

Reduced Order Modelling of a Reynolds number 10^6 jet flow using Machine Learning approaches

A. R. Murali

Ecole Centrale de Lyon, CNRS, Univ Claude Bernard Lyon 1, INSA Lyon, Laboratoire de Mécanique des Fluides et d'Acoustique, UMR5509, 69130, Ecully, France

M. L. K. Kennedy

Institut für Bauingenieurwesen, Technische Universität Berlin, Gustav-Meyer-Allee 25, 13355 Berlin

Vasily Gryazev, Umberto Armani, Annabel P. Markesteijn, Vassili Toropov
School of Engineering and Material Science, Queen Mary University of London, London

S. Elnaz Naghibi

School of Architecture, Computing and Engineering, University of East London, London

Marc C. Jacob

Ecole Centrale de Lyon, CNRS, Univ Claude Bernard Lyon 1, INSA Lyon, Laboratoire de Mécanique des Fluides et d'Acoustique, UMR5509, 69130, Ecully, France

Reinhard Hinkelmann

Institut für Bauingenieurwesen, Technische Universität Berlin, Berlin

Sergey A. Karabasov

School of Engineering and Material Science, Queen Mary University of London, London

The extraction of the most dynamically important coherent flow structures using reduced order models (ROM) is a challenging task in various fluid dynamics applications. In particular, for high-speed round jet flows, the axisymmetric pressure mode of interest is known to be responsible for sound radiation at small angles to the jet axis and dominant contribution to the jet noise peak. In this work the axisymmetric pressure mode of the Navier-Stokes solution of a high speed jet flow at low frequency is reconstructed from simulation data using popular Machine Learning (ML) methods, whose output can later be exploited for data-driven design of effective turbulent acoustic source models. The data used as input for the ML techniques are derived from the Large Eddy Simulation database obtained by application of the high-resolution CABARET method accelerated on GPU cards for flow solutions to NASA Small Hot Jet Acoustic Rig (SHJAR) jets. The SHJAR simulation database is fed to Spectral Proper Orthogonal (SPOD), and the resulting time coefficients of the turbulent pressure fluctuations are the targets of the three machine learning methods put to test in this work. The first Machine Learning method used is the Feed-forward Neural Networks technique, which was successfully implemented for a turbulent flow over a plunging aerofoil in the literature. The second method is based on the application of Genetic Programming, which is a symbolic regression method well-known in optimisation research, but it has not been applied for turbulent flow reconstruction before. The third method, commonly known as Echo State Networks (ESNs), is a time series prediction and reconstruction method from the field of Reservoir Computing. A report on the attempts to apply these methods for approximation and extrapolation of the turbulent flow signals are discussed.

I. Introduction

In various areas of fluid mechanics from aerospace and wind engineering to geophysical flow modelling, insights into the complex flow problems afforded by developing fast reduced-order models are important for fast parameter

studies as well as control and optimisation purposes [1–4]. Many flows of interest are turbulent with a wide range of space-time scales, hence, the typical goal of data-driven reduced-order models is to approximate only the most dynamically important coherent structures in the flow. Due to the progress in high-resolution experimental and numerical simulation methods, large amounts of space and time resolved data have been accumulated over the years, which can be used to separate deterministic and stochastic dynamics of such flows [5–7], thereby facilitating the development of reduced-order models in high-Reynolds numbers.

The first step in many data-driven models is to separate the temporal variability from the spatial field distribution, mode by mode, to reduce complexity of the original four-dimensional (space-time) flow solution. Several ways have been developed in the literature to achieve such modal decomposition, which include Dynamic Mode Decomposition (DMD) and Proper Orthogonal Decomposition (POD) and their modifications [8, 9]. In particular, the POD methods starting from the original work of Lumley [10] and the more recent modifications [11, 12] allow arranging the modes in the order of their energy. In addition, advanced POD modifications such as Spectral POD (SPOD) in the time domain [11] and in the frequency domain [13] allow decomposing different modes by their degree of unsteadiness, i.e. the frequency content. On one hand, the frequency-domain SPOD method allows strictly separating the POD modes according to frequency, which is important for the characterisation of spatiotemporal coherent structures in turbulence [14]. On the other hand, the time-domain SPOD approach enables a user-controlled filtering of the POD correlation matrix leading to smooth temporal SPOD coefficients which are typically easier to reconstruct using machine learning methods in comparison to the unfiltered fluctuations. The dominant modes resulting from such flow decomposition serve to reproduce the most energetic and dynamic features of the turbulent flow of interest at the same time.

Using these advanced POD approaches, several methods to approximate the dynamics of the temporal modal coefficients were developed in the literature, including Galerkin projection and its variants [15] and sparse regression of nonlinear dynamics [16]. Notably, Carlberg et al. [17] reported that reduced-order models may exhibit unstable behaviour when trying to reproduce the complex dynamics of the governing Navier-Stokes equations. Furthermore, despite the progress in the development of more stable reduced-order models [18, 19], the application of these techniques to reconstruct dynamics of high Reynolds-number flows still remains a challenge.

Among various machine learning methods, artificial neural networks have reached a state of maturity and are widely used for reconstructing the dynamics of high-dimensional systems [20, 21]. In particular, a special sub-class of neural network methods called Feedforward Neural Networks (FNN) was found very effective for modelling a range of dynamical systems [22]. In [23], the FNN technique was applied to reduced order modelling of a turbulent flow over a cylinder and a plunging aerofoil, where the flow modal decomposition was combined with the regression analysis of the resulting temporal modal coefficients. As an input, a sufficiently long dataset of Large Eddy Simulations (LES) was used for training and validation of the data-driven model. To improve performance of the regression step, the temporal modal coefficients were obtained from the time-domain SPOD method [11]. The method was demonstrated to predict reliably the flow field beyond the training window and with larger time increments than those used by the full order model, thereby showing the robustness of the reduced-order model based on FNN. In a more recent work [24], the time-domain SPOD in combination with a two-stage conservative and restrictive sparsification procedure was applied to reconstruct the temporal modal coefficients of the flow over a circular cylinder undergoing vortex-induced vibration. The cross-validation procedure was applied to a Particle Image Velocimetry (PIV) dataset reporting good results for accurately reconstructing the most dynamically important vibration mode.

In the current work, the modern data-driven modelling approaches based on SPOD are pushed to the limit by considering a high-speed axisymmetric jet flow, which has a few orders of magnitude larger Reynolds number than the cylinder and the aerofoil flow modelled by Lui and Wolf [23]. The jet corresponds to the isolated unheated static jet at conditions of the NASA SHJAR (Small Hot Jet Acoustic Rig) experiment. The jet Reynolds number based on the nozzle exit diameter is around one million and the acoustic Mach number is 0.9 (Set Point 7 jet case) [25]. Notably, in comparison with the plunging aerofoil flow considered in [23], the high-speed jet flow is inherently broadband and not associated with any distinct tonal frequency. Hence, following previous work on low-rank dynamics of unforced turbulent jets, the objective of reduced-order modelling of the high-speed jet here is reconstructing the axisymmetric pressure mode of the Navier-Stokes solution at low frequency, which can later be used for data-driven design of effective turbulent acoustic source models. In particular, for high-speed round jet flows, the axisymmetric pressure mode of interest is known to be responsible for sound radiation at small angles to the jet axis and dominant contribution to the jet noise peak.

The data-driven modelling will be based on the LES solutions of the NASA jet, which were previously validated in the work of Markesteijn and Karabasov [26]. The LES solutions were obtained using the high-resolution CABARET method [27–31] implemented with asynchronous time stepping [32, 33], and accelerated on Graphics Processing Units

(GPUs) [34–36]. The latter property allows obtaining statistically converged jet LES solutions within a few days on grids of around 100 million cells. For modal decomposition, both the time-domain [11] and the frequency domain [13] SPOD methods will be considered. The temporal modal coefficients extracted from the GPU CABARET solution using the time or frequency domain SPOD methods will be used for training and validation of the reduced order techniques for signal reconstruction.

For studying the ML-approaches, the dataset was tested using three different tools, namely the FNNs, Genetic Programming (GP), and Echo State Networks (ESNs). Among these, GP is a method that originates from the class of Genetic Algorithms (GAs) with the goal of addressing symbolic regression problems. The result of reduced order modelling of GP is thus an analytic function constructed from a zoo of possible function prototypes with guaranteed stability properties. The latter, ESNs, on the other hand is an artificial dynamical oscillator which mimicks the behaviour of a natural chaotic system in the temporal domain [37]. Hence this approach is a way to model the stochastic behaviour of the reduced order modes of the high Re turbulence field in the time domain.

The paper is organised as follows: Section 2 gives an outline of the Large Eddy Simulation dataset of the NASA SHJAR jet flow at Set Point 7 conditions, summarises details of the time- and frequency- domain SPOD methods, and provides examples of the time signals used as an input in the subsequent Machine Learning models.

II. Jet Flow Cases and Computation Methodology

A. Jet Flow Solutions and Spectral Decompositions

In the NASA Small Hot Jet Acoustic Rig (SHJAR) experiment, a single-stream unheated static jet forms downstream of a profiled axisymmetric convergent nozzle with an exit diameter, $D_j = 0.05m$. In particular, the Set Point 7 (SP7) case, which corresponds to an acoustic Mach number $M_a = U_j/a_{amb}$ of 0.9 is considered, where U_j is the jet exit velocity and a_{amb} is the ambient sound speed. The GPU LES solution for this jet case was obtained on a grid of snappy hexa/hanging nodes type, and was sufficiently refined in the jet shear layer region to enable a fast transition to turbulence from the initially laminar inflow boundary condition. The LES domain has 30 Dj radially, 100 Dj axially downstream of the nozzle exit and 10 Dj axially upstream of the nozzle exit. A part of the nozzle was included in the calculation whilst specifying stagnation pressure and temperature boundary condition at the nozzle inlet. Characteristic non-reflecting boundary conditions were specified on all open-domain boundaries, and a wall model was used for all nozzle surfaces. Details of the LES grid resolution and validation results in comparison with the NASA experiment can be found in previous work of Markesteijn et al. [38].

To simplify the postprocessing, the LES pressure solution was interpolated from the original unstructured snappy hexagrid onto a structured cylindrical grid, $(n_x \times n_r \times n_\theta) = (X, Y, Z)$ spanning the axial, radial, and circumferential directions, $x/D_j, r/D_j, \in [0, 10] \times [0, 3] \times [0, 2\pi]$, respectively. Here the origin, $x = 0$ corresponds to the nozzle exit location.

Table 1 summarises the main parameters of the jet flow as well as the interpolation of the LES pressure solution. In total, the LES solution history over 10^3 (978) convective time units D_j/U_j were used, which corresponds to 5348 time frames available for the subsequent reduced-order modelling.

Set Point (SP)	7
Acoustic Mach number M_a	0.9
Jet Mach Number M_j	0.98
Nozzle Temperature Ratio	0.835
Reynolds Number based on jet diameter	1.10^6
Parameters of the cylindrical grid, $dx/D_j = dr/D_j, d\theta[0]$	0.1, 1.4
Sampling frequency, $x(U_j/D_j)$	8.33
Number of time samples	5348

Table 1 The operating conditions of NASA SHJAR jet and the parameters of interpolated cylindrical volume.

After interpolation, the zero-mean pressure time fluctuations, p' are computed:

$$p'(x, r, \theta, t) = p(x, r, \theta, t) - \bar{p}(x, r, \theta) \quad (1)$$

where $\bar{(\cdot)}$ denotes the time average quantity and $(\cdot)'$ denotes the fluctuating part. Instantaneous snapshots of the fluctuating pressure, p' in the nozzle symmetry plane (x - y) and several cross-stream planes (y - z) are shown in Fig. 1. The pressure fluctuations obtained will be used in the modal decomposition analysis.

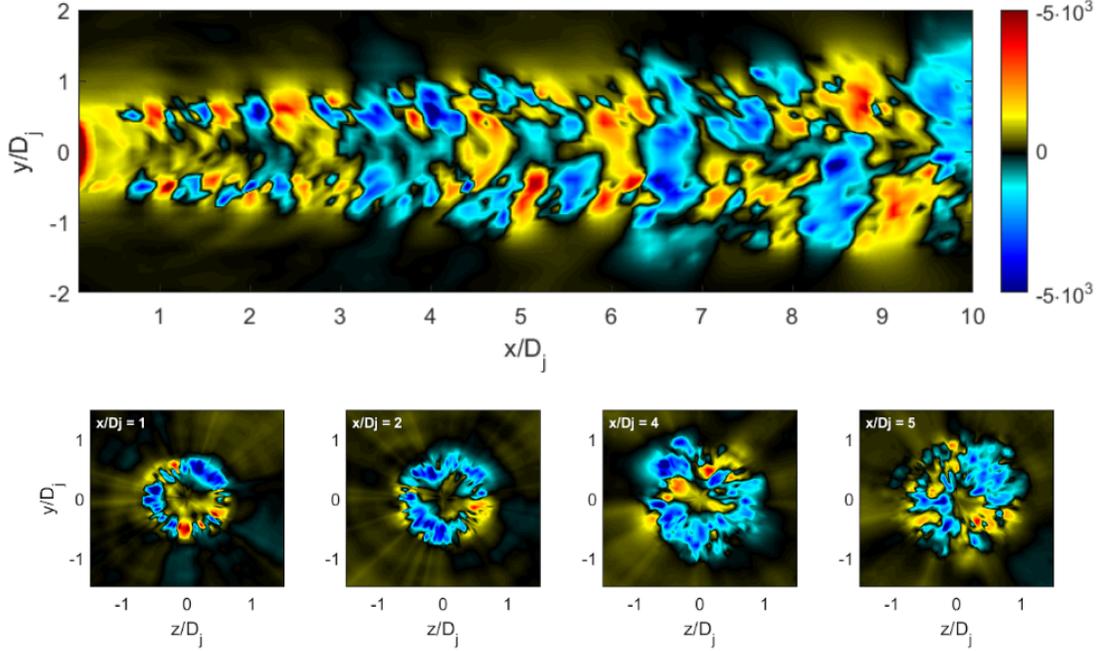


Fig. 1 Instantaneous snapshots of fluctuating pressure p' in the symmetry plane (top) and several cross planes at streamwise locations $x/D_j = 1, 2, 4, 5$ (bottom)

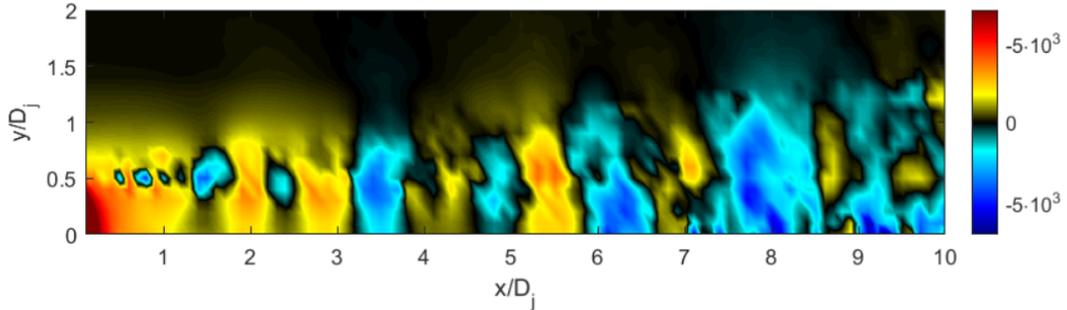


Fig. 2 Instantaneous snapshot of the the fluctuating pressure p' for the Fourier azimuthal mode $m=0$

To reduce dimensionality of the problem, the fluctuating pressure is further subjected to a Fourier transform in the circumferential direction :

$$p'(x, r, \theta, t) = \sum_m p'_m(x, r, t) e^{im\theta} \quad (2)$$

The axisymmetric mode of the Fourier decomposition ($m=0$) is of particular interest because of its importance for jet noise peak at small angles to the jet axis. A typical snapshot of the pressure fluctuation of this mode is shown in Fig. 2. The resulting pressure field, two-dimensional in space and one-dimensional in time, are then decomposed into SPOD modes using frequency domain SPOD approach [13]. The time series of mode coefficients for each frequency is then expanded as

$$\widehat{p}'_m(x, r, f) = \sum_{j=1}^N a_j(f) \phi_j(x, r, f) \quad (3)$$

where $a_j(f) = \langle \phi_j(x, r, f), \widehat{p}'_m(x, r, f) \rangle$, and $\phi_j(x, r, f)$ are the eigenvectors of the cross-spectral density tensor D defined as the Fourier transform of the space-time correlation tensor. Here, $\langle \cdot \rangle$ denotes inner product over spatial volume Ω according to $\langle \mathbf{u}, \mathbf{v} \rangle = \int_{\Omega} \mathbf{v}^*(\mathbf{x}, f) \mathbf{W}(\mathbf{x}) \mathbf{u}(\mathbf{x}, f) d\mathbf{x}$ with \mathbf{W} being a positive-definite Hermitian tensor of appropriate dimension and asterisk superscript being the Hermitian complex conjugate. To calculate the cross-spectral density tensor from the original space-time signals, a Discrete Fourier Transform (DFT) is applied using Welch's method [35], to help the convergence of spectral estimates [13]. In this method, DFT is applied on N_b overlapping blocks of the original space-time data with N_f snapshots in each block. The cross-spectral density tensor at frequency f_k can be approximated by the average

$$\mathbf{D}_{f_k} = \frac{\Delta t}{s N_b} \sum_{n=1}^{N_b} \left(\widehat{p}'_m(x, r, f_k)^{(n)} \right) \left(\widehat{p}'_m(x, r, f_k)^{(n)} \right)^* \quad (4)$$

where Δt is the sampling time between snapshots, $s = \sum_{j=1}^{N_f} \omega_j^2$ and ω_j is a window function to reduce spectral leakage due to non-periodicity of the data in each block. The corresponding time coefficients at each frequency and for each mode are computed here using the recent method developed in Nekkanti and Schmidt [39]. These time coefficients are denoted as $a_j^f(t)$ in the rest of the paper.

Notably, the possibility to extract flow information and associate it to specific frequencies, which is important for aeroacoustics phenomena naturally analysed in the frequency domain, motivates the interest in the frequency-domain SPOD method. Figure 5 shows the spatial distribution of the first SPOD modes for four typical frequencies. The frequencies are reported in terms of Strouhal number $St = f D_j / U_j$, where f is dimensional frequency. Similar to the SPOD analysis of a similar turbulent round jet flow in the literature [17], the spatial mode distributions tend to acquire finer structures, which move closer to the initial jet shear layers as the frequency increases.

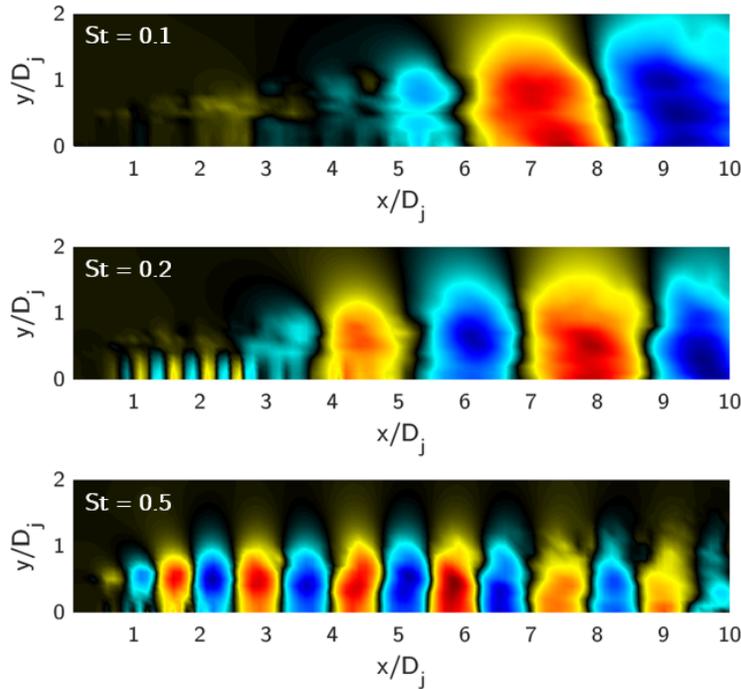


Fig. 3 Spatial distributions of the frequency-domain SPOD mode 1, $\psi(x, r, f)$ for the first Fourier azimuthal mode $m = 0$ of the fluctuating pressure p' computed for $St = [0.1, 0.2, 0.5]$.

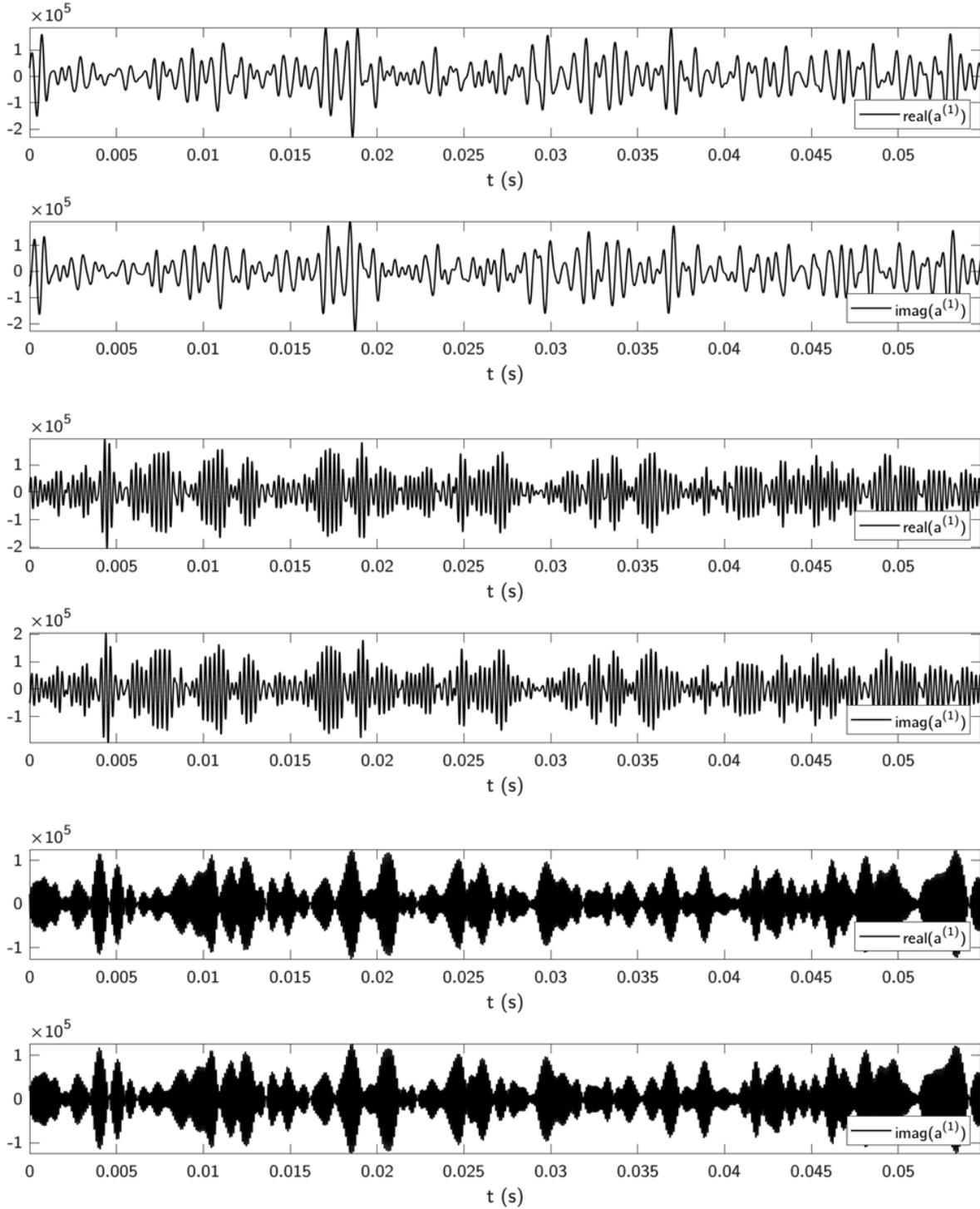


Fig. 4 Time histories of the temporal mode coefficients of the frequency-domain SPOD mode 1, $a_1(t)$ for the Fourier azimuthal mode $m = 0$ of the fluctuating pressure p' computed for $St = [0.1, 0.2, 0.5]$.

Figure 4 shows the temporal mode coefficients of the corresponding SPOD modes for the same four frequencies as in Fig.3. The temporal coefficients of the SPOD modes show broadband behaviour, and their dominant oscillation frequencies tend to increase as the reference frequency of the SPOD mode increases. This behaviour is typical of

the frequency-domain SPOD method, where different flow realisations, which correspond to different time scales are decoupled as separate SPOD modes [11].

The above temporal mode coefficients extracted from the same pressure time series using the two SPOD techniques at different filtering and frequency parameters are the signals to be reconstructed, and extrapolated using the Machine Learning approaches. The following subsections detail these machine learning techniques. The Power Spectral Density (PSD) for the expansion coefficient associated with the leading SPOD mode at $St = 0.1, 0.2,$ and 0.5 are calculated in accordance with [13], complemented by an analysis of sensitivity related to the length of the initial signal. Figure 5 demonstrates the comparison of the PSD obtained using the Welch’s method, both for the signal’s full length and for half of its total length.

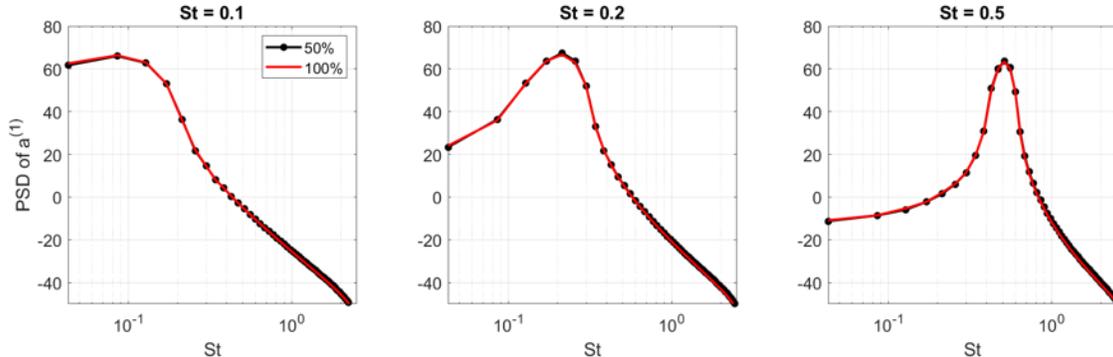


Fig. 5 PSD of the individual expansion coefficients of the leading SPOD mode at $St = [0.1, 0.2, 0.5]$ calculated for 100% and 50% of the signal length.

III. Machine Learning approaches

A. Artificial Neural Networks

Artificial Neural Networks (ANNs) is a popular tool from the machine learning paradigm that has properties of a universal function approximator [40, 41]. By construction, ANN is a network of interconnected nodes where each node represents a set of weights and biases, used to obtain a linear combination of the inputs to that node. An activation function then introduces non-linearity to the output and the network as a whole represents a parameterised nonlinear function [37]. The weights and biases being optimizable by gradient descent enable the network to be fitted onto a dataset with arbitrary dimensionality. The optimization process is essentially the ‘training’ phase and once trained, the network then acts as a smooth differentiable function whose computational cost for forward calculation is cheaper than the training steps.

In this work, the ANN framework developed by Lui and Wolf [42] is applied to test ANN’s performance, where the signal derived from Proper Orthogonal Decomposition (POD) of transient flow data is extrapolated in time to enable stable reconstruction of data for a longer time. Temporal signals of the POD modes are treated as a solution of a time-domain Ordinary Differential Equation (ODE) problem where ODE plays the role of a dynamical system model implicit in Navier Stokes equations. The reconstruction is done by learning the time-gradients of signals from the data using Feedforward Neural Networks (FNN). Following Lui and Wolf [23], the input to FNN at each time step is the POD time-coefficient and the output is the reconstructed time gradient, which is then used to extrapolate the signal in time by numerical integration using a high-order Runge-Kutta scheme. In the suggested algorithm [23], a 10th order-accurate compact scheme is used for obtaining the reference gradient signal from a suitably normalised input signal. The computed reference values are then used for training in the FNN-based regression module where a random search algorithm was implemented to explore the possible network configurations to find a suitable set of hyperparameters. The random search algorithm was designed to find an optimal network structure, which can best represent the derivative field in the domain. The space of degrees of freedom for the search algorithm was selected so as to limit the total computational time in line with recommendations in the ANN literature. For example, the random search algorithm was limited to 100 random models. The range of values allowed for the number of hidden layers

was from 6 to 10, the number of elements per layer (layer width) was in the range of 10 to 64 and the regularisation parameter was set in a range of 10^{-2} to 10^{-6} . A total of 800 time steps were then used for training all models and cases.

For each of the random configurations generated, the mean absolute training error at the last training iteration was used as the performance metric. Once the network structure is finalised by the search algorithm through minimising the mean absolute training error, the chosen network reconstructs the time-gradient values. The latter values are used as the right-hand-side of the ODE, which is solved with an explicit fourth order Runge-Kutta method to advance the signal in time. The above numerical framework, commonly called the Neural ODE, is equivalent to solving a time evolving system,

$$\frac{\partial p(t)}{\partial t} = NN(p(t)) \quad (5)$$

where $p(t)$ is the time-varying signal vector of POD coefficients and $NN(\cdot)$ is the neural network prediction for the time gradient.

Hence, while operating on a select number of modes as simultaneous input to the ANN, the solution process is equivalent to time integration along a high-dimensional trajectory along the signal coordinates corresponding to the POD mode coefficients and the stability criterion is that the system approaches a limit cycle, equivalent to a multi-dimensional dynamical system. On one hand, to avoid intersection points of trajectory, which are detrimental to the quality of training dataset due to target multiplicity for the same collocation points, the domain should be sufficiently sparse. On the other hand, a largely sparse domain will also leave the network untrained with spurious gradients in the regions where the optimizer does not operate. These random gradients divert the reconstructed signal out of a stable cycle and lead to a diverging signal. Hence, as guided by previous empirical studies, the ANN model corresponding to the system of ODEs of 8 POD signals as input is reconsidered in this work.

The suggested ANN approach will now be applied for the frequency-domain spectral POD coefficients of the pressure fluctuations of the jet flow described in Section 2.1.

B. Genetic Programming

Genetic Programming (GP) [40, 41, 43, 44] is a method that originates from the class of Genetic Algorithms (GA) with the goal of addressing symbolic regression problems. While GA typically uses a string of numbers to represent the solution, GP creates a population of models, usually in the shape of graphs or trees, which are iteratively evolved and improved through the application of so-called genetic operators. The ultimate goal of GP is to solve an approximation problem by searching through highly fit models in the space of all possible approximations that can be generated combining variables and functions provided by the user, referred to as “primitives” or “building blocks”. If a random sampling and clustering of primitives result in poor exploration of the space of all possible approximations, GP relies on an iterative process inspired by natural evolution to steer the assembling of new models (so called search in the model space) towards expressions that might minimise the selected error metrics, e.g. Root Mean Square Error (RMSE), on the training data set. By definition, models returned by GP are symbolic expressions and are global, i.e. they apply on the whole variable range defined by the training data set. The initial step of a GP run is the creation of the initial population of approximations, which is performed either randomly or deterministically exploiting preexisting knowledge. In the former case, the generation of a set of GP models (or “trees”) varying in shape and size is based on random or pseudo-random events to extensively explore the space of models defined by the building blocks/primitives provided by the user. In the latter, existing approximations are used to “seed” the initial population in the hope that GP might further refine them. Once the initial population is generated, the evolution of a highly fit model proceeds through the iterative action of genetic operators and model evaluation by a fitness function, which determines the quality of the evolved model and selects the set of models that undergo the next round of genetic operations. The procedure always leads to a stable approximation provided that primitives are selected properly. For example, addition, subtraction, multiplication, sine and cosine are allowed for a bounded model to be extrapolated.

Models’ fitness, which is typically an RMSE or Mean Square Error (MSE) - based quality score, has a dominant role in steering the evolution towards better and better solutions. It depends directly and critically on the values of numerical coefficients of the models. Therefore it is of special importance to distinguish the mathematical structure of a model as a separate object, defined as the model tree stripped of the numerical coefficients. As an example, Figure 7a shows the mathematical structure of the full model $a_3 + (a_1 z_1 / z_2 + a_2 z_3)^2$, which is shown in Fig.6.

The genetic programming method used in this paper is the HyGP implementation developed by Armani [43, 44]. Still founded on the basic principles and guidelines provided by Koza [40], the current HyGP implementation can be classified in the subcategory of “hybrid GP” or “memetic GP” [45, 46] as a result of its additional features such as

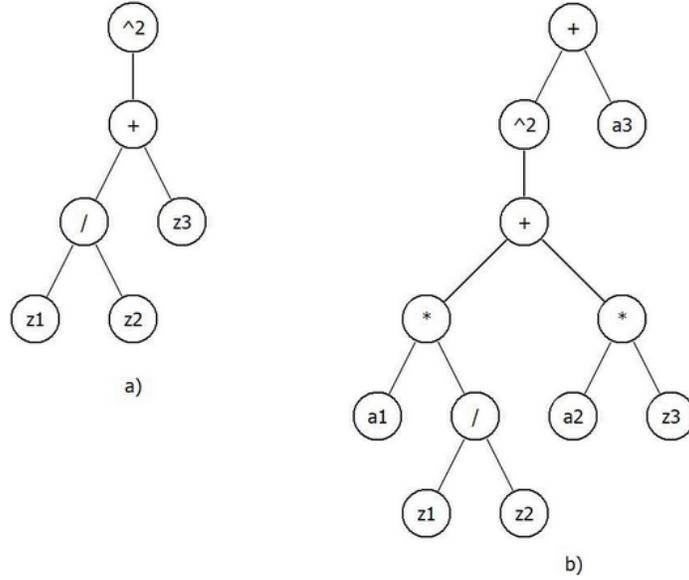


Fig. 6 Example of a GP model and its mathematical structure: (a) the mathematical structure where the tree is in a “parameterless” state and (b) the corresponding full model.

the exploitation of Baldwinian learning. In accordance with the latter approach, the classic genetic operators such as reproduction, mutation and crossover are still used, but they are applied on the “parameterless models”, i.e. the mathematical structures of the approximations, such as in the example depicted in Fig.6a, as opposed to the standard GP methods which apply genetic operators on the full models Fig.6b. Consequently, the numerical coefficients are not modified by genetic operators and are instead optimised independently. The optimisation of the numerical coefficients of HyGP is performed through a separate two-step process: first by using a population-based optimisation algorithm (Particle Swarm Optimisation [47]) to robustly explore the search space and then by applying a deterministic method (Sequential Quadratic Programming or SQP [48]) to fine tune the values of the parameters. A schematic description of the logical steps featured in the suggested HyGP algorithm is sketched in Fig.7.

Since GP/HyGP are relatively less well-known in comparison to ANN especially in the area of fluid dynamics, additional details on the technical implementation of the HyGP algorithm are provided in Appendix A. Most of the hyperparameters defining the HyGP setup (Table 2 in Appendix A) were selected based on previous experience with HyGP and hence no specific optimisation procedure other than trial and error was adopted. An optimisation strategy was used to calibrate the fitness function coefficient a_5 (Eqs. A3-A4 in Appendix A) that defines the relative contribution of the difference in statistical signal properties such as variance and mean over the entire design space between the GP prediction and the input signal. The optimal value of the coefficient was selected running preliminary GP regression tests on the pressure signal training dataset. A Pareto front was used to understand the balance between two objectives, local accuracy (RMSE) and the variance of the evolved models. The final coefficient value was obtained using a posteriori selection from the Pareto subset closest to the origin of the axes of the objectives. In most cases the selected optimal value was $a_5 = 0.01$. To check sensitivity of this parameter, these results were compared with the GP results for other values of the parameter, such as $a_5 = 0.001$, which in turn corresponds to a higher RMSE contribution. It should be noted that the optimal definition of the fitness function objective coefficients of the GP method as well as the choice of hyperparameters themselves is a weak point of the current HyGP implementation as a modelling tool, which critically relies on the user’s experience.

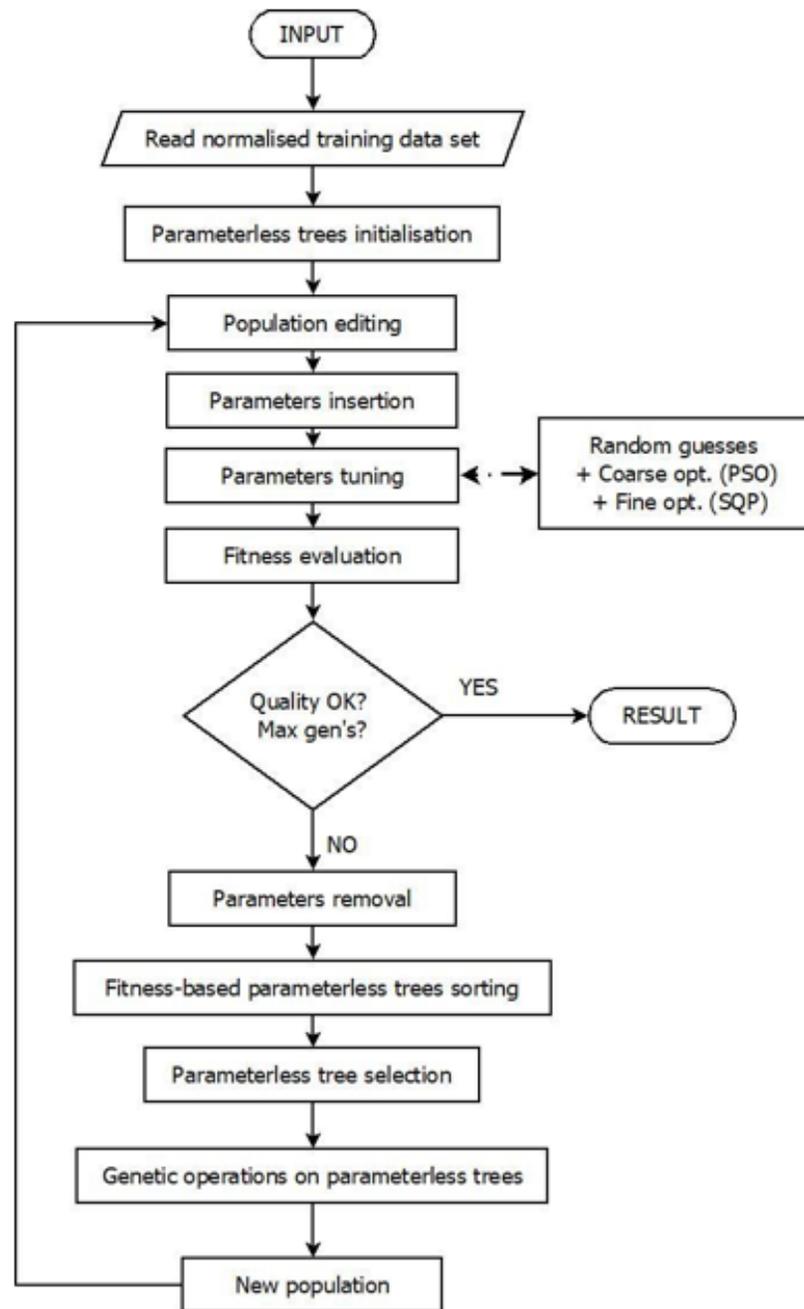


Fig. 7 Flowchart of HyGP algorithm.

C. Reservoir computing and Echo State Networks (ESNs)

In recent years, Reservoir Computing (RC) has garnered interest as a machine learning approach for estimating and forecasting dynamical systems. RC algorithms utilise a recurrent neural network (RNN) comprising of interconnected neurons known as the reservoir. The reservoir is fed using an input layer providing with the observed data and an output layer that evaluates the network states [49]. Seminal papers associated with reservoir computing [50–52] introduced Echo State Networks (ESNs) which is one class of preliminary forms of reservoir computing as an effective Machine

Learning paradigm for processing sequential data and demonstrated its effectiveness in time-series prediction.

ESNs deploys a high-dimensional numerical oscillator called the Reservoir resembling a Recurrent Neural Network. Given a datapoint from an input timeseries, called the driver signal, the reservoir matrix is used to transform the data to a required number of features which are then weighted and read out of the reservoir to synthesize an output signal. The feature space in a reservoir, which is an RNN, are the output from each of the recurring layers. The teacher signal is fed into the output neurons of the network, which feed the reservoir using the output feedback mechanism. The reservoir then operates recurrently to generate feature outputs in each of its recurring layers, which are then read out by the same output layer which feeds the reservoir. It has been shown that after an initial transient operation, the output layer tends to adapt the feedback to the reservoir in such a way that the internal recurrent layers 'echoes' the relevant features for the teacher signals [53]. Such a method differs from a conventional Neural Network in the fact that the activation of neurons are controlled by their weights in the nodes for a Neural Network which are not all active to the same magnitude during a forward propagation. But for ESNs, the reservoir is actively generating the feature outputs in an RNN fashion, thus resembling an artificial dynamical system, whose state spaces are sampled to approximate a given signal.

ESN was initially proposed as an alternative to address the exploding and vanishing gradient problems encountered by RNNs by Jaeger [54]. Also, ESNs have already been demonstrated as a viable method to learn temporal behaviour of chaotic phenomena and to extrapolate the signal beyond their Lyapunov times. In recent years, ESNs have been increasingly employed in the realm of fluid dynamics, as evidenced by various studies including the works of Huhn and Magri [55], Doan et al. [56], Margazoglou and Magri [57], Racca and Magri [58]. However, the method is not explored in the context of reduced order modelling and for the same of high Reynolds number turbulence as in this work. Since the model order reduction results in simplified modes and time series signals whose properties can be attributed to specific physical phenomena, as in the case of SPOD, utilising ESNs ability to capture their stochastic behaviour can aid in high-fidelity reconstruction of turbulent fields using few relevant reduced order modes.

The ESN implemented in this study follow the principles and methods described in Jaeger et al.'s [59] work on deploying ESNs with leaky integrator neurons. The implemented structure of the ESN consists of an input layer, a reservoir consisting of N neurons and the output layer. The appropriate size of reservoir (N) should consider both the length of the training data, denoted as T , and the complexity of the task at hand. Generally, N falls within the range of $T/10$ to $T/2$ [60]. The input layer of the ESN receives the input signals, and the output layer generates the output signal.

For testing the ESN model, we generate a network with 800 reservoir neurons with sparsity of 10%. The model was trained with 2000 time steps, followed by testing over 3348 time steps, employing different combinations of global control parameters to achieve optimal prediction of time coefficient of SPOD modes. The training stage of ESN encompasses minimization of mean squared error between the input signal and the target output.

The selection of control parameters in the ESN model poses a challenging task. These hyperparameters play a crucial role in determining the performance and behaviour of the model. Several studies [61–63] incorporated supplementary optimization methods like Bayesian optimization to improve the performance of ESN. However, these optimization techniques failed to offer a comprehensive analysis of hyperparameter sensitivities and struggled to effectively tackle the prediction task in the current investigation. One key control parameter is the spectral radius which is the maximum eigenvalue of the reservoir matrix (W) and its value should be smaller than unity to satisfy the echo state property [59]. Another important control parameter is the input scaling which influences weighting of the input vector, and it was observed from experiments that higher values of input scaling resulted in amplified signal amplitudes making the model sensitive to noise in the input data. Furthermore, the leakage rate of the network determining the speed at which internal states of the reservoir evolves is also a significant control parameter that governs the reservoir dynamics. While the aforementioned control parameters are pivotal in determining the dynamical characteristics of the ESN reservoir, the size of reservoir N ascertain the capacity of the model [59]. Nevertheless, larger reservoir sizes may result in overfitting and also higher computational costs.

Our approach of modelling the reduced order turbulent signal focuses on the extrapolation capability. Hence unlike conventional approach of training ESN to approximate the signal for the time span of the training data is not sufficient. Alternatively, we train ESN using bandpassed white noise signal deployed as a driver signal. This ensures that for extrapolation, any length of the white noise signal can be generated independently, filtered to the same bandwidth and used to drive the ESN prediction. For training using bandpassed white noise, the input layer is supplied with the noise signal and the POD mode coefficients are supplied as the teacher signal. Hence the process is equivalent to a least square fit of the teacher signal using the latent space provided by the white noise which is ideally infinite. The selection of features and their weighting to approximate the signal via the echo state property is hence the training process. During extrapolation, the same band passed white noise signal but with longer time duration is used to drive the reservoir and the output is recorded as the prediction. The schematic of a complete framework depicting the deployment of Echo

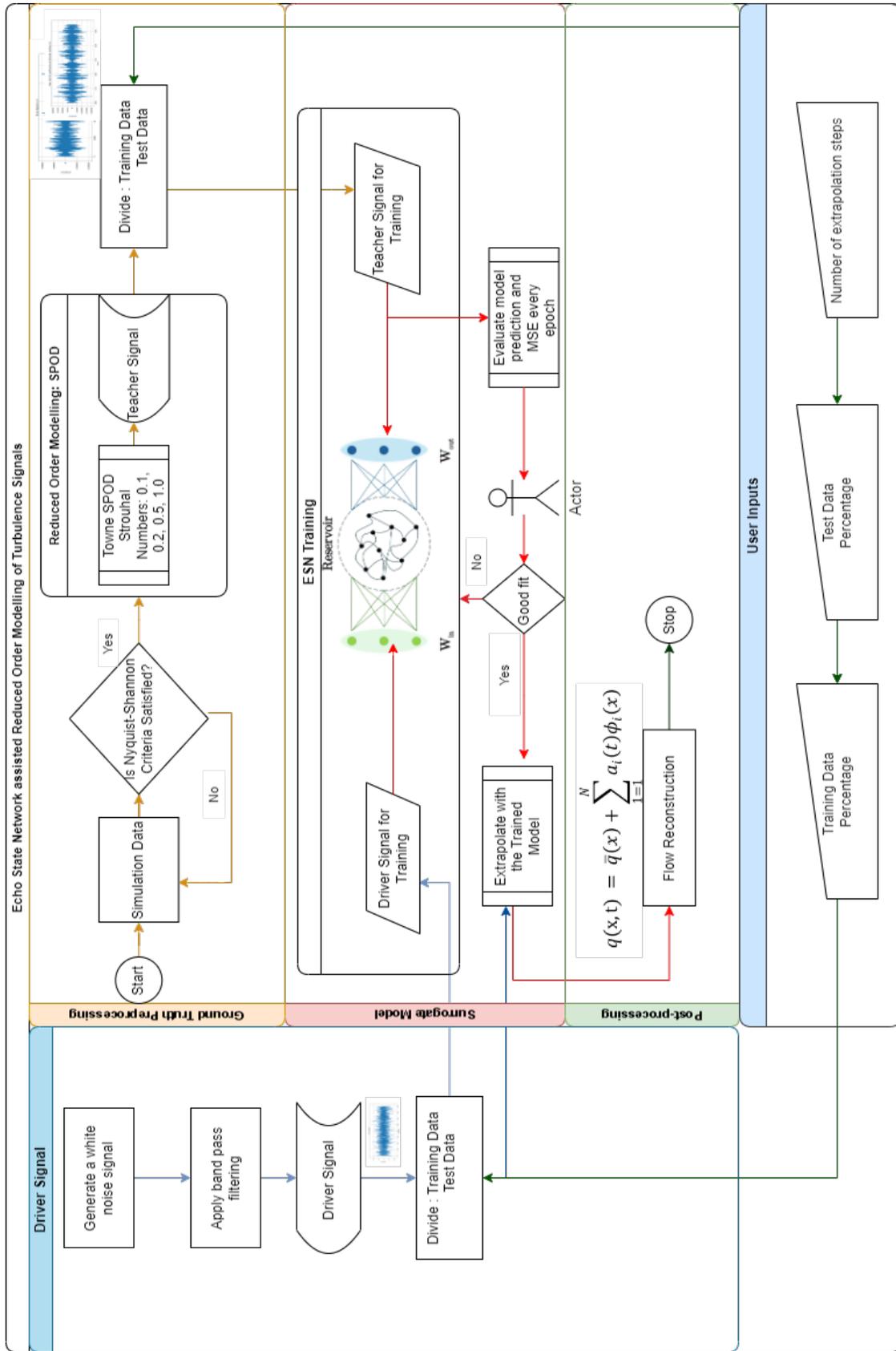


Fig. 8 Schematic diagram of ESN based reduced order modelling of turbulence signals

State Networks (ESNs) to assist reduced-order turbulence modelling is given in Fig 8.

That being the underlying mechanisms for the three ML approaches, we now present the results as they are deployed for the high Re turbulence signals.

IV. Results

The frequency-domain decomposition[13] requires a long enough input time history for the output dataset to reach statistical convergence. Hence, for comparing the ANN and GP methods with the frequency-domain SPOD method, the entire available dataset of 5348 points (Table 1) was used to generate the SPOD time coefficients, in this case for a range of frequencies corresponding to Strouhal numbers from $St = 0.1$ to $St = 1.0$. The first 3000 time samples were used for the training data set, and the remaining part is used for testing. As frequency of the SPOD signal increases, the reconstruction at a fixed data sampling rate becomes increasingly more challenging. Hence, the ML methods are first tested on the low frequency SPOD signal, which corresponds to the peak of the near-field pressure spectrum of high speed jets at $St = 0.1$ and then moving to higher frequencies.

A. Using ANN method

The ANN method applied to the original SPOD signal at $St = 0.1$ without any filtering leads to a fast divergence of the numerical prediction. In an attempt to stabilise the ANN solution, an additional limited bandwidth filtering around the central frequency $St = 0.1$ (1787 Hz) was applied to investigate sensitivity of performance of the ANN method to spectral complexity of the input signal. Result of applying the ANN method to the limited-bandwidth filtered SPOD signals at frequency bands of 1000 Hz are shown in Fig.9. While the ANN method is able to reconstruct the SPOD signal filtered with the much lower frequency window, the solution rapidly diverges when the window is extended to 1000 Hz. Hence, it is concluded that the ANN method originally suggested by Lui and Wolf [23] for turbulent flow signal reconstruction is too unstable to be applicable for the time-frequency analysis of the high-speed jet flow, which is characterised by a rich spectral content. The divergence of ANN based extrapolation, to our knowledge, is understood as due to the accumulation of error with each time step prediction. This is due to Neural ODE framework which predicts the gradient of the signal as a function of state value which inevitably accrues a numerical error at best. Hence such a framework does not theoretically guarantee a stable extrapolation to enable prediction for large time spans.

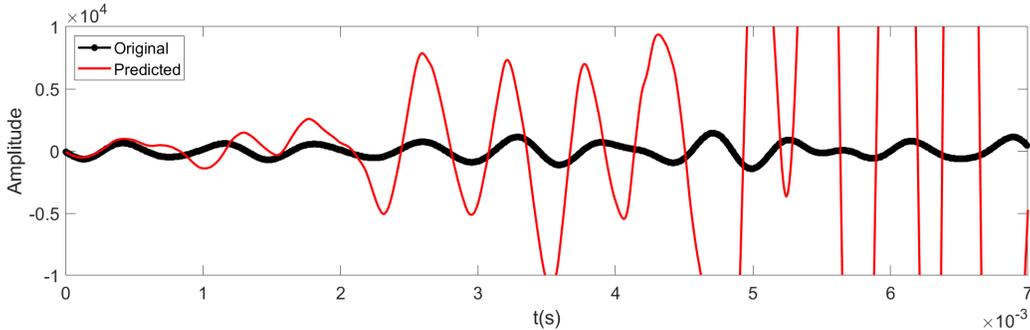


Fig. 9 ANN solution of the bandwidth limited temporal coefficient of the first SPOD mode at $St = 0.1$ at a filter frequency window of 1000 Hz. The ANN method consistently showing diverging behaviour with unconstrained growing amplitude.

B. Using GP method

Figures 12 and 13 compare predictions of the HyGP method of the SPOD signals for frequencies, $St = [0.1, 0.2, 0.5]$ with the input signals in the time and frequency domain. Appendix B provides the corresponding numerical expressions in each case as well as the computing time required by the GP model to generate these expressions during the training process. One of the advantages of the GP model is that after the training is completed it provides an explicit numerical description of the reconstructed time signal. In comparison to the ANN method, the GP solution was found stable for all frequencies for the entire duration of tested SPOD time signals. As noted in Section 3.1, the robustness of the GP method can be attributed to the lack of reliance on the smoothness/integrability property of input signals.

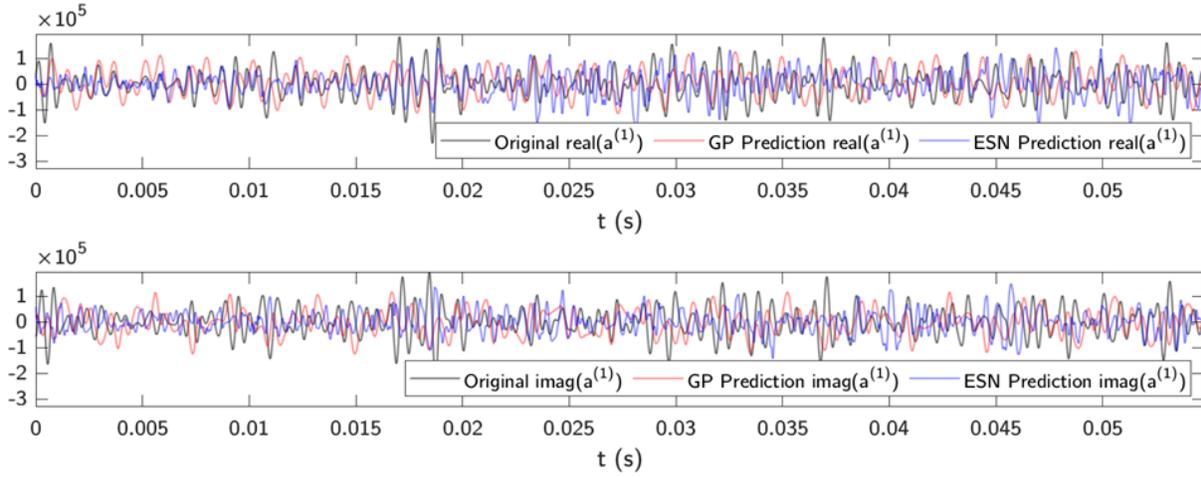


Fig. 10 Comparison of the HyGP and ESN reconstructions of the real (top) and imaginary (bottom) parts of SPOD time coefficients at Strouhal number $St = 0.1$ against the reference signal.

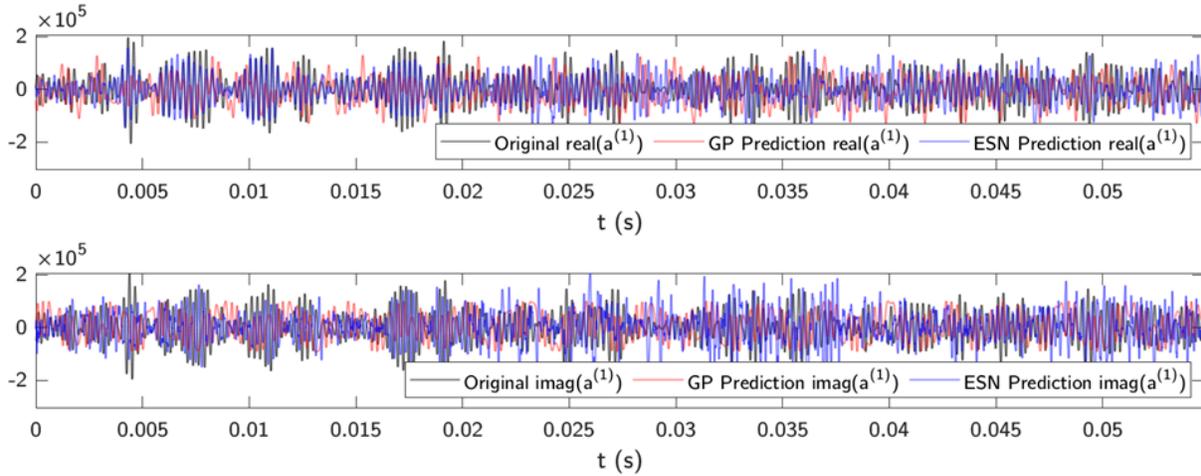


Fig. 11 Comparison of the HyGP and ESN reconstructions of the real (top) and imaginary (bottom) parts of SPOD time coefficients at Strouhal number $St = 0.2$ against the reference signal.

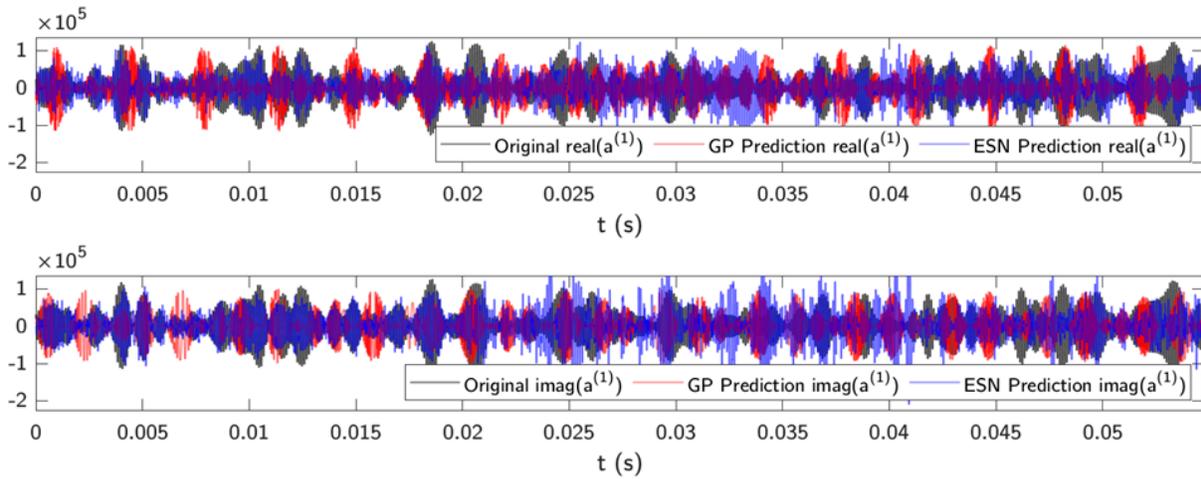


Fig. 12 Comparison of the HyGP and ESN reconstructions of the real (top) and imaginary (bottom) parts of SPOD time coefficients at Strouhal number $St = 0.5$ against the reference signal.

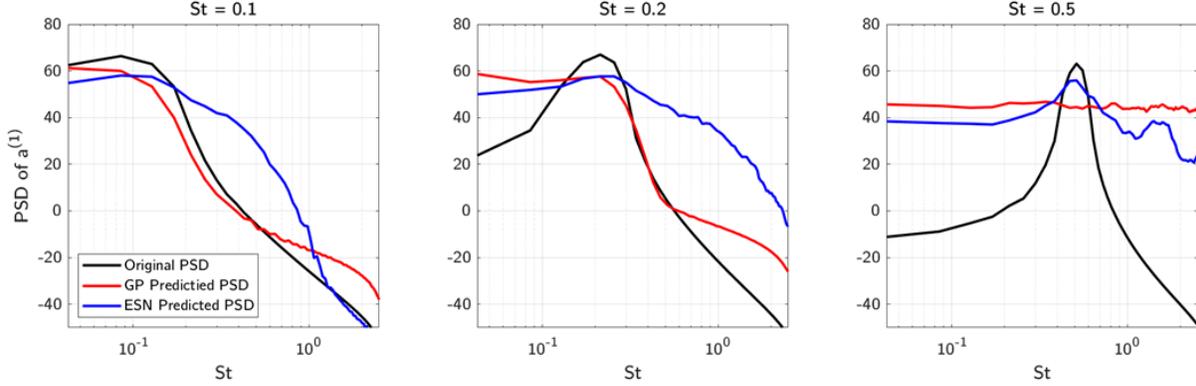


Fig. 13 PSD comparison of the combined complex time coefficients, predicted using GP and ESN, for the leading SPOD mode at $St = [0.1, 0.2, 0.5]$, against that of original signal. GP predictions are seen to maintain higher fidelity compared to ESN predictions. For higher Strouhal numbers, the ESN predictions tend to perform better with tendency to capture the correct central frequency.

It can be noted that predictions of the GP method for the SPOD signals at the low frequency $St = 0.1$ and at the dominant frequency of high-speed jet noise at the far field, $St = 0.2$, are in good agreement with the reference signals. However, as the central frequency of the SPOD signal increases further to $St = 0.5$ and $St = 1$, the GP solution fails to capture the high-frequency part of the spectra. The lack of accuracy for high frequency signal reconstruction of the GP method can be associated with a limitation of the data sampling rate. In the current numerical experiments, it was found that the GP model requires at least 20 time points per period of the frequency of interest for good reconstruction of statistics of a turbulent signal. It can be noted that, despite the high-frequency limitation, performance of the current GP-based reduced-order model of the first SPOD coefficient of axisymmetric pressure mode may already be sufficient for capturing the dominant dynamics relevant for peak jet noise at small angles to the jet flow axis. A detailed error analysis of the reconstructed pressure solution of the GP method is provided in the following section.

C. Using Echo State Networks

The results of ESN predictions are shown together with GP predictions and original signals in Fig.12 and 13. The ESN were trained using first 2000 time points from the dataset. The signal was then extrapolated to 5348 time steps, to compare against the full available dataset and was then used to extrapolate to 40,000 time steps to check its stability. It has been found that the networks prediction showed reasonable agreement with the training data and for comparison with the 5348 timesteps for data in the time domain. The signal comparison of the frequency domain also has a reasonable agreement and better agreement with data than ANN and GP. For extrapolation, the network showed stable reconstruction capability without exploding or vanishing amplitude as would be expected from typical ANN/RNN performances. However, on reconstruction of the full signal using reconstructed signals for both imaginary and real parts of the selected SPOD Strouhal numbers, ESNs tend to generate spurious broadband components. The performance of ESN indicate a higher preference to learn the higher Strouhal number data as compared to lower Strouhals and is able to provide a stable reconstruction in all cases.

V. Conclusions

Three machine learning techniques, Artificial Neural Network (ANN) and Genetic Programming (GP), and Echo State Networks (ESNs) are implemented for reconstruction of turbulent pressure fluctuations from Large Eddy Simulations of a Mach 0.9 isolated round jet at Reynolds number 10^6 corresponding to the NASA SHJAR experiment. The focus of reduced-order modelling has been to capture the dominant flow dynamics corresponding to the jet noise peak at small angles to the jet flow axis. Hence, the goal of reduced-order modelling here has been to reconstruct the most energetic component of the axi-symmetric pressure mode, as extracted using Proper Orthogonal Decomposition (POD). Two Spectral POD techniques were applied, which correspond to the time- and frequency-domain analyses as suggested in the literature. The time-domain POD uses a filtering of the input time signals in the correlation matrix space, thereby making the signal narrower band and smoother in comparison with the POD signals decomposed frequency-by-frequency for

the same length and sampling rate. At the same time, the frequency-domain POD signals support important space-time correlations as typical of spatiotemporal turbulent structures.

The ANN method, which is based on learning from the data and solving a set of effective Neural Ordinary Differential Equations was found insufficient to reconstruct the time modal coefficients of the frequency-domain SPOD signal even for the lowest considered frequency, i.e. $St = 0.1$. The inherent broadband nature of the SPOD signal is reasoned for the ANN instability in this case.

Attempting to model the signals using ESNs were more successful in comparison with ANN approach. While using white noise driver signal, the network was able to successfully capture the broadband central frequencies. However, the predicted signals tend to be more broadband than the teacher signal with the effect more prominent for low frequency signals. The effect was found to improve for higher frequencies where GP predicted signals are deteriorating in their fidelity. Nevertheless, the spurious broadband components from each mode predictions are a significant error contribution towards the final reconstruction of the flow and hence the framework needs further improvement in terms of training dynamics.

In comparison with this, the lack of band limited behaviour and signal smoothness that proved critical for ANN did not impair the GP capability to return a stable approximation model for all frequencies. In this regard, GP proves to be a robust modelling technique, which allows to accurately capture the low-frequency mode of the turbulent signal that is typical of the dominant flow dynamics relevant for noise generation at small angles to the jet axis. However, accuracy of GP deteriorates at high frequencies, approximately from $St = 0.5$ where ESN prediction tend to still capture the central frequencies.

While the GP's stochastic nature brings increased robustness to the model in comparison with ANN and ESN, the same also determines an increase of the quality assessment effort. Evolution implies variability of final models, and so the selection of the best model has to be necessarily done manually or automatically. For example, automatic strategies for hyperparameter optimisation, such as Random Search used for ANN, could represent a possible approach. If the final application of the GP model is known, specific and quantifiable requirements and their hierarchy of importance are known, specific objectives can be incorporated in the fitness function in order to promote the generation of fit-for-purpose models, as done in this study. Alternatively, the final selection might rely on the creation of a Pareto front corresponding to all the models generated by GP. Last but not least, it is worth noting that GP provides an explicit expression for the reconstructed signal at the end of the GP process, regardless of the empirical adjustments used for its generation. Such expressions can be used as a surrogate model of turbulent fluctuating quantities of interest as an improved nonlinear forcing model in linearised modelling frameworks as well as to accelerate convergence of statistical source models in acoustic analogies.

References

- [1] Brunton, S. L., Noack, B. R., and Koumoutsakos, P., "Machine Learning for Fluid Mechanics," 2019.
- [2] Brunton, S. L., and Noack, B. R., "Closed-loop turbulence control: Progress and challenges," *Applied Mechanics Reviews*, Vol. 67, No. 5, 2015, p. 050801.
- [3] Naghibi, S. E., Karabasov, S. A., Jalali, M. A., and Sadati, S. H., "Fast spectral solutions of the double-gyre problem in a turbulent flow regime," *Applied Mathematical Modelling*, Vol. 66, 2019, pp. 745–767.
- [4] Naghibi, S. E., Karabasov, S. A., and Kamenkovich, I., "Zonal jets in the Southern Ocean: A semi-analytical model based on scale separation," *Ocean Modelling*, Vol. 183, 2023, p. 102198.
- [5] Lee, M., Zhu, Y., Li, L. K., and Gupta, V., "System identification of a low-density jet via its noise-induced dynamics," *Journal of Fluid Mechanics*, Vol. 862, 2019, pp. 200–215.
- [6] Sieber, M., Paschereit, C. O., and Oberleithner, K., "Stochastic modelling of a noise-driven global instability in a turbulent swirling jet," *Journal of Fluid Mechanics*, Vol. 916, 2021, p. A7.
- [7] Foldes, R., Camporeale, E., and Marino, R., "Low-Dimensional Representation of Intermittent Geophysical Turbulence with High-Order Statistics-Informed Neural Networks (H-SiNN)," *Physics of Fluids*, Vol. 36, No. 2, 2024, p. 026607. <https://doi.org/10.1063/5.0179132>.
- [8] Taira, K., Hemati, M. S., Brunton, S. L., Sun, Y., Duraisamy, K., Bagheri, S., Dawson, S. T., and Yeh, C.-A., "Modal analysis of fluid flows: Applications and outlook," *AIAA journal*, Vol. 58, No. 3, 2020, pp. 998–1022.

- [9] Begiashvili, B., Groun, N., Garicano-Mena, J., Le Clainche, S., and Valero, E., “Data-driven modal decomposition methods as feature detection techniques for flow problems: A critical assessment,” *Physics of Fluids*, Vol. 35, No. 4, 2023.
- [10] Lumley, J. L., “The structure of inhomogeneous turbulent flows,” *Atmospheric turbulence and radio wave propagation*, 1967, pp. 166–178.
- [11] Sieber, M., Paschereit, C. O., and Oberleithner, K., “Spectral Proper Orthogonal Decomposition,” *Journal of Fluid Mechanics*, Vol. 792, 2016, pp. 798–828. <https://doi.org/10.1017/jfm.2016.103>.
- [12] Ribeiro, J. H. M., and Wolf, W. R., “Identification of coherent structures in the flow past a NACA0012 airfoil via proper orthogonal decomposition,” *Physics of Fluids*, Vol. 29, No. 8, 2017.
- [13] Towne, A., Schmidt, O. T., and Colonius, T., “Spectral Proper Orthogonal Decomposition and Its Relationship to Dynamic Mode Decomposition and Resolvent Analysis,” *Journal of Fluid Mechanics*, Vol. 847, 2018, pp. 821–867. <https://doi.org/10.1017/jfm.2018.283>.
- [14] Karami, S., and Soria, J., “Analysis of coherent structures in an under-expanded supersonic impinging jet using spectral proper orthogonal decomposition (SPOD),” *Aerospace*, Vol. 5, No. 3, 2018, p. 73.
- [15] Rowley, C. W., Colonius, T., and Murray, R. M., “Model reduction for compressible flows using POD and Galerkin projection,” *Physica D: Nonlinear Phenomena*, Vol. 189, No. 1-2, 2004, pp. 115–129.
- [16] Brunton, S. L., Proctor, J. L., and Kutz, J. N., “Discovering governing equations from data by sparse identification of nonlinear dynamical systems,” *Proceedings of the national academy of sciences*, Vol. 113, No. 15, 2016, pp. 3932–3937.
- [17] Carlberg, K., Bou-Mosleh, C., and Farhat, C., “Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations,” *International Journal for numerical methods in engineering*, Vol. 86, No. 2, 2011, pp. 155–181.
- [18] Östh, J., Noack, B. R., Krajnović, S., Barros, D., and Borée, J., “On the need for a nonlinear subscale turbulence term in POD models as exemplified for a high-Reynolds-number flow over an Ahmed body,” *Journal of Fluid Mechanics*, Vol. 747, 2014, pp. 518–544.
- [19] Protas, B., Noack, B. R., and Östh, J., “Optimal nonlinear eddy viscosity in Galerkin models of turbulent flows,” *Journal of Fluid Mechanics*, Vol. 766, 2015, pp. 337–367.
- [20] San, O., and Maulik, R., “Extreme learning machine for reduced order modeling of turbulent geophysical flows,” *Physical Review E*, Vol. 97, No. 4, 2018, p. 042322.
- [21] Wan, Z. Y., Vlachas, P., Koumoutsakos, P., and Sapsis, T., “Data-assisted reduced-order modeling of extreme events in complex dynamical systems,” *PloS one*, Vol. 13, No. 5, 2018, p. e0197704.
- [22] Rudy, S. H., Kutz, J. N., and Brunton, S. L., “Deep learning of dynamics and signal-noise decomposition with time-stepping constraints,” *Journal of Computational Physics*, Vol. 396, 2019, pp. 483–506.
- [23] Lui, H. F., and Wolf, W. R., “Construction of Reduced-Order Models for Fluid Flows Using Deep Feedforward Neural Networks,” *Journal of Fluid Mechanics*, Vol. 872, 2019, pp. 963–994. <https://doi.org/10.1017/jfm.2019.358>.
- [24] Schubert, Y., Sieber, M., Oberleithner, K., and Martinuzzi, R., “Towards robust data-driven reduced-order modelling for turbulent flows: application to vortex-induced vibrations,” *Theoretical and Computational Fluid Dynamics*, Vol. 36, No. 3, 2022, pp. 517–543.
- [25] Bridges, J., and Wernet, M., “Establishing consensus turbulence statistics for hot subsonic jets,” *16th AIAA/CEAS aeroacoustics conference*, 2010, p. 3751.
- [26] Markesteijn, A. P., and Karabasov, S. A., “CABARET solutions on graphics processing units for NASA jets: Grid sensitivity and unsteady inflow condition effect,” *Comptes Rendus Mécanique*, Vol. 346, No. 10, 2018, pp. 948–963.
- [27] Goloviznin, V. M., and Samarskii, A. A., “Finite difference approximation of convective transport equation with space splitting time derivative,” *Matematicheskoe modelirovanie*, Vol. 10, No. 1, 1998, pp. 86–100.
- [28] Karabasov, S. A., and Goloviznin, V. M., “Compact accurately boundary-adjusting high-resolution technique for fluid dynamics,” *Journal of Computational Physics*, Vol. 228, No. 19, 2009, pp. 7426–7451.

- [29] Chintagunta, A., Naghibi, S., and Karabasov, S., “Flux-corrected dispersion-improved CABARET schemes for linear and nonlinear wave propagation problems,” *Computers & Fluids*, Vol. 169, 2018, pp. 111–128.
- [30] Faranosov, G. A., Goloviznin, V. M., Karabasov, S. A., Kondakov, V. G., Kopiev, V. F., and Zaitsev, M. A., “CABARET method on unstructured hexahedral grids for jet noise computation,” *Computers & Fluids*, Vol. 88, 2013, pp. 165–179.
- [31] Semiletov, V. A., Yakovlev, P. G., Karabasov, S. A., Faranosov, G. A., and Kopiev, V. F., “Jet and jet–wing noise modelling based on the CABARET MILES flow solver and the Ffowcs Williams–Hawkings method,” *International Journal of Aeroacoustics*, Vol. 15, No. 6-7, 2016, pp. 631–645.
- [32] Semiletov, V., and Karabasov, S. A., “CABARET scheme with conservation-flux asynchronous time-stepping for nonlinear aeroacoustics problems,” *Journal of Computational Physics*, Vol. 253, 2013, pp. 157–165.
- [33] Semiletov, V., and Karabasov, S., “CABARET scheme for computational aero acoustics: extension to asynchronous time stepping and 3D flow modelling,” *International Journal of Aeroacoustics*, Vol. 13, No. 3-4, 2014, pp. 321–336.
- [34] Markesteijn, A., and Karabasov, S., “Simulations of co-axial jet flows on graphics processing units: the flow and noise analysis,” *Philosophical Transactions of the Royal Society A*, Vol. 377, No. 2159, 2019, p. 20190083.
- [35] Markesteijn, A. P., Gryazev, V., Karabasov, S. A., Ayupov, R. S., Benderskiy, L. A., and Lyubimov, D. A., “Flow and noise predictions of coaxial jets,” *AIAA Journal*, Vol. 58, No. 12, 2020, pp. 5280–5293.
- [36] Gryazev, V., Kalyan, A., Markesteijn, A., and Karabasov, S., “Broadband shock-associated noise modelling for high-area-ratio under-expanded jets,” *The Journal of the Acoustical Society of America*, Vol. 150, No. 2, 2021, pp. 1534–1547.
- [37] Gauthier, D. J., Boltt, E., Griffith, A., and Barbosa, W. A. S., “Next Generation Reservoir Computing,” *Nature Communications*, Vol. 12, No. 1, 2021, p. 5564. <https://doi.org/10.1038/s41467-021-25801-2>.
- [38] Markesteijn, A. P., Kamliya Jawahar, H., Karabasov, S. A., and Azarpeyvand, M., “GPU CABARET Solutions for 30P30N Three-element High-lift Airfoil with Slat Modifications.” *AIAA AVIATION 2021 FORUM*, American Institute of Aeronautics and Astronautics, VIRTUAL EVENT, 2021. <https://doi.org/10.2514/6.2021-2115>.
- [39] Nekkanti, A., and Schmidt, O. T., “Frequency–time analysis, low-rank reconstruction and denoising of turbulent flows using SPOD,” *Journal of Fluid Mechanics*, Vol. 926, 2021, p. A26.
- [40] Koza, J., “On the programming of computers by means of natural selection,” *Genetic programming*, 1992.
- [41] Toropov, V., Polynkin, A., Armani, U., and Alvarez, L., “Application of Metamodel Building by Genetic Programming to Industrial Optimization Problems,” *IV European Congress on Computational Mechanics (ECCM IV): Solids, Structures and Coupled Problems in Engineering, Paris*, 2010, pp. 16–21.
- [42] Lui, H. F. S., and Wolf, W. R., “Construction of Reduced Order Models for Fluid Flows Using Deep Feedforward Neural Networks,” *Journal of Fluid Mechanics*, Vol. 872, 2019, pp. 963–994. <https://doi.org/10.1017/jfm.2019.358>.
- [43] Armani, U., Khatir, Z., Khan, A., Toropov, V. V., Polyinkin, A., Thompson, H., Kapur, N., and Noakes, C. J., “Control of Physical Consistency in Meta-model Building by Genetic Programming,” *Proceedings of the Second International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering*, 2011.
- [44] Armani, U., “Development of A Hybrid Genetic Programming Technique for Computationally Expensive Optimisation Problems,” Ph.D. thesis, University of Leeds, 2014. URL <https://hygprogramming.wordpress.com/>.
- [45] Martens, I., “Symbolic Regression for Space Applications: Differentiable Cartesian Genetic Programming Powered by Multi-objective Memetic Algorithms,” , 2022. URL <https://arxiv.org/abs/2206.06213>.
- [46] Kommenda, M., Burlacu, B., and Kronberger, G., “Parameter identification for symbolic regression using nonlinear least squares,” *Genetic Programming and Evolvable Machines*, Vol. 21, 2020, pp. 471–501. <https://doi.org/10.1007/s10710-019-09371-3>, URL <https://doi.org/10.1007/s10710-019-09371-3>.
- [47] Kennedy, J., and Eberhart, R., “Particle swarm optimization,” *Proceedings of the IEEE International Conference on Neural Networks*, Vol. 4, IEEE, 1995, pp. 1942–1948.
- [48] Madsen, K., Nielsen, H. B., and Sondergaard, J., “Robust subroutines for non-linear optimization,” Tech. Rep. IMM-REP-2002-02, Technical University of Denmark, 2002.

- [49] Gauthier, D. J., Bollt, E., Griffith, A., and Barbosa, W. A. S., “Next generation reservoir computing,” *Nature Communications*, Vol. 12, No. 1, 2021, p. 5564. <https://doi.org/10.1038/s41467-021-25801-2>, URL <https://www.nature.com/articles/s41467-021-25801-2>.
- [50] Jaeger, H., “The “echo state” approach to analysing and training recurrent neural networks-with an erratum note,” *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, Vol. 148, 2001.
- [51] Jaeger, H., and Haas, H., “Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication,” *Science*, Vol. 304, No. 5667, 2004, pp. 78–80. <https://doi.org/10.1126/science.1091277>, URL <https://www.science.org/doi/10.1126/science.1091277>.
- [52] Maass, W., Natschläger, T., and Markram, H., “Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations,” *Neural Computation*, Vol. 14, No. 11, 2002, pp. 2531–2560. <https://doi.org/10.1162/089976602760407955>, URL <https://direct.mit.edu/neco/article/14/11/2531-2560/6650>.
- [53] Jaeger, H., and Haas, H., “Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication,” *Science*, Vol. 304, No. 5667, 2004, pp. 78–80. <https://doi.org/10.1126/science.1091277>, URL <https://www.science.org/doi/abs/10.1126/science.1091277>.
- [54] Jaeger, H., “The “echo state” approach to analysing and training recurrent neural networks-with an erratum note,” *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, Vol. 148, No. 34, 2001, p. 13.
- [55] Huhn, F., and Magri, L., “Gradient-free optimization of chaotic acoustics with reservoir computing,” *Physical Review Fluids*, Vol. 7, No. 1, 2022, p. 014402.
- [56] Doan, N. A. K., Polifke, W., and Magri, L., “Short-and long-term predictions of chaotic flows and extreme events: a physics-constrained reservoir computing approach,” *Proceedings of the Royal Society A*, Vol. 477, No. 2253, 2021, p. 20210135.
- [57] Margazoglou, G., and Magri, L., “Stability analysis of chaotic systems from data,” *Nonlinear Dynamics*, Vol. 111, No. 9, 2023, pp. 8799–8819.
- [58] Racca, A., and Magri, L., “Data-driven prediction and control of extreme events in a chaotic flow,” *Physical Review Fluids*, Vol. 7, No. 10, 2022, p. 104402.
- [59] Jaeger, H., Lukoševičius, M., Popovici, D., and Siewert, U., “Optimization and applications of echo state networks with leaky-integrator neurons,” *Neural Networks*, Vol. 20, No. 3, 2007, pp. 335–352. <https://doi.org/10.1016/j.neunet.2007.04.016>, URL <https://linkinghub.elsevier.com/retrieve/pii/S089360800700041X>.
- [60] Jaeger, H., “A Tutorial on Training Recurrent Neural Networks, Covering BPPT, RTRL, EKF and the “Echo State Network” Approach,” GMD Report 159, German National Research Center for Information Technology, St. Augustin, Germany, Oct. 2002.
- [61] Maat, J. R., Gianniotis, N., and Protopapas, P., “Efficient Optimization of Echo State Networks for Time Series Datasets,” *2018 International Joint Conference on Neural Networks (IJCNN)*, IEEE, Rio de Janeiro, 2018, pp. 1–7. <https://doi.org/10.1109/IJCNN.2018.8489094>, URL <https://ieeexplore.ieee.org/document/8489094/>.
- [62] Racca, A., and Magri, L., “Robust Optimization and Validation of Echo State Networks for learning chaotic dynamics,” *Neural Networks*, Vol. 142, 2021, pp. 252–268. <https://doi.org/10.1016/j.neunet.2021.05.004>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0893608021001969>.
- [63] Trierweiler Ribeiro, G., Guilherme Sauer, J., Fraccanabbia, N., Cocco Mariani, V., and Dos Santos Coelho, L., “Bayesian Optimized Echo State Network Applied to Short-Term Load Forecasting,” *Energies*, Vol. 13, No. 9, 2020, p. 2390. <https://doi.org/10.3390/en13092390>, URL <https://www.mdpi.com/1996-1073/13/9/2390>.
- [64] Poli, R., Langdon, W. B., and McPhee, N. F., *A field guide to genetic programming (with contributions by J. R. Koza)*, Published via <http://lulu.com> and freely available at <http://www.gp-field-guide.org.uk>, 2008. URL <http://www.gp-field-guide.org.uk>.

Appendix A. Details of the HyGP implementation

The workflow of the HyGP algorithm includes the following elements:

Input normalisation

Training data set provided by the user is linearly transformed so that the normalised values of the independent variable (design space) and dependent variable (target values) lie within the interval of (1, 11). The normalisation of the independent and dependent variables is performed to avoid zero value, which is problematic in case the division operation is used in GP models, and also to scale the training data to a predefined design space.

Genetic Operations

Reproduction: A user-defined percentage of the sorted population (best scoring individuals) is copied into the new generation. Copies are eliminated, and to preserve population constant size the void is filled with individuals generated from scratch using the same procedure used for population initialization. Numerical coefficients of the best individual are retained and used as an additional starting guess for the tuning process in the following generation. **Crossover:** It is used to generate a user-defined percentage of the new population. Tournament selection (in which a pool of 3 randomly selected models is temporarily created and the tournament is won by the model with the best fitness) is used to select the individuals that undergo crossover (uniform probability distribution). Parent 1 selection is performed on individuals selected from the reproduction pool, whereas Parent 2 selection considers the whole population to preserve genetic variability. Crossover points are selected randomly, first selecting a random depth or level in the tree-shaped model using a uniform probability distribution in the selection and then randomly selecting one node among the ones sharing the same depth or level. If the new potential offspring have depths larger than the maximum allowed, new crossover points are selected. **Mutation:** consists of point mutation for even generations and subtree mutation for odd generations. Tournament selection (as used in crossover) is used to choose the individuals undergoing mutation. Nodes are selected randomly using the same approach adopted for crossover.

Editing

The editing operation aims at turning all GP models that grow indefinitely in absolute value (diverge) into bounded models. Prior to parameters insertion and fitness evaluation, each parameterless individual is scanned for the presence of diverging subtrees (for example a polynomial). To avoid indefinite growth during HyGP model extrapolation analysis, these diverging subtrees are edited. When a variable is found to be the argument of addition, subtraction, multiplication, exponential, quadratic and cubic terms, the variable node is replaced by a sinusoidal function of the same variable (for example x is replaced by $\sin(x)$).

Tuning (Hybrid Approach)

Numerical coefficients are inserted in the parameterless individuals using the procedure detailed in [48]. The aim is to maximise the number of degrees of freedom of the full models, their so-called “expressivity”, while keeping the number of numerical parameters to a minimum, which in turn is linked to the tuning computational cost. As an example, the division between two variables z_1/z_2 in the tree in Fig. 8 requires a single parameter rather than two. In the first step, a vector of random values is used as the first guess, except for the best individual which inherits the same numerical coefficients in the previous generation. Notably, the user may opt to use more than one set of random values as the first guess to increase the robustness of the search. In the second step, the set of random/inherited values are used as starting points for a Particle Swarm Optimisation algorithm (PSO), which is responsible for improving the initial guesses. In the third and final step, the set of numerical coefficients improved by PSO are further refined by a deterministic algorithm called Sequential Quadratic programming (SQP). The cost function that drives the optimization of the numerical coefficients is root mean square error (RMSE) for PSO and sum of the square errors for SQP. Specifically for the SQP cost function, to reduce the risk of overfitting for sinusoidal functions, angular frequencies of sine and cosine nodes, which are not themselves arguments of other functions except for addition, subtraction, and multiplication by -1 , are detected. These angular frequencies are then tuned imposing an increasing penalisation in the optimisation search if they tend to get higher than $\omega_{\max}i = 0.5\pi/\Delta z_i$, where Δz_i is the increase in the independent variable z_i (for example, time). It is in fact assumed that frequency of sine and cosine terms can reach the maximum of a quarter of the sampling frequency used to generate the training data. So in case sine and cosine nodes are found, the minimization problem

addressed by SQP is recast as follows: Find x that minimizes $F_{\text{SQP}}(x)$:

$$F_{\text{SQP}}(x) = \frac{1}{2} \sum_{j=1}^m (\hat{f}_j(x) - f_j)^2 + \sum_{i=1}^{n_{\text{var}}} \sum_{k=1}^{n_{\text{afr},i}} g_{i,k}(a_{i,k}, \omega_{\text{max},i}) \quad (\text{A1})$$

where x is the set of unknown parameter values of the HyGP individual \hat{f} being tuned, f_j is the observed output for training point j , \hat{f}_j is the output produced by the GP individual in training point j , m is the number of training points (size of the training data set), n_{var} is the number of independent variables and $n_{\text{afr},i}$ is the number of parameters recognized as angular frequencies found for variable i in the individual undergoing tuning. The penalisation terms $g_{i,k}$ that direct the search for x away from high angular frequencies are defined as functions of the particular numerical coefficients in x recognised as angular frequencies $a_{i,k}$:

$$g_{i,k}(a_{i,k}, \omega_{\text{max},i}) = \begin{cases} 0 & \text{if } a_{i,k} \in [-\omega_{\text{max},i}, \omega_{\text{max},i}] \\ e^{(|a_{i,k}| - \omega_{\text{max},i})^2} - 1 & \text{if } a_{i,k} \notin [-\omega_{\text{max},i}, \omega_{\text{max},i}] \end{cases} \quad (\text{A2})$$

No similar constraints on numerical coefficients for sinusoidal terms are implemented in PSO, whose search for optimal numerical coefficients can span the interval $[1, 11]$. No similar constraints on numerical coefficients for sinusoidal terms are implemented in PSO, whose search for optimal numerical coefficients can span the interval $[1, 11]$.

Selection and Fitness function

The fitness function is different from that of the classic GP which works merely based on error-based metrics defined as a function of provided data and GP model response. In order to (i) encourage the evolution of smooth mathematical expressions, (ii) avoid uncontrolled increase of models' size not linked to error reduction ('bloating' [64]), and (iii) curb computational cost linked to tuning of the fitness values the fitness value $F(i, t)$ of GP model i at generation t is defined as a weighted sum of different terms or objectives, following an approach used for multi-objective optimization in evolution-based algorithms:

$$F(i, t) = a_1 F_1(i, t) + a_2 F_2(i, t) + a_3 F_3(i, t) + a_4 F_4(i, t) + a_5 F_5(i, t) \quad (\text{A3})$$

$$a_1 + a_2 + a_3 + a_4 + a_5 = 1 \quad (\text{A4})$$

where $F_1(i, t) = \text{RMSE}(i, t)$ is the root mean square error of individual i at generation t evaluated on the given training data set, $F_2(i, t) = N_{\text{tuning parameters}}$ is the number of numerical coefficients (parameters) in the individual, $F_3(i, t) = 10^6 N_{\text{corrections}}$ with $N_{\text{corrections}}$ is the number of operations (nodes) of the individual not defined on the training data set (i.e., division by zero), $F_4(i, t) = N_{\text{nodes}}$ is the number of nodes that the individual is made of (model size) and $F_5(i, t) = \exp(\alpha_1 |f_{\text{max}} - \hat{f}(x_{f_{\text{max}}})| / (|f_{\text{max}}| + 1)) + \exp(\alpha_2 |f_{\text{min}} - \hat{f}(x_{f_{\text{min}}})| / (|f_{\text{min}}| + 1)) + \exp(\alpha_3 |mean(f) - mean(\hat{f})| / (|mean(f)| + 1)) + \exp(\alpha_4 |var(f) - var(\hat{f})| / |var(f)|)$.

Here, f_{max} and f_{min} are the maximum and minimum values of the target function in the training data set observed in points $x_{f_{\text{max}}}$ and $x_{f_{\text{min}}}$, and $\hat{f}(x_{f_{\text{max}}})$ and $\hat{f}(x_{f_{\text{min}}})$ are the corresponding values of the i -th GP individual at these points. The terms var and mean stand for variance and mean of the given training data (f) and of the GP individual output on the training data (\hat{f}). The coefficients of the exponential arguments used are $\alpha_1 = 1$, $\alpha_2 = 1$, $\alpha_3 = 10$, $\alpha_4 = 10$.

The last term in the fitness function in Eq. A3 is an additional objective introduced to better capture the training data extremes, mean and variance in addition to the local accuracy given by the RMSE-based fitness function. This additional objective is introduced with the aim of steering GP evolution towards models that better match the statistical properties of the original input signal, as illustrated in the results section. Table 2 list the values of the main HyGP hyperparameters used in all experiments performed for the SPOD signals.

Appendix B. Run times and expressions of the GP model to reconstruct the time modal coefficients of the frequency-domain SPOD signals

All GP calculations were run on a laptop computer with linux OS, processor Intel(R) Core(TM) i7-4710HQ, 2.5 GHz in sequential mode, and the indicated time is the wall clock time required for GP to generate the numerical expression in each case. Examples of the GP model output to approximate the turbulent SPOD signals at frequencies corresponding

Table 2 HyGP parameters used for reconstruction of time modal coefficients of frequency-domain SPOD (Towne et al.'s approach)

Hyperparameter	Value
Population size M	50
Number of generations	50
Number of independent evolutions (runs)	5
Editing	Enabled
Binary primitives	+, -, *
Unary primitives	sin, cos, shift (translation by a constant)
PSO particles	30
PSO iterations	10
Reproduction rate	20%
Crossover rate	40%
Mutation rate	40%
Coefficient a_2 (tuning parameters)	0.001
Coefficient a_3 (corrections)	0.01
Coefficient a_4 (size)	0.01
Size of training data set (and DoE type)	3000
Size of testing data set (and DoE type)	5348 (first 3000 identical to training data set)
Sampling frequency (Hz) (normalised space)	300

to $St=0.1, 0.2, 0.5,$ and 1.0 as functions of the normalised time variable, $Z1$ are provided below.

$$\begin{aligned}
 St &= 0.1, \quad \text{run time} = 1284 \text{ s} \\
 Y(Z1) &= 6.2132813177e + 00 \\
 &+ 1.5816113444e + 00 \times \cos \left(1.3521723310e + 00 \times Z1 - 5.8317634022e + 00 \times Z1 \right. \\
 &+ 9.8650043832e + 00 \times \sin(5.8087979615e + 00 \times Z1) \\
 &+ 5.5662596779e + 00 \times Z1 - 3.2698471150e + 00 \times Z1 \\
 &+ 8.4864471317e + 00 \times \cos(1.3284981389e + 00 \times Z1) \left. \right) \\
 &+ 1.1410925588e + 00 \times \sin \left(6.9281489991e + 00 \times \cos \left(1.0535105789e + 00 \times Z1 \right. \right. \\
 &- 2.9200839133e + 00 \times Z1 + 3.9347628306e + 00 \times Z1 + 7.6717547294e + 00 \times Z1 \left. \right) \\
 &\times 8.6970827737e + 00 \times Z1 \left. \right) \\
 &+ 1.0000000000e + 00 \times \cos \left(9.8248552517e + 00 \times Z1 - 3.2263526136e + 00 \times Z1 \right. \\
 &+ 5.1265043393e + 00 \times Z1 + 3.5640234755e + 00 \times Z1 \\
 &+ \left(\left(9.2992381900e + 00 + 6.2156951133e + 00 \times Z1 \right)^1 \right. \\
 &\left. \left. - 6.4269411591e + 00 \times \sin(3.8683018621e + 00 \times Z1) \right) \right) \left. \right) \quad (6)
 \end{aligned}$$

$$\begin{aligned}
& \text{St} = 0.2, \quad \text{run time} = 1608 \text{ s} \\
& Y(Z1) = 6.3489902274e + 00 \\
& \quad + 9.7831494828e - 01 \times \sin \left(8.6496650843e + 00 \times Z1 \right) \\
& \quad \times 2.4117795259e + 00 \times \sin \left(9.9552547286e + 00 \times \sin \left(5.3926160646e + 00 \times Z1 \right) \right) \\
& \quad \times 1.0001328845e + 00 \times \sin \left(1.0990716252e + 01 \times Z1 \right) \\
& \quad + 9.2249751253e + 00 \times \sin \left(1.0071058518e + 00 \times Z1 \right) \\
& \quad + 9.7186564817e - 01 \times \sin \left(1.0156430829e + 01 \times \sin \left(1.6470705803e + 00 \times Z1 \right) \right) \\
& \quad - \left(4.8645036779e + 00 + 3.2365409302e + 00 \times Z1 \right)^1 \\
& \quad + \left(6.6436008721e + 00 + 4.2657790931e + 00 \times \sin \left(9.2479876782e + 00 \times Z1 \right) \right)^1 \\
& \quad + \left(4.6894846547e + 00 \times Z1 + 2.5877947705e + 00 \times Z1 \right. \\
& \quad \left. + 2.1958686276e + 00 \times Z1 \right) \times 1.2220579641e + 00 \times \sin \left(3.6646869468e + 00 \times Z1 \right) \\
& \quad - 1.1406041372e + 00 \times \sin \left(3.3343808250e + 00 \times Z1 \right)
\end{aligned} \tag{7}$$

$$\begin{aligned}
& \text{St} = 0.5, \quad \text{run time} = 1632 \text{ s} \\
& Y(Z1) = 5.4773568325e + 00 \\
& \quad + 1.1886089006e + 00 \times \cos \left(6.5730036615e + 00 \times \cos \left(3.7551455443e + 00 \right. \right. \\
& \quad \left. \left. \times \sin \left(3.6590388299e + 00 \times (Z1 \times Z1) \right) \right) \right) \\
& \quad + \left(2.4829021118e + 00 \times Z1 - 8.6056615309e + 00 \times Z1 \right) \\
& \quad - \left(2.5819287949e + 00 + 1.0999264473e + 01 \times Z1 \right)^1 \\
& \quad \times \left(7.7305648311e + 00 + \left(1.0000000000e + 00 + 5.6919515213e + 00 \times Z1 \right)^1 \right)^1 \\
& \quad - \left(9.3247888150e + 00 + 5.7084314549e + 00 \times Z1 \right)^1 \\
& \quad - 1.0000000000e + 00 \times \sin \left(1.0165013835e + 01 \times \sin \left(1.0000000000e + 00 \times Z1 \right) \right) \\
& \quad \times 2.6890803179e + 00 \times Z1 \\
& \quad - 1.0000000000e + 00 \times \cos \left(7.8306693236e + 00 \times Z1 + 8.9796782181e + 00 \right. \\
& \quad \left. \times \sin \left(7.2855177977e + 00 \times Z1 \right) \right)
\end{aligned} \tag{8}$$

$$\begin{aligned}
& St = 1.0, \quad \text{run time} = 1406 \text{ s} \\
& Y(Z1) = 6.1315496780e + 00 \\
& \quad + 1.0000000000e + 00 \times \sin \left(7.9577247137e + 00 \times Z1 \right) \\
& \quad + 1.0000000000e + 00 \times \sin \left(\left(1.0999975788e + 01 \right. \right. \\
& \quad \left. \left. + \left(9.2976921411e + 00 \times Z1 - 1.1001000000e + 01 \times Z1 \right)^1 \right) \right) \\
& \quad \times \left(2.3102626909e + 00 \times \sin \left(5.9864785922e + 00 \times Z1 \right) \right) \\
& \quad \times \left(1.6233382948e + 00 + 1.0000000000e + 00 \times Z1 \right)^1 \left) \right) \\
& \quad + 1.0000000000e + 00 \times \cos \left(4.0094577122e + 00 \right. \\
& \quad \left. \times \sin \left(1.0581024175e + 01 \times Z1 \right) \right)
\end{aligned} \tag{9}$$

Appendix C. Details of the ESN implementation

This section aims to give a comprehensive overview of the ESN model implemented in this study.

Teacher Signal

Prior to ESN training, it is essential to normalise the amplitudes of turbulence signals, which serves as the ground truth. This normalisation process adjusts the values to fall within the range of -1 and 1, aligning with the output range of the hyperbolic tangent activation function utilised in the model. Moreover, normalising the data facilitates faster convergence and reduces the likelihood of training instabilities.

Driver Signal

The driver signal that is to be input into to the model to initiate the training of ESN should similarly be normalised as the teacher signal. Furthermore, for the current proposed approach, it is recommended to filter the generated white noise signal (driver signal) to the same bandwidth as the teacher signal.

The ESN and Training

Before commencement of ESN training, the user can determine how the data should be divided for initialization, training, and the test set. This allows user to optimize the training process based on the characteristics of the dataset and desired outcomes. In the absence of the user input, the data is divided equally for both training and testing. The current implemented model has been tested for long term extrapolation therefore the user is afforded the option to specify extrapolation up to 40000 time steps. Correspondingly the generated driver signal shall also be split.

An ESN is characterised by a RNN comprising of an input layer, reservoir, and the output layer. ESNs are based on the fundamental concept of utilising a fixed, and randomly generated recurrent layer called the "reservoir." This reservoir should be substantially large to effectively capture the intricate dynamics of the input data. Notably, only the network's output layer undergoes training, resulting in a considerably streamlined and expedited training process when compared to conventional RNNs.

The reservoir state at the time t with N reservoir units, N_{in} inputs and N_{out} outputs is updated based on the state update equation given by Eq. 10.

$$\tilde{x}(t) = \tanh(W^{in}u(t) + Wx(t-1) + W^{fb}y(t)) \tag{10}$$

where $\tilde{x}(t)$ is the reservoir state of dimension N , $u(t)$ is the input signal. The network takes input and combines it with the hidden states from the previous time step, $t-1$, using two weight matrices: one for the input, denoted as $W^{in} \in R^{N \times N^{in}}$, and another for the reservoir, denoted as $W \in R^{N \times N}$. The values of input weights are sampled such that they fall in the range of $[-a, a]$, where a denotes input scaling of the network, a crucial hyperparameter. The reservoir

weight matrix W is sparsely initialized. Generally, the model performance is not significantly impacted by the sparsity of the reservoir, making it a secondary parameter for optimization. Nevertheless, utilizing sparse matrix representations allows for expedited updates of the reservoir states. The resulting linear combination acts as the input for the chosen non-linear activation function, which, in our case, is the hyperbolic tangent function (\tanh).

$W^{fb} \in \mathbb{R}^{N \times N^{out}}$ denotes the output feedback matrix. However, in cases where output feedback is not necessary, W^{fb} is set to zero. Followed by the state updating, extended system state is determined by the concatenation of the current state and previous states as shown in Eq. 11.

$$x(t) = \gamma x(t-1) + (1-\gamma)\tilde{x}(t) \quad (11)$$

Where γ is an important hyperparameter called leakage rate. This parameter regulates the influence of previous states on the current state. Finally, the output signal is then determined by the output layer of the network based on the following equation Eq. 12.

$$y(t) = W^{out}x(t) \quad (12)$$

The following table gives an overview of the hyperparameters used for the reconstruction of time modal coefficients of frequency-domain SPOD using ESN.

Table 3 ESN hyperparameters for the reconstruction of time modal coefficients of frequency-domain SPOD

Hyperparameter	Value
Number of reservoir neurons	800
Spectral Radius	0.99
Leakage Rate	0.5
Sparsity	10%
Input Scaling (all cases except St 0.2 [imag] and St 0.5)	0.5
Input Scaling (St 0.2 [imag] and St 0.5)	0.7