Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 59-5 (2025) 85–90

Data-Driven Prediction of Anode Relative Humidity and Voltage in an Open-Cathode PEM Fuel Cell Stack Using the Koopman Operator

Adwoa S. Adunyah, Carrie M. Hall. *

**Illinois Institute of Technology, Chicago, IL 60616, USA (e-mail: chall9@iit.edu)*

Abstract: Control of anode relative humidity is crucial for optimal hydration and performance in proton exchange membrane (PEM) fuel cells. Accurate predictive models can be essential to enabling real-time control of humidity. This study applies the Koopman operator with radial basis function (RBF) and time-delay embeddings as observables in an extended dynamic mode decomposition (EDMD) framework, to predict anode relative humidity and stack voltage in a 5kW open-cathode PEM fuel cell stack. The performance of the Koopman approaches is compared to a NARX neural network. The Koopman model with time delay embeddings as the basis function consistently outperformed that with RBF across all prediction horizons investigated. Particularly at a 5-step prediction horizon, the RBF EDMD recorded RMSE of 0.61% and 1.17V for humidity and voltage respectively while the time-delay EDMD demonstrated exceptional performance, achieving RMSE on the order of 10^{-13} for both outputs. The NARX model with random data division showed competitive results (RMSE of 0.77% and 1.05V) but deteriorated in performance (RMSE of 2.97% and 2.01V) when tested on unseen data. These findings highlight the effectiveness of the Koopman operator, especially with time-delay embeddings as coordinate basis. Additionally, the linearity of the Koopman model enables easy integration with established linear control strategies.

Keywords: PEM fuel cell, Koopman operator, data-driven modeling, NARX neural network

1. INTRODUCTION

Proton exchange membrane (PEM) fuel cells represent a promising clean energy technology due to their high efficiency and reduced environmental impact compared to fossil fuels. Achieving optimal performance in PEM fuel cells relies on effective water management, as membrane hydration is crucial to their operation. Experimental studies have shown that membrane humidity has a more significant impact on performance than stack temperature (Morner & Klein, 2001). (Zeng et al., 2019) found through experiments that high airflow rates led to a decrease in cell voltage due to excessive membrane dehydration. Given that membrane hydration is influenced by the relative humidity at both the anode and cathode, precise control of anode humidity is key to ensuring optimal fuel cell performance.

Control strategies often rely on physics-based models for predicting relative humidity in PEM fuel cells. (Pukrushpan et al. 2004) developed a model to predict relative humidity in the anode and cathode based on conservation of mass. Building on this approach, (Headley, 2013) proposed a dynamic subdivided model to analyze variation in relative humidity. However, the nonlinear nature of the PEM fuel cell stack makes the development of physics-based models challenging. Recently, machine learning approaches have gained traction, as they can simplify model development while maintaining predictive accuracy. Artificial neural networks (ANN) and support vector machines (SVM) have been used to predict membrane hydration, and ANN was shown to perform better than SVM (Legala et al., 2022). Similarly, (H. Huo et al., 2023) developed an ANN model to predict membrane water content, demonstrating superior performance over SVM. Despite their effectiveness, traditional ANNs are static models that lack memory, limiting their ability to capture dynamic behaviors. Dynamic neural networks, such as the nonlinear autoregressive with exogenous inputs (NARX) network, overcome this limitation by incorporating feedback mechanisms to handle temporal dependencies in the data. (Da Costa Lopes et al., 2015) used a NARX network to accurately predict voltage in a control-oriented PEM fuel cell model. Cho et al. (2022) applied a NARX network in a model predictive control framework to optimize membrane hydration. However, the nonlinear nature of these networks often necessitates the use of complex control algorithms or linearization techniques to integrate with linear control theory.

Linear representations are therefore preferable due to the availability of well-established linear control theory. A promising approach to achieve this is the Koopman operator, introduced by (Koopman & Neumann, 1932), which enables linear representation of nonlinear dynamics in an infinite-dimensional space of observables. Finite-dimensional approximation of the Koopman operator can be achieved with dynamic mode decomposition (DMD). Standard DMD, however, relies on linear state measurements, limiting its ability to capture complex nonlinearities. Extended DMD addresses this by using nonlinear observables or basis functions, to lift the states to a higher-dimensional space, where the dynamics can be more accurately approximated as linear (Korda & Mezić, 2016). The success of EDMD depends on the choice of observables; poorly selected observables may

10.1016/j.ifacol.2025.07.086

miss essential dynamics, while excessive observables risk overfitting. Additionally, observable selection is often system-specific and requires tailoring to each application. While the optimal set of observables is not known a priori and is usually determined through the use of trial basis functions, radial basis functions (RBF) are commonly used. In fuel cell modeling, (D. Huo et al., 2023) applied the Koopman operator to model an open-cathode PEM fuel cell stack, employing RBF as observables, but did not predict water activity in this study.

Time-delay embedding, which involves incorporating past state information into dynamical system models, has emerged as an effective coordinate basis for finite approximation of the Koopman operator (Pan & Duraisamy, 2020). This approach is supported by the Takens embedding theorem (Takens, 1980). By augmenting a system's states with delayed versions, each dimension reflects a different time delay, effectively capturing the system's temporal dynamics. This concept is integral to NARX models where the delayed targets and inputs enhance predictive accuracy.

The primary contribution of this paper is the application of the Koopman operator for predicting the anode relative humidity in an open-cathode PEM fuel cell stack, with a specific focus on comparing the performance of RBF and time-delay embeddings as basis functions for finite approximation of the Koopman operator. In this study the effectiveness of these basis functions in predicting anode relative humidity and voltage in a 5kW open-cathode PEM fuel cell stack is evaluated. The performance of the Koopman models is then compared to that of a NARX neural network.

2. EXPERIMENTAL CONFIGURATION

2.1 Experimental Setup

The models are based on the experimental setup shown in Figure 1. The fuel cell stack is a 5kW open-cathode stack from Horizon, featuring 120 cells in series, each with an active area of 150 cm². Air is supplied by four blowers operating within a pulse width modulation (PWM) range of 20% to 90%. Hydrogen of 99.99% purity is sourced from a pressurized tank. Each cell has four anode channels and 81 cathode channels. The stack includes a manufacturer-built controller that monitors current, voltage, and temperature. A mass flow controller from Aalborg measures hydrogen inlet pressure and flow rate, while a Vaisala humidity sensor, installed via a gas sampling cell, measures the dew point temperature at the anode outlet.

2.2 Experimental Test Procedure

To generate training and testing data, experimental tests were conducted at varying current loads across three fan PWM settings—20%, 55%, and 90%—referred to as dataset 1, dataset 2, and dataset 3, respectively. As shown in Figure 2, the maximum allowable current load decreases with reduced PWM to prevent overheating of the PEM fuel cell stack. Datasets 1 and 2 provided transient data for training the Koopman model, while dataset 3's transient data were used for testing. The PEM fuel cell stack first underwent a 30-minute warm-up to ensure adequate humidification before the test. During this time, the load was initially set to 0A, while the Aalborg mass flow controller recorded hydrogen gas flow rate, temperature, and pressure. Simultaneously, the in-built controller monitored the current, voltage, and stack temperature, and a humidity sensor at the anode outlet measured the dew point temperature. The dew point data, combined with stack temperature, was used to calculate the relative humidity. Data were logged for approximately 5 minutes at each load for datasets 1 and 2, while dataset 3 involved an extended 10-minute logging period to ensure adequate load coverage for time-delay embedding analysis. Data from all instruments were synchronized in LabVIEW at a consistent 0.5-second sampling rate.

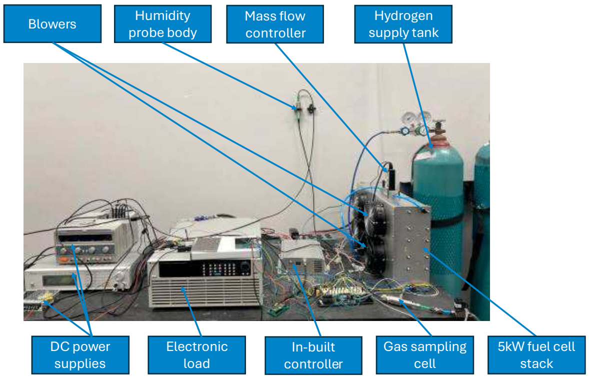

Figure 1: Experimental setup

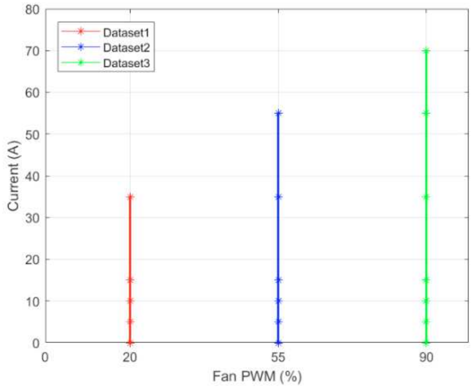

Figure 2: Load test points

3. MODEL DEVELOPMENT

3.1 Koopman Model

Consider a discrete-time nonlinear controlled dynamical system

$$x_{k+1} = f(x_k, u_k) \tag{1}$$

where k is the current step, $x \in R^n$ represents the state of the system with n dimensions, $u \in R^m$ is the control input with

m dimensions, and f is the transion mapping. The Koopman operator, $\mathcal{K}:\mathcal{F}\rightarrow\mathcal{F}$ is a linear operator that acts on nonlinear observable functions $\emptyset: R^n \rightarrow \mathrm{R}^{\mathrm{N}}$, where $\mathcal{F}$ is a space of observables invariant under the action of the Koopman operator. The operator can be expressed by

$$(\mathcal{K}\emptyset)(x) = \emptyset(f(x)) \tag{2}$$

where $\emptyset(f(x))$ is the basis function of the future state obtained by applying the transition mapping f to the current state.

In this system, the states and inputs are defined as

$$x = [RH_{an}, V_{st}, T_{st,} P_{H_2,}] \tag{3}$$

$$u = [PWM, I_{st}]$$

where RH_{an} is the anode relative humidity, V_{st} is the stack voltage, T_{st} is the stack temperature and $P_{H_2,}$ is the hydrogen inlet pressure. The inputs are the fan's PWM and the stack current. I_{st}.

The predictors being sought are of the form

$$z_{k+1} = Az_k + Bu_k \tag{4}$$

$$y = Cz_k$$

where $z \in R^N$ with (typically) N>>n, and y is the prediction of z, $B \in R^{Nxm}$ and $C \in R^{nxN}$. The state z is known as the lifted state since it evolves on a higher dimensional lifted space. The lifted state z is an augmented state of the original state with its transformed coordinates and as such the lifted state z can be described as

$$z = \left[x_1, x_2, x_3, x_{4,} \emptyset_5(x) \dots, \emptyset_N(x)\right]^T \tag{5}$$

3.1.1 RBF as Basis Function

Radial basis functions (RBF) are a common choice of observables for finite approximation of the Koopman operator. The thin plate spline (TPS) RBF was selected as the preferred RBF due to its simplicity, as it avoids the need for additional scaling parameters required by other RBF types. To implement TPS RBF, the radial distance, r is computed between the data and randomly generated centers. The TPS RBF can be calculated by

$$\emptyset_{TPS}(r) = r^2 ln(r) \tag{6}$$

Four RBF feature dimensions (5, 10, 25, and 50) were investigated to determine the optimal parameters for the Koopman model. These values were chosen to provide a structured comparison across a range of model complexities, while keeping computational cost manageable. The dimension with the lowest average RMSE was selected as the optimal model. Considering a trade-off between prediction accuracy and computational cost, an RBF dimension of 25 was selected as optimal. An RBF dimension of 100 was also tested but was discarded, as it led to overfitting.

3.1.2 Time-Delay Embedding as Basis Function

Delay embedding is the process of lifting a time series signal x(t) into a higher dimensional space by stacking it with time-shifted copies of itself x(t-τ) (Dylewsky et al., 2020). To implement Koopman with delay embedding the states of the system are arranged into snapshot matrices as shown in (7) and (8).

$$X = [x_1, x_2, \dots, x_{h-1}] \tag{7}$$

$$X' = [x_2, x_3, \dots, x_h] \tag{8}$$

The subscripts represent the time step. For instance, x_h is the state at time step h. New augmented data matrices are created with the form:

$$Z = \begin{bmatrix} \vert & \vert & \vert & \vert \\ x_1 & x_2 & \dots & x_{\mathrm{h}-d} \\ \vert & \vert & \vert & \vert \\ x_2 & x_3 & \dots & x_{\mathrm{h}-d+1} \\ \vert & \vert & \vdots & \vert \\ x_d & x_{d+1} & \dots & x_{\mathrm{h}-1} \end{bmatrix} \tag{9}$$

$$Z' = \begin{bmatrix} \vert & \vert & \vert & \vert \\ x_2 & x_3 & \dots & x_{\mathrm{h}-d+1} \\ \vert & \vert & \vert & \vert \\ x_3 & x_4 & \dots & x_{\mathrm{h}-d+2} \\ \vert & \vert & \vdots & \vert \\ x_{d+1} & x_{d+2} & \dots & x_h \end{bmatrix} \tag{10}$$

The input matrix though not augmented, is time-shifted according to

$$\Upsilon_{\mathrm{shifted}} = [u_{d+1}, u_2, \dots, u_{h-1}] \tag{11}$$

in which d is the number of delay embeddings.

Two hyperparameters are key in time-delay embeddings: the number of delay embeddings d and the embedding period T^d. For uniformly sampled data, the embedding period is given by $T^d = d\Delta t$, where Δt is the sampling interval. In this study, Δt is set to 0.5s to align with sampling rate of the in-built controller of the PEM fuel cell stack. Four different delay embeddings (5, 10, 15, 20) were tested, selected to explore how varying embedding length impacts prediction accuracy. It was found that 10 delays representing a 5-second embedding period provided the best balance of accuracy and computational efficiency.

3.1.3 State Space Model Determination

Once the lifted states are obtained, the DMD with control (DMDc) algorithm can then be applied to determine the system matrices. For system identification with DMDc, the following steps are required (Proctor et al., 2014).

First, the lifted state variables are collected into a snapshot pair of matrices

$$Z = [z_1, z_2, \dots, z_{h-1}] \tag{12}$$

$$Z' = [z_2, z_3, \dots, z_h] \tag{13}$$

where Z represents the lifted data for the current time step and Z' represents the time-shifted lifted states. The inputs to the system are similarly arranged into a single matrix given by

$$\Upsilon = [u_1, u_2, \dots, u_{h-1}] \tag{14}$$

For time-delay embedding, $\Upsilon_{shifted}$ as described in Equation (11) is used. An augmented matrix is then built as below

$$Z' = [A\ B]\begin{bmatrix} Z \\ \Upsilon \end{bmatrix} = G\Omega \tag{15}$$

A truncated singular value decomposition (SVD) is then performed on the input space Ω with truncation p set to match the number of rows in Ω for this study.

$$\Omega \approx \tilde{U}\tilde{\Sigma}\,\tilde{V}^* \tag{16}$$

G can then be approximated as

$$G \approx Z'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}^* \tag{17}$$

Similarly, A and B can be approximated as

$$A \approx Z'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}_1{}^*$$

$$B \approx Z'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}_2{}^* \tag{18}$$

where $\tilde{U}_1 \in R^{Nxp}$, $\tilde{U}_2 \in R^{lxp}$ and $\tilde{U} = \left[\tilde{U}_1\ \tilde{U}_2\right]^T$

The matrix C can be computed by (Williams et al., 2015)

$$C = XZ^{\dagger} \tag{19}$$

where X is the data for the current time step for the original states as described in Equation (7) and † denotes the pseudoinverse. In the case of time-delay embeddings, the matrix C is computed by

$$C = X_{shifted}Z^{\dagger} \tag{20}$$

where $X_{shifted}$ is the time-shifted version of X according to the number of delay embeddings. $X_{shifted}$ is structured as

$$X_{shifted} = [x_1, x_2, \dots, x_{h-d}] \tag{21}$$

3.2 NARX Neural Network Model

A NARX model is a recurrent dynamic network that relates current value of a time series to past values of the target and input. The defining equation of NARX is given by

$$y_t = f(y_{t-1}, y_{t-2}, \dots, y_{t-p}, x_{t-1}, x_{t-2}, \dots, x_{t-q}) \tag{22}$$

where y is the target, x is the input, subscripts denote timesteps, and p and q are the respective delays applied to the target and input. The target values y can either be the true output values or the network's predicted outputs. Feeding true values back into the network creates a series-parallel (open-loop) configuration, while using predicted outputs define a parallel (closed-loop) configuration.

The NARX model was developed using MATLAB's Deep Learning toolbox. The model utilizes current (I_{st}), PWM, stack temperature (T_{st}), hydrogen inlet mass flow rate ($\dot{m}_{H_2}$), and hydrogen pressure (P_{H_2}) as input, with anode relative humidity (RH_{an}) and stack voltage (V_{st}) as target outputs. The choice of these inputs was informed by the system's physical behavior. Training employed the Levenberg-Marquardt algorithm with both random and indexed division approaches to assess robustness. For random division, the entire dataset was split randomly into 70%, 15% and 15% for training, validation and testing tests respectively. For indexed division, datasets 1 and 2 were dedicated to training and validation, with dataset 3 reserved for testing to ensure no overlap between training and test sets, like the Koopman approach. The model architectures explored ranged from 2 to 16 neurons (2^n, n =1 to 4) and delays 5,10 and 25 on the inputs and targets. Each architecture was trained 10 times to account for variability due to random weight initialization, and the model with the lowest mean squared error (MSE) on the validation set was selected as the optimal model. Early stopping in the toolbox helped prevent overfitting by halting training when validation loss plateaued. The optimal architecture for the indexed division utilized 4 neurons with 5 delays, while the random division performed best with 8 neurons and 5 delays.

4. RESULTS AND DISCUSSION

4.1 Koopman Test Results

The optimal RBF EDMD and Time-Delay EDMD models were evaluated on dataset 3 across four prediction horizons, and the average RMSE are summarized in Table 1. The prediction horizon determines how much the model relies on its own predictions for forecasting. Shorter horizons are more dependent on true data, minimizing error, while longer horizons may lead to error accumulation due to increased reliance on the model's own predictions. The time-delay EDMD consistently outperformed RBF EDMD in predictive accuracy across all prediction horizons. Specifically for a 5-step horizon, the RBF EDMD recorded an RMSE of 0.61% and 1.17V for relative humidity and stack voltage respectively. Remarkably, the time-delay EDMD with a 5s embedding period RMSE on the order of 10^{-13} for both outputs. Figures 3 and 4 illustrate the models' predictions over the 5-step prediction horizon for anode relative humidity and stack voltage respectively. While increasing the number of delays could improve performance further, it comes at a significant computational cost. For example, using 50 delays at a 50-step horizon reduces error to as low as 0.0014 but requires over 150 seconds of computation, an impractical time for real-time control. In contrast, RBF EDMD is limited by overfitting as dimensionality increases. Thus, when computational resources permit, time-delay EDMD presents a promising basis for achieving significantly higher accuracy.

In terms of computational cost, the RBF model requires more training time due to RBF center optimization, but its computation time is low once trained. Conversely, time-delay embedding involves a shorter training time but can incur higher computational demands on the trained model, particularly as the number of delay embeddings increases. On

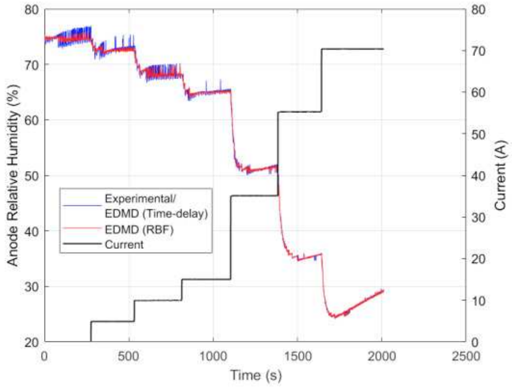

Figure 3: Anode humidity prediction for Koopman models

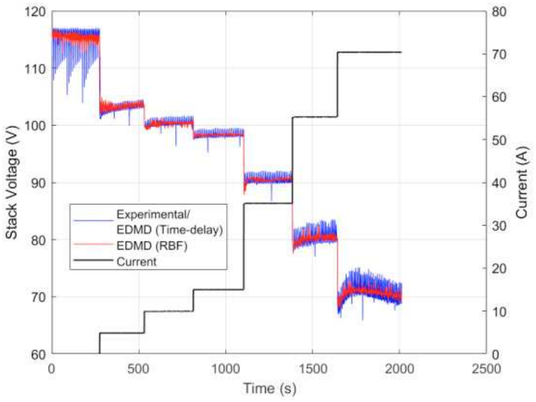

Figure 4: Stack voltage prediction for Koopman models

average, the 25-dimensional optimal RBF EDMD took 5s to compute across all prediction horizons while the time-delay EDMD with 10 delay embeddings took 6.31s (Table 2). However, the marked improvement in accuracy offered by time-delay EDMD may justify this slight increase in computation time. The computations were done on a 1.7GHz Intel Core i7 with 16GB RAM

4.2 NARX Network Test Results

Figures 5 and 6 depicts the results of NARX model in closed-loop mode using indexed and random division over a 5-step prediction horizon for anode relative humidity and stack voltage respectively. When the data are randomly divided, the model performs better, benefiting from exposure to a diverse set of temporal patterns that appear in both training and test sets, enhancing generalization. In contrast, indexed division, where the training and test data are sequentially separated, limits the model's exposure to relevant patterns in the test data, leading to decreased performance. For anode relative humidity, the RMSE decreases from 2.97% with indexed division to 0.77% with random division, and for stack voltage, it drops from 2.01V to 1.05V. Particularly with indexed division, the anode relative humidity exhibits noisier predictions. This likely stems from the model being trained on one set of temporal patterns and tested on another, reducing the ability to capture the data's variability. In contrast, stack voltage is less affected which may be due to its overall dynamics following a broader pattern that the model can capture, enabling better generalization even with indexed division. A possible solution is to train separate NARX models for each output, allowing model parameters to be tuned according to the specific dynamics of each target.

Table 1: Normalized combined RMSE for test set

Model	Prediction horizon			
	5	10	25	50
RBF EDMD	0.0177	0.0192	0.0230	0.0262
Time-Delay EDMD	0.0000	0.0147	0.0159	0.0209
NARX (random)	0.0206	0.0218	0.0266	0.0375
NARX (indexed)	0.0452	0.0614	0.0835	0.0973

Table 2: Computation time for models

Model	Prediction Horizon			
	5	10	25	50
RBF EDMD	4.40s	4.95s	4.97s	5.71s
Time-Delay EDMD	5.59s	5.70s	6.42s	7.54s
NARX (random)	20.43s	9.56s	5.58s	4.96s
NARX (indexed)	18.80s	9.00s	5.06s	4.81s

4.3 Comparison of Koopman and NARX models

The results reveal that the NARX network, when employing a random data division approach, demonstrates comparable performance to the Koopman operator with RBF as basis. This suggests that exposing the NARX model to portions of the test data during training enables it to effectively capture the system's dynamics. However, the computation time of the NARX network is significantly higher for shorter prediction horizons. However, as the prediction horizon increases, the NARX model becomes more computationally efficient. This is because, at longer horizons, the NARX model requires fewer resets, reducing the overall computational burden. For indexed data division, the NARX model struggles to match the performance of Koopman-based methods, as it lacks exposure to the test set patterns during training. In comparison, Koopman approaches maintain consistent performance even when tested on completely unseen data. Furthermore, the NARX model lacks interpretability, whereas Koopman-based methods provide a clear framework for analyzing important system properties such as stability and controllability. The linearity of the Koopman operator also makes it easier to integrate with linear control strategies.

5. CONCLUSION

In this study, the performance of the Koopman-based models with RBF and time-delay embedding as basis functions, and the NARX neural network for predicting anode relative humidity and stack voltage of a 5kW open-cathode PEM fuel cell stack is evaluated. Time-delay EDMD achieves the highest accuracy outperforming both RBF EDMD and the NARX model. RBF EDMD struggles with overfitting, while NARX model performs well with random division but lacks interpretability and struggles with generalization under indexed indivision. While time-delay EDMD may have a slightly higher computational cost compared to RBF EDMD, it still offers significant benefits in terms of accuracy and system insights. Overall, the Koopman approach, particularly with time-delay as basis, provides consistent performance and easier integration with linear control strategies, making it more suitable for real-time control applications.

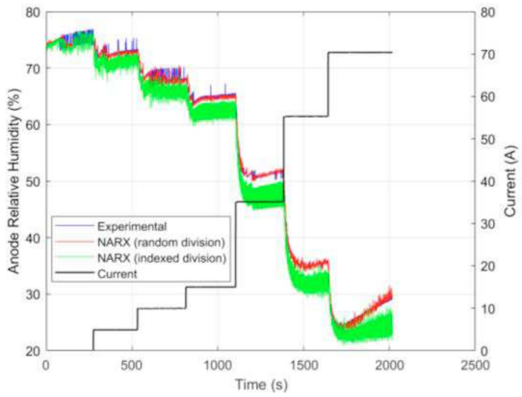

Figure 5: Anode relative humidity prediction for NARX model

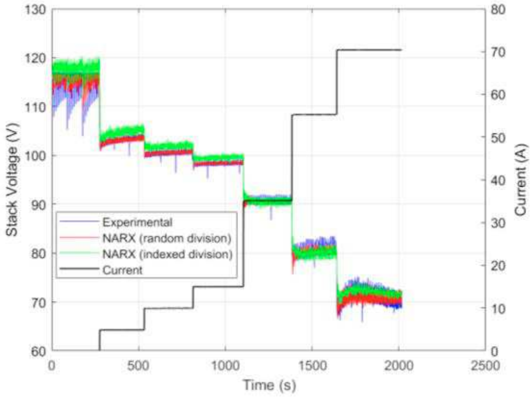

Figure 6: Stack voltage prediction for NARX model

ACKNOWLEDGMENTS

This material is based on work supported by the National Science Foundation, United States under Grant No. 2135735.

REFERENCES

Cho, Y., Hwang, G., Gbadago, D.Q. & Hwang, S. (2022). Artificial neural-network-based model predictive control for optimal operating conditions in proton-exchange membrane fuel cells. *Journal of Cleaner Production, 380*(2), 135049.

Da Costa Lopes, F., Watanabe, E. H., & Rolim, L. G. B. (2015). A Control-Oriented Model of a PEM Fuel Cell Stack Based on NARX and NOE Neural Networks. *IEEE Transactions on Industrial Electronics*, *62*(8), 5155–5163.

Dylewsky, D., Kaiser, E., Brunton, S. L., & Kutz, J. N. (2020). *Principal component trajectories for modeling spectrally-continuous dynamics as forced linear systems*.

Han, I., & Chung, C. (2016). Performance prediction and analysis of a PEM fuel cell operating on pure oxygen using data-driven models : A comparison of artificial neural network and support vector machine. *International Journal of Hydrogen Energy*, 2–11.

Headley, A. J. (2013). *Dynamic Subdivided Relative Humidity Model of a Polymer Electrolyte Membrane Fuel Cell APPROVED BY SUPERVISING COMMITTEE : May*.

Huo, D., Peng, Q., & Hall, C. M. (2023). Koopman-Based Modeling of an Open Cathode Proton Exchange Membrane Fuel Cell Stack. *IFAC-PapersOnLine*, *56*(3), 67–72.

Huo, H., Chen, J., Wang, K., Wang, F., Jin, G., & Chen, F. (2023). State Estimation of Membrane Water Content of PEMFC Based on GA-BP Neural Network. *Sustainability (Switzerland)*, *15*(11).

Koopman, B. O., & Neumann, J. V. (1932). *Dynamical Systems of Continuous Spectra* (Vol. 18, Issue 3).

Korda, M., & Mezić, I. (2016). *Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control*.

Legala, A., Zhao, J., & Li, X. (2022). Machine learning modeling for proton exchange membrane fuel cell performance. *Energy and AI*, *10*(July), 100183.

Morner, S. O., & Klein, S. A. (2001). Experimental evaluation of the dynamic behavior of an air-breathing fuel cell stack. *Journal of Solar Energy Engineering, Transactions of the ASME*, *123*(3), 225–231.

Pan, S., & Duraisamy, K. (2020). On the structure of time-delay embedding in linear models of non-linear dynamical systems. *Chaos*, *30*(7).

Proctor, J. L., Brunton, S. L., & Kutz, J. N. (2014). *Dynamic mode decomposition with control*.

Pukrushpan, J. T., Peng, H., & Stefanopoulou, A. G. (2004). Control-oriented modeling and analysis for automotive fuel cell systems. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, *126*(1), 14–25.

Takens, F. (1980). *Detecting strange attractors in turbulence*.

Williams, M. O., Kevrekidis, I. G., & Rowley, C. W. (2015). A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science,*25, 1307-1346.

Zeng, T., Zhang, C., Huang, Z., Li, M., Chan, S. H., Li, Q., & Wu, X. (2019). Experimental investigation on the mechanism of variable fan speed control in Open cathode PEM fuel cell. *International Journal of Hydrogen Energy*, *44*(43), 24017–24027.