AIAA SciTech Forum
6-10 January 2020, Orlando, FL
AIAA Scitech 2020 Forum

10.2514/6.2020-1533

Multivariate Recurrent Neural Network Models for Scalar and Distribution Predictions in Unsteady Aerodynamics

Qingzhao Wang,* Carlos E. S. Cesnik,† and Krzysztof J. Fidkowski‡
University of Michigan, Ann Arbor, MI, 48109, USA

In this work, recurrent neural networks (RNNs) are used to create output-based reduced-order models for unsteady aerodynamic problems. Two RNN architectures are employed for model creation and are evaluated on accuracy and generalization. The target test case consists of a pitching and plunging NLR 7301 airfoil in the transonic flow regime in the presence of shock motion. RNNs learn the underlying relationship between the inputs (in this study, the Mach number, mean angle of attack, and pitching and plunging motion parameters) and the outputs (in this study, the lift coefficient and pressure coefficient distributions) from a set of time series samples. Various techniques are implemented and investigated for improving the generalization ability of RNN models in predicting both scalar and distributed quantities.

Nomenclature

A	=	amplitude
A_i	=	amplitude for individual oscillations
A_m	=	mean amplitude
α_m	=	mean angle of attack
$\Delta\alpha$	=	amplitude of pitch angle
B_f, B_i, B_o, B_C	=	biases for forget gate, input gate, output gate and cell update
c	=	chord length
C_1	=	amplitude parameter for randomlike signal
$C_{L,\alpha}$	=	ratio of Fourier transforms of lift coefficient fluctuation and pitch motion
C_t	=	cell state at time t
$\hat{C}_t$	=	cell candidate at time t
C_{px}	=	x-component of pressure coefficient
C_{py}	=	y-component of pressure coefficient
C_{sx}	=	x-component of viscous stress coefficient
C_{sy}	=	y-component of viscous stress coefficient
d	=	delays in NARX neural networks
F_{base}	=	base function for randomlike signal
F_{GP}	=	function for Gaussian pulse type signal
F_{noise}	=	analytical noise function for randomlike signal
F_{rand}	=	function for randomlike signal
F_{SIH}	=	function for superimposed harmonic signal
f_i	=	frequency for individual oscillations
f_m	=	mean frequency
f_t	=	forget gate at time t
h_{t-1}, h_t	=	hidden states of LSTM recurrent neural networks at time t-1 and t
Δh	=	amplitude of vertical displacement
i	=	index
i_t	=	input gate at time t
k	=	reduced frequency based on full chord
l_{eff}	=	coefficient for Gaussian pulse type signal

*Postdoctoral Research Fellow, Department of Aerospace Engineering, qzwang@umich.edu, AIAA Member.
†Clarence L. "Kelly" Johnson Collegiate Professor of Aerospace Engineering, cesnik@umich.edu, AIAA Fellow.
‡Associate Professor, Department of Aerospace Engineering, kfid@umich.edu, AIAA Senior Member.

m	=	number of elements in each column of S
N	=	number of single harmonics in superimposed harmonic signal
n_s	=	number of snapshots
o_t	=	output gate at time t
$\boldsymbol{P}$	=	POD mode coefficients
R_f, R_i, R_o, R_C	=	weights for recurrent states for forget, input and output gates and cell update
$\boldsymbol{S}$	=	snapshots matrix
$s_{1/2}$	=	half of simulation time
t	=	time
t_0	=	initial time
t_{s1}, t_{s2}	=	scaled time
tanh	=	hyperbolic tangent function
$\mathbf{u}$	=	inputs of NARX recurrent neural networks
$\boldsymbol{U}$	=	unitary matrix composed of left singular vectors of $\boldsymbol{S}$
um	=	number of input delays in NARX neural networks
$\boldsymbol{V}$	=	unitary matrix composed of right singular vectors of $\boldsymbol{S}$
W_f, W_i, W_o, W_C	=	weights for inputs for forget, input and output gates and cell update
x_t	=	inputs to recurrent neural networks at time t
$\mathbf{y}$	=	outputs of recurrent neural networks
yn	=	number of feedback delays in NARX neural networks
$\boldsymbol{\sigma}$	=	sigmoid function
$\boldsymbol{\Sigma}$	=	diagonal matrix composed of singular values of $\boldsymbol{S}$
Φ	=	phase for oscillations
θ_i	=	modeling parameters in cross-validation
$\circ$	=	Hadamard product operation

I. Introduction

Although high-fidelity Computational Fluid Dynamics (CFD) analysis has become indispensable in design, optimization, and control of complicated aerodynamic systems, its application to industrial problems encounters challenges due to the considerable cost. Reduced-order modeling aims to provide a surrogate model that requires low to moderate cost while maintaining acceptable accuracy and robustness. The losses in accuracy resulting from model-order reduction need to be quantified in order for these models to be useful surrogates. The present work explores the effects of influential factors and evaluates the predictive ability of the reduced-order models (ROMs) that are generated using the recurrent neural network (RNN), a specific class of artificial neural networks (ANN), for unsteady aerodynamic applications.

Neural network-based ROMs approximate an aerodynamic system response by establishing a relationship between the inputs and outputs directly, avoiding the difficulties in tackling the governing equations and their discretization in CFD codes. In addition, the abilities of neural network models to simulate large-scale, highly nonlinear problems can meet the industrial needs for fast high-fidelity solutions.

Different types of RNNs have been developed and applied to performance analysis and design and optimization for dynamic systems. Suresh et al. [1] used recurrent multilayer perceptron networks to predict the lift coefficient at high angles of attack for a rotor blade and to identify its dynamic stall effects. Following work [2] that used the recurrent radial basis function neural network to investigate the structural responses and limit cycle oscillations of a pitching airfoil forced by a transonic flow, Winter and Breitsamter [3] computed the unsteady aerodynamic responses of the NLR 7301 airfoil to pitching and plunging coupled excitations, which matched well with the full-order CFD solution obtained from an Euler equation-based solver. The generation of training signals was addressed with the amplitude modulated pseudo-random binary signal to cover a representative spectrum of the excited amplitudes and frequency components within a relatively short time interval. The resultant signal contains sufficient information to capture the nonlinear features of the underlying system and reduces the computational cost in the training phase. The aforementioned RNN-based ROMs are all discretized in time. Mannarino and Mantegazza [4] proposed the use of continuous time recurrent neural networks to eliminate the influence of time step size on the subsequent analysis. Such a continuous time RNN model has been applied to an inviscid flow over a wing experiencing limit-cycle oscillations.

In most prior studies of RNN applications to unsteady aerodynamics [1, 5–8], the inputs are the variation of pitching

or plunging motions in time, while the bulk flow conditions have only been incorporated as inputs in a few test cases [1, 7]. Suresh et al. [1] trained RNNs using four data sets corresponding to four mean angles of attack, and predicted the lift coefficient responses to two different mean angles of attack. Kou and Zhang [7] proposed a multi-kernel neural network model to predict the aerodynamic loads at constant and varying Mach numbers. The prediction ability of RNNs has primarily been tested on integral quantities. Inspired by the work of Park et al. [9], Lindhorst et al. [10] and Winter and Breitsamter [11] combined the Proper Orthogonal Decomposition (POD) and neural networks to model the time-varying pressure distribution. One of the objectives of the present work is to create a multivariate RNN-based ROM with high accuracy and generalization ability for multi-step ahead predictions of distributed quantities in aerodynamic applications.

Generalization refers to the ability to predict outputs for unseen data. Implementation of the commonly-known methods, such as pruning [12], regularization [13] and early stopping [14], has shown to reduce generalization errors. May et al. [15] proposed a variant to the tangent plane algorithm for real-time recurrent learning, which pruned superfluous weights and led to good generalization over a wide range of network sizes. As the neural network becomes deeper, these traditional methods are no longer able to explain the fact that more hidden units can generalize well [14, 16, 17]. It is shown that the excess of hidden units promotes a good fit in the highly-nonlinear region of the model and early stopping prevents overfitting for the less-nonlinear region [14]. Neyshabur et al. [16] examined candidate complexity measures, and commented that some combination of sharpness and norms seems to reveal generalization ability. Zhang et al. [17] conducted systematic experiments to show the generalization behavior of large models can hardly be explained by traditional complexity measures.

The generalization ability of unsteady ROMs is discussed in this work pertaining to two RNN structures – long short-term memory (LSTM) [18] and nonlinear autoregressive network with exogenous inputs (NARX) [19] RNNs. Both RNNs were introduced to mitigate the vanishing gradient problem [20, 21] in the gradient-based algorithms for long-term dependency. Their structure allows the network to convey information from the distant past. LSTM has been widely used with much success. Recently, a variant of NARX RNN with exponential delays has been proposed and shown to outperform LSTM in some cases [22]. While this technique works well for general time-dependent problems, it is advantageous to apply the physics-based approach [23] when relevant information can be extracted from the studied system.

The remainder of the present paper is outlined as follows. Section II provides theoretical fundamentals of the POD representation of distributions and the LSTM and NARX RNNs. Section III defines the test cases used to demonstrate the application of RNNs to a transonic flow with moving shocks present. Section IV shows and discussed the representative results, followed by the concluding remarks in Section V.

II. Theoretical Fundamentals

A. POD Representation of Distributed quantities

Examples of inputs to the neural network model include the flow conditions, shape parameters, etc., while the outputs can be scalars, distributions, or fields. For the distribution variables, the model is further reduced by using basis vectors obtained from the POD method to represent the snapshots of distributions. For unsteady applications, the snapshots consist of the distributions at a series of time points.

Given a set of snapshots, $\boldsymbol{S} = [\boldsymbol{S}_1, \boldsymbol{S}_2, ...\boldsymbol{S}_{n_s}]$, with each $\boldsymbol{S}_i$ a column vector of m elements, the POD basis vectors can be obtained by a singular value decomposition as:

$$\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T. \tag{1}$$

Since the number of time steps (snapshots) is usually large when a number of sample time series are concatenated, m is less than n_s in the unsteady applications. The columns of $\boldsymbol{U}$ are the left singular vectors, namely the POD modes, and the columns of $\boldsymbol{V}$ are the right singular vectors of the matrix $\boldsymbol{S} \in \mathbb{R}^{m \times n_s}$. $\boldsymbol{\Sigma}$ is a diagonal matrix with m singular values for the diagonal entries in descending order.

In order to further reduce the order of the system, the number of POD modes is truncated based on the cumulative sum of the squared singular values, so that the selected modes contain sufficient energy of the spectrum, while avoiding overfitting and decreasing the computational cost. The amplitudes of the POD mode coefficients are treated as the targets in the RNN training process. Once the RNN-based ROM is created, the POD mode coefficients for independent input data can be predicted. The predicted POD coefficients are then used to reconstruct the distribution with the extracted

POD basis vectors:

$$\boldsymbol{S}_i \approx \boldsymbol{U}\boldsymbol{P}_i, \tag{2}$$

where $\boldsymbol{S}_i$ denotes the reconstructed distribution at a time snapshot i, which can be responses with respect to some testing inputs that do not belong to the training data sets. $\boldsymbol{P}_i$ is the set of POD mode coefficients predicted by RNNs.

B. Recurrent Neural Networks

Artificial neural networks imitate biological brains. Information travels between the artificial neurons. Based on the properties and complexity of specific problems, various neural network architectures have been developed and different machine learning algorithms have been applied to process the information received.

The goal of neural network-based ROMs is to predict the outputs for some unknown data set. To achieve this goal, neural networks learn from the available data set and build connections between neurons via transfer functions and weights and biases.

Implementation of neural network-based ROMs is inherently different for steady and unsteady problems because of the time dependency involved in the unsteady data. For unsteady applications associated with time series, recurrent neural networks are widely used. The architecture of an RNN neural network is shown in Figure 1 with x_t the input and h_t the direct output. In a RNN, a feedback loop is formed so that the previous outputs also contribute to the prediction of the output at the next time step. In practice, the simple RNNs perform poorly in learning the long-term dependency due to the vanishing gradient problem in gradient-based learning algorithms and backpropagation.

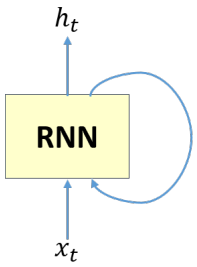


Fig. 1 Diagram of recurrent neural networks.

LSTM is a popular RNN architecture that alleviates the vanishing gradient problem. Figure 2 illustrates an LSTM neural network with unrolled LSTM units to represent the feedback loop in Figure 1. The input at the current time step x_t and the hidden state (the output of the hidden unit) at the previous time step h_{t-1} are combined to output the hidden state h_t that is fed into the next time step. C_t is the cell state (memory unit), a key element in LSTM networks. Inclusion of the cell state allows the gradients to flow unchanged, while still keeping the ability to add, remove, and modify information by three gates as shown in Figure 3. In the LSTM building block, there are four interacting layers, much more complex compared to only one in a simple RNN unit. With x_t and h_{t-1} input into the unit, first a forget gate f_t functions to decide which information will be thrown away from the cell state C_{t-1} by applying a sigmoid layer. Next, an input gate i_t employs the sigmoid function to choose what will be updated. The candidate update to the cell state $\hat{C}_t$ is created through a hyperbolic tangent function layer. The cell state is then updated by combining the output from the forget layer and the selected cell candidate. Lastly, an output gate o_t with the sigmoid function determines what will be output and the hidden state h_t is obtained by modulating the cell state C_{t-1} using the hyperbolic tangent function. These steps can be written in mathematical expressions as

$$f_t = \sigma(W_f x_t + R_f h_{t-1} + B_f), \tag{3}$$

$$i_t = \sigma(W_i x_t + R_i h_{t-1} + B_i), \tag{4}$$

$$\hat{C}_t = \tanh(W_C x_t + R_C h_{t-1} + B_C), \tag{5}$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \hat{C}_t, \tag{6}$$

$$o_t = \sigma(W_o x_t + R_o h_{t-1} + B_o), \tag{7}$$

$$h_t = o_t \circ \tanh C_t, \tag{8}$$

where W, R and B represent the weights for inputs, weights for recurrent states, and the biases respectively, with subscripts indicating the corresponding layer. σ denotes the sigmoid function, and tanh is the hyperbolic tangent activation function. The Hadamard product operation is marked by $\circ$.

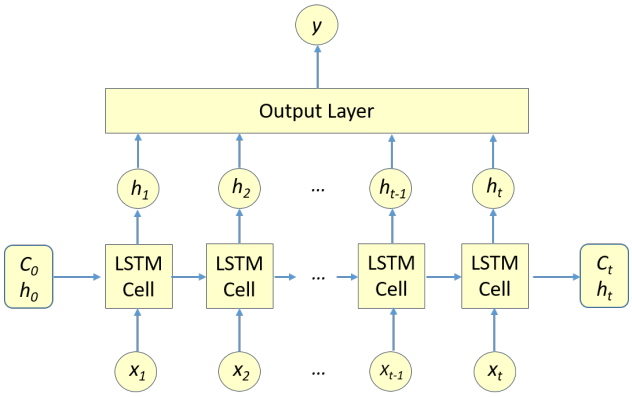


Fig. 2 Structure of an LSTM network with unrolled LSTM cells.

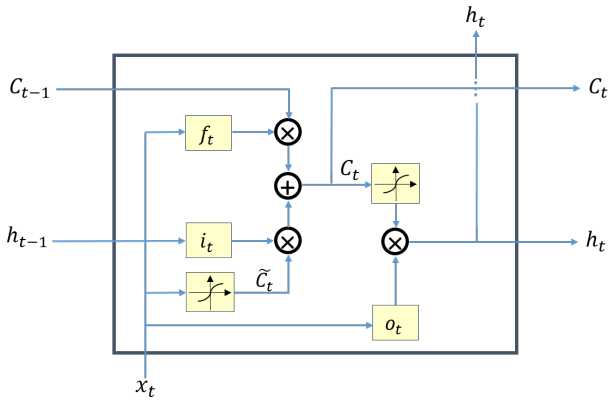


Fig. 3 Structure of an LSTM building block.

The special structure of the memory state and three regulators enables LSTM to convey information over a long-time span. In LSTM, the feedback is from the output of the hidden state, and information travels between adjacent time steps with less resistance. In contrast, the NARX framework addresses the long-term dependency by building external paths to connect to the past directly as seen in Figure 4. The corresponding mathematical model is

$$\mathbf{y}(t) = f\left(\mathbf{u}(t), \mathbf{u}(t-d_{u1}), \ldots, \mathbf{u}(t-d_{um}), \mathbf{y}(t-d_{y1}), \ldots, \mathbf{y}(t-d_{yn})\right), \tag{9}$$

where there are m input delay states represented by d_{u1} through d_{um}, and n feedback delay states denoted by d_{y1} through d_{yn}. The selection of the input and output states imposes significant influence on the ROM predictive performance. Both physics- and statistics-based methods can be applied to the effective states selection. The number of delay states and the span of each delay state are flexible and should reflect time dependency within the sequence data.

C. Cross-validation Error Estimate

Cross-validation is a routinely implemented approach to assessing the generalization ability of unsteady ROMs. The core concept of this method is to divide the available data set into training and testing subsets. In contrast to the fitting error computed as a difference between the true and approximated responses to the training inputs, the performance evaluated on the testing set is indicative of the predictive ability of the trained model. In addition to the training and testing subsets, a separate validation subset could be reserved as a measure of the generalization ability during the offline phase. By monitoring the predictive performance on the validation subset, multiple times of random initialization can be conducted to avoid being trapped in a local minimum, therefore enhancing generalization ability. In the case that the

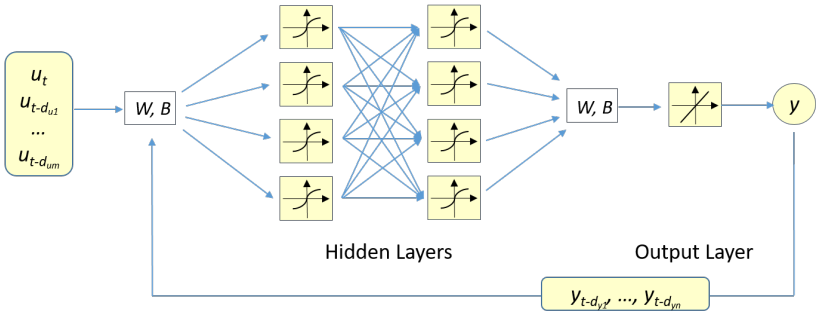


Fig. 4 Architecture of the NARX model used for unsteady computations.

predicted responses are extremely sensitive to initialization or not improved over considerably many initialization trials, it suggests that the structure of the neural network requires adjustment.

The k-fold cross-validation exhibits a balance between the computational cost and sufficiency of representative test samples. In unsteady applications, the k-fold cross-validation evolves into k-time-block cross-validation [24]. Leave-one-out cross-validation is a special case of the k-fold cross-validation. Due to the high computational cost involved, the leave-one-out method is often undesirable. In the present work, the training samples are composed of a series of signals of different types and features. A variety of cross-validation subsets can be derived with respect to the type of signals and time-blocks within an individual signal. Let n denote the number of generic subsets and suppose the root-mean-square-error represents the error estimate. Then, the approximated prediction error using the cross-validation approach is expressed as

$$err_{\mathrm{CV}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}[\hat{y}(\boldsymbol{x}_i,\theta_i) - y(\boldsymbol{x_i})]^2}, \tag{10}$$

where $\hat{y}$ represents the predicted outputs. $\boldsymbol{x}_i$ and θ_i are the inputs and the modeling parameters for the i^{th} tested subset, respectively.

III. Benchmark Problem

A. CFD Model Description

The turbulent simulation of the NLR 7301 airfoil forced by pitching and plunging motions is performed using an in-house CFD code xflow [25]. The incoming flow is at a Reynolds number of 2.1×10^6 and a Mach number between 0.65 and 0.75. The pitching motion excitation varies within a range of 3.7° around a mean angle between 0° and 2.0°, and the plunging motion excitation changes with a variation of 0.15 chord lengths around a mean vertical displacement of 0. In all types of signals, the maximum reduced frequency is limited to 3.0 based on the full chord. As seen in Figure 5, the shock motions in the transonic flow regime are captured by the high-fidelity CFD simulations. The complexity in this phenomenon also poses a challenge in creating the RNN-based ROM.

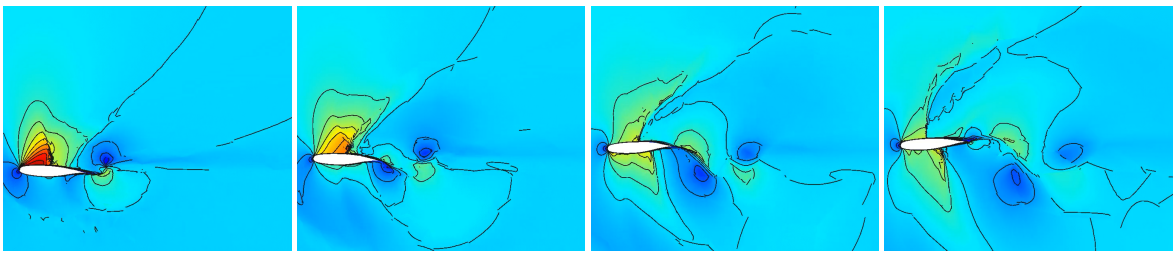


Fig. 5 Snapshots with equal time intervals for high-fidelity CFD simulation of shock motions.

B. RNN Training Signals

For the training data generation, the signal cannot be overly noisy, in order to maintain solution stability. Nevertheless, to build a ROM with good generality, the frequency components and associated amplitudes should span a wide spectrum, with low correlations between data points. The training signals used in this problem encompass 24 single harmonics with constant frequencies and amplitudes at 3 bulk flow conditions, 6 superimposed harmonic signals corresponding to 6 flow conditions with each signal composed of 8 individual single harmonics, 6 Gaussian pulse shaped signals and two sets of random-like signals with each set having 6 signals for 6 distinct flow conditions. To create a compact but representative sample set, the bulk flow conditions and the individual single harmonics used in the superimposed harmonic signal are discretized, permuted and combined via Latin hypercube sampling technique.

The superimposed harmonic signal can be expressed by

$$F_{\mathrm{SIH}}(t) = A_m + \sum_{i=1}^{N} A_i \sin(2\pi f_i t + \phi_i) \sin(2\pi f_m t), \tag{11}$$

where A_m and f_m denote the central amplitude and frequency. N is the number of harmonics in the summation. A_i, f_i, and ϕ_i represent the amplitude, frequency and phase for each harmonic. There are $2N$ dominant frequencies laid out symmetrically about the central frequency f_m. This behavior is explained by showing that the product of two sine functions can be decomposed into the sum of two cosine functions. The phases of the two cosine functions become the sum and difference of the original phases respectively:

$$\sin(2\pi f_i t + \phi_i) \sin(2\pi f_m t) = \frac{\cos(2\pi(f_i - f_m)t + \phi_i) - \cos(2\pi(f_i + f_m)t + \phi_i)}{2}. \tag{12}$$

The signal of the Gaussian pulse shape is generated by the following formula:

$$F_{\mathrm{GP}}(t) = A_m + A\mathrm{e}^{-t_{s1}^2} \cos\left(50 t_{s1} - 8\pi \mathrm{e}^{-2t_{s1}^2}\right), \tag{13}$$

where t_{s1} is the scaled time defined by

$$t_{s1} = \frac{t - s_{1/2} - t_0}{s_{1/2}} l_{\mathrm{eff}}. \tag{14}$$

Since a stable CFD solution is more likely with a signal with gradually increased gradients in time, this type of signal is adjusted to contain most information in the middle of the total time length by using two parameters: the half length of the total simulation time $s_{1/2}$ and the coefficient l_{eff}. It is expected that the generated ROM can identify all the frequency components and retrieve the correct amplitudes in the signal.

The randomlike signal is created by imposing noise to a single harmonic as

$$F_{\mathrm{rand}}(t) = A_m + \frac{(F_{\mathrm{base}}(t) + C_1 F_{\mathrm{noise}}(t))}{1 + C_1}, \tag{15}$$

where C_1 is a parameter used to adjust the amplitude of the composed randomlike signal. F_{base} and F_{noise} represent the single harmonic and noisy portions respectively with the following formula:

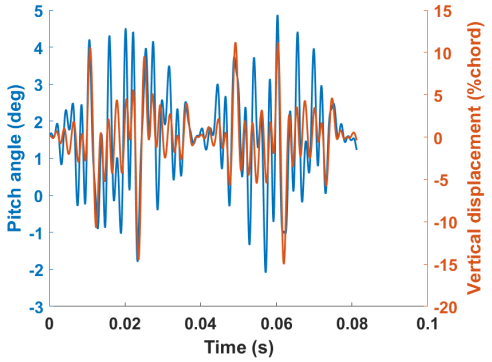
$$F_{\mathrm{base}}(t) = A \sin(2\pi f t), \tag{16}$$

$$F_{\mathrm{noise}}(t) = A \sin(\mathrm{e}^{\sin(t_{s2})} {t_{s2}}^{0.7}) \sin(\sin(t_{s2})) {t_{s2}}^{0.5}, \tag{17}$$

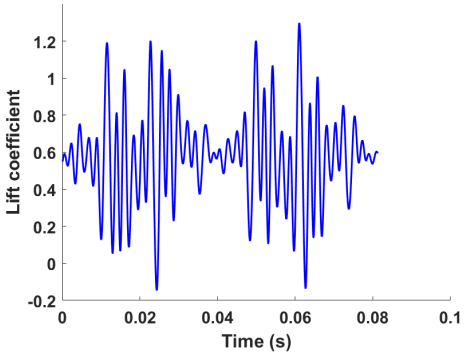
where the scaled time $t_{s2} = \frac{t}{C_2}$. The parameter C_2 can be used to adjust the frequency of the noise.

Both the pitching and plunging excitations are generated using the above formula, but they differ in the parameter values to account for the coupling effects of phases and the combination of distinct frequencies and amplitudes. Figures 6, 7 and 8 show sample signals for the superimposed harmonics, Gaussian pulse, and randomlike types. The lift coefficient responses are also displayed.

Close examination of all the single harmonic training cases reveals that under most test conditions, single harmonic excitations induce lift responses of the same type. In specific cases, especially when the frequencies are low, the lift responses present dominant and secondary frequency components, constituting challenging cases for both training and testing. Therefore, it is essential to incorporate sample signals with specific combinations of input values that lead to characteristic highly-nonlinear behavior to ensure generalization ability of the ROMs.

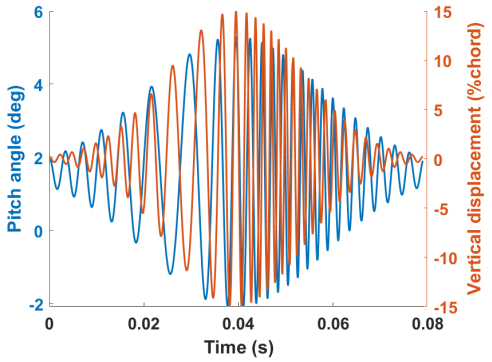


(a) Excitations

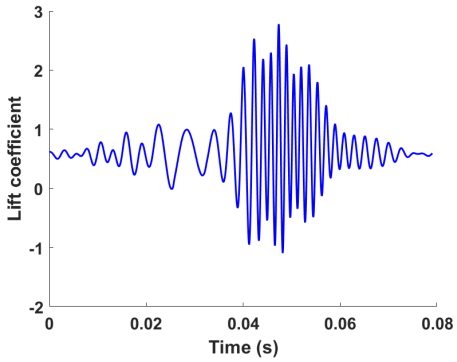


(b) Lift coefficient

Fig. 6 Superimposed harmonics excitations and lift coefficient response.

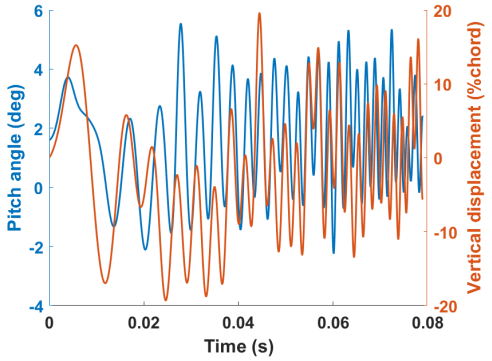


(a) Excitations

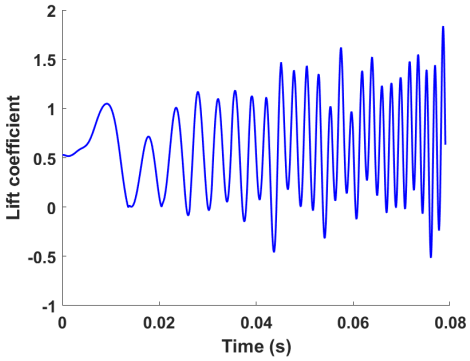


(b) Lift coefficient

Fig. 7 Gaussian pulse type excitations and lift coefficient response.



(a) Excitations



(b) Lift coefficient

Fig. 8 Randomlike excitations and lift coefficient response.

C. ROM Testing Cases

In the testing cases, both the flow boundary conditions and the pitching and plunging motion parameters are perturbed within the trained ranges to avoid extrapolation, which tends to yield poor predictions. These perturbations are designed in such a way that both the individual and coupling effects of the influential parameters can be fully investigated. It is also of interest to evaluate the predictive performance of the trained ROMs on extreme cases. Only the single harmonic

signals are used for testing in order to identify the crucial components that are hard to capture by ROMs. These single harmonic test cases are divided into three categories: pitch-only, plunge-only, and pitch-and-plunge-coupled motions. The detailed definition covers the range of excitation parameters and contains the combination of extreme values. The decoupled cases enable convenient analysis of responses in the frequency domain. The pitching-only cases consist of a series of excitations with gradually increasing pitch angles, and the plunging-only cases are composed of signals with varying vertical displacement. For each described case, 10 reduced frequencies ranging from 0.1 to 3 are distributed with increasing increments. Coupled cases are defined with randomly-mixed motion parameters. In the training set, all the single harmonic excitations contain 4 cycles and 100 time steps for each. Nonetheless, it is noted during test case generation that for some selected excitations, the responses need more than 4 cycles to resolve the initial transient and achieve periodic oscillations. The evident nonlinearity in the lift responses results from the phenomenon of the shocks appearing and disappearing alternately near the leading edge of the airfoil.

IV. Results

Not only does the present work aim to create RNN-based ROMs with good performance, but it also focuses on the investigation of factors that influence the generalization ability in the frame of LSTM and NARX RNNs applied to unsteady aerodynamics. Modeling and computation of both scalar and distribution variables is pursued using different RNN structures. The integral aerodynamic coefficients are obtained by both direct simulations and integration of predicted distributions, i.e. pressure coefficient and viscous stress coefficient, along the airfoil surface.

A. Scalar Prediction

NARX RNNs use the previous delay states in the fashion of external inputs. Thus, the selection of effective delay states is critical in building a relationship between the inputs and outputs with time dependency. For the NARX RNNs created to yield the presented results, 11 input delays have been incorporated with a maximum delay of 245 time steps and increasing time intervals between adjacent delay states. There are 3 hidden layers in the RNN model and each layer contains 10 neurons.

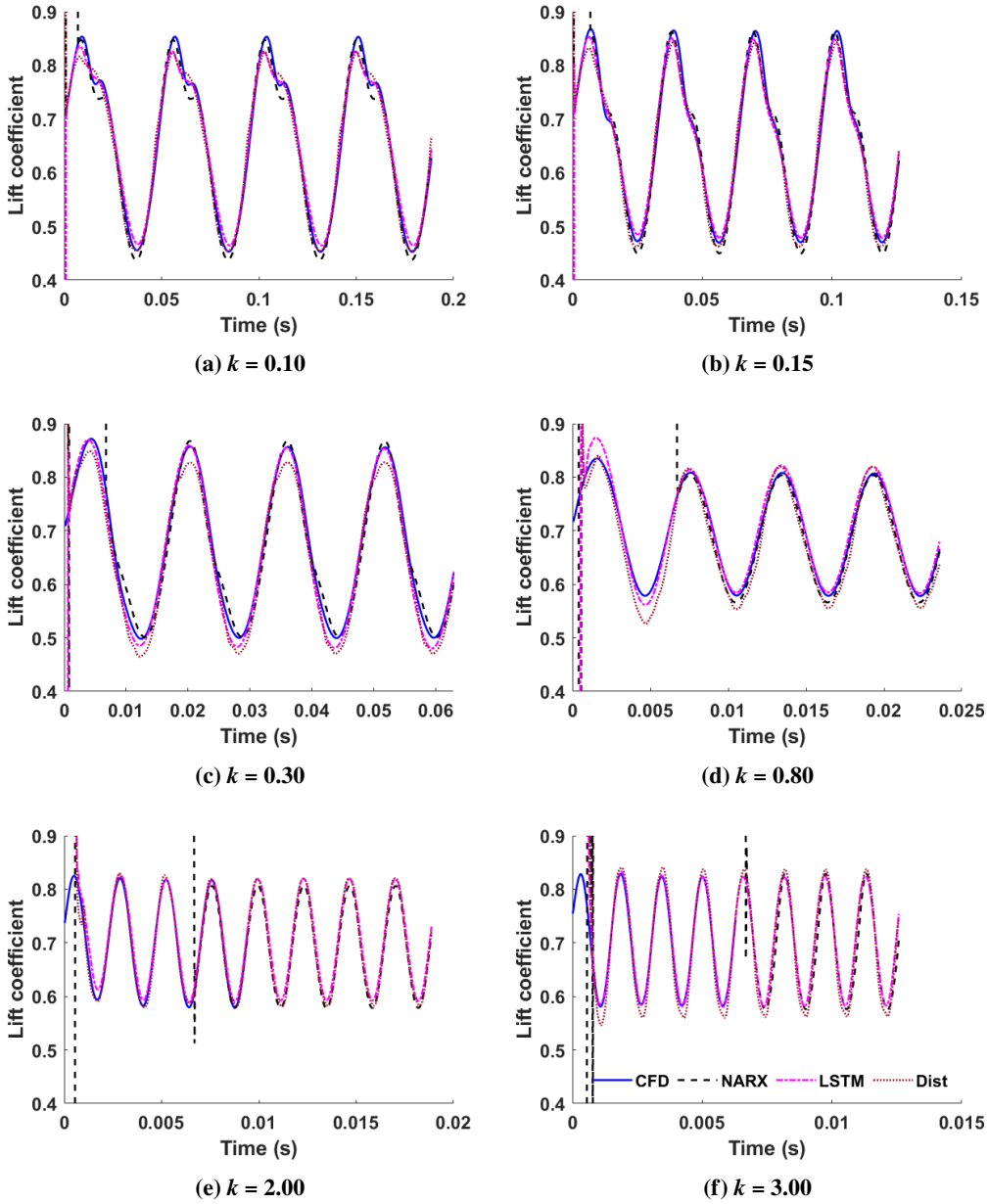
LSTM RNNs are widely applied due to the large capacity of data and validated effectiveness in regulating the information that travels through the hidden units. The number of hidden units reflects the computational capability of ROMs. Therefore, the choice of hidden units plays a crucial role in generalization ability. There should be sufficient hidden units to capture the dominant nonlinear features, but not be excessive to overfit.

The presented RNN models are trained using a Matlab® code. During the training process, the LSTM RNN exhibits excellent capacity for incorporating large amounts of data, while the NARX RNN becomes less efficient as the number of delay states and the size of samples increase. Lift coefficient is the scalar used to demonstrate the implementation of RNN ROMs and to evaluate their performance. To predict lift coefficient responses, an LSTM RNN and a NARX RNN are trained with targets being the lift coefficient itself, leading to two RNN ROMs denoted as direct LSTM ROM and direct NARX ROM, respectively. In addition, different LSTM RNNs are created to simulate the pressure and viscous stress distributions, which are integrated along the airfoil surface to obtain the lift coefficient. With the targets being distributions, these RNN models are named distribution ROMs. Representative results are selected from testing motion categories to show the performance the trained models.

1. Pitch Only

The pitch-only case for performance demonstration is conducted under a Mach number of 0.705 and a mean angle of attack of 1.99°, both of which reflect a small deviation from the sampled flow conditions for training. The pitch angle has a fixed amplitude of 1.54° and 6 reduced frequencies including the minimum and maximum values in the training range. The corresponding lift coefficient responses are displayed in Figure 9. It is apparent that the nonlinear features near the lift coefficient peaks, which are due to long-lasting excitations at low frequencies, have been captured by the direct NARX and LSTM ROMs and distribution ROMs that are marked by "Dist" and created using the LSTM structure.

Comparing the results of individual ROM predictions with the CFD solution reveals that NARX presents an excellent ability to detect nonlinearity at very low frequencies, although it underestimates the secondary frequency component and overestimates the troughs in specific cases. Since the time step size is unified to retain time dependency for all of the cases, the high frequency responses need more cycles to feed NARX RNN enough delay states for prediction. Due to the same requirement and limitation in NARX applications, the response curves exhibit spikes before generating meaningful outputs. As the plots in Figure 9 show, direct LSTM ROM yields the best prediction in most cases. This direct LSTM



(a) k = 0.10 (b) k = 0.15

(c) k = 0.30 (d) k = 0.80

(e) k = 2.00 (f) k = 3.00

Fig. 9 Lift coefficient responses to pitch excitations with Mach = 0.705, α_m = 1.99°, and $\Delta\alpha$ = 1.54°. (a) k = 0.10, (b) k = 0.15, (c) k = 0.30, (d) k = 0.80, (e) k = 2.00, (f) k = 3.00.

model tends to represent initial transient of the studied problem and agrees with the high-fidelity solution as the purely periodic response is achieved. In contrast to the direct LSTM ROM, prediction by distribution LSTM ROMs lowers the mean of lift coefficients at low frequencies and exaggerates the amplitude of lift coefficients at high frequencies.

A conclusion can be drawn that the direct ROMs outperform the distribution ROMs for simulating integrated quantities in terms of smoothness and accuracy. Nonetheless, the distribution ROM is desired to track the complicated shock motions and to provide more details about the solution. In addition, the accuracy can be enhanced further by improving predictions of the pressure and the viscous stress distributions. Therefore, primary analysis and error estimation are demonstrated with the results obtained from the distribution RNN ROMs.

With verified performance, the distribution ROMs are adopted to investigate the effects of reduced frequency in the pitch-only case. Figure 10 collects the lift coefficients predicted at 6 reduced frequencies. As the reduced frequency

increases, the nonlinear phenomenon is mitigated and the mean lift coefficient is increased. The amplitude of the responses also increases except for the highest frequency modeled.

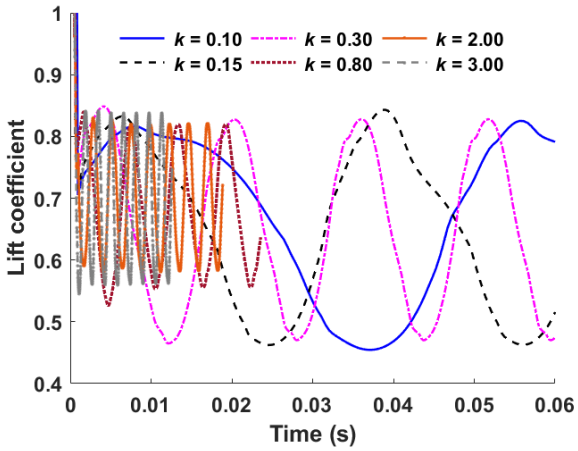
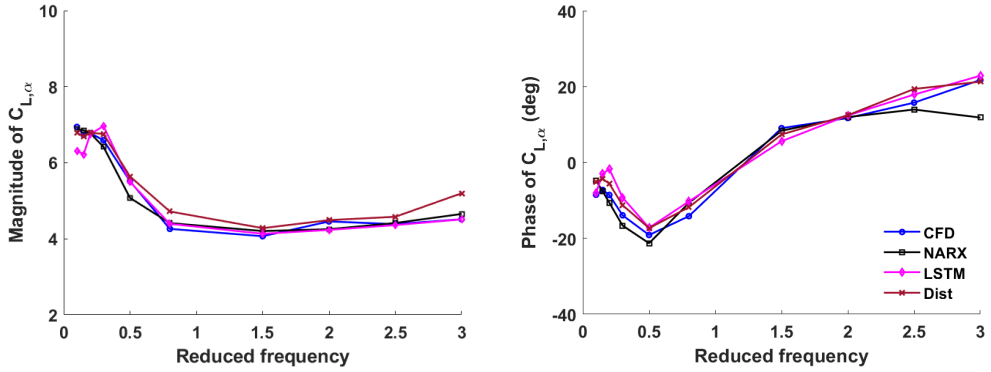


Fig. 10 Lift coefficient responses with varying reduced frequency at Mach = 0.705, α_m = 1.99°, and $\Delta\alpha$ = 1.54°.

To analyze the frequency response with respect to the pitch angle, a frequency transfer function (FRF) [26] of lift coefficients is implemented. Figure 11 provides both the magnitude and the phase of $C_{L,\alpha}$. The magnitude decreases until a reduced frequency of 1.5, and slightly increases at the maximum reduced frequency of 3.0, which is consistent with the observation in the time domain. At the highest reduced frequency, the use of distribution ROMs leads to an overestimated amplitude, while the direct NARX ROM yields a mismatched phase. The degraded performance occurring at extreme conditions implies inefficient sampling and training within certain intervals containing the extrema.



(a) Magnitude **(b) Phase**

Fig. 11 Frequency response function of lift coefficients in the pitch-only case.

Further investigation on the influence of varying pitch angle can be conducted by comparing the time histories of lift responses under specified pitch angle amplitudes. In Figure 12, the bulk flow condition remains the same and the reduced frequency is assigned to be 0.1. The amplitude of the pitch angle varies from 0.1° to 3.14°. Predicted lift coefficients present a shift from the CFD solution as the pitch angle increases. A discrepancy in the envelope appears for the amplitude of 3.14°, suggesting performance degradation in extreme conditions.

2. Plunge Only

Following the analysis for the pitch-only case, the ROM is also evaluated for plunge-only cases. The flow is excited at a Mach number of 0.72 and a mean angle of attack of 0.5° with the maximum vertical displacement o 13.1% of

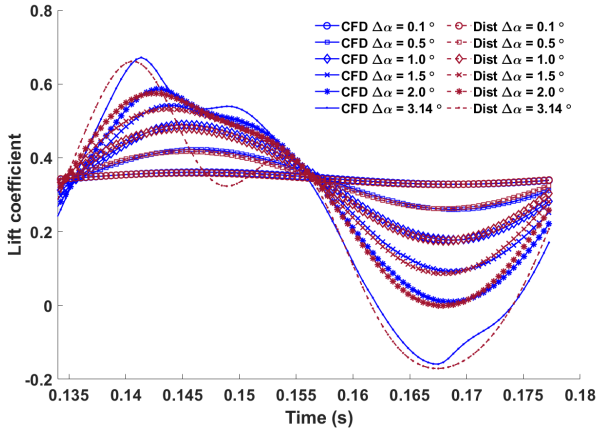


Fig. 12 Lift coefficient responses to varying pitch angle at Mach = 0.75, α_m = 0.05°, and k = 0.1.

the chord length. Figure 13 exhibits the lift coefficient predictions obtained by employing different RNN ROMs. The accuracy of the direct NARX ROM increases as the reduced frequency increases, while the direct LSTM ROM results in slight phase shifts at low frequencies. The distribution ROM prediction aligns with the CFD solution, but leads to shrunken lift coefficients at high frequencies. Focusing on predictive performance of distribution ROMs, the time history plots at representative reduced frequencies in Figure 14 depict the increasing trend of the lift coefficients with increasing reduced frequency, which is reflected by FRF of the lift coefficients in Figure 15. FRF also reveals the phase variation of the lift response with respect to the vertical displacement.

3. Coupled Motion

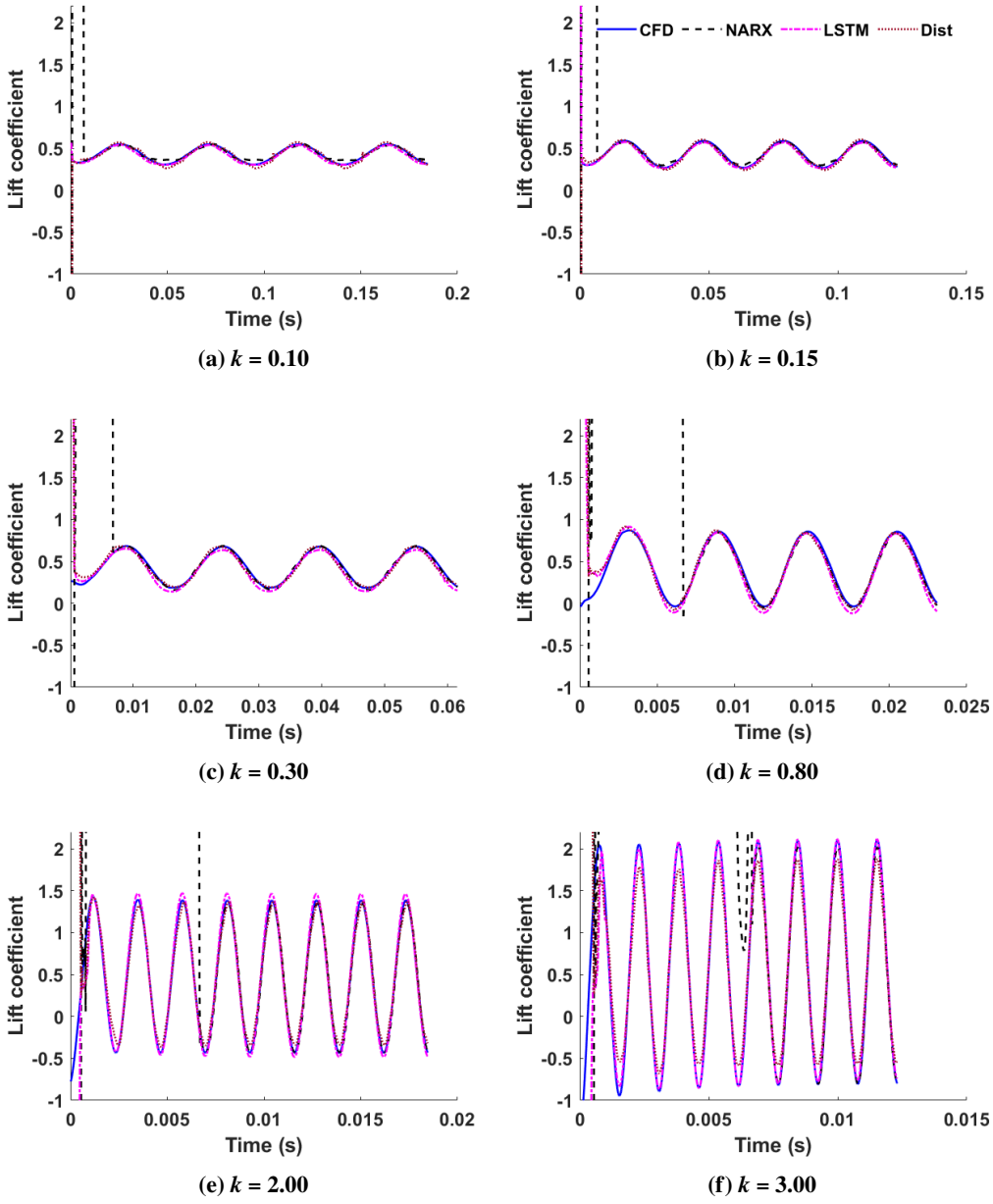
The cases with coupled excitations incorporate the relative phase between pitching and plunging motions. Assigned motion parameters are described in Table 1 as well as the tested bulk flow conditions. The time history plots are displayed in Figure 16. At high reduced frequencies, despite the large amplitudes of motions, the predicted lift coefficient closely follows the CFD solution. It is noted that the initial transient has also been captured by the trained ROMs. The response to the low frequency coupled motion in Case 3 is compared with a plunge-only case with the same flow condition and motion frequency. For this created plunge-only case, the direct NARX ROM is able to extract the phase of the response signal accurately, but underestimates the amplitude. While the direct LSTM ROM and the distribution ROMs yield closer amplitudes, but cause phase shifts. This observed behavior of the ROMs suggests that the sample set needs to be supplemented by additional excitation signals.

Table 1 Mean flow conditions and motion parameters for the coupled cases

Case	Mach	α_m (°)	$\Delta\alpha$ (°)	Δh (% c)	ϕ (°)	k
1	0.66	0.4	3.2	12.8	30	1.5
2	0.725	1.571	1.047	1.571	120	2.749
3	0.75	0.05	3.14	3.14	−42	0.3

B. Distribution Prediction

The results for lift coefficient responses have shown the ability of RNN-based ROMs to simulate multivariate unsteady problems. Specifically, the distribution ROM prediction has been evaluated by comparing the calculated lift coefficient with the high-fidelity solution and the direct ROM simulation. To gain insight into the potential of RNNs in constructing distributed quantities and to improve the accuracy of distribution predictions and derived integrals, it is necessary to explore the related training process and to analyze the predicted distributions in detail. Since the airfoil



(a) k = 0.10 **(b) k = 0.15**

(c) k = 0.30 **(d) k = 0.80**

(e) k = 2.00 **(f) k = 3.00**

Fig. 13 Lift coefficient responses to plunge excitations with Mach = 0.72, α_m = 0.5°, and Δh = 13.1%c. (a) k = 0.10, (b) k = 0.15, (c) k = 0.30, (d) k = 0.80, (e) k = 2.00, (f) k = 3.00.

surface consists of 1530 nodes, training all the nodes over the entire simulation time length for all the samples is tedious and impractical. The distributed outputs in the training sets, which are pressure and viscous stress in this problem, are first decomposed to obtain POD basis vectors. Based on the spectrum of singular values or the individual mode contributions to the integral quantities, critical modes are selected and the coefficients of these modes become the new training outputs. Because of the POD representation, the dimension of the outputs in RNNs is reduced significantly.

For distribution simulations, only LSTM is used, primarily due to the limitation of NARX in efficiency when processing large data sets. The cutoff number of POD modes for pressure is 460 with the cumulative sum of square singular values being 99.99%. Figures 17a and 17b compare the pressure coefficients reconstructed by 20 modes and 460 modes, respectively. For a node at 0.3 of the chord length, the pressure coefficient time history is displayed in Figure 17a. Results with 20 modes can capture the trend in general, but do not align well with the high-fidelity solution at the bottom of each cycle. Pressure history recovered by 460 modes suggests a good agreement with the true solution. Selecting a time point corresponding to the bottom of the oscillation cycle, the reconstructed pressure distribution along the airfoil surface with all modes also outperforms that with only 20 modes in terms of capturing the details of shock motions. The overall root mean square error is reduced from 0.032 to 0.030.

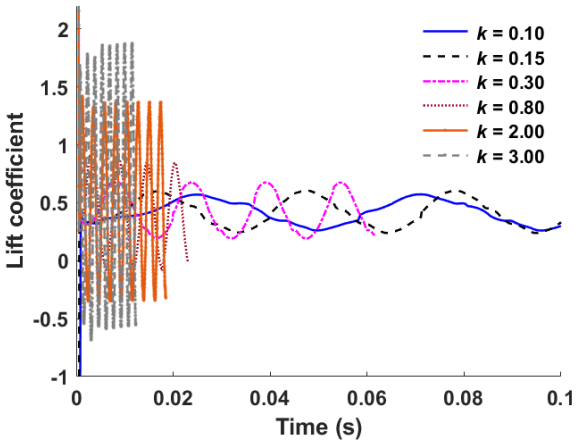
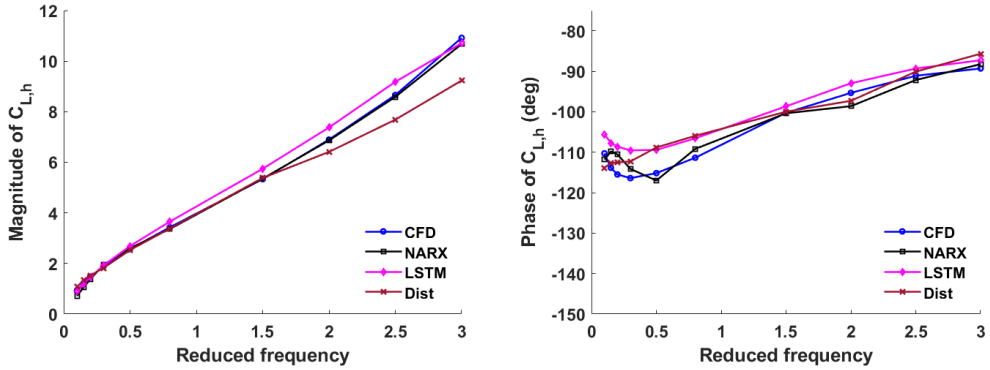


Fig. 14 Lift coefficient responses with varying reduced frequency at Mach = 0.72, α_m = 0.5°, and Δh = 13.1%c.

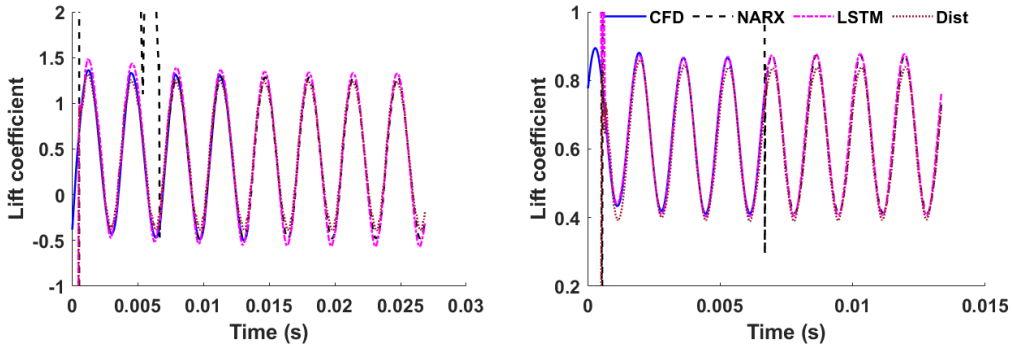


(a) Magnitude **(b) Phase**

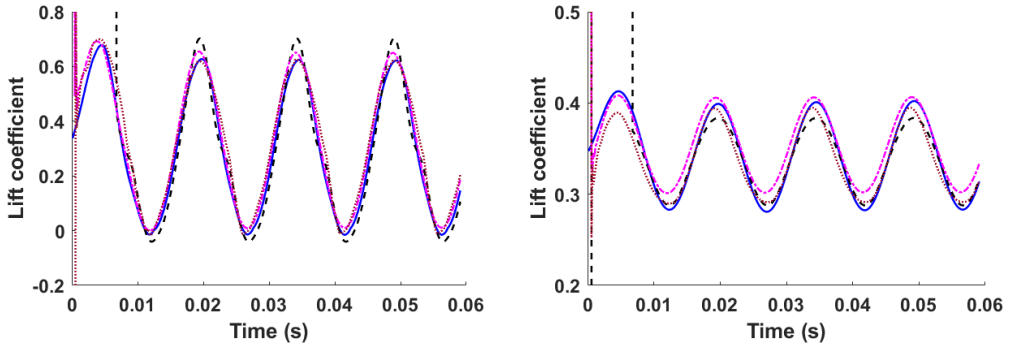
Fig. 15 Frequency response function of lift coefficients in the plunge-only case.

Incorporating more POD modes enhances the ability of ROMs to capture the detailed information of the pressure distribution. To obtain an accurate lift coefficient, the viscous stress is also simulated by separate ROMs for x and y components. Integrating pressure and viscous stress along the airfoil surface yields the lift coefficient time history in Figure 18. The results with different numbers of POD modes are visually indistinguishable, which is consistent with the small difference in the overall root mean square errors of pressure. For all the results presented elsewhere, the maximum number of POD modes determined by the threshold has been adopted for training the distributions in order to account for more details without knowledge of the testing cases during the offline phase.

A single case is selected from each of the three excitation categories to present the predicted distributions in terms of the two orthogonal components of pressure and viscous stress, respectively. It should be noted that pressure is simulated as a whole and then decomposed into x and y components to compute the lift coefficient, while viscous stress is decomposed into two components for training which are simulated separately. Training pressure instead of its two components not only cuts computational time by a half, but also improves the accuracy by avoiding the accumulated error arising from training two ROMs. The estimated training time for pressure ROM is 1.5 hours with single GPU. While it takes nearly 2.5 hours to train viscous stress ROMs, since approximately 750 POD modes are employed for both x and y components. With the predicted POD mode coefficients, distributions have been reconstructed. Figures 19, 20, and 21 show the results at the time corresponding to the peak lift coefficient in the high-fidelity solution. For the pitch-only case in Figure 19, the x component of the pressure coefficient has a good agreement with the CFD solution, but there is a discrepancy at the location where the shock occurs for the y component. The viscous stress is much

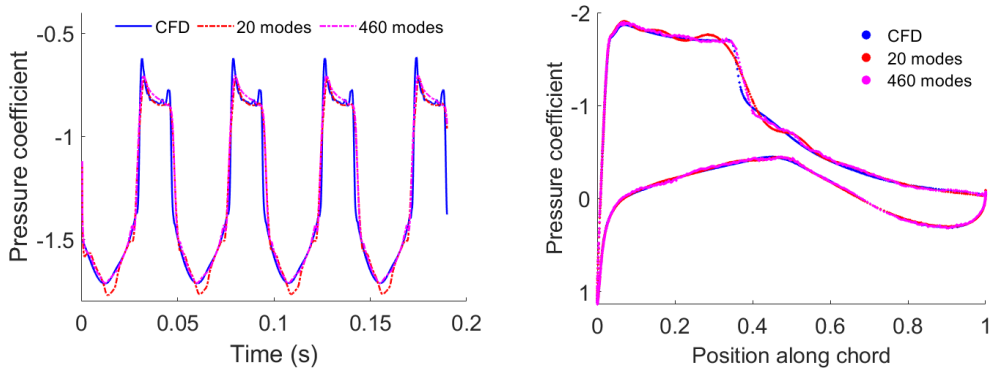


(a) Coupled case 1 (b) Coupled case 2



(c) Coupled case 3 (d) Plunge in Case 3

Fig. 16 Lift coefficient responses to coupled excitations. (a) Coupled case 1, (b) Coupled case 2, (c) Coupled case 3, (d) Plunge in Case 3.



(a) Time history (b) Distribution along airfoil surface

Fig. 17 Reconstructed pressure coefficients at a specified location and at a specific time instant.

smaller in magnitude compared to pressure, and contributes less to the lift coefficient. Nonetheless, accurate viscous stress prediction has a significant effect on drag and moment coefficients, which should also be considered for industrial applications. The two viscous ROMs representing two components are still able to capture the distributions in the presence of large numerical noise. For the plunge-only case displayed in Figure 20, the prediction error is increased for the y component of pressure. The predicted x component of viscous stress preserves the envelope of the CFD result, but misrepresents the details. It is more challenging to predict the coupled case with motion parameters close to the bounds of the training signal definition. The predicted pressure exhibits small deviations from the CFD data and wiggles along

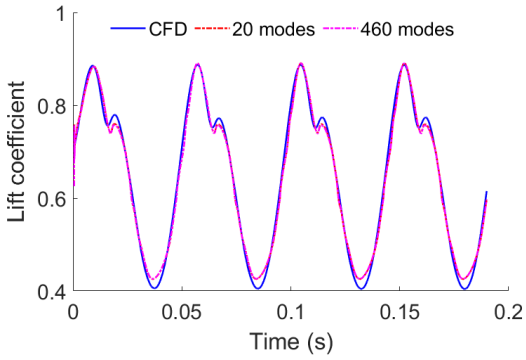
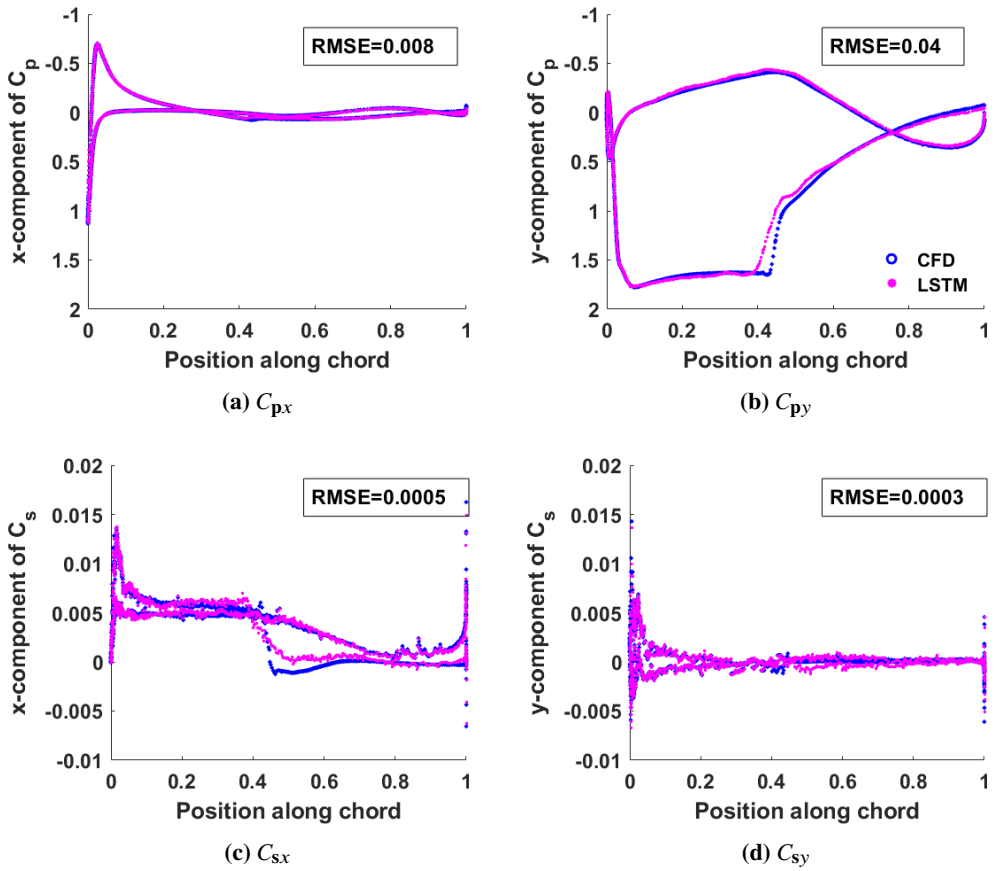


Fig. 18 Effects of the number of POD modes on lift coefficient responses.

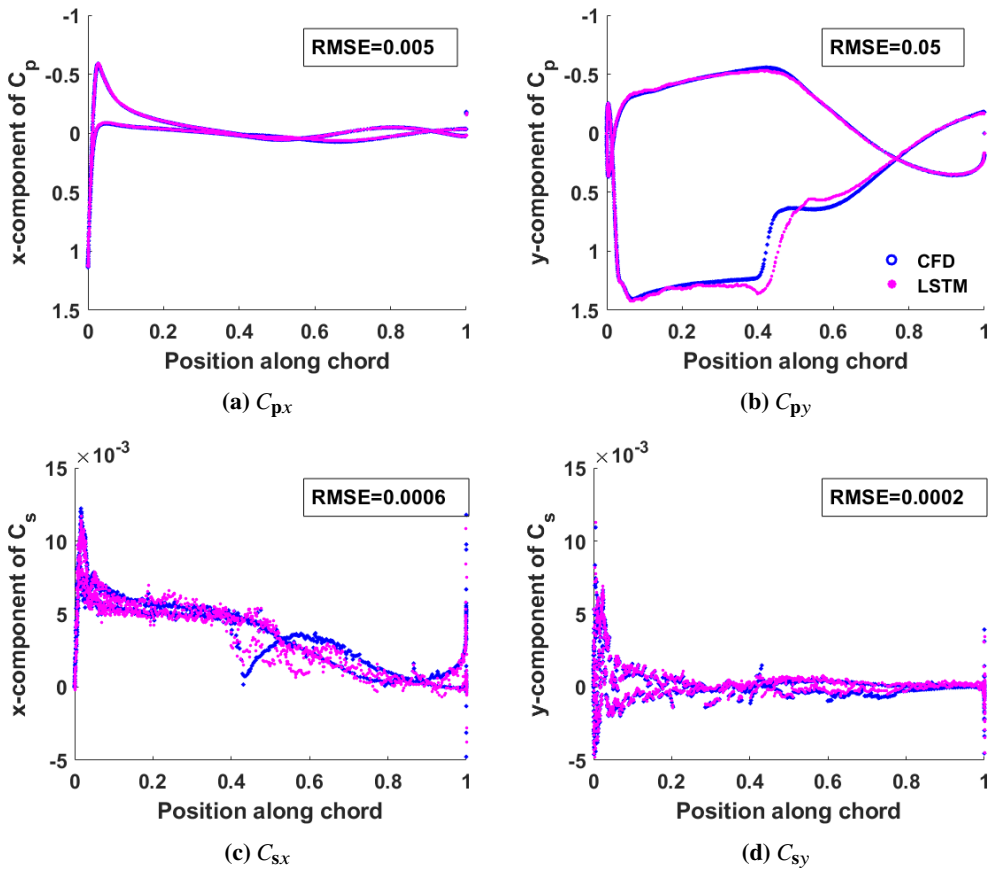
the full surface due to noise. Prediction of viscous stress does not show performance degradation compared to the other two test cases.



(a) C_{px} **(b)** C_{py}

(c) C_{sx} **(d)** C_{sy}

Fig. 19 Predictions of pressure and viscous stress distributions for the pitch-only case at k = 0.1. (a) C_{px}, (b) C_{py}, (c) C_{sx}, (d) C_{sy}.



(a) C_{px} **(b)** C_{py} **(c)** C_{sx} **(d)** C_{sy}

Fig. 20 Predictions of pressure and viscous stress distributions for the plunge-only case at k = 0.1. (a) C_{px}, (b) C_{py}, (c) C_{sx}, (d) C_{sy}.

C. Prediction Error Estimation

Cross-validation has become a standard means of evaluating the generalization ability of the trained models. In the RNN ROMs used throughout this study, all available samples have been included for training due to the compactness of the training set and the representative feature in each sample. Root mean square errors of the lift coefficients after initial transients have been chosen as the metric for evaluating the error estimation technique. Aligned with the fact that testing cases only contain single harmonics, the cross-validation error estimation is performed by only leaving one of the single harmonics in the training set out each time, leading to 24 approximated errors for one trial of the cross-validation process. The errors predicted for individual single harmonic samples are presented in Figure 22. The corresponding pressure and viscous stress ROMs are created using the same modeling settings as the original distribution ROMs trained with all the samples. The influence of random initialization is taken into account by carrying out the cross-validation error estimation multiple times. Two trials are selected to demonstrate the application to lift coefficient prediction, yielding overall estimated root mean square errors of 0.065 and 0.071, respectively. Both trials show high prediction errors for the cases representing a combination of high pitch angles, extreme plunge motions and high reduced frequencies that are used to generate the 6th, 14th and 22nd samples. The effect of randomness in initialization between two trials is reflected on the 2nd sample. A more indicative error estimate should be computed by taking the average of the approximated errors over multiple trials, which takes considerably computational efforts.

The comparison in Figure 23 assesses the accuracy of the estimated prediction errors with the true prediction errors as reference. The errors between the predicted lift coefficients and the high-fidelity lift coefficients are below the bound set by the cross-validation error for the pitch-only case with all tested reduced frequencies. Under the condition with large plunge motions described by the plunge-only case, the distribution ROM prediction leads to large errors at high reduced frequencies, which confirms the phenomenon illustrated by the time history plot in Figure 13 that presents an underestimated amplitude. The coupled case with high-amplitude motions shows an error exceeding the approximated

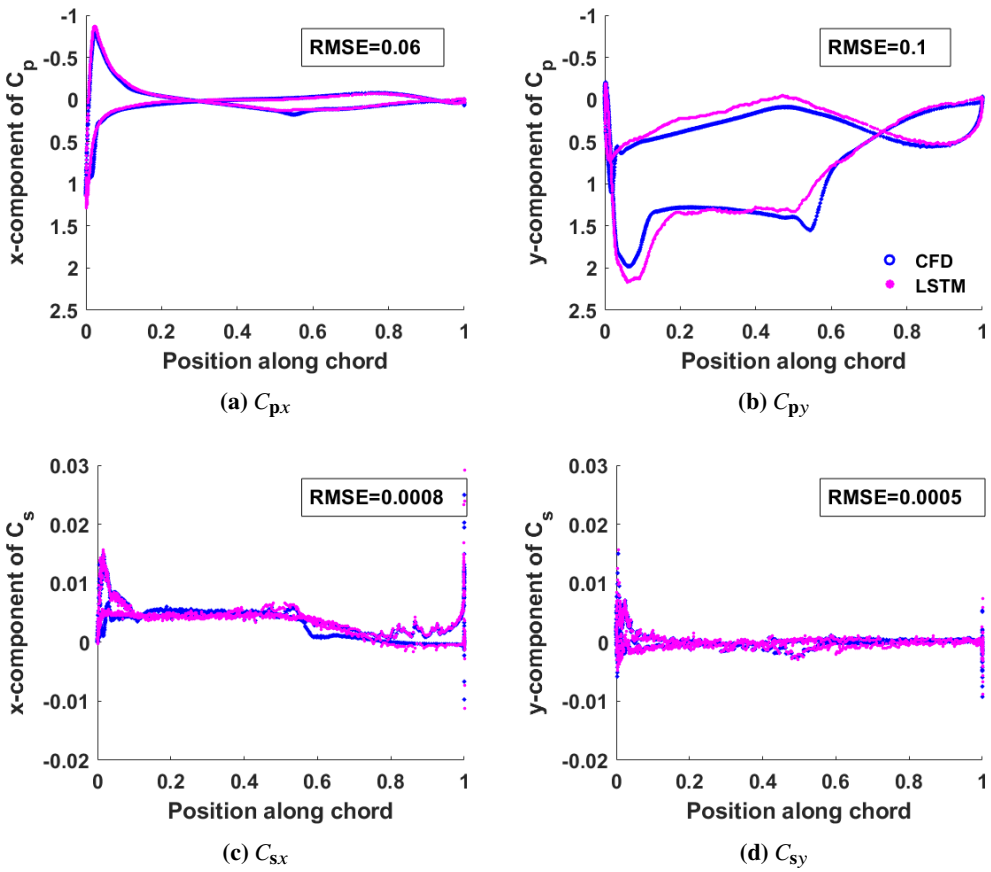


Fig. 21 Predictions of pressure and viscous stress distributions for the coupled motion case. (a) C_{px}, (b) C_{py}, (c) C_{sx}, (d) C_{sy}.

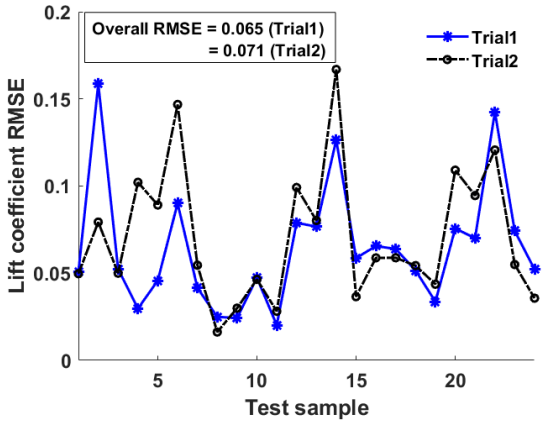


Fig. 22 Prediction errors obtained from cross-validation for individual training samples.

errors. Part of the reason lies in the large values of lift coefficients for this case. Nonetheless, the cross-validation error is an overall estimate that intends to account for all possible test cases. The more important interpretation of this outlier is that the ROM lacks training samples in the neighborhood of the extrema. Except for the extreme cases, the cross-validation error estimate is a reliable approximation of the errors for unknown test cases.

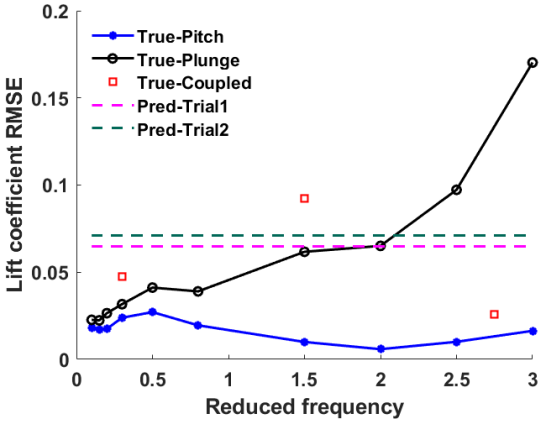


Fig. 23 Comparison of approximated prediction error with true prediction errors.

V. Concluding Remarks

Multivariate unsteady ROMs created by training RNNs with time series inputs achieve good predictions for both scalar and distribution variables of aerodynamic responses to pitching and plunging excitations. POD representation of distributions reduces the computational cost significantly. Reconstruction by the predicted POD coefficients leads to predicted distributions with desirable accuracy, provided that sufficient POD modes are used. Simulation results suggest that including more POD modes is effective in capturing details, but improvement in the integrated quantities is limited. Building an RNN ROM by treating pressure as one variable results in more accurate lift coefficients than creating separate RNN ROMs for two pressure coefficient components.

Predictive performance of the trained ROMs is generally good in simulating the responses to single harmonic excitations. Performance degradation is seen as the bulk flow conditions and motion parameters are varied towards the extreme values defined in the training set. There exists parameter combinations that can induce significant nonlinearity, which should be properly and carefully designed and sampled during the offline stage. Viscous stress distributions have also been simulated to yield better predictions of the lift coefficient responses. Numerical noise makes training difficult in both accuracy and efficiency due to the increased number of POD modes that have to be incorporated. One of the goals to create the present ROMs is to simulate the full time history including the initial transient of the flow. Since the RNN ROMs are built on "memories", for high frequency cases, the initial transient has been overtaken by the impractical outputs due to the lack of delay states. This phenomenon is prominent in NARX predictions.

Fitting accuracy in the training set is a biased measure of the prediction ability. Therefore, cross-validation error estimation should be conducted to evaluate the predictive performance on unknown data. The other approach employed in this paper is to create different ROMs as cross reference. It is expected that the lift coefficients calculated by integration of distributions predicted by distribution ROMs are close to those predicted by direct ROMs. Thus, proximity in lift coefficients to the direct ROMs has been assessed to determine the distribution ROMs that should be selected.

Given the desired accuracy, the created ROMs can play a significant role in real-time applications. For instance, the computational cost in simulating this highly-nonlinear, unsteady aerodynamics problem has been reduced from one hour on a 24-processor machine to a few seconds for the single harmonic test cases.

Future work will include retraining the model with enriched samples. All the test cases shown in this paper are single harmonics. In order to simulate random signals, the coupling between motions and the hysteresis effects of the responses need a thorough representation in the training set. As one way to retain time dependency in the responses, the sequences of the input data have been interpolated to possess the same time step size. Alternatives include converting the time units so that the training signals are discretized with same increments under a different unit system, or incorporating time values into the RNN models. To evaluate the predictive performance more efficiently, analytical error estimation could be considered, as cross-validation becomes exhausting with the increased dimension and complexity of the studied problems.

Acknowledgments

The material in this paper is based on work supported by Airbus in the frame of the Airbus/Michigan Center for Aero-Servo-Elasticity of Very Flexible Aircraft. Special thanks to Hans Bleecke and Reik Thormann for their expert advice and review of this paper.

References

[1] Suresh, S., Omkar, S. N., Mani, V., and Guru Prakash, T., "Lift Coefficient Prediction at High Angle of Attack Using Recurrent Neural Network," *Aerospace Science and Technology*, Vol. 7, No. 8, 2003, pp. 595, 602. doi:10.1016/S1270-9638(03)00053-1.

[2] Zhang, W., Wang, B., Ye, Z., and Quan, J., "Efficient Method for Limit Cycle Flutter Analysis Based on Nonlinear Aerodynamic Reduced-Order Models," *AIAA Journal*, Vol. 50, No. 5, 2012, pp. 1019, 1028. doi:10.2514/1.J050581.

[3] Winter, M., and Breisamter, C., "Reduced-Order Modeling of Unsteady Aerodynamic Loads Using Radial Basis Function Neural Networks," *Deutscher Luft- und Raumfahrtkongress*, 2014. Augsburg, Germany, 2014.

[4] Mannarino, A., and Mantegazza, P., "Nonlinear Aeroelastic Reduced Order Modeling by Recurrent Neural Networks," *Journal of Fluids and Structures*, Vol. 48, 2014, pp. 103, 121. doi:10.1016/j.jfluidstructs.2014.02.016.

[5] Faller, W. E., and Schreck, S. J., "Real-Time Prediction of Unsteady Aerodynamics: Application for Aircraft Control and Manoeuvrability Enhancement," *IEEE Transactions on Neural Networks*, Vol. 6, No. 6, 1995, pp. 1461–1468. doi:10.1109/72.471362.

[6] Ignatyev, D. I., and Khrabrov, A. N., "Neural Network Modeling of Unsteady Aerodynamic Characteristics at High Angles of Attack," *Aerospace Science and Technology*, Vol. 41, 2015, pp. 106 – 115. doi:https://doi.org/10.1016/j.ast.2014.12.017, URL http://www.sciencedirect.com/science/article/pii/S1270963814002703.

[7] Kou, J., and Zhang, W., "Multi-Kernel Neural Networks for Nonlinear Unsteady Aerodynamic Reduced-Order Modeling," *Aerospace Science and Technology*, Vol. 67, 2017, pp. 309 – 326. doi:https://doi.org/10.1016/j.ast.2017.04.017, URL http://www.sciencedirect.com/science/article/pii/S1270963816307891.

[8] Winter, M., and Breitsamter, C., "Nonlinear Identification via Connected Neural Networks for Unsteady Aerodynamic Analysis," *Aerospace Science and Technology*, Vol. 77, 2018, pp. 802 – 818. doi:https://doi.org/10.1016/j.ast.2018.03.034, URL http://www.sciencedirect.com/science/article/pii/S1270963818301810.

[9] Park, K. H., Jun, S. O., Baek, S. M., Cho, M. H., Yee, K. J., and Lee, D. H., "Reduced-Order Model with an Artificial Neural Network for Aerostructural Design Optimization," *Journal of Aircraft*, Vol. 50, No. 4, 2013, pp. 1106–1116. doi:10.2514/1.C032062.

[10] Lindhorst, K., Haupt, M. C., and Horst, P., "Efficient Surrogate Modelling of Nonlinear Aerodynamics in Aerostructural Coupling Schemes," *AIAA Journal*, Vol. 52, No. 9, 2014, pp. 1952–1966. doi:10.2514/1.J052725.

[11] Winter, M., and Breitsamter, C., "Efficient Unsteady Aerodynamic Loads Prediction Based on Nonlinear System Identification and Proper Orthogonal Decomposition," *Journal of Fluids and Structures*, Vol. 67, 2016, pp. 1 – 21. doi:https://doi.org/10.1016/j.jfluidstructs.2016.08.009, URL http://www.sciencedirect.com/science/article/pii/S0889974616301074.

[12] Giles, C. L., and Omlin, C. W., "Pruning Recurrent Neural Networks for Improved Generalization Performance," *Trans. Neur. Netw.*, Vol. 5, No. 5, 1994, pp. 848–851. doi:10.1109/72.317740, URL http://dx.doi.org/10.1109/72.317740.

[13] Hirasawa, K., Kim, S., Hu, J., Murata, J., Han, M., and Jin, C., "Improvement of Generalization Ability for Identifying Dynamical Systems by Using Universal Learning Networks," *Neural networks*, Vol. 14, No. 10, 2001, p. 1389—1404. doi:10.1016/s0893-6080(01)00117-4, URL https://doi.org/10.1016/S0893-6080(01)00117-4.

[14] Caruana, R., Lawrence, S., and Giles, L., "Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping," *Proceedings of the 13th International Conference on Neural Information Processing Systems*, 2000, pp. 381–387. URL http://dl.acm.org/citation.cfm?id=3008751.3008807.

[15] May, P., Zhou, E., and Lee, C. W., "Improved Generalization in Recurrent Neural Networks Using the Tangent Plane Algorithm," *International Journal of Advanced Computer Science and Applications*, Vol. 5, 2014. doi:10.14569/IJACSA.2014.050317.

[16] Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N., "Exploring Generalization in Deep Learning," *CoRR*, Vol. abs/1706.08947, 2017. URL http://arxiv.org/abs/1706.08947.

[17] Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O., "Understanding Deep Learning Requires Rethinking Generalization," *CoRR*, Vol. abs/1611.03530, 2016. URL http://arxiv.org/abs/1611.03530.

[18] Hochreiter, S., and Schmidhuber, J., "Long Short-Term Memory," *Neural Computation*, Vol. 9, No. 8, 1997, pp. 1735–1780. doi:10.1162/neco.1997.9.8.1735.

[19] Lin, T., Horne, B. G., Tino, P., and Giles, C. L., "Learning Long-Term Dependencies in NARX Recurrent Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 7, No. 6, 1996, pp. 1329–1338. doi:10.1109/72.548162.

[20] Bengio, Y., Simard, P., and Frasconi, P., "Learning Long-Term Dependencies with Gradient Descent is Difficult," *IEEE Transactions on Neural Networks*, Vol. 5, No. 2, 1994, pp. 157–166. doi:10.1109/72.279181.

[21] Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J., "Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies," *A Field Guide to Dynamical Recurrent Neural Networks*, edited by S. C. Kremer and J. F. Kolen, IEEE Press, 2001.

[22] DiPietro, R. S., Navab, N., and Hager, G. D., "Revisiting NARX Recurrent Neural Networks for Long-Term Dependencies," *CoRR*, Vol. abs/1702.07805, 2017. URL http://arxiv.org/abs/1702.07805.

[23] Wang, Q., Medeiros, R. R., Cesnik, C. E. S., Fidkowski, K., Brezillon, J., and Bleecke, H. M., "Techniques for Improving Neural Network-based Aerodynamics Reduced-order Models," 2019. doi:10.2514/6.2019-1849, URL https://arc.aiaa.org/doi/abs/10.2514/6.2019-1849.

[24] Bergmeir, C., and Benítez, J. M., "On the Use of Cross-Validation for Time Series Predictor Evaluation," *Information Sciences*, Vol. 191, 2012, pp. 192–213. doi:https://doi.org/10.1016/j.ins.2011.12.028, URL http://www.sciencedirect.com/science/article/pii/S0020025511006773, data Mining for Software Trustworthiness.

[25] Fidkowski, K. J., and Roe, P. L., "An Entropy Adjoint Approach to Mesh Refinement," *SIAM Journal on Scientific Computing*, Vol. 32, No. 3, 2010, pp. 1261–1287. doi:10.1137/090759057.

[26] Thormann, R., and Widhalm, M., "Linear-Frequency-Domain Predictions of Dynamic-Response Data for Viscous Transonic Flows," *AIAA Journal*, Vol. 51, No. 11, 2013, pp. 2540–2557. doi:10.2514/1.J051896, URL https://doi.org/10.2514/1.J051896.