass estimate, top right shows viscous damping prediction, left middle plot shows linear stiffness prediction, right middle plot shows α, bottom left shows $\beta\gamma$ prediction and bottom right plot shows $\beta\delta$ prediction.

such a low NMSE value, the actual and predicted responses are indistinguishable. Of greater interest is the high degree of nonlinear behaviour present in the responses, especially from between around 0.4 s to 1.1 s; it is easy to understand why linear SI would struggle to return anything other than a high NMSE value.

3.7. Model test and figure-of-merit

Once the BW system parameters had been identified using the process detailed in the previous two sections, the predicted system could then be tested against the two fixed test datasets, namely the random phase multi-sine and the sine-sweep signal. The test datasets were sampled at 750 Hz and the data that were generated using the estimated system parameters were sampled at 15 kHz then downsampled by a factor of 20 to also give a 750 Hz sampling frequency.

3.7.1. Random phase multi-sine testing set

Fig. 16 shows the actual and predicted displacement response plots for the random phase multi-sine test dataset. At first glance, it may seem that, on this timescale, there is a near-identical match. Closer examination reveals that there is a slight

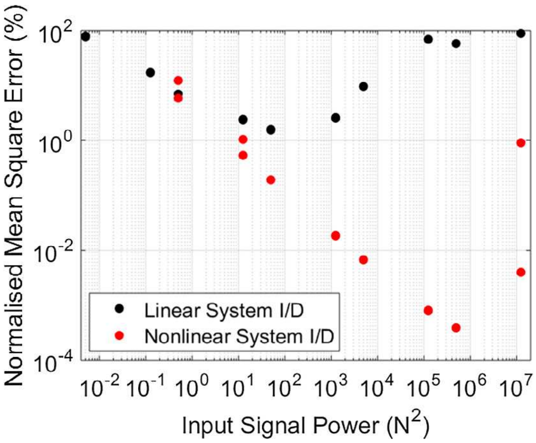


Fig. 13. Plot of Normalised Mean Square Error vs. Input Power for linear and nonlinear SI for chirp input.

mismatch of responses at the start of the signals. Fig. 17 shows a zoomed version of Fig. 16 for the first second of time, and the mismatch is a little clearer. The mismatch is likely to be a result of issues relating to the transient response. It would be possible to improve the prediction by fine-tuning the initial conditions on this test dataset, whilst retaining the previously identified model parameters. Even with the slight initial mismatch, the estimated model returned a very creditable FoM of 4.75×10^{-5} m; this corresponds to an NMSE value of 0.51%.

3.7.2. Sine-sweep testing set

Fig. 18 shows the actual and predicted displacement response plots for the sine-sweep test dataset. As with the random phase multi-sine test signal, there initially appears to be a near-identical match. Closer examination reveals that there is a slight mismatch of responses just after they peak at around 145 s. Fig. 19 shows a zoomed version of Fig. 18 for the time period between 145 s and 160 s. The FoM for the sine-sweep is 1.02×10^{-5} m, which corresponded to a very impressive NMSE value of 0.024%. This success may, in some part, be due to the use of a chirp input for the identification of the system parameters. In fact, the EO approach here gave the lowest cost solution for Benchmark One at the time of the workshop.

3.8. Discussion

The work on the first benchmark, presented in this part of the paper, focussed on highlighting the role played by the force input in the nonlinear SI process and the need for performing a linear system identification before undertaking its nonlinear counterpart. By conducting SI, the added value of the nonlinear identification can be demonstrated and nonlinear behaviour can be highlighted.

It is hoped that the work has demonstrated that, whilst random inputs are entirely appropriate for linear SI, they should be used with caution for nonlinear identification due to their tendency to linearise. It was shown here, that chirp inputs of corresponding power resulted in significant reduction in NMSE along with reduced variability in identified parameters.

The results from the two fixed test datasets found that the system was identified with 0.51% NMSE for the random phase multi-sine data and 0.024% MSE; the corresponding FoMs were 4.75×10^{-5} m and 1.02×10^{-5} m. These results were obtained using JADE optimisation in combination with a high-amplitude linear chirp input excitation.

Note that the results presented here do not include confidence intervals for the parameter estimates. As discussed earlier in the paper, this is not a weakness of the evolutionary approach, as witnessed by the study in [20] which specifically deals with a Bouc-Wen system. The confidence intervals were not computed here because the objective of the benchmarking exercise was simply to obtain the FoM for the various approaches and report those.

Finally, if the results of this section had been presented as a scientific study in other circumstances, they would be considered lacking in a very important respect, i.e. the method and the results are not compared with any competing methods and the approach selected is not evaluated on other datasets. These considerations are vital in a machine learning study in almost all contexts except the one here; the need for local comparison is obviated by the fact that the overall workshop aim was to provide comparisons between the approaches of different participants; the consideration of other datasets is clearly not necessary given the specific focus on the VUB benchmarks.

As discussed earlier, although not covered in any detail here, the discipline of Bayesian inference also offers the possibility of a general framework for NLSI. As well as general references given in the previous section, the reader may wish to consult

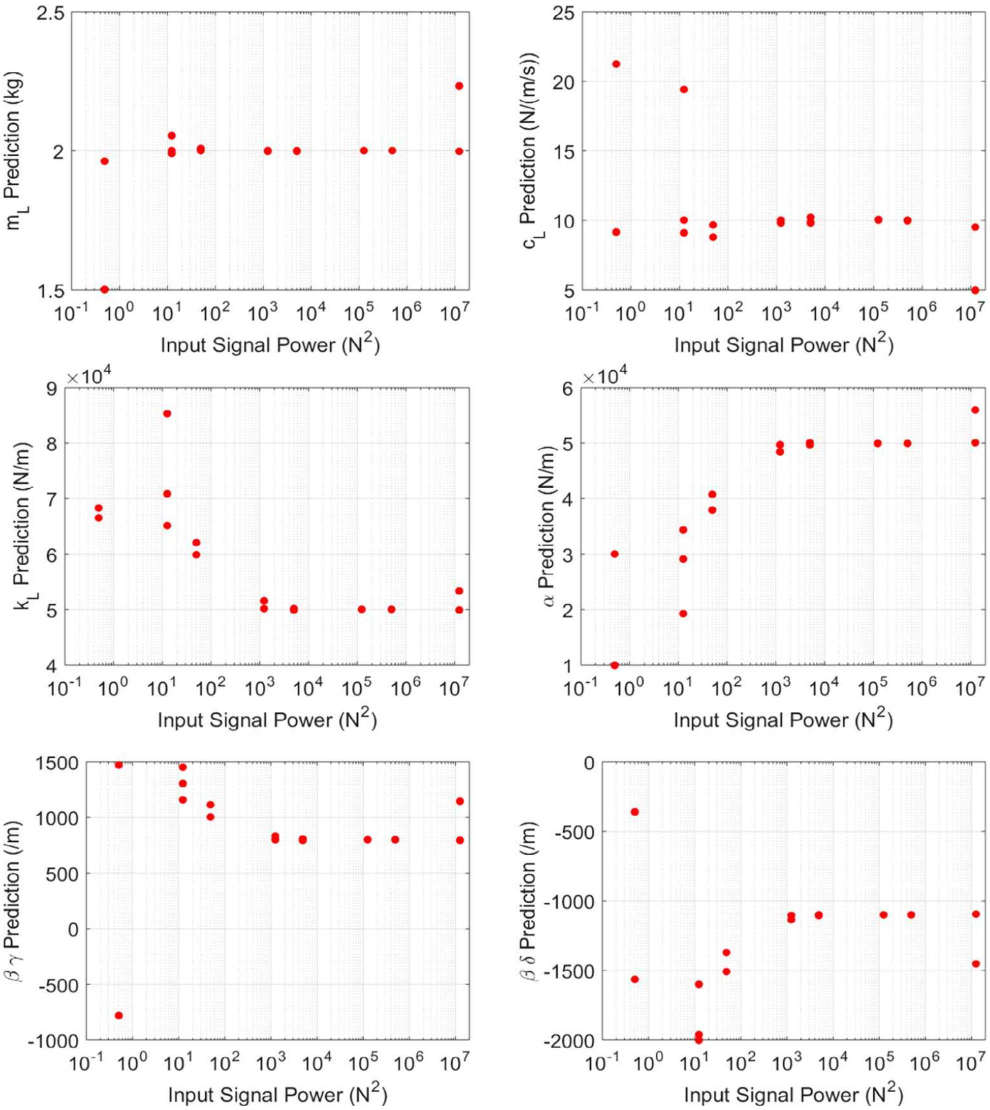


Fig. 14. Plots of Parameter Estimates vs. Input Signal Power for nonlinear SI for chirp input: Top left is the mass estimate, top right shows viscous damping prediction, left middle plot shows linear stiffness prediction, right middle plot shows α, bottom left shows $\beta\gamma$ prediction and bottom right plot shows $\beta\delta$ prediction.

Table 3
Nonlinear parameters giving the lowest NMSE from the JADE optimisation using a 1000 N chirp input.

Parameter name	Predicted value
m_L	2.009 kg
c_L	9.9698 N/(m/s)
k_L	4.9984 × 104 N/m
α	4.9985 × 104 N/m
$\beta\gamma$	799.6/m
$\beta\delta$	−1099.9/m
υ	1

the recent [44,40,41], which illustrate the use of *Approximate Bayesian Computation*, which can simultaneously estimate parameters and weigh the evidence for competing model forms. Two of the papers also focus on a BW system (including an experimental dataset), so can form a useful basis for comparison with the current software.

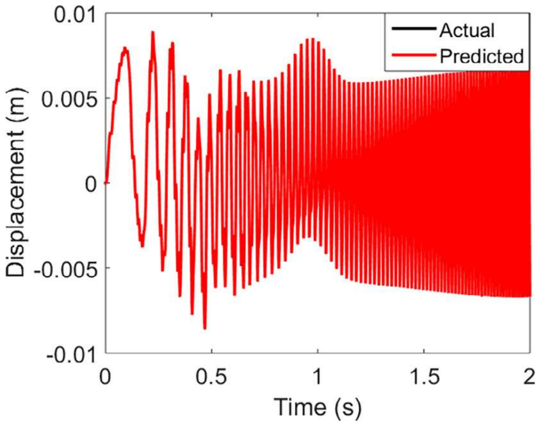


Fig. 15. Actual displacement response plotted with best model predicted displacement response for a 1000 N chirp forcing input.

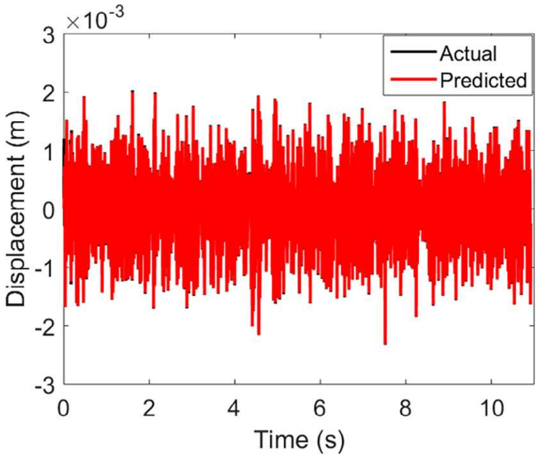


Fig. 16. Actual displacement response plotted with best model predicted displacement response for random phase multi-sine test dataset.

4. Benchmark two: a Wiener-Hammerstein system with process noise

4.1. Introduction

This benchmark presents a Wiener-Hammerstein (WH) electronic circuit where the main challenge results from high process noise, which is the dominant noise distortion [45]. The problem is to identify a Wiener-Hammerstein model, which is a block-structured system, as shown in Fig. 20, comprising two linear time-invariant (LTI) blocks in series, separated by a static nonlinear function.

There are precursors for the uses of EO approaches in WH identification. The approach proposed here is very different from the one reported in [46], where the evolutionary optimisation is used only for the pole-zero allocation problem reported in [47]. The approaches that are presented in [48,49] are more similar to the method presented in this paper. However, [48] only considers the problem where the LTI blocks are represented by a FIR model, and a simplified differential evolution algorithm is used. The method presented in [49] uses a biosocial culture algorithm. It is comparable to the approach presented here, although it requires more hyperparameters to be selected by the user.

In brief, the benchmark data have been generated from an electronic circuit. The first LTI block can be described well with a third-order lowpass filter. The second LTI subsystem is designed as an inverse Chebyshev filter which has a transmission zero within the excited frequency range, making the inversion of the filter difficult. The static nonlinearity $f(x)$ has been realised with a diode-resistor network, resulting in a saturation nonlinearity. The problem is difficult because the high level of process noise e_x will potentially bias the parameters.

The LTI blocks are represented in notation by $R(Z^{-1}, \underline{w}_1)$ and $S(Z^{-1}, \underline{w}_2)$, where Z^{-1} represents the backward shift operator or Z-transform variable and $\underline{w}_1$ and $\underline{w}_2$ are the model parameters for the blocks.

For the identification approach chosen here, each linear block will be represented by an ARX model of the form,

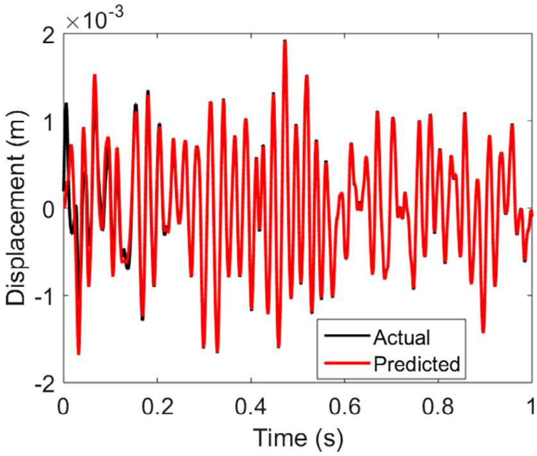


Fig. 17. Zoomed (first second) actual displacement response plotted with best model predicted displacement response for random phase multi-sine test dataset.

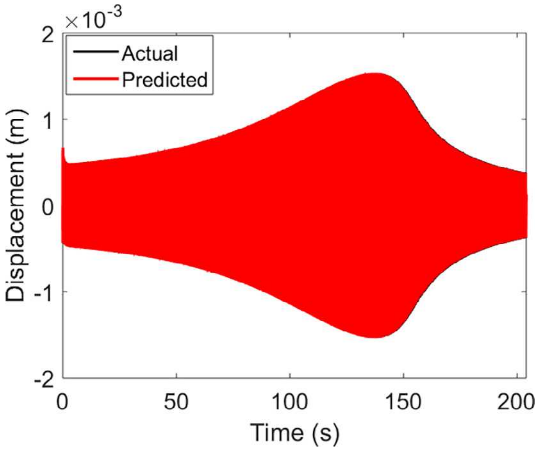


Fig. 18. Actual displacement response plotted with best model predicted displacement response for sine-sweep test dataset.

$$y_i = \sum_{j=1}^{n_y} a_j y_{i-j} + \sum_{j=1}^{n_x} b_j x_{i-j} + e_i \tag{24}$$

and the problem will be to estimate the parameters $\underline{w}_i = (\underline{a}_i, \underline{b}_i)$ for each block. The nonlinear function will be represented here by a sum over sigmoids,

$$f(z) = c_0 + \sum_{i=1}^{nc} c_{1i} \tanh(c_{2i} z + c_{3i}) \tag{25}$$

with an option to use a polynomial representation included in the code.

One of the issues with identifying WH models is that the representation is not unique. For example, the system representation is invariant under an exact two-parameter group of transformations,

$$R \to K_1 R$$

$$S \to K_2 S$$

$$f(z) \to \frac{1}{K_2} f\left(\frac{1}{K_1} z\right) \tag{26}$$

and this presents a type of 'gauge' that can (if so desired) be used to fix scales by setting $b_1 = 1$ in both of the relevant ARX models. Another source of non-uniqueness is generated by the fact that a delay τ on one of the linear blocks, can be compensated by a lead $-\tau$ in the other [50].

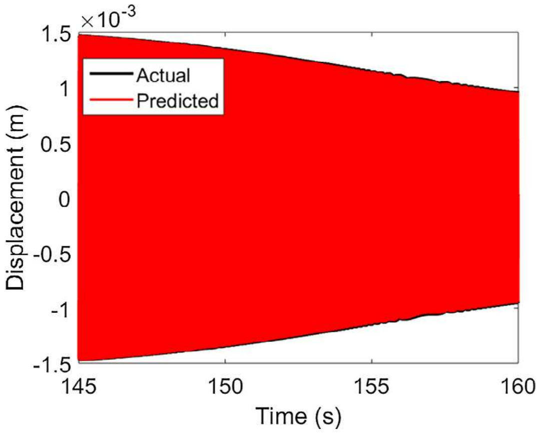


Fig. 19. Zoomed (around peak) actual displacement response plotted with best model predicted displacement response for sine-sweep test dataset.

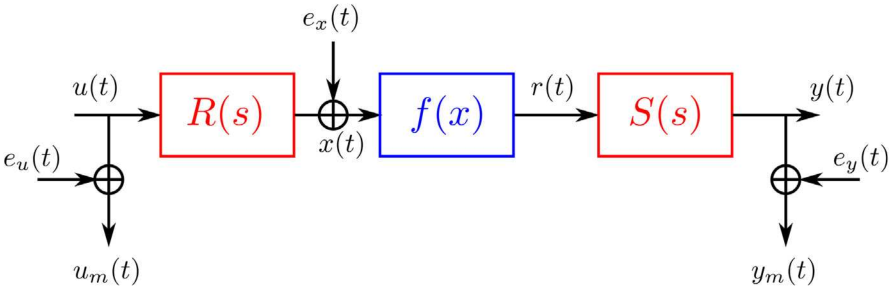


Fig. 20. Schematic of a Wiener-Hammerstein System.

The identification problem is thus to estimate the parameters of the LTI blocks and those characterising the static nonlinearity. As for Benchmark One, the problem will be framed in terms of optimisation and an evolutionary scheme is chosen. Once again, the algorithm is an extension of the basic DE algorithm – in this case, SADE (*Self-Adaptive Differential Evolution*) is used. The algorithm has been chosen because, as a population-based approach, it is resistant to getting captured in local minima, and also because it is versatile in terms of the cost function used for minimisation. However, in the study presented here, the cost function is the ordinary least-squares error. In order to make the paper self-contained, the details of the SADE extension to DE are briefly presented.

4.2. Self-Adaptive Differential Evolution (SADE)

As stated in the last section when discussing JADE, a potential weakness of the standard DE algorithm is that it requires the prior specification of a number of *hyperparameters*. Apart from the population size, maximum number of iterations, etc., the algorithm needs *a priori* specification of the scaling factor F and crossover probability C_r. An algorithm which establishes 'optimum' values for the hyperparameters during the course of the evolution is clearly desirable. Such an algorithm is available in the form of the Self-Adaptive Differential Evolution (SADE) algorithm [51,52]; the description and implementation of the algorithm here largely follows [52].

The development of the SADE algorithm begins with the observation that Storn and Price, the originators of DE, arrived at five possible strategies for the mutation operation [19]:

1. *rand1*: $M = A + F(B - C)$
2. *best1*: $M = X^* + F(B - C)$
3. *current-to-best*: $M = T + F(X^* - T) + F(B - C)$
4. *best2*: $M = X^* + F(A - B) + F(C - D)$
5. *rand2*: $M = A + F(B - C) + F(D - E)$

where T is the current trial vector, X^* is the vector with (currently) best cost and (A, B, C, D, E) are randomly-chosen vectors in the population distinct from T. F is a standard (positive) scaling factor. The SADE algorithm also uses multiple variants of the mutation algorithm as above; however these are restricted to the following four:

1. *rand1*
2. *current-to-best2*: $M = T + F(X^* - T) + F(A - B) + F(C - D)$
3. *rand2*
4. *current-to-rand*: $M = T + K(A - T) + F(B - C)$

In the strategy *current-to-rand*, K is defined as a coefficient of combination and would generally be assumed in the range $[-0.5, 1.5]$; however, in the implementation of [52] and the one used here, the prescription $K = F$ is used to essentially restrict the number of tunable parameters. The SADE algorithm here uses the standard crossover approach, except that at least one crossover is forced in each operation on the vectors. If mutation moves a parameter outside its allowed (predefined) bounds, it is pinned to the boundary. Selection is performed exactly as in DE; if the trial vector has smaller (or equal) cost to the target, it replaces the target in the next generation.

The adaption strategy must now be defined. First, a set of probabilities are defined: $\{p_1, p_2, p_3, p_4\}$, which are the probabilities that a given mutation strategy will be used in forming a trial vector. These probabilities are initialised to be all equal to 0.25. When a trial vector is formed during SADE, a roulette wheel selection is used to choose the mutation strategy on the basis of the probabilities (initially, all equal). At the end of a given generation, the numbers of trial vectors successfully surviving to the next generation from each strategy are recorded as: $\{s_1, s_2, s_3, s_4\}$; the numbers of trial vectors from each strategy which are discarded are recorded as: $\{d_1, d_2, d_3, d_4\}$. At the beginning of a SADE run, the survival and discard numbers are established over the first generations, this interval is called the *learning period* (and is another example of a hyperparameter). At the end of the learning period, the strategy probabilities are updated by,

$$p_i = \frac{s_i}{s_i + d_i} \tag{27}$$

After the learning period, the probabilities are updated every generation but using survival and discard numbers established over a moving window of the last N_L generations. The algorithm thus adapts the preferred mutation strategies. SADE also incorporates adaption or variation on the hyperparameters F and C_r. The scaling factor F mediates the convergence speed of the algorithm, with large values being appropriate to global search early in a run and small values being consistent with local search later in the run. The implementation of SADE used here largely follows [51] and differs only in one major aspect, concerning the adaption of F. Adaption of the parameter C_r is based on accumulated experience of the successful values for the parameter over the run. It is assumed that the crossover probability for a trial is normally distributed about a mean $\overline{C}_r$ with standard deviation 0.1. At initiation, the parameter C_r is set to 0.5 to give equal likelihood of each parent contributing a chromosome. The crossover probabilities are then held fixed for each population index for a certain number of generations and then resampled. In a rather similar manner to the adaption of the strategy probabilities, the C_r values for trial vectors successfully passing to the next generation are recorded over a certain greater number of generations and their mean value is adopted as the next $\overline{C}_r$. The record of successful trials is cleared at this point in order to avoid long-term memory effects. The version of the algorithm here adapts F in essentially the same manner as C_r but uses the Gaussian $N(0.5, 0.3)$ for the initial distribution. At this point, the reader might legitimately argue that SADE has simply replaced one set of hyperparameters (F, C_r) with another (duration of the learning period, etc.). In fact, because DE and SADE are heuristic algorithms, there is no analytical counter to this argument. However, the transition to SADE is justified by the fact that the algorithm appears to be very robust with respect to the new hyperparameters.

A benchmark of the SADE algorithm against the standard DE on a Bouc-Wen identification can be found in [15]; it is shown there that SADE offers a very large speedup in terms of convergence, very similar to the JADE algorithm used for Benchmark One.

4.3. Algorithm verification

Unlike the other two VUB benchmarks, the current authors had not attempted to identify block-structured systems before, and thus considered that the developed algorithm code required verification on a known problem. The system chosen was taken from [53] and comprised the two LTI blocks,

$$R(Z^{-1}, \underline{\theta}_1) = \frac{0.216Z^{-1}}{1 - 1.579Z^{-1} + 0.67Z^{-2}}$$

$$S(Z^{-1}, \underline{\theta}_2) = \frac{5.0Z^{-1}}{1 - 0.875Z^{-1}} \tag{28}$$

and static nonlinearity,

$$f(z) = 5z + 20z^2 + 50z^3 \tag{29}$$

Data were generated using a white noise input. In order to fit the model, the correct number of lags n_x and n_y were specified and ten SADE runs were carried out with a population of 90 individuals; 1000 generations were used. The exercise was rather time-consuming and took of the order of three hours; however, the best solution reached an NMSE of 1.12×10^{-28} and arrived at the model,

$$\hat{R}(Z^{-1}, \underline{\theta}_1) = \frac{0.3468Z^{-1}}{1 - 1.579(10)Z^{-1} + 0.67(11)Z^{-2}}$$

$$\hat{S}(Z^{-1}, \underline{\theta}_2) = \frac{1.031Z^{-1}}{1 - 0.875(10)Z^{-1}}$$

$$\hat{f}(z) = 15.177z + 37.666z^2 + 58.655z^3 \tag{30}$$

No parameters were fixed, and the resulting non-uniqueness of the model is clearly visible in the numerator parameters. The denominator parameters were estimated very accurately (the bracketed quantities after the parameters indicate the number of following zeroes). Multiplying up the numerator parameters and the linear term in the nonlinear function showed that the 'linear gain' corresponds perfectly with the original system. Fig. 21 shows the evolution of the SADE cost function across the ten runs; in five of the runs, SADE converged to the 'global' minimum, in one of the runs it did not quite converge, and in four of the runs it arrived at a local minimum.

4.4. Benchmark identification

Having established that the SADE algorithm worked on WH identification problems, attention moved to the actual VUB benchmark. As part of the benchmark exercise, participants were invited to design input signals for the identification, which could then be run through the benchmark circuit in order to generate training data. The current authors did not design an input, but decided to simply choose an appropriate dataset from among those created. The data set chosen was that stored as '*WH_CombinedZeroMultisineSinesweep.mat*' which used a period of a random multisine followed by a swept-sine excitation. The input from the whole dataset is shown in Fig. 22.

The total dataset contained 57,346 points of input and output. A subset comprising the first 10,000 points of each record was chosen for the training data, which meant that the excitation was only the random multisine. The data were sampled at 78,125 Hz; Fig. 23 shows the frequency domain information pertaining to the training data selected, up to the Nyquist frequency.

Because of the amount of time needed to run SADE on a 10,000 point training set, it was impractical to determine the orders of the ARX models in the linear blocks via cross-validation; for this reason input and output orders of 6 lags were used as it was anticipated that this should be adequate to accurately capture the behaviour of the third-order filters in the circuit. As discussed with respect to Benchmark One, it is good practice to establish a baseline for the identification by fitting a linear model. Of course, the linear modelling did not require SADE, an ordinary least-squares approach was used. So that the linear model captured the same temporal range as the nonlinear model, 12 input and output lags were used.[5]

In order to judge the results of the Benchmark Two identification, two noise-free test datasets were provided: a random multi-sine and a chirp; the idea was to report the FoM on the two test sets. In the case of the (12, 12) linear ARX fit, the FoM values on the two test sets were found to be 0.057 and 0.022 (*simulation* results: multi-sine and chirp, respectively). These results proved a little inferior to results from a *Best Linear Estimate* (BLE) model (see [54], for an explanation of the concept), fitted by other participants in the workshop; Fig. 24 shows comparisons between the true and predicted results on the two test sets.

The SADE algorithm was then used to identify the system. As discussed above, ARX(6, 6) models were used for the linear blocks and four sigmoid functions were used in order to estimate the static nonlinearity (sigmoids were chosen as the static nonlinearity was known to have a saturation characteristic). Altogether, this gave a model with 37 tunable parameters. Because SADE is a nonlinear optimisation algorithm, it can be sensitive to the initial ranges chosen for the parameters; because of this, a number of runs were carried out in order to guide the choice of the initial ranges; these were chosen to be $[-1, 1]$ for all ARX parameters and $[-100, 100]$ for all sigmoid parameters. Because of the size of the problem, only one final SADE run was carried out; this used a population of 370 individuals and ran for 1000 generations; this took around 12 h. The converged model gave FoM values of 0.044 on the multi-sine test set and 0.020 on the chirp test set. The predictions on the test sets for the nonlinear SADE model are shown in Fig. 25.

The results for the full nonlinear model on the multi-sine test set are better than a naive linear least-squares, but not as good as the BLE; the results on the chirp test set are better than both the naive linear model and the BLE. However, the improvements over a linear model are not at all marked. This begs the question: does the SADE model actually make use of the static nonlinearity? One can answer this question by making a forward run through the full WH model and plotting the input to the static nonlinearity, z_1, against the output, z_2. The results of this exercise are shown in Fig. 26. It is clear that the SADE WH model is actually nonlinear; furthermore, the saturation characteristic is clearly captured.

[5] This was a point on which an element of carelessness was identified after the workshop. While time constraints before the workshop precluded the optimisation of the lag numbers on the full nonlinear model, guidance could have been obtained from the optimal lag numbers for a linear model, as the linear model estimation was very fast. Cross-validation analysis after the workshop gave optimal values of the lag numbers as (8, 8) for the linear model. In the light of this analysis, the choice of (6, 6) for the LTI linear blocks in the full nonlinear model does not seem inappropriate.

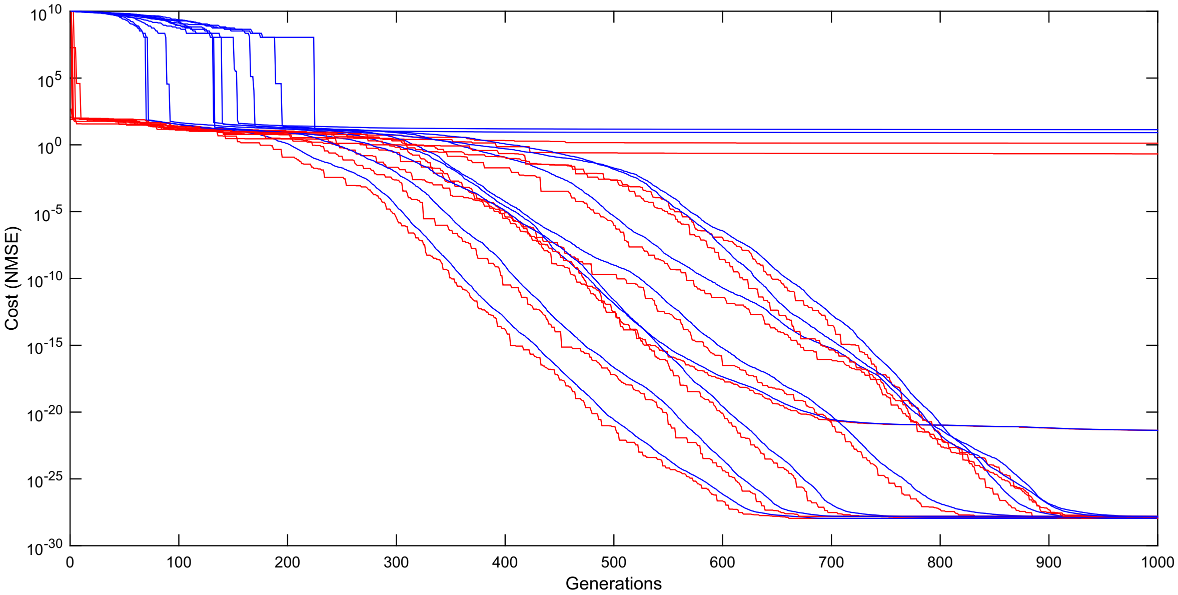


Fig. 21. SADE cost functions across the ten runs used for the verification problem (red – lowest cost per generation; blue average cost per generation). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

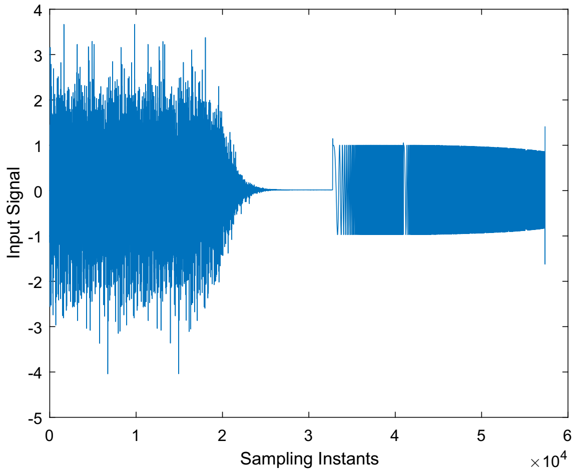
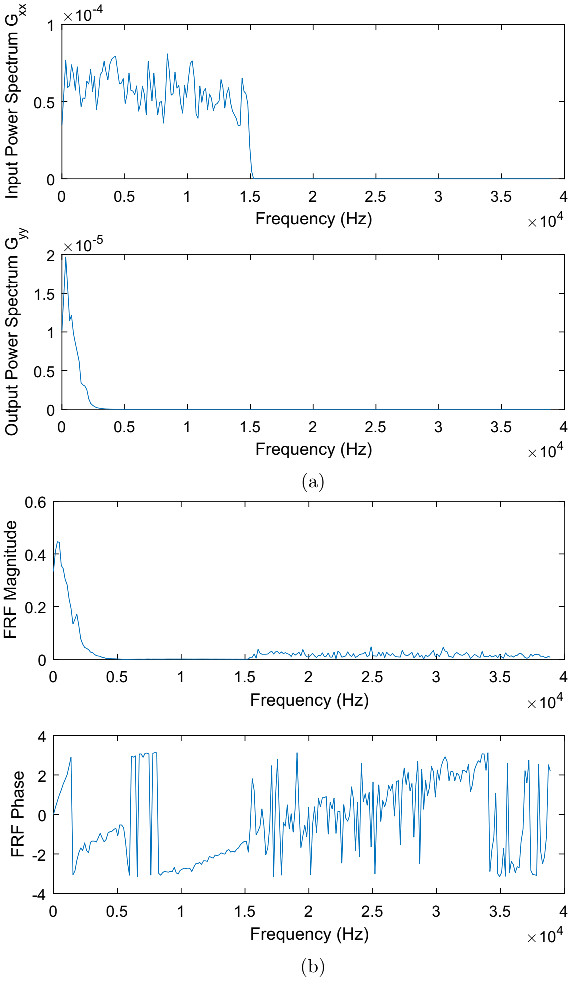


Fig. 22. Input data from Benchmark Two training data: *WH_CombinedZeroMultisineSinesweep.mat*.

4.5. Discussion

The results for Benchmark Two are provided here as a first attempt at identification of a block-structured system by the authors. Although the results are far from perfect, it is hoped that the proposed method is of interest. Given that the algorithm performed very well on a noise-free simulation, it is a fair assumption that the poorer results on the actual benchmark are because of the very high level of internal process noise added. There is room for improvement in several areas. One issue here is that the ARX model orders are hyperparameters and should have been determined in a principled manner; however, the very slow nature of the algorithm precluded that here. One of the reasons for the slow convergence was that the algorithm generated individuals for the initial population completely randomly. It was subsequently realised that this generated many unstable linear blocks, and the corresponding models were essentially wasted in terms of genetic material. As mentioned in the introduction, this paper has been written as a record of the authors' participation in the benchmark exercise, and only shows the results of the original submissions. However, a more rigorous variant of the algorithm was developed later and among the advances in that algorithm was a constraint that the initial population contained only stable models



(a)

(b)

Fig. 23. Frequency domain representation of Benchmark Two identification/training data: (a) input and output spectra; (b) FRF magnitude and phase.

[55]. When the improved SADE algorithm was applied to the 'Silverbox' benchmark (much less process noise than the VUB benchmark) [56], it achieved results consistent with the previous best attempts.

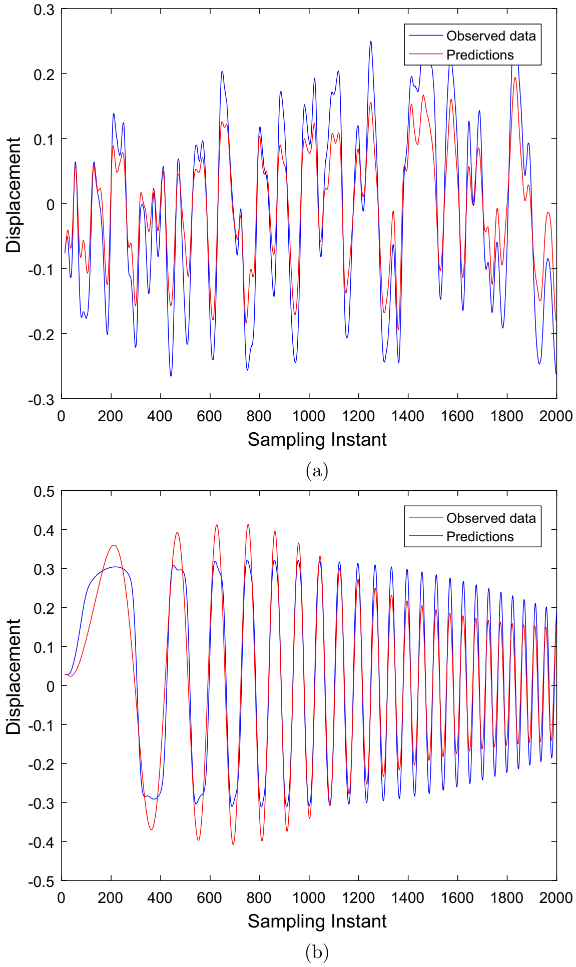
Because the optimisation method here only uses 'forward' runs through the WH model, the transmission zero in the second LTI block has no effect on the identification; this is a strength of the approach. One of the other possibilities for the SADE approach is to use more general objective functions without too much disruption to the core algorithm. For example it may be possible to implement a maximum likelihood variant [57,58].

5. Benchmark three: cascaded tanks

The third of the VUB benchmarks was based on the identification of a physical experiment involving the vertical flow of water between tanks [59]. The system was considered challenging because it contained an unmeasured state, but mainly because only a small training set (1024 points) was given.

A combination of physical modelling and machine learning techniques were employed here to address the cascaded tanks problem. This form of *grey-box* (see Introduction) model involved first fitting a physical white-box model. Then, the outputs of the white-box were used to provide additional feature-rich inputs to a black-box model – in this case a Gaussian Process (GP) regression model.

For the cascaded tanks system, the physics of the problem is understood, but not fully. Because of the detailed behaviour of the fluid – particularly under the overflow condition – it is not possible to fully capture the behaviour of the system in a physical model. However, the partial physics models which are available are good approximations to the true behaviour of the system. This makes this form of grey-box modelling a sensible choice, the white-box model can almost be thought of as a strong *prior* for the subsequent Bayesian method.



(a)

(b)

Fig. 24. Results of linear ARX(12, 12) fit to Benchmark Two test sets: (a) random multi-sine, (b) chirp.

5.1. White-box modelling

Two different white-box models were established to investigate the effect of a more sophisticated model on the predictive performance. When white-box models are formed from known physical behaviour they should perform well in extrapolation in addition to any interpolation capability. If large discrepancies are seen between training and testing errors, this could indicate that the physics of the system chosen fails to capture the true behaviour.

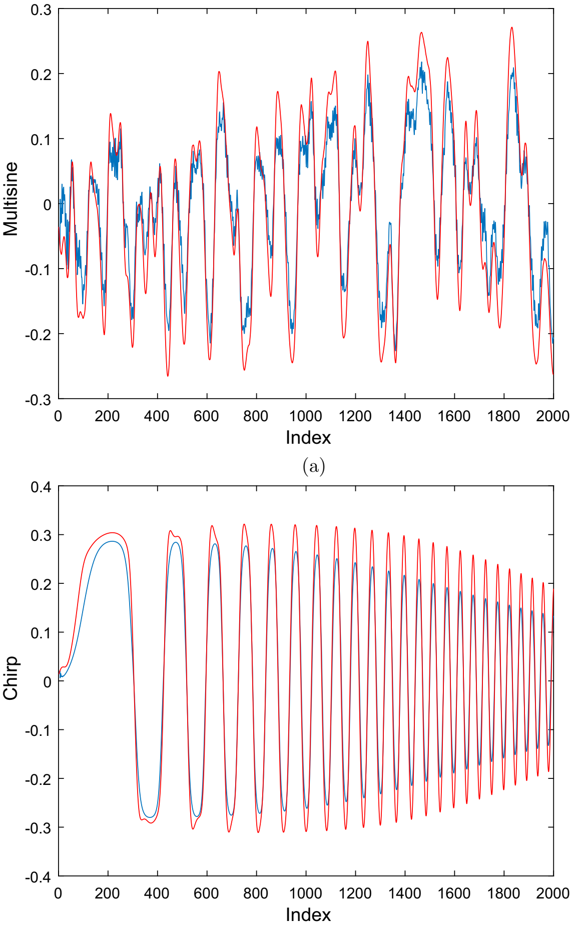
When the system is not in overflow, and assuming Bernoulli's principle holds, the state equations of the system can be written as,

$$\begin{aligned}
\dot{z}_1(t) &= -k_1\sqrt{z_1(t)} + k_4x(t) + w_1(t)\\
\dot{z}_2(t) &= k_2\sqrt{z_1(t)} - k_3\sqrt{z_2(t)} + w_2(t)\\
y(t) &= z_2(t) + e(t)
\end{aligned} \tag{31}$$

When the system is in overflow, the model can be described by the following equations,

$$\begin{aligned}
\dot{z}_1(t) &= -k_1\sqrt{z_1(t)} + k_4x(t) + w_1(t)\\
\dot{z}_2(t) &= \begin{cases} k_1\sqrt{z_1(t)} - k_3\sqrt{z_2(t)} + w_2(t), & z_1(t) \leqslant 10\\ k_1\sqrt{z_1(t)} - k_3\sqrt{z_2(t)} + k_5x(t) + w_3(t), & z_1(t) > 10 \end{cases}
\end{aligned} \tag{32}$$

In both these cases, $x(t)$ is the input to the system, k_n represents the nth parameter of the system and w_n and e are the noise terms. z_1 and z_2 are state variables representing the liquid levels in the two tanks; only z_2 is measured, giving the observable output $y; z_1$ is thus an unmeasured state. The noise terms are assumed zero in the white-box model, allowing the black-box to compensate for noise. It is important to form a white-box model without noise, as this can distort the residuals which the black box is trying to fit.



(a)

(b)

Fig. 25. Results of SADE nonlinear WH fit to Benchmark Two test sets: (a) random multi-sine, (b) chirp.

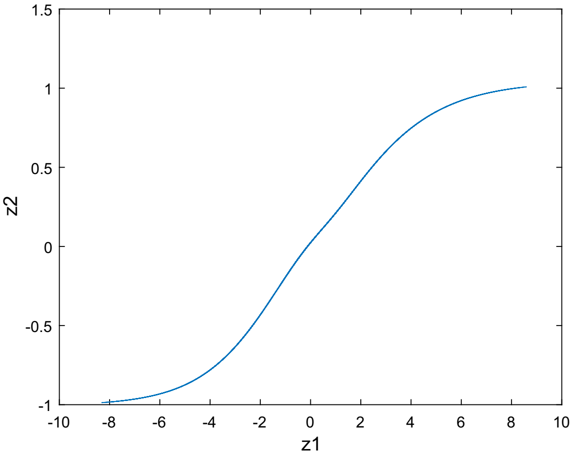


Fig. 26. Graph of input versus output for static nonlinearity in full WH model on chirp test set.

The equations in (32) define the first physical model to be fitted: Model 1. This model was extended to include losses in the system due to friction or geometry; this involved adding additional terms, which introduced two more parameters which required fitting. The extended physics model is shown below in Eq. (33) (from this point on, explicit time dependence in the variables will be omitted),

$$\begin{aligned}\dot{z}_1 &= -k_1\sqrt{z_1} + k_5 z_1 + k_4 x + w_1\\ \dot{z}_2 &= \begin{cases} k_1\sqrt{z_1} - k_5 z_1 + k_6 z_2 - k_3\sqrt{z_2} + w_2, & z_1 \leqslant 10\\ k_1\sqrt{z_1} - k_5 z_1 + k_6 z_2 - k_3\sqrt{z_2} + k_5 x + w_3, & z_1 > 10\end{cases}\end{aligned} \tag{33}$$

The equations shown in (33) define the extended physics model, referred to as Model 2. As in the other benchmarks considered here, the process of fitting the parameters of these models is cast as a multidimensional optimisation with respect to a cost function. The normalised mean-squared error (NMSE), (6), between the predictions and the true system outputs is once more used as the cost function.

Four different optimisation schemes were compared:

- Standard differential evolution (DE) [19] using the classic random binary crossover.
- Particle swarm optimisation (PSO) [60] using sigmoid decreasing inertia weights [61].
- Quantum-behaved particle swarm (QPSO) [62].
- Krill herd (KH) optimisation [63].

The latter two methods represent more recent approaches in optimisation which will be briefly described. The QPSO method is based on the same principles as the regular particle swarm but the dynamics of each particle is changed from the classic formulation to one where every particle is treated in a quantum manner. As DE has been described in a little detail, *Particle Swarm Optimisation* deserves a brief explanation as it is the basis of two of the methods here [60]. In many ways, PSO is one of the simplest iterative population-based optimisation algorithms. The PSO algorithm is motivated by the flocking behaviour of flights of birds in search of food. Each particle (bird) i in the population (flock) is represented by a vector of its position and velocity (y_i, v_i); the update rules for a generation are simply,

$$v_i = v_i + c_1 X_1 (y_i^* - y_i) + c_2 X_2 (y^* - y_i)$$

$$y_i = y_i + v_i \tag{34}$$

where y_i^* is the best position experienced so far by particle i, measured in terms of some cost function $J(y_i)$, and y^* is the best position of *any* particle. c_1 and c_2 are learning factors (hyperparameters) and X_1 and X_2 are randomly generated. The particles are thus steered according to their individual experience and that of the flock. The basic algorithm encoded in Eq. (34) belongs to a simpler class of biologically-inspired algorithms than the DE-inspired family, which are essentially motivated by real-coded genetic algorithms; the PSO variants are not dependent on operations like mutation and crossover. The PSO variants are thus interesting to consider alongside of the DE variants. In the ‘quantum’ version of the PSO algorithm, the trajectory of the particles is dependent upon, both, an attractor that is a random combination of the global best and previous best position for each particle, and a term relating to the particles’ potential fields. This change in the position update procedure arguably encourages better exploration properties for the QPSO over the PSO.

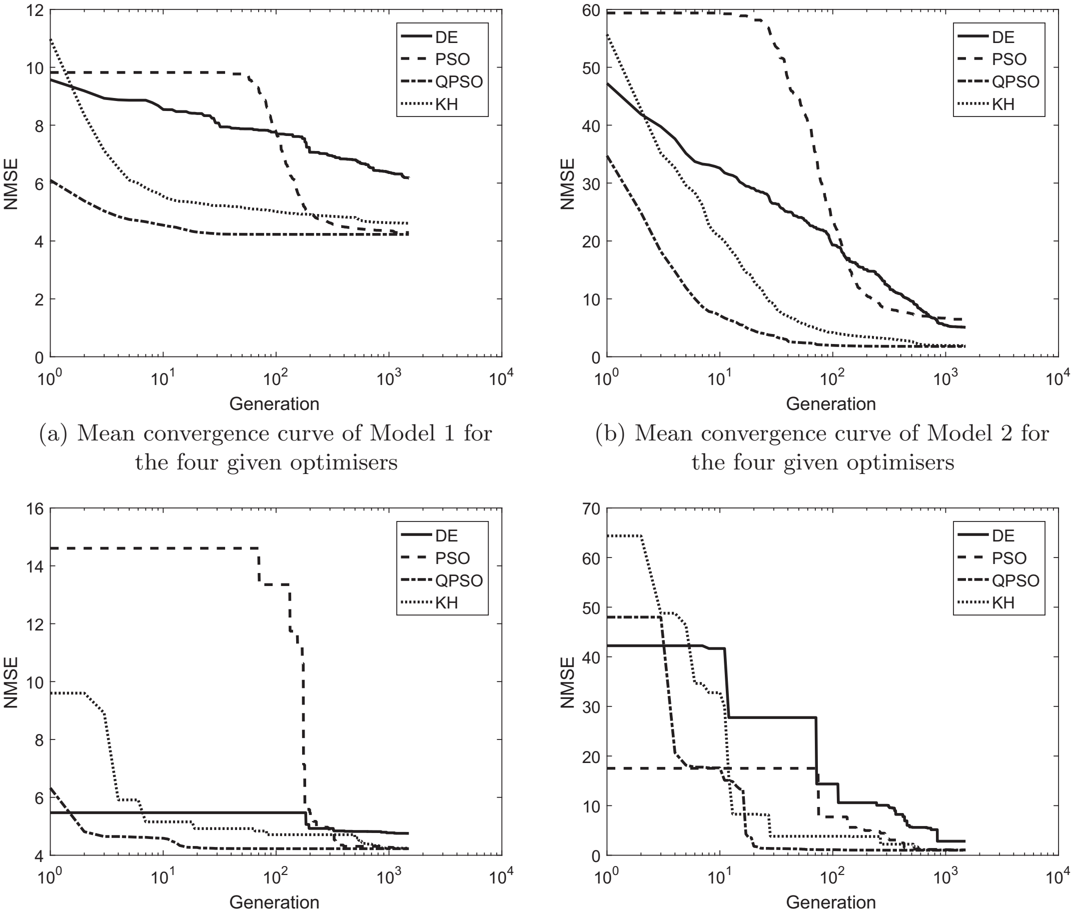
The krill herd is another population-based optimiser which aims to mimic the behaviour of Antarctic krill, it has been shown that the krill move according to three factors: the movement of neighbouring krill, a foraging action, and physical diffusion [64]. This leads to the updates of the particle positions in the KH algorithm being a combination of vectors modelling these three factors. The particles are affected by: the motion of neighbouring particles and the global best particle, a weighted average of the global best solution and the centre of mass of all individuals (food location), which relates to the foraging motion; the physical diffusion is modelled by a random perturbation on particle positions.[6]

Each of the optimisation schemes was run 50 times, with 1500 generations and a population size of 500. The best run and average run results were compared using the NMSE of Eq. (6); this gives a value of zero for a perfect model or 100 if the predictions, $\hat{y}$, are set to the mean of the true outputs. The results of the optimisation are shown in Fig. 27. It can be seen that the first model initialises with a lower NMSE; this is likely to be a result of the lower-dimensional parameter space. However, the mean and best errors in training are significantly improved with the addition of the loss terms in Model 2. As the dimensionality of the parameter space increases, the differentiation between the optimisation schemes becomes more apparent; for this particular problem, the QPSO method not only provides the best overall training and testing error, it also exhibits very fast convergence speeds.

Model 1 shows good performance, with a NMSE below 5; in this case the ability of both the PSO and QPSO methods to find a good set of parameters is clearly seen (Table 4).

The addition of the loss parameters in Model 2 allowed a significant improvement of over three percentage points in the training error and over four points in the model prediction error, as shown in Table 5. Since the additional model terms have physical meaning, an improvement in model behaviour was expected, but not to the extent seen, given the relatively small effect the friction and geometry losses have on the system. As stated earlier, the model was considered unlikely to overfit in

[6] This last benchmark has been used as an opportunity to illustrate a number of recent optimisation algorithms for the identification problem. As observed earlier, in general terms, the ‘no free lunch’ theorems [1] suggest that, averaged over all possible types of problem, no one algorithm will win overall; however, it is interesting to consider newer algorithms for the SI context. The krill algorithm has been chosen somewhat arbitrarily from the (literal) zoo of nature-inspired algorithms, including those based on: ants, bats, bees, dolphins, fireflies, flowers, monkeys, etc.; such algorithms tend to bear more of a family resemblance to the PSO types of algorithm than the DE types.



(a) Mean convergence curve of Model 1 for the four given optimisers

(b) Mean convergence curve of Model 2 for the four given optimisers

(c) Best convergence curve of Model 1 for the four given optimisers

(d) Best convergence curve of Model 2 for the four given optimisers

Fig. 27. Comparison of mean and best convergence curves for both the physical models. Tested with the four different optimisers: differential evolution, particle swarm, quantum particle swarm, and krill herd.

Table 4
Normalised mean-square errors for each optimisation scheme during its best run in training and the model prediction error on the test set, for that best set of parameters, for Model 1.

Optimiser	Training	Testing
DE	4.5815	6.8107
PSO	**4.2276**	**5.9309**
QPSO	**4.2276**	5.9313
KH	4.2447	5.9329

Table 5
Normalised mean square errors for each optimisation scheme during its best run in training and the model prediction error on the test set for that best set of parameters using Model 2.

Optimiser	Training	Testing
DE	2.5116	3.9537
PSO	1.7416	3.1210
QPSO	**1.0174**	**1.7759**
KH	1.0804	1.7816

the traditional sense of an overcomplicated model, due to the additional terms being directly related to physical phenomena. If the terms relating to the losses had been negligible, this should have led to the optimisation scheme returning very small values for these parameters and little improvement in NMSE would be seen. The more accurate the model of the physics used, the better the performance of that model, provided the model incorporates true physical behaviour.

5.2. Black box modelling

For the black-box model used to augment the physical white box, a Gaussian Process (GP) regression model [12,65] has been used; in particular, the GP-NARX model of [66]. The GP-NARX model is a combination of the standard GP regression model with a nonlinear auto-regressive framework. That is, combinations of lags in the input and output dimensions are used to form a higher-dimensional input to the GP model; in addition to this, a lag multiplier is introduced to increase the spacing between data points. For the sake of completeness, a very condensed description of the GP-NARX model is provided here.

5.2.1. Gaussian process NARX models

The basic premise of a Gaussian process (GP) is to perform inference over *functions* directly, as opposed to inference over *parameters* of a function. In short, a GP is a distribution over functions, which is conditioned on training data so that the most probable functions are the best fits to the data.

Let $X = [\underline{x}_1, \underline{x}_2 \dots \underline{x}_N]^T$ denote a matrix of multivariate training inputs, and $\underline{y}$ denote the corresponding vector of training outputs. The input vector for a testing point will be denoted by the column vector $\underline{x}^*$ and the corresponding (unknown) output by y^*. A Gaussian process prior is formed by assuming a (Gaussian) distribution over *functions*,

$$f(\underline{x}) \sim \mathcal{GP}(m(\underline{x}), k(\underline{x}, \underline{x})) \tag{35}$$

where $m(\underline{x})$ is the *mean function* and $k(\underline{x}, \underline{x}')$ is a positive-definite *covariance function.*

One of the defining properties of the GP is that the density of a finite number of outputs from the process, both observed and unobserved, is multivariate normal. This property, combined with standard results for Gaussian distributions, can be used to condition unobserved points on observed training points: this mechanism effectively fits the GP to the training data.

Following a Bayesian approach, the prior mean can be assumed to be zero (see [65] for a discussion). Assuming a Gaussian noise model with variance σ_n^2, the joint distribution for training and testing values is,

$$\begin{pmatrix} \underline{y} \\ y^* \end{pmatrix} \sim \mathcal{N}\left(\underline{0}, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X, \underline{x}^*) \\ K(\underline{x}^*, X) & K(\underline{x}^*, \underline{x}^*) + \sigma_n^2 \end{bmatrix}\right) \tag{36}$$

where $K(X,X)$ is a matrix whose i,jth element is equal to $k(\underline{x}_i, \underline{x}_j)$. Similarly, $K(X, \underline{x}^*)$ is a column vector whose ith element is equal to $k(\underline{x}_i, \underline{x}^*)$, and $K(\underline{x}^*, X)$ is the transpose of the same.

In order to make use of the above, it is necessary to re-arrange the joint distribution $p(\underline{y}, y^*)$ into a conditional distribution $p(y^*|\underline{y})$. Using standard results for the conditional properties of a Gaussian reveals [65],

$$y^* \sim \mathcal{N}(m^*(\underline{x}^*), k^*(\underline{x}^*, \underline{x}^*)) \tag{37}$$

where

$$m^*(\underline{x}^*) = K(\underline{x}^*, X)[k(X,X) + \sigma_n^2 I]^{-1}\underline{y} \tag{38}$$

is the *posterior mean* of the GP and,

$$k^*(\underline{x}^*, \underline{x}') = k(\underline{x}^*, \underline{x}') - K(\underline{x}^*, X)[K(X,X) + \sigma_n^2 I]^{-1} K(X, \underline{x}') \tag{39}$$

is the posterior variance.

Thus the GP model provides a posterior distribution for the unknown quantity y^*. The mean from Eq. (37) can then be used as a 'best estimate' for a regression problem, and the variance can also be used to define confidence intervals. The covariance function used here is the squared-exponential function, and its hyperparameters, augmented by the noise parameter σ_n^2, can be readily found through maximum likelihood estimation. For considerably more details on GPs than this short description allows, see [65].

To apply GPs in the NARX framework, one simply performs a GP regression of the form,

$$y_i = F(y_{i-1}, \dots, y_{i-n_y}; x_i, \dots, x_{i-n_x}) \tag{40}$$

i.e. regress the current system output on a range of past system inputs and outputs.[7] After fitting the GP-NARX model, one way to test it is to compute *one step ahead* (OSA) predictions, which exclusively use the training data up to that time (as discussed earlier in Section 3.1, this is *prediction* in the electrical engineering/control community), i.e.

$$y_i^* = F(y_{i-1}, \dots, y_{i-n_y}; x_i, \dots, x_{i-n_x}) \tag{41}$$

[7] In this implementation of NARX, the current input is also included in the model.

and then compute an error measure. However, a more demanding test is to compute the *Model Predicted Output* (MPO), which uses predicted y^* values instead of observed y values (and thus corresponds to *simulation* in EE/control terminology). It is defined by,

$$y_i^* = F(y_{i-1}^*, \ldots, y_{i-n_y}^*; x_i, \ldots, x_{i-n_x}) \tag{42}$$

and this test can be conducted on testing data as well as training data, which is an important consideration in the more general context of machine learning.

The introduction of the NARX structure to the GP model presents two key challenges; the first is the addition of hyperparameters associated with the number of lags in the model and the lag multiplier; the second is that the action of feeding outputs back onto the input in the GP NARX model breaks one of the basic assumptions of the GP – that there is no noise on the input. The issue of identifying the number of lags in the model would normally be addressed by the use of a validation set or a method such as leave-one-out (LOO) cross-validation [22]. In the case of the cascaded tanks system, the small dataset (1024 points) and the lack of a validation set meant that an alternative approach had to be taken. One of the key stages of fitting a GP model is the minimisation of the negative log marginal likelihood of the process through optimisation of the hyperparameters from the covariance function [65]; since the number of options for the lag hyperparameters was small here, a manual search based on the negative log marginal likelihood of the training data allowed selection of the lag hyperparameters. This manual search led to the identification of 15 lags in both input and output with a lag multiplier of two. From this, the inputs to the GP were taken as $x_t, x_{t-3}, \ldots, x_{t-30}$ and at $y_{t-1}, y_{t-3}, \ldots, y_{t-31}$. A squared-exponential kernel with automatic relevance determination (ARD) was used for the GP covariance structure; the hyperparameters for this can be optimised using a conjugate gradient descent [67,68] or an evolutionary search. On the other matter, concerning input noise; in the case of the GP-NARX models here, the noise levels estimated are very small and so should have little effect when fed back onto the input. The limit being that, if there is no noise on the output and the predictions have zero error there is no difference between OSA and MPO predictions (and the true outputs). Current work is progressing on incorporating methods to propagate the uncertainty introduced in this step, where it is thought that the noise level is significant.

5.2.2. Results: black box

Figs. 28(a) and (b) show the ability of the GP NARX model to perform very well in the OSA (*prediction*) case with an NMSE of only 0.0565 and all observations lying within the 3σ confidence intervals. The results in the MPO (*simulation*) case, however, have an NMSE of 4.6174. This performance is better than the initial white-box model (Model 1), but with the addition of the loss terms in Model 2, the white-box outperforms the GP-NARX black box. The MPO predictions of the GP-NARX model also have a number of test points lying outside the 3σ confidence intervals, which indicates that the model is overconfident in its predictions. This is likely to be a result of the fact that, in training, true values for the output data are fed into the model, this effectively trains the model to make OSA predictions; this would not be an issue if the OSA prediction error was approximately zero; however, in the MPO case, the effect of feeding back noise/error on the outputs to the inputs can be clearly seen to be detrimental to model performance.

5.3. Grey box modelling

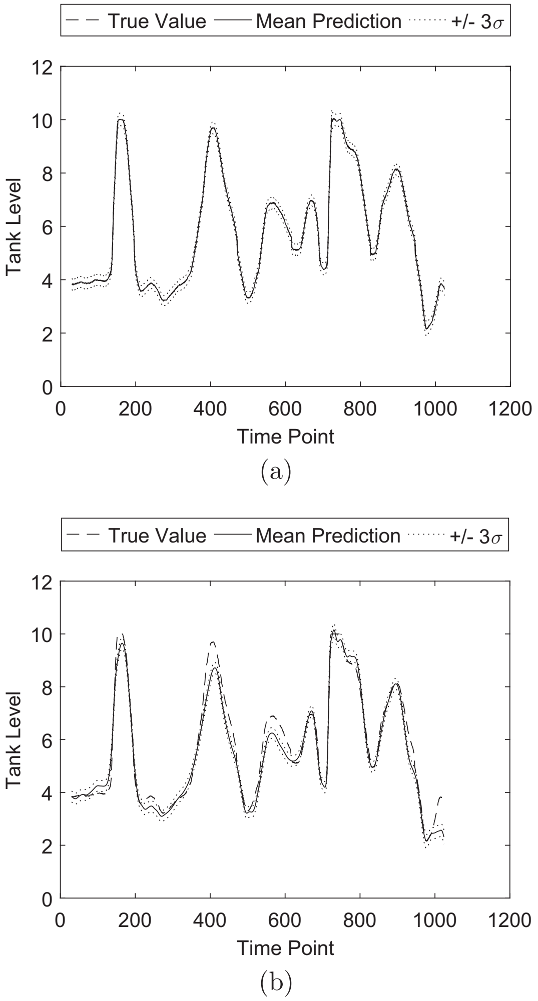
The aim of building a grey-box model as discussed in the introduction (also, see [69]), is to make use of all the prior knowledge as to the system structure which is encoded in the white-box models and then improve predictive performance of the model with the addition of a black-box component. One way to do this would be to treat the white box as a mean function of the process and attempt to fit the residuals of the model using a black-box algorithm, as in,

$$\mathbf{y_p} = \overbrace{f(X)}^{\text{White Box}} + \overbrace{\epsilon(X)}^{\text{Black Box}} \tag{43}$$

The alternative to this is to use the information that is encoded in the white box model as an additional strongly correlated input to the black box, as in Eq. (44); this allows a nonlinear relationship to be established between the white-box outputs and the true system outputs. This incorporation of the white-box outputs as inputs to the black-box model retains a good signal-to-noise ratio which can be lost when only fitting the residuals.

$$\mathbf{y_p} = \overbrace{g(X, \underbrace{f(X)}_{\text{White Box}})}^{\text{Black Box}} \tag{44}$$

Since the error in the white-box models was low, it was found that the formulation of the grey box in (44) was a far more effective model. The grey box was tested with outputs from both white-box models to establish if the quality of the white-box model fit affected the performance of the grey box significantly. It was hypothesised that if the white box were sufficiently accurate, such that it had predictions which were on the noise floor of the data, the addition of a black-box component would be unable to either improve the predictive performance or would be susceptible to overfitting, despite the Bayesian formulation of the GP models.



(a)

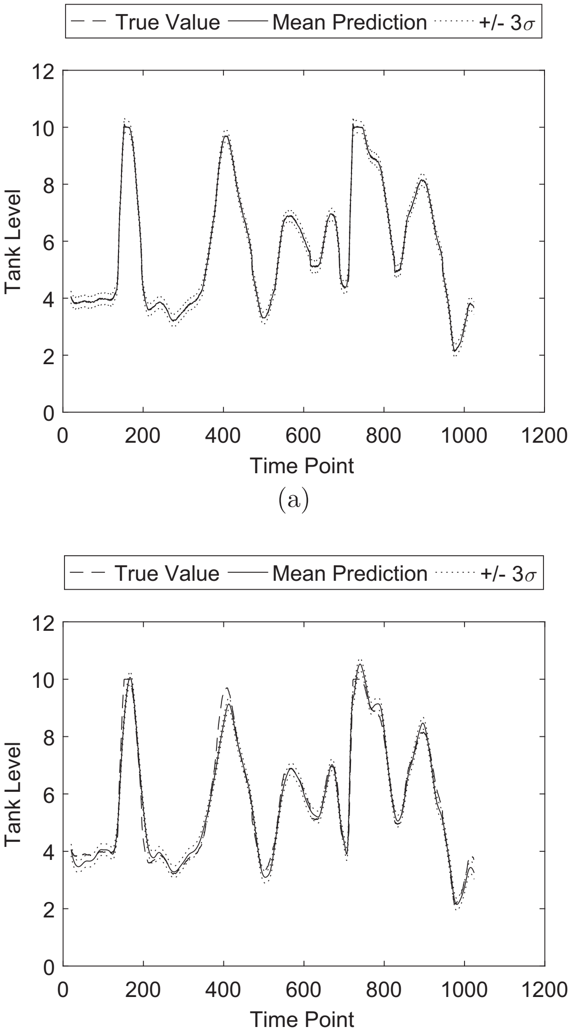
(b)

Fig. 28. Comparison of measured and predicted outputs for cascaded tanks problem using Gaussian process black-box model: (a) One-step-ahead predictions (NMSE 0.0565); (b) Model-predicted output (NMSE 4.6174).

It was again necessary to determine the lag hyperparameters for the GP-NARX model, and this was accomplished as before, with the result being two lags on the inputs, ten on the outputs, and a lag multiplier of two. These lag hyperparameters were used for both grey-box models to allow comparison; since the lags relate to the dynamic behaviour of the model there should be minimal difference in the dynamics of the two white-box models due to their similar formulation. When checking that this assumption was valid, it was found that the same set of lags were optimal in both cases with respect to the marginal likelihood.

Using the outputs of the Model 1 white box as informative inputs to the GP-NARX model yields the results seen in Fig. 29. As before, very good performance is seen in the OSA case, with an NMSE of 0.0529. There is also a significant improvement over both white-box Model 1, and the black-box model, with the MPO error lowered to 2.4621. This indicates that the white-box model is able to provide useful information about the input space to the GP-NARX model, which is not explicitly present in the training dataset. It should be noted that this model, although offering good improvement in terms of the NMSE metric, has a large overestimation of water level in the MPO prediction around time point 750; there, the model clearly predicts a tank level over 10.0, despite this being physically impossible. This error demonstrates the importance of clear engineering interpretations of model outputs to ensure model validity.

Fig. 30 shows the outputs of the second grey-box model in the OSA and MPO case. Here the inputs of white-box Model 2 are used as informative inputs. The structure reduces the NMSE in the OSA case further to 0.0442, and in the MPO case the NMSE is lowered to 0.8178. The key improvement is seen around time point 450 where the second grey-box model shows much better performance than the GP-NARX black box, or the first grey box. It is expected that it is this area of the input space which is not well explored in the training data; however, the accuracy of the white-box Model 2 is able to provide information in this area.



(a)

(b)

Fig. 29. Comparison of measured and predicted outputs for cascaded tanks problem using grey-box model based on white-box Model 1: (a) One-step-ahead predictions (NMSE 0.0529); (b) Model-predicted output (NMSE 2.4621).

A summary of the best MPO performance of every model tested is shown in Table 6. The performance of the grey-box models shows significant improvement over both the white-box models which provide the additional informative inputs, and the use of just a black-box model. In fact the grey-box model incorporating white box model 2 provided the lowest FoM achieved in the workshop.

5.4. Discussion

Although the results shown here establish this form of grey-box modelling as a powerful approach to the NLSI problem, there are a number of caveats that must be addressed before it can be implemented. The first of these is the requirement for well-understood physics that can be accurately described (at least up to a point) in a series of state equations. There are many systems considered across engineering in which the governing equations may not be as easily elicited as in the cascaded tanks; this is the main reason why NLSI is not simply a matter of machine learning.

It is worth considering the complexity of a model required to allow good performance of a system; for example, one could ask if the use of a linear dynamic model as a white box would be sufficient for fitting the grey box with good predictive performance. The authors believe that, provided the white box is not incorrect in such a way as it confounds the structure of the residuals, even a simple white box will aid the performance of the grey box. It remains to be seen if the use of models which simplify the physics of a system (e.g. finite element methods) will leave enough structure in the residuals for the machine learning method to fit.

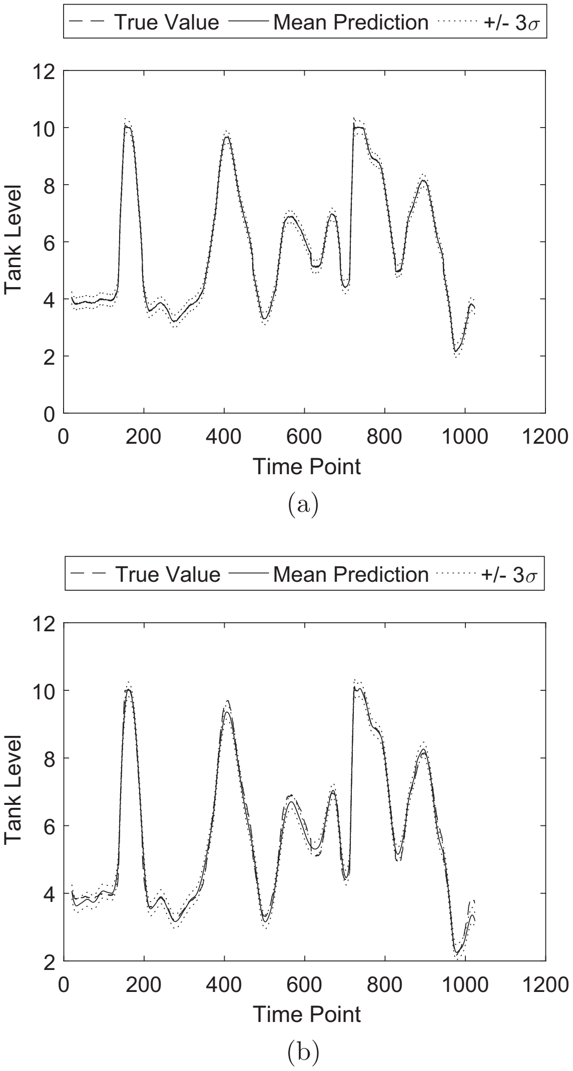


Fig. 30. Comparison of measured and predicted outputs for cascaded tanks problem using grey-box model based on white-box Model 2 (the extended physics model): (a) One-step-ahead predictions (NMSE 0.0529); (b) Model-predicted output (NMSE 2.4621).

Table 6
Best NMSE and FoM scores for all five models tested in the MPO case.

	White 1	White 2	Black	Grey 1	Grey 2
Best NMSE	5.9309	1.7759	4.6174	2.4621	**0.8178**
Best FoM	0.5115	0.2799	0.4550	0.3319	**0.1913**

It should be noted that the simulation of the white-box component of the model was computationally inexpensive for the cascaded tanks system. If the white-box portion of the model is computationally very expensive, there must be some investigation as to the cost-benefit of increasing model fidelity when the machine learning method will attempt to fit the more complex physics that is not modelled.

In summary, when conducting NLSI in the presence of well-understood physics, the incorporation of this prior knowledge in any machine learning technique is a powerful tool. The grey-box framework presented here is an intuitive and modular approach that has several advantages, these are: retention of engineering insight in the processes that are well understood in the white box and ability to compare differing black-box methods without the requirement to refit the white-box component (this is especially valuable where modelling the physical system is computationally expensive).

6. Conclusions

It would be unduly repetitive to summarise the local conclusions from the various studies contained in this paper. The overall conclusions can therefore be quite brief. Following an initial discussion on the nature of nonlinear system identification in general, the results from three case studies on benchmark problems are presented here. In all cases, an evolutionary optimisation scheme of some flavour provides a good, if not excellent, solution. In fact, the approaches presented here gave the best results of the workshop on two of the three benchmarks (One and Three). This supports the contention, expressed in the paper's introduction, that general frameworks for nonlinear system identification are beginning to emerge, potentially moving the discipline forward from its 'toolbox' phase. Some general elements of good practice are highlighted along the way, among them is the fact that careful design of an 'optimal' input excitation can make a serious difference to the results obtained. Another important lesson is that a grey-box model combining a physical white-box and a nonparametric machine learning algorithm can provide a predictive model superior to white- or black-box models alone. While the evolutionary approaches have succeeded in all cases here, it is important to note that equally general frameworks are emerging based on Bayesian machine learning. Although not explored here for reasons of space, such frameworks offer the possible advantage of combining parameter estimation with model selection. In order to see how such models can be applied to the benchmarks here, the reader has only to consult some of the other papers in this special issue.

Acknowledgements

The support of the UK Engineering and Physical Sciences Research Council (EPSRC) through grant reference numbers EP/J016942/1 and EP/K003836/2 is gratefully acknowledged. In addition, Tim Rogers is grateful to Ramboll Oil and Gas for financial support and Geoff Holmes similarly thanks Innovate UK.

References

[1] D. Wolpert, W. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997) 67.
[2] L. Ljung, System Identification: Theory for the User, second ed., Prentice Hall, 1999.
[3] T. Söderstrom, P. Stoica, System Identification, new ed., Prentice Hall, 1994.
[4] D. Ewins, Modal Testing: Theory, Practice and Application, Wiley, Blackwell, 2000.
[5] K. Worden, G. Tomlinson, Nonlinearity in Structural Dynamics: Detection, Modelling and Identification, Institute of Physics Press, 2001.
[6] G. Kerschen, K. Worden, A. Vakakis, J.-C. Golinval, Past, present and future of nonlinear system identification in structural dynamics, Mech. Syst. Signal Process. 20 (2006) 505–592.
[7] J.-P. Nöel, G. Kerschen, 10 years of advances in nonlinear system identification in structural dynamics: a review, in: Proceedings of ISMA 2016-International Conference on Noise and Vibration Engineering, 2016.
[8] G. Klir, Uncertainty and Information: Foundations of Generalized Information Theory, John Wiley and Sons, 2005.
[9] G. Pillonetto, G.D. Nicolao, A new kernel-based approach for linear system identification, Automatica 46 (2010) 81–93.
[10] G. Pillonetto, F. Dinuzzo, T. Chen, G.D. Nicolao, L. Ljung, Kernel methods in system identification, machine learning and function estimation: a survey, Automatica 50 (2014) 657–682.
[11] S. Billings, Nonlinear System Identification: NARMAX, Methods in the Time, Frequency, and Spatio-Temporal Domains, Wiley-Blackwell, 2013.
[12] D. Barber, Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012.
[13] C. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), first ed. 2006., Springer, New York, 2007.
[14] K. Worden, J. Hensman, W. Staszewski, Natural computing for mechanical systems research: a tutorial overview, Mech. Syst. Signal Process. 25 (2011) 4–11.
[15] K. Worden, G. Manson, On the identification of hysteretic systems, Part I: fitness landscapes and evolutionary identification, Mech. Syst. Signal Process. 29 (2012) 201–212.
[16] R. Bouc, Forced vibration of mechanical systems with hysteresis, in: Proceedings of the 4th Conference on Non-linear Oscillation, Prague, Czechoslovakia, 1967.
[17] Y.-K. Wen, Method for random vibration of hysteretic systems, J. Eng. Mech. Div. 102 (1976) 249–263.
[18] M. Yar, J. Hammond, Parameter estimation for hysteretic systems, J. Sound Vib. 117 (1987) 161–172.
[19] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (1997) 341–359.
[20] K. Worden, W. Becker, On the identification of hysteretic systems, Part II: Bayesian sensitivity analysis and parameter confidence, Mech. Syst. Signal Process. 29 (2012) 213–227.
[21] D. Mackay, Information Theory, Inference and Learning Algorithms, Cambridge University Press, 2003.
[22] C. Bishop, Pattern Recognition and Machine Learning, Springer, 2013.
[23] J. Beck, Statistical system identification of structures, in: Proceedings of 5th International Conference on Structural Safety and Reliability, ASCE, New York, 1919, pp. 1395–1402.
[24] J. Beck, L. Katafygiotis, Updating models and their uncertainties. I: Bayesian statistical framework, ASCE J. Eng. Mech. 124 (1998) 455–461.
[25] J. Beck, S.-K. Au, Bayesian updating of structural models and reliability using Markov chain Monte Carlo simulation, ASCE J. Eng. Mech. 128 (2002) 380–391.
[26] J. Beck, K.-V. Yuen, Model selection using response measurements: Bayesian probabilistic approach, ASCE J. Eng. Mech. 130 (2004) 192–203.
[27] M. Muto, J. Beck, Bayesian updating and model class selection for hysteretic structural models using stochastic simulation, J. Vib. Control 14 (2008) 7–34.
[28] M. Girolami, Bayesian inference for differential equations, Theoret. Comput. Sci. 408 (2008) 4–16.
[29] B. Calderhead, M. Girolami, D. Higham, Is it safe to go out yet? Statistical inference in a zombie outbreak model, Preprint, University of Strathclyde, Department of Mathematics and Statistics, 2010.
[30] K. Popper, The Logic of Scientific Discovery, Basic Books, 1959.
[31] C.C. de Wit, H. Olsson, K. Aström, P. Lischinky, New model for control of systems with friction, IEEE Trans. Autom. Control 40 (1995) 419–425.
[32] T. Piatkowski, Dahl and LuGre dynamic friction models – the analysis of selected properties, Mech. Mach. Theory 73 (2014) 91–100.
[33] J. Swevers, F. Al-Bender, C. Ganesman, T. Prajogo, An integrated friction model structure with improved presliding behavior for accurate friction compensation, IEEE Trans. Autom. Control 45 (2000) 675–686.

[34] K. Worden, C. Wong, U. Parlitz, A. Hornstein, D. Engster, T. Tjahjiwidodo, F. Al-Bender, D. Risos, S. Fassois, Identification of pre-sliding and sliding friction dynamics: grey box and black box models, Mech. Syst. Signal Process. 21 (2007) 514–524.

[35] A. Visitin, Differential Models of Hysteresis, Springer, Berlin, 1994.

[36] J. Judd, Analytical Modeling of Wood-frame Shear Walls and Diaphragms (Masters Thesis), Brigham Young University, 2005.

[37] T. Baber, Y. Wen, Random vibrations of hysteretic degrading systems, ASCE J. Eng. Mech. 107 (1981) 1069–1089.

[38] T. Baber, M. Noori, Random vibration of degrading pinching systems, ASCE J. Eng. Mech. 111 (1985) 1010–1026.

[39] T. Baber, M. Noori, Modeling general hysteresis behaviour and random vibration applications, J. Vib. Acoust. Stress Reliab. Des. 108 (1986) 411–420.

[40] A. Abdessalem, N. Dervilis, D. Wagg, K. Worden, Model selection and parameter estimation in structural dynamics using approximate Bayesian computation, Mech. Syst. Signal Process. 99 (2018) 306–325.

[41] A. Abdessalem, N. Dervilis, D. Wagg, K. Worden, Model selection and parameter estimation of dynamical systems using a novel variant of approximate Bayesian computation, Mech. Syst. Signal Process. 2018 (submitted for publication).

[42] J. Noël, M. Schoukens, Hysteretic Benchmark With a Dynamic Nonlinearity, Technical Report: Aerospace and Mechanical Engineering Department, University of Liége, 2015.

[43] J. Zhang, A. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (2009) 945–958.

[44] A. Abdessalem, N. Dervilis, D. Wagg, K. Worden, Identification of nonlinear dynamical systems using approximate Bayesian computation based on a sequential Monte Carlo sampler, in: Proceedings of 27th International Conference on Noise & Vibration Engineering, Leuven, 2016.

[45] M. Schoukens, J. Noël, Wiener-Hammerstein Benchmark with Process Noise, Technical Report:, ELEC Department, Vrije Universiteit Brussel, Brussels, Belgium, 2015.

[46] M. Schoukens, G. Vandersteen, Y. Rolain, F. Ferranti, Fast identification of Wiener-Hammerstein systems using discrete optimization, IET Electron. Lett. 50 (2014) 1942–1944.

[47] J. Sjöberg, L. Lauwers, J. Schoukens, Identification of Wiener-Hammerstein models: two algorithms based on the best split of a linear model applied to the SYSID'09 benchmark problem, Control Eng. Pract. 20 (2012) 1119–1125.

[48] O. Dewhirst, D. Simpson, N. Angarita, R. Allen, P. Newland, Wiener-Hammerstein parameter estimation using differential evolution: application to limb reflex dynamics, in: International Conference on Bio-inspired Systems and Signal Processing, 2010, pp. 271–276.

[49] A. Naitali, F. Giri, Wiener-Hammerstein system identification – an evolutionary approach, Int. J. Syst. Sci. 47 (2016) 45–61.

[50] M. Schoukens, R. Pintelon, Y. Rolain, Identification of Wiener-Hammerstein systems by a nonparametric separation of the best linear approximation, Automatica 50 (2014) 628–634.

[51] A. Qin, P. Suganthan, Self-adaptive differential evolution algorithm for numerical optimization, in: 2005 IEEE Congress on Evolutionary Computation, 2005, pp. 1785–1791.

[52] V. Huang, A. Qin, P. Suganthan, Self-adaptive differential evolution algorithm for constrained real-parameter optimization, in: IEEE Congress on Evolutionary Computation, Vancouver, Canada, 2006, pp. 17–24.

[53] S. Billings, S. Fakhouri, Identification of a class of nonlinear systems using correlation analysis, Proc. IEE 125 (1978) 691–697.

[54] J. Sjöberg, J. Schoukens, Initializing Wiener-Hammerstein models based on partitioning of the best linear approximation, Automatica 48 (2012) 353–359.

[55] M. Schoukens, K. Worden, Evolutionary identification of block-structured systems, in: Proceedings of IMAC XXXV – 35th International Modal Analysis Conference, 2017.

[56] A. Marconato, J. Sjöberg, J. Suykens, J. Schoukens, Identification of the Silverbox benchmark using nonlinear state-space models, in: 16th IFAC Symposium on System Identification (SYSID), 2012, pp. 632–637.

[57] C. Chen, S. Fassois, Maximum likelihood identification of stochastic Wiener-Hammerstein-type non-linear systems, Mech. Syst. Signal Process. 6 (1992) 135–153.

[58] A. Hagenblad, L. Ljung, A. Wills, Maximum likelihood identification of Wiener models, Automatica 44 (2008) 2697–2705.

[59] M. Schoukens, P. Mattsson, T. Wigren, J.-P. Noël, Cascaded Tanks Benchmark Combining Soft and Hard Nonlinearities, Technical Report: ELEC Department, Vrije Universiteit Brussel, Brussels, Belgium, 2016.

[60] R. Eberhart, J. Kennedy, et al., A new optimizer using particle swarm theory, in: Proceedings of the 6th International Symposium on Micro Machine and Human Science, New York, vol. 1, 1995, pp. 39–43.

[61] R. Malik, T. Rahman, S. Hashim, R. Ngah, New particle swarm optimizer with sigmoid increasing inertia weight, Int. J. Comput. Sci. Secur. 1 (2007) 35–44.

[62] J. Sun, B. Feng, W. Xu, Particle swarm optimization with particles having quantum behavior, in: Congress on Evolutionary Computation, 2004.

[63] A. Gandomi, A. Alavi, Krill herd: a new bio-inspired optimization algorithm, Commun. Nonlinear Sci. Numer. Simul. 17 (2012) 4831–4845.

[64] E. Hofmann, E. Haskell, J. Klinck, C. Lascara, Lagrangian modelling studies of Antarctic krill (*Euphausia Superba*) swarm formation, ICES J. Mar. Sci.: J. Conseil 61 (2004) 617–631.

[65] C. Rasmussen, C. Williams, Gaussian Processes for Machine Learning, The MIT Press, 2006.

[66] K. Worden, W. Becker, T. Rogers, E. Cross, On the confidence bounds of Gaussian process NARX models and their higher-order frequency response functions, Mech. Syst. Signal Process. 104 (2018) 188–233.

[67] R. Fletcher, C. Reeves, Function minimization by conjugate gradients, Comput. J. 7 (1964) 149–154.

[68] Y. Dai, Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, SIAM J. Optimiz. 10 (1999) 177–182.

[69] O. Nelles, Nonlinear System Identification: from Classical Approaches to Neural Networks and Fuzzy Models, Springer Science & Business Media, 2013.