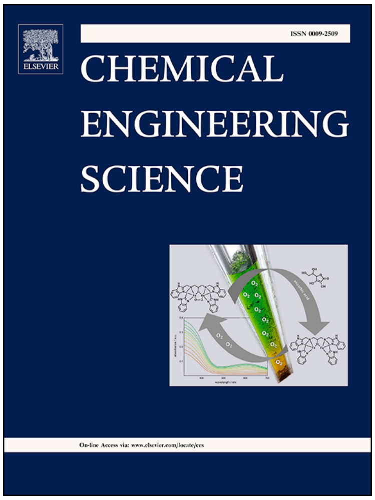

Nonlinear Model Predictive Control of a Multiscale Thin Film Deposition Process Using Artificial Neural Networks

Grigoriy Kimaev, Luis A. Ricardez-Sandoval

PII: S0009-2509(19)30608-6
DOI: https://doi.org/10.1016/j.ces.2019.07.044
Reference: CES 15126

To appear in: *Chemical Engineering Science*

Received Date: 6 April 2019
Revised Date: 3 July 2019
Accepted Date: 18 July 2019

Nonlinear Model Predictive Control of a Multiscale Thin Film Deposition Process Using Artificial Neural Networks

Grigoriy Kimaev, [1]Luis A. Ricardez-Sandoval

Dept. of Chemical Engineering, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1

Abstract

The purpose of this study was to employ Artificial Neural Networks (ANNs) to develop data-driven models that would enable the shrinking horizon nonlinear model predictive control of a computationally intensive stochastic multiscale system. The system of choice was a simulation of thin film formation by chemical vapour deposition. Two ANNs were trained to estimate the system's observables. The ANNs were subsequently employed in a shrinking horizon optimization scheme to obtain the optimal time-varying profiles of the manipulated variables that would meet the desired thin film properties at the end of the batch. The resulting profiles were validated using the stochastic multiscale system and a good agreement with the predictions of the ANNs was observed. Due to their observed computational efficiency, accuracy, and the ability to reject disturbances, the ANNs seem to be a promising approach for online optimization and control of computationally demanding multiscale process systems.

Keywords: nonlinear model predictive control, shrinking horizon optimization, multiscale stochastic system, machine learning, artificial neural network

[1] Corresponding author: laricard@uwaterloo.ca; Phone: +1(519)-888-4567 x38667; Fax: +1(519)-888-4347

1. Introduction

Numerous industrial chemical engineering processes are comprised of coupled phenomena which span several temporal and spatial scales, for example: semiconductor doping, protein binding to cell membrane, the production of thin films by ion beam sputtering, thermal atomic layer deposition or chemical/physical vapour deposition for optical devices, biomedical coatings, solar cells, integrated circuits and micro-electro-mechanical systems (MEMS), as well as heterogeneous catalysis with interactions between a fluid phase and catalyst surface, and continuous and plug-flow protein crystallization in pharmaceutical applications (R. A. Adomaitis, 2003; Chaffart, Rasoulian, & Ricardez-Sandoval, 2016; Cheimarios, Kokkoris, & Boudouvis, 2010; Christofides & Armaou, 2006; Christofides, Armaou, Lou, & Varshney, 2009; Crose, Kwon, Tran, & Christofides, 2017; Crose, Sang-Il Kwon, Nayhouse, Ni, & Christofides, 2015; Kwon, Nayhouse, Christofides, & Orkoulas, 2013, 2014; Kwon, Nayhouse, Orkoulas, & Christofides, 2014; Lee, Mohr, Kwon, & Wu, 2018; J. Li, Croiset, & Ricardez-Sandoval, 2015; Raimondeau & Vlachos, 2000; Salciccioli, Stamatakis, Caratzoulas, & Vlachos, 2011; Zhang, Ding, & Christofides, 2019). While the phenomena and structure at the microscale directly affect the final product's properties and performance, in the industrial setting only the macroscale quantities (e.g. inlet concentration(s), inlet flow rate(s), temperature, etc.) are available for the control of multiscale processes by the manipulation of the macro and microscale phenomena to ensure that the product meets design specifications (Christofides & Armaou, 2006; Christofides et al., 2009; J. Huang, Zhang, Orkoulas, & Christofides, 2011).

In order to expedite the product development cycle and lessen the cost associated with the design of materials, stochastic multiscale models have been developed to gain insight on the interactions between different spatial and temporal scales and identify the underlying mechanisms that impact process performance (Raymond A. Adomaitis, 2010a, 2010b; Crose, Zhang, Tran, & Christofides, 2018; Kwon, Nayhouse, & Christofides, 2015; Lam & Vlachos, 2001; Raimondeau & Vlachos, 2000; Salciccioli et al., 2011; Vlachos, 1997, 2005). Such models consist of coupled simulations of multiple temporal and spatial scales and provide sufficiently accurate results at a high computational cost. While the macroscopic scale phenomena are adequately represented by continuous deterministic equations such as Navier-Stokes equations of motion (Bird, Stewart, & Lightfoot, 2002), a realistic depiction of the microscale events requires the usage of non-closed-form expressions to properly capture the random (stochastic) behaviour of matter at atomistic and molecular scales (Schroeder, 2007). Popular schemes for the atomistic and molecular simulations include Metropolis Monte Carlo (MMC), kinetic Monte Carlo (kMC), molecular dynamics (MD), simulated annealing, and others (Frenkel & Smit, 2001). The coupling between scales is usually implemented through the boundary conditions. The non-closed-form expressions are used to “inform” the calculations at the continuum scale and thus increase the accuracy of the predictions. However, the observables produced by stochastic multiscale models can exhibit the noise that has been propagated from the discrete simulations. Furthermore, while the results of such models can be accurate and realistic, they often cannot be used for optimization and online control due to their high computational intensity, which can be mainly attributed to evaluating the non-closed-form expressions.

In order to diminish the computational cost associated with the stochastic multiscale models based on first principles, data-driven models such as Kriging, Wiener-Hammerstein, reduced-order models, and others (B. Ding & Ping, 2012; Jalali, Nieuwenhuyse, & Picheny, 2017; Kleijnen, 2017; Lawrynczuk, 2016; Narasingam & Kwon, 2018) can be employed to find the empirical relationships (e.g. linear and nonlinear correlations) between the manipulated and/or process state variables and the control variables (Garg, Corbett, Mhaskar, Hu, & Flores-Cerrillo, 2017; Garg & Mhaskar, 2017; Hasenauer, Jagiella, Hross, & Theis, 2015; Kimaev & Ricardez-Sandoval, 2017; Kwon, Nayhouse, Orkoulas, et al., 2014; Meidanshahi, Corbett, Adams II, & Mhaskar, 2017; Narasingam & Kwon, 2018; Oladyshkin & Nowak, 2012; Rasoulian & Ricardez-Sandoval, 2014, 2015a, 2015b, 2016; Siddhamshetty, Wu, & Kwon, 2018; Solvason, 2011; Solvason, Chemmangattuvalappil, & Eden, 2010; Wong, Chee, Li, & Wang, 2018; Yuan, Jiao, Quddus, Kwon, &

Mashuga, 2019). In the data-driven system identification approach, the accuracy in the responses of observables to manipulated variables is preserved, while at the same time the computational cost associated with the stochastic multiscale models is substantially alleviated. However, finding empirical correlations can completely disregard the underlying fundamental relationships between the manipulated variables and the observed responses. Thus, a notable drawback of data-driven models can be their inability to predict the correct responses to perturbations that have not been used during the development of empirical models, especially for extrapolation, i.e. when the new inputs lie significantly far away from the ranges of input values used to construct the data-driven models (Svozil, Kvasnicka, & Pospichal, 1997).

It has been shown that data-driven models based on machine learning techniques such as Artificial Neural Networks (ANNs) are capable of accurately and efficiently predicting system behaviour and responses to perturbations (Fausett, 1993). ANNs can establish complex relationships between multiple manipulated and response variables, but they require large datasets for accurate model training. When the provided datasets are small, ANNs are prone to overfitting, i.e. they memorize the relationships between datapoints, which impedes their ability to make the correct predictions when provided with inputs that were not used in training. However, when properly trained, a particularly appealing feature of ANNs is their capacity to make reasonably accurate predictions even when they are provided with data that has not been used during model development (Seagaran, 2007).

During the recent decades, there has been a resurgence of research interest in ANNs (and other machine learning techniques), with numerous applications in nonlinear signal processing, modelling time series data, medical diagnosis, electrical load forecasting, and hydrologic models, in addition to many other fields (Akkisetty, Lee, Reklaitis, & Venkatasubramanian, 2010; Antanasijevic, Pocajt, Peric-Grujic, & Ristic, 2014; Bashir & El-Hawary, 2009; Chitsazan, Nadiri, & Tsai, 2015; Chow & Leung, 1996; Dayhoff & DeLeo, 2001; Du Tao, Wang Xiuli, & Wang Xifan, n.d.; Fausett, 1993; C.-M. Huang & Yang, 2001; Khanmohammadi, Tutun, & Kucuk, 2016; Lang, Waibel, & Hinton, 1990; Lapedes & Farber, 1987; Lian, Zeng, Yao, Tang, & Chen, 2016; Oliveira, 2004; Seagaran, 2007; Sutskever & Hinton, 2010; Tiwari & Chatterjee, 2010; Zhao-Yang Dong, Bai-Ling Zhang, & Qian Huang, 2001). A recent overview of machine learning applications in chemical engineering can be found in (Venkat Venkatasubramanian, 2019). The existing applications of ANNs that are highly relevant to the field of chemical engineering include process fault diagnosis, process modelling and control, the design of fuel additives and catalysts, predicting two-phase flow patterns, optimization of thermal atomic layer deposition, as well as other important problems in computational fluid dynamics, food chemistry, renewable energy, bioresources, etc. (Alexandridis & Sarimveis, 2005; Bach-Andersen, Rømer-Odgaard, & Winther, 2018; Chaffart & Ricardez-Sandoval, 2018; Y. Ding et al., 2019; K. Huang, Zhan, Chen, & Lü, 2003; Krishnapura & Jutan, 2000; Kumar & Venkateswarlu, 2012; G. Li et al., 2018; Nandi, Ghosh, Tambe, & Kulkarni, 2001; Ozkaya, Demir, & Bilgili, 2007; Sarkar & Modak, 2003; Shaikh & Al-Dahhan, 2003; Sharma, Das, & Samanta, 2006; Sundaram, Ghosh, Caruthers, & Venkatasubramanian, 2001; V. Venkatasubramanian & Vaidyanathan, 1992; Venkat Venkatasubramanian, 2009, 2019; Venkat Venkatasubramanian & Chan, 1989; Watanabe, Matsuura, Abe, Kubota, & Himmelblau, 1989; Wu et al., 2019; Yıldız, Uzun, Ceylan, & Topcu, 2016).

Recently, ANNs have been applied to a stochastic multiscale model that simulated thin film deposition (Chaffart & Ricardez-Sandoval, 2018). In that study, a stochastic partial differential equation (SPDE) simulated thin film surface evolution and was coupled to partial differential equations (PDEs) that modelled the continuum scale phenomena. The ANNs were used to efficiently predict the coefficients of the SPDE based on the time-dependent manipulated variables. The application of ANNs to the model enabled online optimization and control of the thin film deposition process. To the best of our knowledge, that work was the first application of ANNs for multiscale stochastic systems. However, as far as we are aware, there are no applications where ANNs are used to completely replace stochastic multiscale models and, thanks to their computational efficiency, enable online optimization and control of these systems.

Motivated by this gap in knowledge, we trained ANNs to capture the behaviour of a stochastic multiscale model that coupled kinetic Monte Carlo (kMC) with PDEs to simulate thin film formation by chemical vapour deposition. To assess the performance of ANNs for closed-loop control of batch multiscale process systems (such as thin film production), we applied the ANNs in a nonlinear model predictive control (NMPC) framework that used shrinking horizon optimization (Joseph & Hanratty, 1993; Rasoulian & Ricardez-Sandoval, 2015a, 2016; Soni & Parker, 2004; Thomas, Kardos, & Joseph, 1994) to identify the optimal profiles of the manipulated variables that would allow the final thin film to meet the specified product requirements. We subsequently validated the optimal solutions using the full stochastic multiscale system and tested the robustness of ANN models to disturbances that were not used in model development.

The remainder of this paper is organized as follows. The next section contains an overview of the multiscale system that was used in the present analysis to generate the data for training and validating the Artificial Neural Networks (ANNs). The shrinking horizon optimization problem is posed after the multiscale model overview. After that, an overview of the ANNs is presented, followed by a description of the structures of the networks used to capture the responses of the multiscale model's observables to manipulated variables. Next, in the results section, we solve the shrinking horizon optimization problem using ANNs under various conditions and analyze the performance of the ANNs, including their ability to reject disturbances. We conclude with a discussion of the advantages and potential drawbacks of applying ANNs to capture the behaviour of multiscale stochastic systems.

2. The multiscale model and its optimization

In this section, we present a brief overview of the stochastic multiscale model employed in this study. For a detailed description of the calculations, parameters and their values, we refer the reader to our previous publications (Kimaev & Ricardez-Sandoval, 2017, 2018). Further in this section, we present the shrinking horizon optimization problem formulation.

2.1 A brief overview of the stochastic multiscale model

The stochastic multiscale model of thin film formation by chemical vapour deposition used for the purposes of this study consists of two scales coupled through a boundary condition. The macroscale simulation is comprised of momentum, energy and mass transport equations, all simplified to one spatial dimension by taking advantage of radial symmetry (Christofides et al., 2009; Lam & Vlachos, 2001):

$$\frac{\partial}{\partial\tau}\left(\frac{\partial f}{\partial\eta}\right)=\frac{\partial^3 f}{\partial\eta^3}+f\frac{\partial^2 f}{\partial\eta^2}+\frac{1}{2}\left[\frac{\rho_b}{\rho}-\left(\frac{\partial f}{\partial\eta}\right)^2\right] \tag{1}$$

$$\frac{\partial T}{\partial\tau}=\frac{1}{Pr}\frac{\partial^2 T}{\partial\eta^2}+f\frac{\partial T}{\partial\eta} \tag{2}$$

$$\frac{\partial x}{\partial\tau}=\frac{1}{Sc}\frac{\partial^2 x}{\partial\eta^2}+f\frac{\partial x}{\partial\eta} \tag{3}$$

Equations (1)-(3) model the transport of gas particles within the layer of chemical vapour immediately above the thin film, i.e. the gas layer between the bulk chemical vapour (where the dimensionless distance, η, is considered to approach infinity) and the substrate (η→0). Equations (1)-(3) are subject to the following boundary conditions in the bulk (η→∞):

$$T=T_{bulk} \tag{4}$$

$$\frac{\partial f}{\partial\eta}=1 \tag{5}$$

$$x=\mathbb{X} \tag{6}$$

and at the thin film surface (η→0):

$$T=\mathbb{T} \tag{7}$$

$$f=0 \tag{8}$$

$$\frac{\partial f}{\partial\eta}=0 \tag{9}$$

$$\frac{\partial x}{\partial\eta}=\frac{Sc(R_a-R_d)}{\sqrt{2a\mu_b\rho_b}} \tag{10}$$

The microscale of thin film formation is simulated by the kinetic Monte Carlo (kMC) model that considers adsorption, desorption, and migration events. The rates of these events are calculated by equations (11), (12) and (13), respectively (Christofides et al., 2009; Lam & Vlachos, 2001):

$$W_a = \frac{S_0 P x|_{\eta\to 0}}{\sqrt{2\pi m R_g \mathbb{T}} C_{tot}} N^2 \tag{11}$$

$$W_d = \sum_{i=1}^{5} M_i k_{d0} e^{-\frac{iE + E_d}{R\mathbb{T}}} \tag{12}$$

$$W_m = \sum_{i=1}^{5} M_i k_{d0} e^{-\frac{iE + E_m}{R\mathbb{T}}} \tag{13}$$

The decision as to which of the three possible surface events will be performed is determined by the relative magnitudes of the rates W_a, W_d and W_m and a uniform random number from the interval (0,1), as described in (Vlachos, 1997).

The coupling of the discrete and continuum scales is implemented through the boundary condition (10) of the mass transfer equation (3). Figure 1 depicts the coupling process. The net number of atoms adsorbed on the thin film, $N_a - N_d$, is tracked by the kMC code and is subsequently used to update the net rate of adsorption, $R_a - R_d$, which in turn is used to calculate the rate of change of the gas precursor mole fraction x with respect to the dimensionless distance away from the substrate η, i.e. ${\partial x}/{\partial \eta}\big|_{\eta\to 0}$ of equation (10). Once this update is complete, the rate of change of the gas precursor mole fraction with respect to the dimensionless time τ, i.e. ${\partial x}/{\partial \tau}$ of equation (3), can also be updated, and the value of the gas precursor mole fraction at the thin film surface, $x|_{\eta\to 0}$, can be obtained from equation (3). This value is subsequently used to update the rate of adsorption, W_a, which is used by the kMC simulation to adjust the probabilities of adsorption, desorption and migration events on the thin film surface. Once the kMC code has run for a pre-specified coupling time interval, $\Delta\mathcal{T}$, the update (described above) is carried out again. This feedback loop between the two scales (see Figure 1) is repeated until the final simulation batch time is reached.

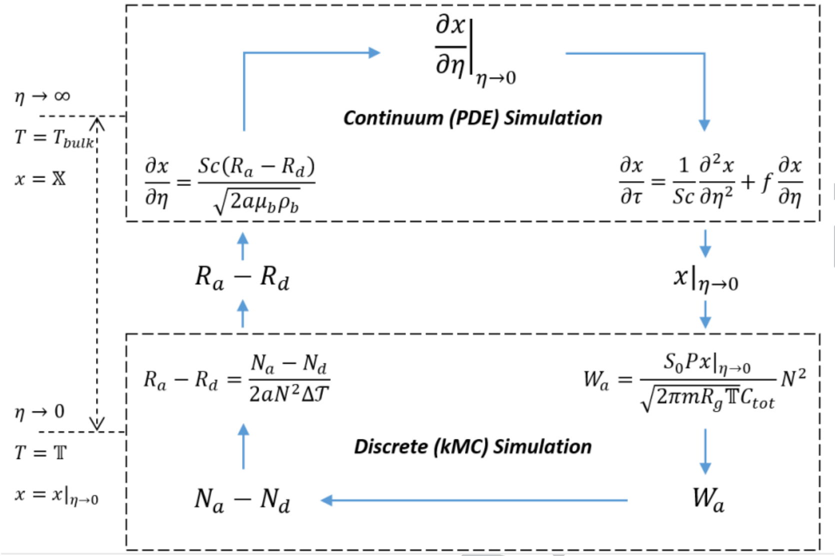
$\frac{\partial x}{\partial \eta}\Big|_{\eta\to 0}$
Continuum (PDE) Simulation
$\eta \to \infty$
$T = T_{bulk}$
$x = \mathbb{X}$
$\frac{\partial x}{\partial \eta} = \frac{Sc(R_a - R_d)}{\sqrt{2a\mu_b\rho_b}}$
$\frac{\partial x}{\partial \tau} = \frac{1}{Sc}\frac{\partial^2 x}{\partial \eta^2} + f\frac{\partial x}{\partial \eta}$
$R_a - R_d$
$x|_{\eta\to 0}$
$R_a - R_d = \frac{N_a - N_d}{2aN^2\Delta\mathcal{T}}$
$W_a = \frac{S_0 P x|_{\eta\to 0}}{\sqrt{2\pi m R_g \mathbb{T} C_{tot}}} N^2$
$\eta \to 0$
$T = \mathbb{T}$
$x = x|_{\eta\to 0}$
Discrete (kMC) Simulation
$N_a - N_d$
W_a

Figure 1: The simulation framework for the stochastic multiscale model of thin film formation by chemical vapour deposition

In this stochastic multiscale model, the variables that can be manipulated are the inlet precursor mole fraction $\mathbb{X}$ of boundary condition (6) and the substrate temperature $\mathbb{T}$ of boundary condition (7). The observables of this multiscale model are roughness, R, growth rate, Gr, and thickness, Th. These properties can be calculated from the following equations (Christofides et al., 2009; Raimondeau & Vlachos, 2000):

$$R = 1 + \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\left(|h_{i+1,j} - h_{i,j}| + |h_{i-1,j} - h_{i,j}| + |h_{i,j+1} - h_{i,j}| + |h_{i,j-1} - h_{i,j}|\right)}{2N^2} \tag{14}$$

$$Gr = \frac{N_a - N_d}{N^2 \Delta\mathcal{T}} \tag{15}$$

$$Th = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} h_{i,j} \tag{16}$$

where $h_{i,j}$ is the number of chemical vapour particles adsorbed at a lattice site (i,j) and N is the size of the square lattice (hence, N^2 is the total number of sites on the lattice). Roughness is considered as the average number of broken bonds on the surface of the thin film (Christofides et al., 2009; Raimondeau & Vlachos, 2000). The units of R and Th are monolayers (ML), while the units of Gr are ML/s (Luo, Heun, Kennedy, Wollschläger, & Henzler, 1994). Note that a perfectly smooth film would have R = 1.0 ML. Also note that Gr and Th are related, where the former is the time derivative of the latter, since the term $(N_a - N_d)/N^2$ represents the change in Th over a coupling time interval during which the kMC model has been simulating adsorption, desorption and migration events, and $\Delta\mathcal{T}$ is the duration of the coupling interval (measured in seconds).

2.2 Nonlinear model predictive control (NMPC) framework

For the semiconductor industry, it is important to produce films that are smooth and sufficiently thick, since both properties contribute to electrical conductivity. For the purposes of optical and biomedical coatings, low roughness and sufficient thickness are likewise significant: smooth films would minimize light scattering or prevent bacteria from adhering to the surface, while sufficient thickness would control the transmission of light through the optical device or ensure a sufficient lifetime for the biomedical device before the film wears off. Furthermore, high Gr would contribute to the profitability of the manufacturer by satisfying customer demand in a timely manner. However, achieving low R values and high Gr and Th values requires optimal control actions in the manipulated variables (the inlet precursor mole fraction $\mathbb{X}$ and the substrate temperature $\mathbb{T}$).

Since the production of thin films is a batch process with nonlinear relationships between the variables, it is amenable to nonlinear model predictive control (NMPC). In a closed-loop control framework the measurements would become available as the time until the end of the batch continues to diminish. Hence, the shrinking horizon optimization scheme (Joseph & Hanratty, 1993) is best suited for finding the optimal profiles in $\mathbb{T}$ and $\mathbb{X}$ that would satisfy the final product requirements. In the shrinking horizon optimization approach, the goal is to drive the plant to some desired target while using the measurements from the plant to monitor the performance and adjust the optimal control actions as necessary. The optimization is iterative and requires a model to predict the plant's responses. In each iteration, only the first control action from the optimal profile is implemented on the actual plant. When a measurement from the plant becomes available, it is used as the initial condition for the next iteration of shrinking horizon optimization, where a new set of optimal control actions will be found. The optimization scheme proceeds until the end of the production batch is reached.

Equation (17) describes the formulation of the shrinking horizon optimization problem that would employ the coupled stochastic multiscale kMC-PDE model to find the optimal time-dependent $\mathbb{T}$ and $\mathbb{X}$ profiles that would maximize the average Gr during the production batch (and, consequently, the total Th) while satisfying the maximum allowed roughness at the end of the batch:

$$\max_{\mathbb{T}(t_i),\mathbb{X}(t_i)} \quad \frac{1}{n}\sum_{i=1}^{n} Gr\big(\mathbb{T}(t_i),\mathbb{X}(t_i)\big)$$

Subject to:

Coupled kMC-PDE model, equations (1)-(16)

$$t_i = i\Delta t; \quad \forall i = 0,1,2,\dots n$$

$$R\big(\mathbb{T}(t_n),\mathbb{X}(t_n)\big) \le R^* \qquad (17)$$

$$\mathbb{T}_{min} \le \mathbb{T}(t_i) \le \mathbb{T}_{max}$$

$$\mathbb{X}_{min} \le \mathbb{X}(t_i) \le \mathbb{X}_{max}$$

$$\mathbb{T}(0) = \mathbb{T}_0$$

$$\mathbb{X}(0) = \mathbb{X}_0$$

$$-\Delta\mathbb{T} \le \mathbb{T}(t_{i+1}) - \mathbb{T}(t_i) \le \Delta\mathbb{T}$$

$$-\Delta \mathbb{X} \leq \mathbb{X}(t_{i+1}) - \mathbb{X}(t_i) \leq \Delta \mathbb{X}$$

where t_i represents the timepoints at which the observables are measured (i.e. at the end of every sampling interval), n is the total number of sampling intervals, Δt is the length of the sampling intervals, and R^* is the maximum allowed roughness constraint at the end of the batch. $\mathbb{T}_{min}$ and $\mathbb{T}_{max}$ are the minimum and maximum allowed temperature values, respectively, and $\mathbb{X}_{min}$ and $\mathbb{X}_{max}$ are the minimum and maximum allowed precursor mole fraction values, respectively. The settings $\mathbb{T}_0$ and $\mathbb{X}_0$ are the initial conditions for the manipulated temperature and precursor mole fraction, respectively, while $\Delta\mathbb{T}$ and $\Delta\mathbb{X}$ are the maximum allowed changes in the respective manipulated variables between adjacent sampling intervals.

Note that the shrinking horizon NMPC framework has to find the optimal control actions during the operation of the plant (i.e. online). Hence, having a computationally efficient method of identifying the optimal control action profiles is essential. However, due to the presence of non-closed-form calculations at the discrete scale, the full kMC-PDE model comprised of equations (1)-(16) is computationally intensive, and its application in equation (17) is computationally taxing. Thus, to make the problem of equation (17) computationally feasible, we trained the ANN models to predict the response variables $R(\mathbb{T}(t_i),\mathbb{X}(t_i))$ and $Gr(\mathbb{T}(t_i),\mathbb{X}(t_i))$ and replaced the full kMC-PDE model with the ANNs. The training procedure and the structure of the ANNs are discussed in section 3.

3. Artificial neural networks (ANNs)

3.1 Overview of ANNs

As briefly mentioned in the introduction, artificial neural networks (ANNs) are information processing systems that are data-driven and can identify complex relationships between the controlled and the response variables (Fausett, 1993; Hornik, 1991; McCulloch & Pitts, 1943; V. Venkatasubramanian & Vaidyanathan, 1992; Venkat Venkatasubramanian & Chan, 1989). ANNs consist of an input layer, one or several hidden layers, and an output layer. In each ANN layer, there are one or multiple processing units called “neurons”. The input layer serves to perceive the incoming variable data (e.g. control actions of a process) and pass it to the other layers in the network. In the hidden layer(s), the information is processed, and the output layer eventually produces a prediction of the response variable(s). The processing of information by the hidden and output layers is done as follows: each neuron accepts and combines the outputs from all the neurons in the preceding layer, where the combination is a weighted sum of each neuron’s output scaled by the corresponding strength of the connection (referred to as *weight*) between the neuron that is accepting the information and each neuron in the preceding layer. The output of a neuron in a hidden layer or in the output layer of an ANN may be represented as follows:

$$\chi_\ell = \varphi(\Sigma_\wp(\omega_{\ell,\wp}\chi_\wp) + \beta_\ell) \qquad (18)$$

where χ_ℓ is the output of neuron ℓ in the layer of interest, φ is the transfer function associated with the neuron ℓ (this function can also be called the *activation* or *output* function and it acts on $(\Sigma_\wp(\omega_{\ell,\wp}\chi_\wp) + \beta_\ell)$, i.e. the net input to the neuron ℓ), $\omega_{\ell,\wp}$ is the weight associated with the connection between the neuron ℓ and some neuron $\wp$ in the previous layer (i.e. the input layer or a hidden layer), $\chi_\wp$ is the output of the neuron $\wp$, $\Sigma_\wp$ signifies the weighted sum of the outputs of all neurons $\wp$ in the previous layer, and β_ℓ denotes the bias that is independent from the previous layer’s output but nevertheless contributes to the activation of the neuron ℓ in addition to the weighted sum $\Sigma_\wp(\omega_{\ell,\wp}\chi_\wp)$. Common choices for the neuron activation functions include the simple linear function, also called the *identity* function, the binary step function (i.e. the Heaviside function), and the sigmoid functions, such as the hyperbolic tangent function (Fausett, 1993; McCulloch & Pitts, 1943).

The values of the weights $\omega_{\ell,\wp}$ and the biases β_ℓ of an ANN are identified and refined during the iterative training on the input and output datasets. The training method of choice for ANNs with at least one hidden layer is called *backpropagation*. It is a gradient-based technique that aims to find $\omega_{\ell,\wp}$ and β_ℓ that would minimize the objective function, which is usually defined as the sum of squared errors of the network's predictions against the actual magnitudes of the response variables. The iterative training continues until the values converge to a local optimum (Svozil et al., 1997). It has been shown that the Levenberg-Marquardt algorithm, a second-order optimization method, can arrive at one of the lowest errors between the actual and predicted responses (Hagan & Menhaj, 1994; Levenberg, 1944; Marquardt, 1963; Svozil et al., 1997). Hence, this algorithm is frequently chosen for ANN training.

It is necessary to find the weights and biases that would allow the ANN to generalise, i.e. make reasonably accurate predictions about the input data that was not seen during training but is similar to what was used for ANN development. In order to achieve good generalisation capabilities instead of finding $\omega_{\ell,\wp}$ and β_{ℓ} that work well only for the seen-before data, a technique called *early stopping* is used for ANN training (Borggaard & Thodberg, 1992; Svozil et al., 1997). In this approach, a test dataset that is never used for training/adjusting the weights and biases is employed to calculate the generalisation error, i.e. the error between the ANN predictions of responses to the inputs never used during training and the actual responses to these inputs. As the iterative training proceeds, the generalisation error on the test dataset is monitored and when it starts to increase, the training is terminated because the growth of this error indicates the commencement of the memorization of the training dataset.

Just as a biological neural network (e.g. the brain) needs exposure to large amounts of information to learn to identify patterns accurately (e.g. in some cases, the expertise has to be acquired over a lifetime of human experience), ANNs also require large datasets for accurate weight and bias training. The need for the large datasets constitutes a disadvantage of the ANN-based approach to system identification, since such datasets may not be readily available for chemical engineering systems (Venkat Venkatasubramanian, 2019). This challenge can be circumvented when a mathematical model of the process is available, which is the case in the present work. Another possible disadvantage of using ANNs is the potential need for multiple training attempts, since the randomly chosen initial guesses for the weights and biases can have a strong impact on the trained network's prediction accuracy. In some cases, poor initial guesses may prevent the convergence in weights and biases altogether. In addition, the existence of multiple local optima in the non-convex objective function (Gandhi, Joshi, Jayaraman, & Kulkarni, 2007; Svozil et al., 1997) may complicate the decision-making process as to which of the possible acceptable ANNs should actually be used to model the system's behaviour. Furthermore, using large datasets for training, using complicated network structures with many neurons and hidden layers, and potentially repeating the training attempts all contribute to lengthy training times and increase the computational cost of ANN identification.

However, the compelling advantages of the ANN-based approach to system identification and modelling that compensate for the possible drawbacks are the computational efficiency of ANNs, as well as their ability to describe nonlinear relationships between inputs/parameters and responses, their potential to generalise and predict responses to never-before-seen input data, and their robustness towards noisy inputs (Svozil et al., 1997).

3.2 Training of ANNs for stochastic multiscale systems

In order to train the ANNs, we used the full multiscale model (with the lattice size N = 50) to generate a dataset of 400,000 points (one point per second), where each point consisted of the manipulated variables (the substrate temperature $\mathbb{T}$ and the inlet precursor mole fraction $\mathbb{X}$), and the corresponding outputs that are needed to evaluate the objective function and constraints considered in the shrinking horizon optimization formulation (17), i.e. roughness R and growth rate Gr. Note that other thin film characteristics, such as porosity or radial distribution function, could also be used to train ANNs via a methodology similar to the one presented here, but those metrics were beyond the scope of this work and thus were not calculated by the multiscale model.

The temperature was constrained to the range between 800 K and 1200 K since it was representative of the industrial thin film production, whereas the maximum allowed step in $\mathbb{T}$ was 50 K (to avoid unrealistically high temperature transitions/thermal shock). For similar reasons, the precursor mole fraction was constrained to the range between 1×10^{-6} and 4×10^{-6} with the maximum allowed change in $\mathbb{X}$ being 5×10^{-7}. The values of $\mathbb{T}$ and $\mathbb{X}$ were held constant for at least 25 seconds. The minimum time interval was chosen as 25 seconds to enable the multiscale model to reach steady state in the observables R and Gr before the next change in the manipulated variables would occur. The purpose of generating such a large dataset was to ensure that the ANNs would have enough information to obtain weights and biases that would make generalized predictions (i.e. interpolate accurately on the respective ranges of the manipulated variables rather than memorize the responses to the training data). Furthermore, the large dataset was intended to enable the ANN training algorithm to average out the effects of stochastic noise in the observables. The CPU time required to simulate the full multiscale model with the lattice size N = 50 for 400,000 seconds was approximately 72 hours. Note that all calculations in this work were performed on a computer with 16 GB of RAM and an Intel® Xeon® E5-2620 v4 processor running at 2.10 GHz.

We trained multiple-input single-output nonlinear autoregressive ANNs that accepted external input and, upon training completion, were able to forecast the response variables based on the current inputs (i.e. the current values of $\mathbb{T}$ and $\mathbb{X}$) and their own past predictions (i.e. the past predictions of R and Gr were supplied to the respective ANN via closed-loop feedback). The dataset of 400,000 points was randomly subdivided into training, validation and test subsets: 70%, 15% and 15% of the 400,000 points were assigned to each respective subset. The training subset was used for fitting the weights and biases of the ANNs (via the Levenberg-Marquardt backpropagation with a mean square error objective function), the validation subset was used to evaluate the ANN models' quality of fit and adjust the weights and biases, and the test subset was used only to decide when to stop training to prevent the memorization of data.

As per common practice, the hyperbolic tangent sigmoid transfer function was used for the neurons in the hidden layer(s) of the ANNs, and a linear transfer function was utilized for the neurons in the output layer of the ANNs. The above procedure was implemented using the Neural Net Time Series toolbox in MATLAB.

To accelerate the completion of ANN training and to mitigate the potentially detrimental effect of poor initial guesses for weights and biases on convergence to reasonably accurate ANNs, the training was carried out in the open loop mode; the feedback loop in each ANN was closed only after the training was completed. In other words, during training, the ANNs did not rely on their own past predictions of the observables to make the new predictions – instead, the known observable values at the previous timestep were used for the ANNs to make predictions for the current timestep. After the training was completed, the feedback loop within the ANNs was closed so that the only external inputs required by the ANNs were the manipulated variables $\mathbb{T}$ and $\mathbb{X}$, whereas the networks' prior predictions of the observables were used for the future predictions.

To identify the networks that can potentially generate acceptable predictions, 9 ANNs for each of the two observables were trained, with network architectures that had one to three hidden layers, and two, five and ten neurons per hidden layer. Each network was trained on the same dataset with 400,000 points (described above). Next, the performance of the networks was assessed on a new dataset with 1,600,000 points that was never before seen by the networks. This dataset was obtained after approximately 260 hours of simulating the full multiscale model, and it was generated using the same restrictions on $\mathbb{T}$ and $\mathbb{X}$ as for the 400,000-point dataset. The new dataset was larger than the training dataset to confirm that the trained ANNs could predict the responses to previously unseen inputs accurately. Each of the 9 proposed networks for each of the two observables calculated the responses to the 1,600,000 manipulated variable values in 55 ± 0.5 seconds, regardless of the network's structure. The statistics of the percent relative errors between the predicted responses of R and Gr and the 1,600,000 measured values have been summarized in Tables 1 and 2, respectively. Similarly, the probability distributions of the errors have been plotted in Figures 2 and 3, respectively. Note that in these Tables and Figures, different network structures have been specified using the number of hidden layers and the number of neurons in each layer; for example, the ANN with 1 hidden layer and 2 neurons in the layer was referred to as *1 layer, 2 neurons* while the ANN with 3 hidden layers and 10 neurons per layer was called *3 layers, 10 neurons*.

Table 1: Statistics on the percent relative errors between 1,600,000 predicted and actual roughness (R) values

	Statistics on the percent relative errors						
ANN structure	min	max	median	mean	std. dev.	variance	skewness
1 layer, 2 neurons	-53.73	45.56	0.57	0.42	7.67	58.78	-0.41
2 layers, 2 neurons	-54.91	49.61	0.45	0.51	7.02	49.25	-0.52
3 layers, 2 neurons	-49.75	45.29	0.91	0.95	7.59	57.59	-0.44
1 layer, 5 neurons	-55.47	45.57	0.25	0.06	6.55	42.86	-0.77
2 layers, 5 neurons	-55.21	97.17	0.20	0.06	6.46	41.72	-0.33
3 layers, 5 neurons	-54.94	210.72	0.21	0.05	6.53	42.63	0.89
1 layer, 10 neurons	-54.99	144.64	0.21	0.04	6.51	42.38	-0.14
2 layers, 10 neurons	-53.80	86.83	0.32	0.19	6.38	40.76	-0.38
3 layers, 10 neurons	-53.50	1097.68	0.22	2.09	40.08	1606.59	19.20

Table 2: Statistics on the percent relative errors between 1,600,000 predicted and actual growth rate (Gr) values

	Statistics on the percent relative errors						
ANN structure	min	max	median	mean	std. dev.	variance	skewness
1 layer, 2 neurons	-23.25	44.62	0.02	0.09	2.46	6.07	1.26
2 layers, 2 neurons	-23.15	44.43	0.01	0.07	2.44	5.95	1.23
3 layers, 2 neurons	-23.06	44.49	0.02	0.08	2.44	5.95	1.23
1 layer, 5 neurons	-21.63	41.32	0.01	0.07	2.42	5.88	1.17
2 layers, 5 neurons	-19.53	39.50	0.01	0.06	2.31	5.34	0.90
3 layers, 5 neurons	-20.92	41.25	0.01	0.07	2.42	5.85	1.08
1 layer, 10 neurons	-22.10	43.36	0.00	0.06	2.43	5.91	1.22
2 layers, 10 neurons	-20.69	41.88	-0.02	0.03	2.33	5.43	1.12
3 layers, 10 neurons	-20.49	39.62	0.02	0.07	2.33	5.44	1.03

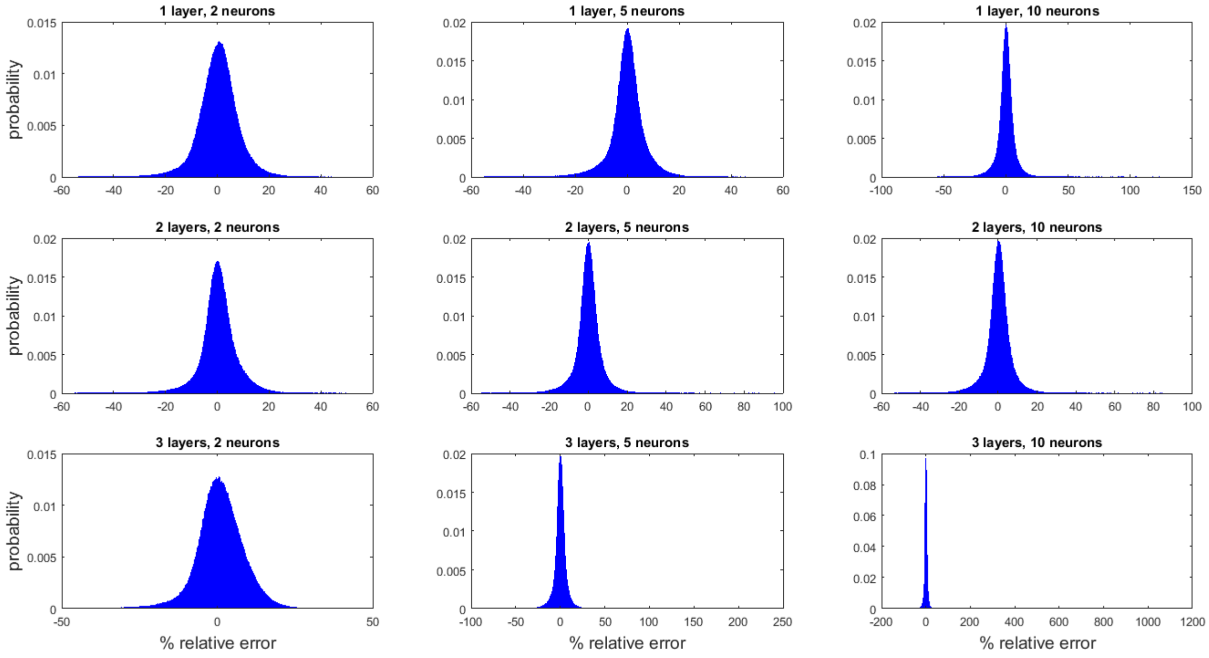

Figure 2: Probability distributions of the percent relative errors between 1,600,000 predicted and actual roughness (R) responses

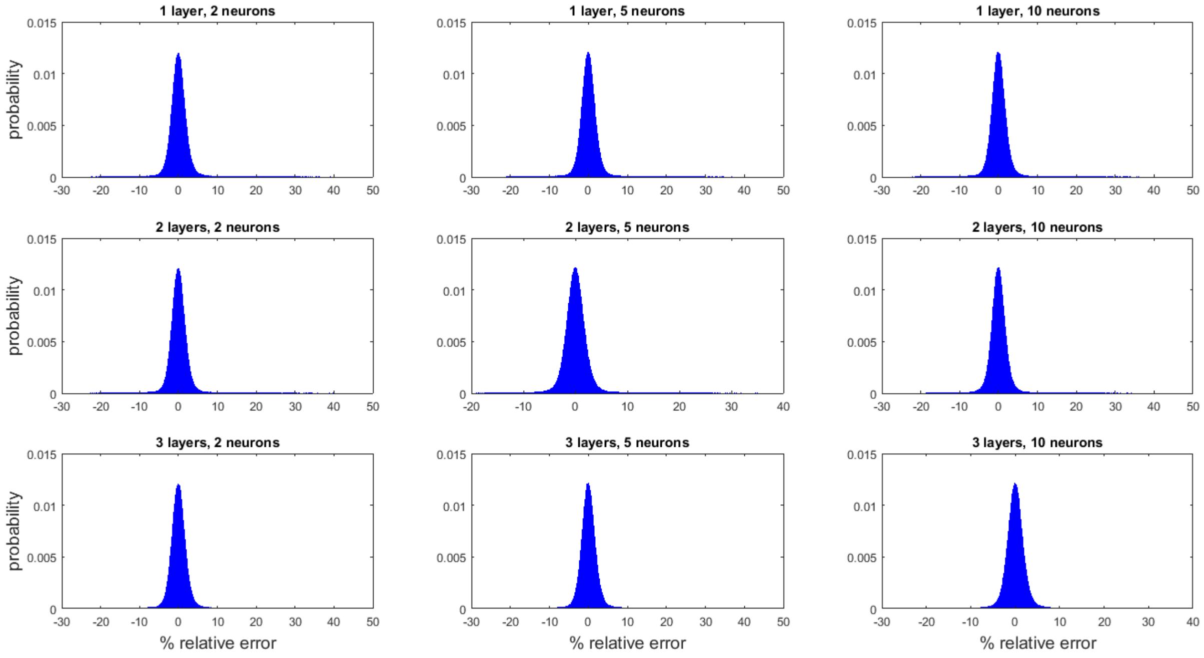

Figure 3: Probability distributions of the percent relative errors between 1,600,000 predicted and actual growth rate (Gr) responses

Based on the data presented in Table 1 and Figure 2, the ANN with 1 hidden layer and 5 neurons was chosen to predict roughness (R). It can be seen in Table 1 that the lowest mean and standard deviation of relative percent errors were demonstrated by all networks with 5 neurons per hidden layer and the network with 1 hidden layer and 10 neurons (e.g. the mean errors were between 0.04% and 0.06% for these ANNs). However, among these four networks, all the ANNs larger than the network with 1 hidden layer and 5 neurons demonstrated relatively high maximum possible percent errors (from 86.83% for the network with 2 layers and 10 neurons per layer up to 1,097.68% for the network with 3 layers and 10 neurons per layer). The presence of such high possible maximum errors was also the reason why the corresponding probability distributions in Figure 2 appeared narrow and sharp.

While no significant differences could be observed from Figure 3 among the error distributions from the ANNs that predicted the growth rate (Gr) responses, Table 2 showed that the network with 2 hidden layers and 5 neurons in each layer demonstrated superior generalization abilities with the lowest possible minimum (-19.53%) and maximum (39.50%) relative percent errors, the second-lowest median (0.01%) and mean (0.06%) errors, and the lowest standard deviation (2.31%), variance (5.34) and skewness (0.90) of all the proposed ANN architectures. The corresponding architectures of the ANNs chosen to predict roughness and growth rate for the present multiscale model have been depicted in Figures 4 and 5, respectively.

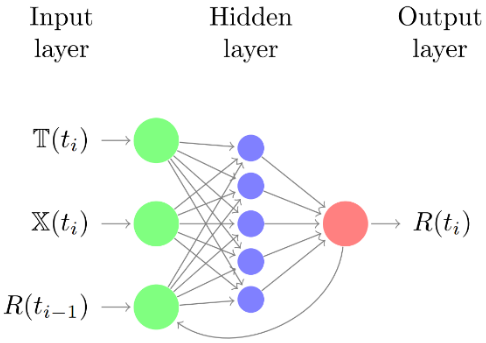

Figure 4: A schematic of the chosen "1 layer, 5 neurons" ANN structure for predicting roughness (R). Note that the output of the ANN is used for autoregression via the closed-loop feedback.

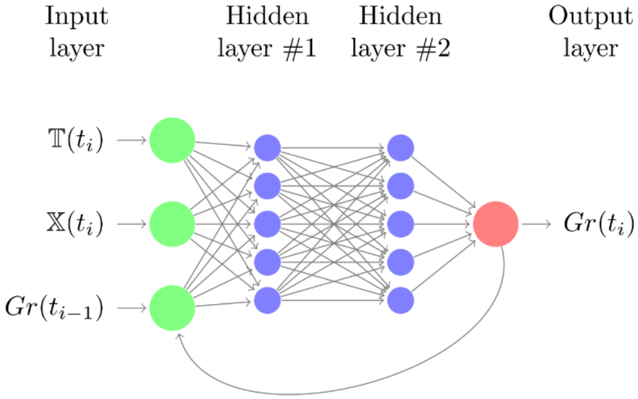

Figure 5: A schematic of the chosen "2 layers, 5 neurons" ANN structure for predicting growth rate (Gr). Note that the output of the ANN is used for autoregression via the closed-loop feedback.

4. Results and discussion

In this section, we present the results of applying the ANNs chosen in section 3.2 for the shrinking horizon optimization problem of equation (17) in section 2.2. The shrinking horizon optimization problem was solved using the second order gradient-based interior-point constrained optimization algorithm implemented in the *fmincon* function in MATLAB. For the interior point algorithm in *fmincon*, the constraint and optimality tolerances were 10^{-6} and the step tolerance was 10^{-10} (i.e. the default MATLAB values). Furthermore, to speed-up the calculations, *fmincon* was set to compute the gradients in parallel in all scenarios presented below. In equation (17), the objective function had no additional weights, and the batch time was set to range from 0 s to 450 s in order to allow the NMPC algorithm to sample an adequate number of decision variables while considering the possibility of having external and/or internal events that may impact the operation. Identical to the training dataset characteristics described in section 3.2, the temperature parameters were set as follows: $\mathbb{T}_{min}$ = 800 K, $\mathbb{T}_{max}$ = 1,200 K and $\Delta\mathbb{T}$ = 50 K. The precursor mole fraction parameters were also identical to the characteristics of the training dataset: $\mathbb{X}_{min}$ = 1×10^{-6}, $\mathbb{X}_{max}$ = 4×10^{-6} and $\Delta\mathbb{X}$ = 5×10^{-7}. Note that the initial conditions for the temperature and precursor mole fraction were set to $\mathbb{T}_0 = \mathbb{T}_{min}$ and $\mathbb{X}_0 = \mathbb{X}_{max}$, respectively; this was done to maximize the value of W_a in equation (11), i.e. the rate of adsorption of chemical vapour particles onto the substrate/thin film surface, thereby promoting thin film growth. Hence, these are reasonable settings for the start of the production batch that would aim to manufacture the thin films in a timely fashion. In the scenarios presented below, up to three final roughness targets (R^*) were considered: 1.8 ML, 3.0 ML and 5.0 ML. The first two targets were used as industrially relevant criteria for smooth high-performance thin films since an ideal film would have R = 1.0 ML, whereas the third value was proposed to test the performance of the ANN models for higher roughness targets subject to large stochastic noise. Note that in all scenarios, the manipulated variables were held piecewise constant for the duration of the corresponding sampling intervals Δt (25 s in Scenarios A-C and 10 s in Scenario D, see below).

4.1 Scenario A: No online measurements

In the first scenario, there was no feedback from the full multiscale model (i.e. the plant model) provided to the ANNs. Instead, the ANNs were used within the gradient-based constrained optimization algorithm (mentioned previously) with the same constraints as in equation (17) to find the optimal time-dependent control action profiles that would, according to the ANN predictions, satisfy the final roughness targets R^* of 1.8 ML, 3.0 ML and 5.0 ML. Once found, the profiles were used to run the plant model in the open loop batch mode to validate the solutions. This control scheme has been presented in Figure 6a, where $R_{kMC}(t_i)$ and $Gr_{kMC}(t_i)$ denote the values of the observables produced by the plant model, while R_{ANN} (t_i) and $Gr_{ANN}(t_i)$ represent the predictions of the ANN models. Note that the sampling interval Δt was set to 25 s, i.e. the manipulated variables $\mathbb{T}$ and $\mathbb{X}$ were allowed to be changed no sooner than every 25 s. Consequently, for the horizon of 450 s the total number of the sampling intervals, n, was 18. The results are presented in Table 3 and in Figures 7-9. Note that in Table 3, as well as in Tables 4-6 in sections 4.2-4.4, $R_{ANN}(t_n)$ refers to the roughness prediction by the ANN model at time t_n (i.e. the end of the optimization horizon), $R_{kMC}(t_n)$ refers to the final roughness measurement from the plant, $\overline{Gr_{ANN}}$ refers to the average of the growth rate values predicted by the corresponding ANN over the entire optimization horizon, and $\overline{Gr_{kMC}}$ refers to the average of the growth rate values collected from the plant model over the entire horizon.

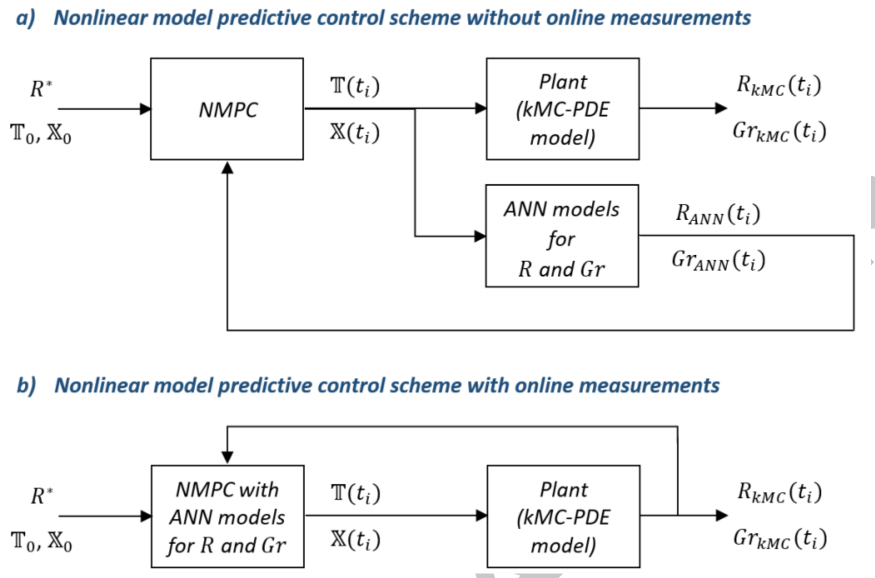

Figure 6: The open-loop (a) and closed-loop (b) shrinking horizon NMPC schemes employed in Scenarios A-D. Scenario A used the open-loop scheme (a), while Scenarios B-D used the closed-loop scheme (b).

Table 3: The performance of ANN models for the case of no online measurements and the sampling interval Δt = 25 s.

	$\mathbb{T}(t_n)$ (K)	$\mathbb{X}(t_n)$	$\overline{Gr_{ANN}}$ (ML/s)	$\overline{Gr_{kMC}}$ (ML/s)	$R_{ANN}(t_n)$ (ML)	$R_{kMC}(t_n)$ (ML)
R^* = 1.8 ML	1200	1.22×10^{-6}	31.66	31.59	1.80	1.83
R^* = 3.0 ML	1106	4.00×10^{-6}	36.37	36.24	3.00	3.00
R^* = 5.0 ML	1006	4.00×10^{-6}	36.76	36.65	5.00	7.60

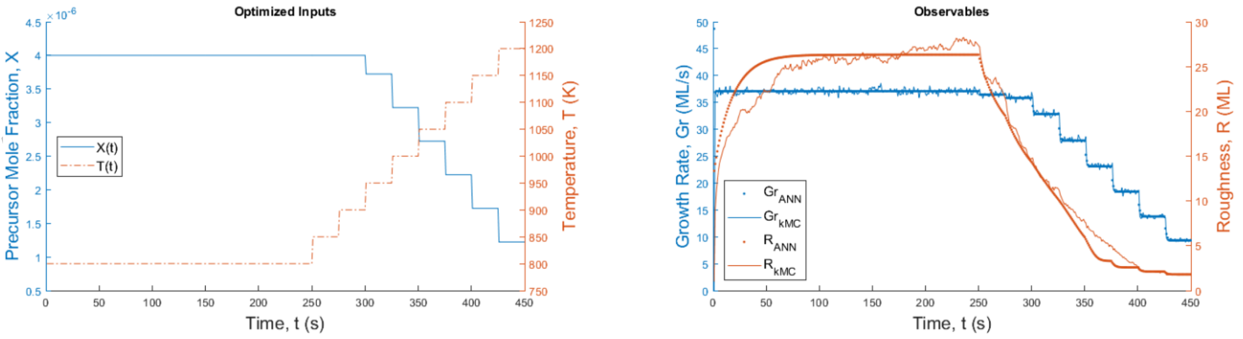

Figure 7: Optimal profiles of $\mathbb{T}$ and $\mathbb{X}$ and the responses in R and Gr, no online measurements, Δt = 25 s, R^* = 1.8 ML

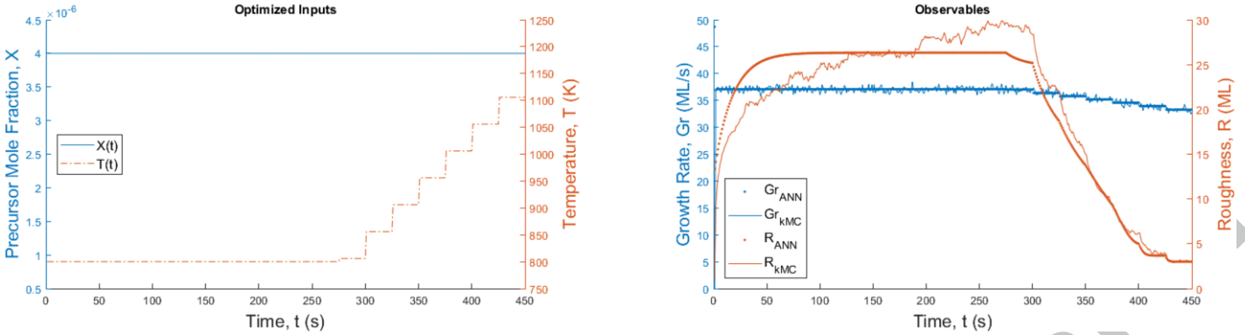

Figure 8: Optimal profiles of $\mathbb{T}$ *and* $\mathbb{X}$ *and the responses in* R *and* Gr*, no online measurements,* Δt *= 25 s,* R^* *= 3.0 ML*

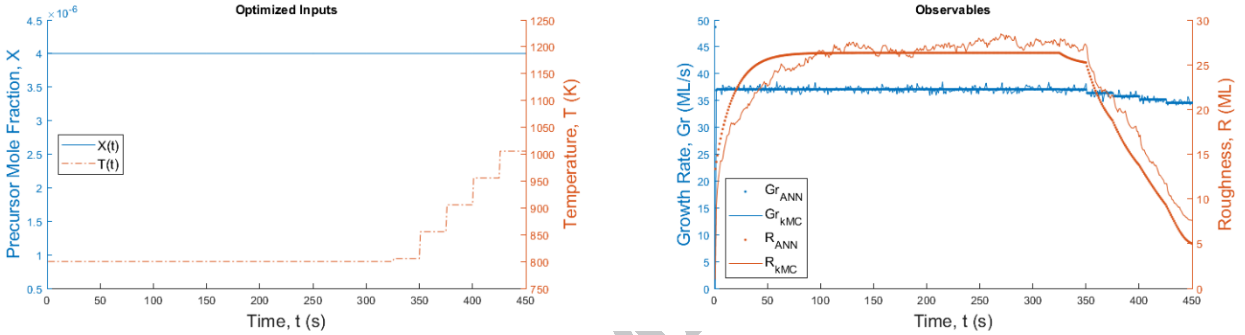

Figure 9: Optimal profiles of $\mathbb{T}$ *and* $\mathbb{X}$ *and the responses in* R *and* Gr*, no online measurements,* Δt *= 25 s,* R^* *= 5.0 ML*

Figures 7-9 show that different profiles of the manipulated variables have been obtained depending on the three target roughness values R^* (1.8 ML, 3.0 ML and 5.0 ML). A close overlap of the final roughness trajectories (approximately the last 50 seconds of the batch) can be observed in Figures 7 (R^* = 1.8 ML) and 8 (R^* = 3.0 ML). Note that for R^* = 1.8 ML, the final roughness value from the multiscale model was 1.83 ML, slightly higher than the R^* target (see Table 3). This can be explained by the presence of stochastic noise in the full multiscale model – the noise is present even at 1,200 K. The most detrimental effect of the noise, as well as potentially model-plant mismatch, was observed for high roughness target values, i.e. R^* = 5.0 ML, where the ANN for the roughness significantly underestimated $R_{kMC}(t_n)$ at 1,006 K: the $R_{kMC}(t_n)$ value was 52% greater than $R_{ANN}(t_n)$, i.e. 7.60 ML versus 5.00 ML, respectively. Note that for all R^* targets, the roughness ANN overestimated the rate of dynamic response of the system to the changes in the temperature. This is most apparent in Figure 9, where the ANN predictions of roughness reached steady state at 800 K before the multiscale model did (a similar effect is observed in Figures 7 and 8), and when the temperature started to increase at the sampling intervals $i \geq 14$ ($t \geq 326$ s), the response of R_{ANN} was much faster than of R_{kMC}. For the case of R^* = 5.0 ML, the poor accuracy of ANN predictions of roughness at the end of the batch could be attributed to the noisy training data (Svozil et al., 1997) at relatively low temperatures (under 1,050 K). During training, the weights and biases of the ANN for roughness likely converged to the values that captured the average responses of noisy low-temperature (e.g. under 1,050 K) roughness, resulting in model-plant mismatch and suboptimal ANN performance.

Moreover, the ANN for the growth rate predicted the responses accurately and, in all the scenarios in this work, it did not overestimate $\overline{Gr_{kMC}}$ by more than 0.4 % (this error occurred for R^* = 3.0 ML in Table 3, compare 36.37 ML/s from the ANN to 36.24 ML/s from the multiscale model). Note that in Table 3, all results for $\overline{Gr_{ANN}}$ slightly overestimated the corresponding $\overline{Gr_{kMC}}$ values, consistent with the fact that the ANN for growth rate with 2 hidden layers and 5 neurons per layer exhibited positive mean and median relative percent errors, as well as positive skewness of the errors (see the *2 layers, 5 neurons* results in Table 2).

Furthermore, the strength of the dependence of the observables on the manipulated variables can be qualitatively observed by comparing the optimized control inputs in Figures 7-9 to the corresponding time-dependent profiles of the observables. In Figures 8 and 9, only the substrate temperature $\mathbb{T}$ was manipulated in the attempt to achieve R^* = 3.0 ML and R^* = 5.0 ML, respectively. In both cases, roughness started to decrease as soon as the temperature began to increase, but the growth rate decreased only at the temperatures greater than or equal to 850 K, and it did so at a much lower rate than the roughness trajectory. Hence, in this multiscale system, temperature has a greater impact on roughness than growth rate, but it is related to both observables nonetheless. An opposite qualitative relationship appears to exist between the precursor mole fraction ($\mathbb{X}$) and the observables: comparing Figures 7 and 8, it can be observed that the major difference is the manipulation of $\mathbb{X}$ in Figure 7 (R^* = 1.8 ML). In the scenario of Figure 7, both $\mathbb{X}$ and $\mathbb{T}$ were manipulated, but the precursor mole fraction caused the majority of the change in the growth rate trajectory (the growth rate at the end of the batch was 9 ML/s in Figure 7, which was 3.7 times lower than 33 ML/s in Figure 8). Comparing Figure 8 to Figure 7, the final temperature was increased by less than 100 K, so the majority of the effect on the growth rate can be attributed to the precursor mole fraction manipulations. Therefore, both the roughness and the growth rate ANNs were able to capture the different strengths of connections between each manipulated variable and the corresponding observables, and the ANNs could adjust the manipulated variables depending on the final roughness constraint. Furthermore, the ANN models demonstrated superior computational efficiency. On average, the optimization was completed in 75 seconds after approximately 2,300 evaluations of the ANN models. However, the average computational time to evaluate the full kMC-PDE model using the optimal profiles of the manipulated variables was 580 seconds. In other words, if the full kMC-PDE model were to be used in the optimization scheme and evaluated 2,300 times, the total computational time would be almost 18,000 times longer than with the ANN models. Thus, the ANN models efficiently provided reasonable estimates and, as a result, were used for further studies.

4.2 Scenario B: Online measurements

The second scenario was the full shrinking horizon NMPC technique, where R and Gr measurements from the plant model (referred to as *online measurements* in Figures 6b and 10-12, as well as in Table 4) were collected in real time once at the end of every sampling interval (Δt = 25 s) and then used as the initial conditions for the respective ANNs in the next iteration of the shrinking horizon optimization. This is summarized in Figure 6b, where the feedback between the plant model and the shrinking horizon optimization is represented by the dotted line and the feedback between the plant and the ANN models is represented by the dashed line. Figures 10-12 and Table 4 summarize the results of the study.

Table 4: The performance of ANN models for the case with online measurements and Δt = 25 s.

	$\mathbb{T}(t_n)$ *(K)*	$\mathbb{X}(t_n)$	$\overline{Gr_{ANN}}$ *(ML/s)*	$\overline{Gr_{kMC}}$ *(ML/s)*	$R_{ANN}(t_n)$ *(ML)*	$R_{kMC}(t_n)$ *(ML)*
R^* = 1.8 ML	1200	1.22×10^{-6}	31.66	31.57	1.80	1.79
R^* = 3.0 ML	1106	4.00×10^{-6}	36.38	36.24	3.00	2.97
R^* = 5.0 ML	1006	4.00×10^{-6}	36.66	36.53	5.33	5.96

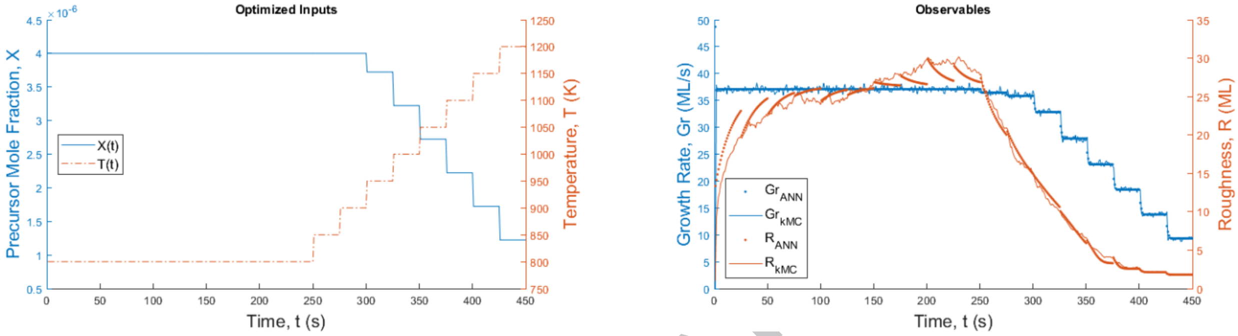

Figure 10: Optimal profiles of $\mathbb{T}$ and $\mathbb{X}$ and the responses in R and Gr, with online measurements, Δt = 25 s, R^* = 1.8 ML

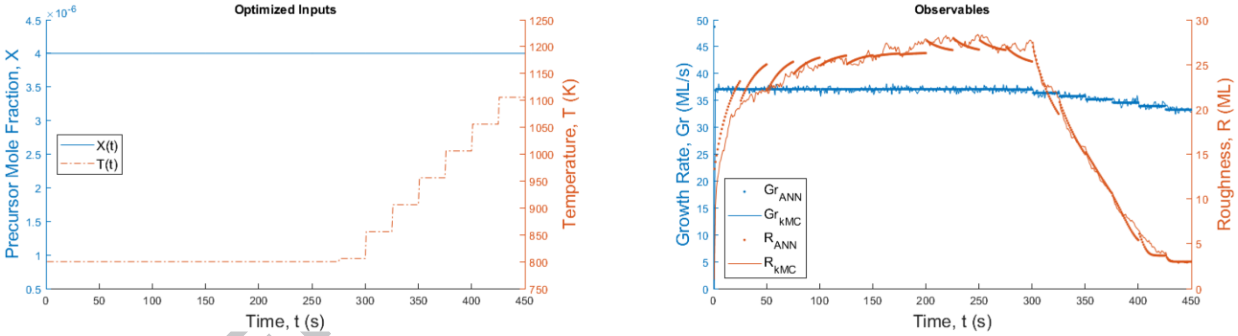

Figure 11: Optimal profiles of $\mathbb{T}$ and $\mathbb{X}$ and the responses in R and Gr, with online measurements, Δt = 25 s, R^* = 3.0 ML

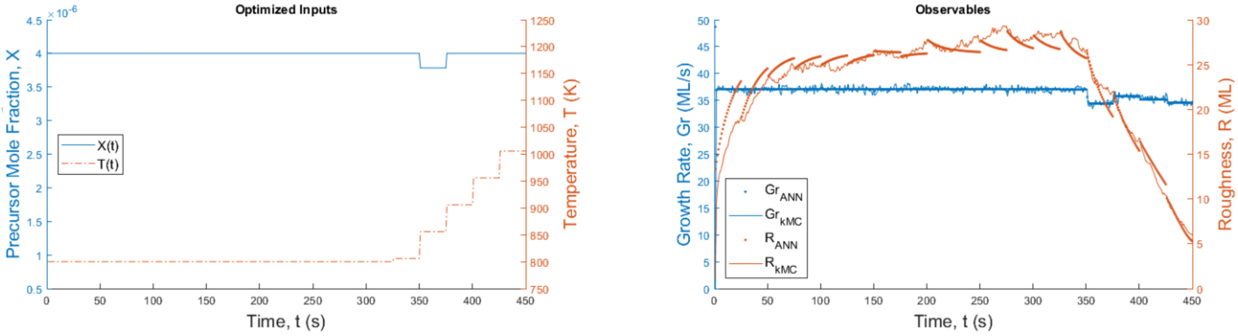

Figure 12: Optimal profiles of $\mathbb{T}$ and $\mathbb{X}$ and the responses in R and Gr, with online measurements, Δt = 25 s, R^* = 5.0 ML

Unlike in Figures 7-9, the time-varying profiles of R_{ANN} appear discontinuous because every 25 seconds the measurement from the multiscale model was provided as the initial condition for the neural network predictions. We observed close agreement between $R_{ANN}(t_n)$ and $R_{kMC}(t_n)$ values for R^* = 1.8 ML (1.80 ML vs 1.79 ML, respectively, see Table 4 and Figure 10) and R^* = 3.0 ML (3.00 ML vs 2.97 ML, respectively, see Table 4 and Figure 11). The growth rate predictions obtained using ANN and the stochastic multiscale model also exhibited good agreement. Furthermore, the final temperatures $\mathbb{T}(t_n)$ obtained for the case with online measurements were identical (rounded to nearest integer) to the case without the online measurement for each roughness target R^*. Since the targets R^* = 1.8 ML and R^* = 3.0 ML were satisfied by the plant, the ANNs were shown to provide highly accurate predictions for the low-noise regime of the multiscale model.

However, the target R^* = 5.0 ML (see Figure 12) was not satisfied, neither by the ANN model nor by the plant. The mismatch between ANN roughness predictions and the data from the plant was significant: $R_{ANN}(t_n)$ was 5.33 ML and $R_{kMC}(t_n)$ was 5.96 ML, as per Table 4. Unlike in the previous scenario, here the online measurements were supplied as the initial conditions for the ANN. While the temperature was set to 800 K (until 325 seconds of batch time, $i \leq$ 13), the ANN predictions were not able to detect an issue, expecting to meet the roughness constraint of 5.0 ML because the ANN overpredicted the rate of dynamic response of the model in the noisy regime. As depicted in Figure 12, as the temperature started to be increased (at 326 seconds of batch time, $i \geq$ 14), the initial conditions for the shrinking horizons started to be higher than previously anticipated by the ANN for roughness. Consequently, not only did the plant exceed the roughness constraint of 5.0 ML, but so did the ANN model, because the mismatch between the plant measurements and the ANN was too great for the ANN to overcome and reach the roughness target of 5.0 ML while abiding by the constraint on the maximum temperature increase between two successive sampling intervals (i.e. $\Delta\mathbb{T}$ = 50 K). Since the temperature ramp began late in the time horizon, there were not enough decisions left to continue to increase the temperature to meet the R^* = 5.0 ML requirement.

Note that for online optimization with embedded ANNs it may be necessary to use more efficient nonlinear programming and dynamic optimization software (e.g. DONLP2, IPOPT, etc.) instead of MATLAB's *fmincon* function, particularly if the computational resources are limited. It was observed that the shrinking horizon optimization problem's CPU time and the solution accuracy were sensitive to the solver type and the specified tolerances, as well as the accuracy of the initial guess. For example, with the tolerance of 10^{-6} (i.e. the default *fmincon* value), 75 seconds of CPU time were required initially and the CPU time for optimization decreased to less than 25 seconds only when there remained 7 (or less) decisions to be made on the shrinking horizon. However, with the first order optimality tolerance set to 10^{-1}, the optimization required approximately 15 seconds of CPU time to make the initial prediction of optimal control actions, but the average growth rate decreased by approximately 15% compared to the presented solutions. Furthermore, when the active set algorithm was used instead of interior point and the initial guess was close to the optimal solution, the CPU time at the start of optimization was less than 3 seconds and the manipulated variables' profiles were optimal. Therefore, the optimization tuning parameters will need to be adjusted based on the problem under consideration and the goals to be achieved by NMPC.

4.3 Scenario C: Disturbance rejection

Furthermore, we present a variation of the scenario B (section 4.2), where we studied the effect of a large disturbance in both manipulated variables. Using the same optimization procedure as in section 4.2, we added a disturbance not seen during ANN training. For the sampling interval i = 4, i.e. the interval that started at 76 s, disturbances to both the temperature and the precursor mole fraction were introduced and held constant for the entire interval; the disturbances were 3 times greater than the maximum allowed amount during data generation and training, i.e. the change in $\mathbb{T}$ was 150 K and the change in $\mathbb{X}$ was -1.5×10^{-6}. Note that online measurements from the plant continued to be collected and provided to the ANN models with the sampling interval of Δt = 25 s. Also, the manipulated variables were again held constant for the duration of each sampling interval. Since the online measurements of the observables R and Gr were used by the shrinking horizon optimization as initial conditions on every iteration, any disturbances to the observables were captured by the NMPC framework. After obtaining the measurements, the framework chose the control actions that best satisfied the objective function and the constraints of equation (17) while accounting for the measured R and Gr through the initial conditions. The results have been summarized in Table 5 and Figures 13-15.

Table 5: The performance of ANN models for the case with online measurements, disturbance at 76 s and Δt = 25 s.

	$\mathbb{T}(t_n)$ *(K)*	$\mathbb{X}(t_n)$	$\overline{Gr_{ANN}}$ *(ML/s)*	$\overline{Gr_{kMC}}$ *(ML/s)*	$R_{ANN}(t_n)$ *(ML)*	$R_{kMC}(t_n)$ *(ML)*
R^* *= 1.8 ML*	1200	1.22×10^{-6}	29.94	29.90	1.80	1.77
R^* *= 3.0 ML*	1106	4.00×10^{-6}	34.65	34.55	3.00	2.86
R^* *= 5.0 ML*	1006	4.00×10^{-6}	34.96	34.88	5.06	6.58

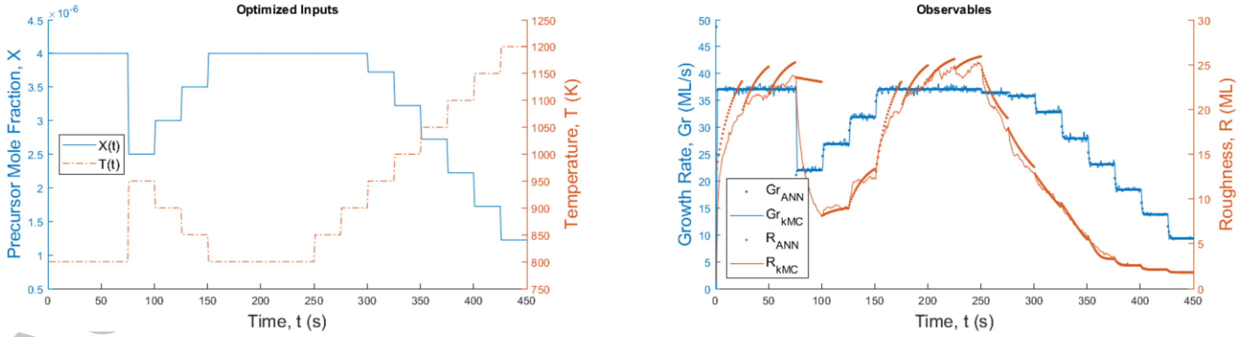

Figure 13: Optimal profiles and the responses, with online measurements and a disturbance, Δt = 25 s, R^* = 1.8 ML

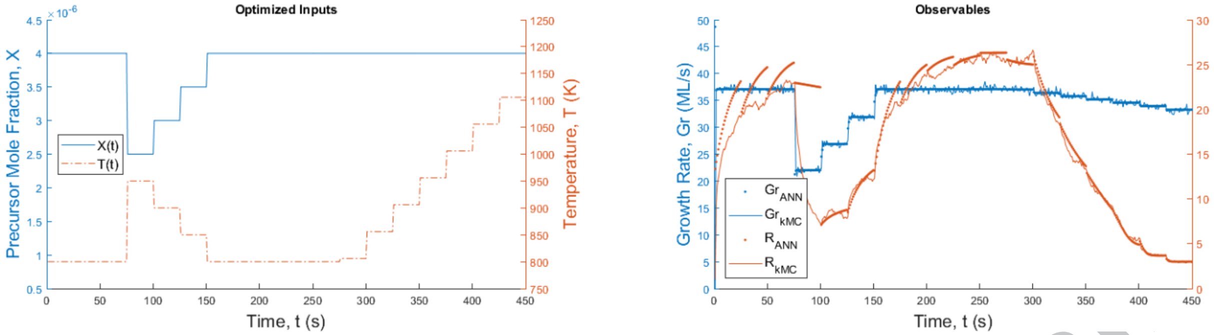

Figure 14: Optimal profiles and the responses, with online measurements and a disturbance, Δt = 25 s, R^* = 3.0 ML

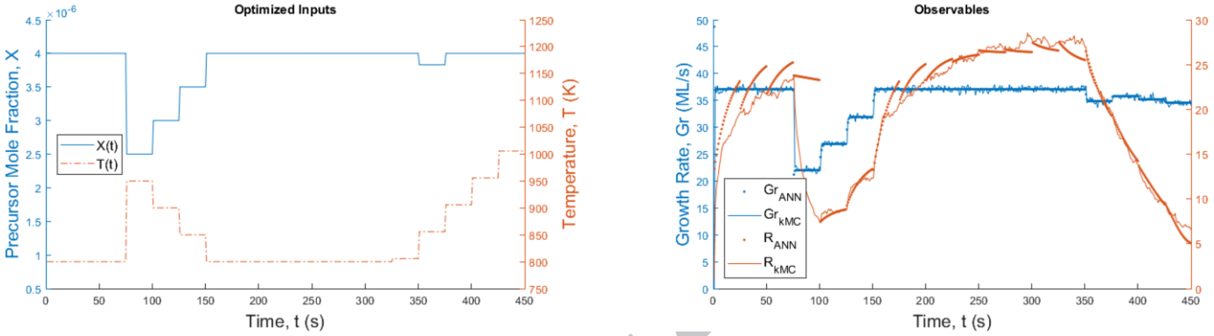

Figure 15: Optimal profiles and the responses, with online measurements and a disturbance, Δt = 25 s, R^* = 5.0 ML

It can be seen from the optimal profiles in Figures 13-15 that ANNs reacted to disturbances efficiently by decreasing the temperature and increasing the precursor mole fraction during the time from 101 seconds to 150 seconds (i.e. intervals i = 5 and 6) back to the original settings before manipulating the control inputs again later in order to meet the roughness target. This robustness to disturbances which were not seen during ANN models' development is a beneficial feature for online process control, where unexpected disturbances may significantly affect plant operations. However, due to poor performance in the noisy regime, the R^* = 5.0 ML target (see Figure 15) was not met neither by the ANN model for roughness (5.06 ML was the value of $R_{ANN}(t_n)$, as per Table 5), nor by the plant (6.58 ML was the measured $R_{kMC}(t_n)$, per Table 5).

4.4 Scenario D: Decreasing the sampling interval

To further investigate the ANN's behaviour at high roughness values, we attempted to mitigate the issue of ANN underprediction of the final measured roughness for the target R^* = 5 ML by decreasing the sampling interval Δt to 10 seconds instead of 25 seconds, making n = 45 for the total batch time of 450 s. Measurements from the plant were collected and provided to the ANN models every 10 seconds. The results have been provided in Table 6 and Figure 16.

Table 6: The performance of ANN models for the case with online measurements and Δt = 10 s. For brevity, only the results for R^* = 5.0 ML are reported.

	$\mathbb{T}(t_n)$ *(K)*	$\mathbb{X}(t_n)$	$\overline{Gr_{ANN}}$ *(ML/s)*	$\overline{Gr_{kMC}}$ *(ML/s)*	$R_{ANN}(t_n)$ *(ML)*	$R_{kMC}(t_n)$ *(ML)*
R^* *= 5.0 ML*	1054	4.00×10^{-6}	36.77	36.62	4.53	5.00

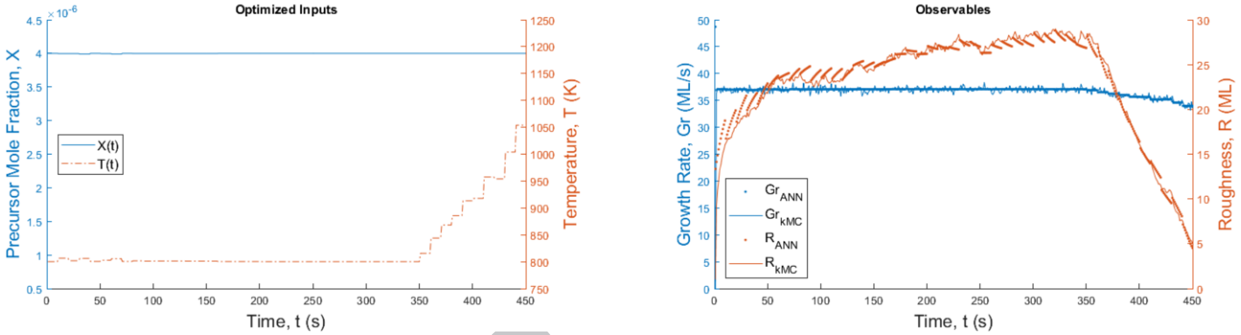

Figure 16: Optimal profiles and the responses, with online measurements, Δt = 10 s, R^* = 5.0 ML

Note that the training data was based only on Δt = 25 s. However, the ANNs are recursive: the future predictions depend only on the past predictions and the current manipulated variables, whereas the batch time is not an input to the ANNs. Hence, it is possible to use the ANN models for more frequent sampling. After decreasing the sampling interval, the ANN models tended to predict a faster response to temperature changes than what was actually observed from the full multiscale model. For this case, $T(t_n)$ was 1,054 K (as per Table 6), higher than 1,006 K obtained previously for the scenarios A-C with R^* = 5.0 ML. The obtained $R_{kMC}(t_n)$ was 5.00 ML (see Table 6 and Figure 16), and $R_{ANN}(t_n)$ was even lower (4.53 ML, see Table 6). Therefore, decreasing the sampling interval duration to Δt = 10 s was conducive to meeting the R^* = 5.0 ML constraint. While in this case the target roughness specification was met by the plant, it should be noted that the stochastic noise present at the temperatures of approximately 1,050 K (and below) may also produce roughness values greater than 5.0 ML. Also, it should be emphasized that R^* = 5.0 ML is a relatively high roughness value and that industrial applications require smoother films, for which situations the ANN models have demonstrated accurate predictions and reliable performance (see the results in sections 4.1-4.3). Note that lower roughness targets (1.8 ML and 3 ML) were tested for this case and satisfactory performance was observed, but the results are not shown here for brevity.

5. Conclusions

In this study, multiple-input single-output nonlinear autoregressive artificial neural networks (ANNs) with external input were developed to predict the responses of a stochastic multiscale chemical engineering system and subsequently conduct online shrinking horizon nonlinear model predictive control (NMPC). A model of thin film formation by chemical vapour deposition was used as the case study. The manipulated variables were the substrate temperature and the inlet precursor mole fraction, whereas the response variables of interest were the thin film roughness and growth rate. The ANN models captured system behaviour very accurately in the low-noise (i.e. high temperature) regimes and provided orders-of-magnitude computational savings over the full multiscale model. Furthermore, the ANN models provided good closed-loop performance since they were able to reject and compensate for the disturbances to the manipulated variables that were three times greater than what was seen in the ANN training data. The recommendations for future work include training the ANNs under parametric uncertainty and conducting moving horizon estimation as well as parameter and state estimation using the ANN models.

Acknowledgements

The authors would like to gratefully acknowledge the Natural Sciences and Engineering Research Council of Canada (NSERC) for the financial support.

References

Adomaitis, R. A. (2003). A Reduced-Basis Discretization Method for Chemical Vapor Deposition Reactor Simulation. *Mathematical and Computer Modelling*, *38*(1–2), 159–175.

Adomaitis, Raymond A. (2010a). Development of a multiscale model for an atomic layer deposition process. *Journal of Crystal Growth*, *312*(8), 1449–1452. https://doi.org/10.1016/j.jcrysgro.2009.12.041

Adomaitis, Raymond A. (2010b). Multiscale modeling and optimization of an atomic layer deposition process for nanomanufacturing applications. In *IFAC Proceedings Volumes* (Vol. 43, pp. 859–864). Elsevier. https://doi.org/10.3182/20100705-3-BE-2011.00142

Akkisetty, P. K., Lee, U., Reklaitis, G. V., & Venkatasubramanian, V. (2010). Population balance model-based hybrid neural network for a pharmaceutical milling process. *Journal of Pharmaceutical Innovation*, *5*(4), 161–168. https://doi.org/10.1007/s12247-010-9090-2

Alexandridis, A., & Sarimveis, H. (2005). Nonlinear adaptive model predictive control based on self-correcting neural network models. *AIChE Journal*, *51*(9), 2495–2506. https://doi.org/10.1002/aic.10505

Antanasijevic, D., Pocajt, V., Peric-Grujic, A., & Ristic, M. (2014). Modelling of dissolved oxygen in the Danube River using artificial neural networks and Monte Carlo Simulation uncertainty analysis. *Journal of Hydrology*, *519*(1), 1895–1907. https://doi.org/10.1016/j.jhydrol.2014.10.009

Bach-Andersen, M., Rømer-Odgaard, B., & Winther, O. (2018). Deep learning for automated drivetrain fault detection. *Wind Energy*, *21*(1), 29–41. https://doi.org/10.1002/we.2142

Bashir, Z. A., & El-Hawary, M. E. (2009). Applying wavelets to short-term load forecasting using PSO-based neural networks. *IEEE Transactions on Power Systems*, *24*(1), 20–27. https://doi.org/10.1109/TPWRS.2008.2008606

Bird, R. B., Stewart, W. E., & Lightfoot, E. N. (2002). *Transport Phenomena* (2nd ed.). John Wiley & Sons.

Borggaard, C., & Thodberg, H. H. (1992). Optimal minimal neural interpretation of spectra. *Analytical Chemistry*, *64*(5), 545–551. https://doi.org/10.1021/ac00029a018

Chaffart, D., Rasoulian, S., & Ricardez-Sandoval, L. A. (2016). Distributional Uncertainty Analysis and Robust Optimization in Spatially Heterogeneous Multiscale Process Systems. *AIChE Journal*, *62*(7), 2374–2390. Retrieved from https://doi.org/10.1002/aic.15215

Chaffart, D., & Ricardez-Sandoval, L. A. (2018). Optimization and Control of a Thin Film Growth Process: A Hybrid First Principles/Artificial Neural Network Based Multiscale Modelling Approach. *Computers and Chemical Engineering*, *119*, 465–479. https://doi.org/10.1016/j.compchemeng.2018.08.029

Cheimarios, N., Kokkoris, G., & Boudouvis, A. G. (2010). Multiscale modeling in chemical vapor deposition processes: Coupling reactor scale with feature scale computations. *Chemical Engineering Science*, *65*(17), 5018–5028. https://doi.org/10.1016/j.ces.2010.06.004

Chitsazan, N., Nadiri, A. A., & Tsai, F. T.-C. (2015). Prediction and structural uncertainty analyses of artificial neural networks using hierarchical Bayesian model averaging. *Journal of Hydrology*, *528*, 52–62. https://doi.org/10.1016/j.jhydrol.2015.06.007

Chow, T. W. S., & Leung, C. T. (1996). Neural network based short-term load forecasting using weather compensation. *IEEE Transactions on Power Systems*, *11*(4), 1736–1742. https://doi.org/10.1109/59.544636

Christofides, P. D., & Armaou, A. (2006). Control and optimization of multiscale process systems. *Computers and Chemical Engineering*, *30*(10–12), 1670–1686.

Christofides, P. D., Armaou, A., Lou, Y., & Varshney, A. (2009). *Control and Optimization of Multiscale Process Systems*. (W. S. Levine, Ed.) (1st ed.). New York, NY: Birkhäuser Boston.

Crose, M., Kwon, J. S.-I., Tran, A., & Christofides, P. D. (2017). Multiscale modeling and run-to-run control of PECVD of thin film solar cells. *Renewable Energy*, *100*, 129–140. https://doi.org/10.1016/j.renene.2016.06.065

Crose, M., Sang-Il Kwon, J., Nayhouse, M., Ni, D., & Christofides, P. D. (2015). Multiscale modeling and operation of PECVD of thin film solar cells. *Chemical Engineering Science*, *136*, 50–61. https://doi.org/10.1016/j.ces.2015.02.027

Crose, M., Zhang, W., Tran, A., & Christofides, P. D. (2018). Multiscale three-dimensional CFD modeling for PECVD of amorphous silicon thin films. *Computers and Chemical Engineering*, *113*, 184–195. https://doi.org/10.1016/j.compchemeng.2018.03.011

Dayhoff, J. E., & DeLeo, J. M. (2001). Artificial neural networks: Opening the black box. *Cancer*, *91*(8 SUPPL.), 1615–1635.

Ding, B., & Ping, X. (2012). Dynamic output feedback model predictive control for nonlinear systems represented by Hammerstein – Wiener model. *Journal of Process Control*, *22*(9), 1773–1784. https://doi.org/10.1016/j.jprocont.2012.07.011

Ding, Y., Zhang, Y., Kim, K., Tran, A., Wu, Z., & Christofides, P. D. (2019). Microscopic modeling and optimal operation of thermal atomic layer deposition. *Chemical Engineering Research and Design*, *145*, 159–172. https://doi.org/10.1016/j.cherd.2019.03.004

Du Tao, Wang Xiuli, & Wang Xifan. (n.d.). A combined model of wavelet and neural network for short term load forecasting. In *Proceedings. International Conference on Power System Technology* (Vol. 4, pp. 2331–2335). IEEE. https://doi.org/10.1109/ICPST.2002.1047201

Fausett, L. (1993). *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications* (1st ed.). Upper Saddle River, NJ: Prentice Hall.

Frenkel, D., & Smit, B. (2001). *Understanding Molecular Simulation: From Algorithms to Applications* (2nd ed.). Academic Press.

Gandhi, A. B., Joshi, J. B., Jayaraman, V. K., & Kulkarni, B. D. (2007). Development of support vector regression (SVR)-based correlation for prediction of overall gas hold-up in bubble column reactors for various gas–liquid systems. *Chemical Engineering Science*, *62*(24), 7078–7089. https://doi.org/10.1016/J.CES.2007.07.071

Garg, A., Corbett, B., Mhaskar, P., Hu, G., & Flores-Cerrillo, J. (2017). Subspace-based model identification of a hydrogen plant startup dynamics. *Computers and Chemical Engineering*, *106*, 183–190. https://doi.org/10.1016/j.compchemeng.2017.05.020

Garg, A., & Mhaskar, P. (2017). Subspace Identification-Based Modeling and Control of Batch Particulate

Processes. *Industrial & Engineering Chemistry Research*, *56*(26), 7491–7502. https://doi.org/10.1021/acs.iecr.7b00682

Hagan, M. T., & Menhaj, M. B. (1994). Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, *5*(6), 989–993. https://doi.org/10.1109/72.329697

Hasenauer, J., Jagiella, N., Hross, S., & Theis, F. J. (2015). Data-Driven Modelling of Biological Multi-Scale Processes. *Journal of Coupled Systems and Multiscale Dynamics*, *3*(2), 101–121. https://doi.org/10.1166/jcsmd.2015.1069

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, *4*(2), 251–257. https://doi.org/10.1016/0893-6080(91)90009-T

Huang, C.-M., & Yang, H.-T. (2001). Evolving wavelet-based networks for short-term load forecasting. *IEE Proceedings - Generation, Transmission and Distribution*, *148*(3), 222. https://doi.org/10.1049/ip-gtd:20010286

Huang, J., Zhang, X., Orkoulas, G., & Christofides, P. D. (2011). Dynamics and control of aggregate thin film surface morphology for improved light trapping: Implementation on a large-lattice kinetic Monte Carlo model. *Chemical Engineering Science*, *66*(23), 5955–5967. https://doi.org/10.1016/j.ces.2011.08.020

Huang, K., Zhan, X.-L., Chen, F.-Q., & Lü, D.-W. (2003). Catalyst design for methane oxidative coupling by using artificial neural network and hybrid genetic algorithm. *Chemical Engineering Science*, *58*(1), 81–87. https://doi.org/10.1016/S0009-2509(02)00432-3

Jalali, H., Nieuwenhuyse, I. Van, & Picheny, V. (2017). Comparison of Kriging-based algorithms for simulation optimization with heterogeneous noise. *European Journal of Operational Research*, *261*(1), 279–301. https://doi.org/10.1016/j.ejor.2017.01.035

Joseph, B., & Hanratty, F. W. (1993). Predictive Control of Quality in a Batch Manufacturing Process Using Artificial Neural Network Models. *Industrial and Engineering Chemistry Research*, *32*(9), 1951–1961. https://doi.org/10.1021/ie00021a019

Khanmohammadi, S., Tutun, S., & Kucuk, Y. (2016). A New Multilevel Input Layer Artificial Neural Network for Predicting Flight Delays at JFK Airport. *Procedia - Procedia Computer Science*, *95*, 237–244. https://doi.org/10.1016/j.procs.2016.09.321

Kimaev, G., & Ricardez-Sandoval, L. A. (2017). A comparison of efficient uncertainty quantification techniques for stochastic multiscale systems. *AIChE Journal*, *63*(8), 3361–3373. https://doi.org/http://doi.org/10.1002/aic.15702

Kimaev, G., & Ricardez-Sandoval, L. A. (2018). Multilevel Monte Carlo for noise estimation in stochastic multiscale systems. *Chemical Engineering Research and Design*, *140*, 33–43. https://doi.org/10.1016/j.cherd.2018.10.006

Kleijnen, J. P. C. (2017). Regression and Kriging metamodels with their experimental designs in simulation: A review. *European Journal of Operational Research*, *256*(1), 1–16. https://doi.org/10.1016/j.ejor.2016.06.041

Krishnapura, V. G., & Jutan, A. (2000). A neural adaptive controller. *Chemical Engineering Science*, *55*(18), 3803–3812. https://doi.org/10.1016/S0009-2509(00)00034-8

Kumar, B. S., & Venkateswarlu, C. (2012). Estimating biofilm reaction kinetics using hybrid mechanistic-neural network rate function model. *Bioresource Technology*, *103*(1), 300–308. https://doi.org/https://doi.org/10.1016/j.biortech.2011.10.006

Kwon, J. S.-I., Nayhouse, M., & Christofides, P. D. (2015). Multiscale, Multidomain Modeling and Parallel Computation: Application to Crystal Shape Evolution in Crystallization. *Industrial & Engineering Chemistry Research*, *54*(47), 11903–11914. https://doi.org/10.1021/acs.iecr.5b02942

Kwon, J. S.-I., Nayhouse, M., Christofides, P. D., & Orkoulas, G. (2013). Modeling and Control of Protein Crystal Shape and Size in Batch Crystallization. *AIChE Journal*, *59*(7), 2317–2327. https://doi.org/https://doi.org/10.1002/aic.14039

Kwon, J. S.-I., Nayhouse, M., Christofides, P. D., & Orkoulas, G. (2014). Modeling and control of crystal shape in continuous protein crystallization. *Chemical Engineering Science*, *107*, 47–57. Retrieved from https://doi.org/10.1016/j.ces.2013.12.005

Kwon, J. S.-I., Nayhouse, M., Orkoulas, G., & Christofides, P. D. (2014). Crystal shape and size control using a plug flow crystallization configuration. *Chemical Engineering Science*, *119*, 30–39. https://doi.org/10.1016/j.ces.2014.07.058

Lam, R., & Vlachos, D. (2001). Multiscale model for epitaxial growth of films: Growth mode transition. *Physical Review B*, *64*(3), 035401.

Lang, K. J., Waibel, A. H., & Hinton, G. E. (1990). A time-delay neural network architecture for isolated word recognition. *Neural Networks*, *3*(1), 23–43. https://doi.org/10.1016/0893-6080(90)90044-L

Lapedes, A., & Farber, R. (1987). Nonlinear signal processing using neural networks: Prediction and system modelling. In *IEEE international conference on neural networks, San Diego, CA, USA, 21 Jun 1987*. Retrieved from https://www.osti.gov/servlets/purl/5470451

Lawrynczuk, M. (2016). Nonlinear predictive control of dynamic systems represented by Wiener–Hammerstein models. *Nonlinear Dynamics*, *86*, 1193–1214. https://doi.org/10.1007/s11071-016-2957-0

Lee, D., Mohr, A., Kwon, J. S.-I., & Wu, H.-J. (2018). Kinetic Monte Carlo modeling of multivalent binding of CTB proteins with GM1 receptors. *Computers & Chemical Engineering*, *118*, 283–295. https://doi.org/https://doi.org/10.1016/j.compchemeng.2018.08.011

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, *2*(2), 164–168. Retrieved from https://doi.org/10.1090/qam/10666

Li, G., Liu, Z., Li, J., Fang, Y., Liu, T., Mei, Y., & Wang, Z. (2018). Application of general regression neural network to model a novel integrated fluidized bed gasifier. *International Journal of Hydrogen Energy*, *43*(11), 5512–5521. https://doi.org/10.1016/j.ijhydene.2018.01.130

Li, J., Croiset, E., & Ricardez-Sandoval, L. (2015). Carbon nanotube growth: First-principles-based kinetic Monte Carlo model. *Journal of Catalysis*, *326*, 15–25. https://doi.org/10.1016/j.jcat.2015.03.010

Lian, C., Zeng, Z., Yao, W., Tang, H., & Chen, C. L. P. (2016). Landslide Displacement Prediction With Uncertainty Based on Neural Networks With Random Hidden Weights. *IEEE Transactions on Neural Networks and Learning Systems*, *27*(12), 2683–2695.

Luo, E. Z., Heun, S., Kennedy, M., Wollschläger, J., & Henzler, M. (1994). Surface roughness and conductivity of thin Ag films. *Physical Review B*, *49*(7), 4858–4865.

Marquardt, D. W. (1963). An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, *11*(2), 431–441. Retrieved from https://doi.org/10.1137/0111030

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133. https://doi.org/10.1007/BF02478259

Meidanshahi, V., Corbett, B., Adams II, T. A., & Mhaskar, P. (2017). Subspace model identification and model predictive control based cost analysis of a semicontinuous distillation process. *Computers and Chemical Engineering*, *103*, 39–57. https://doi.org/10.1016/j.compchemeng.2017.03.011

Nandi, S., Ghosh, S., Tambe, S. S., & Kulkarni, B. D. (2001). Artificial neural-network-assisted stochastic process optimization strategies. *AIChE Journal*, *47*(1), 126–141. https://doi.org/10.1002/aic.690470113

Narasingam, A., & Kwon, J. S.-I. (2018). Data-driven identification of interpretable reduced-order models using sparse regression. *Computers & Chemical Engineering*, *119*, 101–111. https://doi.org/https://doi.org/10.1016/j.compchemeng.2018.08.010

Oladyshkin, S., & Nowak, W. (2012). Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion. *Reliability Engineering and System Safety*, *106*, 179–190. https://doi.org/10.1016/j.ress.2012.05.002

Oliveira, R. (2004). Combining first principles modelling and artificial neural networks: A general framework. *Computers and Chemical Engineering*, *28*(5), 755–766. https://doi.org/10.1016/j.compchemeng.2004.02.014

Ozkaya, B., Demir, A., & Bilgili, M. S. (2007). Neural network prediction model for the methane fraction in biogas from field-scale landfill bioreactors. *Environmental Modelling and Software*, *22*(6), 815–822. https://doi.org/10.1016/j.envsoft.2006.03.004

Raimondeau, S., & Vlachos, D. G. (2000). Low-Dimensional Approximations of Multiscale Epitaxial Growth Models for Microstructure Control of Materials. *Journal of Computational Physics*, *160*(2), 564–576.

Rasoulian, S., & Ricardez-Sandoval, L. A. (2014). Uncertainty analysis and robust optimization of multiscale process systems with application to epitaxial thin film growth. *Chemical Engineering Science*, *116*, 590–600.

Rasoulian, S., & Ricardez-Sandoval, L. A. (2015a). A robust nonlinear model predictive controller for a multiscale thin film deposition process. *Chemical Engineering Science*, *136*, 38–49.

Rasoulian, S., & Ricardez-Sandoval, L. A. (2015b). Robust multivariable estimation and control in an epitaxial thin film growth process under uncertainty. *Journal of Process Control*, *34*, 70–81.

Rasoulian, S., & Ricardez-Sandoval, L. A. (2016). Stochastic nonlinear model predictive control applied to a thin film deposition process under uncertainty. *Chemical Engineering Science*, *140*, 90–103.

Salciccioli, M., Stamatakis, M., Caratzoulas, S., & Vlachos, D. G. (2011). A review of multiscale modeling of metal-catalyzed reactions: Mechanism development for complexity and emergent behavior.

Chemical Engineering Science, *66*(19), 4319–4355.

Sarkar, D., & Modak, J. M. (2003). ANNSA: a hybrid artificial neural network/simulated annealing algorithm for optimal control problems. *Chemical Engineering Science*, *58*(14), 3131–3142. https://doi.org/10.1016/S0009-2509(03)00168-4

Schroeder. (2007). *An Introduction to Thermal Physics*. Pearson Education.

Seagaran, T. (2007). *Programming Collective Intelligence: Building Smart Web 2.0 Applications* (1st ed.). O'Reilly Media.

Shaikh, A., & Al-Dahhan, M. (2003). Development of an artificial neural network correlation for prediction of overall gas holdup in bubble column reactors. *Chemical Engineering and Processing: Process Intensification*, *42*(8–9), 599–610. https://doi.org/10.1016/S0255-2701(02)00209-X

Sharma, H., Das, G., & Samanta, A. N. (2006). ANN–based prediction of two-phase gas– liquid flow patterns in a circular conduit. *AIChE Journal*, *52*(9), 3018–3028. https://doi.org/10.1002/aic.10922

Siddhamshetty, P., Wu, K., & Kwon, J. S.-I. (2018). Optimization of simultaneously propagating multiple fractures in hydraulic fracturing to achieve uniform growth using data-based model reduction. *Chemical Engineering Research and Design*, *136*, 675–686. https://doi.org/https://doi.org/10.1016/j.cherd.2018.06.015

Solvason, C. C. (2011). *Integrated Multiscale Chemical Product Design using Property Clustering and Decomposition Techniques in a Reverse Problem Formulation*. Auburn University.

Solvason, C. C., Chemmangattuvalappil, N. G., & Eden, M. R. (2010). Multi-Scale Chemical Product Design using the Reverse Problem Formulation. *Computer Aided Chemical Engineering*, *28*, 1285–1290.

Soni, A. S., & Parker, R. S. (2004). Closed-Loop Control of Fed-Batch Bioreactors: A Shrinking-Horizon Approach. *Industrial & Engineering Chemistry Research*, *43*(13), 3381–3393. https://doi.org/10.1021/ie030535b

Sundaram, A., Ghosh, P., Caruthers, J. M., & Venkatasubramanian, V. (2001). Design of fuel additives using neural networks and evolutionary algorithms. *AIChE Journal*, *47*(6), 1387–1406. https://doi.org/10.1002/aic.690470615

Sutskever, I., & Hinton, G. (2010). Temporal-Kernel Recurrent Neural Networks. *Neural Networks*, *23*(2), 239–243. https://doi.org/10.1016/J.NEUNET.2009.10.009

Svozil, D., Kvasnicka, V., & Pospichal, J. (1997). Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems*, *39*(1), 43–62. https://doi.org/10.1016/S0169-7439(97)00061-0

Thomas, M. M., Kardos, J. L., & Joseph, B. (1994). Shrinking Horizon Model Predictive Control Applied to Autoclave Curing of Composite Laminate Materials. In *American Control Conference* (pp. 505–509).

Tiwari, M. K., & Chatterjee, C. (2010). Uncertainty assessment and ensemble flood forecasting using bootstrap based artificial neural networks (BANNs). *Journal of Hydrology*, *382*(1–4), 20–33. https://doi.org/10.1016/j.jhydrol.2009.12.013

Venkatasubramanian, V., & Vaidyanathan, R. (1992). Diagnosing noisy process data using neural networks. In *IFAC Symposia Series* (pp. 547–552).

Venkatasubramanian, Venkat. (2009). DROWNING IN DATA: Informatics and modeling challenges in a data-rich networked world. *AIChE Journal*, *55*(1), 2–8. https://doi.org/10.1002/aic.11756

Venkatasubramanian, Venkat. (2019). The promise of artificial intelligence in chemical engineering: Is it here, finally? *AIChE Journal*, *65*(2), 466–478. https://doi.org/10.1002/aic.16489

Venkatasubramanian, Venkat, & Chan, K. (1989). A neural network methodology for process fault diagnosis. *AIChE Journal*, *35*(12), 1993–2002. https://doi.org/10.1002/aic.690351210

Vlachos, D. (1997). Multiscale Integration Hybrid Algorithms for Homogeneous–Heterogeneous Reactors. *AIChE Journal*, *43*(11), 3031–3041.

Vlachos, D. (2005). A Review of Multiscale Analysis: Examples from Systems Biology, Materials Engineering, and Other Fluid–Surface Interacting Systems. *Advances in Chemical Engineering*, *30*, 1–61.

Watanabe, K., Matsuura, I., Abe, M., Kubota, M., & Himmelblau, D. M. (1989). Incipient fault diagnosis of chemical processes via artificial neural networks. *AIChE Journal*, *35*(11), 1803–1812. https://doi.org/10.1002/aic.690351106

Wong, W. C., Chee, E., Li, J., & Wang, X. (2018). Recurrent Neural Network-Based Model Predictive Control for Continuous Pharmaceutical Manufacturing. *Mathematics*, *6*(11). https://doi.org/10.3390/math6110242

Wu, Z., Tran, A., Ren, Y. M., Barnes, C. S., Chen, S., & Christofides, P. D. (2019). Model predictive control of phthalic anhydride synthesis in a fixed-bed catalytic reactor via machine learning modeling. *Chemical Engineering Research and Design*, *145*, 173–183. https://doi.org/10.1016/j.cherd.2019.02.016

Yıldız, Z., Uzun, H., Ceylan, S., & Topcu, Y. (2016). Application of artificial neural networks to co-combustion of hazelnut husk–lignite coal blends. *Bioresource Technology*, *200*, 42–47. https://doi.org/https://doi.org/10.1016/j.biortech.2015.09.114

Yuan, S., Jiao, Z., Quddus, N., Kwon, J. S.-I., & Mashuga, C. V. (2019). Developing Quantitative Structure–Property Relationship Models To Predict the Upper Flammability Limit Using Machine Learning. *Industrial & Engineering Chemistry Research*, *58*(8), 3531–3537. https://doi.org/10.1021/acs.iecr.8b05938

Zhang, Y., Ding, Y., & Christofides, P. D. (2019). Multiscale computational fluid dynamics modeling of thermal atomic layer deposition with application to chamber design. *Chemical Engineering Research and Design*, *147*, 529–544. https://doi.org/10.1016/j.cherd.2019.05.049

Zhao-Yang Dong, Bai-Ling Zhang, & Qian Huang. (2001). Adaptive neural network short term load forecasting with wavelet decompositions. In *2001 IEEE Porto Power Tech Proceedings (Cat. No.01EX502)* (Vol. vol.2, p. 6). IEEE. https://doi.org/10.1109/PTC.2001.964731

Highlights

- Artificial Neural Networks (ANNs) were trained on stochastic multiscale model data
- ANNs were used in online nonlinear model predictive control scheme
- ANNs provided accurate predictions for industrially relevant observable values
- ANN computational costs were orders of magnitude lower than the original model
- The accuracy of ANNs deteriorated for observable values subject to stochastic noise

Declaration of interests

☒ 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.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: