Available online at www.sciencedirect.com

Chemical Engineering Research and Design

journal homepage: www.elsevier.com/locate/cherd

Machine learning-based predictive control using on-line model linearization: Application to an experimental electrochemical reactor

Junwei Luo[a], *Berkay Çıtmacı*[a], *Joon Baek Jang*[a], *Fahim Abdullah*[a], *Carlos G. Morales-Guio*[a,*], *Panagiotis D. Christofides*[a,b,**]

[a] *Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095-1592, USA*
[b] *Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095-1592, USA*

ARTICLE INFO

Article history:
Received 22 July 2023
Received in revised form 9 August 2023
Accepted 9 August 2023
Available online 14 August 2023

Keywords:
Recurrent neural network (RNN) modeling
Koopman operator-based linearization
Electrochemical reactors
Model predictive control (MPC)
Real-time control
Transfer learning

ABSTRACT

The electrochemical reaction-based process, a new type of chemical process that can generate valuable products using renewable electricity, is a sustainable alternative to the traditional chemical manufacturing processes. One promising research area of electrochemical reaction processing is to reduce carbon dioxide (CO_2) into carbon-based products, which can contribute to closing the carbon cycle if CO_2 is directly captured from the atmosphere. In this work, we demonstrate a model predictive control (MPC) scheme that uses a neural network (NN) model as the process model to implement real-time multi-input-multi-output (MIMO) control in an electrochemical reactor for CO_2 reduction. Specifically, a long short-term memory network (LSTM) model is developed using historical experimental data of the electrochemical reactor to capture the nonlinear input-output relationship as an alternative to the complex, first principles-based model. Furthermore, the Koopman operator method is used to linearize the LSTM model to reduce the nonlinear optimization step in the MPC to a well-understood and easy-to-solve quadratic programming (QP) problem. The performance of the LSTM model, Koopman-based optimization, and MPC using the linearization of the LSTM model are evaluated with various simulations as well as open-loop and closed-loop experiments. As the results, the proposed MPC scheme can drive the two output states, that are concentrations of the products (C_2H_4 and CO), to their desired setpoints by computing optimal input variables (surface potential and electrode rotation speed) in real-time in closed-loop experiments. Furthermore, a transfer learning-based method is utilized to update the NN model to handle process variability.

1. Introduction

In today's chemical manufacturing industry, fossil fuels serve as the primary energy source for the chemical industry, leading to significant energy consumption and greenhouse gas emissions (Boulamanti et al., 2017). Alternatively, there has been increasing interest in electrochemical reactions, such as converting carbon dioxide (CO_2) into carbon-based fuels and chemicals with electricity, as a means to mitigate CO_2 emissions. This approach holds the potential for leveraging electricity generated from renewable resources as an

* Corresponding author.
** Corresponding author at: Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095-1592, USA.
E-mail addresses: moralesguio@ucla.edu (C.G. Morales-Guio), pdc@seas.ucla.edu (P.D. Christofides).
https://doi.org/10.1016/j.cherd.2023.08.017
0263-8762/

energy source for large-scale chemical manufacturing, furthermore contributing to global-scale renewable energy storage and closing the anthropogenic carbon cycle (De Luna et al., 2019). Although electrochemical conversion of waste CO_2 is very promising, several challenges hinder the widespread adoption of electrochemical reactors on an industrial scale. Perhaps most importantly, the conversion of CO_2 through electrochemistry requires significant energy consumption (Sullivan et al., 2021). Researchers have focused on improving energy efficiency in electrochemistry through the development of more efficient and selective catalysts through nanostructuring, doping of transition metals, utilization of single-atom catalysts, etc. (Nitopi et al., 2019; Li et al., 2020; Kim et al., 2023; Cao et al., 2022) as well as the design of devices to reduce the overall cell potential and address parasitic carbonation problems (Xie et al., 2022; Zhang et al., 2022b; Ramdin et al., 2023). On the other hand, discussions on process scale-up have been limited so far (Richard et al., 2023). We have identified another critical challenge in scaling up electrolyzers to be the absence of advanced process control schemes for electrochemical reactors due to the complex and nonlinear nature of electrochemical processes. Since the realization of an economically viable electrochemical process will require optimization in process integration and cascade reactor train (Ramdin et al., 2021; Ozden et al., 2021; Fan et al., 2023), the development of a control scheme to regulate individual electrochemical reactor units is necessary.

To address this issue, Çıtmacı et al. (2022a) proposed a feedback control scheme using proportional-integral (PI) controllers utilizing a support vector regression-based (SVR) hybrid model as a state estimator. This approach enabled real-time state estimation for a PI controller and, subsequently, implementation of single-input-single-output (SISO) control in a gastight rotating cylinder electrode (RCE) cell. Building on this work, Çıtmacı et al. (2023) introduced a recurrent neural network (RNN) model as an improved state estimator, surpassing the performance of the SVR model. This RNN model captured relationships between process variables and gas product concentration and allowed for the implementation of multi-input-multi-output (MIMO) control using PI control techniques for the same RCE reactor. Alongside the classical control strategies, model predictive control (MPC) methods have emerged as vital components in industrial process control design (Qin and Badgwell, 2003; Lee, 2011). MPC offers the advantage of computing optimal control actions by anticipating future output states, making it a powerful tool for multivariable control while considering process constraints and nonlinearities, for example, (Holkar and Waghmare, 2010).

Although the specific application of MPC in electrochemical reactors is limited, MPC has been widely used in various research areas, including chemical reactors, battery management, and self-driving cars. For instance, Richalet (1993) provided a comprehensive discussion on implementing MPC for a crude oil distillation unit in the petroleum industry. Furthermore, Chavan et al. (2018) explored MPC design for a multivariable distillation column, demonstrating superior performance compared to PI-based control through MATLAB simulations using the Wood and Berry Model. MPC has also been applied to develop a battery management system, which is similar to the application of an electrochemical reactor in the sense that both tasks involve manipulating electrochemical reactions, even though the battery management system focuses on storing and releasing electricity instead of generating products using electrical potential. For example, Pozzi et al. (2020) proposed a nonlinear MPC design based on the electrochemical models capturing the internal phenomena of the battery to solve the charge unbalancing problem in lithium-ion cells connected in series. These applications have demonstrated the ability of MPC to control systems with electrochemical reactions. Considering the advantages of MPC over classic control strategies (e.g., PI control) with respect to explicitly handling actuator and state constraints, multivariable interactions and nonlinearities, it is potentially practical and valuable to leverage the application of MPC to control electrochemical reactors.

Implementing MPC in electrochemical reactors poses two major challenges that need to be overcome: model accuracy and computational expense. The accuracy of the model prediction is crucial for the performance of MPC. Ideally, a first-principles-based model that accurately captures the underlying phenomena in the electrocatalytic system would be optimal. However, such models are often unavailable for practical cases. To this end, in our research, we focus on a data-driven approach to model the process system. Data-driven modeling offers a systematic approach that can be applied to any process system if sufficient data quantity and quality are ensured. One of the significant examples of data-driven modeling is machine learning (ML), which is a class of techniques that can be generally applied to various systems without the need for formulating specific physical patterns discovered in experiments (Dobbelaere et al., 2021). Classical ML methods, including SVR, linear regression, Gaussian process regression, and decision trees, have been widely utilized for modeling tasks (Xi et al., 2007; Singh et al., 2017; Bhadriraju et al., 2019; Hewing et al., 2019; Zhang et al., 2022a). Additionally, deep learning methods, employing neural network (NN) structures, have demonstrated superior performance in capturing nonlinear and complex systems compared to classical ML methods. As a result, NN modeling has drawn significant attention and has been applied in recent research works (Núñez et al., 2019; Shin et al., 2020). Considering the nonlinearity and complexity of the electrochemical reactor and to facilitate the modeling process, a NN method is utilized to model the system, which has been demonstrated to be an effective technique for this specific reactor in Çıtmacı et al. (2023).

While the use of nonlinear data-driven models in MPC has shown promising performance in various research studies, implementing MPC with a nonlinear model generally involves solving a non-convex optimization problem. This complexity often results in high computational costs and unstable gradient concerns. Xie and Ren (2020) demonstrated in their work that using RNN models in MPC can be highly accurate yet intractable to solve in real-time, motivating the exploration of linearization approaches to improve the computational efficiency of MPC. Several techniques have been proposed for linearizing nonlinear models, such as Taylor series, piecewise linearization, etc. (Schei, 1997; Lin et al., 2013; Ławryńczuk, 2014). Specifically, the Koopman operator method is developed to be a data-driven approach that can be applied to any nonlinear model (Korda and Mezić, 2018; Proctor et al., 2018; Arbabi et al., 2018). Xie and Ren (2020) showed the Koopman operator method is a type of linearization approach that can have better performance than the classical Taylor series method, particularly when

linearizing over a larger domain. Furthermore, the application of MPC using a linearized model has been studied in-depth in the chemical engineering domain (e.g., Mendis et al., 2019; Yang and Wan, 2022). These results have highlighted the potential of employing MPC with on-line linearized models in practical control applications. By leveraging efficient linearization techniques, NN-based MPC can potentially be applied to control the electrochemical reactor effectively.

Motivated by the above considerations, this study aims to develop an advanced process control scheme using MPC with suitable process models for an electrochemical CO_2 reduction reactor. Specifically, a neural network model is initially constructed using reactor data to capture the nonlinear complex input-output relation of the reactor, followed by on-line linearization of the NN model using the Koopman operator method to reduce the computational cost of MPC. The control design is applied experimentally to the electrochemical reactor. The paper is organized into the following sections: Section 2 introduces the background information of this study, including the mathematical notation used in this paper, the overall design of the process, and the equipment setup. Section 3 elaborates on the technical details for the design and development of a NN model. Section 4 discusses the Koopman operator method and the procedure of using it to linearize the NN model in real-time. Finally, Section 5 reports the results of this study including simulation, open- and closed-loop experiments.

2. Preliminaries

2.1. Notation

For a matrix $\mathbf{M}$, the notation $\mathbf{M}^{-1}$ is used to represent the inverse of the matrix $\mathbf{M}$ and $\mathbf{M}^{\dagger}$ denotes the pseudoinverse of matrix $\mathbf{M}$. x, $\hat{x}$, and u are the controlled outputs of the process control system (i.e., the productivity of the reactor for the targeting species), the prediction of the process output given by the process model (i.e., the NN model), and the inputs (control actions) calculated by the process control system (i.e., applied potential, rotation speed, and current), respectively.

2.2. Process overview

The overall objective of our process is to electrochemically reduce CO_2 into valuable chemical products and fuels. A copper electrode is used in this process because it is the only known single-element catalyst that can reduce CO_2 into C_{2+} hydrocarbons and alcohol products, which are energy-dense and valuable, with a considerable production rate (Popović et al., 2020). However, the process of electrochemical CO_2 reduction on copper is intricate, which results in the production of 17 different chemicals through a series of complex reaction pathways (Nitopi et al., 2019). Among multiple factors contributing to the complex reaction mechanisms, mass transport and reaction kinetics play critical roles. Specifically, the transport phenomena in the diffusion boundary layer are directly related to the residence time of the reactant CO_2 and intermediates near the catalyst surface as well as the adsorption on and desorption from the catalyst, which determine the selectivity of final products. On the other hand, the reaction kinetics on the catalyst surface is related to the number of electrons transferred across the surface, which can be manipulated by the applied potential. Therefore, we aim to control the selectivity of electrochemical CO_2 reduction by controlling the aforementioned two input factors, potential and electrode rotation speed. Applying real-time control to any process requires on-line measurements of the process outputs. In this work, the productivity of four gas-phase products (i.e., hydrogen (H_2), carbon monoxide (CO), methane (CH_4), and ethylene (C_2H_4)) can be monitored in real-time using a gas chromatograph (GC). The overall reaction formulas producing these four products are summarized as follows:

$$2CO_2 + 8H_2O + 12e^- \longrightarrow C_2H_4 + 12OH^- \quad (1a)$$

$$CO_2 + H_2O + 2e^- \longrightarrow CO + 2OH^- \quad (1b)$$

$$CO_2 + 6H_2O + 8e^- \longrightarrow CH_4 + 8OH^- \quad (1c)$$

$$2H_2O + 2e^- \longrightarrow H_2 + 2OH^- \quad (1d)$$

Finally, the production rates of CO and C_2H_4 are chosen to be the control outputs to be regulated by the process control system. These two outputs are influenced differently by the input variables; specifically, the production rate of CO is highly correlated to the rotation speed, and the production rate of C_2H_4 is strongly influenced by the applied potential (Çıtmacı et al., 2023). On the platinum anode, the water oxidation reaction to produce oxygen occurs and counter-balances the charge from the half-cell reactions on the working electrode.

2.3. Electrochemical reactor setup

The gastight RCE cell was designed to examine how mass transport and reaction kinetics affect the electrochemical reduction of CO_2 while ensuring a gastight environment for the real-time detection of gas products (Jang et al., 2022). As shown in Fig. 1, the experimental reactor consists of two reaction chambers separated by an anion-exchange membrane preventing the crossover of products. The cathode is the working electrode in the cylindrical geometry carrying out the CO_2 reduction reaction while the Pt foil anode works as the counter electrode. Before each experiment, polycrystalline Cu RCE was mechanically and electrochemically polished following the procedure described in Jang et al. (2022) followed by roughening of the surface via electrochemical redox cycling in the presence of chloride ions (Roberts et al., 2015). The preparation for this catalyst is the same as in our previous work (Çıtmacı et al., 2023). Both the working and the counter electrodes are immersed in 0.2 M potassium bicarbonate electrolyte solutions. During the experiment, the CO_2 gas is directly bubbled into the electrolyte in both chambers with a fixed volumetric flow rate of 20 mL/min. Subsequently, the dissolved CO_2 molecules are transported to the reacting surface on the cathode to be reduced to various products. The potentiostat manipulates the potential applied to the working electrode against a reference Ag/AgCl electrode and records the electrical current passed between the working and the counter electrode. The control of the mass transport properties in the reactor is made possible by magnetically coupling the shaft where the RCE is mounted to another magnet connected to the modulated speed rotator (MSR) outside the reactor.

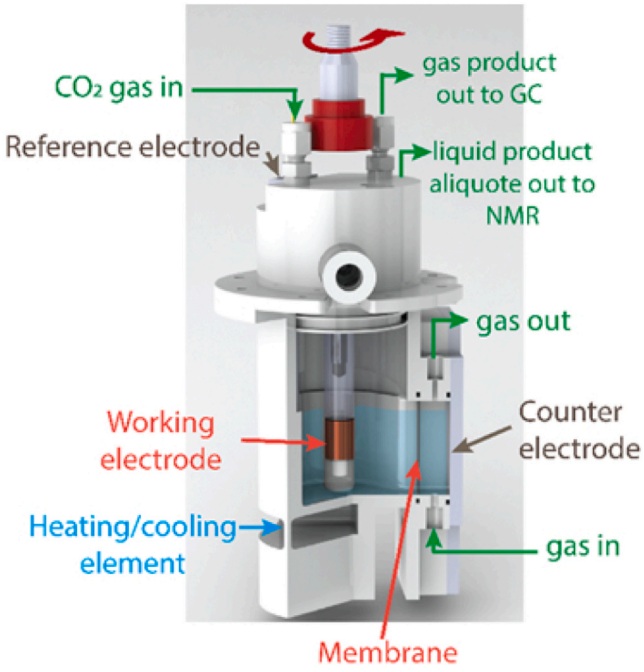

Fig. 1 – The experimental setup of the gastight rotating cylinder electrode (RCE) cell.

2.4. *Model predictive control*

MPC is an advanced control strategy used in various industrial processes. It involves utilizing a dynamic mathematical model of the system to predict its future state or output behavior and optimize control actions by iteratively solving an optimization problem over a defined time horizon. Specifically, MPC determines the optimal control actions to minimize a specified cost function while satisfying system constraints. The design of the MPC in this work can be mathematically defined as the following optimization problem:

$$\mathscr{J} = \min_{u \in S(\Delta)} \int_{t_k}^{t_{k+N_h}} L(\hat{x}(t), u(t))\,dt \tag{2a}$$

$$\text{s.t. } \hat{x}(t) = F_{nn}(x(t), u(t)) \tag{2b}$$

$$L(\hat{x}(t), u(t)) = (\hat{x}(t) - x_r)^T Q (\hat{x}(t) - x_r) + (u(t) - u_r)^T R (u(t) - u_r) \tag{2c}$$

$$u(t) \in U, \quad \forall\ t \in \left[t_k, t_{k+N_h}\right) \tag{2d}$$

$$\hat{x}(t_k) = x(t_k) \tag{2e}$$

$$|u(t_k) - u(t_{k-1})| \le u_c \tag{2f}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are the output states and control actions (calculated by the model predictive control system), respectively. The set U represents the control action space that defines the upper and lower bounds of the m control actions applied to the reactor. The absolute difference between the control actions to be applied in the next control period from the instance time $u(t_k)$ and control action applied in the current control period $u(t_{k-1})$ is bounded by the vector u_c containing absolute boundaries for m control actions (in this particular case, $m = 2$ as we have two manipulated inputs and dimension of $u(t_k)$ is 2). Furthermore, x_r and u_r are the reference values for the output states and control actions. Q and R represent the weight parameters (both are positive definite matrices) of the penalty terms for the output states and control actions, respectively, in the quadratic cost function $L(x, u)$. Therefore, by minimizing the cost function L with an appropriate manipulated input trajectory, the reactor can be driven to the desired set-point given by x_r by applying the first calculated control action $u(t_k)$ at each sampling time, and then repeating this process in the next sampling time. Finally, F_{nn} is the NN model, N_h is the prediction horizon, and the set $S(\Delta)$ comprises of piecewise constant functions having a period of Δ.

In this work, the outputs of the process to be regulated by the model predictive control system, x, are the production rates of CO, C_2H_4, and H_2. Specifically, we are aiming to get the productivity of CO and C_2H_4 to a certain set-point while minimizing the productivity of the side product hydrogen from the competing hydrogen evolution reaction.

3. Neural network modeling

To account for the complexity of the electrochemical reaction mechanism and fill in the lack of a first-principles model, a neural network (NN) model is developed to capture the dynamic response of the output states under various input conditions. Subsequently, the trained NN model is utilized as the process model of the MPC to estimate the output states over a certain time horizon known as the prediction horizon N_h. This section describes the design and development of the NN model for this purpose.

3.1. *Data collection*

The data set used to develop the NN model is similar to the one reported in Çıtmacı et al. (2023), and three types of experiments (i.e., open-loop steady input, step changes, and closed-loop experiments) are performed to collect the data. Specifically, constant inputs (applied potential and catalyst rotation speed) are applied to generate some portion of the training set data, which provides information about the expectation of the steady state output values under certain input conditions in addition to the dynamic trends while reaching respective steady states. In the second type of experiment, step-change inputs of random amplitudes are applied, but the input actions remain in a predefined range throughout the experiment. Finally, the closed-loop experimental results from Çıtmacı et al. (2023) are included in the data set. Although the controller type used in Çıtmacı et al. (2023) is different from the one in this work, the underlying physico-chemical phenomena are the same. Thus, including those results can help the model to capture the dynamic behavior of the system more efficiently.

GC is used to monitor the outlet concentrations of the gas products in real-time during data collection. Specifically, the GC takes a gas sample injection and quantifies the production rates of the four gas-phase products every 1300 s during the experiment. Analyzing the injected gas sample takes 15 min, and the GC needs to cool down for 400 s before taking the next injection. Therefore, only four data points can be collected from a one-hour duration of the experiment. As a result, there are a total of about 200 GC measurements collected at the end of data generation experiments for the training, which is not enough to train a neural network model. To address this problem, a 3rd-order polynomial regression based on three consecutive GC measurements is

applied to determine a probable output data trajectory between every GC measurement using the inputs measured every second. More details about this data enhancement process are reported in Çıtmacı et al. (2023).

3.2. Long short-term memory networks

Among many ML methods that can be used to capture nonlinear processes, the RNN family has been proven to be an effective modeling strategy for time-series forecasting tasks. Recently, RNN models have become popular in the research area of process modeling and control and have been applied in many academic and industrial works (Han et al., 2015; Wu et al., 2019). The long short-term memory network (LSTM) is one of the well-developed NN models that belongs to the RNN family. It shares the major design of architecture with other types of RNN models that have information flowing in two directions to capture the time-dependent relationship within the training data (Ren et al., 2022). Furthermore, the LSTM model has its special "gates design" to store the historical information and determine how to use it to predict the output (Hochreiter and Schmidhuber, 1997).

The architecture of the LSTM model used in this work is shown in Fig. 2. The model is developed to predict the output state at the next consecutive sampling time using p historical state predictions and control actions. Therefore, there are only three outputs given by the model, which represent production rates of CO, C_2H_4, and H_2 (in ppm) at the $p+1$ time step. Specifically, the LSTM layer maps the time-sequence input containing the historical state prediction and control actions to 180 hidden states. Subsequently, a dropout layer with a 30% dropout rate of the hidden states is inserted to prevent overfitting, and the remaining hidden states are densely connected to the output nodes.

Remark 1. The number of hidden states and percentage rate of dropout are included in the hyperparameters of the LSTM model. Therefore, their value can be found following the general hyperparameter tuning process. Specifically, in this work, we performed a random search to locate those values. More precise methods to perform the hyperparameter tuning include cross-validation and grid search. More details about hyperparameter tuning can be found in Feurer and Hutter (2019).

Remark 2. In the area of machine learning, preventing the model from overfitting the data is an important task. Overfitting refers to the situation where the NN model can perform well with the training data set but fails to maintain good performance for data outside the training set. Several factors contribute to this problem, and a critical one is when the NN model has too many weight parameters, which can result in allowing the model to memorize the training data instead of extracting underlying trends from it. One method to reduce overfitting is the regularization (Girosi et al., 1995), with the dropout method used in this work being one example of a regularization method (Wan et al., 2013; Srivastava et al., 2014).

3.3. Model training

Based on prior knowledge of the experimental reactor, the input sequence of the LSTM model is designed to contain one hour (3600 s) of historical information. However, if the data is formatted on a per-second basis, each sequence in the data set will contain 3600 elements, which results in unnecessarily high computational costs in both time and space consumption. Therefore, the time step of the data sequence is designed to be on a per-hundred seconds basis to reduce the length of the input sequence to 36 elements. The time space in the input sequence is preserved in the output sequence, and since the output sequence only contains 1 time step, the overall function of the LSTM model is to use the one-hour historical information up to the instant to predict the output states at 100 s later in the future (i.e., the output sequence has a shape of (1,3), where 3 is the number of output species).

This sliding window algorithm was employed to create a training dataset from a collection of 35 experiments. Specifically, a window of one hour was used as the input for the training data, and the output of the LSTM was determined to be the production rates of the target species at 100 s after the final time step of the input sequence. The window was systematically slid by a stride of 100 s, and the first 1000 (seconds of) measurements in each experiment were skipped to enhance the reliability of the training data. Therefore, the input of the LSTM model has a shape of (36, 6),

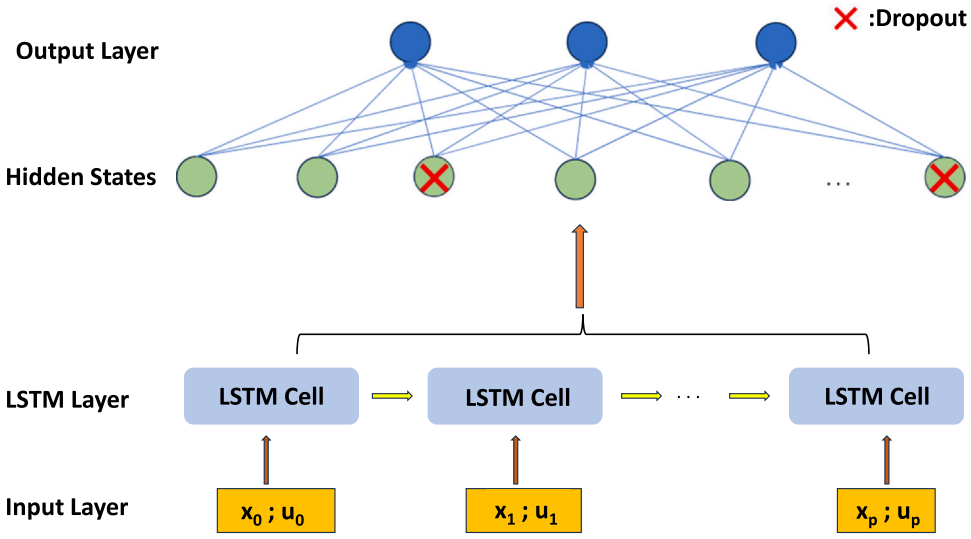

Fig. 2 – The architecture of the LSTM model used in this work that processes the input sequence with a LSTM layer and yields the prediction for the output states at the next time step (i.e., 100 s later from the instantaneous point in time).

where 6 denotes the input features (i.e., surface potential, rotation speed, current, and previous states of production rates for C_2H_4, CO, and H_2) measured at the respective time step. The sliding window algorithm is applied to 18 experimental data sets to generate the data sequence to develop the LSTM model. When developing an NN model for time-series forecasting problems, it is crucial to ensure that the validation data retains a certain level of independence from the training data to avoid potential information leakage. To address this concern, we randomly allocated results from 5 out of the total 18 experiments as the testing set, while the remaining experiments were assigned to the training set. This training set is further divided into two parts before the model training for ratio validation purposes, using the train-test ratio of 70:30. Finally, the Scikit-learn Minmax Scaler was utilized to normalize the data.

In this study, the LSTM model was trained using the TensorFlow API. The model was optimized with the NADAM optimizer. As the data did not provide dense coverage of the overall operating conditions, it was crucial to maintain generalization and prevent overfitting. Therefore, we applied L2 regularization to the LSTM layer with a factor of 0.07 and performed 30% recurrent dropout within the LSTM cells. The mean squared error was selected to be the cost function to evaluate the model performance. The LSTM model underwent training for 45 epochs with a batch size of 32. Additionally, a callback function was utilized to capture the best-performing weights based on minimization of the validation loss throughout the training process. As a result, the training and validation loss of the trained LSTM model were found to be 0.0028 and 0.00456, respectively.

Remark 3. The length of the input sequence (i.e., 3600 s) was found based on the combination of experimental observations and hyperparameter tuning. Specifically, from the experiment, we found that the dead time of the process can vary up to 2000 s, which meant, to capture the delay of the reactor, the length of the input sequence should be at least 2000 s. Starting from there, we tuned the length of the input sequence and found that, with the length of 3600 s, the LSTM model can capture well the dynamics of the process. Notably, increasing the length of the input sequence will result in higher computational cost, and since the length of the input sequence can be considered as a part of the hyperparameter tuning, the cross-validation method was used in this step.

3.4. *Model performance*

The trained model demonstrates significantly low training and validation losses, indicating its successful training. To further assess the model's performance, a comparison is made between the model's predictions, based on input data recorded from a validation experiment, and the corresponding output state measurements. Fig. 3 is an example of such a comparison, with solid curves representing the predictions made by the LSTM model and dashed curves representing the measured trends during the experiment. The close alignment between the curves depicted in Fig. 3 highlights the model's adequate prediction capabilities.

Once the model demonstrated its proficiency in predicting the dynamic behavior of the output state, we proceeded to evaluate its ability to accurately capture the reactor's steady state performance. The electrochemical reactor is an inherently stable process in the operating region of interest, meaning that regardless of the initial output state conditions, the application of the same constant control actions throughout a period of time should lead to the convergence of the outlet species concentrations to the same steady state every time. The prediction results of the LSTM model for an open-loop experiment are shown in Fig. 4. It can be observed that, regardless of the starting point of the trend, the predictions consistently converge to the same steady state for all three output states under a fixed control action. However, due to the stochastic nature of the electrochemical reaction and other experimental uncertainties, there exists a variance in the steady state. Therefore, the steady state given by the LSTM is ideally the average of the steady state values obtained if the experiment is repeated with the same fixed control inputs.

4. Koopman operator-based linearization of RNN model

The motivation behind the exploration of a method to linearize the neural network model for utilization in MPC arises from the fact that NN-based MPC involves solving a constrained optimization problem with a highly nonlinear NN model. Consequently, this optimization problem becomes a challenging nonlinear optimization task, which remains a topic of considerable mathematical exploration without a definitive approach for effective resolution. As a result, solving the nonlinear optimization problem within a reasonable time frame (certainly, within the process sampling time for real-time control purposes) might not be possible, which renders this type of nonlinear MPC application impractical for many industrial processes. On the other hand, the development of an MPC framework with a linearized system is a well-established approach. By approximating the NN model with a linear system on-line and at each sampling time, an MPC can be formulated as a quadratic programming problem, which lends itself to efficient solution techniques. This implies that if we can effectively approximate the NN model with a linear system, the NN model-based MPC can be solved quickly and efficiently, and applied to real-world applications. This section presents the systematic process utilized in this project, drawing inspiration from the work of Xie and Ren (2020), to linearize the RNN-based process model and integrate it into an MPC.

4.1. *Koopman operator theory*

Xie and Ren (2020) presented a method to linearize an RNN model based on the principles of the Koopman operator theory. The Koopman operator theory, initially proposed by Bernard Koopman in the 20th century (Koopman, 1931; Koopman and Neumann, 1932), plays an important role in analyzing, modeling, and controlling nonlinear processes. The core concept of the Koopman operator theory involves mapping inputs of a nonlinear function into a higher-dimensional feature space, thereby obtaining a linear approximation of the nonlinear system (Koopman, 1931). In other words, the Koopman operator can linearize an arbitrary finite-dimensional nonlinear system at the cost of expanding its dimensionality up to infinity. Notably, this concept is also similar to the idea of feature engineering, in the machine learning terminology, which serves as a fundamental aspect

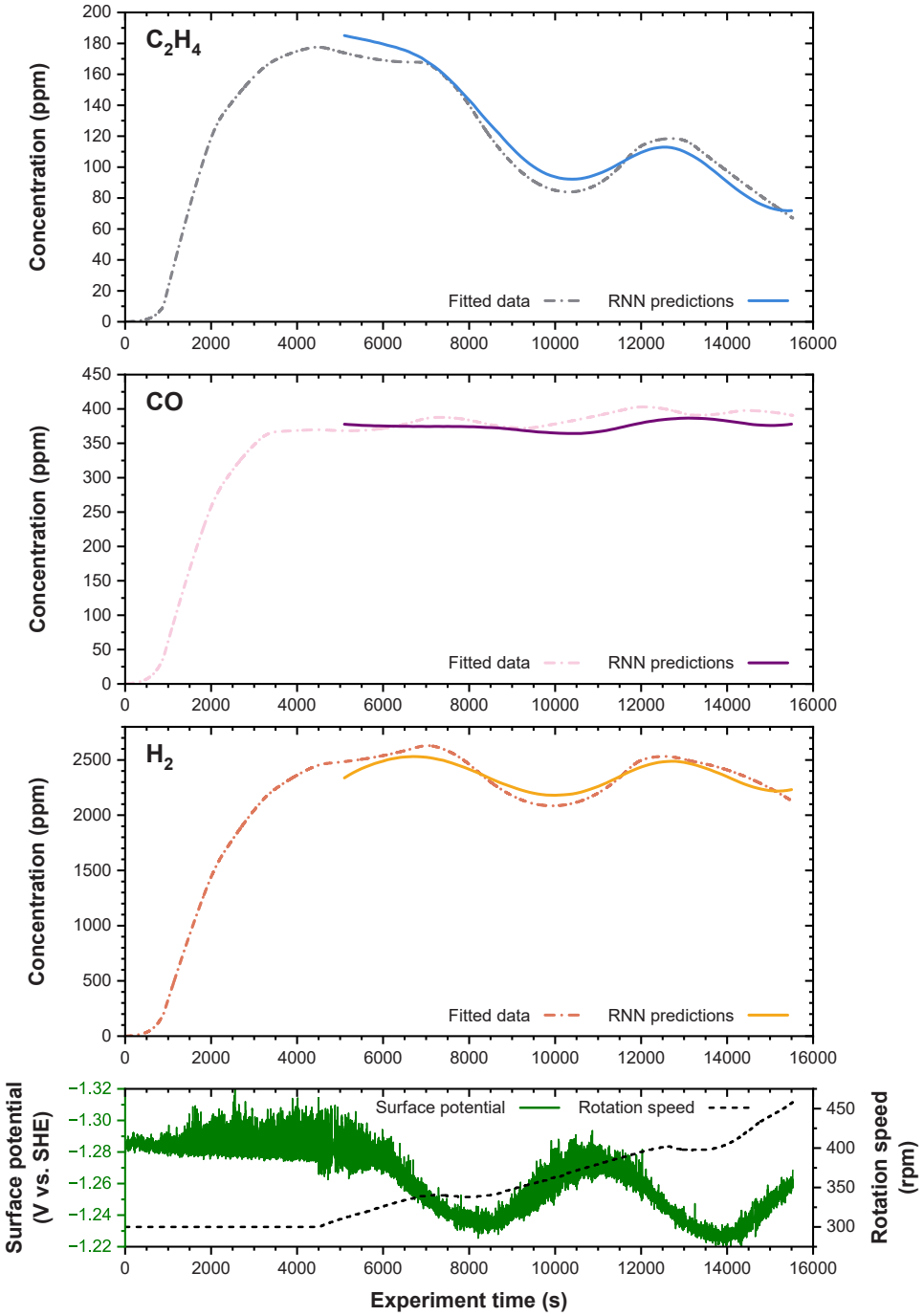

Fig. 3 – LSTM predictions of C_2H_4, CO, and H_2 concentrations compared to the reference data in the testing set. Inputs (surface potential and electrode rotation speed) used for the prediction are shown at the bottom.

in various ML models such as support vector machines (SVM) (Cortes and Vapnik, 1995).

The Koopman operator can be defined mathematically with the following equations:

$$x_{k+1} = \mathbf{f}(x_k) \tag{3a}$$

$$\mathscr{K} g(x_k) \triangleq g(\mathbf{f}(x_k)) = g(x_{k+1}) \tag{3b}$$

where Eq. (3a) is the discrete representation of a nonlinear dynamical system, and the function **f** captures the output state evolution of the system from an arbitrary time step k. Eq. (3b) is the definition of the Koopman operator $\mathscr{K}$, where $g(\cdot)$ are a set of scalar functions named the observables. From Eq. (3b), it can be easily proven that $\mathscr{K}$ is a linear operator, which allows finding the eigen decomposition of $\mathscr{K}$ and rewriting the evolution of the nonlinear system as a linear combination based on the eigen decomposition of $\mathscr{K}$ as follows:

$$\mathscr{K}\phi_j(x) = \lambda_j \phi_j(x), \quad j = 1, 2, \ldots, \infty \tag{4a}$$

$$\mathscr{K} g(x_k) = \sum_{j=1}^{\infty} \lambda_j \phi_j(x_k) \mathbf{v}_j \tag{4b}$$

where λ_j, ϕ_j, $\mathbf{v}_j$ are known as the eigenvalues, the eigenfunctions, and the mode of the Koopman operator $\mathscr{K}$.

The discussion about Koopman operator theory so far has been centered around an autonomous system with time-varying inputs. However, to allow using this method in a dynamic control system requires extending the Koopman theory to be able to handle a non-autonomous system

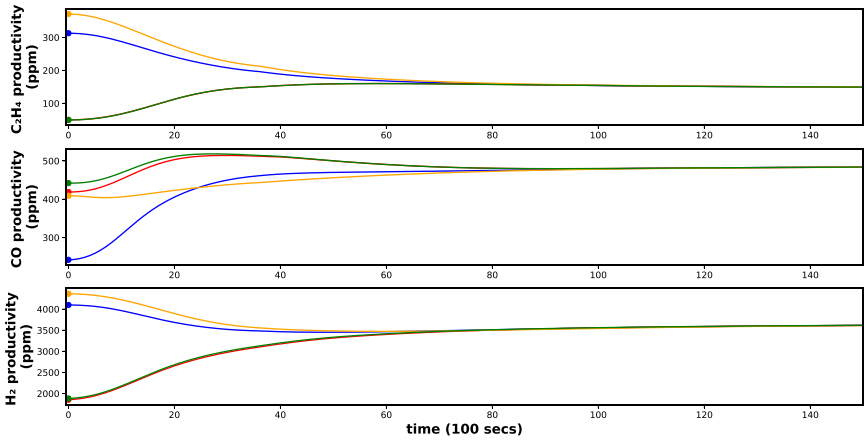

Fig. 4 – Open-loop simulation using the trained LSTM model with consistent fixed inputs from various initial states. The predicted trends for different initial states are represented in different colors.

including time-varying control inputs. In Proctor et al. (2018), a generative Koopman with inputs and control (KIC) method was proposed to generalize the application of Koopman operator theory to non-autonomous systems. Specifically, the KIC method defined a new representation of the Koopman operator as follows:

$$x_{k+1} = \mathbf{f}(x_k, u_k) \tag{5a}$$

$$\mathscr{K} g(x_k, u_k) \triangleq g(\mathbf{f}(x_k, u_k), u_{k+1}) = g(x_{k+1}, u_{k+1}) \tag{5b}$$

where u_k is the input applied at the k^{th} time step, and Eq. (5a) is the discrete representation of any nonlinear system accepting external inputs. There are other works proposing different formulations for Koopman operator with inputs (e.g., Korda and Mezić, 2018), and the core ideas shared around those methods involve augmenting the states x and the inputs u into the same matrix and use it to form the observables instead of just the states, which all allow linearizing the nonlinear system using the method applied to an autonomous system.

Remark 4. The method of constructing the observables g is an essential research area of Koopman operator theory, and there are significant efforts on this subject, such as using a nonlinear function to augment the state measurements (Rowley et al., 2009; Williams et al., 2015; Brunton et al., 2016; Proctor et al., 2018). In this work, we define the observables to be the output states of the system, such that $g(x) = x$.

Remark 5. The Koopman operator can also be applied to a dynamic system with continuous representation. However, the focus of our mathematical analysis and investigation in this work is centered on the discrete representation, as the LSTM model can be considered as a discrete approximation of the underlying nonlinear dynamic system. Therefore, applying the Koopman operator to a discrete nonlinear dynamic system fits better to the application of the Koopman operator to the LSTM model.

4.2. *Dynamic mode decomposition*

Although the Koopman operator method suggests linearizing a nonlinear system into an infinite-dimensional linear system, it is practical to work with a finite dimension that is high enough to achieve the desired accuracy. Considering this, the Dynamic Mode Decomposition (DMD) method, first proposed in Schmid (2010), is an effective method to provide a finite-dimensional approximation of the Koopman operator. Specifically, the DMD method is a data-driven method that requires obtaining measurements to start with. We define $O_k = g(x_k)$ to be the observation of a nonlinear system and $O_k^+ = O_{k+1}$ to be the observation one time-step after O_k. By performing experiments or simulations with the nonlinear system, time-sequence data can be collected for the observations and yield:

$$\mathbf{O} = \left[O_0, O_1, \ldots, O_{n_s}\right], \quad \mathbf{O}^+ = [O_0^+, O_1^+, \ldots, O_{n_s}^+] \tag{6}$$

where n_s is the total number of samples. Notably, the notation O_1 is not necessarily the next time step of O_0. Subsequently, the DMD of the nonlinear system based on the measurements can be found as the eigen decomposition of the linear mapping matrix $\mathbf{A}$ that forms the equation,

$$\mathbf{O}^+ = \mathbf{AO} \tag{7}$$

The analytical solution of Eq. (7) yields the matrix $\mathbf{A}$ as $\mathbf{A} = \mathbf{O}^+\mathbf{O}^\dagger$. Finally, the eigenvalues and eigenvectors of $\mathbf{A}$ are the approximation of the eigenvalues and the mode of the Koopman operator, respectively.

For a nonlinear system with inputs, the Dynamic Mode Decomposition with control (DMDc) method was proposed in Proctor et al. (2016), which includes the measurements of the control actions $\mathbf{O}_u = \left[u_0, u_1, \ldots, u_{n_s}\right]$ to compute the linear mapping matrix $\mathbf{G} = \mathbf{O}^+\begin{bmatrix}\mathbf{O}\\ \mathbf{O}_u\end{bmatrix}^\dagger$ defined for the DMDc method. Similarly, the singular value decomposition of $\mathbf{G}$ can provide a finite approximation of KIC. Eventually, an arbitrary non-autonomous nonlinear system defined as Eq. (5a) can be linearized with the DMDc method into the following system:

$$\mathbf{G} = [\mathbf{AB}] \tag{8a}$$

$$x_{k+1} = \mathbf{A}x_k + \mathbf{B}u_k \tag{8b}$$

$$y_k = \mathbf{C}x_k + \mathbf{D}u_k \tag{8c}$$

Furthermore, the process of computing the matrix $\mathbf{G}$ involves solving a linear least-squares problem, which can be solved more effectively in practice with the regression method rather than finding the analytical solution (Korda and Mezić, 2018). Therefore, the extended dynamic mode decomposition

(EDMD) method proposed in Williams et al. (2015) introduced a regression procedure to approximate the Koopman operator.

4.3. *Linearization of LSTM model and performance evaluation*

The Koopman operator theory and EDMD method are utilized to linearize the LSTM model because they are data-driven and independent of the form of the nonlinear model (Arbabi and Mezic, 2017; Xie and Ren, 2020). The first step of implementing these methods is to collect time sequence trajectories of the LSTM model. Following the procedure of Proctor et al. (2018); Xie and Ren (2020), at the k^{th} time step, we first define the vectors $y = [\hat{x}_{k+1}, \hat{x}_{k+2}, ..., \hat{x}_{k+N_t}]^\top$, $\hat{x} = [\hat{x}_k, \hat{x}_{k+1}, ..., \hat{x}_{k+N_t-1}]^\top$, and $u = [u_k, u_{k+1}, ..., u_{k+N_t-1}]^\top$, where N_t is the distance between the farthest time step contained in the linearization samples and the k^{th} time step. Notably, the historical information that is used by the LSTM model to make predictions up to the k^{th} time step is available at the time t_k. Thus, the prediction $\hat{x}_{k+1}$ can be computed using the LSTM model. Furthermore, by adding the new prediction and the next control action u_{k+1} while removing the first element of the LSTM input, the vector y can be obtained by iteratively running the LSTM model.

In this work, we applied a constraint on how much the input actions can be changed from one sampling time to the next, which is mathematically defined by the following equations:

$$u_k = [v_k, r_k, c_k] \tag{9a}$$

$$c_k = C(v_k, r_k) \tag{9b}$$

$$u_d \triangleq [v_{k+1} - v_k, r_{k+1} - r_k] \tag{9c}$$

$$|u_d| \le u_c = [v_b, r_b] \tag{9d}$$

where v_i, r_i, and c_i are the surface potential, rotation speed, and the current value given by the reactor at the i^{th} sampling time. The surface potential and rotation speed are the control actions that can be manipulated during the experiment, and the current varies as a consequence of these control actions. v_b and r_b are positive numbers referring to the maximum absolute step changes allowed per time step for the potential and rotation speed and are equal to 0.01 V and 30 RPM, respectively.

The data to linearize the LSTM model is generated with respect to the constraints of Eq. (9). Specifically, we first determined the number (N_s) and the length (N_t) of the time-sequence data. Then, we randomly generate N_t control actions starting from the same initial control action u_0 that obey the constraints of Eq. (9) and run the LSTM model to generate one time sequence of "measurements" of y. This process is repeated N_s times to obtain N_s sequences. Tu (2013) pointed out that, due to the reduction of the problem into a linear regression formulation, the data set used to perform the DMD-based method does not need to retain the sequential order of the data points (i.e., the rows of the data matrices can be shuffled such that, for example, the last row of the target vector y, $\hat{x}_{k+N_t}$ can instead be moved to be the first row, as long as the last rows of $\hat{x}$ and u, $\hat{x}_{k+N_t-1}$ and u_{k+N_t-1} are also moved to be their first rows, respectively). Therefore, the data matrices y, x, and u with a shape of ($N_s \times N_t$) can be reshaped into three vectors containing ($N_s \times N_t$) elements (also note $n_s = N_s \times N_t$), as long as the triplets of x_i, u_i, and y_i remain the same. With this data structure, the linear least-square regression problem to find **G** can be easily solved by using the Scikit-learn linear regression function without fitting the intercept. The pseudocode to implement the linearization of LSTM in our work is represented in algorithm 1.

Algorithm 1. Procedure of linearizing the LSTM model.

Input: $v_b \ge 0, r_b \ge 0, NN$ ▷ NN: the initial inputs of LSTM with shape (1, 36, 6)
Input: x_{train} , y_{train} ▷ Define empty arrays to store linearization samples
for $i \leftarrow 1$ **to** N_s **do**
  $NN_i \leftarrow NN$ ▷ Initialize the LSTM input by making a deepcopy of NN
  for $t \leftarrow 1$ **to** N_t **do**
    $v_i \leftarrow NN_i[0, -2, 0] + rand(-v_b, v_b)$ ▷ $rand(l, h)$: randomly pick number between l and h
    $r_i \leftarrow NN_i[0, -2, 1] + rand(-r_b, r_b)$
    $c_i \leftarrow C(v_i, r_i)$ ▷ $C(\cdot)$: eq. 9b
    $NN_i[0, -1, :3] = [v_i, r_i, c_i]$
    $\hat{x}_{i,t} = LSTM(NN_i)$ ▷ $LSTM(\cdot)$: LSTM prediction with shape (3)
    $x_{\text{train}}.append(NN_i[0, -1, :])$
    $y_{\text{train}}.append(\hat{x}_{i,t})$
    $NN_i[0, :-1, :] \leftarrow NN_i[0, 1:, :]$
    $NN_i[0, -1, 3:] \leftarrow \hat{x}_{i,t}$
  end for
end for
$[A, B] \leftarrow LG(x_{\text{train}}, y_{\text{train}})$ ▷ $LG(\cdot)$: Sickit-learn linear regression

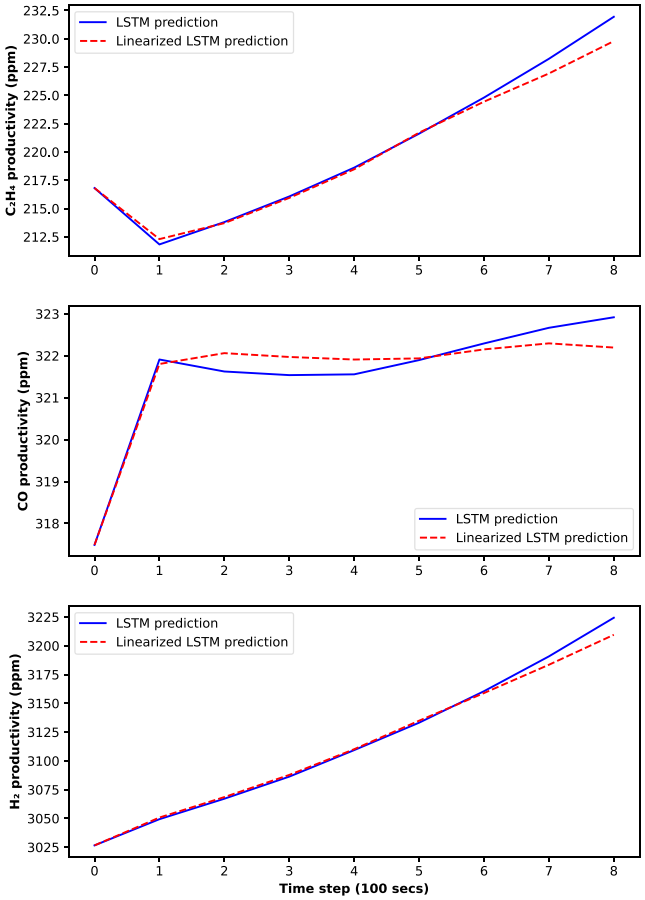

Fig. 5 – Comparison between the linearized model prediction (dashed curve) and the original LSTM model prediction (solid curve) over a sampling period.

The prediction given by the linearized model was compared with the original LSTM prediction. Specifically, the initial inputs to activate the LSTM model are randomly cropped from existing experimental results and provided to algorithm 1 to generate a linearized model. Subsequently, the linearized model was utilized to make predictions over a time span based on a sequence of control actions randomly picked within the step change constraint and compared to the prediction given by the original LSTM model using the same control sequence. The comparison is shown in Fig. 5, where the prediction given by the original LSTM model over a time span of 800 s is represented in the blue solid curve, while the prediction given by the linearized model is denoted in the red dashed curve. The predictions given by the two models are close to each other, which supports that the linearized model can approximate the LSTM model adequately.

Remark 6. The most significant advantage of the proposed NN modeling plus on-line linearization method is taking advantage of the high degree of freedom given by the NN model to facilitate the modeling process. Specifically, we did initially attempt to develop a linear model for the electrochemical reactor, but we found such a model to be ineffective due to the complexity and nonlinearity of the process. In particular, since the electrochemical reactor is a highly nonlinear process, to represent it as a single linear model yields a model with a very high error that is not sufficiently accurate for control purposes. Although this issue can be mitigated to an extent by separating the operational region into multiple sub-regions that can each be described by a local linear system, how to determine the sub-regions (e.g., how many sub-regions are needed, and how to determine the boundaries of each sub-region) is a difficult task. In contrast, building a neural network model and then linearizing it on-line to reduce the computational burden of the MPC calculations proved to be a viable and effective approach since the neural network model can achieve a much lower error due to its ability to capture nonlinear behavior, and the Koopman linearization is aimed at achieving the lowest possible error in a pre-defined segment of the operating region (over n_s data points only).

Remark 7. Notably, for this work, the current flowing at a fixed applied potential depends on the electrolyte solution resistance due to the Ohmic loss. The solution resistance between the working and the reference electrode is determined from the electrochemical impedance spectroscopy (EIS) and is around 6.5 ± 0.3 Ω in the RCE cell setup when using 0.2 M potassium bicarbonate electrolyte. Although the value is practically constant, there are slight variances from experiment to experiment and during the experiment, while it is measured only before and after the experiments (Çıtmacı et al., 2023). Therefore, the measured current value will not be the same with the same control action, and thus, provide additional information for our LSTM model to learn the electrochemical reactor system better. The current value is measured and recorded during the experiment, and those measurements are used to train the LSTM model. However, when collecting samples for the Koopman-based linearization of the LSTM model, the value of the current needs to be approximated with the correlation between c_k and the control actions denoted as transformation

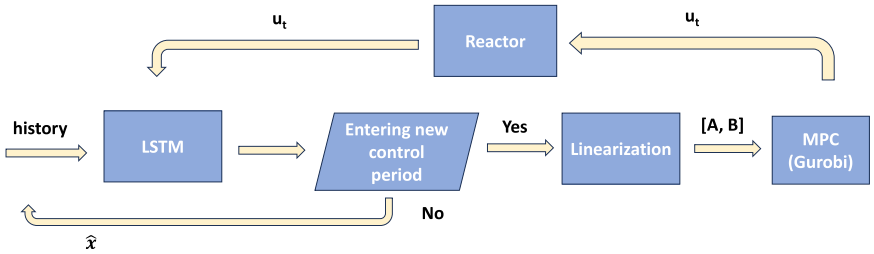

Fig. 6 – The overall workflow of the MPC in this work. The LSTM model is used as a state estimator when the MPC is not activated. Once entering a new sampling time, the MPC is activated and computes the control action for the reactor with the linearization of the LSTM model.

C in Eq. (9b). In simulations, this value was approximated using the average resistance obtained from various experiments. For the closed-loop experiments, the resistance value was measured right before starting the experiment and used to anticipate the current value in the prediction horizon.

5. Closed-loop experiments

The details of implementing the linearized NN-based MPC for the electrochemical reactor are presented in this section. As a quick recap, referring to Eq. (2), the main objective of the MPC in this work is to drive the productivity of C_2H_4 and CO to their specific set-points while suppressing the productivity of H_2. The set-points for C_2H_4 and CO are selected to be 147 ppm and 478 ppm, respectively, such that $x_r = [147, 478, 0]$. Furthermore, by replacing the LSTM model used in Eq. (2d) with the linearized model, the overall optimization problem within the MPC becomes a quadratic programming problem, which is convex and can be solved efficiently. In the closed-loop experiment, the MPC is operated in a sample-and-hold manner, which means it will give the optimum control action over a certain control period (i.e., 100 s in this work), and the control action will be held fixed during the control period. The overall workflow of the MPC is demonstrated in Fig. 6. Specifically, the LSTM model worked as the state estimator throughout the experiment. When entering a new sampling time, algorithm 1 was used to compute the linearization of the LSTM model for the specific time-instant, which was then used to find the MPC control action by solving a quadratic programming (QP) problem. The Gurobi optimizer was used to solve the optimization problem in this work.

Remark 8. We note that the LSTM model used to provide the output state information in the MPC is a nonlinear process model, which causes the MPC to be a nonlinear, non-convex optimization problem. However, when the LSTM model is linearized, the MPC will only have linear constraints which, along with the quadratic objective function used, makes the MPC optimization problem become a quadratic program (QP) that can be solved very efficiently with readily available solvers such as Gurobi.

5.1. *Implementation of the MPC in the experimental setup*

The LSTM model (without linearization) worked as the state estimator in the closed-loop experiments by predicting the instantaneous reactor productivity. When the processing of a new GC measurement is finished, the LSTM model has to reinitialize its prediction, which means that it uses the state measurements in the input of LSTM instead of the state prediction given by the LSTM in the previous iteration. This reinitialization is expected to prevent the accumulation of prediction errors. Specifically, the output states are estimated through 3rd-order polynomials based on the 3 consecutive GC measurements up to the newest measurement and used as the input of the LSTM model to predict the output state at 100 s after the newest GC injection made. Note that the GC measurement has a delay of 15 min because it takes 15 min to separate and analyze the sample taken from the injection. Therefore, through the reinitialization, the LSTM predicted the output state 15 min ago again, and needs to run iteratively using the reinitialized prediction to correct all the predictions for the previous 15 min. Furthermore, since this 3rd-order polynomial approximation can only be activated once every 21 min, it cannot be used as the process model or state estimator that requires to be able to give prediction every 100 s. But once the 3rd-order polynomial approximation is activated, it can estimate the output states for the last 1 hr effectively and accurately.

The Laboratory Virtual Instrument Engineering Workbench (LabVIEW) software was utilized to digitalize the electrochemical reactor in this work. LabVIEW is a graphical programming language that allows a user to develop a user interface to monitor the system and develop control systems to implement the control actions in the working equipment. Although LabVIEW also allows users to develop simple programs (e.g., PI controller), it is technically challenging to implement the aforementioned workflow that involves using the NN model, linearization, and optimization in LabVIEW. Therefore, a data pipeline was developed to allow information to flow between a Python script operating the workflow and the LabVIEW controlling the operating equipment.

The options for pipeline design include reading the data from a real-time updated csv file or data transfer through a database. Since opening a real-time updated csv file to read data might disturb the process of writing data, this option is not optimal. On the other hand, sending data from LabVIEW to a database is an easy task that is already combined into our automation scheme using the Clean Energy Smart Manufacturing Innovation Institute's (CESMII) Smart Manufacturing Innovation Platform (SMIP) as discussed in Çıtmacı et al. (2022b). Specifically, the SMIP can work as a database to store and organize our data at defined endpoints for each piece of equipment, and the use of start and end dates for query and mutation of SMIP's data transfer protocol, GraphQL, makes it a perfect candidate for data transfer application. In short, this data flow is designed to use LabVIEW as the edge device performing process control and

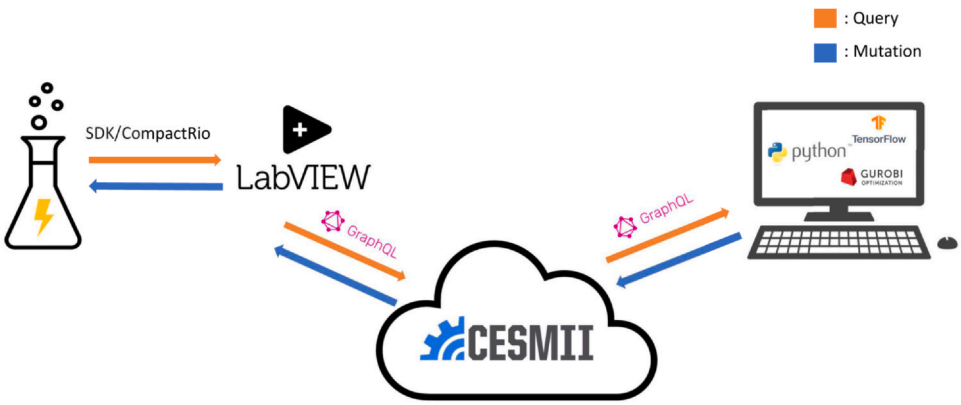

Fig. 7 – Data flow between the experimental setup and local Python script through SMIP for MPC calculations.

monitor tasks, SMIP as a cloud database for data management, and a high-performance computer running Python interpreters as a back-end server. The data transfer protocol is shown in Fig. 7.

When performing an experiment, constant physical properties, such as solution resistance, open circuit potential, etc., are measured before the electrolysis and sent to the SMIP at the beginning of the experiment. Process data collected through LabVIEW, such as applied potential, surface potential, rotation speed, and current, are mutated to the SMIP every 2 s. The reactor was put to run in open-loop for the first 7000 s of the experiment because, at the beginning state of the experiment, the rector does not reach the equilibrium giving higher variance in its productivity. Thus, data collected at this stage is expected to have a different distribution from the rest of the experiment and is excluded from the LSTM training. Control actions applied to the reactor at this stage are fixed to be – 1.22 V for potential and 300 RPM for rotation speed.

After letting the reactor run in open-loop for the first 6298 s, the MPC will be activated, and the Python script queries the last one hour of process data every 100 s to form the initial input for the LSTM model. Subsequently, the LSTM was linearized to compute the first control action, while the original LSTM model estimations along with input values are mutated to SMIP. LabVIEW script also queries those values from the SMIP to feed the new input values to the potentiostat and modulated speed rotator to implement the new applied potential and rotation speed. From then on, the experiment was run in closed-loop.

5.2. *Closed-loop experiments*

The result of the closed-loop experiment is summarized in Fig. 8, where the dots are the GC measurements collected in the experiment and the dashed curves are the approximated output states with probable experimental trajectory method employing 3rd-order polynomials. The productivity evolution of C_2H_4 and CO_2 is shown in the top figure, and their set-points, which are 147 and 478 ppm, respectively, are denoted with the dotted straight lines. A reference control action u_r = [– 1.28 V, 600 RPM], stated in Eq. (2c), was used in the MPC objective function to achieve better control performance. The reference control action u_r was found using the trained LSTM model. As discussed above, the LSTM model is stable, such that it can be used to find the theoretical control actions that can give the targeted steady state. The weight matrix was chosen to be $Q = \text{diag}(0.01,\ 0.01,\ 1 \times 10^{-6})$ and $R = \text{diag}(1 \times 10^{4},\ 1.0/1200)$. The weight parameters were tuned based on the simulation and experiment results.

The design of the weight matrices considered scaling their importance on the cost function. For example, when designing the matrix Q, the first two parameters are the weight of C_2H_4 and CO, respectively, which are equally important in our control scheme. The value 0.01 in the Q matrix was used to prevent the cost value from becoming too big. On the other hand, the cost for the H_2 is much less than the other two weights because driving the outputs to the set-point is the first priority for the control system. Reducing the productivity of the side product H_2 can maximize the energy efficiency of our reactor. However, it is physically impossible to eliminate the H_2 production, which means if the weight parameter for the H_2 is too high, the MPC will allow the two target states to be away from the set-point as the trade-off to reach the optimum defined by the MPC objective function. Drawn from the understanding of the reactor, we considered the reactor operated energy efficiently if the productivity of H_2 was kept below 4000 ppm. Since the states are squared in the objective function, using 1×10^{-6} will give a cost equal to 16 if the productivity of H_2 is at 4000 ppm, which will dominate the MPC with the outputs approaching the set-points, and thus, the productivity of H_2 was not included in the objective function. The weight matrix R was calculated to balance the speed of the convergence of the states to the steady state and the magnitude of the control actions.

The prediction horizon of the MPC (N_h) is 8 times steps (i.e., 800 s in the future), and the length of the time sequence, N_t, collected for linearization is designed to be equal to $N_h + 2$. The number of linearization samples, N_s, is taken to be 30. Theoretically, linear regression can be more accurate with increasing amounts of data. On the other hand, increasing N_t and N_s also requires more time to collect the sample which makes the linearization more computationally expensive. Notably, the sampling step can be processed in parallel, which means the computational time is independent of the size of N_s if there are sufficient amounts of parallel processors. However, since the processing of the time sequence is iterative, the computation time for linearization is

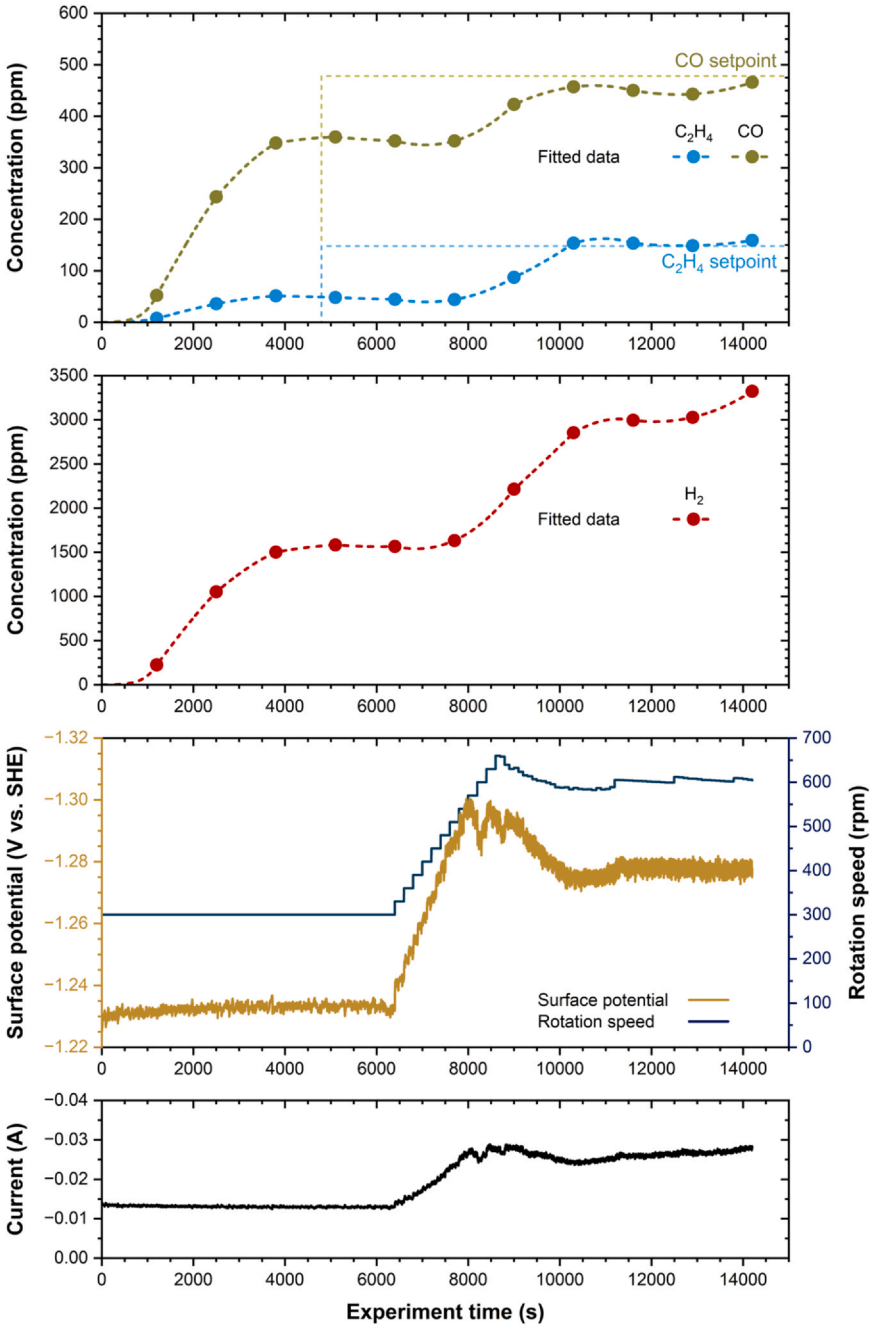

Fig. 8 – Output responses and control actions in the closed-loop experiment controlled by the MPC using the linearization of the LSTM model.

bounded by the size of N_t. In this work, the maximum allowable step changes in u reduce the required number of sequences in our linearization sample. In our implementation, 30 sequences collected with the Monte Carlo method turned out to be sufficient for the linearization task. With this choice, the MPC successfully drives the outputs to the set-point while maintaining the H_2 production rate below 4000 ppm.

5.3. *Model retrain using transfer learning*

The MPC design in this work created a feedback loop by using real-time measurements to re-initiate the LSTM model with measurement feedback and improve closed-loop system robustness. However, a correction algorithm that uses the feedback information to improve the LSTM model was not implemented, which means the control scheme demonstrated in this work is based on the assumption that the LSTM model can capture the real process accurately, and that the process behavior does not change significantly. However, in real-world applications, the system is very likely to perform differently from the model prediction due to the variance of the application. The reported control scheme in this work should be able to handle this slight variance, as long as the variance is not significant enough to have the steady state shifting on a very different condition. However,

sometimes the process may have very different behavior than the data collected to develop the neural network model. This problem is usually called the data (process) shift problem (for example, due to catalyst activity variation as a new catalyst is introduced every certain number of experiments), and more actions need to be taken to account for the data shift problem in model update.

To this end, we introduced a model retrain procedure based on the transfer learning concept, which has been applied to various engineering research works (Xiao et al., 2023; Munoz et al., 2023), to update the process model efficiently when the data shift problem is detected. To imitate this data shift problem in our reactor, we changed the polycrystalline Cu RCE to a new one and followed the same procedure to synthesize nanopores, but the resulting performance was different. Specifically, the new catalyst was more active and had a higher selectivity towards the production of C_2H_4. Various experiments were conducted with the new catalyst, which gave the result that with the control action at – 1.28 V and 600 RPM, the output for C_2H_4 increased to about 200 ppm but remained unchanged for CO.

Subsequently, the LSTM model was retrained off-line based on the data collected from the new experiments. Of course, the new data set is smaller in size compared to the original data set. Therefore, if we just train a new model after including the new data into the original data set, the newly trained LSTM is very likely to count heavily on the old data set and represent less of the performance of the new catalyst. To account for this, we used the idea of transfer learning, which is a scheme to fine-tune a pre-trained model to make it fit better to a new data set. Since the underlying physico-chemical phenomena do not change with the catalyst change, the available LSTM is a good pre-trained model that captures the critical dynamic relations from the previous training.

Specifically, the training of all layers in the pre-trained LSTM model except the output layer was frozen with the assumption that the ground truth physical relationship is captured in the LSTM layer. This assumption was made based on the fact that the same reactor is used, and only the catalyst is different with respect to its morphology. If more significant changes are considered (like different reactor configurations), then one may have to consider retraining additional weight parameters. Subsequently, the model is trained with only the new experimental results, which consist of results from three closed-loop experiments ranging from 8 to 10 h using the new catalyst and the original LSTM model. The model is trained with 10 epochs because it is common in transfer learning to train the model with a small number of epochs to prevent it from overfitting the new data, especially when the size of the new data set is small. While the computational time to train the original LSTM model was about 5 h using Google Collab (i.e., using both GPU and CPU), the update of the existing LSTM network model required only around 15 min. Eventually, the retrained model preserved a stable behavior and predicted the new u_r to be – 1.26 V and 650 RPM. This result matches our expectation for the new catalyst since the C_2H_4 productivity is more correlated to the applied potential while CO productivity is more correlated to the rotation speed. Based on the experiment observation, the new u_r should decrease the potential to reduce the productivity of C_2H_4 to better approach the set-point. However, reducing the potential will also reduce the productivity of CO even if it is more correlated to rotation speed. The rotation speed then needs to be increased slightly to compensate for the loss in CO productivity. Thus, we concluded that the transfer learning-based retrain process calibrates the LSTM model in the correct direction and moved on to using it to perform closed-loop MPC experiments of controlling the reactor with the new catalyst. The result of the closed-loop experiment is shown in Fig. 9 which demonstrates that the MPC with the retrained model can stabilize the reactor outputs to the desired set-points.

Remark 9. In addition to the retraining method that corrects the model off-line, on-line correction may be implemented to improve the MPC performance using real-time measurements. For example, the extended Kalman filter method can be a good candidate to be considered for this task. To implement this method, the Kalman correction factor should be added to the LSTM model for real-time estimation. Furthermore, since the Koopman method can be applied to any nonlinear model, the overall workflow of the model linearization and MPC implementation does not need to be changed to include the Kalman filter correction. Developing this correction step for an MPC of an electrochemical reactor is one of our future objectives.

Remark 10. The computational cost of solving the original MPC optimization problem in each sampling time (which is 100 s in our experimental implementation) depends on the choice of optimization solver and its setup. Specifically, in our work, we used the COIN-or IPOPT optimizer (Wächter, 2009) to solve the nonlinear optimization problem and use 300 iterations as the maximum number of iterations. With this setup, it took at most 300 s to solve the LSTM-based MPC optimization problem for this application, which significantly exceeds the sampling time of 100 s. On the other hand, the QP for the MPC resulting from the on-line linearization procedure can always be solved in a few seconds with the Gurobi optimizer. In this work, we did not find it necessary to implement a parallel computing scheme in the MPC calculations, as the overall MPC runtime without parallel computing was around 10 s.

Remark 11. The retraining correction requires collecting new data from the process, which may introduce a certain delay to update the MPC. Consider the case where the data shift problem is just detected, and the collected data is not enough to retrain the process model, the MPC should be deactivated and switched to a backup controller (e.g., classical proportional integral controller) to ensure that the process operates safely.

Remark 12. We claim three main advantages of using the MPC approach compared to a traditional steady-state control approach (where the inputs are just set to steady-state values corresponding to the desired set-points from the beginning of the experiment; steady-state input values are predicted by the original LSTM model). The first advantage is that the use of MPC enables minimizing the production rate of H_2 during operation. Secondly, the overall transient approach to the set-points can be optimized via MPC. Finally, the use of measurement-feedback control achieves robustness against disturbances and reactor variability. The robustness of the MPC to reactor variability can be observed in Fig. 9, where the LSTM model suggests operating the

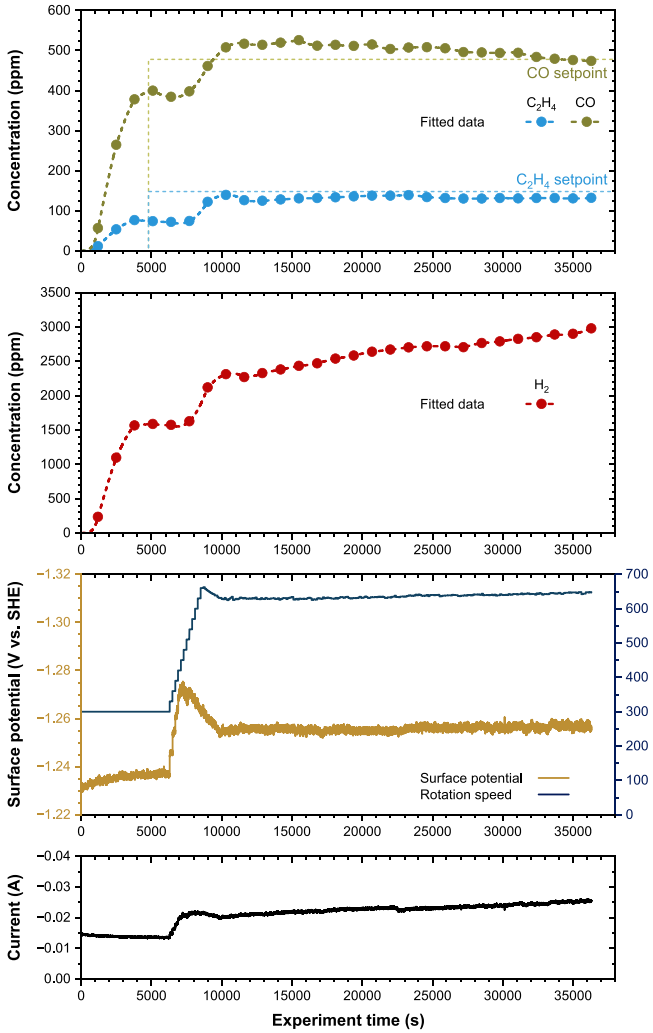

Fig. 9 – Output responses and control actions with new catalyst controlled by the MPC using the retrained model.

reactor with a steady-state surface potential of – 1.26 V, but the MPC actually dictates the steady-state surface potential to be – 1.25 V, which is about 10% of the allowable range, to drive the process outputs to the desired set-points.

6. Conclusion

A procedure to apply neural network model-based MPC to perform real-time multivariable control for an experimental electrochemical reactor was presented; the approach involves on-line linearization of the neural network model and is applicable to broad classes of chemical processes. Specifically, in this study, an LSTM neural network model was used to capture the nonlinear dynamic input-output relationship to control an electrochemical reactor that converts CO_2 to valuable chemical products. The Koopman operator method was found to be able to linearize the LSTM model efficiently (in terms of computational effort) and effectively (in terms of model performance). Based on that method, a systematic approach was developed to linearize a neural network model using the Scikit-learn linear regression function, which is efficient and easy to implement. Open-loop simulations were performed to evaluate the performance of the original LSTM and linearized LSTM models, and the MPC developed based on the linearization of the LSTM model was applied to control the experimental electrochemical reactor. As the closed-loop results demonstrated, the MPC calculates the optimal control actions with reasonable computation cost and successfully drives the process outputs to desired set-point values. Furthermore, a transfer-learning scheme was introduced to account for the data shift problem (owing to catalyst activity variability every time a new catalyst is introduced) by updating the LSTM model using new process measurement data. The transfer-learning method was demonstrated to be able to update the original LSTM model with a limited amount of new data and computational resources. Finally, the updated LSTM model and the resulting MPC were demonstrated to resolve the data shift problem by driving the process outputs to the desired set-points in a closed-loop experiment under the new experimental (catalyst) conditions.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We would like to gratefully acknowledge financial support from the U.S. Department of Energy, through the Office of Energy Efficiency and Renewable Energy (EERE), under the

Advanced Manufacturing Office Award Number DE-EE0007613. Financial support from the National Science Foundation through the award CBET-2140506 is also gratefully acknowledged.

References

Arbabi, H., Mezic, I., 2017. Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator. SIAM J. Appl. Dyn. Syst. 16, 2096–2126.

Arbabi, H., Korda, M., Mezić, I., 2018.A data-driven koopman model predictive control framework for nonlinear partial differential equations, In: 2018 IEEE Conference on Decision and Control (CDC), 6409–6414.

Bhadriraju, B., Narasingam, A., Kwon, J.S.I., 2019. Machine learning-based adaptive model identification of systems: application to a chemical process. Chem. Eng. Res. Des. 152, 372–383.

Boulamanti, A., Moya, J.A., et al., 2017.Energy efficiency and GHG emissions: Prospective scenarios for the chemical and petrochemical industry. Report 9789279657344, EU Science Hub.

Brunton, S.L., Brunton, B.W., Proctor, J.L., Kutz, J.N., 2016. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. PLOS One 11, e0150171.

Cao, Y., Chen, S., Bo, S., Fan, W., Li, J., Jia, C., Zhou, Z., Liu, Q., Zheng, L., Zhang, F., 2022. Single atom bi decorated copper alloy enables C-C coupling for electrocatalytic reduction of CO_2 into C2+ products. Angew. Chem. Int. Ed., e202303048.

Chavan, S., Birnale, N., Deshpande, A.S., 2018.Design and simulation of model predictive control for multivariable distillation column, In: 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 764–768.

Çıtmacı, B., Luo, J., Jang, J.B., Canuso, V., Richard, D., Ren, Y.M., Morales-Guio, C.G., Christofides, P.D., 2022a. Machine learning-based ethylene concentration estimation, real-time optimization and feedback control of an experimental electrochemical reactor. Chem. Eng. Res. Des. 185, 87–107.

Çıtmacı, B., Luo, J., Jang, J.B., Korambath, P., Morales-Guio, C.G., Davis, J.F., Christofides, P.D., 2022b. Digitalization of an experimental electrochemical reactor via the smart manufacturing innovation platform. Digit. Chem. Eng. 5, 100050.

Çıtmacı, B., Luo, J., Jang, J.B., Morales-Guio, C.G., Christofides, P.D., 2023. Machine learning-based ethylene and carbon monoxide estimation, real-time optimization, and multivariable feedback control of an experimental electrochemical reactor. Chem. Eng. Res. Des. 191, 658–681.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20, 273–297.

De Luna, P., Hahn, C., Higgins, D., Jaffer, S.A., Jaramillo, T.F., Sargent, E.H., 2019. What would it take for renewably powered electrosynthesis to displace petrochemical processes? Science 364, eaav3506.

Dobbelaere, M.R., Plehiers, P.P., Van de Vijver, R., Stevens, C.V., Van Geem, K.M., 2021. Machine learning in chemical engineering: strengths, weaknesses, opportunities, and threats. Engineering 7, 1201–1211.

Fan, L., Zhao, Y., Chen, L., Chen, J., Chen, J., Yang, H., Xiao, Y., Zhang, T., Chen, J., Wang, L., 2023. Selective production of ethylene glycol at high rate via cascade catalysis. Nat. Catal.

Feurer, M., Hutter, F., 2019. Hyperparameter optimization. Automated machine learning: Methods, systems, challenges. Springer, pp. 3–33.

Girosi, F., Jones, M., Poggio, T., 1995. Regularization theory and neural networks architectures. Neural Comput. 7, 219–269.

Han, H.G., Zhang, L., Hou, Y., Qiao, J.F., 2015. Nonlinear model predictive control based on a self-organizing recurrent neural network. IEEE Trans. Neural Netw. Learn. Syst. 27, 402–415.

Hewing, L., Kabzan, J., Zeilinger, M.N., 2019. Cautious model predictive control using gaussian process regression. IEEE Trans. Control Syst. Technol. 28, 2736–2743.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780.

Holkar, K., Waghmare, L.M., 2010. An overview of model predictive control. Int. J. Control Autom. 3, 47–63.

Jang, J., Rüscher, M., Winzely, M., Morales-Guio, C.G., 2022. Gastight rotating cylinder electrode: toward decoupling mass transport and intrinsic kinetics in electrocatalysis. AIChE J. 68.

Kim, B., Tan, Y.C., Ryu, Y., Jang, K., Abbas, H.G., Kang, T., Choi, H., Lee, K.S., Park, S., Kim, W., et al., 2023. Trace-level cobalt dopants enhance co2 electroreduction and ethylene formation on copper. ACS Energy Lett. 8, 3356–3364.

Koopman, B.O., 1931. Hamiltonian systems and transformation in Hilbert space. Proc. Natl. Acad. Sci. 17, 315–318.

Koopman, B.O., Neumann, J.V., 1932. Dynamical systems of continuous spectra. Proc. Natl. Acad. Sci. 18, 255–263.

Korda, M., Mezić, I., 2018. Linear predictors for nonlinear dynamical systems: koopman operator meets model predictive control. Automatica 93, 149–160.

Ławryńczuk, M., 2014. Computationally efficient model predictive control algorithms: a neural network approach. volume 3. Studies in Systems, Decision and Control Series. Springer,.

Lee, J.H., 2011. Model predictive control: Review of the three decades of development. Int. J. Control, Autom. Syst. 9, 415–424.

Li, M., Wang, H., Luo, W., Sherrell, P.C., Chen, J., Yang, J., 2020. Heterogeneous single-atom catalysts for electrochemical CO_2 reduction reaction. Adv. Mater. 32, 2001848.

Lin, M.H., Carlsson, J.G., Ge, D., Shi, J., Tsai, J.F., 2013. A review of piecewise linearization methods. Math. Probl. Eng. 2013, 101376.

Mendis, P., Wickramasinghe, C., Narayana, M., Bayer, C., 2019. Adaptive model predictive control with successive linearization for distillate composition control in batch distillation, In: 2019 Moratuwa Engineering Research Conference (MERCon), 366–369.

Munoz, S.A., Park, J., Stewart, C.M., Martin, A.M., Hedengren, J.D., 2023. Deep transfer learning for approximate model predictive control. Processes 11, 197.

Nitopi, S., Bertheussen, E., Scott, S.B., Liu, X., Engstfeld, A.K., Horch, S., Seger, B., Stephens, I.E., Chan, K., Hahn, C., et al., 2019. Progress and perspectives of electrochemical CO_2 reduction on copper in aqueous electrolyte. Chem. Rev. 119, 7610–7672.

Núñez, F., Langarica, S., Díaz, P., Torres, M., Salas, J.C., 2019. Neural network-based model predictive control of a paste thickener over an industrial internet platform. IEEE Trans. Ind. Inform. 16, 2859–2867.

Ozden, A., Wang, Y., Li, F., Luo, M., Sisler, J., Thevenon, A., Rosas-Hernández, A., Burdyny, T., Lum, Y., Yadegari, H., Agapie, T., Peters, J.C., Sargent, E.H., Sinton, D., 2021. Cascade CO_2 electroreduction enables efficient carbonate-free production of ethylene. Joule 5, 706–719.

Popović, S., Smiljanić, M., Jovanovič, P., Vavra, J., Buonsanti, R., Hodnik, N., 2020. Stability and degradation mechanisms of copper-based catalysts for electrochemical CO_2 reduction. Angew. Chem. 132, 14844–14854.

Pozzi, A., Zambelli, M., Ferrara, A., Raimondo, D.M., 2020. Balancing-aware charging strategy for series-connected lithium-ion cells: a nonlinear model predictive control approach. IEEE Trans. Control Syst. Technol. 28, 1862–1877.

Proctor, J.L., Brunton, S.L., Kutz, J.N., 2016. Dynamic mode decomposition with control. SIAM J. Appl. Dyn. Syst. 15, 142–161.

Proctor, J.L., Brunton, S.L., Kutz, J.N., 2018. Generalizing Koopman theory to allow for inputs and control. SIAM J. Appl. Dyn. Syst. 17, 909–930.

Qin, S.J., Badgwell, T.A., 2003. A survey of industrial model predictive control technology. Control Eng. Pract. 11, 733–764.

Ramdin, M., De Mot, B., Morrison, A.R., Breugelmans, T., Van Den Broeke, L.J., Trusler, J.P., Kortlever, R., De Jong, W., Moultos, O.A., Xiao, P., Webley, P.A., Vlugt, T.J., 2021. Electroreduction of CO_2/CO to C2products: process modeling, downstream

separation, system integration, and economic analysis. Ind. Eng. Chem. Res. 60, 17862–17880.
Ramdin, M., Moultos, O.A., van den Broeke, L.J., Gonugunta, P., Taheri, P., Vlugt, T.J., 2023. Carbonation in low-temperature CO_2 electrolyzers: causes, consequences, and solutions. Ind. Eng. Chem. Res. 62, 6843–6864.
Ren, Y.M., Alhajeri, M.S., Luo, J., Chen, S., Abdullah, F., Wu, Z., Christofides, P.D., 2022. A tutorial review of neural network modeling approaches for model predictive control. Comput. Chem. Eng. 165, 107956.
Richalet, J., 1993. Industrial applications of model based predictive control. Automatica 29, 1251–1274.
Richard, D., Jang, J., Çıtmacı, B., Luo, J., Canuso, V., Korambath, P., Morales-Leslie, O., Davis, J.F., Malkani, H., Christofides, P.D., Morales-Guio, C.G., 2023. Smart manufacturing inspired approach to research, development, and scale-up of electrified chemical manufacturing systems. iScience 26, 106966.
Roberts, F.S., Kuhl, K.P., Nilsson, A., 2015. High selectivity for ethylene from carbon dioxide reduction over copper nanocube electrocatalysts. Angew. Chem. 127, 5268–5271.
Rowley, C.W., Mezić, I., Bagheri, S., Schlatter, P., Henningson, D.S., 2009. Spectral analysis of nonlinear flows. J. Fluid Mech. 641, 115–127.
Schei, T.S., 1997. A finite-difference method for linearization in nonlinear estimation algorithms. Automatica 33, 2053–2058.
Schmid, P.J., 2010. Dynamic mode decomposition of numerical and experimental data. J. Fluid Mech. 656, 5–28.
Shin, Y., Smith, R., Hwang, S., 2020. Development of model predictive control system using an artificial neural network: a case study with a distillation column. J. Clean. Prod. 277, 124124.
Singh, B., Sihag, P., Singh, K., 2017. Modelling of impact of water quality on infiltration rate of soil by random forest regression. Model. Earth Syst. Environ. 3, 999–1004.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1929–1958.
Sullivan, I., Goryachev, A., Digdaya, I.A., Li, X., Atwater, H.A., Vermaas, D.A., Xiang, C., 2021. Coupling electrochemical CO_2 conversion with CO_2 capture. Nat. Catal. 4, 952–958.
Tu, J.H., 2013.Dynamic mode decomposition: Theory and applications. Ph.D. thesis. Princeton University.
Wächter, A., 2009.Short tutorial: Getting started with ipopt in 90 min, In: Dagstuhl Seminar Proceedings, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
Wan, L., Zeiler, M., Zhang, S., LeCun, Y., Fergus, R., 2013. Regularization of neural networks using dropconnect. Int. Conf. Mach. Learn., PMLR 1058–1066.
Williams, M.O., Kevrekidis, I.G., Rowley, C.W., 2015. A data–-driven approximation of the Koopman operator: extending dynamic mode decomposition. J. Nonlinear Sci. 25, 1307–1346.
Wu, Z., Tran, A., Rincon, D., Christofides, P.D., 2019. Machine learning-based predictive control of nonlinear processes. part I: theory. AIChE J. 65, e16729.
Xi, X.C., Poo, A.N., Chou, S.K., 2007. Support vector regression model predictive control on a hvac plant. Control Eng. Pract. 15, 897–908.
Xiao, M., Hu, C., Wu, Z., 2023. Modeling and predictive control of nonlinear processes using transfer learning method. AIChE J. 69, e18076.
Xie, K., Ozden, A., Miao, R.K., Li, Y., Sinton, D., Sargent, E.H., 2022. Eliminating the need for anodic gas separation in CO_2 electroreduction systems via liquid-to-liquid anodic upgrading. Nat. Commun. 13, 1–9.
Xie, S., Ren, J., 2020. Linearization of recurrent-neural-network-based models for predictive control of nano-positioning systems using data-driven Koopman operators. IEEE Access 8, 147077–147088.
Yang, S., Wan, M.P., 2022. Machine-learning-based model predictive control with instantaneous linearization–a case study on an air-conditioning and mechanical ventilation system. Appl. Energy 306, 118041.
Zhang, T., Li, S., Zheng, Y., 2022a. Implementable stability guaranteed lyapunov-based data-driven model predictive control with evolving gaussian process. Ind. Eng. Chem. Res. 61, 14681–14690.
Zhang, Z., Lees, E.W., Ren, S., Mowbray, B.A., Huang, A., Berlinguette, C.P., 2022b. Conversion of reactive carbon solutions into CO at low voltage and high carbon efficiency. ACS Cent. Sci. 8, 749–755.