Title: Physics-informed Recurrent Neural Networks for Linear and Nonlinear Flame Dynamics
Authors: Vikas Yadav, Abdulla Ghani

Abstract

This paper demonstrates the ability of recurrent neural networks (RNNs) to predict the linear and the nonlinear response of a premixed laminar flame to incoming velocity perturbations. We develop data-driven models, which require the velocity and heat release rate fluctuations as input data. Both time series are obtained from Direct Numerical Simulations (DNS) of a laminar flame. The length of the signals, and, hence, the cost of the simulation, is comparable to those used in the linear framework of System Identification. A more robust type of RNNs, namely long short term memory (LSTM), is employed to reduce the dependency on large datasets. The LSTM framework is modeled as a time series regression problem and four models are trained with decreasing data set lengths. All purely data-driven models accurately predict the unsteady time series of the heat release rate and, hence, the Flame Transfer Functions (FTFs). We further improve the model accuracy by incorporating a physical constraint, namely the low-frequency limit for perfectly-premixed flames, into the LSTM model. This step reduces the required data length compared to the purely data-driven approach. The proposed model, called PI-LSTM, is able to reproduce the linear and the nonlinear FTFs for amplitudes up to $50\,\%$ of the laminar flame based on one numerical simulation, where the length of the time series is $100\,\mathrm{ms}$.

1 Introduction

Prediction of combustion instabilities (CIs) require an accurate description of the flame dynamics, i.e. heat release rate fluctuations $\dot{Q}'$ Ducruix *et al.* (2003); Schuller *et al.* (2020). A key element for linear CI analysis is the Flame Transfer Function (FTF), which describes the heat release rate fluctuations caused by the incoming velocity fluctuations u':

$$\mathcal{F}(\omega) = \frac{\dot{Q}'(\omega)/\bar{\dot{Q}}}{u'(\omega)/\bar{u}}, \quad \omega \in \mathbb{C}, \tag{1}$$

with ω being the angular frequency.

The FTF can be measured experimentally or computed via numerical simulations. In experiments, a loudspeaker introduces linear, monochromatic, acoustic velocity perturbations into the combustion system, while a camera captures the flame response Lieuwen (2005 - 2005); Æsøy *et al.* (2020). This procedure has to be repeated for the entire range of frequencies of interest. For numerical simulations, the workflow is similar, and, because of the frequency sweep, computationally expensive. This situation is even more critical, when the velocity amplitude is varied, which is -practically speaking- out of reach.

Instead of using monochromatic input signals, Polifke (2014) introduced methods from control engineering, called System Identification (SI), to reduce the number of required numerical simulations to a single computational run: tailor-made, broad-band signals are designed that contain a range of frequencies (typically up to $800\,\mathrm{Hz}$) with uniformly-distributed amplitudes to excite the flame. The required computational cost for the SI approach is affordable as the required physical time is in the order of $0.1\,\mathrm{s}$ for laminar flames Ghani & Polifke (2021) and $0.3\,\mathrm{s}$ for turbulent flow configurations typically encountered in gas turbines, Tay-Wo-Chong *et al.* (2011); Hermeth *et al.* (2013).

Despite the success of the SI approach, one difficulty can not be overcome: the inherent linear framework of SI is not applicable for the nonlinear flame response modeled by the Flame Describing Function (FDF):

$$\mathcal{F}(\omega, |u'|) = \frac{\dot{Q}'(\omega, |u'|)/\bar{\dot{Q}}}{u'(\omega)/\bar{u}}, \quad \omega \in \mathbb{C}. \tag{2}$$

Dowling (1997) proposed this extension of the FTF such that it does not only depend on the frequency, but also on the modulation amplitude $|u'|$. With this weakly nonlinear framework it is possible to identify e.g. limit cycle amplitudes, Noiray *et al.* (2008).

Machine Learning (ML) has shown to overcome the limitations of linear descriptions, in particular for fluid dynamics, Brunton *et al.* (2020). In the context of flame dynamics, Tathawadekar *et al.* (2021)

proposed a simple neural network architecture, which learned the time signals of u' and $\dot{Q}'$, and successfully reproduced FTFs/FDFs of a laminar flame previously computed by Jaensch & Polifke (2017). However, training of the ML model required signal lengths of $1\,\mathrm{s}$, which is out of reach in the context of Direct Numerical Simulations (DNS) or Large Eddy Simulations (LES).

This work presents a novel ML model for flame dynamics and is based on Recurrent Neural Networks (RNNs). We demonstrate that our model (i) successfully reproduces the FTF in the linear regime as well as the FDF in the nonlinear regime for amplitudes $u'/\bar{u}$ up to $50\,\%$; (ii) successfully reproduces higher harmonics, Haeringer *et al.* (2019); (iii) requires similar (or even less) computational time as the SI approach; and (iv) introduce physical constraints into the data-driven framework to reduce the lower data limit required for accurate FTF modeling. Hence, this work bridges the gap to nonlinear flame models obtained from numerical simulations at the cost of one, high-amplitude, broad-band forced computation.

The paper is structured as follows: Section 2 briefly describes the target configuration. Section 3 introduces the ML architecture employed in this work, followed by the description of the data preprocessing (Sec. 4) and the training procedure (Sec. 5). Section 6 assess the quality of the model predictions in both the linear and non-linear regime. Finally, we introduce the physical constraints in the ML framework (Sec. 7).

2 Numerical setup for data generation

We briefly describe the target configuration as it has been widely used in previous numerical studies Jaensch & Polifke (2017); Haeringer *et al.* (2019); Ghani & Polifke (2021); Tathawadekar *et al.* (2021). The experimental setup corresponds to the multi-slit burner investigated by Kornilov *et al.* (2007). Methane and air are premixed at an equivalence ratio of 0.8 with an inlet velocity of $0.4\,\mathrm{m\,s^{-1}}$. The inlet temperature was maintained at $293\,\mathrm{K}$ and the plate, on which the flame is stabilized, is set to a temperature of $373\,\mathrm{K}$. Two-dimensional numerical simulations were performed using the fully compressible solver *AVBP* developed by Cerfacs/IFPen. Chemical kinetics were modeled by a two-step chemical scheme of Franzelli *et al.* (2010). The flame is resolved with 10 grid points and the term DNS used in this work refers to flame-resolved DNS. We run the simulations for one second sampled at an average rate of $\Delta t = 1 \times 10^{-6}\,\mathrm{s}$. The input signal for velocity forcing is the same for all cases shown here and is the same as used in Tathawadekar *et al.* (2021). The global heat release rate $\dot{Q}'$ is obtained by integrating the heat release rate over the numerical domain.

3 Neural networks for flame dynamics modeling

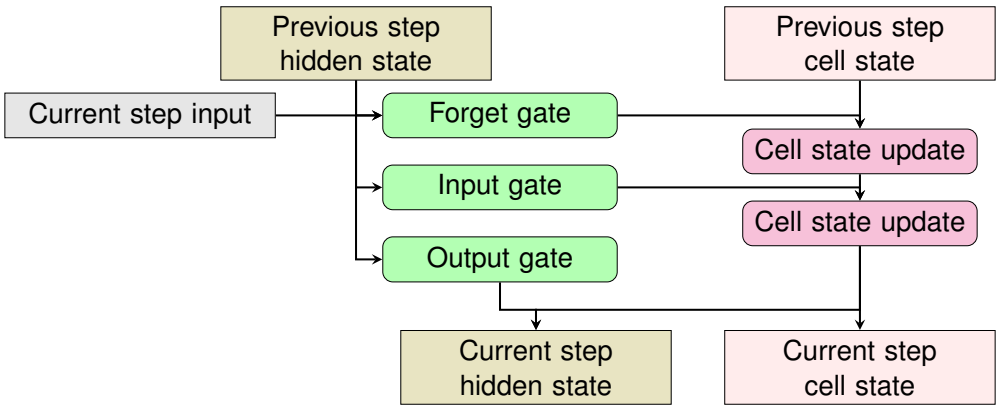
In this section, we recapitulate the evolution of the ML architecture used in this work and briefly explain its specific features.

3.1 Artificial neural network (ANN)

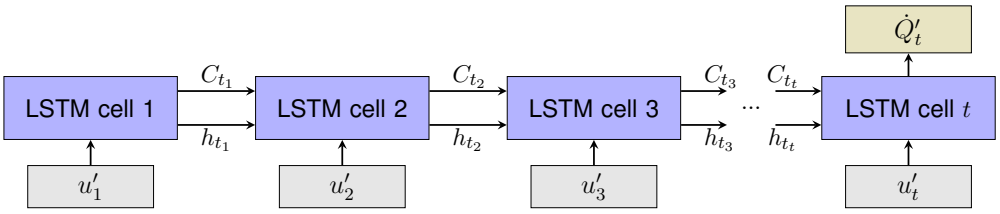
ANNs are composed of layers of basic computational units called neurons. Each neuron receives n inputs (x_i). The neuron linearly combines the inputs and their associated weights W_i together with a bias b. For mapping the computation to a nonlinear space, the linear block is passed through an activation function Φ:

$$y = \Phi\left(\sum_{i=1}^{n} W_i x_i + b\right). \tag{3}$$

The network build from a single neuron is called a perceptron, but it alone cannot capture the nonlinear dynamics of a system. Therefore, neurons are combined in the form of layers called Multi-Layer Perceptrons (MLP). When the layers of neurons are connected in a sequential manner, the network is termed as a Feed-Forward Neural Network (FFNN). The weights and biases associated with the neurons of the networks are the model parameters, which are optimized during the training process.



(a) Components of a single LSTM cell, depicting flow of information through three gates and cell state.



(b) LSTM network mapping t previous input states $(u_1, u_2, ..., u_t)$ to t^{th} output state $(\dot{Q}'_t)$.

Figure 1: Simplified flow chart depicting the flow of information in an LSTM cell.

We use the Mean Squared Error (MSE) as the loss function for the optimization and evaluation of the model:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\dot{Q}'_p(i) - \dot{Q}'(i))^2, \tag{4}$$

where $\dot{Q}'_p$ is the predicted heat release rate and N is the total number of data points. In this work, a regression model is built for the prediction of $\dot{Q}'$. The algorithm stops the optimization process as soon as the tolerance for the minimal error is reached.

The FFNN handles the previous time steps as individual features instead of a dependent sequence, which is an essential aspect of any dynamical system. This problem is overcome by using Recurrent Neural Networks (RNNs) that store information from many previous time steps Zhang & Man (1998).

3.2 Recurrent neural network (RNN)

Unlike FFNNs, RNNs have a feedback loop in the neurons, which is recursively used for the future time steps (called hidden state). In other words: RNNs make partially use of the output of the previous iteration as an input for the next prediction. Hence, it is a well-suited model for sequential data such as DNS time series: a feature that is not given by MLPs. The possibility of passing past information through the neurons, make RNNs the best choice for sequential and time-series data Zhang & Xiao (2000).

However, studies of Hochreiter (1998) and Pascanu *et al.* (2013) point out the shortcomings of RNNs. The latter can not capture the delay in a system because of the vanishing and exploding gradients during the gradient descent optimization. To tackle this problem of exploding and vanishing gradients, a more sophisticated RNN variant, namely *Long short-term memory* network (LSTM), was developed by Hochreiter & Schmidhuber (1997).

3.3 Long short-term memory (LSTM)

LSTM avoids the long-term dependency problem in sequential and time-series data. In contrast to classical RNNs, the hidden state in LSTMs is modified through multiple activation functions referred as to gates: the update, forget and output gates. In addition to the hidden state, a cell state is also present, which accounts for the long-term information. The gates control the information flow of the previous and the current step, e.g. if the input gate value is close to one and the forget gate value is close to zero, the information is removed from the current step. The information from the gates are then passed to the cell state as shown in Fig. 1(a). The cell state carries this information to further time steps. The output gate, on the other hand, combines the long-term information in the cell state together with the current step input to give the new hidden state.

The parameters that control the gates are subject to the optimization procedure (called hyperparameter tuning). The final computation after the output gate and the cell state can be expressed as:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{5}$$

$$h_t = o_t * \tanh(C_t) \tag{6}$$

where

C_t is the updated cell state;

C_{t-1} is the cell state at the previous step;

f_t, i_t and o_t are the information given by the forget, input and output gate, respectively;

h_t is the final output from the LSTM cell.

A more detailed explanation of the LSTM model can be found in Hochreiter & Schmidhuber (1997). In this work, we use u' as the input signal and train the LSTM model for predicting $\dot{Q}'$. Figure 1(b) illustrates one such model having a single layer of LSTM cells using t time steps of u' for predicting $\dot{Q}'$ at the t^{th} step. In the following, we describe the preprocessing steps required to train the LSTM network with time series from the DNS.

4 Data preprocessing

We present the data preprocessing strategy necessary prior to the model training.

4.1 Data splitting and sampling

The LSTM model uses the time series of u' and $\dot{Q}'$ and does not depend on additional information of the combustor. We train four models for a forcing amplitude of $u'/\bar{u} = 0.5$ and reduce the data length continuously. The cases are summarized in Tab. 1.

	Data length			
	1 s	0.5 s	0.25 s	0.125 s
Case	A	B	C	D

Table 1: Summary of the four cases with varying data length for LSTM training. All models are submitted to a forcing amplitude of $u'/\bar{u} = 0.5$.

The data sets of the Cases A, B, C and D are divided into train, validation, and test sets (Fig. 2). 20 % of the selected data is used for model validation and the remaining data from the 1 s data set is used as testing data to evaluate the model performances. The sampling rate of the signals is 1×10^6 Hz, which comes from the time steps of the DNS. Since we are interested in analyzing only frequencies up to <1000 Hz, we decrease the sample rate to be 10 times higher than the frequencies of interest so that there is no deficit of data while building the ML model (10 000 Hz).

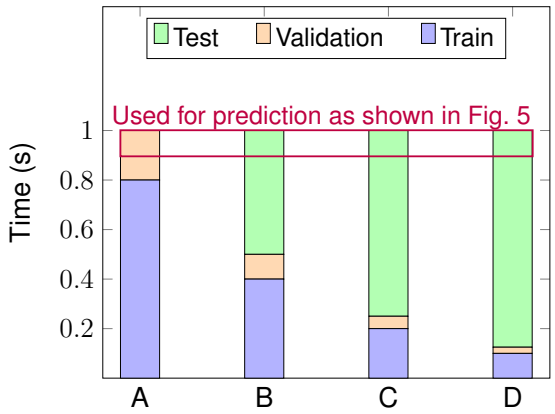


Figure 2: Data splitting for different cases: Case A: $1\,\mathrm{s}$ is used for training with $20\,\%$ of it for validation. Similarly, in Case B: $0.5\,\mathrm{s}$, Case C: $0.25\,\mathrm{s}$ and Case D: $0.125\,\mathrm{s}$ data are used to build the model.

4.2 Determination of the input signal history length

Owing to the time-delayed nature of the thermoacoustic system, we train the model for generating a non-linear map ($\mathcal{G}$) between the history of u' and the current time step $\dot{Q}'$.

$$\dot{Q}'(t) = \mathcal{G}(u'(t)|u'(t-\Delta t), ..., u'(t-n\Delta t)), \quad \forall t. \tag{7}$$

For finding this map, we first need to identify the length of the history of u'.

Tathawadekar *et al.* (2021) set the history data length to $10\,\mathrm{ms}$. Here, we look for a physical approach to determine this value. To identify the characteristic time scale of the system, we perform a DNS simulation by exciting the steady-state flame with a unit impulse and analyze the flame impulse response. Figure 3 shows that the flame reaches, after excitation, the steady state after around $30\,\mathrm{ms}$. Hence, we use the history of $30\,\mathrm{ms}$ of the u' signal to benefit from the long-term dependencies of LSTMs.

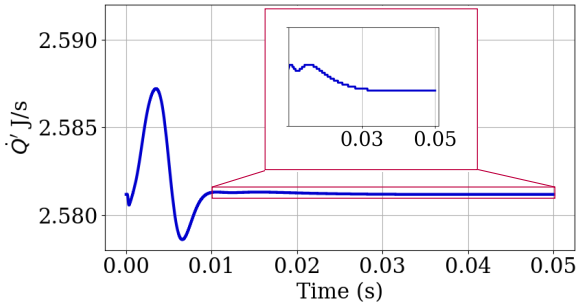


Figure 3: Response of the flame to a unit impulse. The heat release rates reach a steady-state after $30\,\mathrm{ms}$, which is the history data length for LSTMs.

5 Training of the LSTM models

After data preparation, the training of the LSTM framework is performed. The initial weights are randomly chosen and are optimized using the Adaptive momentum estimation (Adam) algorithm, which is the state-of-the-art gradient-based algorithm used for deep learning applications Kingma & Ba (2014). The model is regularized using the dropout strategy: it removes neurons from the layers randomly during the training process. This avoids training a too specialized model for the a specific trained data set, which would otherwise not generalize well on unknown datasets. During training, it is essential to have the dropout technique applied only to the non-recurrent part of the network. Application of dropout to the recurrent part puts constraints on the memory capacity of LSTMs, thereby, limiting their performance.

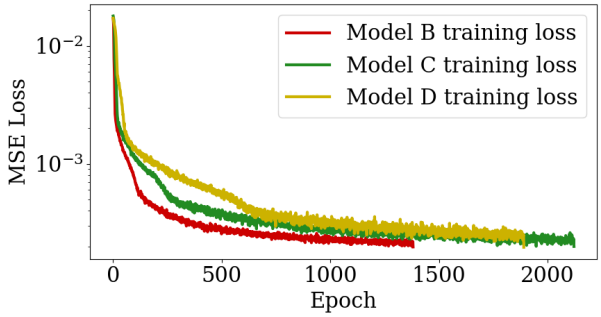


Figure 4: Evolution of the loss function during model training.

The individual steps of the training procedure are:

1. Initially, we start with the reference case of Tathawadekar *et al.* (2021), where they used 70, 35 neurons. In our case, instead of neurons, we used 70, 35 LSTM cells.
2. As discussed in Sec. 4.2, we use $30\,\mathrm{ms}$ of data history of the u' signal as the input size to the architecture.
3. The learning rate is kept small at $1e-5$ to avoid overshooting the minima and the model is trained for more iterations.
4. Dropout can be set to a standard value between 0.2 and 0.5. In this work, we arbitrarily choose the value as 0.3.
5. During training, we set the batch size to a standard value of 32. This means, the training set will be divided in batches of 32 samples and the optimization parameters will be updated after each batch is passed to the algorithm. This speeds up the convergence process.
6. We train the model for around 3000 epochs until the loss function reaches a threshold value. The threshold values is set after observing the convergence of the loss function.

With the initial cell size of 70, 35, the loss function reaches a plateau at 4×10^{-4} and the algorithm stops learning. For improving the accuracy, we increased the cell size. We found that the cell size of 100, 50 was capable of reducing the accuracy to 1×10^{-4} or even lower for larger data sets. Increasing the cell size greater than 100, 50 would increases the training time and does not improve the model performance (not shown). At the same time, training the model for longer epochs after reaching the plateau would not improve the accuracy. In this regard, we define a custom early stopping criteria such that the training is stopped as soon as the loss value reaches the order of 1×10^{-4}. Figure 4 shows the evolution of the training loss until the model is optimized. Owing to larger training data length, model B converges faster than model C and D. After the training process terminates having reached the tolerance, the models are saved and they are later evaluated against the testing set. The set of hyperparameters for the models are summarized in Tab. 2.

Table 2: Hyperparameters of the models for Cases A, B, C and D.

Hyperparameter	Models A,B,C,D
`#Hidden layers`	2
`#LSTM cells`	$100, 50$
`Learning rate`	$1e-5$
`Dropout`	0.3
`Batch size`	32

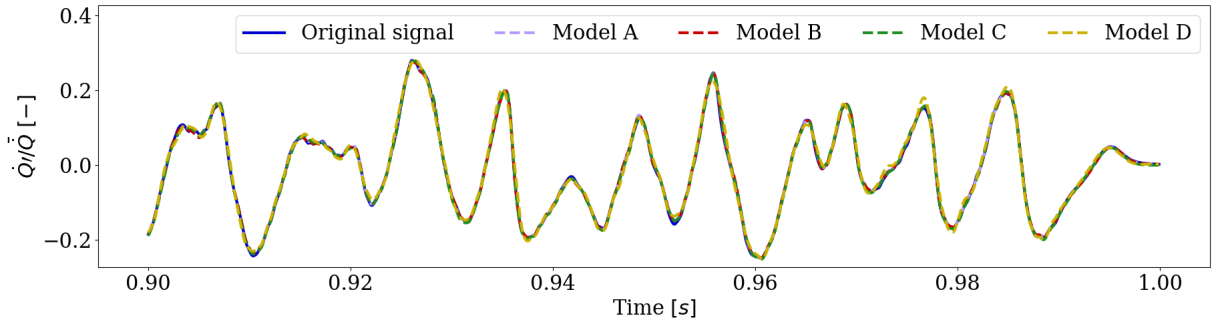


Figure 5: Comparison of the time series of the predicted and computed values by the DNS of the global heat release rate on the testing set (red box in Fig 2).

6 Performance of LSTM for time series and FTF prediction

In this section, we assess the model performance by analyzing its capacity to perform time series to time instances unseen in the training procedure. Next, we present the predicted FTFs in the linear and non-linear regime and show that higher harmonics are well-captured by the models.

6.1 Time series prediction

In order to check how well the models generalize on unseen data, we use the testing data in the range from 0.9-1.0 s (red box in Fig 2). The heat release rate predictions using the four models are visualized in Fig. 5. All the models perform well in predicting the $\dot{Q}'$ signal. The MSE calculated on the entire dataset for models A, B, C and D are 2.99×10^{-5}, 2.5×10^{-5}, 5.0×10^{-5} and 2.16×10^{-4} respectively. Model D, which uses the least amount of training data, generalizes considerably too, with a few small fluctuations (e.g. 0.978 s). This shows that the models capture the nonlinear flame dynamics reasonably well with 125 ms of simulation data.

6.2 Flame Transfer Function prediction

The models are further assessed by estimating the FTF of each model in the linear and the non-linear regime. The models are submitted to mono-frequent forcing with various amplitudes. Sinusoidal signals containing frequencies from 10 Hz to 500 Hz are considered with amplitudes $u'/\bar{u}$ of 0.1 (linear regime, also used in the experiments) and 0.5 (non-linear regime).

6.2.1 Linear regime

First, we check the model responses to mono-chromatic excitations in the linear regime. All four models perform very close to the experimental values of Kornilov *et al.* (2009) (Fig. 6). The phase prediction overlaps the original values for all four models except for higher frequencies, where the gain is almost negligible. Although model D has the least amount of training data (and hence less computational time of the DNS), its performance in estimating the FTF is in very good agreement with that of the models A, B and C.

6.2.2 Non-linear regime

The model performances are now assessed by forcing with high-amplitude sinusoidal waves at $u'/\bar{u} = 0.5$. Figure 7 shows the predicted FTFs of models B, C and D as compared to the simulation results of Jaensch & Polifke (2017) and Model A having the entire data length of 1 s. All models reproduce well the the gain and the phase. Model D shows some minor deviations in the peak gain. For all cases, the low-frequency limit (Polifke & Lawn (2007)) is well-reproduced.

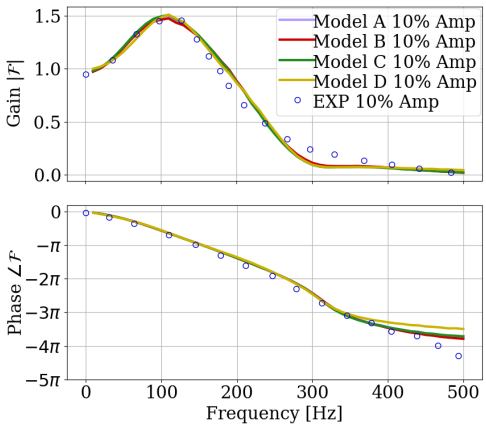


Figure 6: FTFs for $10\,\%$ forcing amplitude.

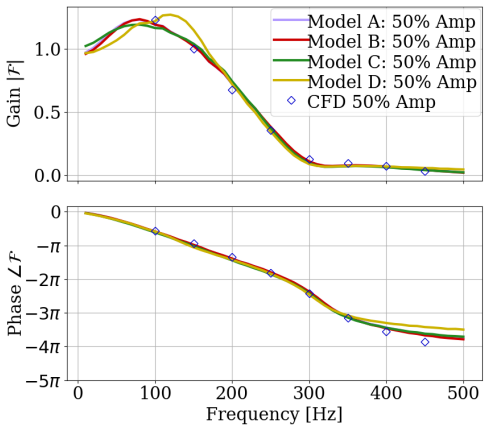


Figure 7: FTF for $50\,\%$ forcing amplitude.

6.3 Frequency content

The trained models are further assessed by analyzing the frequency content of the predicted responses. For this, we consider, for instance, the model predictions of the sinusoidal wave at $100\,\mathrm{Hz}$ and investigate the Fast Fourier Transform (FFT) content in the heat release rate for two amplitudes (Fig. 8). By training the LSTM with broadband signals, the models learn to predict the higher harmonics behavior. When the excitation amplitude is small (Fig. 8), the response of the model is close to pure sinusoidal with negligible contribution from higher harmonics implying the linear nature of the system. But as the excitation amplitude increases to $u'/\bar{u} = 0.5$, the system becomes nonlinear and higher harmonics start to appear as shown in Fig. 8. This behaviour is also accurately captured by our models.

7 Physics-informed gain correction in the minimal data limit

In order to test the modeling limit of LSTMs, we reduce the training data length to $100\,\mathrm{ms}$, which we label as Case E. For smaller data sets, the FTFs are not physical with significant error losses (not shown) and therefore not considered. As the data length reduces, the low-frequency content in the data reduces and, hence, the LSTM framework is not able to reproduce the low-frequency response of the flame accurately (Fig. 9). This is also supported by the estimated low MSE values of 8×10^{-4} for Case E. To overcome this problem, we introduce a physical constraint formulated by Polifke (2014):

$$\lim_{\omega \to 0} \mathcal{F}(\omega) = 1. \tag{8}$$

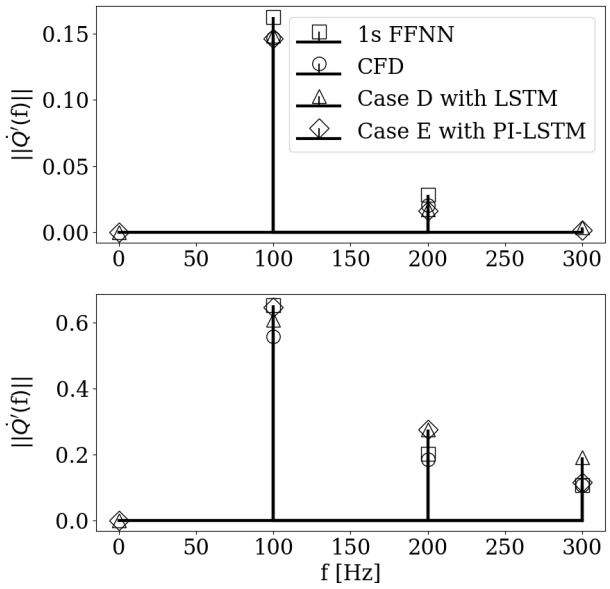


Figure 8: Frequency content of the heat release rate predicted by the model C and PINN model at frequency = 100 Hz at $u'/\bar{u} = 0.1$ (top panel) and 0.5 (bottom panel).

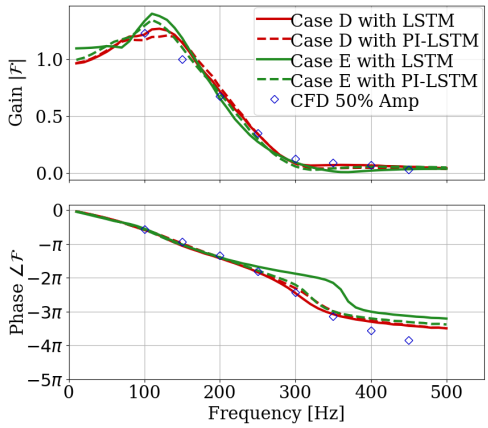


Figure 9: FTF for Case D and E with $50\,\%$ forcing amplitude.

This is known as the low-frequency limit for perfectly-premixed flames and refers to a gain of 1 and a phase of 0. To incorporate this limit during training, the model is given same values of u' and we expect the model to predict the exact same value for $\dot{Q}'$. Any deviation is penalized and this is added to the cost function as follows:

$$\mathcal{J} = \alpha_1 \frac{1}{N} \sum_{i=1}^{N} (\dot{Q}'_p(i) - \dot{Q}'(i))^2_{data} + \alpha_2((\dot{Q}'_p - \dot{Q}')^2)_{physics} \tag{9}$$

In this work, we have equally weighted the MSE induced by the data and the physics in Eq. 9 ($\alpha_1 = \alpha_2 = 0.5$). The physical constraints added in the network improves the MSE on the dataset from 8×10^{-4} to 4×10^{-4} for Case E. Correspondingly, the FTF prediction improved for the case E (Fig. 9). This shows that our Physics-informed LSTM (PI-LSTM) improves the FTF prediction in the minimal data limit. At the same time, the higher harmonics are recovered (Fig. 8). However, PI-LSTMs did not improve the FTF models trained with less than $100\,\mathrm{ms}$. Note that for the same configuration, the lower data limit using linear SI methods is around $70\,\mathrm{ms}$, Ghani & Polifke (2021).

8 Conclusion

We demonstrate the ability of LSTMs to predict the linear and the nonlinear FTF of a laminar premixed flame by performing one single DNS at high-amplitude forcing. Based on the computed time series of the velocity and heat release rate fluctuations, four models are trained with several data set lengths ($500\,\mathrm{ms}$, $250\,\mathrm{ms}$ and $125\,\mathrm{ms}$). All models performed reasonably well compared to the FTFs computed via DNS. In an attempt to further reduce the required input data, we found that the minimal data limit is $100\,\mathrm{ms}$). For this case, the predictions of the FTF started to deviate in the low-frequency limit. Hence, we incorporated this piece of physical knowledge into the LSTM architecture called the Physics-informed LSTM (PI-LSTM). The PI-LSTM improved the model accuracy, but could not lower the minimal data limit required to FTF modeling.

Currently, the PI-LSTM model uses equally weighted scaling parameters, which are not further analyzed in this work. Future work will define the scaling parameters as design parameters, which will be subject to optimization during the training process. We also plan to test the PI-LSTM model on a turbulent flow configuration, which will challenge the ML architecture due to noisy data sets.

Declaration of Interests. The authors report no conflict of interest.

References

ÆSØY, EIRIK, AGUILAR, JOSÉ G, WISEMAN, SAMUEL, BOTHIEN, MIRKO R, WORTH, NICHOLAS A & DAWSON, JAMES R 2020 Scaling and prediction of transfer functions in lean premixed h2/ch4-flames. *Combust. Flame* **215**, 269–282.

BRUNTON, STEVEN L., NOACK, BERND R. & KOUMOUTSAKOS, PETROS 2020 Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **52** (1), 477–508.

DOWLING , A. 1997 Nonlinear self-excited oscillations of a ducted flame. *J. Fluid Mech.* **346**, 271 – 290.

DUCRUIX, SÉBASTIEN, SCHULLER, THIERRY, DUROX, DANIEL & CANDEL, SÉBASTIEN M. 2003 Combustion Dynamics and Instabilities: Elementary Coupling and Driving Mechanisms. *J. Propuls. Power* **19** (5), 722–734.

FRANZELLI, B., RIBER, E., SANJOSÉ, M. & POINSOT, T. 2010 A two-step chemical scheme for kerosene–air premixed flames. *Combust. Flame* **157** (7), 1364–1373.

GHANI, ABDULLA & POLIFKE, WOLFGANG 2021 Control of intrinsic thermoacoustic instabilities using hydrogen fuel. *Proc. Combust. Inst.* **38** (4), 6077–6084.

HAERINGER, MATTHIAS, MERK, MALTE & POLIFKE, WOLFGANG 2019 Inclusion of higher harmonics in the flame describing function for predicting limit cycles of self-excited combustion instabilities. *Proc. Combust. Inst.* **37** (4), 5255–5262.

HERMETH, SEBASTIAN, STAFFELBACH, GABRIEL, GICQUEL, LAURENT YM & POINSOT, THIERRY 2013 Les evaluation of the effects of equivalence ratio fluctuations on the dynamic flame response in a real gas turbine combustion chamber. *Proc. Combust. Inst.* **34** (2), 3165–3173.

HOCHREITER, SEPP 1998 The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertain. Fuzziness Knowlege-Based Syst.* **6** (2), 107–116.

HOCHREITER, SEPP & SCHMIDHUBER, JÜRGEN 1997 Long short-term memory. *Neural Comput.* **9** (8), 1735–1780.

JAENSCH, STEFAN & POLIFKE, WOLFGANG 2017 Uncertainty encountered when modelling self-excited thermoacoustic oscillations with artificial neural networks. *International Journal of Spray and Combustion Dynamics* **9** (4), 367–379.

KINGMA, DIEDERIK & BA, JIMMY 2014 Adam: A method for stochastic optimization. *ICLR* .

KORNILOV, V.N., ROOK, R., TEN THIJE BOONKKAMP, J.H.M. & DE GOEY, L.P.H. 2009 Experimental and numerical investigation of the acoustic response of multi-slit bunsen burners. *Combust. Flame* **156** (10), 1957–1970.

KORNILOV, V.N., SCHREEL, K.R.A.M. & DE GOEY, L.P.H. 2007 Experimental assessment of the acoustic response of laminar premixed bunsen flames. *Proc. Combust. Inst.* **31** (1), 1239–1246.

LIEUWEN, TIMOTHY 2005 - 2005 *Combustion instabilities in gas turbine engines : operational experience, fundamental mechanisms and modeling*. AIAA J.

NOIRAY, N., DUROX, D., SCHULLER, T. & CANDEL, S. 2008 A unified framework for nonlinear combustion instability analysis based on the flame describing function. *J. Fluid Mech.* **615**, 139–167.

PASCANU, RAZVAN, MIKOLOV, TOMAS & BENGIO, YOSHUA 2013 On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning* (ed. Sanjoy Dasgupta & David McAllester), *PMLR*, vol. 28, pp. 1310–1318. PMLR.

POLIFKE, WOLFGANG 2014 Black-box system identification for reduced order model construction. *Ann. Nucl. Energy* **67**, 109–128, advanced stability analysis for nuclear reactors.

POLIFKE, WOLFGANG & LAWN, CHRIS 2007 On the low-frequency limit of flame transfer functions. *Combust. Flame* **151** (3), 437–451.

SCHULLER, THIERRY, POINSOT, THIERRY & CANDEL, SÉBASTIEN 2020 Dynamics and control of premixed combustion systems based on flame transfer and describing functions. *J. Fluid Mech.* **894**.

TATHAWADEKAR, NILAM, DOAN, NGUYEN ANH KHOA, SILVA, CAMILO F. & THUEREY, NILS 2021 Modeling of the nonlinear flame response of a bunsen-type flame via multi-layer perceptron. *Proc. Combust. Inst.* **38** (4), 6261–6269.

TAY-WO-CHONG, LUIS, BOMBERG, SEBASTIAN, ULHAQ, AHTSHAM, KOMAREK, THOMAS & POLIFKE, WOLFGANG 2011 Comparative validation study on identification of premixed flame transfer function. , vol. 134.

ZHANG, JUN & MAN, K.F. 1998 Time series prediction using rnn in multi-dimension embedding phase space. , vol. 2, pp. 1868 – 1873.

ZHANG, JIA-SHU & XIAO, XIAN-CI 2000 Predicting chaotic time series using recurrent neural network. *Chinese Phys. Lett.* **17** (2), 88–90.