33RD CONGRESS
OF THE INTERNATIONAL COUNCIL
OF THE AERONAUTICAL SCIENCES
STOCKHOLM, SWEDEN, 4-9 SEPTEMBER, 2022

ICAS 2022 SWEDEN

APPLICATION OF GENETIC PROGRAMMING AND ARTIFICIAL NEURAL NETWORK APPROACHES FOR RECONSTRUCTION OF TURBULENT JET FLOW FIELDS

Vasily Gryazev[1], Umberto Armani[1], Amal Roy Murali[2], Annabel P. Markesteijn[1], Sergey A. Karabasov[1], Vassili Toropov[1], S. Elnaz Naghibi[3], Vladimir Riabov[4]

[1]School of Engineering and Material Science, Queen Mary University of London, London
[2]Laboratoire de Mecanique des Fluides et d'Acoustique, École Centrale de Lyon, France
[3]Department of Aeronautics, Imperial College London, London,
[4]Department of Complex Systems, Future University Hakodate, Hakodate, *Hokkaido,* Japan

Abstract

Two Machine Learning (ML) methods are considered for reconstruction of turbulent signals corresponding to the Large Eddy Simulation database obtained by application of the high-resolution CABARET method accelerated on GPU cards for flow solutions of NASA Small Hot Jet Acoustic Rig (SHJAR) jets. The first method is the Feedforward Neural Networks technique, which was successfully implemented for a turbulent flow over a plunging aerofoil in (Lui and Wolf, 2019). The second method is based on the application of Genetic Programming, which is well-known in optimisation research, but has not been applied for turbulent flow reconstruction before. The reconstruction of local flow velocity and pressure signals as well as time-dependent principal coefficients of the Spectral Proper Orthogonal Decomposition of turbulent pressure fluctuations are considered. Stability and dependency of the ML algorithms on the smoothness property and the sampling rate of the underlying turbulent flow signals are discussed.

Keywords: turbulence, genetic programming, artificial neural networks

1. Introduction

Large-scale turbulent jet flows occur in many aerodynamic and aeroacoustic problems. Because of the nature of high-Reynolds number turbulence, these flows possess a rich dynamic behavior, which continues attracting research interest of the fluid dynamics community. Recent progress in novel computer architectures and high-resolution algorithms made it possible to extend Large Eddy Simulations (LES) to billions of degrees of freedom. In addition to having high spatial resolution, such simulations need to capture small temporal scales and evolve over long times to generate meaningful statistical data series. Hence, one of the topical problems in the field is the analysis of the large data sets, which emerge from these calculations, thereby allowing to develop fast reduced-order data-driven models in the context of design optimization and flow control [1]. In many data analysis methods, the starting point is flow modal decomposition such as proper orthogonal decomposition (POD) [2-4] and dynamic mode decomposition (DMD) [5-7], After the flow decomposition step, the dominant modes serve to reproduce the most important dynamics of the physical system in question. Several methods to approximate this dynamics exist including Galerkin projection and its variants [8] and sparse regression of nonlinear dynamics [9]. Notably, [10] reported that reduced-order models developed using some of these techniques exhibit unstable behavior when trying to reproduce the complex dynamics of the governing Navier-Stokes solution. Despite the progress in the development of more stable reduced-order models [11-12], application of these techniques to reconstruct dynamics of complex turbulent flows such as high-speed turbulent jets remains a challenge. Machine learning methods are a promising technique for the development of reduced-order models of high-Reynolds number flows. Neural networks are one popular technique for reconstructing the dynamics of high-dimensional systems [13-14]. Furthermore, a special sub-class of neural network methods called feedforward neural networks (FNN) was found particularly effective for modelling of a range of dynamical systems [15]. More

recently, [16] used FNN to develop a numerical methodology which combines flow modal decomposition and regression analysis. Furthermore, to improve the regression step they used a spectral proper orthogonal decomposition (SPOD) [3]. For the problem of a plunging aerofoil with large areas of turbulent separated flows, the method was demonstrated to predict the flowfield beyond the training window and with larger time increments than those used by the full order model, thereby demonstrating the robustness of the reduced-order model based on FNN. In this work, performance of the reduced order technique based on FNN developed in [16] will be tested for reconstruction of velocity and pressure fluctuations extracted from the Large Eddy Simulations database of high-speed jets [17]. The jets correspond to the two isolated static jet cases from the NASA SHJAR experiment. These are the so-called Set Point 3 and 7 jets, which correspond to isothermal conditions, Reynolds number of around one million, and acoustic Mach numbers of 0.5 and 0.9, respectively. The simulations were performed using the high-resolution CABARET method [18-22], which is characterized by low dispersion and dissipation. The CABARET LES solver was extensively validated for high-speed jet flows [22-24]. The solver was implemented on the Graphics Processing Units (GPUs) to reduce the time to solution to two days on a few GPU cards with LES grids of around 100 million cells [25-27]. In addition to the FNN method, Genetic Programming (GP) is applied for the same task. GP is a method that originates from the class of genetic algorithms (GA) with the goal of addressing symbolic regression problems. Although it is well known in the optimisation literature [28-30], GP has not been widely used in application to low-order modelling of complex fluid dynamics flows so far. In particular, to the best knowledge of the authors, in the current work GP will be applied for the reconstruction of temporal dynamics of high-Reynold number jet flows for the first time.

2. Jet Flow Cases and Methodology

In this section we are presenting the results of the Large Eddy Simulations (LES) and extract the time signals and time coefficients of the SPOD modes which will be used as an input for FNN and GP methods. The jet cases considered in the present study correspond to the conditions of the NASA Small Hot Jet Acoustic Rig (SHJAR) experiment. SHJAR jet is isothermal single-stream static jet, which issues from profiled axi-symmetric convergent nozzles of a small area ratio. Two Set Points (SP) 3 and 7 with acoustic Mach number Ma = 0.5 and 0.9, respectively, are considered. The LES data whose implementation details can be found in [17] are first interpolated onto a $n_x \times n_r \times n_\theta$ structured cylindrical grid spanning $x/D_j,\ r/D_j,\ \theta \in [0,10] \times [0,3] \times [0,2\pi]$, where D_j is the jet diameter. Temporal, spatial separation and simulation parameters are summarised in Table 1.

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

Set Point (SP)	3	7
Acoustic Mach number M_a	0.5	0.9
Jet Mach Number M_j	0.51	0.98
Nozzle Temperature Ratio	0.950	0.835
Reynolds Number based on jet diameter	$0.56 \cdot 10^6$	$1 \cdot 10^6$
Sampling rate $dx/D_j = dr/D_j$	0.1	
dt (s)	$2 \cdot 10^{-5}$	
Number of time samples	5348	

For the reconstruction of the temporal dynamics two flow quantities will be considered, which correspond to fluctuating pressure p' and fluctuating velocity u',

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

where $\overline{(\cdot)}$ denotes the mean quantity and $(\cdot)'$ the fluctuating part.
A snapshot of the flow field in the streamwise cross section along jet axis and transverse planes at different streamwise locations x/D_j which correspond to the fluctuating pressure p' of the SP7 jet is shown in Figure 1.

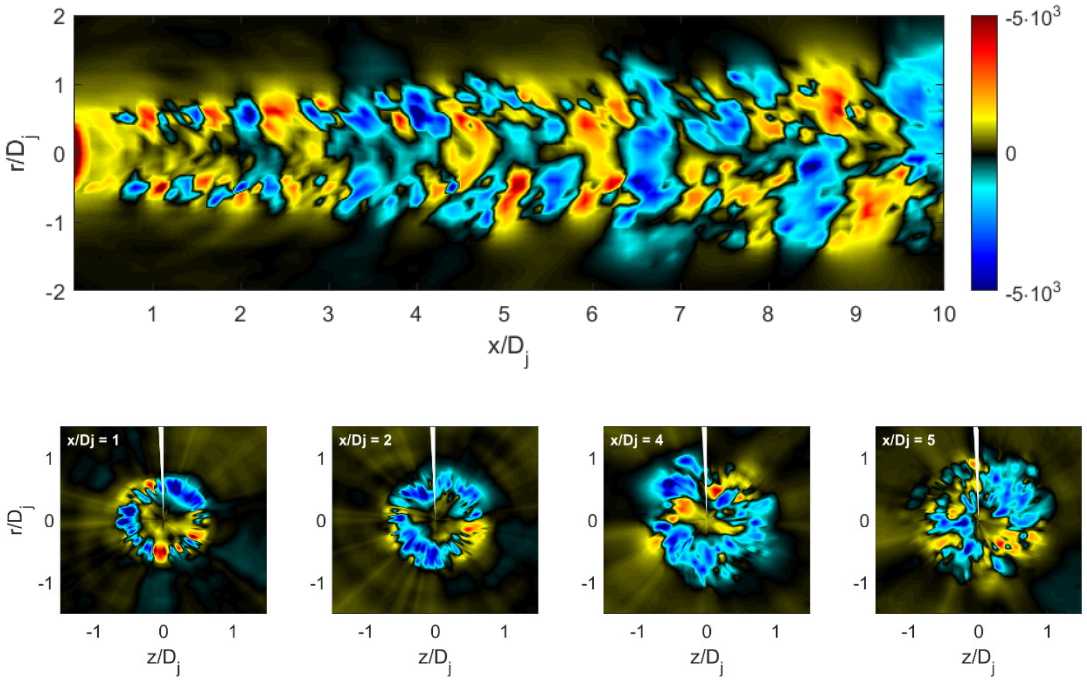


Figure 1 – Instantaneous snapshot of fluctuating pressure in streamwise cross section along the jet axis (top) and transverse planes at streamwise locations $x/D_j = 1,2,4,5$ (bottom)

The next step is to decompose the 4D flow solutions (3D in space and 1D in time) into a series of time signals which capture the dynamic behavior of dominant SPOD modes. The procedure consists of several step. First, due to the periodicity of the jet flow in azimuthal direction, it can be then decomposed into azimuthal Fourier modes,

$$p'(x,r,\theta,t) = \sum_m \widehat{p'}_m(x,r,t)\, exp\,[im\theta] \tag{2}$$

and the result of azimuthal decomposition for mode $m = 0$ is shown in Figure 2.

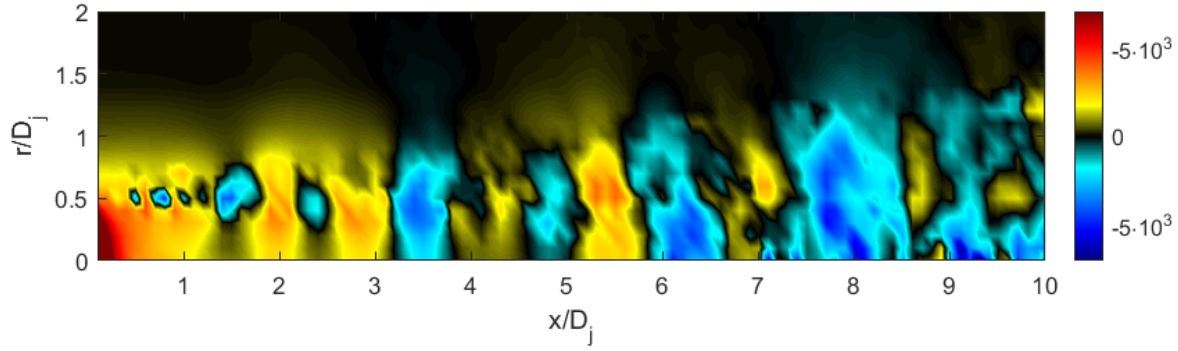


Figure 2 – Instantaneous flow field of fluctuating pressure for azimuthal mode $m = 0$.

Next, we are going to apply SPOD decomposition to split the flow into dominant modes for each frequency band of interest. The Details of SPOD implementation can be found in [31, 32]. The spatial distributions of the SPOD modes of the NASA SP7 jet at azimuthal mode number m = 0 and four typical frequencies are shown in Fig.3. Each SPOD mode corresponds to a time-dependent coefficient and can be extracted for the particular frequency of interest. In this work the frequencies will be reported in terms of the Strouhal number St = fD_j/U_j , where f is a frequency in Hz and U_j is the jet exit velocity.

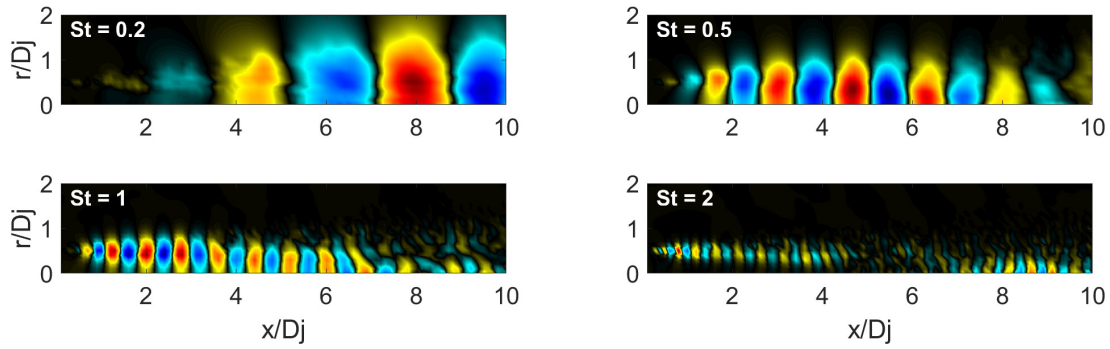


Figure 3 – First SPOD mode of the pressure fluctuations of Ma = 0.9 jet for St = 0.2, 0.5, 1, 2 at azimuthal mode m = **0.**

Having identified the extraction and decomposition procedure we present the local time signals of the fluctuating pressure (Fig.4a,b) and velocity (Fig.4c,d) at two spatial locations on the jet lipline $r/D_j = 0.5$ and one azimuthal angle and time coefficients of the first SPOD mode for two representative frequencies.. As the jet flow spreads downstream the time signal shows significantly higher correlation, especially for fluctuating velocity (Fig.4d). In comparison with pressure and velocity time signals, the SPOD time coefficients are smoother and resemble harmonic behavior at low frequencies (Fig4.e)

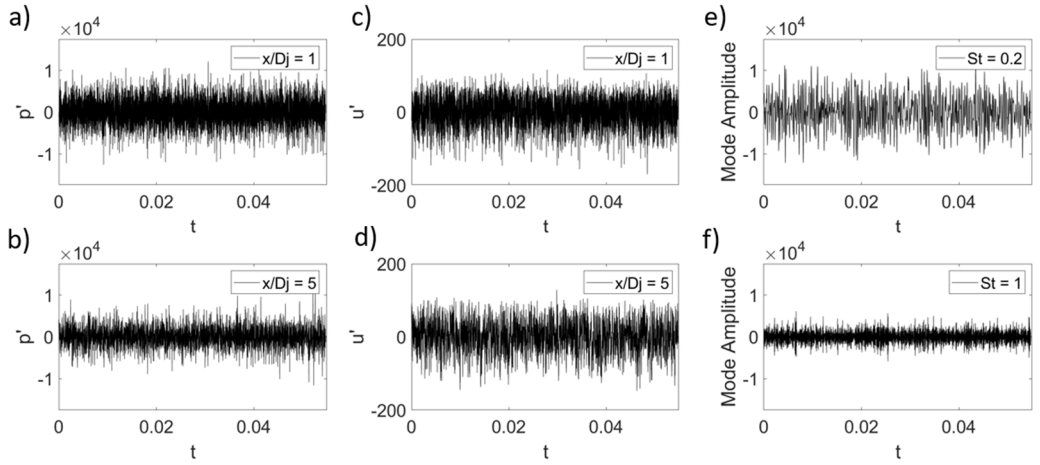


Figure 4 – Local time signals $p'(t, \theta_1)$ at different streamwise locations

2.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) is a popular class of machine learning algorithms that has the properties of a universal function approximator [33,34]. In the simplest of perspectives, ANN is a network of interconnected nodes, connected by the numerical operations whereby information is transferred from one node to the next (Fig.5). Each node numerically represents a set of weights and biases by which that node processes information from its predecessors and the output is made non-linear using a suitable activation function. Hence, from input to output layers, the network as a whole represents a highly nonlinear differentiable function parameterised by its weights and biases

[35].

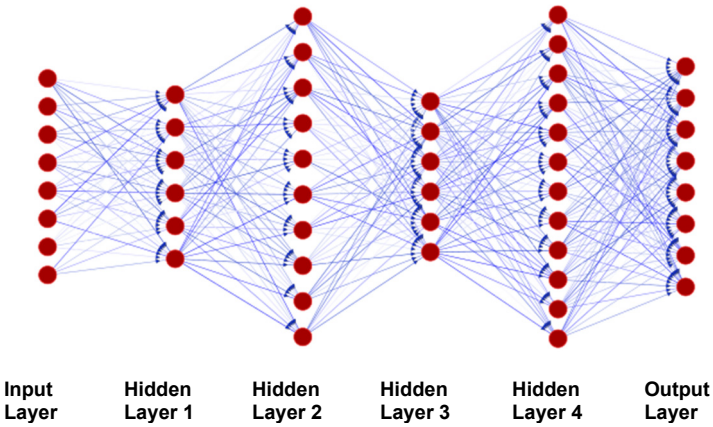


Figure 5 – A neural network of interconnected nodes (red) representing the weights and biases and the connections (blue vectors) representing the flow of information in a forward propagation. The ANN shown has 4 hidden layers of random widths and 8 neurons in the input and output layers.

The forward computations of these interconnected nodes can be recorded in the form of a computational graph and thus can be differentiated using the Automatic Differentiation technique. This enables the network to be optimizable by gradient descent. Here the optimization process whereby the parameters of the network are optimized to fit a function map is called 'training' and the process by which interpolation or extrapolation is done on a data-point which was not presented to the network during training is the 'prediction'. While training, the gradients of a suitable measure of the prediction error is computed with respect to each of the weights and biases, and are optimized iteratively until the error converges. For predicting, the computation from input layer to output layer via intermediate or hidden layers is called the forward propagation. The computational advantage of ANNs lies in the fact that once a network is iteratively trained by making predictions and optimizing for its errors iteratively, the network's prediction error converges to a minimum possible value when the network parameters and structure can be frozen for permanent reuse. Thus, after training, the computation required for prediction over a new data point is that required for a single forward propagation only and prediction cost scales linearly with the number of predictions to be made. This is significantly cheaper than the iterative training process. Given that they are highly non-linear differentiable functions optimizable by gradient descent and with expressivity scalable with the depth and width of the network, ANNs can effectively replace complex computational pipelines.

That being the numerical property of a Neural Network, they are the first choice for any interpolation tasks that involves high dimensional or non-linear maps. In this work, we chose to test the efficacy of an ANN based framework proposed by Lui and Wolf. [16] in regeneration of the time accurate turbulence signals. This choice is motivated by the fact that the framework has favorable numerical properties because it was tailored to the Proper Orthogonal Decomposition (POD) method with additional spectral smoothing [3]. However, our end goal is to work with turbulent signals, which may not necessarily come from the specific filtered POD version but, for example, from the SPOD framework which allows decoupling the signal into well-defined physical space-time modes [31]. In this case, we may be outside the confidence region for the ANN algorithm unlike in the original application. Nevertheless, we share the insights from our parametric study of the method and critical evaluation of its performance.

As explained in detail by Lui and Wolf. [16] an ANN based Reduced Order Model (ROM) employs ANN as a function approximator for the temporal oscillations of dominant POD modes of the flow. But rather than making predictions of the signal via extrapolating in the time domain using an ANN, the prediction made by ANN is for the time gradient and input is the signal amplitudes. The process then is then equivalent to solving the ODE

$$\frac{\partial P(t)}{\partial t} = NN\big(p(t)\big) \tag{3}$$

where, $p(t)$ is the time-varying pressure signal and $NN(\cdot)$ is the neural network prediction. The signal is then reconstructed by time marching using a fourth order explicit Runge-Kutta scheme. The advantage of such a framework is two-fold. First, if the input signal is sufficiently smooth, the time gradient will be bounded. Hence both input and output domains for the network are bounded and the network always operates over the same range of values irrespective of the time coordinate. This means that, if the time stepping process is stable, the network always is interpolating from the dataset even while predicting the gradient outside the training window. The second advantage is due to the properties of SPOD/POD modes in strongly periodic flows. They can be identified to have two main characteristics. The first is that the SPOD modes in the simple case of a cylinder flow occurs in pairs [3], thereby having correlation in the input to the ANN. This effectively means that when multiple signals are inputted to the ANN simultaneously, the gradients calculated during backpropagation have comparatively lower competition between optimization for individual signals than a case with completely uncorrelated signals. The second property is that SPOD modes generated by the filtering technique have nearly tonal Fourier structure and resembles an amplitude modulated signal. This is a crucial factor for ANNs since they are known to learn frequency contents progressively with training iterations and the number of iterations required for the ANN to learn higher frequencies grows quadratically [35]. Hence with nearly tonal input signals, the probability that the training error converges in practical number of iterations are higher than in the case of a broadband signal. Given these characteristics of input signals, the ANNs chosen by a random hyper-parameter search were shown to be able to extrapolate the temporal coefficients of SPOD and POD modes by Lui and Wolf. [16].

For testing the efficacy of Artificial Neural Networks in recreating the turbulent signals, we reuse the code provided by the original authors. Here we strip the framework of the Proper Orthogonal Decomposition module which uses a smoothing of the correlation matrix and input time-series signals directly to the high order differentiation module where the time gradients are computed using a 10th order compact scheme. The result is then inputted to the ANN based regression module where a Random search algorithm explores the possible network configurations to find a suitable set of hyperparameters for the ANN. Here, for each of the random configurations generated, the Mean Absolute Training Error at the last training iteration is used as the performance metric. Once the network structure is finalized by the search algorithm by minimizing Mean Absolute Training Error, the chosen network is employed by a fourth order explicit Runge-Kutta method to extrapolate the signal from the initial time step. Hence, in practice, the random search algorithm attempts to find an optimal network structure which best overfit the training dataset. Here the degrees of freedom for the Random Search algorithm are chosen so as to limit the total computational time when compared to the Genetic Programming approach.

Now, since multiple signals are substituted to an ANN simultaneously, 6 to 10 for stable performance in our attempts, the trajectory of the signal in this multi-dimensional space is the effective input. The coordinates of this trajectory being represented by the signal values, the corresponding target at each point of the trajectory are the time gradients. Since these gradients are already known from the future time-step at any given signal location, they can be numerically calculated for the purpose of training the network. Once the network accurately captures the gradients along the trajectory with sufficient resolution, a time marching algorithm can be initialized at the starting point of the original trajectory to reconstruct the signal. If in the high dimensional space, the trajectory in the dataset had already formed a closed loop, this means that the time marching algorithm will be able to capture the limit-cycle of the signal leading to a stable extrapolation with significant statistical similarity with the original signal. However, if the trajectory in the dataset is open ended, or more precariously, ending in the vicinity of a previous point of the trajectory with incongruent gradient direction, this can lead to unstable extrapolation even without any numerical issues.

In the view of the optimization process, a randomly initialized neural network will have random predictions of the above said gradient field. They are expected to be smooth and continuous from the properties of ANNs. If trained using a stochastic gradient descent approach, the optimizer traverses along the signal trajectory and optimizes the neural network at each location to reduce the squared error between the true and predicted gradients. Again, since the neural network is a smooth differentiable function, this optimization results in the optimization of the network's gradient

prediction in the vicinity of the trajectory. Assuming an approximately uniform radius of influence of optimization on the distribution of predicted gradients close to the trajectory locations – an assumption logically valid only after a successful pretraining where the sensitivity of the network parameters to new input data is minimal – the stability of this process requires sufficient sparsity of the trajectory in its domain. If instead the signals are confined to a closed space, the optimization process revisits the same vicinity of the signal trajectory at multiple instances to reinforce multiple target gradients. This difference of target gradients for the same datapoint will destabilize the training process with a non-convergent training error. In case of a batch gradient descent process, the effective optimization may suffer the same way as in the case of multi-objective optimization with competing objectives. This issue will be aggravated in the case of high frequency signals, since the probability of the trajectory colliding with itself increases in a lower dimensional space. On the other hand, a largely sparse domain will also leave the network untrained with random gradient distribution in the regions where the optimizer did not operate. Both these effects can lead the time stepping algorithm to predict trajectories to either stray far outside of the intended domain, or to a region of negligible gradient leading to a constant value output, or to a wrong limit cycle. These modes of failures were a characteristic of instability with the ANN based extrapolation process for high frequency turbulence signals.

To understand the stability of ANN based method and compare its performance with Genetic Programming, the former has been first tested on local signals corresponding to velocity and pressure fluctuations of the SP7 jet case. We find that preprocessing of the turbulent signal by smoothing is of paramount importance for keeping the ANN algorithm stable. The best practices are then used to reconstruct the time-dependent coefficients of the SPOD decomposition of the azimuthal pressure mode of the same jet as discussed in the results section.

2.2 Genetic Programming

Genetic programming (GP) [28,29,30,40] 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 computer programs, usually in the shape of graphs or trees, which are iteratively evolved through the application of so-called genetic operators and evaluated to develop a solution or metamodel. In approximation problems, a program represents an empirical model used to approximate a response function. For instance, Figure 6a shows a program representing the expression $(z_1/z_2 + z_3)^2$.

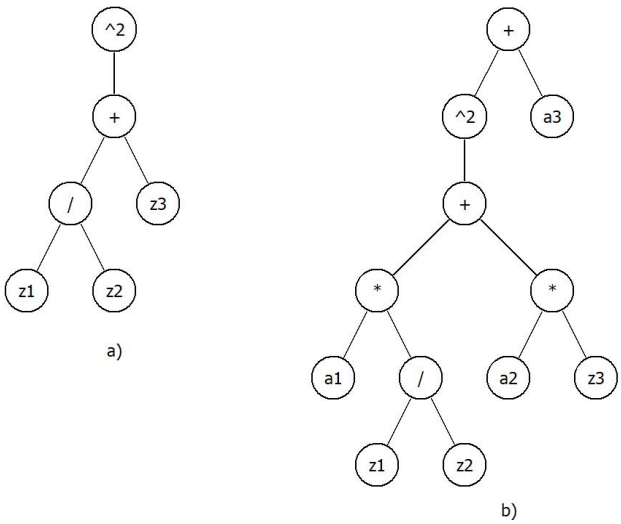


Figure 6 – HyGP tree structures for the expression $(z_1/z_2 + z_3)^2$.
In a) the tree is in a “parameterless” state, as no numerical parameters are present,
In b) numerical parameters a_i are inserted for full expressivity.

These programs can be created in a generic and hierarchical manner while varying in size and shape. The ultimate goal of GP is to solve a problem by searching highly fit computer programs in the space of all possible programs that can be a solution to the problem. This aspect is key to

finding near global solutions by fostering many solutions that may potentially be close to minima (local or global). In GP, the creation of the initial population is performed randomly in an attempt to extensively explore the space of models defined by the building blocks provided by the user. Subsequently, the evolution of a highly fit model proceeds autonomously through the iterative action of genetic operators and model evaluation by a fitness function, which determines the quality of the evolved model. The fitness function has a dominant role in steering the evolution towards plausible solutions.

The genetic programming code used in this paper is the HyGP implementation developed by Armani [30,40]. Despite being founded on the basic principles and guidelines provided by Koza [28], HyGP implementation can be classified in the subcategory of “hybrid GP” or “memetic GP” [36,37] as a result of its additional features such as the exploitation of Baldwinian learning. Hence, the common GP building blocks can be found despite being slightly modified to embrace the hybrid approach. The genetic operators used are the classic operators: reproduction, mutation and crossover. However, they are performed on mathematical expressions stripped of their numerical parameters (referred to as “structures” or “parameterless individuals” in the following), which are optimized in two steps, using first a population-based optimization algorithm (Particle Swarm Optimisation [38]) and subsequently a deterministic method (Sequential Quadratic Programming or SQP [39]) to finely refine the values of the parameters. A schematic description of the logical steps featuring in the HyGP algorithm is sketched in Fig. 7.

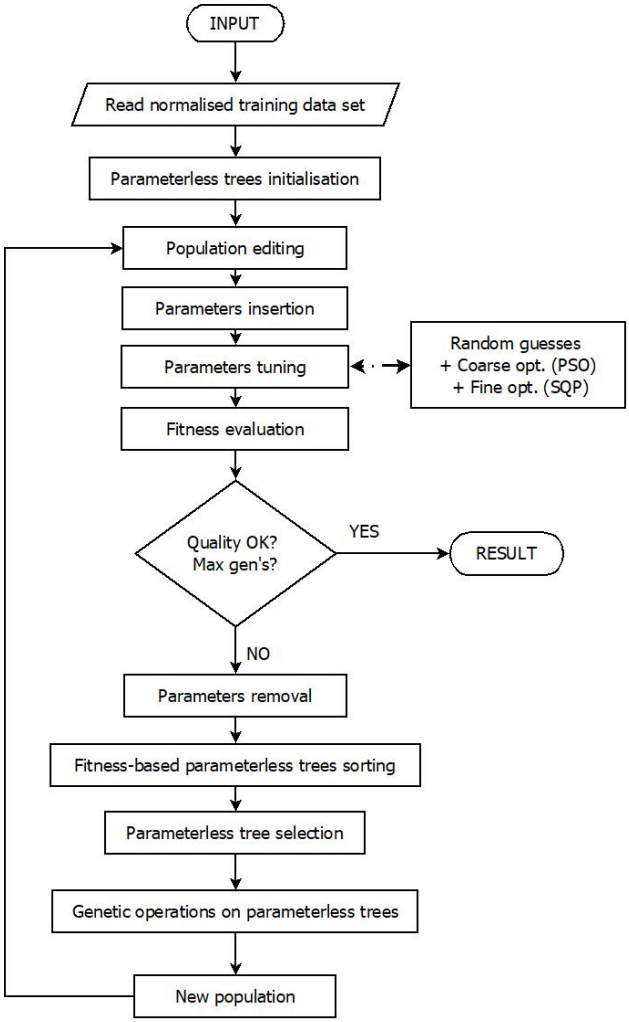


Figure 7 – Flowchart of HyGP algorithm.

More details about genetic programming can be found in [40, 41]. Below are given the main highlights of the HyGP code used per process involved:

- **Input normalization**
  Training data set is provided by the user linearly scaled so that independent (points in the

design space) and dependent variable values (target values) once normalized span the range [1.0,11.0]

- **Genetic operators**
  - 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 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 (size 3) 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 (uniform probability distribution) then randomly selecting one node among the ones sharing the same depth (uniform probability distribution). 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 (size 3) is used to choose the individual undergoing mutation. Nodes are selected randomly using the same approach adopted for crossover.

- **Editing**
  The editing operation aims at turning all GP models 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 unbounded growth during HyGP model extrapolation analysis, these diverging subtrees are edited (modified): when a variable is found to be the argument of addition, subtraction, multiplication, exponential, square and cube, the variable node is replaced by a sinusoidal function of the very 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 [39, p. 131]. The aim is to maximize the expressivity of the individuals keeping to a minimum the number of numerical parameters, which in turn is linked to the tuning computational cost (using the tree in Fig. 6 as an example, the division between two variables z_1 / z_2 requires a single parameter not two). Initially, a vector of random values is used as a first guess (the user may opt to use more than one set of random values to increase the robustness of the search), except for the best individual which, replicated to the new generation, inherits the same numerical coefficients it had in the previous generation. As a second step, the set of random/inherited values are used as starting points for a Particle Swarm Optimisation algorithm (PSO), whose function is to improve the initial guesses. As a third and final step, the set of numerical coefficients improved by PSO is further refined by a deterministic algorithm, 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 sin() and cos() nodes which are not themselves arguments of other functions except addition, subtraction and multiplication by -1 are detected and they are tuned imposing increasing penalization for angular frequencies higher than $\omega_{max\ i} = 0.5\ \pi/\Delta z_i$, which comes from assuming that frequency of sin() and cos() terms may reach as a maximum a quarter of the sampling frequency used to generate the building data set. So in case sin() and cos() nodes are found, the minimization problem addressed by SQP is recast as follows:

$$\textit{find } \mathbf{x}$$
$$\textit{minimize } F_{SQP}(x) = \frac{1}{2}\sum_{j=1}^{m}\left(\hat{f}_j(x) - f_j\right)^2 + \sum_{i=1}^{n_{var}}\sum_{k=1}^{n_{afr\,i}} g_{i,k}\left(a_{i,k}, \omega_{max,i}\right) \qquad (4)$$

where **x** is the set of unknown parameters' 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 $n_{afr\,i}$ 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 $a_{i,k}$:

$$g_{i,k}(a_{i,k}, \omega_{max\,i}) = \left\{ 0 \; if \; a_{i,k} \in [-\omega_{max,i}, \omega_{max,i}] ; \; e^{(|a_{i,k}|-\omega_{max,i})^2} - 1 \; if \; a_{i,k} \notin [-\omega_{max,i}, \omega_{max,i}] \right\} \tag{5}$$

No overfitting countermeasures for sinusoidal terms are implemented in PSO, whose search for optimal numerical coefficients can span the interval [1.0, 11.0].

- **Selection and Fitness function**
  It is different from classic GP fitness function which works merely based on an error-based metrics defined as a function of provided data and GP model response. In order to encourage the evolution of smooth mathematical expressions, to avoid 'bloating' [41] and to curb computational cost linked to tuning the fitness values $F(i,t)$ of HyGP model or "individual" 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:

$$F(i,t) = a_1 F_1(i,t) + a_2 F_2(i,t) + a_3 F_3(i,t) + a_4 F_4(i,t) + a_8 F_8(i,t) \tag{6}$$
$$a_1 + a_2 + a_3 + a_4 + a_8 = 1 \tag{7}$$

  where:
  $F_1(i,t) = RMSE(i,t)$ is the root mean square error of individual i at generation t evaluated on the given training data set

  $F_2(i,t) = N_{tunig\;parameters}$ is the number of numerical coefficients (parameters) in the individual

  $F_3(i,t) = 10^6 N_{corrections}$ where $N_{corrections}$ is the number of operations (nodes) of the individual not defined on the training data set (i.e. division by zero)

  $F_4(i,t) = N_{nodes}$ is the number of nodes that the individual is made of (model size)

$$F_8(i,t) = exp\left(\frac{\left|f_{max} - \hat{f}(x_{f\,max})\right|}{|f_{max}| + 1}\right) + exp\left(\frac{\left|f_{min} - \hat{f}(x_{f\,min})\right|}{|f_{min}| + 1}\right) + exp\left(10\frac{\left|mean(f) - mean(\hat{f})\right|}{|mean(f)| + 1}\right) + exp\left(10\frac{\left|var(f) - var(\hat{f})\right|}{|var(f)|}\right)$$

  where f_{max} and f_{min} are the maximum and minimum values of the target function in the training data set observed in points $x_{f\,max}$ and $x_{f\,min}$, and $\hat{f}(x_{f\,max})$ and $\hat{f}(x_{f\,min})$ are the corresponding values of the i-th GP individual in such 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 last term in the fitness function in Eq.6 is an additional objective introduced to better capture training data extremes, mean and variance on top of 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 signal provided as input, as it will be detailed in the results section.

To ease model interpretation and analysis, all the GP models generated for the current study have

been reported in the Appendix together with the corresponding computing time.

3. Results

3.1 Artificial Neural Networks

A. Reconstruction of local time-series signals from turbulent pressure fluctuations

The local time series of pressure signals from various locations of the jet were used as an input to the ANN based extrapolator. The signals first tested were the raw signals which yielded no stable extrapolation. Band-passed version of the original signal was then generated by choosing a mean frequency of 2000 Hz and bandwidth of 500 Hz as shown in Fig. 8 Here the objective was to test the ANN performance at nearly tonal signal at sufficiently lower frequency mimicking the properties of SPOD signals in the original framework. Nevertheless, the ANNs are unable to obtain a stable reconstruction as shown in Fig. 9, and hence are to be considered unsuitable for the purpose.

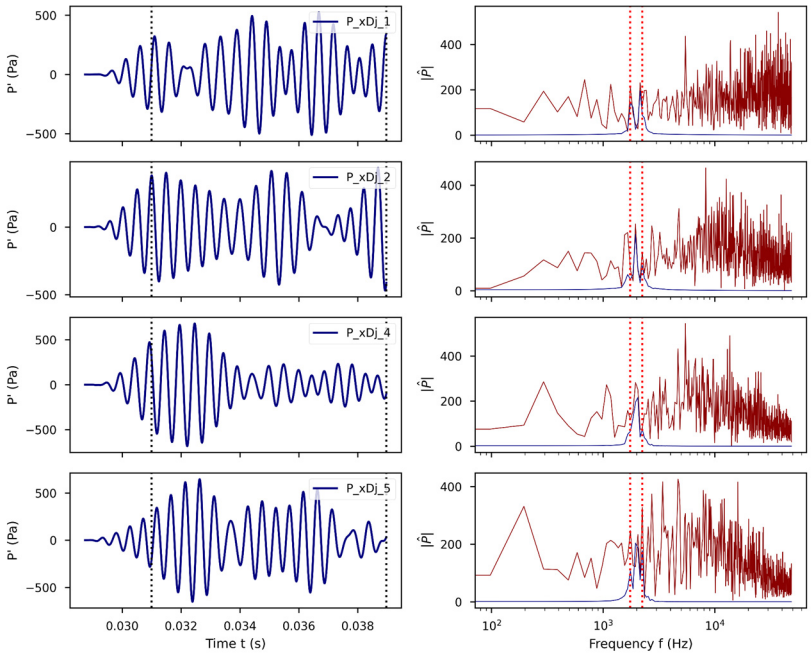


Figure 8 – The set of four signals from the axial location of the jet shear layer corresponding to locations $x/Dj = [1,2,4,5]$ after applying the bandpass filter with mean frequency at 2000 Hz and bandwidth of 500 Hz are shown above. The bandpass signal and its FT are shown in blue(—) and the FT of original signal is shown in red(—). Dotted lines indicate truncation of the signal in time domain (left) before training and bandwidth of filter used in frequency domain (right).

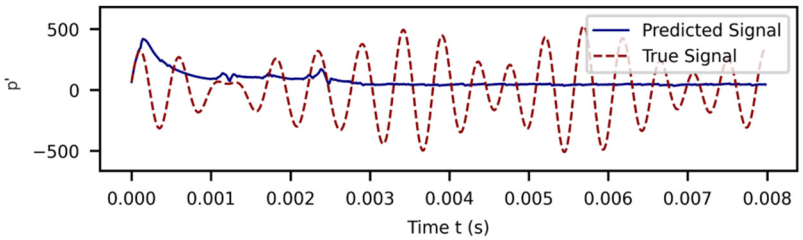


Figure 9 – The signal is reconstructed using ANN based extrapolation for the case of bandpass filter with mean frequency at 2000 Hz and bandwidth of 500 Hz showing the mode of failure where the gradient prediction attenuates to zero. The same behavior is obtained for all four signal inputs and are hence omitted.

B. Reconstruction of time-dependent principle coefficients for the space-frequency modes

The SPOD signals of the axisymmetric mode of the fluctuating pressure were inputted to the ANN based extrapolation routine to study its efficacy at various spectral compositions of the input. Here the signals are obtained by using SPOD following Towne et al. [31] of the flow solution utilizing the entire flow domain. Thus time-series signals are the temporal coefficients of spatial modes obtained at a particular Strouhal number. Our tests indicate that ANN method is sensitive to both the mean frequency and the bandwidth, with stability increasing as both parameters decrease. Since the time-series data of an unfiltered SPOD signal also contains a broader bandwidth, we perform further filtering of signals where peak frequency and bandwidth are used for a grid search with stability as objective. The spectral contents of inputted data before and after filtering are shown in Fig. 10 and the result for ANN based reconstruction in Fig. 11.

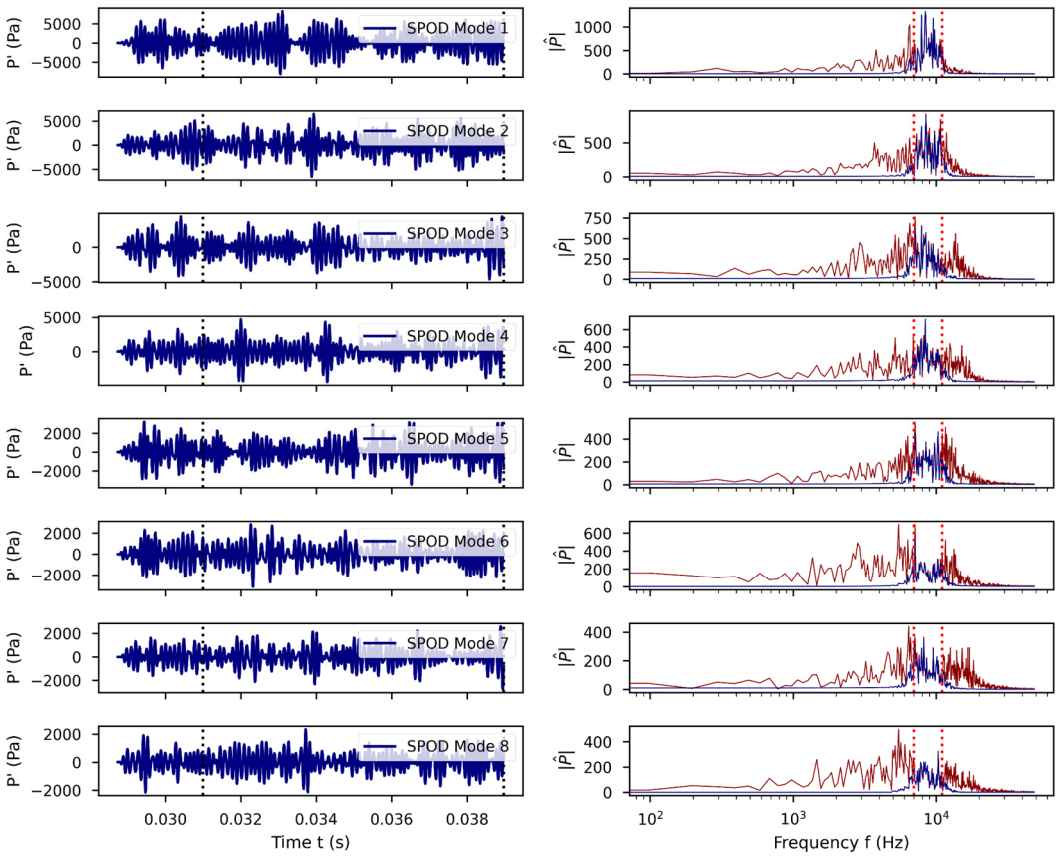


Figure 10 – Input data of SPOD temporal coefficient for St=0.5, showing the original and filtered Fourier Transforms for bandpassing at 9000 Hz and bandwidth of 4000 Hz. The bandpass signal and its FT magnitude are shown in blue(—) and the FT of original signal is shown in red(—). Dotted lines indicate truncation of the signal in time domain (left) before training and bandwidth of filter used in frequency domain (right).

From the result of extrapolation for varying filter parameters, it can be inferred that the ANN method is unable to handle the bandwidth for the SPOD signals where SPOD refers to the algorithm by Towne et al. [31] With a reduced bandwidth of 4000 Hz, the extrapolation is only stable for a few initial time steps and then the signal develops low frequency contents with large amplitude values. A further reduction of the mean frequency to 2000 Hz and bandwidth to 500 Hz results in a more stable process as shown in Fig. 12 and 13, thus demonstrating that the frequency content is the primary factor affecting stability. An extreme case of lower mean frequency of 1000 Hz and narrow bandwidth of 100 Hz are also attempted and the inputs and results are as shown in Fig. 14 and 15 respectively. Here the bandpassing results in a sufficiently smooth signal with a tonal structure and the ANN based extrapolator follows the same trajectory as the input signals. Nevertheless, at the prescribed mean frequency and bandwidth, the method does not hold promise to help in stable extrapolation of signals for relevant applications and therefore further statistical analysis of the predicted signals are omitted.

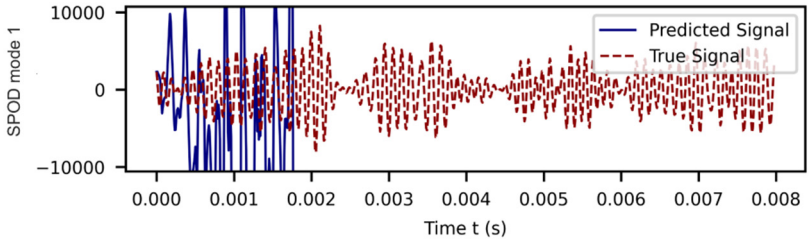


Figure 11 – The signal is reconstructed using ANN based extrapolation for SPOD mode 1 showing the mode of failure where the signal strays to the exterior of the signal amplitude boundaries. The same behavior is obtained for all eight signals.

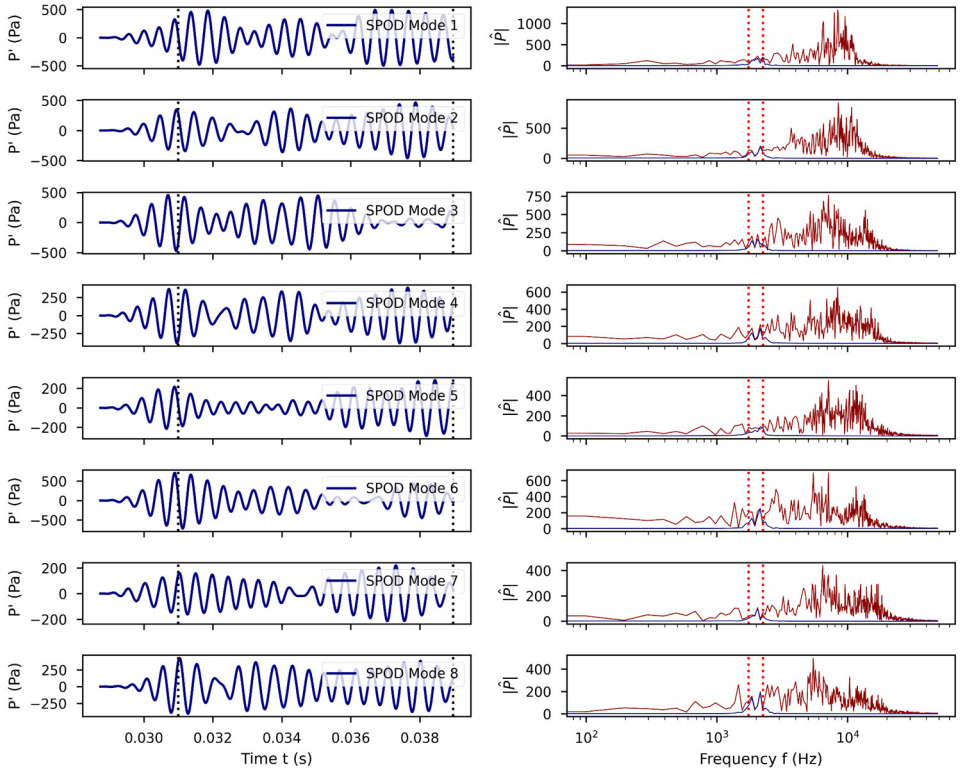
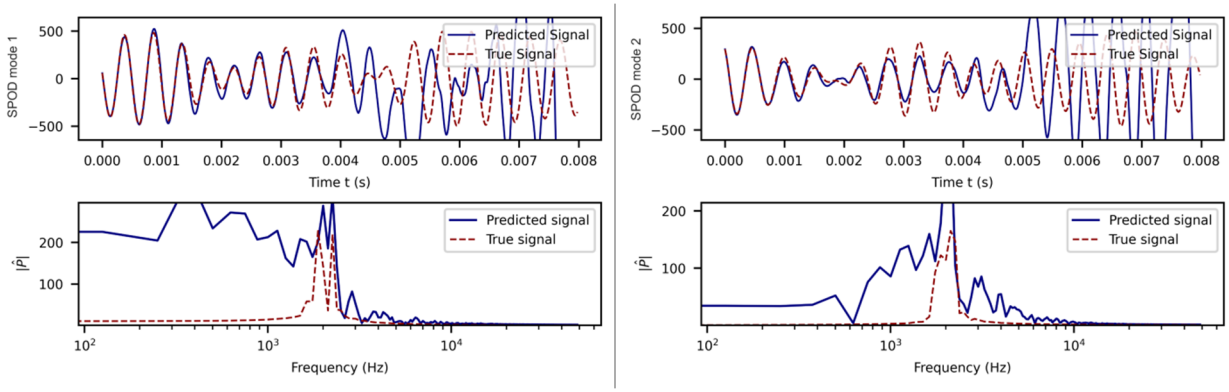


Figure 12 – Input data of the SPOD temporal coefficient for St=0.5, showing the original and filtered Fourier Transforms for bandpassing at 2000 Hz and bandwidth of 500 Hz. The bandpass signal and its FT magnitude are shown in blue(—) and the FT of original signal is shown in red(—). Dotted lines indicate truncation of the signal in time domain (left) before training and bandwidth of filter used in frequency domain (right).



(a) (b)

Figure 13 – ANN based reconstruction of the SPOD mode, a) mode 1 and b) mode 2, time signals band passed at mean frequency 2000 Hz and bandwidth 500 Hz. Top: ANN based reconstruction (—) of a bandpassed SPOD signal (--) showing improved stability; Bottom: The Fourier transform of

the predicted signal (—) and that of original signal (--).

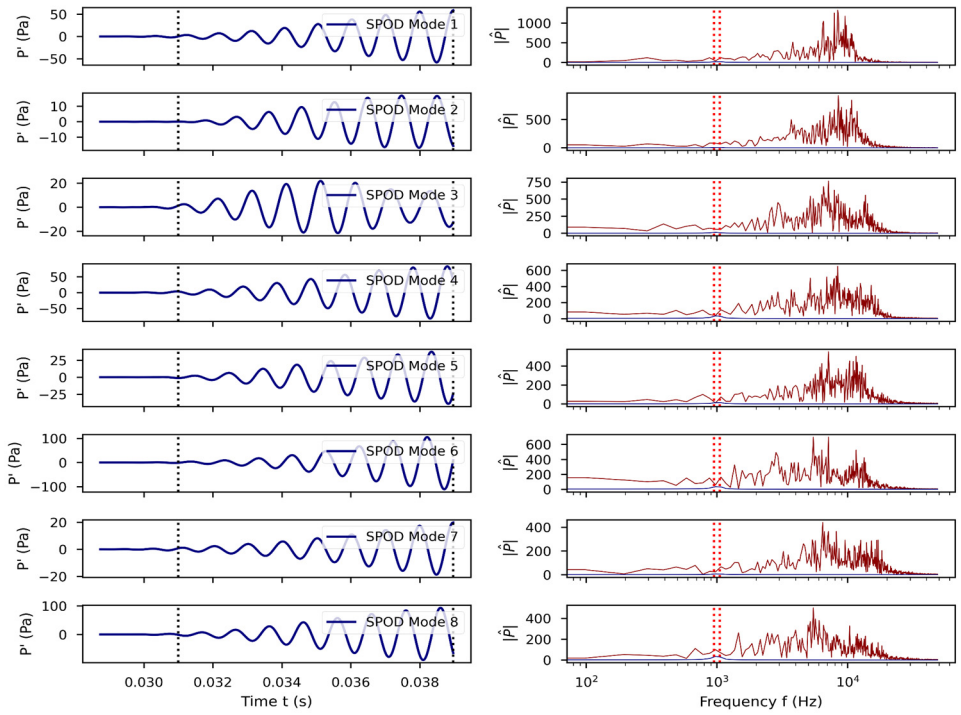
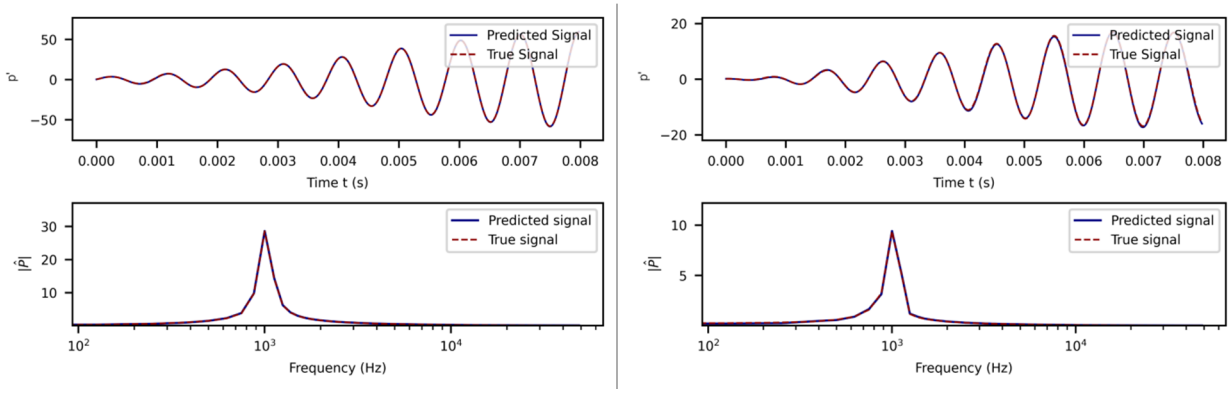


Figure 14 – Input data of SPOD modes 1 to 8, temporal coefficient for St=0.5, showing the original and filtered Fourier Transforms for an extreme case of bandpassing at 1000 Hz and a narrow bandwidth of 100 Hz. The bandpass signal and its FT magnitude are shown in blue(—) and the FT of original signal is shown in red(—). Dotted lines indicate truncation of the signal in time domain (left) before training and bandwidth of filter used in frequency domain (right).



(a) (b)

Figure 15 – ANN based reconstruction of the SPOD modes, a) mode 1 and b) mode 2, time signals band passed at mean frequency 1000 Hz and bandwidth 100 Hz. Top: ANN based reconstruction (—) of a bandpassed SPOD signal (--), showing perfect training accuracy due to strong bandpass filter; Bottom: The Fourier transform of the predicted signal (—) and that of original signal (--) are in good agreement.

3.2 Genetic Programming

To ease comparison with ANNs extrapolation performance, HyGP was tested on the same SP7 jet flow modelling problems, such as local velocity and pressure signals and the SPOD mode 1 as a function of time for specific combinations of azimuthal angle, radial position (normalised with respect to nozzle diameter D_j) and longitudinal position (normalised with respect to the nozzle diameter D_j), for a total of 13 test cases as detailed below:

- SP7 jet $u'(t)$ for one azimuthal angle θ_1 at $r/D_j = 0.5$ and $x/D_j = 1,2,4,5$
- SP7 $p'(t)$ for one azimuthal angle θ_1 at $r/D_j = 0.5$ and $x/D_j = 1,2,4,5$
- SP7 SPOD mode 1 of the pressure for different Strouhal numbers (St=01, 0.2, 0.5, 1, 2)

For all the test cases the HyGP experiments were set up using the same values of all the hyperparameters not relating to the definition of the multiobjective fitness function. In Table 2 the values of these common hyperparameters are shown.

Table 2 – HyGP parameters used to evolve SP7 models.

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 (full factorial)
Size of testing data set (and DoE type)	5348 (full factorial) - first 3000 identical to training data set
Sampling frequency (Hz) (normalised space)	300

The parameters and values reported in the table were selected based on previous experience with HyGP, so no specific optimisation other than trial and error was adopted.
A different approach was used to define the coefficient a_8 that defines the relative contribution or weight of the difference in variance and mean trend of the evolved models on the overall fitness of the model, balancing therefore model local accuracy (RMSE) with statistical properties like variance and mean over the design space. The optimal value of the coefficient was selected through preliminary GP experiments on the $u'(t)$ and $p'(t)$ SP7 test cases using coefficient values ranging a spectrum of 4 orders of magnitude (from 0.5 to 0.0001). A Pareto front was used to understand the balance between the two objectives, local accuracy and variance of the evolved models, for each experiment set up with a specific value of the coefficient a_8. The selected coefficient value was taken from the Pareto subset closest to the origin of the objectives' axes after analysis of the behavior of the corresponding models (a posteriori selection).
To give a more practical idea of the optimisation procedure adopted based on Pareto front, Fig.16 shows the bidimensional Pareto front on the Variance-RMSE space of the GP experiments on SP7 velocity $u'(t)$ signal run for coefficient a_8 values of 0.5, 0.1, 0.01, 0.001, 0.0001. Five different clusters corresponding to the independent evolutions (runs) of each experiment are clearly visible.

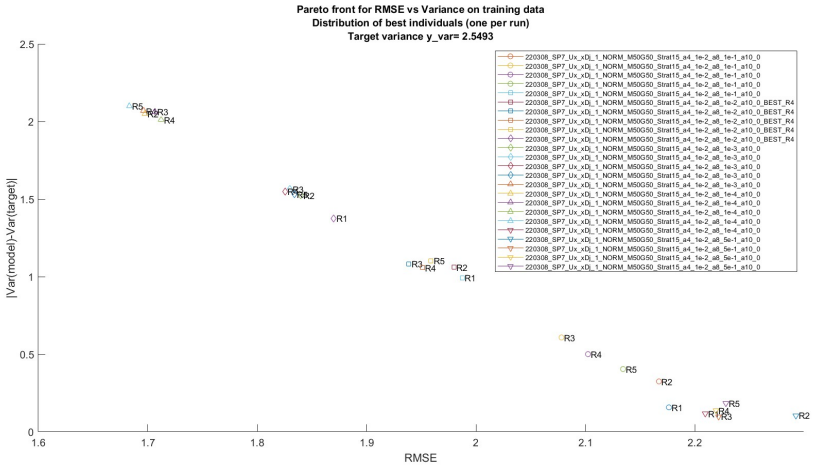


Figure 16 – Pareto front of five HyGP experiments for SP7 jet $u'(t)$ set up with different values of a_8.

The selected value was a_8 = 0.01: the corresponding HyGP model superimposed to the original $u'(t)$ signal is shown in Fig. 17. The vertical line at Z1=11 represents the end of the training data sets, so all the data beyond the line is extrapolated and compared with the test data set.

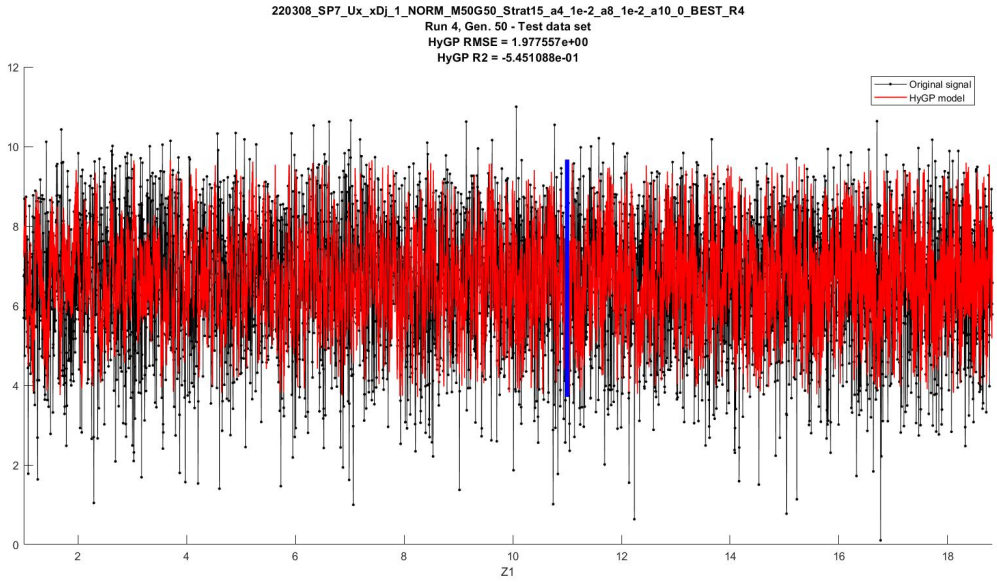


Figure 17 – SP7 jet $u'(t)$ signal superimposed to selected HyGP model

Similar behaviour was shown for all cases of SP7 velocity and pressure signals, so no further details are provided on GP capability on these test cases.
The same approach used to model SP7 velocity and pressure signals was used to generate SPOD models, with the only difference that coefficient a_8 was not directly optimised for all these cases, preferring to rely on the range of best values already identified during the preliminary analyses performed on SP7 velocity $u'(t)$ and pressure $p'(t)$. In the following figures the best results returned by HyGP for the first SPOD mode of SP7 jet are presented whereas the corresponding expression of the model is reported in the Appendix.

- **SP7 SPOD mode 1, St=0.1 (coefficient a_8=0.1)**

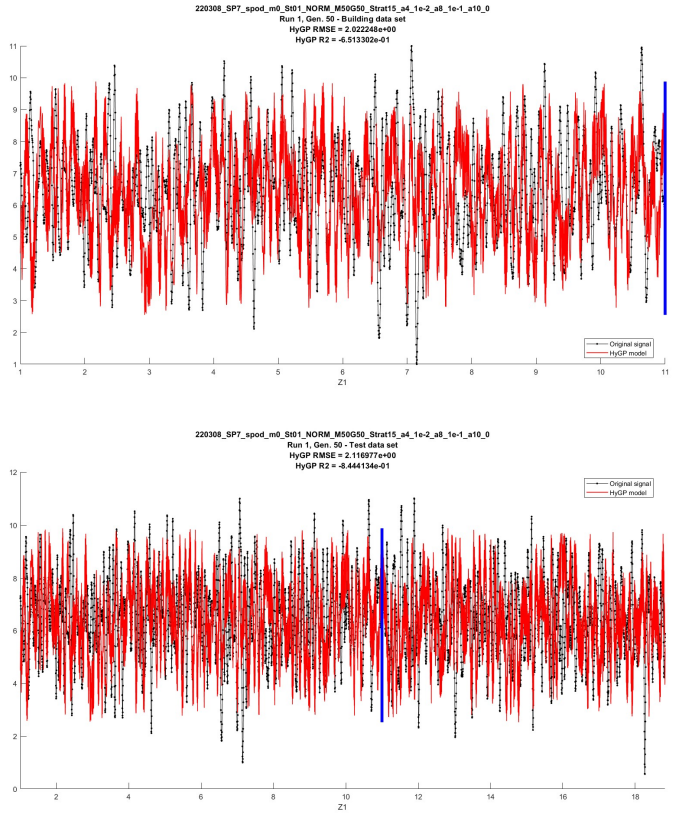
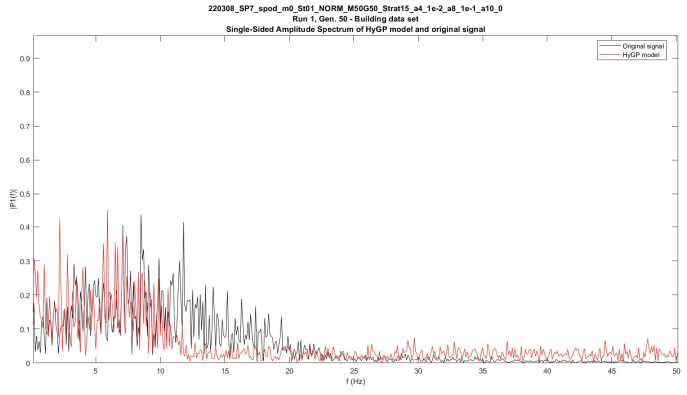


Figure 18 – Plots of SP7 jet first SPOD mode at St = 0.1 original signal superimposed to best HyGP model: training (top) and test data sets (bottom, after vertical line) in the normalised space.

In Fig. 18 the curves of the first SPOD mode at St=0.1 original signal and the best HyGP model are shown, whereas in Fig. 19 the corresponding spectrum in the normalised space is presented.



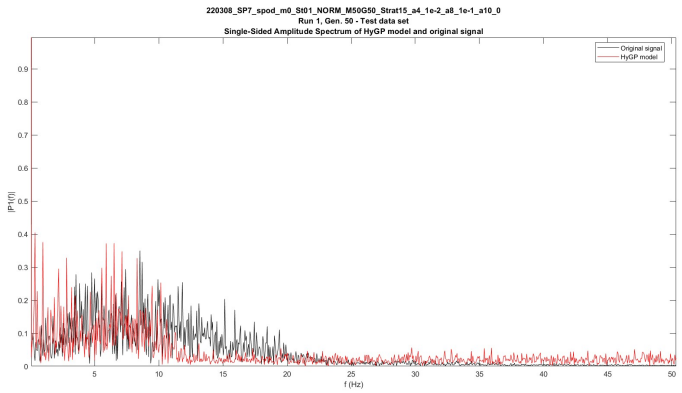


Figure 19 – Plots of frequency spectra of the first SPOD mode at St=0.1 original signal and HyGP model on training and test data sets (normalised space)

- **SP7 SPOD mode 0 St=0.2 (coefficient a_8=0.1)**

The curves of the first SPOD mode at St=0.2 original signal and the best HyGP model are shown in Fig. 20, whereas in Fig. 21 the corresponding spectra in the normalised space are shown. Similarly to St=0.1, the frequency spectra of the original signal and the model exhibit a good match.

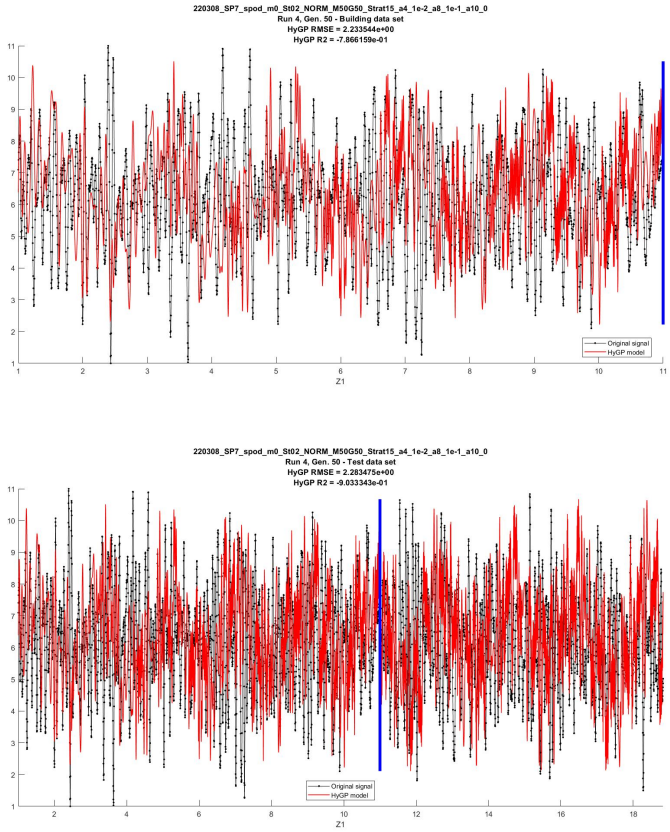


Figure 20 – Plots of the first SPOD mode at St=0.2 original signal superimposed to HyGP model: training (top) and test data sets (bottom, after vertical line) in the normalised space.

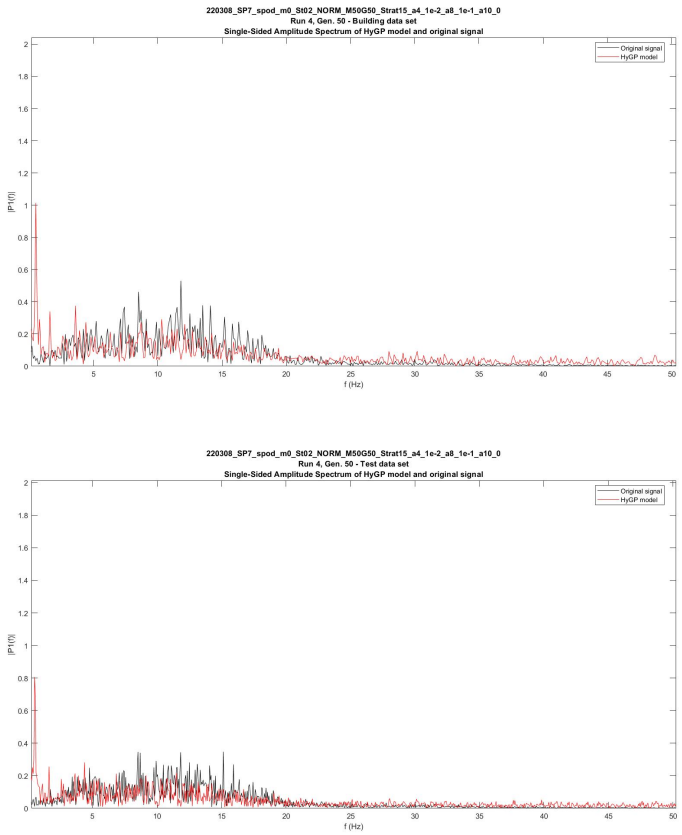
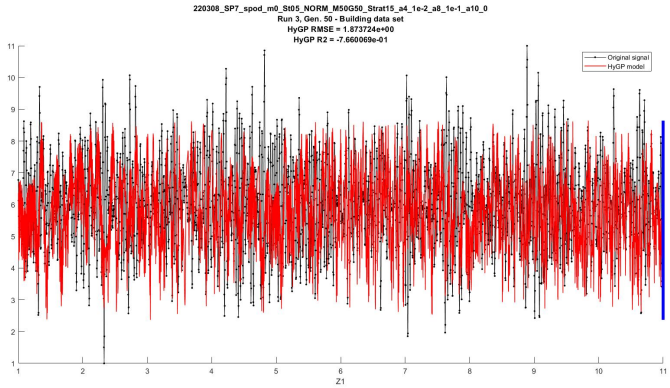


Figure 21 –Plots of frequency spectra of the first SPOD mode at St=0.2 original signal and HyGP model on training and test data sets (normalised space).

- **SP7 SPOD mode 0 St=0.5 (coefficient a_8=0.1)**

The curves of the first SPOD mode at St=0.5 original signal and the best HyGP model are shown in Fig. 22, in Fig. 23 the corresponding spectra in the normalised space are plotted. Differently from the previous cases, the ability of HyGP to reproduce a frequency spectrum similar to the original signal is reduced.



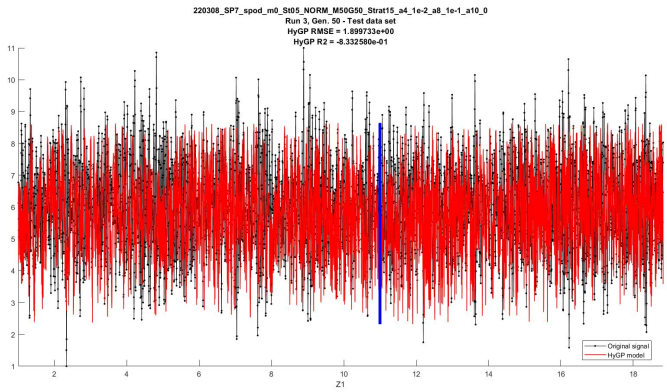


Figure 22 – Plots of the first SPOD mode at St=05 original signal superimposed to HyGP model: training (top) and test data sets (bottom, after vertical line) in the normalised space.

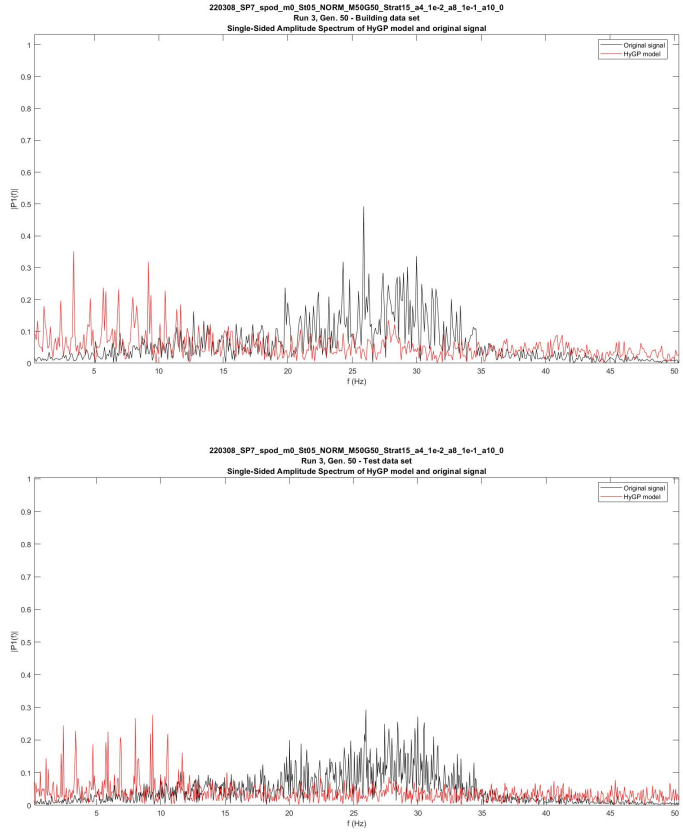


Figure 23 – Plots of frequency spectra of the first SPOD mode at St=05 original signal and HyGP model on training and test data sets (normalised space)

- **SP7 SPOD mode 0 St=1.0 (coefficient a_8=0.1)**

The curves of the first SPOD mode at St=1.0 original signal and the best HyGP model are shown in Fig. 24, in Fig. 25 the corresponding spectra in the normalised space are plotted. The trend of decaying HyGP capability in reproducing the original frequency spectrum is confirmed.

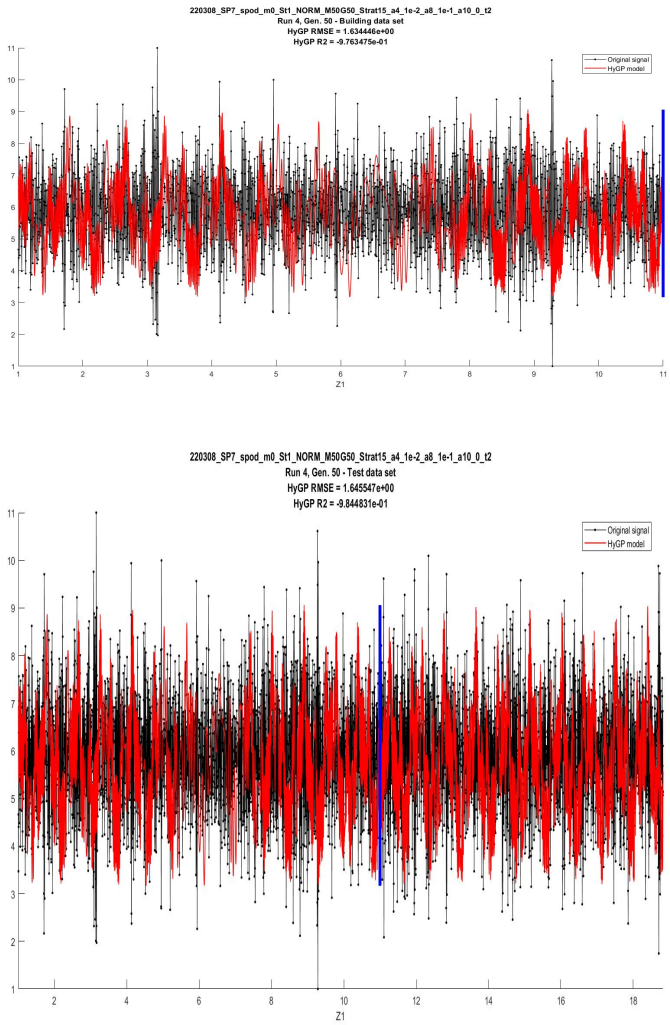
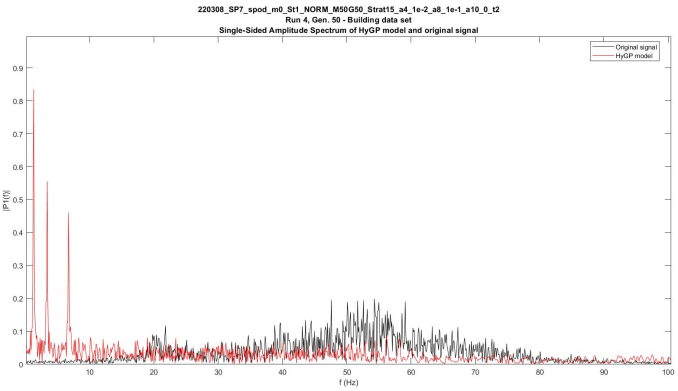


Figure 24 – Plots of the first SPOD mode at St=1 original signal superimposed to HyGP model: training (top) and test data sets (bottom, after vertical line) in the normalised space.



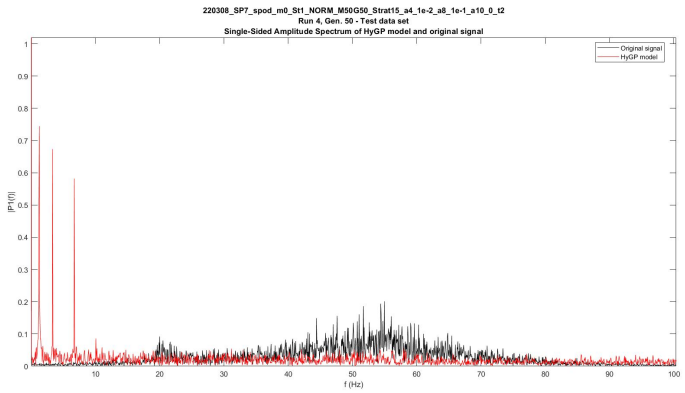


Figure 25 – Plots of frequency spectra of the first SPOD mode at St=1 original signal and HyGP model on training and test data sets (normalised space)

SP7 SPOD mode 0 St=2.0 (coefficient a_8=0.1)

The curves of the first SPOD mode at St=2.0 original signal and the best HyGP model are shown in Fig. 26, in Fig. 27 the corresponding spectra in the normalised space are plotted. HyGP capability in reproducing the original frequency spectrum dramatically worsened for this further increase in Strouhal number.

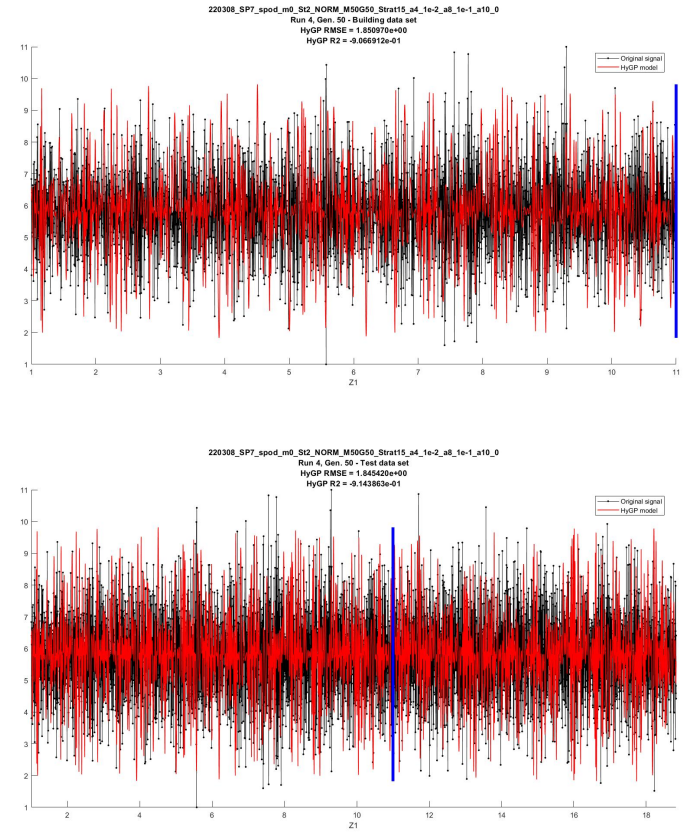


Figure 26 – Plots of the first SPOD mode at St=2 original signal superimposed to HyGP model: training (top) and test data sets (bottom, after vertical line) in the normalised space.

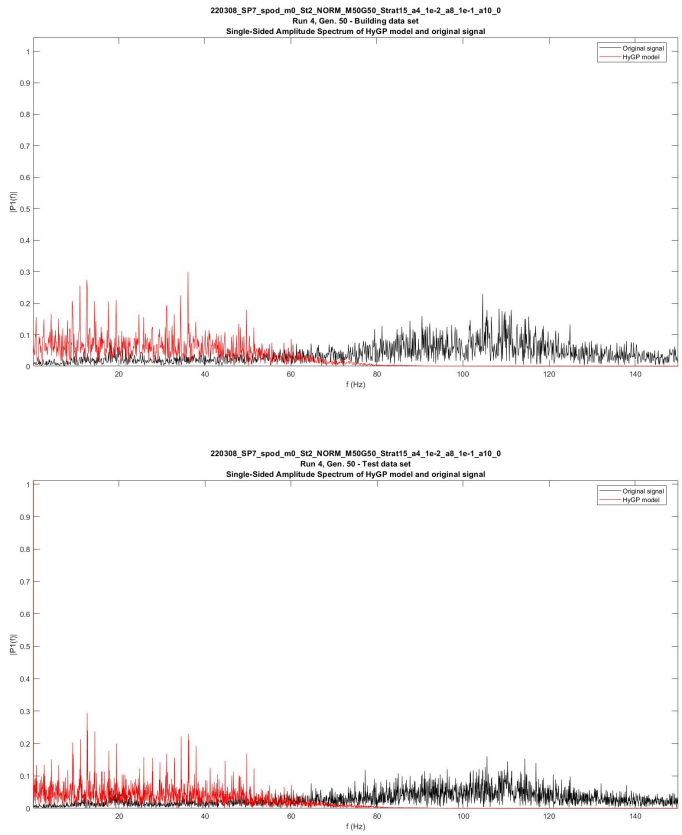


Figure 27 – Plots of frequency spectra of the first SPOD mode at St=2 original signal and HyGP model on training and test data sets (normalised space)

4. Conclusions

To test the Artificial Neural Network (ANN) method for reconstruction of turbulent jet flow signals, we chose the feedforward neural networks framework previously implemented for a turbulent flow around a plunging aerofoil in [16]. In this framework, ANN acts as a differential operator in time on the signal extracted from the Large Eddy Simulation solution, thereby considering just the signal amplitude as input. The framework was previously shown to have various good properties enabling the ANN algorithm to extrapolate periodic time series signals. However, in application to high-speed jet flow, the method is found to be inadequate to handle the resulting high bandwidth signals. This raises a question of why the ANN based framework could successfully learn the signals from a spectrally filtered POD of the datum flow dataset in the original publication. We suggest that the answer to this question is in the properties of the specific POD method, which involves a smoothing of the correlation matrix and subsequent pairing of signals and their tonal nature. This was evident from the fact that a spectral bandpassing of the broadband turbulent signal, rather than applying filtering to the correlation matrix which would consistently affect all trajectories, did not result in a successful extrapolation by the ANN. For the next step, the ANN method was tested on bandpass filtered signals from the Spectral POD (SPOD) method. In comparison with the original POD method of Seiber et al. [3], the selected SPOD method based on the algorithm developed in [31] enables accurate decoupling of the individual frequency modes of the LES dataset. Nevertheless, by applying a bandlimited filter of mean frequency 1000 Hz with bandwidth of 100 Hz it was possible to obtain a stable ANN reconstruction of the original signal. For stability, it was required to include 8 signals corresponding to the time coefficients of different SPOD modes.

It should be emphasized that the capability of ANN to extrapolate the signal was tested within the competition space in terms of compute time. In this activity, the random search algorithm was limited to test 100 random models. The range of values allowed for the number of hidden layers were from 6 to 10, layer widths in the range of 10 to 64 and regularization parameters in the range 10^{-2} to 10^{-6}. A total of 800 times steps were used for training all models and cases. The total compute time was thus kept limited to 11 hours on average for serial random search algorithm,

which is larger than that committed to the GP algorithm for the same task.
A further analysis of the preprocessing method that computes signal trajectories for ANN suggests that multiple signal input to the ANN is equivalent to training the network along a high dimensional trajectory formed by the signal amplitude values. A closed loop trajectory of the signal is thus equivalent to a limit cycle in the dynamical system perspective. For example, it may be suggested that compared to the Spectral POD method in [31] the method of filter-based POD [3] preserves the requited symmetries in the high dimensional trajectory space, thereby producing signals with inherently smoother properties, which makes them better suitable for the ANN algorithm. The smoothing of the correlation matrix is not equivalent to filtering of the individual principle time coefficients, hence, the use of individually filtered SPOD signals did not significantly improve stability of the ANN method for the jet application. Future work will include utilizing the filter-based POD algorithm for the LES solution of high Reynolds number jet case followed by the ANN algorithm.
On the Genetic Programming (GP) front, the lack of signal smoothness that proved critical for ANNs does not impair the GP capability to return an approximation model. So GP proves to be a robust modeling technique provided that primitives are selected properly. In this case addition, subtraction, multiplication, sine and cosine allowed for a bounded model to be extrapolated. In case of random generation of diverging polynomials, the editing operation was able to ensure robustness of the GP search. Moreover, there are no constraints in terms of the number of input signals that have to be provided to GP for it to be effective, as it occurs for ANNs.
Despite the robustness, quality of the generated models appeared to be critically dependent on the definition of the fitness function objectives. specifically the balance between model local accuracy (RMSE) and statistical properties (variance and mean). In previous experiments, not shown in the paper, also the first root and decaying speed of the autocorrelation function were used as objectives, but the additional computational cost did not bring any additional advantage so the approach was eventually abandoned. The optimal definition of the fitness function objective coefficients (weights) as well as the optimal set up of the many hyperparameters are the weak point of GP as a modeling tool, which leaves its effective use critically dependent on the user's experience.
Automatic strategies for hyperparameters optimization are viable, and Random Search as used for ANNs could represent a possible approach. Similar to ANN, GP is affected by a resolution problem. For example, the SPOD signals were generated for increasing Strouhal numbers, meaning that the peak of the spectrum of the principle shifts towards higher frequencies for higher Strouhal numbers. At the same time, the number of sampling points in the training data set was kept the same (3000), so as a result the number of sampled points per period on average decreased for higher frequencies. In other words, the higher the frequency, the fewer the points used to reconstruct it for the fixed sampling rate. Hence, the drop in GP quality model from SPOD St=0.5 to St=2 may be ascribed to this reduction in resolution, and this behavior is also observed in ANNs experiments. Therefore, the key to enable GP to reconstruct high frequency signals is the increase of sampling points per period. It is suggested that to correctly reconstruct a signal GP needs from 20 to 50 points per period of the time frequency which needs to be captured. Notably, previous experiments with band filtering did not show improved results, neither on velocity and pressure signals nor SPOD components, Hence, the under performance of GP for high frequencies of the SPOD signal cannot be related to the broadband nature of the signal. Therefore, the next step in this direction will be to analyze the SPOD signals at a refined sampling rate for which an appropriate LES dataset will need to be generated.
While the GP stochastic nature brings increased robustness to the model in comparison with ANN, 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. If the final application of the 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, which expression can be used as a surrogate model of the original turbulent signal.

Copyright Statement

References

[1] Brunton S. L. and Noack B. R. Closed-Loop Turbulence Control: Progress and Challenges. In Applied Mechanics Reviews. *ASME International*. Vol. 67, Issue 5, 2015. https://doi.org/10.1115/1.4031175

[2] Lumley J. L. The structure of inhomogeneous turbulence. *Atmospheric Turbulence and Wave Propagation*, Moscow: Nauka. pp. 166–178, 1967.

[3] Sieber M. Paschereit C. O. and Oberleithner K. Spectral proper orthogonal decomposition. *Journal of Fluid Mechanics*. *Cambridge University Press*. Vol. 792, pp. 798–828, 2016. https://doi.org/10.1017/jfm.2016.103

[4] 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, Issue 8, p. 085104, 2017. https://doi.org/10.1063/1.4997202

[5] Schmid P. J. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, Vol. 656, pp. 5–28, 2010. https://doi.org/10.1017/s0022112010001217

[6] Tu J. H. Rowley, C. W. Luchtenburg, D. M. Brunton S. L., Kutz J. N. On Dynamic Mode Decomposition: Theory and Applications, 2013. https://doi.org/10.48550/ARXIV.1312.0041

[7] Le Clainche, S., and Vega, J. M. Higher order dynamic mode decomposition to identify and extrapolate flow patterns. *Physics of Fluids,* Vol. 29, Issue 8, p. 084102, 2017. https://doi.org/10.1063/1.4997206

[8] 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, Issues 1–2, pp. 115–129, 2004. https://doi.org/10.1016/j.physd.2003.03.001

[9] 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, Issue 15, pp. 3932–3937, 2016. https://doi.org/10.1073/pnas.1517384113

[10] Carlberg K., Bou-Mosleh C. 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 (2), pp. 155–181, 2011.

[11] Östh J., Noack B. R., Krajnović S., Barros D., 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, pp. 518–544, 2014. https://doi.org/10.1017/jfm.2014.168

[12] Protas B., Noack B. R., Östh, J. Optimal nonlinear eddy viscosity in Galerkin models of turbulent flows. *Journal of Fluid Mechanics,* Vol. 766, pp. 337–367, 2015. https://doi.org/10.1017/jfm.2015.14

[13] San O., and Maulik, R. (2018). Extreme learning machine for reduced order modeling of turbulent geophysical flows, *ArXiv*. https://doi.org/10.48550/ARXIV.1803.00222

[14] Wan Z. Y., Vlachas P., Koumoutsakos P., Sapsis T. Data-assisted reduced-order modeling of extreme events in complex dynamical systems. *D. Durstewitz (Ed.)*,Vol. 13, Issue 5, p. e0197704, 2018. https://doi.org/10.1371/journal.pone.0197704

[15] Rudy S. H., Kutz J. N., Brunton S. L. Deep learning of dynamics and signal-noise decomposition with time-stepping constraints. *ArXiv*, 2018, https://doi.org/10.48550/ARXIV.1808.02578

[16] Lui, H. F. S., and Wolf, W. R. (2019). Construction of Reduced Order Models for Fluid Flows Using Deep Feedforward Neural Networks. ArXiv. https://doi.org/10.48550/ARXIV.1903.05206

[17] 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 M´ecanique,* Vol. 346, 10, ,

pp. 948–963, 2018. https://doi.org/10.1016/j.crme.2018.07.004

[18]Goloviznin V. M., Samarskii A. A., Finite difference approximation of convective transport equation with space splitting time derivative, *Jour Matem. Mod*., Vol.10(1), pp. 86–100, 1998.

[19]Karabasov S. A., and Goloviznin V. M., Compact Accurately Boundary Adjusting high-REsolution Technique for Fluid Dynamics, *Journal of Computational Physics*, Vol.228, pp. 7426-7451, 2009. https://doi.org/10.1016/j.jcp.2009.06.037

[20]Semiletov V. and Karabasov S. A., CABARET scheme with conservation-flux asynchronous time-stepping for nonlinear aeroacoustics problems, *Journal of Computational Physics*, Vol. 253, pp. 157–165, 2013. https://doi.org/10.1016/j.jcp.2013.07.008

[21]Semiletov V.A. and Karabasov S.A., CABARET scheme for computational aero acoustics: extension to asynchronous time stepping and 3D flow modelling, *International Journal of Aeroacoustics*, Vol.13 (3-4), 321-336, 2014. https://doi.org/10.1260/1475-472X.13.3-4.321

[22]Chintagunta A., Naghibi S.E., Karabasov S.A. Flux-corrected dispersion-improved CABARET schemes for linear and nonlinear wave propagation problems, *Computers & Fluids*,Vol.169, pp.111-128, 2017. https://doi.org/10.1016/j.compfluid.2017.08.018

[23]Faranosov G. A., Goloviznin V. M., Karabasov S. A., Kondakov V. G., Kopiev V. F., Zaitsev M. A., CABARET method on unstructured hexahedral grids for jet noise computation, *Computers & Fluids*, Vol. 88, pp. 165–179, 2013. https://doi.org/10.1016/j.compfluid.2013.08.011

[24]Semiletov V. A., Yakovlev P. G., Karabasov S. A., Faranosov G. A., 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, pp. 631–645, 2016. https://doi.org/10.1177/1475472X16659387

[25]Markesteijn A.P., Karabasov S.A., 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, pp. 20190083, 2019. https://doi.org/10.1098/rsta.2019.0083

[26]Markesteijn A. P., Gryazev V., Karabasov S. A., Ayupov R. S., Benderskiy L. A., Lyubimov D. A., Flow and Noise Predictions of Coaxial Jets, *AIAA Journal*, Vol. 58, No. 12, pp. 5280–5293, 2020. https://doi.org/10.2514/1.J058881

[27]Gryazev V., Kalyan A., Markesteijn A.P., Karabasov S.A., Broadband shock-associated noise modelling for high-area-ratio under-expanded jets, *The Journal of the Acoustical Society of America*, Vol. 150 (2), pp. 1534-1547, 2021. https://doi.org/10.1121/10.0005976

[28]Koza J.R. Genetic Programming: On the Programming of Computers by Means of Natural Selection, *MIT press,* Cambridge, USA, 1992.

[29]Toropov V.V., Polynkin A., Armani U., Alvarez L.F. 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, 16-21 May 2010.

[30]Armani U., Khatir Z., Khan A., Toropov V.V., Polyinkin A., Thompson, H., Kapur N., 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.

[31]Towne A., Schmidt O. T., Colonius T. Spectral proper orthogonal decomposition and its relationship to dynamic mode decomposition and resolvent analysis. *Journal of Fluid Mechanics*, Vol. 847, pp. 821867, 2018. https://doi.org/10.1017/jfm.2018.283

[32]Schmidt O. T., Towne A., Rigas G., Colonius T., Brès G. A. Spectral Analysis of Jet Turbulence. *Journal of Fluid Mechanics*, Vol. 855, pp. 953–982, 2018. https://doi.org/10.1017/jfm.2018.675

[33]Sonoda S. and Murata N., Neural network with unbounded activation functions is universal approximator, Appl. Comput. Harmon. Anal., vol. 43, no. 2, pp. 233–268, 2017, doi: https://doi.org/10.1016/j.acha.2015.12.005.

[34]Cybenko G., 1989. Approximation by superpositions of a sigmoidal function. Mathematics of control, signals and systems, 2(4), pp.303-314

[35]Ronen, B., Jacobs, D., Kasten, Y. and Kritchman, S., 2019. The convergence rate of neural networks for learned functions of different frequencies. Advances in Neural Information Processing Systems, 32.

[36]Kommenda, M., Burlacu, B., Kronberger, G. et al. Parameter identification for symbolic regression using nonlinear least squares. Genet Program Evolvable Mach 21, 471–501 (2020). https://doi.org/10.1007/s10710-019-09371-3.

[37]Martens, Izzo, "Symbolic Regression for Space Applications: Differentiable Cartesian Genetic Programming Powered by Multi-objective Memetic Algorithms", European Space Agency, Noordwijk, 2201 AZ, The Netherlands, 2022 https://arxiv.org/abs/2206.06213

[38]J. Kennedy, R. Eberhart, 1995 Particle swarm optimization, Proceedings of the IEEE International Conference on Neural Networks, 4:1942—1948, IEEE.

[39]K. Madsen, H. B. Nielsen, and J. Sondergaard. Robust subroutines for non-linear optimization. Technical report IMM-REP-2002-02, Technical University of Denmark, 2002.

[40]Armani U. Development of A Hybrid Genetic Programming Technique for Computationally Expensive Optimisation Problems, *PhD Thesis*, University of Leeds, 2014. https://hygprogramming.wordpress.com/

[41]R. Poli, W. B. Langdon, N. F. McPhee, *A Field Guide to Genetic Programming (With contributions by J. R. Koza)*, 2008, http://www.gp-fieldguide.org.uk/

Appendix

The models generated by HyGP for SP7 mode 0 St=0.1, 0.2, 0.5, 1.0 and 2.0 are reported in the following. The computing time requested to evolve the model is also indicated (all experiments were run on a laptop computer with linux OS, processor Intel(R) Core(TM) i7-4710HQ 2.50GHz, 2501 Mhz in sequential mode, so the indicated time is the wall clock time)

SP7 SPOD mode 0, St=0.1: Run 1, Generation 50 (time = 1284 s)

Y(Z1) = (6.2132813177e+00 + ((1.5816113444e+00 * (cos(((((1.3521723310e+00 * Z1) - (5.8317634022e+00 * Z1)) + (9.8650043832e+00 * (sin((5.8087979615e+00 * Z1))))) + (((5.5662596779e+00 * Z1) - (3.2698471150e+00 * Z1)) + (8.4864471317e+00 * (cos((1.3284981389e+00 * Z1)))))))) + ((1.1410925588e+00 * (sin(((6.9281489991e+00 * (cos((((1.0535105789e+00 * Z1) - (2.9200839133e+00 * Z1)) + ((3.9347628306e+00 * Z1) + (7.6717547294e+00 * Z1)))))) * (8.6970827737e+00 * Z1))))) + (1.0000000000e+00 * (cos(((((9.8248552517e+00 * Z1) - (3.2263526136e+00 * Z1)) + ((5.1265043393e+00 * Z1) + (3.5640234755e+00 * Z1))) + ((((9.2992381900e+00 + (6.2156951133e+00 * Z1)))^1) - (6.4269411591e+00 * (sin((3.8683018621e+00 * Z1))))))))))

SP7 SPOD mode 0, St=0.2: Run 4, Generation 50 (time = 1608 s)

Y(Z1) = (6.3489902274e+00 + (((9.7831494828e-01 * (sin((8.6496650843e+00 * Z1)))) * (2.4117795259e+00 * (sin((((9.9552547286e+00 * (sin((5.3926160646e+00 * Z1)))) * (1.0001328845e+00 * (sin((1.0990716252e+01 * Z1))))) + (9.2249751253e+00 * (sin((1.0071058518e+00 * Z1)))))))) + ((9.7186564817e-01 * (sin(((((1.0156430829e+01 * (sin((1.6470705803e+00 * Z1)))) - (((4.8645036779e+00 + (3.2365409302e+00 * Z1)))^1)) + (((6.6436008721e+00 + (4.2657790931e+00 * (sin((9.2479876782e+00 * Z1))))))^1)) + ((((4.6894846547e+00 * Z1) + (2.5877947705e+00 * Z1)) + (2.1958686276e+00 * Z1)) * (1.2220579641e+00 * (sin((3.6646869468e+00 * Z1)))))))) - (1.1406041372e+00 * (sin((3.3343808250e+00 * Z1)))))))"

SP7 SPOD mode 0, St=0.5: Run 3, Generation 50 (time = 1632 s)

Y(Z1) = (5.4773568325e+00 + ((1.1886089006e+00 * (cos((6.5730036615e+00 * (cos((((3.7551455443e+00 * (sin((3.6590388299e+00 * (Z1 * Z1))))) + ((((2.4829021118e+00 * Z1) - (8.6056615309e+00 * Z1)) - (((2.5819287949e+00 + (1.0999264473e+01 * Z1)))^1)) * (((7.7305648311e+00 + (((1.0000000000e+00 + (5.6919515213e+00 * Z1)))^1)))^1))) - (((9.3247888150e+00 + (5.7084314549e+00 * Z1)))^1)))))))) - ((1.0000000000e+00 *

(sin(((1.0165013835e+01 * (sin((1.0000000000e+00 * Z1)))) * (2.6890803179e+00 * Z1))))) - (1.0000000000e+00 * (cos(((7.8306693236e+00 * Z1) + (8.9796782181e+00 * (sin((7.2855177977e+00 * Z1))))))))))"

SP7 SPOD mode 0, St=1.0: Run 4, Generation 50 (time = 1406 s)

Y(Z1) = (6.1315496780e+00 + (((1.0000000000e+00 * (sin((7.9577247137e+00 * Z1)))) + (1.0000000000e+00 * (sin(((((1.0999975788e+01 + ((9.2976921411e+00 * Z1) - (1.1001000000e+01 * Z1))))^1) * ((2.3102626909e+00 * (sin((5.9864785922e+00 * Z1)))) * (((1.6233382948e+00 + (1.0000000000e+00 * Z1)))^1)))))))) + (1.0000000000e+00 * (cos((4.0094577122e+00 * (sin((1.0581024175e+01 * Z1))))))))"

SP7 SPOD mode 0, St=2.0: Run 4, Generation 50 (time = 1380 s)

Y(Z1) = (5.8102261719e+00 + (((2.0965308200e+00 * (cos(((6.8116255137e+00 * (sin((2.4128073693e+00 * Z1)))) * (4.1703040938e+00 * (sin((3.6685165274e+00 * (sin((1.6094374972e+00 * Z1))))))))))) - (1.4904010879e+00 * (cos((((5.6656835462e+00 + (5.7143657353e+00 * Z1)))^1))))) * (1.1264370241e+00 * (cos((8.1142724845e+00 * (cos((7.7000794209e+00 * (sin((5.2847809570e+00 * Z1))))))))))))"