Computer Methods in Applied Mechanics and Engineering 418 (2024) 116502

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

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

Inferring unknown unknowns: Regularized bias-aware ensemble Kalman filter

Andrea Nóvoa [a,c], Alberto Racca [b,c,a], Luca Magri [c,d,a,*]

[a] Cambridge University Engineering Dept., Trumpington St, Cambridge, CB2 1PZ, UK
[b] Imperial College London, I-X Centre for AI In Science, White City Campus, 84 Wood Lane, London, W12 7SL, UK
[c] Imperial College London, Aeronautics Dept., Exhibition Road, London, SW7 2AZ, UK
[d] The Alan Turing Institute, 96 Euston Rd, London, NW1 2DB, UK

ARTICLE INFO

Keywords:
Model bias
Sequential data assimilation
Reservoir computing
Ensemble Kalman filter

ABSTRACT

Because of physical assumptions and numerical approximations, low-order models are affected by uncertainties in the state and parameters, and by model biases. Model biases, also known as model errors or systematic errors, are difficult to infer because they are 'unknown unknowns', i.e., we do not necessarily know their functional form *a priori*. With biased models, data assimilation methods may be ill-posed because either (i) they are 'bias-unaware' because the estimators are assumed unbiased, (ii) they rely on an *a priori* parametric model for the bias, or (iii) they can infer model biases that are not unique for the same model and data. First, we design a data assimilation framework to perform combined state, parameter, and bias estimation. Second, we propose a mathematical solution with a sequential method, i.e., the *regularized bias-aware ensemble Kalman Filter* (r-EnKF), which requires a model of the bias and its gradient (i.e., the Jacobian). Third, we propose an echo state network as the model bias estimator. We derive the Jacobian of the network, and design a robust training strategy with data augmentation to accurately infer the bias in different scenarios. Fourth, we apply the r-EnKF to nonlinearly coupled oscillators (with and without time-delay) affected by different forms of bias. The r-EnKF infers in real-time parameters and states, and a unique bias. The applications that we showcase are relevant to acoustics, thermoacoustics, and vibrations; however, the r-EnKF opens new opportunities for combined state, parameter and bias estimation for real-time and on-the-fly prediction in nonlinear systems.

1. Introduction

Data Assimilation (DA) optimally combines uncertain measurements with an imperfect numerical model of the physical quantity of interest in order to find the best estimate of the full physical state and/or model parameters (e.g., [1,2]). DA has been developed in the last decades in the earth sciences and numerical weather prediction (NWP) communities by combining concepts from control theory, dynamic programming, and Bayesian statistics. In the NWP terminology, DA obtains the most likely estimates of the state and/or model parameters (the *analysis*) that are compatible with the background knowledge from the numerical model (the *forecast*), and a set of measured data (the *observations*), as well as with their corresponding uncertainties.

The assimilation framework can be affected by errors in the observations (e.g., due to poorly calibrated or malfunctioning sensors), and/or forecast model errors (e.g., model assumptions, or discretization errors). If the errors are non-aleatoric or with

* Corresponding author at: Imperial College London, Aeronautics Dept., Exhibition Road, London, SW7 2AZ, UK.
E-mail address: l.magri@imperial.ac.uk (L. Magri).

https://doi.org/10.1016/j.cma.2023.116502
Received 6 June 2023; Received in revised form 1 September 2023; Accepted 28 September 2023
Available online 14 October 2023
0045-7825/© 2023 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Nomenclature

Data assimilation symbols

$\boldsymbol{\alpha}$	Model parameters
$\boldsymbol{b}$	Model bias
$\mathbf{C}_{bb}$	Bias error covariance matrix
$\mathbf{C}_{dd}$	Observation error covariance matrix
$\mathbf{C}_{\psi\psi}$	Ensemble error covariance matrix
$\boldsymbol{d}$	Observable data
Δt	Time step
Δt_{d}	Time between observables
$\boldsymbol{\epsilon}$	Gaussian noise
$\mathcal{F}$	Nonlinear differential operator
γ	Bias regularization parameter
$\mathbb{I}_n$	Identity matrix of $n \times n$ dimensions
$\mathcal{J}$	DA cost function
$\mathbf{J}$	Jacobian (gradient of the bias with respect to the analysis state)
$\mathcal{M}$	Nonlinear measurement operator
$\mathbf{M}$	Linear measurement operator
m	Ensemble size
$\mathbf{M}\boldsymbol{\psi}$	Biased model prediction
$\mathcal{N}$	Normal random distribution
N	Augmented state vector length
N_α	Number of inferred parameters
N_ϕ	Number of state variables
N_q	Number of observables
$\boldsymbol{\phi}$	State variables
$\boldsymbol{\psi}$	Augmented state vector
σ_α	Ensemble parameters' initial standard deviation
σ_ϕ	Ensemble states' initial standard deviation
t_{err}	Error metrics time window
$\mathcal{U}$	Uniform random distribution
$\boldsymbol{y}$	Unbiased model prediction

Echo state network symbols

Δt_{ESN}	ESN time step
δ_{r}	Symmetry-breaking constant
$\boldsymbol{g}$	Input normalizing factors
λ	Tikhonov parameter
L	Number of training datasets
N_r	Number reservoir states (neurones)
N_{tr}	Length of the training set
ρ	Spectral radius
$\boldsymbol{r}$	Reservoir state vector
σ_{in}	Input scaling
t_{tr}	Training time
t_{val}	Validation time
$\mathbf{W}$	Reservoir state matrix
$\mathbf{W}_{\mathrm{in}}$	Input matrix
$\mathbf{W}_{\mathrm{out}}$	Output matrix

non-zero mean, they are systematic errors, which are also known as *bias*, and they may be a function of the state, space, time, and/or the environment. The terms 'bias', 'systematic error', and 'model error' are equivalent to each other for the purpose of this paper, i.e., they are informally referred to as the 'unknown unknowns'. Biases in the assimilation can be detected by monitoring

the long-term statistics of the *innovations*, which are the residuals between the observations and the forecast equivalents [3]. If the innovations are not Gaussian in time, the assimilation framework may become biased, and the analysis resulting from a DA scheme may be sub-optimal [4]. Therefore, the bias must be explicitly and carefully handled in operational DA. Nonetheless, traditional DA is designed to adjust the estimates with respect to unbiased Gaussian errors in the model and observations, thus, we refer to these as *bias-unaware* methods. If unbiased observations are assimilated into a biased model with a bias-unaware method, the resulting analysis will be sub-optimal (e.g., [5,6]). Moreover, a biased model can lead to increased analysis errors, which may result in an analysis varying with the assimilation frequency [7].

Current *bias-aware* DA schemes include bias estimation methods that are either (i) off-line, which perform first, bias-unaware DA to estimate the bias from the forecast innovations, and second, a bias-aware DA by correcting the forecast with the bias estimate from the bias-unaware DA; or (ii) adaptive, which estimate the bias as part of the assimilation cycle to remove the bias of the analysis. Adaptive bias-aware schemes are more versatile because they are robust to transient changes in the bias form [8]. In this work, we focus on sequential adaptive schemes. Bias-aware methods have been mainly implemented in variational DA. For instance, in NWP the four-dimensional variational (4D-Var) method was extended to consider explicitly an adaptive model bias in the variational problem. The bias correction in 4D-Var has been deployed in the model space (e.g. [9,10]), as well as in the observation space (e.g. [11,12]). Bias correction in the observations is currently operational at the European Centre for Medium-range Weather Forecasting [13]. Other applications of variational adaptive bias correction schemes can be found in ocean studies (e.g. [14,15]).

Sequential assimilation, also known as real-time DA (e.g., [6]), bypasses the need of storing and post-processing batches of data by assimilating the observations on the fly, which makes them suitable for real-time prediction and control applications. Real-time state and/or parameter estimation is often implemented in a Kalman filter framework [16], which provides the optimal solution in a Bayesian framework for a linear system. Several modifications to the filter allow us to extend and generalize the Kalman filter to nonlinear systems and nonlinear state spaces, such as the extended Kalman filter, particle filters, or ensemble filtering methods (e.g., [2]). Nevertheless, these are *bias-unaware* formulations. Bias estimation in real-time schemes dates back to Friedland [17], who proposed a *separate-bias* Kalman filter scheme, which consists of solving two Kalman filter problems, one for the physical model state and parameter estimation, and another for estimating the parameters of the bias model. The separate-bias Kalman filter, which was initially limited to an adaptive linear biases, was generalized to handle nonlinear and stochastic bias models (e.g. [18–20]). Further, Dee and da Silva [21] extended the separate filter by coupling the two sets of problems using the updated bias estimates to remove the bias of the forecast during the assimilation. However, the coupled filter can become costly for multivariate models because it requires two analysis steps [15]. Rubio et al. [22] proposed a sequential DA with bias correction in a Bayesian framework, which combined reduced-order models and transport map sampling. More recently, da Silva and Colonius [23] deployed an ensemble Kalman filter for aerodynamic flows in which the system dynamics are augmented with a low-rank representation of the flow discretization error. By inferring the parameters of the low-rank model, they could perform a bias-aware estimation of the flow-field under significant discretization bias (which is a subset of the model bias that we are set to tackle in this paper). The need for a pre-defined model for the bias means that the accuracy of the methods is bounded to the complexity and suitability of this pre-defined model. Further, the increase in the number of unknown parameters to infer can affect the conditioning of the minimization problem [5].

Alternatively, hybrid DA methods, which decouple the estimation of the model bias from the assimilation framework to allow more flexibility and complexity in the model of the bias have been deployed in computational mechanics (e.g., [24]). These hybrid methods estimate the model bias using data-driven model identification tools and Kalman filter frameworks (e.g. [25–27]). Haik et al. [28] proposed a parameterized-background-data-weak technique to estimate the bias introduced with model-order reduction tools. Recent advances in machine learning for data-driven modelling allow us to develop surrogate models of dynamical systems using neural networks, either as the full representation of the dynamics, or on top of the pre-existing imperfect physical model as a hybrid model (e.g., [24,29,30]). The hybrid approach is particularly attractive from a model-bias inference perspective because neural networks can be deployed to close the low-order model, i.e., to estimate the bias of the model. The similarities between machine learning and data assimilation have recently encouraged the combination of both inverse problems [31,32]. For instance, Brajard et al. [33] applied an ensemble Kalman filter to a surrogate model provided by a convolutional neural network, to allow the latter to work under sparse observations. In model bias applications, Bonavita and Laloyaux [34] implemented a fully-connected artificial neural network for atmospheric model bias correction in a weakly-constraint 4D-Var framework; and Farchi et al. [35] proposed a network with a combination of convolutional and fully-connected layers in the same variational framework. Training convolutional and fully connected networks may be computationally expensive because they use back-propagation, which makes them expensive for sequential data assimilation. In real-time assimilation, Nóvoa and Magri [6] explained that echo state networks, which are a subset of reservoir computers, are more suitable for real-time DA because their training consists of a computationally-cheap linear regression problem. Nóvoa and Magri [6] employed an echo state network to provide an estimate of the bias of a low-order model, and correct the forecast state prior to applying a square-root ensemble Kalman filter [2]. They showed that the trained network can estimate in time the bias of the state without the need to infer additional parameters. However, even if we have a universal approximator acting as a closure model such that observation data is assimilated into a bias-corrected state, current bias-aware schemes do not ensure that the bias is unique, and the analysis could lead to unrealistically-large bias estimates [13]. Indeed, current filters do not have information on the magnitude of the bias of the forecast (its norm), thus, there is no constraint on the evolution of the bias. This means that the model state and parameters may be markedly inaccurate if the bias in not uniquely inferred.

To summarize, the available adaptive bias-aware methods face two limitations when dealing with unknown biases: (1) they rely on an *a priori* definition of a parameterized model for the bias whose parameters are inferred within the assimilation framework,

and (2) they do not provide a unique estimate of the bias. The overarching objective of this work is to generalize current bias-aware data assimilation methods to perform real-time state, parameter and model bias estimation of low-order models. We implement a three-fold strategy: first, we propose a regularized bias-aware sequential DA framework; second, we provide mathematical solution to the framework with a sequential method, which performs on-the-fly state, parameter and model bias inference, whilst providing a solution with an unique model bias; and third, we generalize the echo state network implementation of Nóvoa and Magri [6] for a more versatile and robust bias estimator. We test the proposed method in nonlinearly coupled oscillators with and without time delays, in which we consider different biases.

This paper is divided into seven sections, including this introduction. Section 2 formalizes the problem and provides an overview of bias-unaware and bias-aware sequential DA. Section 3 introduces and derives the proposed regularized bias-aware ensemble Kalman filter. Section 4 describes the echo state network deployed for model bias estimation and the training approach; and details the derivation of the echo state network's Jacobian. Section 5 describes the implementation of the developed method for combining the proposed data assimilation framework with an echo state network. In Section 6, we test the proposed framework in two numerical scenarios. Section 7 ends the paper with the conclusions.

2. Sequential data assimilation for low-order models

This section provides the motivation as well as the background knowledge on data assimilation required for the proposed regularized bias-aware sequential data assimilation framework for low-order models, which is derived in Section 3. We define the model in Section 2.1, and cast the framework to an ensemble framework into Section 2.2. We provide an overview of the bias-unaware ensemble Kalman filter in Section 2.3 and the current bias-aware assimilation practice in Section 2.4.

2.1. The model

We consider a deterministic dynamical system

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{\phi}}{\mathrm{d}t} &= \mathcal{F}(\boldsymbol{\phi}, \boldsymbol{\alpha}), \\ \boldsymbol{y} &= \mathcal{M}(\boldsymbol{x}, \boldsymbol{\phi}) + \boldsymbol{b}, \end{aligned} \tag{1}$$

where $\boldsymbol{y} \in \mathbb{R}^{N_q}$ are the model *observables*, which describe the dynamics of a measurable physical quantity; $\mathcal{F} : \mathbb{R}^{N_\phi} \to \mathbb{R}^{N_\phi}$ is a nonlinear operator, which is a function of the model parameters $\boldsymbol{\alpha} \in \mathbb{R}^{N_\alpha}$ and operates on the state variables $\boldsymbol{\phi} \in \mathbb{R}^{N_\phi}$; $\mathcal{M} : \mathbb{R}^{N_\phi} \to \mathbb{R}^{N_q}$ is the measurement operator that maps the state variables into the observable space at the spatial locations $\boldsymbol{x}$; and $\boldsymbol{b} \in \mathbb{R}^{N_q}$ is the bias. (The terms 'bias', 'systematic error', and 'model error' are equivalent to each other for the purpose of this paper, i.e., they are informally referred to as the 'unknown unknowns'. In this paper, we will use the term 'bias'.) The bias may arise from modelling assumptions, and from approximations in the operators $\mathcal{F}$ and $\mathcal{M}$. We define the bias of the observable space because this is the space in which the error is measurable. $\mathcal{F}$ can represent the spatial discretization of a partial differential equation and its boundary conditions.

In reality, numerical models are subject to uncertainties in the state and parameters, which are commonly modelled as stochastic processes. Data assimilation can be used to improve our knowledge of the model state and parameters. This process is known as combined state and parameter estimation, which typically requires reformulating the problem with an augmented state vector comprising of the model state variables and the poorly-known parameters (e.g., [36]). For reasons that will become apparent in Section 2.2, we augment further the state vector with the mapped model observables, such that, we define an augmented state vector $\boldsymbol{\psi} = [\boldsymbol{\phi}; \boldsymbol{\alpha}; \mathcal{M}(\boldsymbol{x}, \boldsymbol{\phi})] \in \mathbb{R}^{N=N_\phi+N_\alpha+N_q}$, where the operator $[\ ;\ ]$ indicates horizontal concatenation. This allows us to formally have a linear measurement operator $\mathbf{M}$ in the problem definition, such that (1) can be written as

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{\psi}}{\mathrm{d}t} &= \mathbf{F}(\boldsymbol{\psi}) + \boldsymbol{\epsilon}_\psi, \\ \boldsymbol{y} &= \mathbf{M}\boldsymbol{\psi} + \boldsymbol{b} + \boldsymbol{\epsilon}_y \end{aligned} \tag{2}$$

where $\mathbf{F}(\boldsymbol{\psi}) = [\mathcal{F}(\boldsymbol{\phi}, \boldsymbol{\alpha}); \boldsymbol{0}]$ is the nonlinear operator augmented with the vector of zeroes $\boldsymbol{0} \in \mathbb{R}^{N_\alpha+N_q}$ (as the parameters and observation operator are constant in time); $\boldsymbol{\epsilon}_\psi = [\boldsymbol{\epsilon}_\phi; \boldsymbol{\epsilon}_\alpha; \boldsymbol{\epsilon}_y]$ is Gaussian noise that models the aleatoric uncertainties in the augmented state, which encapsulates the uncertainties in the state variables, $\boldsymbol{\epsilon}_\phi$, in the model parameters, $\boldsymbol{\epsilon}_\alpha$, and in the measurement operator, $\boldsymbol{\epsilon}_y$; and $\mathbf{M} = \left[\mathbf{0} \,\middle|\, \mathbb{I}_{N_q}\right]$ is the vertical concatenation of a matrix of zeros, $\mathbf{0} \in \mathbb{R}^{N_q \times (N_\phi+N_\alpha)}$, and the identity matrix, $\mathbb{I}_{N_q} \in \mathbb{R}^{N_q \times N_q}$. In this work, we refer to $\mathbf{M}\boldsymbol{\psi}$ as the 'biased' model prediction, and to $\boldsymbol{y}$ as the 'unbiased' model prediction.

Sequential DA updates the model state $\boldsymbol{\phi}$ and the parameters $\boldsymbol{\alpha}$ when observation data of the modelled physical quantity are available (e.g. from experiments) by combining the observations with the model estimate $\boldsymbol{y}$. In a sequential framework, the observation data at time t, $\boldsymbol{d}(t)$, is a single measurement realization of the quantity of interest subject to uncertainty, $\boldsymbol{\epsilon}_d$,

$$\boldsymbol{d}(t) = \boldsymbol{d}^{\mathrm{t}}(t) + \boldsymbol{\epsilon}_d, \tag{3}$$

where the superscript 't' stands for 'true', i.e., $\boldsymbol{d}^{\mathrm{t}}(t)$ is the true physical quantity (which we do not know). We assume that the observations are unbiased and Gaussian distributed, i.e. $\boldsymbol{\epsilon}_d \sim \mathcal{N}\left(\boldsymbol{0}, \mathbf{C}_{dd}\right)$, where $\mathcal{N}$ denotes the normal distribution with zero mean and covariance matrix $\mathbf{C}_{dd}$, which is pre-defined as a diagonal matrix assuming that the observations are uncorrelated. Thus,

the expectation of the observables coincides with the true physical state. In a real scenario, this would mean that the experimental devices are calibrated correctly, which is the overarching working assumption in this paper. In this numerical study, we use synthetic observable data generated by adding noise to the *truth* (see Sections 6.2–6.3). Lastly, we define the bias of the observable space as the difference between the truth and the expectation of the model, i.e.,

$$\boldsymbol{b} = \mathbb{E}(\boldsymbol{y}) - \mathbb{E}(\mathbf{M}\boldsymbol{\psi}) = \boldsymbol{d}^{\mathrm{t}} - \mathbb{E}(\mathbf{M}\boldsymbol{\psi}). \tag{4}$$

2.2. Ensemble data assimilation framework

We tackle the DA problem from a Bayesian framework. Our knowledge in a Markovian model prediction at time t_k is quantified by the conditional probability $\boldsymbol{\psi}_k \sim \mathcal{P}(\boldsymbol{\psi}_k|\boldsymbol{\psi}_{k-1}, \mathbf{F})$. This knowledge is updated every time at which we have reference data from observations $\boldsymbol{d}_k$ by using the Bayes rule, such that the *posterior* (i.e., the confidence in our prediction $\boldsymbol{\psi}_k$ after observing $\boldsymbol{d}_k$) is given by

$$\mathcal{P}(\boldsymbol{\psi}_k|\boldsymbol{d}_k, \boldsymbol{b}_k, \mathbf{F}) \propto \mathcal{P}(\boldsymbol{d}_k|\boldsymbol{\psi}_k, \boldsymbol{b}_k, \mathbf{F})\mathcal{P}(\boldsymbol{\psi}_k, \boldsymbol{b}_k, \mathbf{F}), \tag{5}$$

where $\mathcal{P}(\boldsymbol{\psi}_k, \boldsymbol{b}_k, \mathbf{F})$ is the *prior*, which measures the confidence in our model's estimate if there were no available observables $\boldsymbol{d}_k$; and $\mathcal{P}(\boldsymbol{d}_k|\boldsymbol{\psi}_k, \boldsymbol{b}_k, \mathbf{F})$ is the *likelihood*, which measures the confidence that we have in the model prediction. The mode, i.e., most probable value, of $\boldsymbol{\psi}_k$ in the posterior is defined as the best estimator of the state, which is the *analysis* state. In other words, the analysis is the statistically consistent combination of prior physical knowledge and observed data, accounting for the model mismatch.

If the operator $\mathbf{F}$ were linear, the mode of the Bayesian update (5) would be exactly corrected by the Kalman filter equations [16]. For nonlinear cases, a common strategy consists of reformulating the problem with an ensemble approach, in which the prior is estimated by evolving in time a number of m realizations of the state. The ensemble framework is widely implemented because it avoids the computation of adjoint solvers (which are needed in variational DA) and does not need to propagate the covariance matrix in time in contrast to, e.g., the extended Kalman filter (e.g., [37]). The expected value and the covariance matrix of the state are estimated by the ensemble statistics

$$\mathbb{E}(\boldsymbol{\psi}) \approx \overline{\boldsymbol{\psi}} = \frac{1}{m}\sum_{j=1}^{m}\boldsymbol{\psi}_j \quad \text{and} \quad \mathbf{C}_{\psi\psi} = \begin{bmatrix} \mathbf{C}_{\phi\phi} & \mathbf{C}_{\phi\alpha} & \mathbf{C}_{\phi q} \\ \mathbf{C}_{\alpha\phi} & \mathbf{C}_{\alpha\alpha} & \mathbf{C}_{\alpha q} \\ \mathbf{C}_{q\phi} & \mathbf{C}_{q\alpha} & \mathbf{C}_{qq} \end{bmatrix} \approx \frac{1}{m-1}\sum_{j=1}^{m}(\boldsymbol{\psi}_i - \overline{\boldsymbol{\psi}}) \otimes (\boldsymbol{\psi}_i - \overline{\boldsymbol{\psi}}), \tag{6}$$

where $\otimes$ is the dyadic product. We generate the initial ensemble from a Gaussian distribution, i.e, $\boldsymbol{\psi}_j \sim \mathcal{N}(\overline{\boldsymbol{\psi}}, \mathbf{C}_{\psi\psi})$. Each member in the ensemble is forecast individually as

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{\psi}_j}{\mathrm{d}t} &= \mathbf{F}(\boldsymbol{\psi}_j) + \boldsymbol{\epsilon}_\psi, \\ \boldsymbol{y}_j &= \mathbf{M}\boldsymbol{\psi}_j + \boldsymbol{b}_j + \boldsymbol{\epsilon}_y, \qquad \text{for} \quad j = 0, \dots, m-1, \end{aligned} \tag{7}$$

where $\boldsymbol{b}_j$ is the bias of each ensemble member. Because the forecast operator $\mathcal{F}$ is nonlinear, the Gaussian prior may not remain Gaussian after the model forecast, and $\mathbb{E}(\mathcal{F}(\boldsymbol{\psi})) \neq \mathcal{F}(\overline{\boldsymbol{\psi}})$. However, we work under the assumption that the time between analyses Δt_{d} is small enough such that the Gaussian distribution is not significantly distorted, as assumed in nonlinear DA problems [2]. Therefore, the ensemble filter is a nonlinear model that approximates the mode of the Bayesian update.

2.3. Bias-unaware ensemble Kalman filter

For each member in the ensemble, the ensemble Kalman filter (EnKF) minimizes the uncertainty in the augmented state (by reducing the covariance) and the Mahalanobis distance between the model forecast and the measurements. The cost function for each ensemble member is (e.g., [38])

$$\mathcal{J}(\boldsymbol{\psi}_j) = \left\|\boldsymbol{\psi}_j - \boldsymbol{\psi}_j^{\mathrm{f}}\right\|^2_{\mathbf{C}^{\mathrm{f}-1}_{\psi\psi}} + \left\|\mathbf{M}\boldsymbol{\psi}_j - \boldsymbol{d}\right\|^2_{\mathbf{C}^{-1}_{dd}}, \quad j = 0, \dots, m-1. \tag{8}$$

where the superscript 'f' stands for 'forecast', and the precision matrices $\mathbf{C}^{-1}$ weigh the L_2-norms. The state covariance matrix, $\mathbf{C}_{\psi\psi}$, is computed via Eq. (6). The observation error covariance, $\mathbf{C}_{dd}$, is pre-defined from the knowledge of the source of the data (e.g. precision of the measuring devices). Each ensemble member must be updated with a different observations to avoid the underestimation of the analysis covariance [39]. This can be achieved by adding Gaussian noise to the observables such that each j ensemble member is assimilated with $\boldsymbol{d}_j \sim \mathcal{N}\left(\boldsymbol{d}, \mathbf{C}_{dd}\right)$, where $\mathbf{C}_{dd} = \overline{\boldsymbol{\epsilon}_d \otimes \boldsymbol{\epsilon}_d}$.

An analysis state $\boldsymbol{\psi}_j^{\mathrm{a}}$ minimizes (8) if the derivative of the cost function with respect to the state is zero, which leads to the EnKF equations [40]

$$\boldsymbol{\psi}_j^{\mathrm{a}} = \boldsymbol{\psi}_j^{\mathrm{f}} + \mathbf{C}^{\mathrm{f}}_{\psi\psi}\mathbf{M}^{\mathrm{T}}\left(\mathbf{C}_{dd} + \mathbf{M}\mathbf{C}^{\mathrm{f}}_{\psi\psi}\mathbf{M}^{\mathrm{T}}\right)^{-1}\left[\boldsymbol{d}_j - \mathbf{M}\boldsymbol{\psi}_j^{\mathrm{f}}\right], \quad j = 0, \dots, m-1, \tag{9}$$

where the superscript 'a' stands for 'analysis'. From (9) it can be seen that the updates to the forecast are confined to the range of the forecast ensemble error covariance matrix $\mathbf{C}^{\mathrm{f}}_{\psi\psi}$. Therefore, the size and spread of the ensemble are key to the performance of the filter. In the limit of an infinite ensemble, the EnKF (9) is no longer an approximation, and if the operator $\mathbf{F}$ is unbiased, the mean of the analysis ensemble would recover the true state and parameters [2]. However, in practice there is a trade-off between the bias of a model (its accuracy) and its computational requirements. In fact, sequential DA methods are typically employed in real-time and on-the-fly prediction applications, which often use computationally-cheap, thus, potentially biased models e.g. [41].

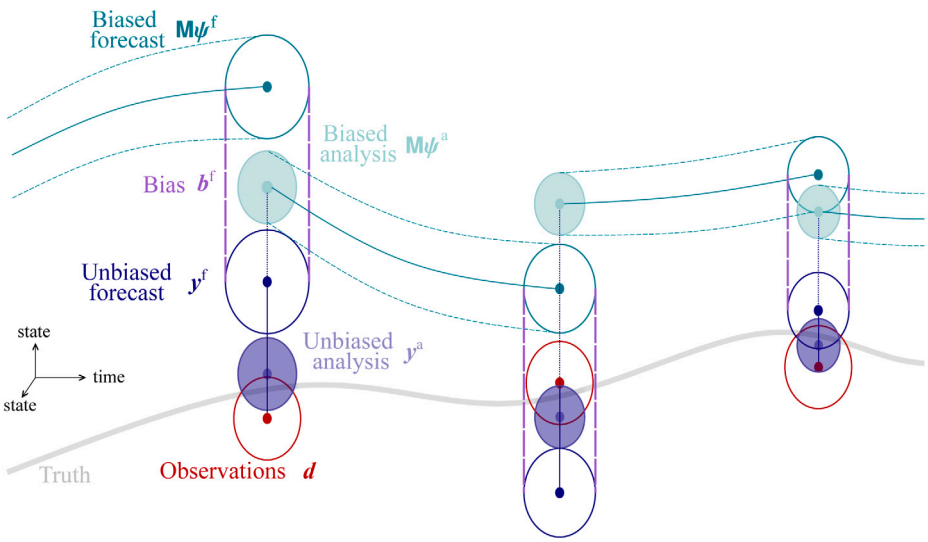

Fig. 1. Conceptual schematic of a sequential bias-aware data assimilation process. Bias-aware DA aims to estimate the truth (grey line) by combining measurable observations (red) and their uncertainties (represented by the circles) with the predictions from a biased model, which propagates in time (teal lines). To assimilate the observations into the biased model, we must provide an estimate of the bias of the forecast (vertical violet dashed lines), which maps the biased forecast and its uncertainty (teal) into an unbiased state (navy). The assimilation between the observations and the unbiased forecast results in the unbiased analysis with a reduced uncertainty (shaded navy). This analysis indirectly updates the biased forecast to a biased analysis (shaded teal), which serves as the updated initial condition for the next forecast step. This process is repeated sequentially in time when observations are available. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

2.4. Bias-aware ensemble Kalman filter

We develop a sequential DA method to predict in real-time complex dynamical systems using low-order models. We define a low-order model as a qualitatively accurate model whose simulation time is of the same order as the timescales of experimental measurements (i.e., they provide real-time predictions), and whose approximations and physical assumptions negatively affect its quantitative accuracy (i.e., low-order models may incur model bias). The model bias must be treated explicitly to mitigate sub-optimal analysis [5]. In a bias-aware framework, we need to provide an estimate of the bias at the time of assimilation, such that the biased forecast is mapped into an unbiased forecast, which, subsequently, can be combined with the observations data. This process is illustrated in Fig. 1.

Current bias-aware methods rely on defining *a priori* a parameterized model to estimate the bias in the assimilation (e.g., [12]). The parameters of the bias model are computed within the ensemble assimilation framework as a parameter estimation problem for $\boldsymbol{\beta}_j$. Thus, the cost function to minimize includes the estimation of the bias model parameters as

$$\mathcal{J}(\boldsymbol{\psi}_j, \boldsymbol{\beta}_j) = \left\|\boldsymbol{\psi}_j - \boldsymbol{\psi}_j^{\mathrm{f}}\right\|^2_{\mathbf{C}^{\mathrm{f}-1}_{\psi\psi}} + \left\|\boldsymbol{y}_j - \boldsymbol{d}_j\right\|^2_{\mathbf{C}^{-1}_{dd}} + \left\|\boldsymbol{\beta}_j - \boldsymbol{\beta}_j^{\mathrm{f}}\right\|^2_{\mathbf{C}^{\mathrm{f}-1}_{\beta\beta}}, \quad j = 0, \dots, m-1, \tag{10}$$

where $\boldsymbol{y}_j = \mathbf{M}\boldsymbol{\psi}_j + \boldsymbol{b}_j(\boldsymbol{\beta}_j)$ is the unbiased estimate, and $\mathbf{C}^{\mathrm{f}}_{\beta\beta}$ is the bias parameters' error covariance matrix. This method is typically implemented in a variational approach (e.g. [9]). Alternatively, the parameters can be inferred through a sequential method using ensemble Kalman filters [23], which consist of augmenting the state vector with the bias parameters.

The accuracy of the solution from (10) depends on the suitability and complexity of the pre-defined functional form for the bias, $\boldsymbol{b}_j(\boldsymbol{\beta}_j)$. Increased complexity implies an increased number of model bias parameters to infer. However, augmenting the parameter estimation variables can affect the conditioning of the minimization problem [5]. To overcome this limitation, Nóvoa and Magri [6] proposed an Echo State Network (ESN), which is a reservoir computer tool, to address the challenge of bias estimation. The ESN bypasses the need of inferring the bias model parameters by autonomously inferring the bias in real time. Therefore, in contrast to the cost function dependent on bias parameters (10), Nóvoa and Magri [6] use a sequential ensemble square-root Kalman filter that minimizes

$$\mathcal{J}(\boldsymbol{\psi}_j) = \left\|\boldsymbol{\psi}_j - \boldsymbol{\psi}_j^{\mathrm{f}}\right\|^2_{\mathbf{C}^{\mathrm{f}-1}_{\psi\psi}} + \left\|\boldsymbol{y}_j - \boldsymbol{d}_j\right\|^2_{\mathbf{C}^{-1}_{dd}}, \quad j = 0, \dots, m-1, \tag{11}$$

where $\boldsymbol{y}_j = \mathbf{M}\boldsymbol{\psi}_j + \boldsymbol{b}^{\mathrm{f}}$, and the ESN estimates the bias forecast $\boldsymbol{b}^{\mathrm{f}}$, which is defined as an ensemble average (see Section 3.2). The network is re-initialized with the mean analysis innovation, $\boldsymbol{d} - \mathbf{M}\overline{\boldsymbol{\psi}}^{\mathrm{a}}$, which allows the reservoir to be synchronized with the physical forecast model. We detail this implementation of the ESN in Section 4.

The bias-aware analysis schemes (10) and (11) are designed to correct the bias of the forecast state (or equivalently in the observations) and, therefore, they reduce the mean analysis increments. However, they do not necessarily improve the analysis

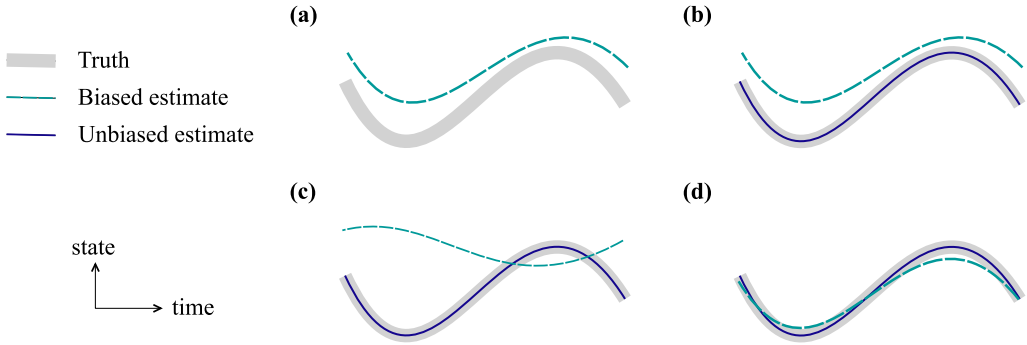

Fig. 2. Pictorial comparison between (a) bias-unaware data assimilation; (b–d) bias-aware assimilation, in which the unbiased estimate (navy) matches the truth (grey). If the bias-aware framework is not regularized, it may indistinguishably recover solutions with (b) unchanged, (c) increased, or (d) reduced bias norm, as compared to the bias-unaware method. The bias is seen as the distance between the biased estimate (dashed blue) and truth. The proposed regularized framework favours the small-norm bias scenario (d) if the bias regularization factor $\gamma > 0$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

solutions, i.e., the assimilation may lead to an analysis state with a larger-norm bias than that of the forecast state. Fig. 2 illustrates the different unbiased analyses that may arise using current bias-aware DA methods. This raises the conceptual question "what is a 'good' unbiased analysis?". We generalize the implementation of the ESN as the bias estimator in Section 4.

3. Regularized bias-aware ensemble Kalman filter (r-EnKF)

In a scenario with a biased model, we judge that an unbiased analysis is 'good' if the unbiased state (i) matches the truth, and (ii) has a small-norm bias, as illustrated in Fig. 2d. We are assuming that the model captures the main physical mechanisms needed to provide an estimate that is compatible with experimental data, i.e., the model is qualitatively accurate. Further, we assume that we can find a combination of parameters in the model that minimizes the norm of the model bias. With this in mind, we formulate a data assimilation problem that not only does it minimize the estimation uncertainty and its distance to the observables, but it also reduces the bias of the estimator. Mathematically, we pose the problem by regularizing (11) such that

$$\mathcal{J}(\boldsymbol{\psi}_j) = \left\|\boldsymbol{\psi}_j - \boldsymbol{\psi}_j^{\mathrm{f}}\right\|^2_{\mathbf{C}^{\mathrm{f}-1}_{\psi\psi}} + \left\|\boldsymbol{y}_j - \boldsymbol{d}_j\right\|^2_{\mathbf{C}^{-1}_{dd}} + \gamma \left\|\boldsymbol{b}_j\right\|^2_{\mathbf{C}^{-1}_{bb}}, \quad \text{for} \quad j = 0, \dots, m-1 \tag{12}$$

where the operator $\|\cdot\|^2_{\mathbf{C}^{-1}}$ is the L_2-norm weighted by the semi-positive definite matrix $\mathbf{C}^{-1}$, and $\gamma \geq 0$ is a bias regularization factor (note that γ could be merged with $\mathbf{C}_{bb}$, but as detailed in Section 5, we set $\mathbf{C}_{bb} = \mathbf{C}_{dd}$ and use the hyperparameter γ to calibrate the relative sizes of the likelihood and the bias). The 'good' unbiased analysis that we define with cost function (12) accounts for the norm of the bias, in contrast to the possible outputs from current bias-aware frameworks which may result in analysis with increased or unchanged model bias (Fig. 2b,c).

An analysis state $\boldsymbol{\psi}_j^{\mathrm{a}}$ minimizes the regularized bias-aware cost function (12) if, for each ensemble member $j = 0, \dots, m-1$,

$$\frac{1}{2}\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\boldsymbol{\psi}_j}\bigg|_{\boldsymbol{\psi}_j^{\mathrm{a}}} = \mathbf{C}^{\mathrm{f}-1}_{\psi\psi}\left(\boldsymbol{\psi}_j^{\mathrm{a}} - \boldsymbol{\psi}_j^{\mathrm{f}}\right) + \frac{\mathrm{d}\boldsymbol{y}_j}{\mathrm{d}\boldsymbol{\psi}_j}\bigg|^{\mathrm{T}}_{\boldsymbol{\psi}_j^{\mathrm{a}}} \mathbf{C}^{-1}_{dd}\left(\boldsymbol{y}_j^{\mathrm{a}} - \boldsymbol{d}_j\right) + \gamma \frac{\mathrm{d}\boldsymbol{b}_j}{\mathrm{d}\boldsymbol{\psi}_j}\bigg|^{\mathrm{T}}_{\boldsymbol{\psi}_j^{\mathrm{a}}} \mathbf{C}^{-1}_{bb}\boldsymbol{b}_j^{\mathrm{a}} = 0. \tag{13}$$

The hyperparameter γ directly affects the gradient on the left-hand side, whence, the adaptability of the analysis state to the magnitude of the bias. With small bias regularization factors, $\gamma \ll 1$, the analysis will evolve slowly with the bias norm, while large values of γ allow large instantaneous updates to the ensemble state and parameters. To mathematically solve (13), we assume that the analysis state is sufficiently close to the forecast such that we linearize the analysis bias as

$$\boldsymbol{b}_j^{\mathrm{a}} \approx \boldsymbol{b}_j^{\mathrm{f}} + \mathbf{J}^{\mathrm{f}}\mathbf{M}\left(\boldsymbol{\psi}_j^{\mathrm{a}} - \boldsymbol{\psi}_j^{\mathrm{f}}\right), \tag{14}$$

where $\mathbf{J}^{\mathrm{f}} = \frac{\mathrm{d}\boldsymbol{b}_j^{\mathrm{f}}}{\mathrm{d}\mathbf{M}\boldsymbol{\psi}_j^{\mathrm{f}}}$ is the Jacobian. Further details on the forecast bias $\boldsymbol{b}_j^{\mathrm{f}}$ are discussed in Section 3.2. Importantly, the linearization (14) makes the minimization problem quadratic. Introducing (14) into (13), grouping the terms in $\boldsymbol{\psi}_j^{\mathrm{a}}$, and exploiting the Woodbury matrix inversion formula [42], yield the regularized bias-aware ensemble Kalman filter (r-EnKF)

$$\boldsymbol{\psi}_j^{\mathrm{a}} = \boldsymbol{\psi}_j^{\mathrm{f}} + \mathbf{K}\left[\left(\mathbb{I} + \mathbf{J}^{\mathrm{f}}\right)\left(\boldsymbol{d}_j - \boldsymbol{y}_j^{\mathrm{f}}\right) - \gamma\mathbf{C}_{dd}\mathbf{C}^{-1}_{bb}\mathbf{J}^{\mathrm{f}}\boldsymbol{b}_j^{\mathrm{f}}\right], \tag{15}$$

where the regularized Kalman gain matrix $\mathbf{K}$ is

$$\mathbf{K} = \mathbf{C}^{\mathrm{f}}_{\psi\psi}\mathbf{M}^{\mathrm{T}}\left[\mathbf{C}_{dd} + (\mathbb{I} + \mathbf{J}^{\mathrm{f}})\mathbf{M}\mathbf{C}^{\mathrm{f}}_{\psi\psi}\mathbf{M}^{\mathrm{T}}(\mathbb{I} + \mathbf{J}^{\mathrm{f}})^{\mathrm{T}} + \gamma\mathbf{C}_{dd}\mathbf{C}^{-1}_{bb}\mathbf{J}^{\mathrm{f}}\mathbf{M}\mathbf{C}^{\mathrm{f}}_{\psi\psi}\mathbf{M}^{\mathrm{T}}\mathbf{J}^{\mathrm{f}^{\mathrm{T}}}\right]^{-1}. \tag{16}$$

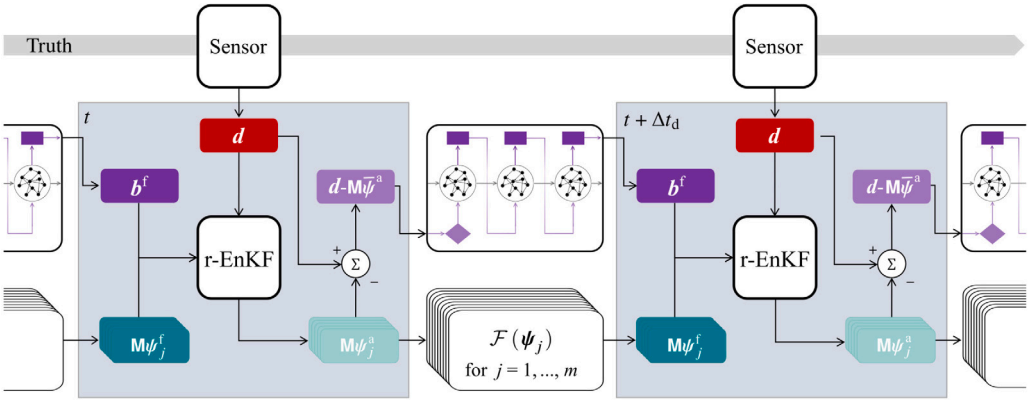

Fig. 3. Schematic of the regularized bias-aware sequential data assimilation with an echo state network (ESN). The low-order model $\mathcal{F}$ propagates each ensemble member to an ensemble of forecast states $\mathbf{M}\boldsymbol{\psi}_j^{\mathrm{f}}$, and the ESN runs autonomously in parallel to estimate the ensemble mean bias, $\boldsymbol{b}^{\mathrm{f}}$. When a sensor provides noisy data $\boldsymbol{d}$, we apply the r-EnKF, which statistically combines the noisy data, the ensemble mean bias, and the forecast ensemble. The r-EnKF results in the analysis ensemble, $\mathbf{M}\boldsymbol{\psi}_j^{\mathrm{a}}$, which gives the new initial conditions for $\mathcal{F}$; and the analysis innovation $\boldsymbol{d} - \mathbf{M}\overline{\boldsymbol{\psi}}^{\mathrm{a}}$ re-initializes the ESN. This process is repeated sequentially every Δt_{d} time between observations.

3.1. Discussion

The r-EnKF Eqs. (15)–(16) constitute a main result of this work as far as sequential data assimilation is concerned. We are defining a regularized bias-aware DA framework, which, in the limiting case of an unbiased (perfect) model, recovers a zero-bias state equivalent to the analysis from the bias-unaware EnKF (9). If the system is unbiased, i.e., $\boldsymbol{b}_j = \boldsymbol{0}$, the r-EnKF (15) simplifies to the bias-unaware EnKF (9). Furthermore, if the bias is independent of the state, i.e., $\mathbf{J} = \mathbf{0}$, the r-EnKF simplifies to the bias-aware EnKF resulting from (11), in which the observations are assimilated into a bias-corrected forecast (e.g. [6]).

Han and Bormann [13] proposed a cost function similar to (12), and derived the constrained form of the variational bias-corrected 4D-Var. This variational method is restricted to pre-defined linear parametrized bias models, whose parameters are inferred in the assimilation analogously to (10). Including a pre-defined bias model in the assimilation can be beneficial if the form of the bias is known *a priori*, and if it can be accurately represented with a pre-defined model.

3.2. Simplification of the bias definition

For brevity, instead of computing an ensemble of biases, $\boldsymbol{b}_j^{\mathrm{f}}$, we work with one echo state network to directly infer the mean bias, $\boldsymbol{b}^{\mathrm{f}}$. Therefore, the ESN forecasts autonomously the bias $\boldsymbol{b}^{\mathrm{f}}$ in time [6]. When observations become available, we update every ensemble member with $\boldsymbol{b}_j^{\mathrm{f}} \equiv \boldsymbol{b}^{\mathrm{f}}$, for $j = 1, \ldots, m$, employing the r-EnKF (15); and finally re-initialize the ESN with the mean analysis innovation, $\boldsymbol{d} - \mathbf{M}\overline{\boldsymbol{\psi}}^{\mathrm{a}}$. This procedure is illustrated in Fig. 3 and detailed in Section 5. Importantly, the transient innovations are not equivalent to the bias, because the model state and parameters are not yet converged. Whereas, at statistical convergence, the ESN prediction and the innovations coincide because of (4), i.e., at convergence, $\langle \boldsymbol{b}^{\mathrm{f}} - (\boldsymbol{d} - \mathbf{M}\boldsymbol{\psi}_j^{\mathrm{f}}) \rangle$ is Gaussian noise, where $\langle \cdot \rangle$ indicates the time-average.

4. Echo state network: an adaptive model for the bias

Echo state networks (ESNs) are a type of recurrent neural networks, which are designed to learn temporal correlations in data through an internal high-dimensional state, the *reservoir* [43,44]. In previous work [6], we showed that ESNs can be used in sequential DA to adaptively infer the bias as the model state and parameters converge in a sequential DA framework. This is achieved by re-initializing the network with the analysis innovations (see Fig. 4b). The motivation behind employing an ESN to model the bias of a low-order model in a sequential Kalman filter framework is three-fold: (i) ESNs are generalized nonlinear auto-correlation functions (Aggarwal [45], p.306), which are universal approximators [46], this means that they can accurately approximate time-varying functions (provided that they are trained on sufficient data); (ii) the ESN can be formulated in the state space; and (iii) training an ESN consists of solving a linear-regression problem [47], which is time-efficient compared to backpropagation with gradient descent methods required in other recurrent architectures, such as long short-term memory networks [45].

In the upcoming subsections, we summarize the state-space description of the ESN (Section 4.1), and the training process (Section 4.2); then, we design a multi-parameter training data generation for a versatile and robust ESN (Section 4.3), and we derive the Jacobian of the ESN, which is required in the r-EnKF (Section 4.4).

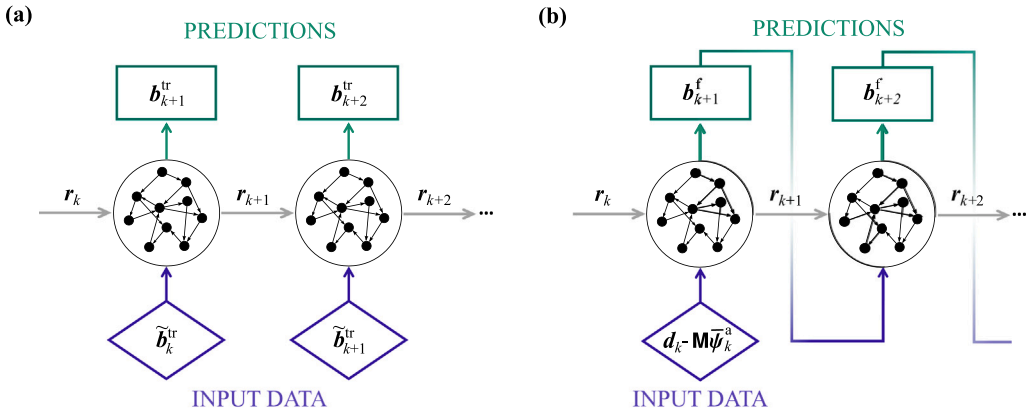

Fig. 4. Schematic of the ESN forecast configurations during (a) training, and (b) assimilation. The subscripts indicate time (e.g., $\boldsymbol{b}_k = \boldsymbol{b}(t_k)$). During training, the ESN is forecast in open-loop, in which the input data (indicated with diamonds) are the noisy training data, and the training aims to match the output to the noise-free training data. During the assimilation, the reservoir is re-initialized at the analysis step with the mean analysis innovation, and, from here, it runs autonomously in closed-loop.

4.1. State-space formulation

The equations governing the time evolution of the bias forecast, $\boldsymbol{b}^{\mathrm{f}}$, given by an ESN are

$$\begin{aligned} \boldsymbol{b}^{\mathrm{f}}(t_{k+1}) &= \mathbf{W}_{\mathrm{out}}\left[\boldsymbol{r}(t_{k+1});1\right] \\ \boldsymbol{r}(t_{k+1}) &= \tanh\left(\sigma_{\mathrm{in}}\mathbf{W}_{\mathrm{in}}\left[\boldsymbol{i}(t_k)\odot\boldsymbol{g};\delta_{\mathrm{r}}\right]+\rho\mathbf{W}\boldsymbol{r}(t_k)\right), \end{aligned} \tag{17}$$

where $\boldsymbol{i}(t_k)$ is the input to the reservoir; $\boldsymbol{r}\in\mathbb{R}^{N_r}$ and $\mathbf{W}\in\mathbb{R}^{N_r\times N_r}$ are the reservoir state and matrix, with a number of reservoir states (or neurons) $N_r \gg N_q$; the operator $[\ ;\ ]$ indicates horizontal concatenation; $\mathbf{W}_{\mathrm{in}}\in\mathbb{R}^{N_r\times(N_q+1)}$ and $\mathbf{W}_{\mathrm{out}}\in\mathbb{R}^{N_q\times(N_r+1)}$ are the input and output matrices, respectively; σ_{in} is the input scaling; ρ is the spectral radius; δ_{r} is a constant used for breaking the symmetry of the ESN (we set $\delta_{\mathrm{r}}=0.1$) [48]; $\boldsymbol{g}=[g_1,\dots,g_{N_q}]$ are the input normalizing factors with $g_q^{-1}=\max(\boldsymbol{b}_q^{\mathrm{tr}})-\min(\boldsymbol{b}_q^{\mathrm{tr}})$, i.e., g_q^{-1} is the range of qth component of the training data $\boldsymbol{b}^{\mathrm{tr}}$ (the superscript 'tr' stands for 'training', which should not be confused with the superscript 't' for 'true'); the operator $\odot$ is the Hadamard product, i.e., component-wise multiplication; and the $\tanh(\cdot)$ operation is performed element-wise. The matrices $\mathbf{W}_{\mathrm{in}}$ and $\mathbf{W}$ are fixed, sparse and randomly generated using a connectivity of 5 (see [43,49], for details), whereas $\mathbf{W}_{\mathrm{out}}$ is obtained through training, which is discussed in Section 4.2. The hyperparameters σ_{in} and ρ are obtained with a recycle validation strategy and a Bayesian optimization following Racca and Magri [49]. The recycle validation is summarized in Appendix A for completeness.

The ESN can evolve in either open or closed loops, in which the main differences affect the inputs to the reservoir, $\boldsymbol{i}(t_k)$ in (17), which in Fig. 4 are illustrated with diamonds. Fig. 4a shows the ESN open-loop configuration employed during training, in which the input is the training bias data, i.e., $\boldsymbol{i}=\tilde{\boldsymbol{b}}^{\mathrm{tr}}$ (the tilde indicates added noise, as detailed in Section 4.3). On the other hand, Fig. 4b shows that, during assimilation, the reservoir is re-initialized at every analysis step with the analysis innovations (which is equivalent to a single open-loop step), i.e., $\boldsymbol{i}=\boldsymbol{d}-\mathbf{M}\overline{\boldsymbol{\psi}}^{\mathrm{a}}$, and, from here, the ESN runs autonomously in closed-loop using the output from the previous step as the input. The implementation is further discussed in Section 5.

4.2. Training the echo state network

A trained ESN is the global minimizer of a quadratic optimization problem [47]

$$\begin{aligned} \min_{\mathbf{W}_{\mathrm{out}}} \quad & \sum_{k=1}^{N_{\mathrm{tr}}-1}\left\|\mathbf{W}_{\mathrm{out}}\left[\boldsymbol{r}(t_k);1\right]-\boldsymbol{b}^{\mathrm{tr}}(t_k)\right\|_2^2+\lambda\left\|\mathbf{W}_{\mathrm{out}}\right\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{r}(t_k)=\tanh\left(\sigma_{\mathrm{in}}\mathbf{W}_{\mathrm{in}}\left[\tilde{\boldsymbol{b}}^{\mathrm{tr}}(t_{k-1})\odot\boldsymbol{g};\delta_{\mathrm{r}}\right]+\rho\mathbf{W}\boldsymbol{r}(t_{k-1})\right), \\ & \boldsymbol{r}(t_0)=\boldsymbol{0}, \end{aligned} \tag{18}$$

where N_{tr} is the length of the training dataset; λ is the Tikhonov regularization hyperparameter; $\boldsymbol{b}^{\mathrm{tr}}(t_k)\in\mathbb{R}^{N_q}$ is the training data at time t_k; and $\tilde{\boldsymbol{b}}^{\mathrm{tr}}$ is the training data with added Gaussian noise, which regularizes the problem (e.g., [49]). The only trainable parameters are the components of the matrix $\mathbf{W}_{\mathrm{out}}$.

During training the ESN is in open-loop configuration, and the aim is to match the outputs (rectangles in Fig. 4a) obtained from the input noisy training data $\tilde{\boldsymbol{b}}^{\mathrm{tr}}$ (diamonds), to the noise-free data $\boldsymbol{b}^{\mathrm{tr}}$ by modifying the weights in $\mathbf{W}_{\mathrm{out}}$. Because the reservoir state $\boldsymbol{r}$ is not a function of $\mathbf{W}_{\mathrm{out}}$, the minimization problem (18) consists of solving the linear system (19) (ridge regression)

$$\left(\mathbf{R}\mathbf{R}^{\mathrm{T}}+\lambda\mathbb{I}_{N_r+1}\right)\mathbf{W}_{\mathrm{out}}^{\mathrm{T}}=\mathbf{R}\mathbf{B}^{\mathrm{T}}, \tag{19}$$

where $\mathbf{R} = \left[[\boldsymbol{r}(t_1);1] \mid \ldots \mid [\boldsymbol{r}(t_{N_{\mathrm{tr}}});1]\right]$ and $\mathbf{B} = \left[\boldsymbol{b}^{\mathrm{tr}}(t_1) \mid \ldots \mid \boldsymbol{b}^{\mathrm{tr}}(t_{N_{\mathrm{tr}}})\right]$. We regularize the ESN by adding Gaussian noise to each component of the input training data, i.e., $\tilde{\boldsymbol{B}}_q = \boldsymbol{B}_q + \mathcal{N}(\boldsymbol{0}, 0.03\mathrm{std}(\boldsymbol{B}_q))$, where $\mathrm{std}(\boldsymbol{B}_q)$ is the standard deviation in time of the qth row of $\mathbf{B}$ [50]. Because the input noise regularizes the ESN, we find that optimizing the Tikhonov parameter does not impact significantly the ESN performance, thus, we set the Tikhonov parameter to $\lambda = 10^{-16}$ for numerical stability.

4.3. Multi-parameter training dataset with data augmentation

The ESN training data must be carefully selected to obtain a robust estimator for the bias. Nóvoa and Magri [6] suggested a single-parameter approach where the ESN is trained on a timeseries of synthetic biases, created from an educated initial guess in the model state and parameters $\boldsymbol{\psi}^0 = (\boldsymbol{\phi}^0, \boldsymbol{\alpha}^0)$ from prior knowledge. This method provides accurate predictions of the bias if the actual bias of the model is similar to the training time series. However, we found that the ESN may diverge if the ensemble is initialized far from $\boldsymbol{\psi}^0$, or if the dynamics of the innovations are significantly different to the single-parameter training dataset. To overcome this, we propose a multi-parameter training approach with data augmentation, which provides more physical information to the ESN.

The multi-parameter method consists of creating a set of L timeseries of the bias data, whose initial guesses for the state and parameters are drawn from a uniform random distribution $\mathcal{U}\left(\boldsymbol{\psi}^0(1-\sigma_L), \boldsymbol{\psi}^0(1+\sigma_L)\right)$, where $\boldsymbol{\psi}^0$ is the mean, and $(1-\sigma_L, 1+\sigma_L)$ are the normalized ranges of the uniform distribution. Then, we increase the number of training datasets by adding $2L$ synthetic bias timeseries with 10% and 1% amplitude of the original data (i.e., we add signals with physical dynamics but with smaller amplitudes). In the machine learning jargon, this is referred to as *data augmentation* [51]. Data augmentation increases the robustness of the ESN for the network to infer the bias in lightly-biased models. (If the ESN is trained with large-norm bias datasets only, in the limiting case of an unbiased model, the ESN may not be able to recover a zero-norm bias.) The total number of training timeseries in the proposed training method is $3L$, thereby training the ESN consists of solving an augmented form of (19) such that

$$\left(\sum_{l=0}^{3L-1} \mathbf{R}_l \mathbf{R}_l^{\mathrm{T}} + \lambda\, \mathbb{I}_{N_r+1}\right) \mathbf{W}_{\mathrm{out}}^{\mathrm{T}} = \sum_{l=0}^{3L-1} \mathbf{R}_l \mathbf{B}_l^{\mathrm{T}}. \tag{20}$$

Although the computational cost of training the ESN increases with this approach, the summations in (20) can be parallelized to minimize the impact of data augmentation in the training cost. Importantly, the ESN is not trained with the 'true' bias, which is assumed to be *a priori* unknown. Instead, we train the ESN with a diverse set of training bias datasets (created from an educated initial guess of the system's dynamics), which provides the ESN with a variety of physical information, and allows the network to reconstruct different biases as the assimilation converges. Thus, with the multi-parameter training we aim to alleviate the limited adaptivity of machine-learned bias models in sequential assimilation, which was discussed by, e.g., [52].

4.4. Jacobian of the echo state network

In this section, we derive the Jacobian required in the r-EnKF Eqs. (15)–(16). For the ESN Eq. (17)

$$\mathbf{J} = \frac{\mathrm{d}\,\boldsymbol{b}^{\mathrm{f}}(t_{k+1})}{\mathrm{d}\,\mathbf{M}\boldsymbol{\psi}^{\mathrm{f}}(t_k)} = \frac{\mathrm{d}\,\boldsymbol{b}^{\mathrm{f}}(t_{k+1})}{\mathrm{d}\,\boldsymbol{i}(t_k)} \frac{\mathrm{d}\,\boldsymbol{i}(t_k)}{\mathrm{d}\,\mathbf{M}\boldsymbol{\psi}^{\mathrm{f}}(t_k)}. \tag{21}$$

The first term in the right-hand side of (21) is the Jacobian of the ESN, i.e., the gradient of the output with respect to the input; and the second term in (21) is the gradient of the input with respect to the forecast biased model estimate. As shown in Fig. 3, we perform an open-loop step at the analysis step, such that the input to the ESN is the mean analysis innovation, i.e.,

$$\boldsymbol{i}(t_k) = \boldsymbol{d}(t_k) - \mathbf{M}\overline{\boldsymbol{\psi}}^{\mathrm{a}}(t_k) \quad \text{and} \quad \mathbf{J} = -\frac{\mathrm{d}\,\boldsymbol{b}^{\mathrm{f}}(t_{k+1})}{\mathrm{d}\,\boldsymbol{i}(t_k)}.$$

In contrast, the ESN runs autonomously in between observations, in which the input to the reservoir is the ESN prediction at the previous time step (see Fig. 4b), thus, the second term in (21) is zero, and $\mathbf{J} = \mathbf{0}$ in closed-loop. Hence, we only require the open-loop ESN Jacobian for the r-EnKF. (Appendix B includes the derivation of the closed-loop form for completeness.)

To derive the Jacobian of the ESN, we split the input and output matrices into $\mathbf{W}_{\mathrm{in}} = \left[\mathbf{W}_{\mathrm{in}}^{(1)} \;\middle|\; \boldsymbol{W}_{\mathrm{in}}^{(2)}\right]$ and $\mathbf{W}_{\mathrm{out}} = \left[\mathbf{W}_{\mathrm{out}}^{(1)} \;\middle|\; \boldsymbol{W}_{\mathrm{out}}^{(2)}\right]$, where $\boldsymbol{W}_{\mathrm{in}}^{(2)}$ and $\boldsymbol{W}_{\mathrm{out}}^{(2)}$ are the last column vectors, which only operate on the constants 1 and δ_{r}, respectively. With this, we can rewrite (17) such that the prediction at t_{k+1} is a function of the quantities at t_k only, i.e.,

$$\boldsymbol{b}^{\mathrm{f}}(t_{k+1}) = \mathbf{W}_{\mathrm{out}}^{(1)} \tanh\left(\sigma_{\mathrm{in}} \mathbf{W}_{\mathrm{in}}^{(1)} \left(\boldsymbol{i}(t_k) \odot \boldsymbol{g}\right) + \sigma_{\mathrm{in}} \delta_{\mathrm{r}} \boldsymbol{W}_{\mathrm{in}}^{(2)} + \rho \mathbf{W} \boldsymbol{r}(t_k)\right) + \boldsymbol{W}_{\mathrm{out}}^{(2)}. \tag{22}$$

In open-loop, the reservoir state is not a function of the input because the input data is the innovation, rather than the prediction at the previous step. Therefore, at the analysis step (i.e., in open-loop), the gradient of the output $\boldsymbol{b}^{\mathrm{f}}(t_{k+1})$ with respect to the input $\boldsymbol{i}(t_k)$ yields

$$\mathbf{J} = -\frac{\mathrm{d}\,\boldsymbol{b}^{\mathrm{f}}(t_{k+1})}{\mathrm{d}\,\boldsymbol{i}(t_k)} = -\mathbf{W}_{\mathrm{out}}^{(1)} \left[\mathbf{T} \odot \left(\sigma_{\mathrm{in}} \mathbf{W}_{\mathrm{in}}^{(1)} \odot \mathbf{G}\right)\right], \tag{23}$$

where $\mathbf{G} = [\boldsymbol{g} \mid \ldots \mid \boldsymbol{g}]^{\mathrm{T}} \in \mathbb{R}^{N_q \times N_r}$; $\boldsymbol{1} \in \mathbb{R}^{N_r}$ is a vector of ones; and $\mathbf{T} = [\boldsymbol{T} \mid \ldots \mid \boldsymbol{T}] \in \mathbb{R}^{N_r \times N_q}$ with

$$\boldsymbol{T} = \boldsymbol{1} - \tanh^2\left(\sigma_{\mathrm{in}} \mathbf{W}_{\mathrm{in}}^{(1)} \left(\boldsymbol{b}(t_k) \odot \boldsymbol{g}\right) + \sigma_{\mathrm{in}} \delta_{\mathrm{r}} \mathbf{W}_{\mathrm{in}}^{(2)} + \rho \mathbf{W} \boldsymbol{r}(t_k)\right).$$

5. Regularized ensemble Kalman filter with an echo state network

The implementation of the regularized bias-aware DA algorithm with an ESN can be divided into three steps: (1) ESN training, (2) ESN and ensemble initialization, and (3) the core of the assimilation. The first step is the training of the echo state network. We use a multi-parameter approach and data augmentation to generate the training set. The pseudo-algorithm 1 summarizes this procedure, which is detailed in Section 4.2. We need observation data sampled at a frequency $\Delta t_{\mathrm{ESN}}^{-1}$ for an interval t_{tr}, which is problem-specific (refer to Appendix D for the values of the problem-specific parameters). We set the ESN time step larger than that of the physical model ($\Delta t_{\mathrm{ESN}} > \Delta t$), to reduce the computational cost added by the ESN. The time step should be, however, small enough to capture the dynamics of the bias.

Algorithm 1 Echo state network training

$\boldsymbol{d}(t_0 : t_{\mathrm{tr}}) \leftarrow$ GetObservations
for $l = [0 : L-1]$ **do** ▷ Multi-parameter data generation
  $\boldsymbol{\phi}_l, \boldsymbol{\alpha}_l = (\boldsymbol{\phi}^0, \boldsymbol{\alpha}^0)\mathcal{U}((1-\sigma_L), (1+\sigma_L))$
  $\mathbf{M}\boldsymbol{\psi}_l(t_0 : t_{\mathrm{tr}}) \leftarrow$ ForecastModel$(\boldsymbol{\phi}_l, \boldsymbol{\alpha}_l)$
  $\mathbf{B} \leftarrow$ Append$(\boldsymbol{d} - \mathbf{M}\boldsymbol{\psi}_l)$
$\mathbf{B} \leftarrow$ Concatenate$(\mathbf{B}, -0.1\mathbf{B}, 0.01\mathbf{B})$ ▷ Augment training data
$\mathbf{W}_{\mathrm{out}} \leftarrow$ TrainESN$(\mathbf{B})$ (20)

The second stage shown in pseudo-algorithm 2 is the ensemble and ESN initialization. As discussed in Section 2.3, the choice of the initial ensemble plays an important role on the assimilation because the analyses are confined to the range of the ensemble covariance, i.e., the analysis update results from a combination of the forecast ensemble members [40]. The ensemble size and spread required for convergence depend on the dimensionality of the dynamical system at hand. We initialized the ensemble state and parameters from a multivariate Gaussian distribution as shown in the pseudo-algorithm 2. The ensemble is forecast in time until statistical convergence. Then, we initialized the ESN with the mean innovations, which is also known as the *washout* phase [47]. The initialization must be performed before the start of the assimilation to update the reservoir state $\boldsymbol{r}$ with the physical bias. For a number of washout steps N_{wash} in open loop (Fig. 4a), we feed input data from observations to the ESN every Δt_{ESN}. The number of washout steps for an accurate initialization of the reservoir is problem-specific.

Algorithm 2 Ensemble and echo state network initialization

$\boldsymbol{\psi}_j^{\mathrm{f}}(t_0) \leftarrow$ InitializeEnsemble$(m, \mathcal{N}(\boldsymbol{\alpha}^0, \sigma_\alpha \mathbb{I}), \mathcal{N}(\boldsymbol{\phi}^0, \sigma_\phi \mathbb{I}))$
$t = t_{\mathrm{startDA}} - N_{\mathrm{wash}} \times \Delta t_{\mathrm{ESN}} - 2\Delta t_{\mathrm{d}}$
for $j = 0 : m-1$ **do** ▷ Parallel forecast
  $\boldsymbol{\psi}_j^{\mathrm{f}}(t) \leftarrow$ ForecastModel$(\boldsymbol{\psi}_j^{\mathrm{f}}(t_0))$ (7)
loop N_{wash} times: ▷ ESN washout
  $\mathbf{B}_{\mathrm{wash}} \leftarrow$ Append$(\boldsymbol{d}(t) - \mathbf{M}\overline{\boldsymbol{\psi}}^{\mathrm{f}}(t))$
  for $j = 0 : m-1$ **do**
    $\boldsymbol{\psi}_j^{\mathrm{f}}(t + \Delta t_{\mathrm{ESN}}) \leftarrow$ ForecastModel$(\boldsymbol{\psi}_j^{\mathrm{f}}(t))$ (7)
  $t = t + \Delta t_{\mathrm{ESN}}$
$\boldsymbol{b}^{\mathrm{f}}(t), \boldsymbol{r}(t) \leftarrow$ OpenLoopESN$(\mathbf{B}_{\mathrm{wash}})$ (17)

Algorithm 3 Regularized bias-aware data assimilation with an echo state network

for every observation $\boldsymbol{d}$ available **do**
  $\boldsymbol{d}(t) \leftarrow$ GetObservations
  $\mathbf{J} \leftarrow -$JacobianOpenLoop$(\boldsymbol{b}^{\mathrm{f}}(t), \boldsymbol{r}(t))$ (23)
  $\boldsymbol{\psi}_j^{\mathrm{a}} \leftarrow$ r-EnKF$(\boldsymbol{\psi}_j^{\mathrm{f}}(t), \boldsymbol{b}^{\mathrm{f}}(t), \boldsymbol{d}(t), \mathbf{J}, \gamma, \mathbf{C}_{bb})$ (15) ▷ Perform assimilation
  if all $\boldsymbol{\alpha}_j^{\mathrm{a}}$ are physical **then for** $j = 0, \ldots m-1$:
    $\boldsymbol{\psi}_j^{\mathrm{a}} = \overline{\boldsymbol{\psi}}^{\mathrm{a}} + 1.002\ (\boldsymbol{\psi}_j^{\mathrm{a}} - \overline{\boldsymbol{\psi}}^{\mathrm{a}})$
  else
    $\boldsymbol{\psi}_j^{\mathrm{a}} = \overline{\boldsymbol{\psi}}^{\mathrm{f}}(t) + 1.05\ (\boldsymbol{\psi}_j^{\mathrm{f}}(t) - \overline{\boldsymbol{\psi}}^{\mathrm{f}}(t))$
  for $j = 0 : m-1$ **do** ▷ Parallel forecast
    $\boldsymbol{\psi}_j^{\mathrm{f}}(t : t + \Delta t_{\mathrm{d}}) \leftarrow$ ForecastEnsemble$(\boldsymbol{\psi}_j^{\mathrm{a}})$ (7)
  $\boldsymbol{b}^{\mathrm{f}}(t : t + \Delta t_{\mathrm{d}}), \boldsymbol{r}(t + \Delta t_{\mathrm{d}}) \leftarrow$ ClosedLoopESN$(\boldsymbol{d}(t) - \mathbf{M}\overline{\boldsymbol{\psi}}^{\mathrm{a}}, \boldsymbol{r}(t))$ (17)
  $t = t + \Delta t_{\mathrm{d}}$

The final step is the core stage of the assimilation, which is illustrated in Fig. 3 and it is detailed in pseudo-algorithm 3. The forecast model and the ESN are iteratively evolved in between observations for a time Δt_{d}. Every Δt_{d}, we (i) compute the open-loop

Table 1
Terminology of the RMS and MAE metrics and their corresponding mathematical expression, and the time-windows in which the errors are computed.

Term	Metric	Term	Time-window
Biased RMS	$\mathrm{RMS}(\boldsymbol{d}^{\mathrm{t}}, \mathbf{M}\boldsymbol{\psi})$	pre-DA	t_{err} before the assimilation
Unbiased RMS	$\mathrm{RMS}(\boldsymbol{d}^{\mathrm{t}}, \boldsymbol{y})$	DA	last t_{err} of the assimilation
True biased RMS	$\mathrm{RMS}(\boldsymbol{d}^{\mathrm{t}}, \mathbf{M}\boldsymbol{\psi}^{\mathrm{t}})$	post-DA	t_{err} after removing the filter
Biased MAE	$\mathrm{MAE}(\boldsymbol{d}^{\mathrm{t}}, \mathbf{M}\boldsymbol{\psi})$		
Unbiased MAE	$\mathrm{MAE}(\boldsymbol{d}^{\mathrm{t}}, \boldsymbol{y})$		

Jacobian of the ESN (23); and (ii) perform the analysis step by combining the ensemble forecast, the bias estimate, the observations, and the Jacobian, as well as the bias regularization factor, γ, and matrix weighting the bias-norm $\mathbf{C}_{bb}$. Because we define the bias in the observable space (see Section 2.1), the observation error covariance matrix $\mathbf{C}_{dd}$ is an appropriate weight in the bias norm. Thus, we set $\mathbf{C}_{bb} = \mathbf{C}_{dd}$. Through this procedure, the r-EnKF (15) obtains the optimal combination between the biased forecast, the bias estimate, and the observations. The analysis indirectly updates the biased ensemble (as illustrated in Fig. 1). Because sequential filters do not enforce physical boundaries on the estimated parameters, we apply the reject-inflate strategy proposed by Nóvoa and Magri [6]. If the resulting parameters are physical, we keep the analysis and apply a multiplicative inflation to the ensemble, motivated by da Silva and Colonius [53]. Alternatively, if the recovered parameters are not physical (e.g. positive time delays), we reject the analysis, and the algorithm returns the forecast ensemble with a stronger inflation with a 1.05 factor. The inflation consists of multiplying the ensemble covariance by an inflation factor close to 1 (1.002 in our case [6]). This is common practice when using small ensemble sizes, which can systematically underestimate the error covariances, thereby potentially leading to covariance collapse [2,54]. Finally, the analysis innovation, i.e., $\boldsymbol{d} - \mathbf{M}\overline{\boldsymbol{\psi}}^{\mathrm{a}}$ is used to re-initialize the ESN with a single open-loop step as illustrated in Fig. 3. The ESN is then evolved in time in closed-loop in parallel to the ensemble forecast, in which each ensemble member is evolved individually (as indicated by the stacking of m models in Fig. 3).

6. Numerical test cases

In this work, we showcase the r-EnKF to predict dynamics given by nonlinear oscillators. Nonlinear oscillators are employed in variety of disciplines, ranging from quantum physics [55] to cardiovascular studies [56], and the synchronization property of coupled oscillators is exploited in a variety of applications such as in laser physics, to increase power output (e.g. [57]), or in nano-mechanics, to remove frequency differences arising from imperfections in fabrication [58]. Further, nonlinear oscillators are extensively used to model vibrations and acoustics (e.g. [59]). Within the field of acoustics, thermoacoustic oscillations, which can also be modelled with nonlinearly coupled oscillators, are a key phenomenon in the fields of aviation and power generation [60–64]. Thermoacoustics are high-amplitude oscillations that arise when an unsteady heat source interacts with the acoustics and hydrodynamics of a system. If the acoustic waves and the heat release are sufficiently in phase, the acoustic energy in the system increases, thus, a thermoacoustic oscillation arises. We showcase the r-EnKF on problems that are relevant to thermoacoustics because they offer computational challenges, such as nonlinearities and time-delays, which are common to a variety of nonlinear systems in engineering and science.

We test the r-EnKF algorithm on two prototypical systems that have been employed in fundamental studies of thermoacoustic instabilities: (1) the van der Pol oscillator, which is a nonlinearly-forced oscillator [65]; and (2) a nonlinearly time-delayed model, which is a set of nonlinearly coupled oscillators (e.g. [66]). The problem-specific parameters of the ESN and DA framework, as well as the model parameters, are reported in Appendix D.

6.1. Error metrics

We use two error metrics to analyse the results. First, we compute the time-evolution of the mean absolute error (MAE) of the biased and unbiased signals, relative to the maximum amplitude of the truth. For two quantities $\boldsymbol{w}, \boldsymbol{z}$ with N_q dimensions, the MAE is defined as

$$\mathrm{MAE}(\boldsymbol{w}, \boldsymbol{z}) = \frac{1}{N_{\mathrm{err}}} \sum_{k=0}^{N_{\mathrm{err}}-1} \sum_{q=0}^{N_q-1} \frac{\left|\boldsymbol{w}_q(t_k) - \boldsymbol{z}_q(t_k)\right|}{\max\left(\boldsymbol{d}_q^{\mathrm{t}}\right)}. \tag{24}$$

The MAE is computed at every time interval t_{err} (which is problem-specific, see Appendix D) to recover a smooth time-evolution of the metric, that is, we analyse the moving-average of the absolute error. Second, we calculate the normalized root mean square error (RMS) as

$$\mathrm{RMS}(\boldsymbol{w}, \boldsymbol{z}) = \sqrt{\frac{\sum_q \sum_k \left(\boldsymbol{w}_q(t_k) - \boldsymbol{z}_q(t_k)\right)^2}{\sum_q \sum_k \left(\boldsymbol{w}_q(t_k)\right)^2}}, \quad \text{for} \quad \begin{array}{l} q = 0, \dots, N_q - 1, \\ k = 0, \dots, N_{\mathrm{err}} - 1 \end{array} \tag{25}$$

Table 1 summarizes the error metrics and their corresponding terminologies, which we use in Sections 6.2 and 6.3 to analyse the results.

6.2. Van der Pol oscillator

The van der Pol model is a forced oscillator with linear damping, whose time evolution is governed by the second-order differential equation

$$\ddot{\eta} + \omega^2 \eta = \dot{q} - \zeta \dot{\eta}, \tag{26}$$

where ω is the angular oscillating frequency, ζ is the damping coefficient, and $\dot{q}$ is a forcing term. When applied to thermoacoustics, η represents the acoustic velocity, and $\dot{q}$ is the heat release rate. Noiray [67] proposed a nonlinear arc-tangent heat release law to model a thermoacoustic limit cycle from experimental data, $\dot{q} = \beta\dot{\eta}\left(1 - \kappa\eta^2(\beta + \kappa\eta^2)^{-1}\right)$, where κ is the nonlinearity coefficient, and β is the forcing strength. Using this heat release law, Eq. (26) can be written as a system of ordinary differential equations as

$$\dot{\eta} = \mu \tag{27a}$$

$$\dot{\mu} = \omega^2 \mu + \dot{\eta}\left(\beta - \zeta - \frac{\beta\kappa\eta^2}{\beta + \kappa\eta^2}\right). \tag{27b}$$

In reference to Section 2.2, the state variables of the system are $\boldsymbol{\phi} = [\mu; \eta]$ and the physical parameters to infer are $\boldsymbol{\alpha} = [\zeta; \beta; \kappa]$. Hence, the augmented state vector is $\boldsymbol{\psi} = [\mu; \eta; \zeta; \beta; \kappa; \eta]$, where the biased model prediction is the acoustic velocity, $\mathbf{M}\boldsymbol{\psi} = \eta$.

To showcase the r-EnKF, we create the reference data, i.e., the *truth*, by prescribing a nonlinear additive bias to the observables created from a set of parameters $\boldsymbol{\alpha}^{\mathrm{t}}$ (reported in Appendix D). In this case, we define

$$d^{\mathrm{t}} = \eta^{\mathrm{t}} + \cos\eta^{\mathrm{t}}. \tag{28}$$

Therefore, the 'true' bias of the model is $b^{\mathrm{t}} = \cos\eta^{\mathrm{t}}$. The observations are then created from the truth by adding Gaussian noise with standard deviation $\sigma_{\mathrm{d}} = 0.01$, which is normalized by $\langle |d^t| \rangle$, i.e., the time-average of the absolute value of the true amplitude.

Further forms of bias and types of observation noise are studied in the time-delayed model in Section 6.3, which is more general. We use the van der Pol model to (1) analyse the effect of the multi-parameter training with data augmentation (Section 6.2.1), and (2) to perform a parametric analysis on the bias regularization factor, γ (Section 6.2.2).

6.2.1. Analysis of the multi-parameter training approach with data augmentation

We start our analysis by addressing a question that we asked in this work: "what is a 'good' unbiased analysis?". We do so by showing the effects of data augmentation and the multi-parameter training approach (see Section 4.2) on the biased and unbiased RMS in the van der Pol oscillator. Fig. 5 shows the truth and the generated multi-parameter observable timeseries for three different L-values (Fig. 5I), as well as their corresponding bias datasets used to train the ESN in the simulations (Fig. 5II). The bias training datasets are the innovations $b_l = d - \mathbf{M}\boldsymbol{\psi}_l$ for $l = 1, \ldots, L$ computed over a training time $t_{\mathrm{tr}} = 1.0$ s, sampled every $\Delta t_{\mathrm{ESN}} = 5 \cdot 10^{-4}$ s. The multi-parameter training ($L > 1$) cases shown in Fig. 5b,c provide rich dynamical information to the ESN during training, which makes the network robust to sudden transient changes in the state and parameters during assimilation (see Section 6.2.2). The effect of the multi-parameter training with data augmentation is analysed in Fig. 6, which shows the biased and unbiased RMS errors (see Table 1) computed after the filter has been removed in four cases: a single-parameter training (a) without data augmentation, and (b) with data augmentation; and two multi-parameter training cases with data augmentation with (c) $L = 10$, and (d) $L = 50$. The L-datasets used for training the ESN correspond to those in Fig. 5. The initial ensemble is identical in each simulation, with an initial biased RMS of 0.51 (pre-DA black line in Fig. 6). The grey line indicates the true biased RMS, as defined in Table 1. Fig. 6 shows that if the bias is not penalized (violet bars in Fig. 6), i.e., when $\gamma = 0$ in (12), the four cases provide solutions with small unbiased RMS (smaller than the true biased RMS); however, their biased RMS remains at a similar level to the pre-DA RMS. This undesired scenario, in which the bias norm does not decrease with the assimilation, is illustrated in Fig. 2b. By increasing γ, we promote a small-norm bias analysis. Fig. 6a shows that the simplest training approach recovers a small biased RMS at $\gamma = 10$ (blue), however, the ESN does not recover accurately the bias of the solution (see the increased RMS of the thinner blue bar). Moreover, both the filter and ESN diverge if the bias regularization factor is increased to $\gamma = 20$. Introducing data augmentation (Fig. 6b), improves the robustness of the ESN, as seen by the small unbiased RMS; however, the recovered solutions in the three γ cases have a larger-norm bias than the initial predictions, thus, the unbiased analysis is not 'good' (this corresponds to the scenario depicted in Fig. 2c). The 'good' unbiased analysis (cf. Fig. 2d) is recovered when we introduce the multi-parameter training. Comparing the multi-parameter training cases in Fig. 5b,c, we see that increasing the size of the training dataset to $L = 50$ improves the filter and ESN performance because the biased RMS are at true biased RMS levels, the unbiased RMS remains small, and the ensemble uncertainty (represented by the error bars) is small for both $\gamma = 10$ and 20. The results improve with L because more training data mean more information given to the ESN (as shown in Fig. 5). Therefore, increasing the information fed to the ESN makes the network robust to sudden transient changes in the state and parameter during the assimilation.

6.2.2. Effects of the training size and the bias regularization factor

Fig. 7 shows the effect of the bias regularization factor γ on the time evolution of the simulations with $L = 50$ in Fig. 6. The bias and pressure timeseries (Fig. 7I,II) show that the ESN washout ($1.98\ \mathrm{s} < t < 2\ \mathrm{s}$) gives an accurate unbiased signal initialization (as seen by the unbiased estimate overlapping the truth, and the ESN estimate matching the innovations). Consequently, if $\gamma = 0$, the cost function (10) is small regardless of the bias norm, and the innovations and parameters do not change significantly throughout assimilation. In contrast, if $\gamma > 0$, the filter provides an analysis with a smaller-norm bias (as seen by the reduction in innovations, and

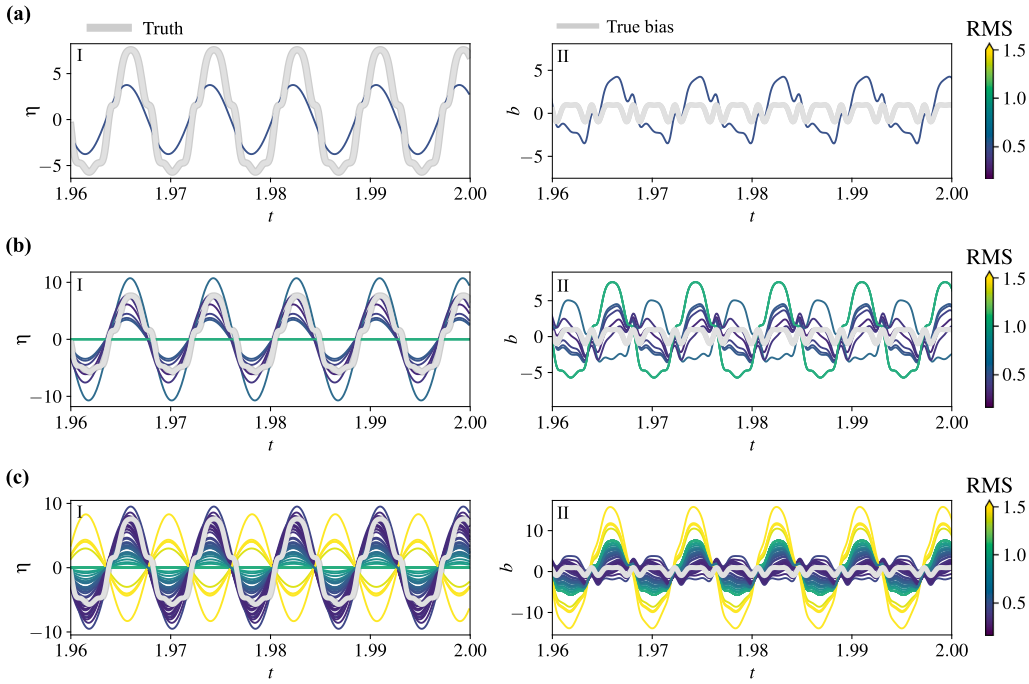

Fig. 5. Van der Pol model. Comparison of the ESN training datasets with (a) single-parameter training, i.e., $L = 1$, and multi-parameter training with (b) $L = 10$, and (c) $L = 50$. (I) Timeseries of the biased observables generated with a multi-parameter approach $\mathbf{M}\boldsymbol{\psi}_l = \eta_l$, for $l = 1, \dots, L$ (colourmap), compared to the unbiased truth d^{t} (thick grey line). (II) The corresponding b_l training datasets for $l = 1, \dots, L$ (colourmap), compared to the true bias $b^{\mathrm{t}} = \cos \eta^{\mathrm{t}}$ (28), (thick grey line). The colourmap shows the biased RMS error $\mathrm{RMS}(\boldsymbol{d}^{\mathrm{t}}, \mathbf{M}\boldsymbol{\psi}_l)$ in the shown time-window. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

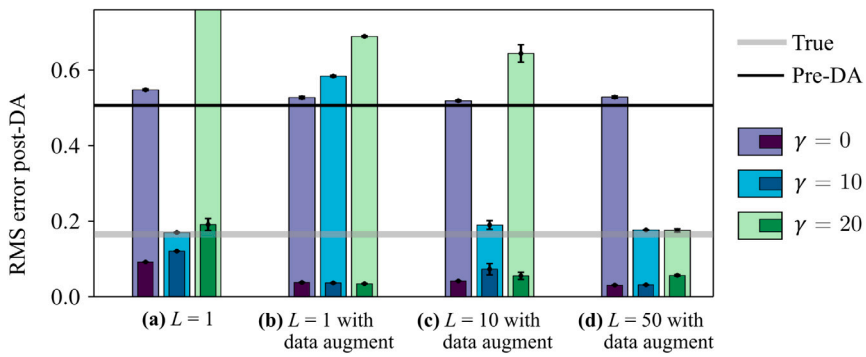

Fig. 6. Van der Pol model. Analysis of the multi-parameter training with data augmentation, and the bias regularization hyperparameter. Comparison between the post-DA RMS errors using (a) single-parameter training, (b) single-parameter training with data augmentation, and multiple-parameter training with data augmentation with (c) $L = 10$ and (d) $L = 50$; with bias regularization factors $\gamma = 0$ (violet), $\gamma = 10$ (blue), and $\gamma = 20$ (green). The thicker bars show the biased error, $\mathrm{RMS}(\boldsymbol{d}^{\mathrm{t}}, \mathbf{M}\overline{\boldsymbol{\psi}})$, and the overlapping thinner bars show the unbiased error, $\mathrm{RMS}(\boldsymbol{d}^{\mathrm{t}}, \overline{\boldsymbol{y}})$. The error bars indicate the spread of the RMS errors in the ensemble. The horizontal lines indicate the true biased RMS, $\mathrm{RMS}(\boldsymbol{d}^{\mathrm{t}}, \mathbf{M}\boldsymbol{\psi}^{\mathrm{t}})$ (grey), and the biased RMS before applying the filter (black). Solutions (c) with $\gamma = 10$, and (d) with $\gamma = 10, 20$ are 'good' unbiased analysis because the unbiased RMS is small and the biased RMS is at true biased RMS levels. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the overlap of the biased estimate with the observations in Fig. 7b,c). As discussed in Section 3, γ directly influences the adaptability of the analysis to the bias norm, which can be seen in Fig. 7III: the instantaneous changes in the model parameters at $\gamma = 20$ are larger than at $\gamma = 10$. (For visualization, the parameters are normalized by their initial ensemble mean, which is not necessarily the solution to the problem, i.e., we do not expect the parameters to converge to 1.) However, the parameter convergence can be hindered if the instantaneous changes are too large.

Therefore, tuning the bias regularization factor γ is key to recovering small-norm bias analyses with the r-EnKF. We perform a parametric analysis on γ, for the 3 sizes of training datasets L. Fig. 8 shows the error metrics of the parametric analysis, as defined in Table 1. To complete the results of state, parameter and bias estimation, we include in Appendix E the convergence of the parameters for the same simulations. With $L = 1$, the filter performance is almost unaffected regardless of γ, as shown by the small variation in

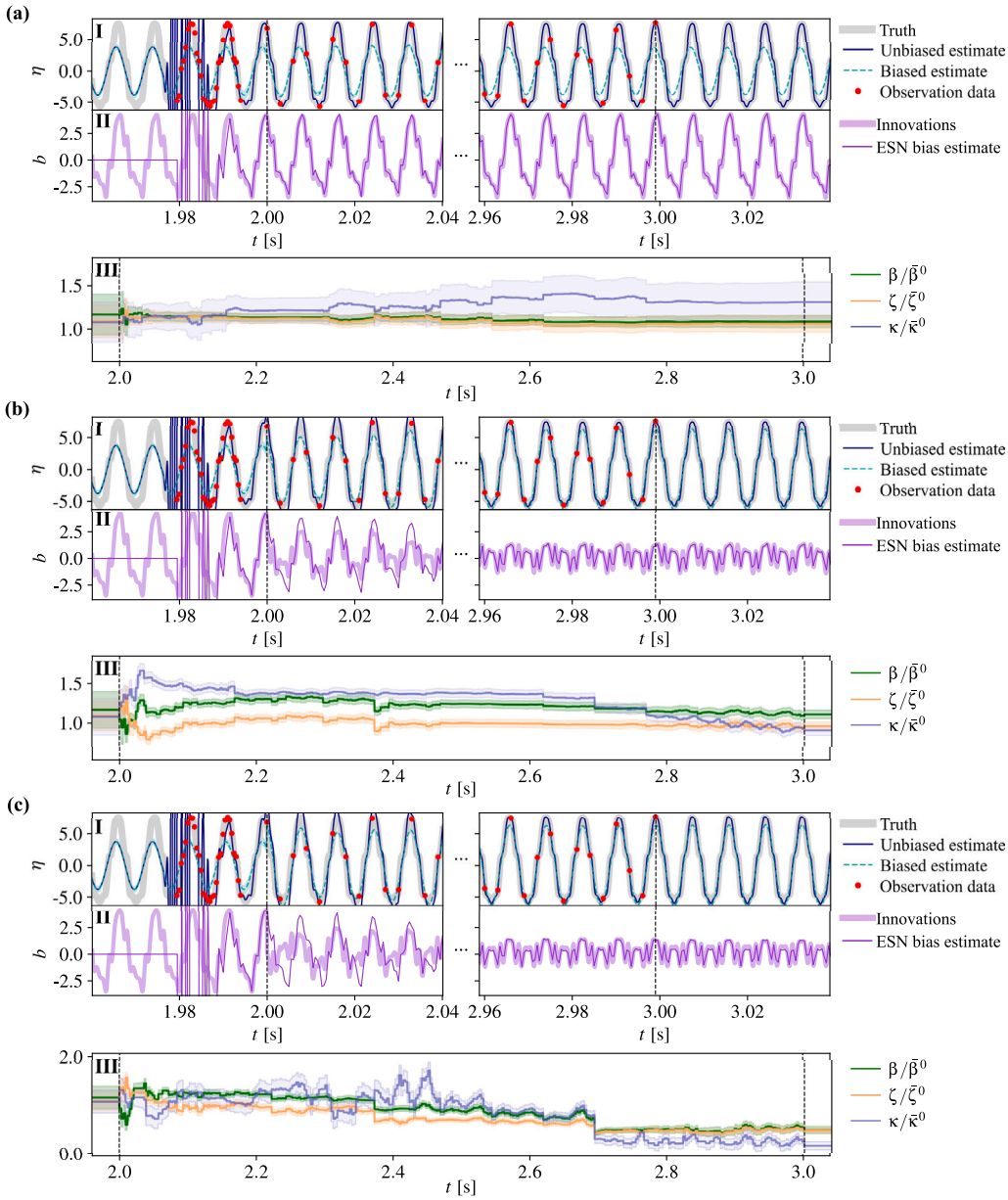

Fig. 7. Van der Pol model. Analysis on the bias regularization factor in the assimilation, (a) $\gamma = 0$, (b) $\gamma = 10$, (c) $\gamma = 20$, with $L = 50$. Time evolution at the start and end of assimilation of (I) the true acoustic velocity (thick grey), unbiased estimate (navy), and biased estimate with its uncertainty (dashed teal); and (II) the innovations (thick light orchid) and ESN prediction (thin dark orchid). (III) Evolution of the inferred parameters and their uncertainty, normalized by their initial ensemble mean values. The assimilation window is indicated by the vertical dashed lines, and the red circles show the observation data. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the RMS and MAE throughout the assimilation in Fig. 8a. Whereas with the designed multi-parameter training (Fig. 8b,c), the filter is versatile and recovers 'good' unbiased analysis, i.e., the filter converges to solutions with biased RMS at true biased RSM levels, and with small unbiased RMS values. Moreover, we see that at the ESN is more robust at $L = 50$, because the prediction remains accurate for a wider range of γ, and the unbiased RMS remains unchanged after the filter is removed. It is important to note that the ESN is only trained once for each of the L-values, i.e., for each row in Fig. 8. Thus, even though the weights of the ESN are static, the network is adaptive enough to recover bias signals of different magnitudes and dynamics.

Overall, the results shown in Sections 6.2.1 and 6.2.2 indicate that (i) providing a rich variety of training datasets to the ESN (i.e. providing $L > 1$) and performing data augmentation is essential to give the framework robustness and versatility; (ii) a proper tuning of the bias regularization factor is key to recovering a small-norm bias solutions with the r-EnKF; and (iii) the ESN is more robust and stable to wider ranges of the bias regularization factor as L increases.

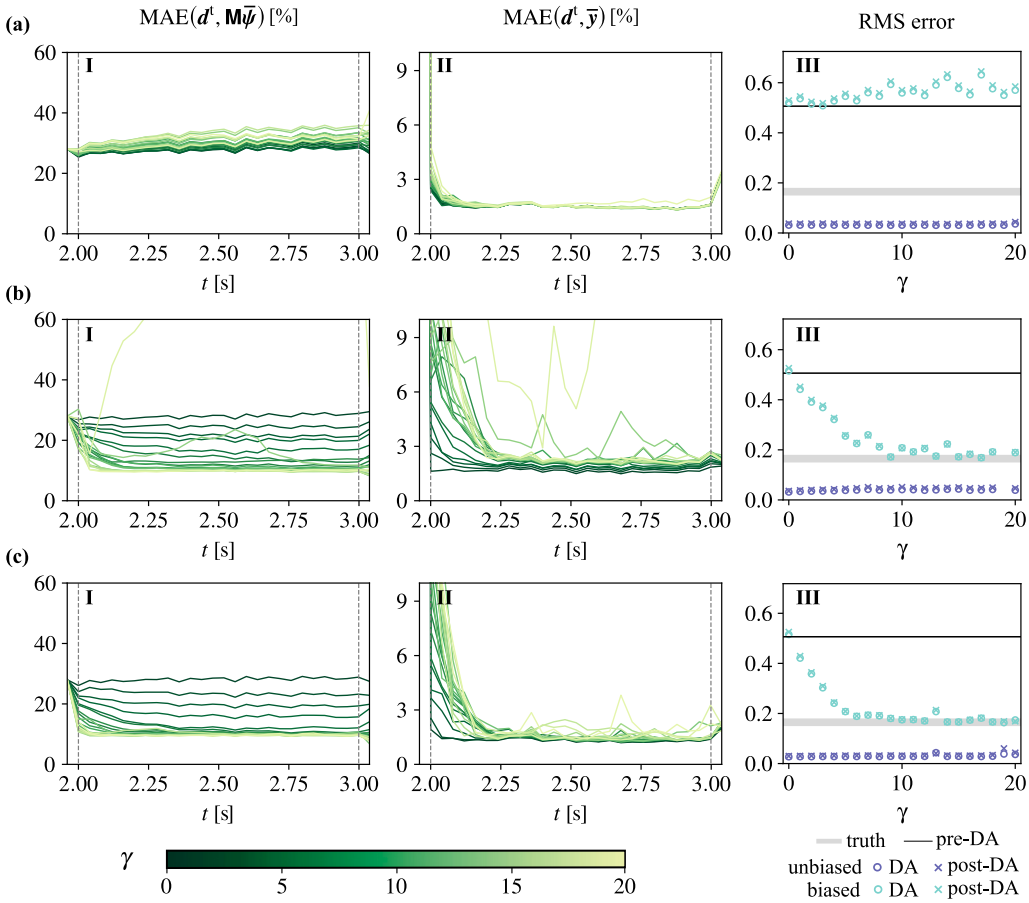

Fig. 8. Van der Pol model. Parametric analysis of the bias regularization factor γ for different numbers of training sets (a) $L = 1$, (b) $L = 10$, and (c) $L = 50$. Evolution of the (I) biased and (II) unbiased MAE; and (III) RMS errors as defined in Table 1: biased RMS (light blue) and unbiased RMS (dark blue) at convergence (circles) and after assimilation (crosses), true biased RMS (thick grey line), and biased RMS before assimilation (thin black line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

6.3. Nonlinearly coupled time-delayed oscillators

In this section, we apply the r-EnKF to a set of coupled oscillators with a nonlinearly time-delayed forcing and nonlinear damping. We use a low-order time-delayed nonlinear thermoacoustic model, which is widely used in fundamental studies of thermoacoustic oscillations (e.g. [68–70]). We formulate the time-delayed problem in the Markovian framework proposed by Huhn and Magri [71]. (The reader is referred to Appendix C for further details.) The discretized system of equations governing the acoustics is

$$\frac{\mathrm{d}\eta_j}{\mathrm{d}t} = A_1 \mu_j \tag{29a}$$

$$\frac{\mathrm{d}\mu_j}{\mathrm{d}t} = -A_2 \eta_j - A_3 \dot{q} \sin\left(A_4 x_\mathrm{h}\right) - A_5 \zeta_j \mu_j, \qquad \text{for} \quad j = 0, \dots, N_m - 1 \tag{29b}$$

$$\frac{\mathrm{d}v_i}{\mathrm{d}t} = 2\tau_v \sum_k \mathsf{D}_{ik} v_k, \qquad \text{for} \quad i = 0, \dots, N_c - 1 \tag{29c}$$

where $\boldsymbol{\eta}$ and $\boldsymbol{\mu}$ are the acoustic velocity and pressure modes resulting from the Galerkin discretization of the acoustic velocity and pressure into N_m acoustic modes; ζ_j is the damping, defined with the modal form $\zeta_j = C_1 j^2 + C_2 \sqrt{j}$ [72]; $\dot{q}$ is the perturbation heat release rate, which is modelled with the time-delayed square-root model [68]

$$\dot{q}(t) = A_6 \beta \left(\sqrt{\left| \frac{1}{3} + A_7 u\left(x_\mathrm{h}, t - \tau\right) \right|} - \sqrt{\frac{1}{3}} \right), \tag{30}$$

with β the heat release strength, and τ the time-delay such that $u\left(x_\mathrm{h}, t - \tau\right)$ is the acoustic velocity at the heat source location at the delayed time; and $\boldsymbol{v}$ are the spectral Chebyshev modes, which result from discretizing the advection equation modelling the history of $u\left(x_\mathrm{h}\right)$ into $N_c + 1$ collocation points with the Chebyshev polynomial matrix $\mathbf{D}$ [73]; and the constants $A_1, \dots, A_7$ are

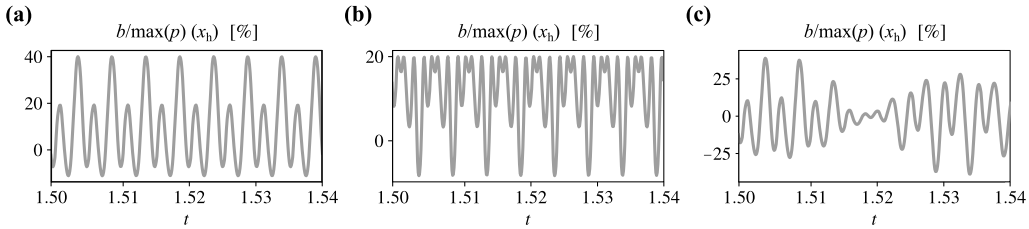

Fig. 9. Nonlinear time-delayed model. Synthetic bias signals at the heat source location ($x_{\mathrm{h}} = 0.2$ m), normalized by the maximum amplitude of the true pressure at x_{h}: (a) linear bias, (b) nonlinear bias, and (c) time-function bias.

specified in Appendix C. The advection equation transforms the time delayed problem into an initial value problem by modelling a dummy variable w travelling in a non-dimensional domain $X \in [0, 1]$ with velocity τ_v^{-1}, with $\tau_v \geq \tau$. The Chebyshev modes are defined such that $v_0(t) = w(X = 0, t) = u(x_{\mathrm{h}}, t)$, and $v_{N_c}(t) = w(X = 1, t) = u(x_{\mathrm{h}}, t - \tau_v)$. Thus, the time-delayed velocity in (30), $u\left(x_{\mathrm{h}}, t - \tau\right) = w(X = \tau/\tau_v, t)$, which is computed by interpolation between Chebyshev modes.

The thermoacoustic parameters to infer are $\boldsymbol{\alpha} = [\beta; \tau]$; the state variables are $\boldsymbol{\phi} = [\boldsymbol{\eta}; \boldsymbol{\mu}; \boldsymbol{\nu}]$; and the observables in the nonlinear time-delayed system are acoustic pressure measurements at $\boldsymbol{x} \in \mathbb{R}^{N_q}$ locations

$$p_q(x_q, t) = -\sum_{j=0}^{N_m-1} \mu_j(t) \sin\left(\frac{\omega_j}{\bar{c}} x_q\right), \quad \text{for} \quad q = 0, \ldots, N_q - 1. \tag{31}$$

With this, augmented the state vector is $\boldsymbol{\psi} = [\boldsymbol{\eta}; \boldsymbol{\mu}; \boldsymbol{\nu}; \beta; \tau; \boldsymbol{p}(\boldsymbol{x})]$, and the biased model prediction is $\mathbf{M}\boldsymbol{\psi} = \boldsymbol{p}(\boldsymbol{x})$. We use $N_m = 10$ for an appropriate representation of the acoustics [71], and $N_q = 6$ equidistant measurement locations, with $x_0 = x_{\mathrm{h}}$ (i.e., the first measurement is at the heat source location), which is consistent with experimental works [74]. Because the time-delay τ is to be inferred and its value changes throughout the assimilation, we define $\tau_v = 0.01$ s as the upper bound of the time-delay to give versatility to the filter (e.g., [75]). We find that $N_c = 50$ collocation points are required to avoid numerical dissipation in the advection equation.

The true pressure is defined as $\boldsymbol{d}^{\mathrm{t}} = \boldsymbol{p}^{\mathrm{t}} + \boldsymbol{b}^{\mathrm{t}}$, where $\boldsymbol{b}^{\mathrm{t}}$ is the pre-defined bias. In this test case, we analyse three different forms of bias of increasing complexity: (i) a linear function of the acoustic pressure with an amplitude shift (32)(a), which could correspond to an offset on the microphones calibration in a real experiment; (ii) a periodic transformation of the pressure (32)(b), which can model, for instance, unclosed higher order acoustic modes (i.e., $N_m > 10$); and (iii) an explicit time-dependent bias (32)(c), which could arise in reality due to unmodelled physics in the low-order model (e.g., entropy waves, or mean flow effects); respectively

$$\boldsymbol{b}^{\mathrm{t}} = \begin{cases} a_1\, \boldsymbol{p}^{\mathrm{t}}(\boldsymbol{x}) + a_2 \max(\boldsymbol{p}^{\mathrm{t}}(x_{\mathrm{h}})). & \text{(a)} \\ a_3 \max(\boldsymbol{p}^{\mathrm{t}}(x_{\mathrm{h}})) \cos\left(\dfrac{a_4\, \boldsymbol{p}^{\mathrm{t}}(\boldsymbol{x})}{\max(\boldsymbol{p}^{\mathrm{t}}(x_{\mathrm{h}}))}\right). & \text{(b)} \\ a_5\, \boldsymbol{p}^{\mathrm{t}}(\boldsymbol{x}) \sin\left(a_6 \pi t\right)^2. & \text{(c)} \end{cases} \tag{32}$$

We set the constant values in (9) to be $a_1 = 0.3, a_2 = 0.1, a_3 = 0.2, a_4 = 2, a_5 = 0.4$, and $a_6 = 2$. Fig. 9 shows the three types of bias in (9) at the first measurement location (x_{h}). In the linear case (Fig. 9a) the bias (32)(a) retains the dynamics of the original signal (a 2-period limit cycle), but with a different mean and amplitude. The periodic dynamics of the original signal are also retained with the periodic bias forcing (32)(b), but with a different frequency spectrum (Fig. 9b). In contrast, Fig. 9c shows how the time-dependent bias (32)(c) is not a limit cycle, but a quasi-periodic bias because, by adding a time-dependent sinusoidal, we are including a new incommensurate frequency to the dynamics.

Finally, to create the observations, we add to each component of the truth, Gaussian noise with a standard deviation of $\sigma_{\mathrm{d}} = 0.01$, which is normalized by the time-average of the mean absolute amplitude. We analyse the robustness of the r-EnKF for increasing values of σ_{d}, as well as the impact of adding white, pink and brown noises of different strengths in Section 6.3.3.

6.3.1. *Analysis of different biases: linear, nonlinear, and time-dependent*

In this section, we discuss the effect of the type of bias dynamics on the assimilation. We start from the linear and nonlinear bias cases. The number of degrees of freedom in the nonlinearly coupled oscillators is larger than in the van der Pol case. Thus, as discussed in Section 2, the initialization of the ensemble (which is given by the ensemble size, m, and the initial ensemble spread) is key to the performance of the filter because the analyses are confined to the range of the forecast ensemble error covariance matrix. We perform the analysis shown in this section for three ensemble sizes. We find that $m = 50$ provides a good trade-off between filter performance and computational cost (results shown in Appendix F). Fig. 10 shows, for the linear (32)(a) and nonlinear (32)(b) biases, the sum of the biased and unbiased RMS (see Table 1) after the filter is removed. The figure shows that bias regularization factors $\gamma \gtrsim 5$ and $\gamma \lesssim 1$ tend to provide inaccurate solutions or even divergence. The areas in dashed dark red show solutions where the combined bias and unbiased RMS is larger than the true signal magnitude. The error is smaller in the linear bias case (Fig. 10a), which is to be expected due to the increased difficulty arising from the nonlinear bias form (Fig. 10b). We define optimal $(L^\star, \gamma^\star)$ pairs as the minimum values of the combined RMS error ($\mathrm{RMS}_b + \mathrm{RMS}_u$) (i.e., the global minima in Fig. 11 are the post-DA

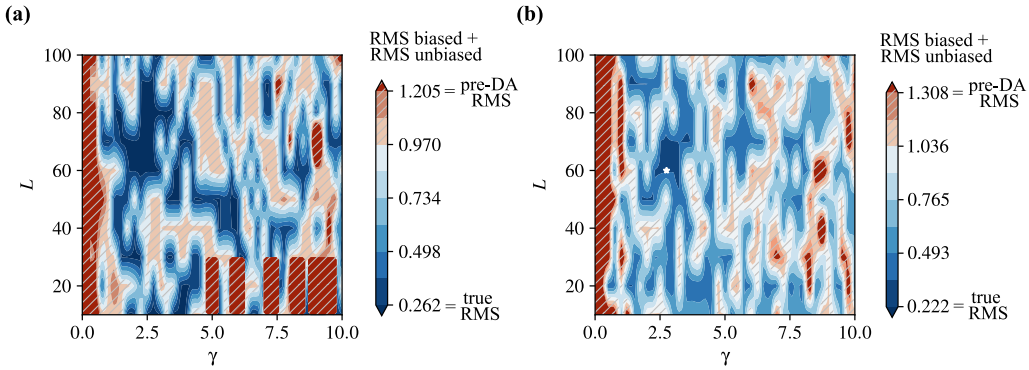

Fig. 10. Nonlinear time-delayed model. Combined bias and unbiased RMS errors after assimilation with (a) linear, and (b) nonlinear biases. Contours for varying training set size, L, and bias regularization factor, γ. The colourmap shows the sum of the bias and unbiased RMS errors, and the minimum combined RMS point is indicated with a white star. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
Nonlinear time-delayed model RMS errors, for the linear, nonlinear, and time-dependent bias cases. Comparison between the true, RMS($\boldsymbol{d}^{\mathrm{t}}, \mathbf{M}\boldsymbol{\psi}^{\mathrm{t}}$), biased RMS($\boldsymbol{d}^{\mathrm{t}}, \mathbf{M}\overline{\boldsymbol{\psi}}$), and unbiased RMS($\boldsymbol{d}^{\mathrm{t}}, \overline{\boldsymbol{y}}$) at assimilation convergence (DA) and after the filter is removed (post-DA).

Bias type	True biased RMS	DA RMS			post-DA RMS		
		$L^\star, \gamma^\star$	Biased	Unbiased	$L^\star, \gamma^\star$	Biased	Unbiased
Linear	0.2623	10, 3.50	0.1761	0.0244	100, 1.75	0.1817	0.0157
Nonlinear	0.2217	60, 2.75	0.2303	0.0799	60, 2.75	0.2279	0.0792
Time-func.	0.2385	30, 0.50	0.2860	0.0590	10, 1.25	0.2534	0.4434

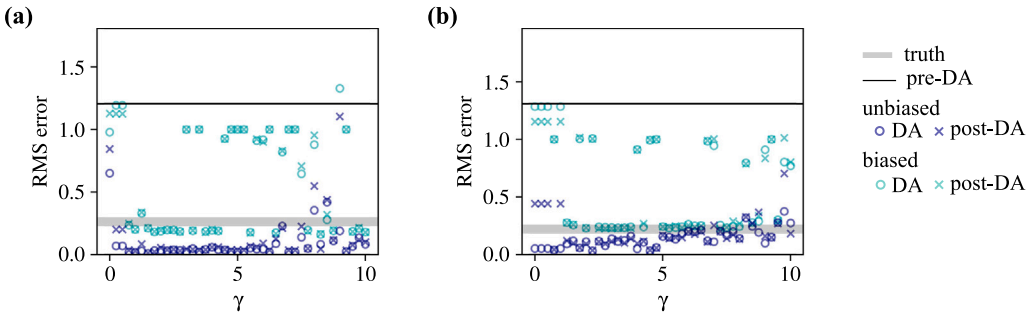

Fig. 11. Nonlinear time-delayed model. RMS errors (as defined in Table 1) with varying bias regularization factor γ, for the (a) linear and (b) nonlinear bias cases, with fixed $L = 70$ and $m = 50$. RMS errors of the biased (light blue) and unbiased (dark blue) solutions at convergence (circles) and after assimilation (crosses), true biased RMS (thick grey line), and biased RMS of the initial ensemble (black line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

($L^\star, \gamma^\star$)). Table 2 reports for each of the bias cases, the true biased RMS, and the biased and unbiased RMS of the optimal solutions at convergence, and after the filter is removed.

Fig. 10 shows how for $L = 70$ (i.e., training the ESN with 70 signals), the r-EnKF provides accurate solutions for a large range of γ (as seen by the blue areas concentrated in this region). We show in Fig. 11 the detailed RMS errors and the inferred parameters at $L = 70$, for the linear (Fig. 11a) and nonlinear (Fig. 11b) biases. The filter sensitivity to γ is higher than that of the van der Pol model in Section 6.2.2. In agreement with the van der Pol results, the biased RMS remains at pre-DA levels when the bias is not penalized, but as γ increases, the biased RMS converges to true-RMS levels. Interestingly, Fig. 11a shows that the r-EnKF finds an analysis solution with smaller-norm bias than the true linear bias, which is indicated by the biased post-DA RMS being smaller than the true biased RMS (see the exact values in Table 2). Indeed, the objective of the proposed r-EnKF algorithm is to recover small-norm bias analyses, and, although we are prescribing a form of bias, this may not be the smallest-norm bias solution available. Thus, the r-EnKF can find a set of model parameters $\boldsymbol{\alpha} = [\beta; \tau]$ (which are reported in Appendix E for completeness) that provides a smaller-norm bias than the true bias (32)(a). Fig. 11b shows qualitatively similar results on the nonlinear bias case: consistent small-norm bias is recovered at $\gamma \in [1.5, 4.5]$, which are given by a unique $\boldsymbol{\alpha}$. The less accurate results are expected due to the increased difficulty in the problem (arising from the nonlinearity of the bias). The predictions remain accurate after the filter has been removed, which means that the filter recovers a 'good' unbiased analysis, and the prediction window of the ESN is long in the linear and nonlinear cases. Fig. 12 shows the detailed timeseries for the optimal ($L^\star, \gamma^\star$) (see Table 2). Fig. 12I compares at

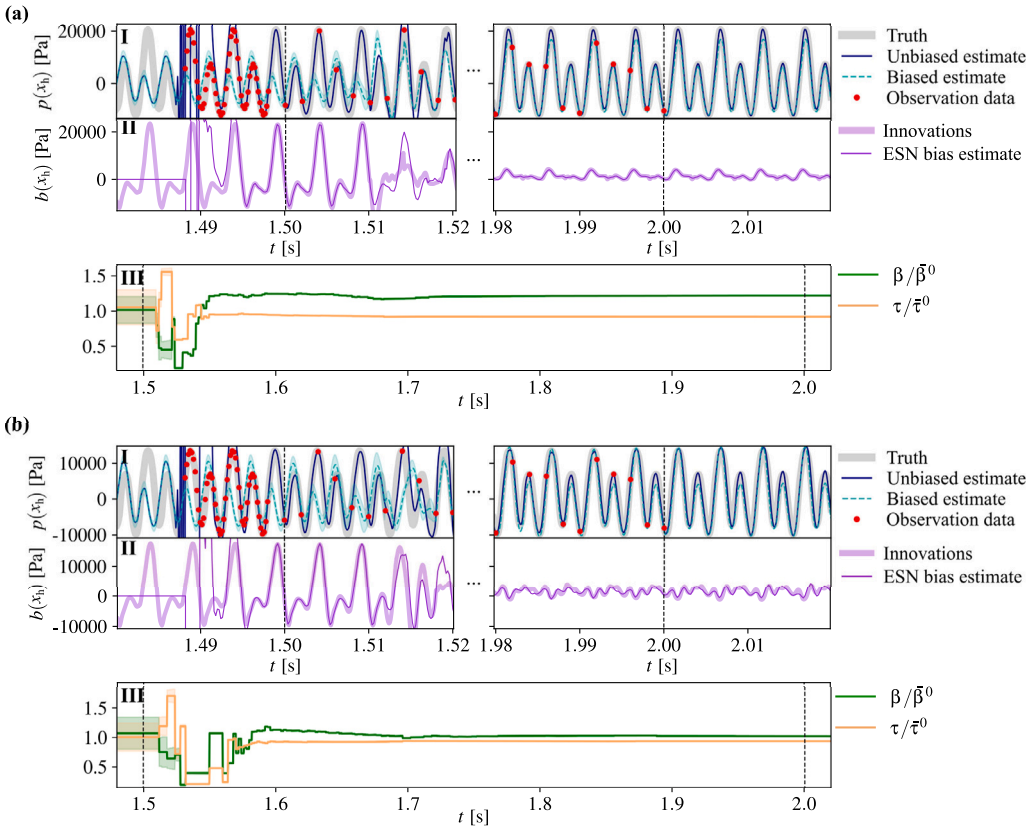

Fig. 12. Nonlinear time-delayed model. Timeseries results with (a) linear bias for $L^\star = 100, \gamma^\star = 1.75$; and (b) nonlinear bias for $L^\star = 60, \gamma^\star = 2.75$. Time evolution at the start and end of assimilation of (I) the true acoustic velocity (thick grey), unbiased estimate (navy), and biased estimate with its uncertainty (dashed teal), and (II) the innovations (thick light orchid) and ESN prediction (thin dark orchid); and (III) time evolution throughout the assimilation of the inferred parameters and their uncertainty, normalized by their initial ensemble mean values. The assimilation window is indicated by the vertical dashed lines, and the red circles show the observation data. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the start and end of the assimilation, the true observable acoustic pressure at the heat source location, $p^{\mathrm{t}}(x_{\mathrm{h}}) + b^{\mathrm{t}}(x_{\mathrm{h}})$, to the filter unbiased estimation, $p(x_{\mathrm{h}}) + b(x_{\mathrm{h}})$, and the biased estimate, $p(x_{\mathrm{h}})$, which is the model estimate before applying the bias correction. The ESN bias estimate $b^{\mathrm{f}}(x_{\mathrm{h}})$ is compared to the true innovations (i.e., the difference between the truth and the biased estimate, $p^{\mathrm{t}}(x_{\mathrm{h}}) + b^{\mathrm{t}}(x_{\mathrm{h}}) - p(x_{\mathrm{h}})$). The agreement between the innovation and the ESN prediction shows that the ESN successfully infers the bias during the assimilation in open-closed loops, and the closed-loop prediction remains accurate after the assimilation ends. The evolution of the parameters throughout the assimilations are shown in Fig. 12III, which indicate that the filter converges rapidly to a combination of thermoacoustic parameters that provides the small-norm bias solution, i.e., with significantly reduced innovations (as seen in Fig. 12II). Importantly, the sampling frequency in real thermoacoustic oscillations experiments is typically 10 kHz (e.g. [76]), which is higher than the sampling required in this framework. As reported in Appendix D, we use a model sampling frequency of 10 kHz, an ESN forecast frequency of 5 kHz, and the assimilation frequency is 0.5 kHz.

Finally, we address the case of the time-dependent bias defined in (32)(c). This is the most challenging scenario because the bias is not only a function of the state, but also an explicit function of time. With the explicit time-dependency, we are adding an incommensurate frequency to the dynamics, thereby creating a quasi-periodic bias. We (i) increase the ESN training time from 0.5 s to 1.5 s to allow the ESN to learn time-varying information (note that the ESN is not aware of the time information); and we (ii) increase the assimilation frequency to 1 kHz to achieve accurate results (which is to be expected because quasi-periodic dynamics are more complex than limit cycles, and a higher sampling rate is required to reconstruct a quasi-periodic signal). Fig. 13a, b show the combined RMS error contours after removing the filter and at the end of the assimilation, respectively. The allowed range of the bias regularization factor is visibly smaller than in the linear and nonlinear biases in Fig. 10, and the combined RMS error is higher after assimilation. However, as shown by the DA RMS contours, the error is small during assimilation, when both DA and post-DA biased RMS are of the same order of magnitude to the true biased RMS (see Table 2). This means that the filter successfully converges to a small-norm solution, but the ESN predictive window is shorter than in the linear and nonlinear baises due to the quasi-periodicity of the time-dependent bias. We show the detailed timeseries results for the optimal post-DA $(L^\star, \gamma^\star)$ in Fig. 14. The biased solution remains close to the truth after the filter is removed (see Fig. 14I). The ESN prediction becomes accurate with

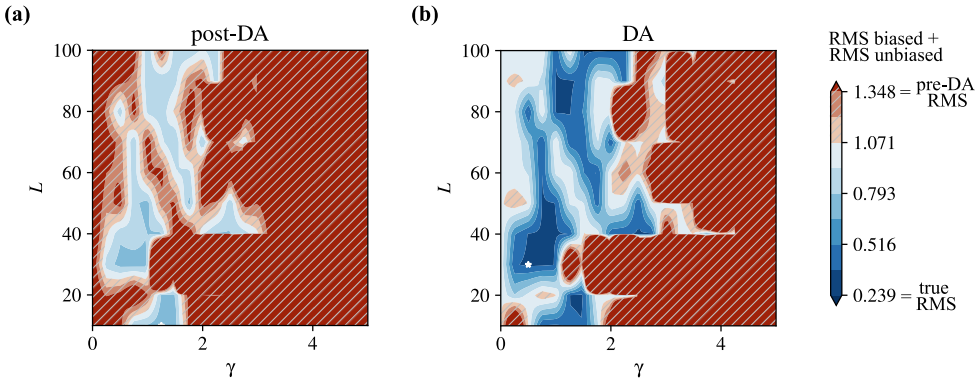

Fig. 13. Nonlinear time-delayed model. Error contours for different (L, γ) combinations for the explicit time-dependent bias. The colourmap shows the sum of the bias and unbiased RMS errors (a) after the filter is removed (post-DA), and (b) at the end of the assimilation (DA). The minimum combined RMS point is indicated with a white star. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

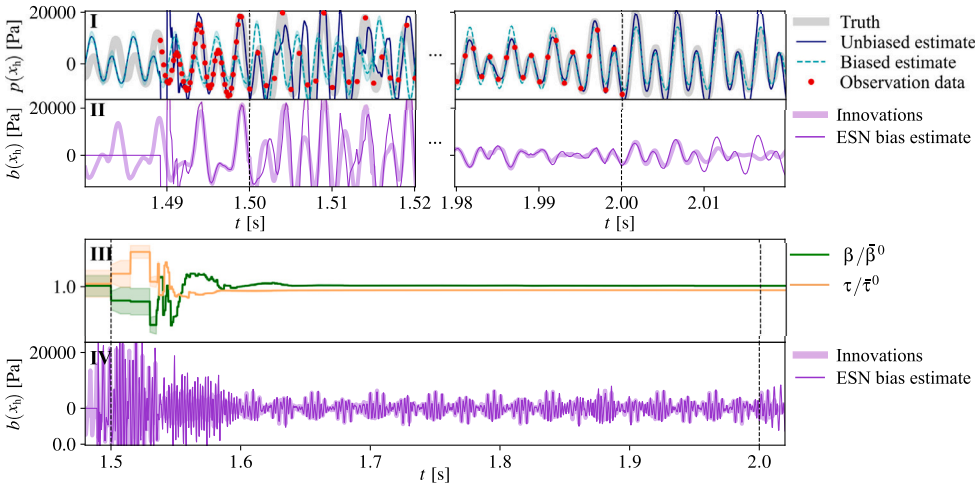

Fig. 14. Nonlinear time-delayed model. Timeseries results with a time-dependent bias for $L^\star = 10, \gamma^\star = 1.25$. Time evolution at the start and end of assimilation of (I) the true acoustic velocity (thick grey), unbiased estimate (navy), and biased estimate with its uncertainty (dashed teal), and (II) the innovations (thick light orchid) and ESN prediction (thin dark orchid); and time evolution throughout the assimilation of (III) the inferred parameters and their uncertainty, normalized by their initial ensemble mean values, and (IV) the innovations and ESN bias prediction. The assimilation window is indicated by the vertical dashed lines, and the red circles show the observation data. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the re-initialization with the analysis innovations (as shown in Fig. 14IV). Assimilating the data with the unbiased signal allows the filter to perform an unbiased analysis, and the bias regularization guides the solution to a small-norm bias estimate (see Section 3).

6.3.2. Performance of the bias-unaware EnKF on the time-varying bias case

In this section, we show how the traditional bias-unaware EnKF performs on a biased model. We show the time-varying bias scenario only for brevity. The results in Fig. 15 are the bias-unaware equivalent to Fig. 14. Both simulations are performed using the same initial ensemble and filter parameters. Noticeably, the magnitude of the innovations is large at the end of the assimilation, and the state estimate does not resemble the truth. Further, it can be seen that the bias-unaware filter attempts to converge, but at $t \approx 1.87$ s there is a sudden instantaneous update in the parameters. We expect this behaviour to occur cyclically due to the time-varying bias. As discussed in Section 2.3, if unbiased data is assimilated into a biased model, the model drift can increase the bias of the analysis, becasue the bias-unaware framework assumes unbiased forecasts.

6.3.3. Robustness of the r-EnKF to the level and colour of the observation noise

In this section, we analyse the impact of the strength and colour of the measurement noise on the assimilation. We perform the robustness study for the case of the nonlinear time-delayed model with nonlinear bias.

In the previous sections the observations were assumed to be subject to small-norm Gaussian noise with $\sigma_d = 0.01$ (see Section 6.3), because we assume that the sensors are well-calibrated. Fig. 16 shows how relaxing this assumption affects the RMS

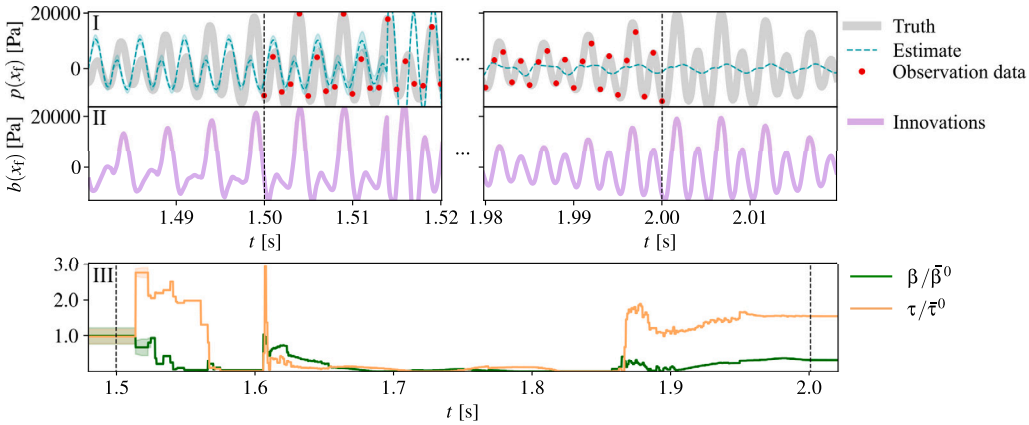

Fig. 15. Nonlinar time-delayed model with a time-dependent bias. Results with a bias-unaware EnKF. Time evolution at the start and end of assimilation of (I) the true acoustic velocity (thick grey), pressure estimate with its uncertainty (dashed teal); and (II) the innovations (thick light orchid). Time evolution throughout the assimilation of (III) the inferred parameters and their uncertainty, normalized by their initial ensemble mean values. The assimilation window is indicated by the vertical dashed lines, and the red circles show the observation data. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

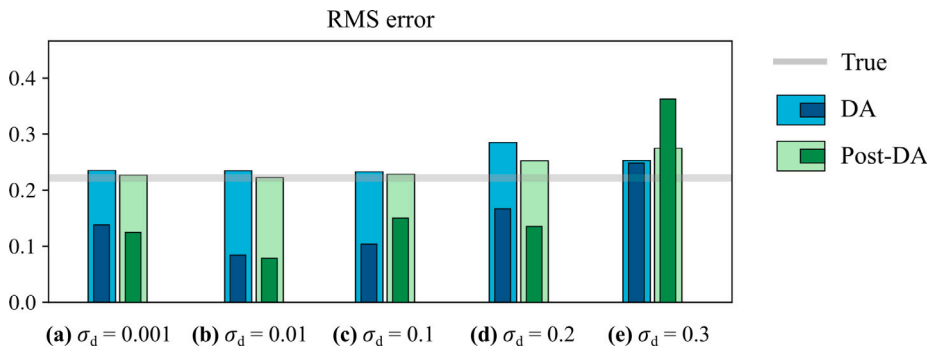

Fig. 16. Nonlinear time-delayed model with a nonlinear bias. Analysis of the effect of increasing the magnitude of the Gaussian additive noise in the assimilation. Biased (thick bars) and unbiased (thin bars) RMS errors at the end of the assimilation (blues) and after the filter has been removed (greens). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

errors at the end of the assimilation. We can see that the biased errors remain at true RMS levels up to a standard deviation in the observations of 10% of the mean absolute amplitude. Moreover, the ESN successfully predicts the bias in the model with observation errors as high as 20%.

Finally, we analyse how the colour of the measurement noise affects the r-EnKF performance. We create white, pink and brown noises in frequency domain, normalize them to have equivalent energy spectrum, and then weight them with a factor of the mean absolute pressure. We consider three factors: 0.1, 0.25, and 0.5, which correspond to a signal-to-noise ratio values of $\mathrm{SNR}_{\mathrm{dB}} = 21.55 \pm 0.21, 13.70 \pm 0.47$, and 7.72 ± 0.48. We compute the signal-to-noise ratio as

$$\mathrm{SNR}_{\mathrm{dB}} = \frac{1}{N_q} \sum_{q=1}^{N_q} 10 \log_{10} \frac{\left\langle {p_q^t}^2 \right\rangle}{\left\langle \left(p_q - \tilde{p}_q\right)^2 \right\rangle}. \tag{33}$$

Fig. 17 shows snapshots of the three coloured noises for the factor 0.25, as well as the corresponding noise-free data. The results for the three noise levels are shown in Fig. 18. The r-EnKF successfully recovers solutions with small biased RMS up to SNR of approximately 14 dB, which is noteworthy because the assumption of Gaussian observation error remains in the assimilation, i.e., the filter has no information about the colour of the noise. These results are key for future application of the r-EnKF in real experimental data, because experimental devices are typically affected by non-Gaussian errors.

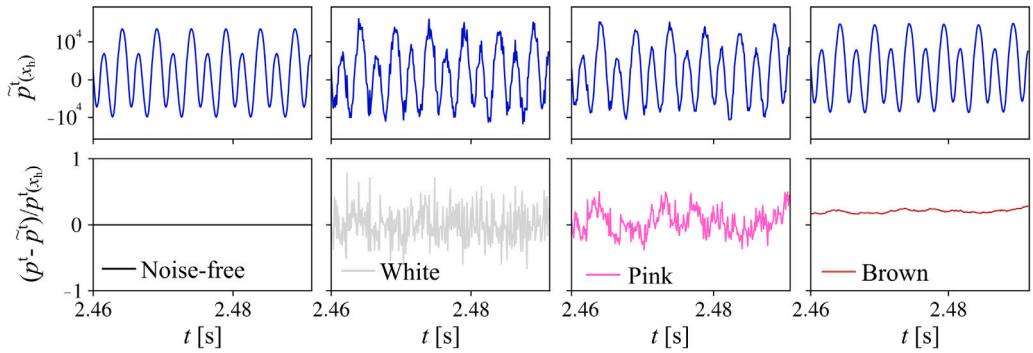

Fig. 17. Nonlinear time-delayed model with a nonlinear bias. Comparison of the noise-free signal with the observation data with additive white, pink, and brown noises. Snapshots of the timeseries of the (a) acoustic pressure at the first microphone, with (b) their corresponding noises with $\mathrm{SNR}_{\mathrm{dB}} = 13.70 \pm 0.47$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

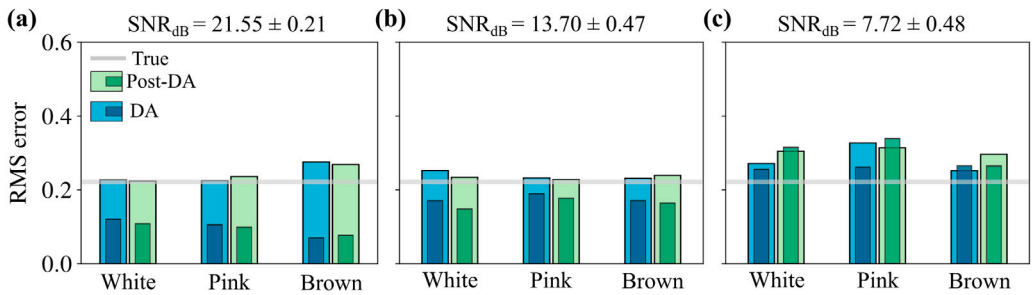

Fig. 18. Nonlinear time-delayed model with a nonlinear bias. Performance of the r-EnKF with white, pink, and brown noises, with (a) $\mathrm{SNR}_{\mathrm{dB}} = 21.55 \pm 0.21$, (b) $\mathrm{SNR}_{\mathrm{dB}} = 13.70 \pm 0.47$, and (c) $\mathrm{SNR}_{\mathrm{dB}} = 7.72 \pm 0.48$. Biased (thick bars) and unbiased (thin bars) RMS errors at the end of the assimilation (blues) and after the filter has been removed (greens). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

7. Conclusions

Upon making physical assumptions and numerical approximations, high-fidelity physical models result in low-order models, which may be affected by uncertainties in the state, parameters, and by model biases. If the model is biased, data assimilation methods can provide state and parameters that are inaccurate, which make the model quantitatively incorrect and not generalizable. Current bias-aware real-time data assimilation methods face two limitations: (i) they cannot ensure that the model bias of the analysis is unique or small in norm; and (ii) they need an *a priori* parametrized model for the bias, whose parameters are inferred within the assimilation. To overcome these limitations, first, we propose a regularized bias-aware sequential assimilation framework, within we derive the regularized bias-aware ensemble Kalman filter (r-EnKF), which is designed to favour small-norm bias analyses, thus, bypassing (i); and second, we infer the model bias using echo state networks (ESNs), which overcome (ii) because they are universal approximators, and we design a multi-parameter training approach for the network.

We test and validate the proposed r-EnKF using two models of nonlinearly coupled oscillators (with and without time-delay), affected by different forms of bias (linear, nonlinear, and time-dependent functions of the state). The results show that the r-EnKF successfully infers a combination of model parameters that provides a physical state with a unique bias, which can be accurately recovered by the ESN. Further, we show that the designed multi-parameter training approach with data augmentation makes the ESN robust and flexible, because the same network can infer biases with different dynamics. We show that the weight of the bias regularization in the assimilation is key to recover small-norm bias analyses because it directly controls the magnitude of the instantaneous changes in state and parameters in every analysis step. Therefore, future work can be focused on the possibility of adaptively tuning the bias regularization weight factor. Finally, we show that the r-EnKF is robust to work with high noise observations as well as with measurements subject to colour noise. The proposed regularized bias-aware ensemble Kalman filter opens new opportunities for real-time prediction in nonlinear systems by performing state, parameter, and bias estimation when data from sensors is available.

Declaration of competing interest

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

Data availability

Data will be made available on request

Acknowledgements

A. N. is supported partly by the EPSRC-DTP, UK, the Cambridge Commonwealth, European and International Trust, UK under a Cambridge European Scholarship, and Rolls Royce, UK. A. R. is supported by the Eric and Wendy Schmidt AI in Science Postdoctoral Fellowship, a Schmidt Futures program. L. M. gratefully acknowledges financial support from the ERC Starting Grant PhyCo 949388. L.M and A.N acknowledge support from the UKRI AI for Net Zero grant EP/Y005619/1.

Code availability

https://github.com/MagriLab/rBA-EnKF.

Appendix A. Recycle validation

Following Racca and Magri [49], we optimize the hyperparameters of the ESN (17) (the input scaling, σ_{in}, and the spectral radius, ρ) through Bayesian optimization and recycle validation (RV). In the RV, the optimal hyperparameters are chosen by minimizing the mean square error of closed-loop predictions over multiple intervals seen during training in open-loop. The number of intervals is known as the number of folds, N_{folds}. Specifically, in each of the time series, we select $N_{\text{folds}} = 4$ of length t_{val}, which is problem-specific and it is reported in Appendix D. The recycle validation algorithm exploits information from multiple parts of the dataset, whilst keeping a low computational cost [49]. This is because RV performs the validation on N_{folds} intervals used in training by keeping a fixed output matrix, $\mathbf{W}_{\text{out}}$, but using the closed-loop configuration rather than the open-loop setting used during training. The mean square error computed through RV is optimized in a hyperparameter space (whose ranges are reported in Appendix D), through Bayesian optimization. We evaluate the error in 16 grid points of the hyperparameter space, and then on 4 additional points, which are selected through the gp-hedge algorithm [77,78].

Appendix B. Derivation of the Jacobian of the ESN

The Jacobian of ESN (17) is the gradient of the output $\boldsymbol{b}^{\text{f}}$ with respect to the input $\boldsymbol{i}$

$$\frac{\mathrm{d}\,\boldsymbol{b}^{\text{f}}(t_{k+1})}{\mathrm{d}\,\boldsymbol{i}(t_k)}, \tag{B.1}$$

where the input $\boldsymbol{i}(t_k)$ in open-loop is data, whereas in the closed-loop the ESN output at the previous time step $\boldsymbol{b}^{\text{f}}(t_k)$ is used as the input. We follow the procedure in Section 4.4 to rewrite (17) as

$$\begin{aligned}\boldsymbol{b}^{\text{f}}(t_{k+1}) &= \mathbf{W}_{\text{out}}^{(1)} \tanh\left(\sigma_{\text{in}}\mathbf{W}_{\text{in}}^{(1)}\left(\boldsymbol{i}(t_k)\odot\boldsymbol{g}\right) + \sigma_{\text{in}}\delta_{\text{r}}\boldsymbol{W}_{\text{in}}^{(2)} + \rho\mathbf{W}\boldsymbol{r}(t_k)\right) + \boldsymbol{W}_{\text{out}}^{(2)}\\ \boldsymbol{r}(t_k) &= \left(\mathbf{W}_{\text{out}}^{(1)\,\text{T}}\mathbf{W}_{\text{out}}^{(1)}\right)^{-1}\mathbf{W}_{\text{out}}^{(1)\,\text{T}}\left(\boldsymbol{b}^{\text{f}}(t_k) - \boldsymbol{W}_{\text{out}}^{(2)}\right).\end{aligned} \tag{B.2}$$

With this, the Jacobian becomes

$$\frac{\mathrm{d}\,\boldsymbol{b}(t_{k+1})}{\mathrm{d}\,\boldsymbol{i}(t_k)} = \mathbf{W}_{\text{out}}^{(1)}\left(\mathbf{T}\odot\left(\sigma_{\text{in}}\mathbf{W}_{\text{in}}^{(1)}\odot\mathbf{G} + \frac{\mathrm{d}\,\boldsymbol{r}(t_k)}{\mathrm{d}\,\boldsymbol{i}(t_k)}\right)\right), \tag{B.3}$$

where the matrices $\mathbf{G}$ and $\mathbf{T}$ are detailed in Section 4.4. The reservoir state $\boldsymbol{r}(t_k)$ is a function of the output from the previous step, $\boldsymbol{b}^{\text{f}}(t_k)$. In closed-loop, $\boldsymbol{i}(t_k) = \boldsymbol{b}^{\text{f}}(t_k)$, and hence

$$\frac{\mathrm{d}\,\boldsymbol{r}(t_k)}{\mathrm{d}\,\boldsymbol{i}(t_k)} = \begin{cases}\rho\mathbf{W}\left(\mathbf{W}_{\text{out}}^{(1)\,\text{T}}\mathbf{W}_{\text{out}}^{(1)}\right)^{-1}\mathbf{W}_{\text{out}}^{(1)\,\text{T}} & \text{if closed-loop.}\\ \mathbf{0} & \text{if open-loop.}\end{cases} \tag{B.4}$$

Appendix C. Nonlinear time-delayed model

The nonlinear time-delayed system analysed in Section 6.3 is illustrated in Fig. C.19. The system models the thermoacoustic oscillations occurring in a laboratory-scale open-ended tube with a compact heat source (also known as the Rijke tube). We employ the low-order model detailed by Juniper [79]. The main modelling assumptions are: (i) the mean flow is a perfect gas with a small Mach number and constant properties, such that mean flow effects are neglected in the acoustic propagation; (ii) the boundary conditions are ideal, i.e. the acoustic pressure is zero at both ends of the tube, as illustrated in Fig. C.19; (iii) the viscous and body forces are negligible; and (iv) the heat source is compact and can be modelled as a point source of sound. Thus, the acoustics in the

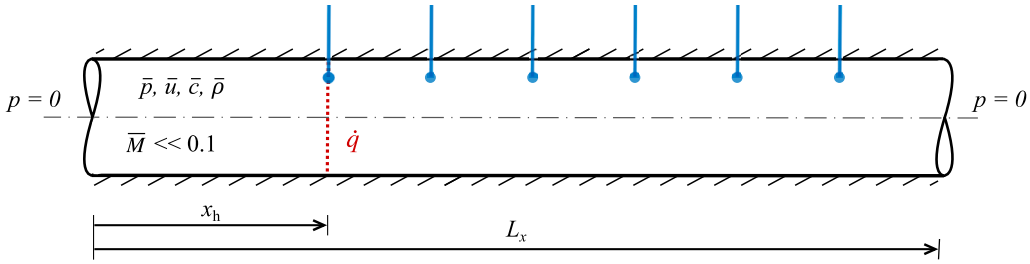

Fig. C.19. Schematic of the nonlinear time-delayed model, which consists of an open-ended tube with a compact heat source. The measurable quantity is the acoustic pressure at different locations in the tube, which are measured by microphones. The blue vertical lines indicate the equidistant microphones, and the heat released by the compact heat source is indicated by the vertical dotted red line.

duct are governed by the one-dimensional linearized momentum and energy conservation equations. The Markovian formulation introduced by Huhn and Magri [71] for the acoustic velocity u and pressure p is

$$\frac{\partial u}{\partial t} + \frac{1}{\bar{\rho}}\frac{\partial p}{\partial x} = 0 \tag{C.1a}$$

$$\frac{\partial p}{\partial t} + \bar{\rho}\,\bar{c}^2\frac{\partial u}{\partial x} = \dot{q}(\Gamma - 1)\delta(x - x_{\mathrm{h}}) - \zeta\frac{\bar{c}}{L_x}p \tag{C.1b}$$

$$\frac{\partial w}{\partial t} - \tau_v\frac{\partial w}{\partial X} = 0 \tag{C.1c}$$

where x_{h} is the heat source location; $\bar{\rho}, \bar{c}$ and $\bar{u}$ are the mean-flow density, speed of sound and velocity, respectively; Γ is the heat capacity ratio; ζ is the damping factor; δ is the Dirac delta distribution, which enforces a compact assumption for the heat release rate $\dot{q}$ [W/m^2], which we model with the simple time-delayed square-root law

$$\dot{q}(t) = \bar{u}\bar{p}\beta\left(\sqrt{\left|\frac{1}{3} + \frac{u\left(x_{\mathrm{h}}, t-\tau\right)}{\bar{u}}\right|} - \sqrt{\frac{1}{3}}\right), \tag{C.2}$$

The advection equation (C.1c) transforms the time-delayed problem into an initial value problem. The equation is used to keep track of the acoustic velocity history at the heat source location, i.e. it acts as a numerical memory. The dummy variable w, which travels in a dimensionless spatial domain X with velocity $-\tau_v$, models the history of the acoustic velocity at the heat source location, x_{h}. We define the boundary condition $w(X = 0, t) = u\left(x = x_{\mathrm{h}}, t\right)$, such that the acoustic velocity at the heat source location at some time $t - \Delta$ is given by $u\left(x_{\mathrm{h}}, t - \Delta\right) = w(0, t - \Delta) = w\left(\tau_v/\Delta, t\right)$. The velocity of the advection is $\tau_v \geq \tau$, because the length of the stored history must be at least equal to the acoustic time delay. (Note that if τ is constant, one can set $\tau_v = \tau$, such that $u\left(x_{\mathrm{h}}, t - \tau\right) = w(1, t)$.)

The acoustic equations are discretized in space with a Galerkin method by taking the natural acoustic modes as the orthogonal spatial basis. The ideal boundary conditions ($p(x = 0) = p(x = L_x) = 0$) are enforced by choosing for the velocity and pressure oscillations the basis

$$u(x,t) = \sum_{j=1}^{N_m}\eta_j(t)\cos\left(\frac{\omega_j}{\bar{c}}x\right) \quad \text{and} \quad p(x,t) = -\sum_{j=1}^{N_m}\mu_j(t)\sin\left(\frac{\omega_j}{\bar{c}}x\right), \tag{C.3}$$

where N_m is the number of the acoustic modes. Finally, we discretize the advection equation in X with a Chebyshev spectral method, using the Chebyshev polynomial matrix $\mathbf{D}$ for $N_c + 1$ collocation points, which are defined as $X_i = 0.5\left(1 - \cos\left(i\pi/N_c\right)\right)$ for $i = 0, \ldots, N_c$. Introducing the Galerkin and Chebyshev discretizations into (C.1) yields the governing equations

$$\frac{\mathrm{d}\eta_j}{\mathrm{d}t} = \frac{\omega_j}{\bar{\rho}\bar{c}}\mu_j \tag{C.4a}$$

$$\frac{\mathrm{d}\mu_j}{\mathrm{d}t} = -\bar{\rho}\,\bar{c}\,\omega_j\eta_j - 2\,\dot{q}\frac{\Gamma - 1}{L_x}\sin\left(\frac{\omega_j}{\bar{c}}x_{\mathrm{h}}\right) - \zeta_j\frac{\bar{c}}{L_x}\mu_j, \quad \text{for} \quad j = 0, \ldots, N_m - 1 \tag{C.4b}$$

$$\frac{\mathrm{d}v_i}{\mathrm{d}t} = 2\tau_v\sum_k \mathsf{D}_{ik}, v_k \quad \text{for} \quad i = 0, \ldots, N_c - 1 \tag{C.4c}$$

where η_j and μ_j are the acoustic velocity and pressure modes resulting from the Galerkin discretization of the acoustic velocity, u, and pressure, p, into N_m acoustic modes with ω_j acoustic frequencies; ζ_j is the damping, defined with the modal form $\zeta_j = C_1 j^2 + C_2\sqrt{j}$ [72].

Appendix D. Simulations' parameters

This appendix summarizes the parameters used for the van der Pol (Table D.3) and nonlinear time-delayed model (Table D.4) simulations.

Table D.3
Parameters used in the van der Pol simulations.

Model	Δt	$1{\cdot}10^{-4}$	s	ω	240π	Hz	β_{limits}	(20, 120)	Hz
	$\bar{\beta}^0$	70.0	Hz	β^{t}	75.0	Hz	κ_{limits}	(0.1, 10.0)	
	$\bar{\kappa}^0$	4.0		κ^{t}	3.4		ζ_{limits}	(20, 120)	Hz
	$\bar{\zeta}^0$	60.0	Hz	ζ^{t}	55.0	Hz			
Filter	Δt_{d}	$30\Delta t$	s	N_q	1		σ_{d}	0.01	
	m	10		t_{err}	0.04	s	σ_α	0.25	
							σ_ϕ	0.25	
ESN	Δt_{ESN}	$5\Delta t$	s	L[a]	10		λ	10^{-16}	
	N_r	100		σ_L	0.5		σ_{in}[b]	$[10^{-5}, 1]$	
	N_{wash}	30		t_{tr}	1.0	s	ρ[b]	[0.7, 1.05]	
	N_{folds}	4		t_{val}	0.01	s			

[a] Indicates that it is the default value but it may be varied in different simulations.
[b] Indicates that the parameter is optimized within the given values;
and if no units are specified, the quantity is dimensionless.

Table D.4
Parameters used in the nonlinear time-delayed model simulations.

Model	Δt	$1{\cdot}10^{-4}$	s	C_1	0.05		$\bar{\beta}^0$	4.0	
	N_c	50		C_2	0.01		$\bar{\tau}^0$	$1.5{\cdot}10^{-3}$	s
	N_m	10		$\bar{u}$	10	m/s	β^{t}	4.2	
	τ_v	10^{-2}	s	$\bar{p}$	1.013	bar	τ^{t}	$1.4{\cdot}10^{-3}$	s
	x_{h}	0.2	m	$\bar{T}$	417.2	K	β_{limits}	(0.1, 5.0)	
	L_x	1.0	m	Γ	1.4		τ_{limits}	$(1{\cdot}10^{-6},\ \tau_v)$	s
Filter	Δt_{d}[a]	$20\Delta t$	s	N_q	6		σ_{d}	0.01	
	m[a]	50		$\boldsymbol{x}$	[0.2, 0.33, 0.47,		σ_α	0.2	
	t_{err}	0.02	s		0.6, 0.73, 0.87]	m	σ_ϕ	0.2	
ESN	Δt_{ESN}	$2\Delta t$	s	L[a]	50		λ	10^{-16}	
	N_r	500		σ_L	0.2		σ_{in}[b]	$[10^{-5}, 10^{-2}]$	
	N_{wash}	50		t_{tr}	0.5	s	ρ[b]	[0.7, 1.05]	
	N_{folds}	4		t_{val}	0.02	s			

[a] Indicates that it is the default value but it may be varied in different simulations.
[b] Indicates that the parameter is optimized within the given values;
and if no units are specified, the quantity is dimensionless.

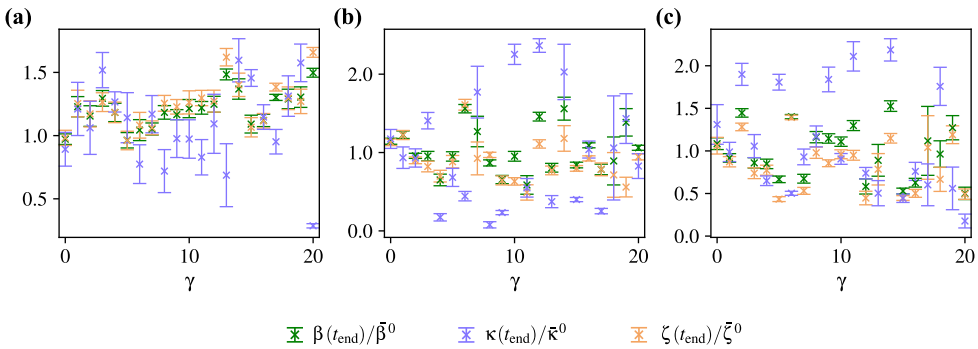

Fig. E.20. Van der Pol model. Distribution of the inferred model parameters with their uncertainties at the end of the assimilation. Results for varying bias regularization factor γ, for different numbers of training sets (a) $L = 1$, (b) $L = 10$, and (b) $L = 50$. The parameters are normalized by the initial ensemble mean.

Appendix E. Parameter convergence results

Fig. E.20 shows the parameter distribution at the end of the assimilation for simulations in Fig. 8. The filter provides solutions with large uncertainties for $L = 1$, and 10 (as seen by the large ensemble spread in Fig. E.20a,b), which indicate non-converged solutions. Further, we see that there are many (β, ζ, κ) combinations that recover 'good' unbiased analysis, which is expected as there are multiple solutions in the van der Pol system (see Eq. (26)).

Fig. E.21 shows the parameter distribution at the end of the assimilation in Fig. 11. We can see that, in contrast to Fig. E.20, the filter consistently converges to a similar set of parameters when the biased and unbiased errors are small, i.e., at the range of bias regularization factor $1 \lesssim \gamma \lesssim 5$ (see Fig. 11).

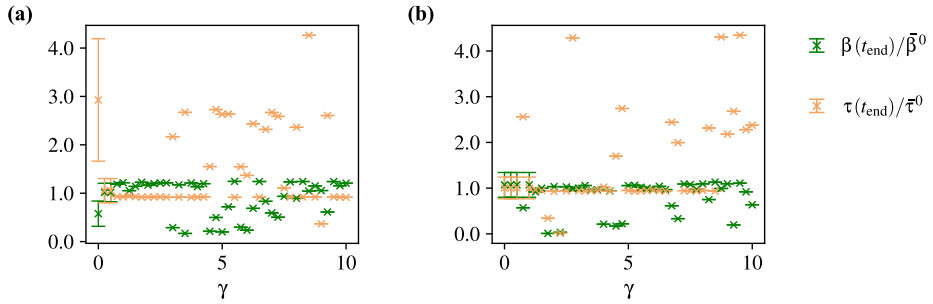

Fig. E.21. Nonlinear time-delayed model. Distribution of the inferred model parameters with their uncertainties at the end of the assimilation. Results for varying bias regularization factor γ, for the (a) linear and (b) nonlinear bias cases at fixed $L = 70$ and $m = 50$. The parameters are normalized by the initial ensemble mean.

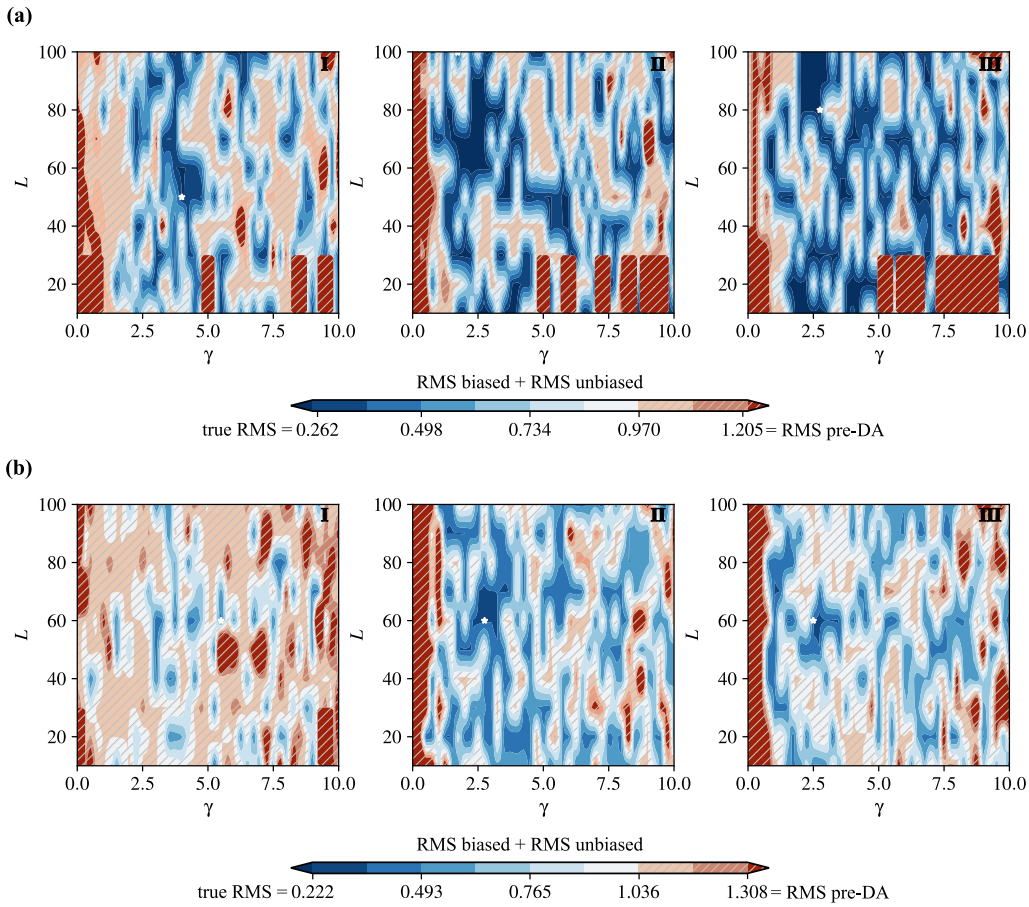

Fig. F.22. Nonlinear time-delayed model. Error comparison of the post-assimilation error metrics for three ensemble sizes (I) $m = 10$, (II) $m = 50$, (III) $m = 80$, for the (a) linear and (b) nonlinear biases. Contours for varying training set size, L, and bias regularization factor, γ. The colourmap shows the sum of the bias and unbiased RMS errors, and the minimum combined RMS point is indicated with a white star. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Appendix F. Analysis of the ensemble size

Fig. F.22 shows the error contours for three ensemble sizes for the nonlinear time-delayed model with linear and nonlinear biases. The difference between $m = 10$ and 50 is significant, whilst increasing the ensemble size to $m = 80$ does not notably vary the results. Thus, we fix $m = 50$ in Section 6.3.1.

References

[1] Albert Tarantola, Inverse Problem Theory and Methods for Model Parameter Estimation, Society for Industrial and Applied Mathematics, ISBN: 9780898715729, 2005, p. 355.
[2] Geir Evensen, Data Assimilation: The Ensemble Kalman Filter, Springer Science & Business Media, 2009.
[3] Leopold Haimberger, Homogenization of radiosonde temperature time series using innovation statistics, J. Clim. 20 (7) (2007) 1377–1403.
[4] Brian D.O. Anderson, John B. Moore, Optimal Filtering, Courier Corporation, 2012.
[5] Dick P. Dee, Bias and data assimilation, Q. J. R. Meteorol. Soc.: J. Atmos. Sci. Appl. Meteorol. Phys. Oceanogr. 131 (613) (2005) 3323–3343.
[6] Andrea Nóvoa, Luca Magri, Real-time thermoacoustic data assimilation, J. Fluid Mech. 948 (2022) A35, http://dx.doi.org/10.1017/jfm.2022.653.
[7] Michael D. Tsyrulnikov, Stochastic modelling of model errors: A simulation study, Q. J. R. Meteorol. Soc.: J. Atmos. Sci. Appl. Meteorol. Phys. Oceanogr. 131 (613) (2005) 3345–3371.
[8] John R. Eyre, Observation bias correction schemes in data assimilation systems: A theoretical study of some of their properties, Q. J. R. Meteorol. Soc. 142 (699) (2016) 2284–2291.
[9] Yannick Trémolet, Accounting for an imperfect model in 4D-var, Q. J. R. Meteorol. Soc.: J. Atmos. Sci. Appl. Meteorol. Phys. Oceanogr. 132 (621) (2006) 2483–2504.
[10] Patrick Laloyaux, Massimo Bonavita, Mohamed Dahoui, J. Farnan, Sean Healy, E. Hólm, S.T.K. Lang, Towards an unbiased stratospheric analysis, Q. J. R. Meteorol. Soc. 146 (730) (2020) 2392–2409.
[11] John C. Derber, Wan-Shu Wu, The use of TOVS cloud-cleared radiances in the NCEP SSI analysis system, Mon. Weather Rev. 126 (8) (1998) 2287–2299.
[12] Thomas Auligné, Anthony P. McNally, Dick P. Dee, Adaptive bias correction for satellite data in a numerical weather prediction system, Q. J. R. Meteorol. Soc.: J. Atmos. Sci. Appl. Meteorol. Phys. Oceanogr. 133 (624) (2007) 631–642.
[13] Wei Han, Niels Bormann, Constrained adaptive bias correction for satellite radiance assimilation in the ecmwf 4d-var system, Technical Report, European Centre for Medium-Range Weather Forecasts Reading, UK, 2016.
[14] Michael J. Bell, Matthew J. Martin, Nancy K. Nichols, Assimilation of data into an ocean model with systematic errors near the equator, Q. J. R. Meteorol. Soc.: J. Atmos. Sci. Appl. Meteorol. Phys. Oceanogr. 130 (598) (2004) 873–893.
[15] Magdalena A. Balmaseda, Dick P. Dee, Arthur Vidard, David L.T. Anderson, A multivariate treatment of bias for sequential data assimilation: Application to the tropical oceans, Q. J. R. Meteorol. Soc.: J. Atmos. Sci. Appl. Meteorol. Phys. Oceanogr. 133 (622) (2007) 167–179.
[16] Rudolph Emil Kalman, A new approach to linear filtering and prediction problems, J. Basic Eng. 82 (1) (1960) 35–45.
[17] Bernard Friedland, Treatment of bias in recursive filtering, IEEE Trans. Automat. Control 14 (4) (1969) 359–367.
[18] Bernard Friedland, Notes on separate-bias estimation, IEEE Trans. Automat. Control 23 (4) (1978) 735–738.
[19] Mario B. Ignagni, Separate bias Kalman estimator with bias state noise, IEEE Trans. Automat. Control 35 (3) (1990) 338–341.
[20] D.H. Zhou, Y.X. Su, Y.G. Xi, Zhong-Jun Zhang, Extension of Friedland's separate-bias estimation to randomly time-varying bias of nonlinear systems, IEEE Trans. Automat. Control 38 (8) (1993) 1270–1273.
[21] Dick P. Dee, Arlindo M. da Silva, Data assimilation in the presence of forecast bias, Q. J. R. Meteorol. Soc. 124 (545) (1998) 269–295.
[22] Paul-Baptiste Rubio, Ludovic Chamoin, François Louf, Real-time Bayesian data assimilation with data selection, correction of model bias, and on-the-fly uncertainty propagation, C. R. Méc. 347 (11) (2019) 762–779.
[23] André F.C. da Silva, Tim Colonius, Flow state estimation in the presence of discretization errors, J. Fluid Mech. 890 (2020).
[24] Francisco Chinesta, Elias Cueto, Emmanuelle Abisset-Chavanne, Jean Louis Duval, Fouad El Khaldi, Virtual, digital and hybrid twins: a new paradigm in data-based engineering and engineered data, Arch. Comput. Methods Eng. 27 (2020) 105–134.
[25] Basile Marchand, Ludovic Chamoin, Christian Rey, Real-time updating of structural mechanics models using Kalman filtering, modified constitutive relation error, and proper generalized decomposition, Internat. J. Numer. Methods Engrg. 107 (9) (2016) 786–810.
[26] Rodrigo Astroza, Andrés Alessandri, Joel P. Conte, A dual adaptive filtering approach for nonlinear finite element model updating accounting for modeling uncertainty, Mech. Syst. Signal Process. 115 (2019) 782–800.
[27] Matthieu Diaz, P-É Charbonnel, Ludovic Chamoin, A new Kalman filter approach for structural parameter tracking: application to the monitoring of damaging structures tested on shaking-tables, Mech. Syst. Signal Process. 182 (2023) 109529.
[28] Willy Haik, Yvon Maday, Ludovic Chamoin, A real-time variational data assimilation method with data-driven model enrichment for time-dependent problems, Comput. Methods Appl. Mech. Engrg. 405 (2023) 115868.
[29] Henry D.I. Abarbanel, Paul J. Rozdeba, Sasha Shirman, Machine learning: Deepest learning as statistical data assimilation problems, Neural Comput. 30 (8) (2018) 2025–2055.
[30] Peter A.G. Watson, Applying machine learning to improve simulations of a chaotic dynamical system using empirical error correction, J. Adv. Modelling Earth Syst. 11 (5) (2019) 1402–1417.
[31] Marc Bocquet, Julien Brajard, Alberto Carrassi, Laurent Bertino, Bayesian inference of chaotic dynamics by merging data assimilation, machine learning and expectation-maximization, Found. Data Sci. 2 (1) (2020) 55.
[32] Alan J. Geer, Learning earth system models from observations: machine learning or data assimilation? Phil. Trans. R. Soc. A 379 (2194) (2021) 20200089.
[33] Julien Brajard, Alberto Carrassi, Marc Bocquet, Laurent Bertino, Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: A case study with the Lorenz 96 model, J. Comput. Sci. 44 (2020) 101171.
[34] Massimo Bonavita, Patrick Laloyaux, Machine learning for model error inference and correction, J. Adv. Modelling Earth Syst. 12 (12) (2020) e2020MS002232.
[35] Alban Farchi, Patrick Laloyaux, Massimo Bonavita, Marc Bocquet, Using machine learning to correct model error in data assimilation and forecast applications, Q. J. R. Meteorol. Soc. 147 (739) (2021) 3067–3084.
[36] Hans Yu, Matthew P. Juniper, Luca Magri, Combined state and parameter estimation in level-set methods, J. Comput. Phys. 399 (2019) 108950.
[37] David González, Alberto Badias, Icíar Alfaro, Francisco Chinesta, Elias Cueto, Model order reduction for real-time data assimilation through extended Kalman filters, Comput. Methods Appl. Mech. Engrg. 326 (2017) 679–693.
[38] Sebastian Reich, Colin Cotter, Probabilistic Forecasting and Bayesian Data Assimilation, Cambridge University Press, Cambridge, ISBN: 978-1-107-70680-4, 2015, http://dx.doi.org/10.1017/CBO9781107706804.
[39] Gerrit Burgers, Peter Jan van Leeuwen, Geir Evensen, Analysis scheme in the ensemble Kalman filter, Mon. Weather Rev. 126 (6) (1998) 1719–1724.
[40] Geir Evensen, The ensemble Kalman filter: Theoretical formulation and practical implementation, Ocean Dyn. 53 (4) (2003) 343–367.
[41] Hans Yu, Matthew P. Juniper, Thomas Jaravel, Matthias Ihme, Luca Magri, Data assimilation and optimal calibration in nonlinear models of flame dynamics, in: ASME Turbo Expo GT2019-92052, 2019.
[42] Gene H. Golub, Charles F. Van Loan, Matrix Computations, JHU Press, 2013.
[43] Herbert Jaeger, Harald Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, Science 304 (5667) (2004) 78–80.
[44] Wolfgang Maass, Thomas Natschläger, Henry Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, Neural Comput. 14 (11) (2002) 2531–2560.
[45] Charu C. Aggarwal, Neural Networks and Deep Learning, Vol. 10, Springer, 2018, 978–3.

[46] Lyudmila Grigoryeva, Juan-Pablo Ortega, Echo state networks are universal, Neural Netw. (ISSN: 0893-6080) 108 (2018) 495–508, http://dx.doi.org/10.1016/j.neunet.2018.08.025.
[47] Mantas Lukoševičius, A practical guide to applying echo state networks, in: Neural Networks: Tricks of the Trade, Springer, 2012, pp. 659–686.
[48] Francisco Huhn, Luca Magri, Learning ergodic averages in chaotic systems, in: International Conference on Computational Science, Springer, 2020, pp. 124–132.
[49] Alberto Racca, Luca Magri, Robust optimization and validation of echo state networks for learning chaotic dynamics, Neural Netw. 142 (2021) 252–268.
[50] Alberto Racca, Luca Magri, Data-driven prediction and control of extreme events in a chaotic flow, Phys. Rev. Fluids 7 (2022) 104402, http://dx.doi.org/10.1103/PhysRevFluids.7.104402.
[51] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016.
[52] Jianyu Liang, Koji Terasaki, Takemasa Miyoshi, A machine learning approach to the observation operator for satellite radiance data assimilation, J. Meteorol. Soc. Japan Ser. II 101 (1) (2023) 79–95.
[53] André F.C. da Silva, Tim Colonius, Ensemble-based state estimator for aerodynamic flows, AIAA J. 56 (7) (2018) 2568–2578.
[54] Peter Jan Van Leeuwen, Comment on "data assimilation using an ensemble Kalman filter technique", Mon. Weather Rev. 127 (6) (1999) 1374–1377.
[55] Mark Dykman, Fluctuating Nonlinear Oscillators: From Nanomechanics to Quantum Superconducting Circuits, Oxford University Press, 2012.
[56] Gary Drzewiecki, Fundamentals of Chaos and Fractals for Cardiology, Springer, 2021.
[57] Rajarshi Roy, K. Scott Thornburg, Experimental synchronization of chaotic lasers, Phys. Rev. Lett. 72 (13) (1994) 2009.
[58] Michael C. Cross, Jeffrey L. Rogers, Ron Lifshitz, Alex Zumdieck, Synchronization by reactive coupling and nonlinear frequency pulling, Phys. Rev. E 73 (3) (2006) 036205.
[59] Hieu Tri Nguyen, Dentcho A. Genov, Hamzeh Bardaweel, Vibration energy harvesting using magnetic spring based nonlinear oscillators: Design strategies and insights, Appl. Energy 269 (2020) 115102.
[60] Lord Rayleigh, The explanation of certain acoustical phenomena, Nature 18 (1878) 319–321, http://dx.doi.org/10.1038/018319a0.
[61] Tim C. Lieuwen, Unsteady Combustor Physics, Cambridge University Press, Cambridge, ISBN: 978-1-139-05996-1, 2012, http://dx.doi.org/10.1017/CBO9781139059961.
[62] Luca Magri, Adjoint methods as design tools in thermoacoustics, Appl. Mech. Rev. 71 (2) (2019) http://dx.doi.org/10.1115/1.4042821.
[63] Raman I. Sujith, Vishnu R. Unni, Complex system approach to investigate and mitigate thermoacoustic instability in turbulent combustors, Phys. Fluids 32 (6) (2020) 061401.
[64] Luca Magri, Peter J. Schmid, Jonas P. Moeck, Linear flow analysis inspired by mathematical methods from quantum mechanics, Annu. Rev. Fluid Mech. 55 (2023) 541–574.
[65] Balth Van der Pol, LXXXVIII. On "relaxation-oscillations", Lond. Edinb. Dublin Philos. Mag. J. Sci. 2 (11) (1926) 978–992.
[66] A.P. Dowling, Nonlinear self-excited oscillations of a ducted flame, J. Fluid Mech. 346 (1997) 271–290, http://dx.doi.org/10.1017/S0022112097006484.
[67] Nicolas Noiray, Linear growth rate estimation from dynamics and statistics of acoustic signal envelope in turbulent combustors, J. Eng. Gas Turb. Power 139 (4) (2017).
[68] Maria A. Heckl, Non-linear acoustic effects in the rijke tube, Acustica 72 (1990).
[69] Koushik Balasubramanian, Raman I. Sujith, Thermoacoustic instability in a Rijke tube: Non-normality and nonlinearity, Phys. Fluids 20 (044103) (2008) http://dx.doi.org/10.1063/1.2895634.
[70] Luca Magri, Koushik Balasubramanian, Raman I. Sujith, Matthew P. Juniper, Non-normality in combustion-acoustic interaction in diffusion flames: a critical revision, J. Fluid Mech. 719 (2013) 183–202.
[71] Francisco Huhn, Luca Magri, Stability, sensitivity and optimisation of chaotic acoustic oscillations, J. Fluid Mech. 882 (2020) A24.
[72] Lev D. Landau, Evgenii M. Lifshitz, Fluid Mechanics, second ed., Pergamon Press, ISBN: 978-0-08-033933-7, 1987, http://dx.doi.org/10.1016/B978-0-08-033933-7.50009-X.
[73] Lloyd N. Trefethen, Spectral Methods in MATLAB, Vol. 10, SIAM, 2000.
[74] Lipika Kabiraj, Raman I. Sujith, Pankaj Wahi, Bifurcations of self-excited ducted laminar premixed flames, J. Eng. Gas Turb. Power 134 (3) (2012) 031502–1–031502–7, http://dx.doi.org/10.1115/1.4004402.
[75] José G. Aguilar Pérez, Sensitivity Analysis and Optimization in Low Order Thermoacoustic Models (Ph.D. thesis), University of Cambridge, 2019.
[76] Lipika Kabiraj, Raman I. Sujith, Nonlinear self-excited thermoacoustic oscillations: intermittency and flame blowout, J. Fluid Mech. 713 (2012) 376–397.
[77] Eric Brochu, Vlad M. Cora, Nando de Freitas, A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning, 2010, arXiv preprint arXiv:1012.2599.
[78] Matthew Hoffman, Eric Brochu, Nando De Freitas, et al., Portfolio allocation for Bayesian optimization, in: UAI, 2011, pp. 327–336.
[79] Matthew P. Juniper, Triggering in the horizontal Rijke tube: non-normality, transient growth and bypass transition, J. Fluid Mech. 667 (2011) 272–308, http://dx.doi.org/10.1017/S0022112010004453.