On the Techniques for Efficient Sampling, Uncertainty Quantification and Robust Control of Stochastic Multiscale Systems

by

Grigoriy Kimaev

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Doctor of Philosophy

in

Chemical Engineering

Waterloo, Ontario, Canada, 2020

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner	Vinay Prasad Professor Chemical and Materials Engineering, University of Alberta
Supervisor	Luis A. Ricardez-Sandoval Associate Professor Chemical Engineering, University of Waterloo
Internal Members	Hector Budman Professor Chemical Engineering, University of Waterloo
	Eric Croiset Chair & Professor Chemical Engineering, University of Waterloo
Internal-external Member	Christopher Nielsen Associate Professor Electrical and Computer Engineering, University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Chapters 1, 2 and 7 were written entirely by me. My supervisor, Luis A. Ricardez-Sandoval, provided suggestions and recommendations regarding their contents.

The results in chapter 3 of this thesis have been published to the *AIChE Journal* on March 7, 2017. 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/10.1002/aic.15702 That paper was written entirely by myself and edited by my supervisor.

The results in chapter 4 of this thesis have been published to the *AIChE Journal* on December 4, 2017. Kimaev, G., & Ricardez-Sandoval, L. A. (2018). "Multilevel Monte Carlo applied to chemical engineering systems subject to uncertainty." *AIChE Journal*, *64*(5), 1651–1661. https://doi.org/10.1002/aic.16045 That paper was written entirely by myself and edited by my supervisor.

The results in section 5.1 of this thesis have been published to the *Chemical Engineering Research and Design* journal on October 13, 2018. 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 That paper was written entirely by myself and edited by my supervisor.

The results in sections 5.2 – 5.4 of this thesis have been published to the *AIChE Journal* on May 8, 2020. Kimaev, G., Chaffart, D., & Ricardez-Sandoval, L. A. (2020). "Multilevel Monte Carlo applied for uncertainty quantification in stochastic multiscale systems." *AIChE Journal*, 66(8), e16262. https://doi.org/10.1002/aic.16262 My fellow PhD Candidate Donovan Chaffart provided

the description of the gap-tooth catalytic pore reactor model (which can be found in section 5.4 of this thesis) and its implementation in MATLAB® code. The rest of the paper was written entirely by myself and edited by my supervisor. This contribution was identified by Dr. Jinfeng Liu (University of Alberta) as the Best Presentation in the session "Advances in Computational Methods and Numerical Analysis" of the 2018 AIChE Annual Meeting in Pittsburgh.

The results in section 6.3 of this thesis have been published to the *Chemical Engineering Science* journal on July 19, 2019. Kimaev, G., & Ricardez-Sandoval, L. A. (2019). "Nonlinear model predictive control of a multiscale thin film deposition process using Artificial Neural Networks." *Chemical Engineering Science*, *207*, 1230–1245. https://doi.org/10.1016/j.ces.2019.07.044 That paper was written entirely by myself and edited by my supervisor.

The results in section 6.4 of this thesis have been published to the *Chemical Engineering Research and Design* journal on June 25, 2020. Kimaev, G., & Ricardez-Sandoval, L. A. (2020). "Artificial Neural Networks for dynamic optimization of stochastic multiscale systems subject to uncertainty." *Chemical Engineering Research and Design*, *161*, 11–25. https://doi.org/10.1016/j.cherd.2020.06.017 That paper was written entirely by myself and edited by my supervisor.

The results in section 6.5 of this thesis have been published to the *Journal of Physical Chemistry C* virtual special issue "Machine Learning in Physical Chemistry" on July 30, 2020. Kimaev, G., & Ricardez-Sandoval, L. A. (2020). "Artificial Neural Network discrimination for parameter estimation and optimal product design of thin films manufactured by chemical vapour deposition." http://dx.doi.org/10.1021/acs.jpcc.0c05250 That paper was written entirely by myself and edited by my supervisor.

Abstract

In order to better understand and leverage natural phenomena to design materials and devices (e.g. biomedical coatings, catalytic reactors, thin conductive films for microprocessors, etc.), stochastic multiscale models have been developed that explicitly model the interactions and feedbacks between the electronic, atomistic/molecular, mesoscopic and macroscopic scales. These models attempt to use the accurate results from the fine scales to inform industrially relevant domain sizes and thereby improve product quality through optimal control actions during industrial manufacturing. However, the presence of stochastic calculations increases the computational cost of such modeling approaches and makes their direct application in uncertainty quantification, optimization and online control challenging. Uncertainty cannot be ignored from simulations, otherwise there will be model-plant mismatch and loss in performance. The added computational intensity necessitates the development of more efficient computational methods that can leverage the accurate predictions of stochastic multiscale models in the industrial setting where accuracy, efficiency and speed are of utmost importance.

A lot of research has been done in the area of stochastic multiscale models over the past few decades, but some gaps in knowledge remain. For instance, the performance of traditional uncertainty quantification techniques such as power series (PSE) and polynomial chaos expansions (PCE) has not been compared in the context of stochastic multiscale systems. Furthermore, a novel sampling technique called Multilevel Monte Carlo (MLMC) sampling emerged from the field of computational finance with the aim of preserving accuracy of estimation of model observables while decreasing the required computational cost. However, its applications in the field of chemical engineering and in particular for stochastic multiscale systems remain limited. Also, the advancements in computing power caused the usefulness of machine learning methods such as

Artificial Neural Networks (ANNs) to increase. Because of their flexibility, accuracy and computational efficiency, ANNs are experiencing a resurgence of research interest, but their application for stochastic multiscale chemical engineering systems are still limited at the moment.

This thesis aims to fill the identified gaps in knowledge. The results of the conducted research indicate that PCE can be more computationally efficient and accurate than PSE for stochastic multiscale systems, but it may be vulnerable to the effects of stochastic noise. MLMC sampling provides an attractive advantage over the heuristic methods for uncertainty propagation in stochastic multiscale systems because it allows to estimate the level of noise in the observables. However, the stochastic noise imposes a limit on the maximum achievable MLMC accuracy, which was not observed for continuous systems that were originally used in MLMC development. ANNs appear to be a very promising method for online model predictive control of stochastic multiscale systems because of their computational efficiency, accuracy and robustness to large disturbances not seen in the training data.

Acknowledgments

First of all, I would like to express my sincere gratitude to my supervisor, Prof. Luis A. Ricardez-Sandoval, for his help and support, patience, suggestions and relentless drive to carry on with the research and publications. Among all the supervisors I've had to date, he is the first one who managed to push me to publish peer-reviewed papers.

I would like to thank my examining committee for taking the time to review my thesis and providing valuable feedback: Prof. Vinay Prasad, Prof. Hector Budman, Prof. Eric Croiset and Prof. Christopher Nielsen.

I wish to acknowledge the financial assistance provided by the Natural Sciences and Engineering Research Council of Canada, the Faculty of Engineering and the Department of Chemical Engineering at the University of Waterloo, and the Waterloo Institute for Nanotechnology.

I want to thank all my friends and colleagues for their time, encouragement, inside jokes, dinners and birthday celebrations we shared, as well as the discussions and arguments we've had on the widest possible range of topics. In particular, I would like to thank Dr. Pendar Mahmoudi, Dr. Navid Bizmark, Dr. Amir Mowla, Dr. Mina Rafiei, Donovan Chaffart, Kavitha Menon, Yue Yuan, Huabei You, Han Wang, Mahshad Valipour, Ilse Mariana Cerrillo Briones, Manuel Alejandro Tejeda Iglesias, Marco Antonio Lucio Hernandez, Yael Izamal Valdez Navarro, Oscar Palma-Flores, Gabriel Patron, Javad Noroozi, Alex Vasile and James Lowman.

Last but certainly not least, I wish to express my appreciation to my dad, mom and brother for their support and patience during my journey towards this milestone. However, I still insist that they should pay for the gown rental. Also, I would like to sincerely thank Sherman, Elena, Danila and Nicholas for their care and help during my studies.

Dedication

To the Self-Existent One.

A knowledge of Him through His self-revelation is the only relief I could find to my desperate search for the meaning of life.

I wish I were perfect in sharing with those I've been privileged to meet the meaning, peace and joy of a relationship with Him and the answers that benefited my brief existence so much.

Table of Contents

List of Figures

List of Tables

List of Acronyms

ANN	Artificial Neural Network
CVD	Chemical Vapour Deposition
kMC	kinetic Monte Carlo
MC	Monte Carlo
MLMC	Multilevel Monte Carlo
MPC	Model Predictive Control
NMPC	Nonlinear Model Predictive Control
NISP	Non-Intrusive Spectral Projection
PCE	Polynomial Chaos Expansion
PSE	Power Series Expansion
UQ	Uncertainty Quantification

List of Symbols

General Purpose Symbols

h_l	MLMC domain discretization step size at level l
C_l	computational cost at MLMC discretization level l
Ns_l	total number of MLMC samples used at level l during preliminary sampling
P_l	approximation of P at MLMC discretization level l
V_l	variance of the estimators of P at different MLMC discretization levels l
$\boldsymbol{y}_{PCE}$	PCE-based prediction of observable values
$\boldsymbol{y}_{PSE}$	PSE-based prediction of observable values
$\mathcal{N}_l$	optimal number of MLMC samples at discretization level l
α_{MLMC}	MLMC Theorem constant for convergence in mean value of estimator
α	empirical constant used to relate the kMC timestep of the CVD model to temperature and lattice size (Chapter 5)
β	empirical constant used to relate the kMC timestep of the CVD model to temperature and lattice size (Chapter 5)
$\beta_{\ell,ANN}$	bias that contributes to the activation of ANN neuron ℓ
β_{MLMC}	MLMC Theorem constant for convergence in variance of estimator
β_i	PCE expansion coefficient
γ_{MLMC}	MLMC Theorem constant for convergence in computational cost
$\mu_{\boldsymbol{u}}$	nominal (mean) values of uncertain parameters $\boldsymbol{u}$
φ_{ANN}	transfer function that acts on the inputs to a neuron in ANN
χ_ℓ	output of a neuron ℓ in ANN

$\omega_{\ell,\mathscr{p}}$	weight associated with the connection between the neuron ℓ and some neuron $\mathscr{p}$
ℓ	index of a neuron in ANN
$\mathcal{M}$	optimal refinement factor of MLMC domain discretization step size h_l
ϵ	user-defined root mean square error tolerance of MLMC
Σ	summation operator
Ψ	the orthogonal basis polynomials for PCE
Ω	support range of w used in the NISP calculation
L	highest discretization level in MLMC
P	observable of a random variable
Y	multilevel estimator of $\mathbb{E}[P]$ in the MLMC scheme
k	kurtosis of the distribution of an observable
l	discretization level in the MLMC scheme
s	skewness of the distribution of an observable
w	weighting function for the calculation of PCE β coefficients by NISP method
y	radial coordinate in the catalytic pore reactor model
$\boldsymbol{u}$	uncertain parameters for PCE or PSE
$\boldsymbol{y}$	observables used for PSE/PCE identification
$\mathcal{N}(\mu,\sigma^2)$	normal distribution with mean μ and standard deviation σ
$\mathbb{E}$	expected value operator
σ	standard deviation of the distribution of an observable
$\boldsymbol{\xi}$	scaled values of the random parameters in PCE

CVD model

C_{tot}	concentration of sites on thin film surface
E_d	energy required for desorption
E_m	energy required for migration
M_i	number of adsorbed surface atoms with a particular number of neighbours
N_a	number of adsorbed atoms during a kMC timestep
N_d	number of desorbed atoms during a kMC timestep
P_{CVD}	chamber pressure
R_a	rate of adsorption (macroscale portion of the CVD model)
R_d	rate of desorption (macroscale portion of the CVD model)
R_g	gas constant
S_0	sticking coefficient
T_{bulk}	bulk temperature of the chemical vapour
T_{surf}	substrate and thin film temperature
W_a	kMC rate of adsorption
W_d	kMC rate of desorption
W_m	kMC rate of migration
X_{bulk}	mole fraction of the chemical species in the bulk
k_{d0}	event frequency constant
$\mu_b \rho_b$	viscosity of the bulk gas multiplied by its density
ρ_b / ρ	density of the bulk divided by gas boundary layer density
Δt_{kMC}	kMC time increment

$\Delta\mathcal{T}$	coupling time interval
E	energy associated with a single bond
Gr	Growth rate of the thin film
N	kMC lattice size (the number of atoms along an edge of the square lattice)
Pr	Prandtl number
R	Roughness of the thin film
Sc	Schmidt number of the depositing chemical species
Th	Thickness of the thin film
a	hydrodynamic strain rate
f	dimensionless stream function
i	number of neighbours of an atom in the kMC simulation
m	precursor molecular weight
x	mole fraction of the depositing chemical vapour species
η	dimensionless distance away from thin film surface
ς	random number used to increment kMC time
τ	dimensionless time, $\tau = 2at$

Mixing Tank model

A_T	area of the mixing tank
H	height of the liquid level in the mixing tank
H_{sp}	setpoint height of the liquid level in the mixing tank
ε	error term of the PI controller in the mixing tank model ($\varepsilon = H_{sp} - H$)
F_{in}	mixing tank inlet flow rate

F_{out}	mixing tank outlet flow rate
C_v	outlet valve coefficient
ρ_w	density of water
g_w	standard gravity of water
K	stem position of the outlet valve
$\overline{K}$	nominal value of K
K_c	PI controller gain
$\tau_{i,tank}$	PI controller integral time constant

Catalytic Pore Reactor model

$\mathcal{L}$	length of the catalytic pore
ρ	radius of the catalytic pore
n	the number of teeth along the length of the catalytic pore, controlled by MLMC
$C_{i,in}$	inlet concentration of species i
$C_{i,out}$	outlet concentration of species i
C_i	concentration of species i (where i can correspond to species A, B or reaction product AB)
D_i	diffusion coefficient of species i
$E_{d,A}$	activation energy of desorption for species A
N_e	number of empty catalytic surface sites in the catalytic pore
N_i	quantity of adsorbed molecules of species i in the catalytic pore

N_r	number of A^* and B^* species pairs adsorbed at adjacent catalyst surface sites ready to react to produce AB in the catalytic pore reactor
τ_ι	kMC time increment
$\omega_{i,ads}$	rate of adsorption of species i
$\omega_{i,cons}$	rate of consumption of species i
$\omega_{i,des}$	rate of desorption of species i
$\omega_{i,prod}$	rate of production of species i
ν	fluid velocity

Wastewater Treatment Plant model

V_r	volume of the bioreactor
c_s	oxygen specific saturation
f_d	fraction of dead biomass that contributes to the substrate
k_{01}	rate of oxygen demand
k_c	rate of biomass death by biological waste
k_d	rate of biomass death by endogenous metabolism
k_{la}	rate of oxygen transfer into the water
k_s	saturation constant
l_b	depth of the middle layer of the settler
l_d	depth of the top layer of the settler
l_r	depth of the bottom layer of the settler
q_r	flow rate of the recycled stream
s_i	substrate concentration in the inlet stream

$x_{i,bio}$	biomass concentration in the inlet stream
$x_{r,bio}$	biomass concentration in the recycled stream
y_{bio}	fraction of substrate converted to biomass
μ_{gr}	specific growth rate of the microorganisms
φ	uncertain exponent of biomass concentration in equations (45) and (46)
ω	the frequency of the disturbance in the inlet stream

Distillation Column model

A_{cond}	Area of the condenser
A_{cool}	Area of the cooler
A_{reb}	Area of the reboiler
D_{col}	Column diameter
F_{22}	Feed flowrate
K_{c1}	Gain of the toluene PI controller
K_{c2}	Gain of the heptane PI controller
N_{feed}	Feed stage number
N_{stages}	Number of stages in the column (the top stage is #1)
P_{cond}	Pressure in the condenser
τ_{I1}	Integral time constant of the toluene controller
τ_{I2}	Integral time constant of the heptane controller
sp_1	Set-point for the PI controller of the of toluene mole fraction in the top stream
sp_2	Set-point for the PI controller of heptane mole fraction in the bottom stream

Chapter 1: Motivation

Numerous processes in systems biology, fluid-surface interactions, materials manufacturing and chemical engineering are inherently multiscale. Some examples of processes that consist of events occurring at several temporal and spatial scales are: heterogeneous catalysis, semiconductor doping, thin film production for integrated circuits, manufacturing of micro-electro-mechanical systems, photovoltaic cells and biomedical devices, as well as protein crystallization for pharmaceuticals.[1–10] Furthermore, all of these processes, and many others, are subject to uncertainty.

While the properties (e.g. lattice structure, roughness, composition, etc.) of the products of the processes mentioned above often directly depend on the microscale phenomena, macroscale quantities (e.g. pressure, concentration, temperature) strongly influence the occurrence and behaviour of the microscale events. For efficient mass production, the macroscale variables are often used for process optimization and control.[11] The behaviour of macroscale quantities can be modelled using the well-known continuum approach and closed-form expressions. However, since bidirectional interactions between the macro and molecular scales exist,[12,13] multiscale modelling techniques were developed in order to enhance model prediction accuracy (i.e. reduce model-plant mismatch) and, therefore, improve the product quality and the overall efficiency of the processes while accounting for their inherent multiscale nature.[1,3,4,13–20]

The coupling of highly different temporal and spatial scales can make the multiscale models stiff (i.e. highly sensitive to small perturbations). Furthermore, the explicit modelling of microscale phenomena introduces several challenges. Simulations of the molecular scale demand the implementation of techniques that model stochastic phenomena (e.g. Molecular Dynamics,

Metropolis Monte Carlo, kinetic Monte Carlo, etc.), since the traditional closed-form equations of continuum modelling cannot explicitly capture the phenomena at small length and time scales such as lateral interactions and many-body effects.[1,13,21] Consequently, the random noise associated with stochastic modelling creates variability in the observables of stochastic multiscale models. Stochastic modelling techniques of the molecular scale phenomena also tend to be quite computationally intensive. As a result, the computational complexity associated with sampling, uncertainty quantification, optimization and control of such models grows and often inhibits their applicability to time-sensitive scenarios such as online control under uncertainty.

To overcome the computational challenges and take advantage of the potential benefits offered by stochastic multiscale models for process and product design, many studies have been published that examine efficient uncertainty propagation, optimization and model predictive control techniques for such systems. Some of the multiscale systems under consideration included epitaxial thin film growth, plasma-enhanced chemical vapour deposition, protein crystallization, heterogeneous catalysis, wood fiber production and flows through porous media.[4–8,18,22–37] Power Series and Polynomial Chaos Expansions (PSE and PCE, respectively) have been used for efficient sampling, uncertainty quantification and robust optimization of stochastic multiscale models.[9,35,37–42] Furthermore, reduced order models, multidomain modelling, parallel computation and data-driven models such as the machine learning technique called Artificial Neural Networks (ANNs) have been applied for the control of stochastic multiscale systems.[17,32,33,35,43–48]

However, to the extent of the author's knowledge, the performance of PSE and PCE techniques has not been compared in the context of uncertainty propagation in a stochastic multiscale model until the work in this thesis has been performed. Also, the current applications of an advanced sampling technique called Multilevel Monte Carlo (MLMC) appear to be limited in the field of

chemical engineering to uncertainty quantification of porous media flows, oil reservoir simulations and enhanced oil recovery.[49–53] The performance of MLMC, PSE and PCE techniques has not been compared for traditional chemical engineering systems. Moreover, MLMC has not been applied to stochastic multiscale models. In addition, the applications of ANNs to stochastic multiscale systems and product design under uncertainty remain limited in number.[31,46–48]

1.1: Research Objectives

The purpose of this thesis is to address the gaps in the literature by presenting the work performed in the areas of sampling, noise estimation and uncertainty quantification of stochastic multiscale systems, as well as their efficient online control under nominal conditions and in the presence of uncertainty using data-driven models.

The following research objectives are pursued in this thesis in order to address the gaps in knowledge:

- Investigate the effect of random noise associated with the multiscale stochastic systems on the performance of common uncertainty quantification techniques, such as PSE and PCE.
- Apply MLMC to estimate the expected values of traditional chemical engineering systems (e.g. wastewater treatment plant, distillation column), assess the variability of the estimates and compare MLMC performance to PSE and PCE.
- Apply MLMC to stochastic multiscale systems (e.g. chemical vapour deposition chamber, catalytic pore reactor) to estimate the variability in the estimated expected values and use MLMC to improve uncertainty quantification in stochastic multiscale systems (e.g. via PCE).

- Develop computationally efficient and robust ANN models to predict the responses of stochastic multiscale models to a wide range of perturbations in the manipulated variables, under nominal conditions and in the presence of parametric uncertainty. Conduct model predictive control of the system by completely replacing the stochastic multiscale model with the ANNs.
- Use ANNs to account for time-invariant parametric uncertainty with a variety of techniques. First, model the statistical moments of the observables and conduct open-loop dynamic optimization. Next, use parameter estimation to discriminate between multiple plausible ANNs and conduct closed-loop control under uncertainty using the best-fit ANN.

1.2: Structure of the Thesis

This thesis is organized as follows:

Chapter 2 presents a literature review of the current applications of uncertainty quantification techniques, multilevel Monte Carlo sampling, and machine learning to the field of chemical engineering in general and specifically to stochastic multiscale systems. The gaps in knowledge that motivate the research presented in this thesis are identified and discussed.

Chapter 3 compares the performance of traditional methods, i.e. power series and polynomial chaos expansions (PSE and PCE, respectively), in uncertainty quantification of a stochastic multiscale system subject to single- and multi-parameter uncertainty. The effect of stochastic noise is highlighted and discussed. The work in Chapter 3 has been partially presented at the 66th Canadian Chemical Engineering Conference in 2016 and has been published in the AIChE Journal.[54]

Chapter 4 applies multilevel Monte Carlo sampling (MLMC) for the estimation of expected values in standard chemical engineering systems subject to parametric uncertainty. A tutorial example explains how the MLMC method samples a continuous system subject to time-invariant uncertainty. Following the tutorial, the performance of MLMC is compared to PSE, PCE and brute-force Monte Carlo sampling (MC) for three traditional chemical engineering systems. This work has been published in the AIChE Journal.[55]

Chapter 5 extends the MLMC method to noise estimation in the observables of stochastic multiscale systems. The results of noise estimation are subsequently used to improve uncertainty quantification via Polynomial Chaos Expansions (PCE). The nuances involved in the adaptation of the MLMC method to stochastic multiscale systems are presented. The performance of MLMC in these systems is qualitatively compared to the results of Chapter 4. This work has been partially presented at the 2018 and 2019 AIChE Annual Meetings and published in the Chemical Engineering Research and Design journal[56] and the AIChE Journal.[57]

Chapter 6 switches the scope of the thesis from sampling and uncertainty quantification of stochastic multiscale systems to their efficient control. Artificial Neural Networks (ANNs) are used to capture and predict the dynamic responses of a stochastic multiscale system to perturbations in the manipulated variables. Furthermore, the chapter presents two approaches for applying ANNs for nonlinear model predictive control (NMPC) of stochastic multiscale systems under parametric uncertainty. ANNs were used to model the statistical moments of system observables and parameter estimation was used to identify a best-fit ANN for closed-loop NMPC. This work has been partially presented at the 68th Canadian Chemical Engineering Conference in 2018 as well as the 2019 AIChE Annual Meeting. This work has been published in the Chemical Engineering Science journal,[58] accepted for publication in the Chemical Engineering Research and

Design journal (acceptance number CHERD-D-20-00647R1) and submitted to the Journal of Physical Chemistry C (submission number jp-2020-05250w).

Chapter 7 summarizes the key outcomes and contributions from the studies performed in Chapters 3-6 and outlines the recommendations for future research.

Chapter 2: Literature Review

Over the past decades, many studies have been published examining different stochastic multiscale systems in chemical engineering as well as various sophisticated mathematical models developed to explain and predict the behaviour of those systems. Furthermore, numerous techniques have been devised for efficient sampling, uncertainty quantification and model predictive control of these systems and models. In addition, Multilevel Monte Carlo sampling and machine learning methods have also been applied to the field of chemical engineering over the past years. However, a review of the literature found some gaps in knowledge that serve as motivation for the research presented in this thesis. This chapter presents the conducted literature review and discusses the identified knowledge gaps.

2.1 Stochastic Multiscale Processes

Numerous processes in systems biology, fluid-surface interactions, materials manufacturing and chemical engineering are comprised of interacting 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.[1–8,26,27,37,59–62]

While the phenomena and structure at the microscale directly affect the final product's properties and performance, in the industrial mass-production setting usually only the macroscale quantities (e.g. inlet concentration(s), inlet flow rate(s), temperature, etc.) are available for the manipulation

of multiscale processes to ensure that the product meets design specifications.[1,2,11,12] Consequently, stochastic multiscale models have been developed to gain insight on and enable the control of the bidirectional interactions between different spatial and temporal scales and identify the underlying mechanisms that impact process performance.[3,4,12,13,15–18,24,28]

Several approaches exist for the implementation of stochastic multiscale models. While the techniques and implementations can differ widely, the ultimate goal of using these models is to expedite the product development cycle, improve product quality and lessen the cost associated with the design of materials. Such models consist of coupled simulations of multiple temporal and spatial scales and provide sufficiently accurate results at a high computational cost. For example, although the simulations of individual particles over industrially relevant domain sizes are computationally infeasible because the quantity of particles in the domain of interest can approach or exceed Avogadro's number, in multiscale simulations the electronic structure of atoms can be used to inform the intermolecular potential calculations, which in turn can improve the understanding of bulk behaviour.[21] Thus, sufficiently accurate results can be obtained at a high but feasible computational cost.

One common approach to classifying simulation scales is by the temporal and spatial domains, e.g. the frequency of events and the length scales over which they occur.[16,21,63] Using this approach, the scales can be classified[16,21,64] as electronic (where the events occur with frequencies shorter than 10^{-12} s over distances less than 10^{-9} m), atomistic (where the event frequency range is 10^{-12}–10^{-10} s and spatial domains are 10^{-10}–10^{-8} m long), mesoscopic (10^{-10}–10^{-2} s and 10^{-9}–10^{-2} m) and macroscopic (where the frequencies are longer than 10^{-2} s and distances exceed 10^{-6} m). It should be noted that these classifications are approximate and that the specified temporal and spatial

ranges overlap between subsequent scales. These overlaps create "handshaking" regions that enable multiscale simulation techniques to exchange information between scales.[16,21]

Different computational approaches must be used to capture the phenomena occurring at different scales. At the electronic scales, quantum mechanics calculations and density functional theory (DFT) are used to calculate the quantities of interest (e.g. reaction activation energies).[65,66] At the atomistic scale, a realistic depiction of the events requires the usage of molecular dynamics (MD) or non-closed-form expressions such as Monte Carlo (MC), kinetic Monte Carlo (kMC) or simulated annealing to properly capture the random (stochastic) behaviour of individual atoms and molecules.[64,67–69] Mesoscale calculations can rely on dissipative particle dynamics, coarse-grained MD, MC or kMC, or Brownian dynamics, to capture the average behaviour of aggregates of individual particles (hence the name "coarse-grained").[68,70]

On the other hand, the continuum assumption is valid at the macroscale, so the corresponding calculations tend to use the well-understood deterministic models based on classical mechanics and the Navier-Stokes equations of motion.[71] Continuum calculations rely on partial differential equations (PDEs) to describe the behaviour of matter and are solved using Finite Element Analysis, Finite Differences, or the Method of Lines. Alternatively, simplifying assumptions may be used to reduce PDEs to ordinary differential equations (ODEs) to further diminish the computational complexity.

Once the simulation scales have been identified, a method of exchanging information between them needs to be devised. One approach is to do sequential modeling that does not consider any feedbacks that may be present between the scales.[21] In this scenario, usually the finest scale is solved first, and then the information is provided to calculations at the larger scale. For example, the reaction activation energies can be approximated using DFT.[59,64,65] Then, the parameter values

can be used to simulate the reactions on a catalytic surface using kMC.[59] Alternatively, different scales can be coupled through a boundary condition or an initial condition, as in the case where a pharmaceutical drug first diffuses through ingested nanovesicles and then enters the body's bloodstream without being reabsorbed by the nanoparticles. In this scenario, the smaller scale (nanoparticle) delivers the drug to the larger scale (bloodstream) over some time period until a steady state is reached due to the depletion of nanoparticles.[72]

However, since feedbacks between different scales frequently occur in nature, computational models that consider these interactions are also necessary.[16,21] One method of developing such a simulation is to couple the subdomains through the "handshaking" region where the subdomains overlap, as in the case of a gas diffusing from bulk towards a substrate in a stagnation flow reactor.[15] PDEs can be used to describe the macroscale gas behaviour, while kMC or MD can be employed to model the microscale phenomena (adsorption, desorption, migration, reaction, etc.). Then, the fine scale can be solved for one timestep, the boundary condition and the macroscale solution can be updated, and the fine scale simulation can be repeated with the data from the updated macroscale solution.[13,15,21]

If it can be assumed that the boundary condition is spatially invariant (e.g. stagnation flow reactor), only one microscale simulation domain is sufficient.[13,15] However, if this assumption does not hold, as in the cases of spatial gradients in the "handshaking" region due to the absence of a gas distributor in a deposition chamber or due to fluid diffusion through a thick membrane or a tubular catalytic reactor, a multigrid approach would be necessary.[21,37,73–75] In such a simulation, the governing equations at the continuum scale would be discretized over the simulation domain and microscale simulations (i.e. "grids") would be conducted at the appropriate discretization points.

The purpose of the aforementioned multiscale approaches is to leverage highly accurate fine scale simulations to improve the accuracy of device-relevant domain sizes. In addition to using the overlapping subdomains, a different multiscale simulation paradigm exists. Rather than create interfaces between disparate spatial and temporal scales, coarse-graining can be used to approximate the average behaviour of conglomerations of individual atoms/molecules and access larger spatial domains using this methodology.[21,76,77] Also, following this paradigm, coarse-time-stepping (accelerated time advancement) and net-event kMC can be employed to access longer temporal scales.[25,59,77] However, careful selection of the coarse timestep is required to ensure numerical stability of the calculation.[78–84]

Since in stochastic multiscale models the non-closed-form expressions are used to "inform" the calculations at the continuum scale, 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. These challenges motivate the development of efficient uncertainty propagation, optimization and control strategies to ensure the industrial relevance of stochastic multiscale models.

Chemical Vapour Deposition (CVD), a unit operation of chemical engineering, is an example of a stochastic multiscale process that has significant industrial importance.[85] It is widely used for manufacturing smooth thin conductive films needed for semiconductor manufacturing.[61] This process is typically represented by multiscale models.[16] It is necessary to ensure that film roughness is within specified constraints, since high roughness will render the thin film unsuitable for integration in electronic devices applications by adversely affecting its electrical

conductivity.[86–90] Thus, the bulk property of conductivity is directly affected by the nanoscale structure of the film. Therefore, it is critical to control the key factors in film formation (e.g. inlet gas concentration, substrate temperature[61]) to keep film roughness within the specifications despite various parametric uncertainties or model structure errors (e.g. uncertain activation energies for adsorption, desorption and migration of atoms, or missing/incorrect equations for the underlying mechanisms). Ignoring the uncertainty in process parameters can lead to significant variability in product quality.[12,21]

2.2 Uncertainty Propagation

Uncertainty is the lack of deterministic knowledge about the values of process parameters, and it is present in every process. Uncertainty arises from two major sources: variability inherent to the underlying natural phenomena ("aleatory" uncertainty) and insufficient knowledge about them ("epistemic" uncertainty).[91,92] Further research into the underlying chemical and physical phenomena can reduce epistemic uncertainty through the development of more accurate models.[91,93] Alternatively, it can be assumed that accounting for parametric uncertainty can compensate for the errors in model equations.

However, unlike epistemic uncertainty, aleatory uncertainty cannot be lessened. Parameter uncertainty can be present at any scale, which motivates the integration of uncertainty quantification in multiscale modelling. Simulations of molecular-scale phenomena demand the implementation of stochastic modelling techniques, since the traditional closed-form equations of continuum modelling cannot explicitly capture the phenomena at small length and time scales such as lateral interactions and many body effects in heterogeneous catalytic systems.[1,13,21] While stochastic modelling of nanoscale phenomena takes molecular events into account more accurately and thereby removes epistemic uncertainty to some degree, the lack of closed-form expressions

adds an additional level of computational expense, simulation complexity, and introduces variability among the results of different simulation runs. The noise in these stochastic multiscale systems can be decreased by averaging the results of multiple simulations[35] at the expense of higher computational costs.

Probabilistic techniques are used to draw reliable conclusions from data that includes uncertainty.[92] Typically, these techniques associate probabilistic bounds or probability distribution functions with the uncertain variables. Since it is not possible to know exactly which numerical value of the uncertain parameter will be manifested in a particular experiment or simulation run, uncertainty must be propagated through the system – the experimental results or computer model calculations have to be obtained using as many uncertain parameter realisations as possible. Consequently, the impact of parameter uncertainty on the response variables is estimated in the form of approximating their bounds or their probability distribution functions, thereby quantifying the uncertainty.[92,94] This estimation establishes an appropriate degree of confidence in the results and conclusions drawn from the analysis. Therefore, uncertainty quantification can aid in ensuring compliance with safety and environmental regulations.[91] It is also highly beneficial to product quality improvement and the design of processes that need to remain insensitive (robust) to uncertainty.

Uncertainty propagation requires either multiple experiments or numerous simulation runs and is therefore the most resource-intensive stage in the uncertainty quantification process. If manifold experiments are conducted during uncertainty analysis, both temporal and economic expenditures are incurred. While computer simulations are available to lessen these expenses, they can still be very time-consuming. Monte Carlo (MC) sampling is a technique that can be used to randomly select the uncertain parameter values for simulations.[94,95] However, its implementation for large-

scale applications is computationally expensive, which motivates the development of more efficient techniques. For example, because of its time-sensitive nature, online process control in the presence of uncertainty would benefit from efficient uncertainty propagation methods.[38]

Power Series Expansion (PSE)[96–98] and Polynomial Chaos Expansion (PCE)[92,94,99–101] are two commonly used mathematical techniques for efficient and fast uncertainty quantification in chemical engineering applications[35,97,98] The techniques have also been applied in other disciplines as well, including but not limited to mechanics[99] and biomechanics[102]. The techniques estimate the observables using different approaches: PSE uses the sensitivities of response variables with respect to uncertain parameters, while PCE relies on an orthogonal basis set of polynomials (each scaled by an appropriate expansion coefficient), to derive low-order closed-form expressions that establish a direct relationship between the uncertain parameter(s) and an observable (output) variable of interest. Since the expansions consist of simple mathematical operations, the uncertainty propagation process can be implemented very rapidly once the sensitivities or the expansion coefficients have been obtained.

The predictive capabilities of PSE[22,34,35,37] and PCE[103–107] have been studied individually in different contexts. The performance of PSE and PCE has been compared for a simulation of a nonlinear dynamical system (batch crystallization process, represented by continuum equations) in the presence of parameter uncertainties[38]; that study suggested that 2nd order PCE produced highly accurate approximations of the non-Gaussian probability distribution function of the observable. However, it appears that this is the only comparison of PSE and PCE currently available in the literature. This gap in the literature motivated the development of the work presented in Chapter 3 of this thesis, i.e. the comparison of PSE and PCE for uncertainty quantification in stochastic multiscale systems.

2.2.1 Polynomial Chaos Expansions

Polynomial Chaos Expansion (PCE)[92] provides a closed-form expression that relates the observable variable to one or multiple uncertain parameters. PCE consists of an orthogonal basis set of polynomials that are functions of the uncertain parameters.[94,100,101] The choice of basis depends on the probability distributions of the uncertain parameters,[94] which have to be known (or assumed) *a priori*. While the polynomials in the basis are known, the expansion coefficients are not, and need to be estimated. Once the coefficients are obtained, the PCE expression can be used for fast uncertainty propagation. A general representation of the PCE is of the form:

$$\boldsymbol{y}_{PCE} = \beta_0\Psi_0 + \sum_{i=1}^{\infty}\beta_i\Psi_1(\boldsymbol{\xi_i}) + \sum_{i=1}^{\infty}\sum_{j=1}^{i}\beta_{ij}\Psi_2(\boldsymbol{\xi_i},\boldsymbol{\xi_j}) + \sum_{i=1}^{\infty}\sum_{j=1}^{i}\sum_{k=1}^{j}\beta_{ijk}\Psi_3(\boldsymbol{\xi_i},\boldsymbol{\xi_j},\boldsymbol{\xi_k}) + \ldots \qquad (1)$$

where β_i are the unknown expansion coefficients and $\Psi(\boldsymbol{\xi})$ are the polynomials in the orthogonal basis.[92] The set of random parameters is represented by $\boldsymbol{\xi}$, with $\boldsymbol{\xi}\epsilon\mathbb{R}^{n_\xi}$. The subscripts of the β_i coefficients do not have a particular significance, other than to uniquely identify the coefficient. The random parameters $\boldsymbol{\xi}$ are related to the set of uncertain parameters $\boldsymbol{u}$, where the latter represent physically significant magnitudes and units. In fact, $\boldsymbol{\xi}$ are the transformed parameters $\boldsymbol{u}$. For example, if the uncertain parameter is normally distributed with mean μ and standard deviation σ, it would be transformed to have the standard normal distribution, $\boldsymbol{u} \sim \mathcal{N}(\mu,\sigma^2) \rightarrow \boldsymbol{\xi} \sim \mathcal{N}(0,1)$. Since an orthogonal basis is used, the random parameters $\boldsymbol{\xi}$ are required to be uncorrelated in the PCE expansion. If the distributions of $\boldsymbol{u}$ (and consequently, $\boldsymbol{\xi}$) are correlated, then after scaling the distributions to standard form $\boldsymbol{\xi} \sim \mathcal{N}(0,1)$ it is necessary to obtain the Cholesky factor of the correlation matrix. The inverse of the Cholesky factor must then be applied to the scaled distributions to obtain uncorrelated random parameter distributions. A more thorough discussion

on such transformations for normal distributions and other types of distributions can be found elsewhere.[92]

The calculation of β_i is the focal point of constructing the expansion. One approach to calculating these coefficients is the stochastic Galerkin method,[94,100] which is an analytical (intrusive) method. This approach has been applied successfully to systems governed by stochastic nonlinear ODEs and PDEs.[100] However, in all cases closed-form equations were available for the application of stochastic Galerkin method, rendering it inadequate for stochastic multiscale simulations that do not have closed-form expressions in at least one of the length and time scales. Numerical (non-intrusive) methods can thus be used to calculate the expansion coefficients for PCE. The standard techniques are Least Squares (LSQ) regression and Non-Intrusive Spectral Projection (NISP). The former method employs the conventional linear algebra techniques which aim to find the model coefficients β_i by minimizing the least-squares errors.[38,92] On the other hand, the NISP method leverages the orthogonality property of the basis polynomials $\Psi(\boldsymbol{\xi})$ for calculating the expansion coefficients β_i. The application of the orthogonality property to both sides of the PCE expression allows to isolate each expansion coefficient;[38] the generalized formulation to compute i^{th} expansion coefficient β_i is given by:

$$\beta_l = \frac{\int_\Omega \boldsymbol{y}(\boldsymbol{\xi})\Psi(\boldsymbol{\xi})w(\boldsymbol{\xi})d\boldsymbol{\xi}}{\int_\Omega \left(\Psi(\boldsymbol{\xi})\right)^2 w(\boldsymbol{\xi})d\boldsymbol{\xi}} \tag{2}$$

where Ω refers to the support range[92] of the weighting function w. The type of orthogonal basis dictates the choice of the weighting function. The types of bases and the appropriate weighting functions have been tabulated previously.[92,100] The values of the uncertain parameters used to obtain the observable variable $\boldsymbol{y}$ from experiment or simulation must correspond to the roots of a polynomial from the orthogonal basis. Note that the choice of the polynomial and the roots can

influence the accuracy of the PCE NISP method.[38] The dimensionality of the integration in equation ((2)) is equal to the number of uncertain parameters involved in the i^{th} term of the PCE expression. The integrals can be approximated using numerical methods such as Gaussian quadrature or Radau quadrature.[108]

When applied to stochastic systems, the calculation of PCE coefficients separates the random and deterministic components of the contributions of the uncertain parameters to the value of the estimated observable.[107] In addition, the PCE coefficients can be used to calculate variance analytically for any expansion order.[100] Also, the use of an orthogonal polynomial basis allows to calculate the expansion coefficients at a low computational cost by the NISP method. This is due to the fact that NISP allows to drastically reduce the number of realisations of the uncertain parameters by limiting them to the number of roots of the polynomial chosen from the basis. However, the very nature of PCE imposes a limitation, since the probability distributions of the uncertain parameters have to be known *a priori*[100] in order to select the proper polynomial basis. In addition, truncating the expansion up to a finite order introduces an error in the estimations[105] and the computational cost associated with constructing PCE rises as the number of uncertain parameters increases. Nevertheless, the last two drawbacks apply to any expansion-based approach (including the Power Series Expansion).

2.2.2 Power Series Expansion

Power Series Expansion (PSE)[96,97] utilizes the sensitivities of the observable variable with respect to the uncertain parameters[37] to construct closed-form expressions that establish a direct relationship between the uncertain parameters and the observable. A general expression for PSE takes the following form:

$$\boldsymbol{y}_{PSE} = \boldsymbol{y}\left(\mu_{u_1} \dots \mu_{u_\infty}\right) + \sum_{i=1}^{\infty} \frac{\partial \boldsymbol{y}}{\partial u_i}\left(u_i - \mu_{u_i}\right) + \sum_{i=1}^{\infty}\sum_{j=i}^{\infty} \frac{1}{2!}\frac{\partial^2 \boldsymbol{y}}{\partial u_i \partial u_j}\left(u_i - \mu_{u_i}\right)\left(u_j - \mu_{u_j}\right) + \dots \qquad (3)$$

where $\mu_{\boldsymbol{u}}$ denote the nominal values of various uncertain parameters $\boldsymbol{u}$. The sensitivities of the observable with respect to the uncertain parameters can be calculated using analytical or numerical differentiation techniques (e.g. finite differences).

The presence of covariance between the uncertain parameters does not add an extra step in the derivation of PSE (which is not the case for PCE). Thus, PSE is simple and intuitive to implement. When calculating the sensitivities for stochastic systems, the perturbations of the uncertain parameters have to be sufficiently large in order to isolate (or, reduce) the effect of stochastic noise from the observable's sensitivity to changes in the uncertain parameters.[109] That is, the sampled values of the uncertain parameters should be selected such that they reflect an actual change in the observables. Therefore, for systems with stochastic noise, it is possible to improve PSE accuracy by increasing the size of the perturbations while keeping the PSE order fixed. However, the perturbation size is limited by the feasible operating range of the uncertain parameters. Consequently, PSE performance may suffer when applied to systems with large variance of the uncertain parameters. Moreover, if the variance of the uncertain parameters is not known *a priori*, several iterations of the PSE approximations may be required to determine the perturbation size that reduces the stochastic noise. An additional drawback of the PSE approach is that the analytical calculation of variance is only possible for the 1st order expansion.[38] The calculation of the output's variance for higher-order PSE expansions require the simulation of the PSE model. Furthermore, PSE may fall short of yielding accurate results in systems where the relationship between the uncertain parameters and the observable is highly nonlinear.

PSE and PCE are not the only computationally efficient methods of uncertainty quantification in existence. More efficient selection of uncertain parameter realisations can be achieved using Latin hypercube sampling[42,110] and response surface methodology.[41,111,112] A recently developed technique called Multilevel Monte Carlo (MLMC) sampling[113] has shown the ability to provide accurate estimations at computational costs that can be several orders of magnitude less than standard MC sampling. The combination of improved speed and high accuracy motivates its application to chemical engineering systems that are subject to uncertainty.

2.2.3 Multilevel Monte Carlo

Multilevel Monte Carlo (MLMC) sampling method was originally developed in the field of computational finance[113] for calculating an observable's expected value at a reduced computational cost by using samples of different accuracy. Many low-cost low-accuracy samples are used as control estimates, and the expected value is refined with a progressively diminishing number of higher-cost higher-accuracy calculations. The accuracy is controlled by refining the discretization of the integration domain (which is oftentimes the time domain). The general expression for calculating the expected value in the observables using MLMC sampling is as follows:

$$\mathbb{E}[P] = \mathbb{E}[P_0] + \sum_{l=1}^{L} \mathbb{E}[P_l - P_{l-1}] \tag{4}$$

where P is the observable (a random variable), $\mathbb{E}[P]$ is its expected value, P_l is the numerical approximation at the discretization level l (with $l = 0$ corresponding to the coarsest discretization), and L is the total number of discretization levels used in MLMC. The samples P_l and P_{l-1} are computed using subsequent fine and coarse discretization levels, respectively, but they both

correspond to the same stochastic sample (i.e. the same uncertain parameter realisations and the same Brownian path[113] for stochastic systems).

The MLMC method enables one to find Y, the multilevel estimator of $\mathbb{E}[P]$ that has the root mean square error less than a user-defined tolerance ϵ. The MLMC method also provides the maximum bound on the computational complexity associated with finding the estimator. However, this information can be obtained by MLMC only when the conditions of the underlying theorem are met:[114]

Theorem 1. Let P denote a random variable, and let P_l denote the corresponding level l numerical approximation. If there exist independent estimators Y_l based on $\mathcal{N}s_l$ Monte Carlo samples, each with expected cost C_l and variance V_l, and positive constants α_{MLMC}, β_{MLMC}, γ_{MLMC}, c_1, c_2, c_3 such that $\alpha_{MLMC} \geq \frac{1}{2}\min(\beta_{MLMC}, \gamma_{MLMC})$, and

i) $|\mathbb{E}[P_l - P]| \leq c_1 2^{-l\alpha_{MLMC}}$

ii) $\mathbb{E}[Y_l] = \begin{cases} \mathbb{E}[P_0], & l = 0 \\ \mathbb{E}[P_l - P_{l-1}], & l > 0 \end{cases}$

iii) $V_l \leq c_2 2^{-l\beta_{MLMC}}$

iv) $C_l \leq c_3 2^{l\gamma_{MLMC}}$

then there exists a positive constant c_4 such that for any $\epsilon < \exp^{-1}$ there is a total number of levels L and a number of samples at every level $\mathcal{N}_l$ for which the multilevel estimator

$$Y = \sum_{l=0}^{L} Y_l \tag{5}$$

has a mean-square-error with bound

$$\mathbb{E}[(Y - \mathbb{E}[P])^2] < \epsilon^2 \tag{6}$$

with a computational complexity C with bound

$$\mathbb{E}[C] \leq \begin{cases} c_4\epsilon^{-2}, & \beta_{MLMC} > \gamma_{MLMC}, \\ c_4\epsilon^{-2}(\log\epsilon)^2, & \beta_{MLMC} = \gamma_{MLMC}, \\ c_4\epsilon^{-2-(\gamma_{MLMC}-\beta_{MLMC})/\alpha_{MLMC}} & \beta_{MLMC} < \gamma_{MLMC}. \end{cases} \quad (7)$$

It is apparent that Theorem 1 requires the knowledge of many constants. In practice, they are not known and need to be estimated using the results of preliminary sampling of the model.[113,114] It has been suggested that the constants α_{MLMC} and β_{MLMC} can be estimated using linear least-squares regression by constructing the following first order[114] polynomials:

$$\begin{aligned} \log_2|Y_l| &= -\alpha_{MLMC}l + c_\alpha\,, \quad l > 0 \\ \log_2 V_l &= -\beta_{MLMC}l + c_\beta\,, \quad l > 0 \end{aligned} \quad (8)$$

where c_α and c_β are constants obtained by linear regression. The constant γ_{MLMC} can be approximated using the expected computational costs at the finest and second-finest levels of discretization:

$$\gamma_{MLMC} = \log_2 \frac{C_L}{C_{L-1}} \quad (9)$$

where the costs C_{L-1} and C_L represent the time required for the simulation to find Y_{L-1} and Y_L during preliminary sampling (described below). Note that α_{MLMC}, β_{MLMC}, and γ_{MLMC} need to be positive. A negative α_{MLMC} would imply that a finer discretization does not improve the accuracy of calculations of the expected value, a negative β_{MLMC} would indicate that the variance of the estimator is not diminishing, and a negative γ_{MLMC} would show that the computational cost of finer discretization levels is less than the cost of computing coarser samples. Furthermore, note that V_l is not the variance of P, but rather it is the variance of the estimators of P at different discretization levels l. However, MLMC has been extended for the calculation of higher order statistical moments of P.[50,115–118] The maximum entropy method[117] that approximates probability distributions by using MLMC-generated estimates of central statistical moments[116] appears to be

the most promising currently available technique for calculating statistical moments and obtaining confidence intervals. Additional details on the MLMC Theorem can be found elsewhere.[113,114,119] Note that in addition to equation (6), ϵ also plays a role in the calculation of the optimal number of samples at every discretization level:[113]

$$\mathcal{N}_l = \left\lceil 2\epsilon^{-2}\sqrt{V_l h_l}\left(\sum_{l=0}^{L}\sqrt{\frac{V_l}{h_l}}\right)\right\rceil \tag{10}$$

where h_l is the step size used in discretizing the domain (typically, it is reduced according to a geometric series[113,114]). Note that $\mathcal{N}_l$ represents the rounded-up result of the calculation in equation (10). The MLMC algorithm has been summarized below.[114,119]

Preliminary sampling:

Step 1: Set the value of L ($L \geq 2$). Set the coarsest integration step size, h_0, and calculate other step sizes as $h_l = \frac{h_0}{2^l}$ (i.e. a geometric series). Initialize the number of samples to use at the defined levels, Ns_l. Note that in Step 1, Ns_l will be the same number for all levels; however, this will change from level to level as MLMC sampling algorithm continues. Set the value of ϵ ($\epsilon < \exp^{-1}$).

Step 2: Obtain the estimates of $\mathbb{E}[Y_l]$, V_l and C_l by sampling the model and tracking the time spent on calculations at each level. Note that since Ns_l is the same for all levels because it has not changed since Step 1, the changes in C_l are exclusively due to the changes in discretization among the levels.

Step 3: Approximate α_{MLMC}, β_{MLMC} and γ_{MLMC} using equations (8) and (9). MLMC can be applied only if all constants are positive and $\alpha_{MLMC} \geq \frac{1}{2}\min(\beta_{MLMC}, \gamma_{MLMC})$, as per Theorem

1. If the conditions are not met, increasing L and/or Ns_l in Step 1 may help to meet the conditions by providing better estimates of α_{MLMC}, β_{MLMC} and γ_{MLMC}.

Prior to proceeding with the *MLMC approximation* (i.e. Steps 4-8 below), reset the value of L to 2. The reset helps to prevent unnecessary sampling if the estimator will converge at a level number smaller than what was chosen for Step 1. The obvious disadvantage of preliminary sampling is that too many samples may have been used for (and too much time has been spent on) some levels, especially the finer discretizations. However, this is likely to happen only for systems that are not computationally intensive.

MLMC approximation:

Step 4: Calculate N_l at every defined level (from 0 to L) using the estimates of V_l, and the values of h_l and ϵ (see equation (10)).

Step 5: For every level where N_l is greater than the corresponding Ns_l, conduct additional sampling to obtain the total of $\mathcal{N}_l$ samples at the level. Update Ns_l to track the total number of samples performed at every level.

Step 6: Update the estimates of V_l and repeat Step 4 (i.e. update $\mathcal{N}_l$ at every level).

Step 7: If $\mathcal{N}_l$ has not increased by more than 1% from Step 4 to Step 6 for every level, proceed to Step 8. Otherwise, go back to Step 5.

Step 8: Check for the robust convergence[114] of $\mathbb{E}[P]$, i.e., if the condition

$$\frac{2^{\alpha_{MLMC}(l-L)}}{2^{\alpha_{MLMC}-1}}\mathbb{E}[Y_l] \leq \frac{\epsilon}{\sqrt{2}} \tag{11}$$

is satisfied for all three finest discretization levels (i.e. $l = L-2$, $L-1$, and L), finish MLMC sampling and calculate $\mathbb{E}[P]$ as per equation (4). Otherwise, add a new discretization level: set $V_{L+1} = \frac{V_L}{2^{\beta_{MLMC}}}$, set Ns_l for the new level to 0, calculate h_{L+1} as $\frac{h_0}{2^{L+1}}$, set $L = L+1$, and go back to Step 4.

Extension to multiple observables:

To enable the MLMC algorithm to estimate multiple observables simultaneously, the following modifications can be made to the steps described above:

Step 2: Obtain the estimates of $\mathbb{E}[Y_l]$ and V_l for every observable.

Step 3: Estimate α_{MLMC} and β_{MLMC} for every observable. All constants need to be positive and the condition $\alpha_{MLMC} \geq \frac{1}{2}\min(\beta_{MLMC}, \gamma_{MLMC})$ needs to be satisfied for every observable.

Step 4: For each observable and each level, calculate the corresponding $\mathcal{N}_l$ values as per equation (10). For each level, find the maximum $\mathcal{N}_l$ among the values calculated at that level for all the observables and set $\mathcal{N}_l$ for the level to the maximum value.

Step 8: Check the convergence of each observable's estimator using the α_{MLMC} value that corresponds to the observable (calculated in Step 3). If condition (11) is not satisfied for at least one observable, define another discretization level. For each observable, set $V_{L+1} = \frac{V_L}{2^{\beta_{MLMC}}}$ using the corresponding β_{MLMC} (calculated in Step 3). Proceed with the rest of Step 8 in the same way as for the case of one estimated observable.

The main limitation of the MLMC method is that the algorithm is heuristic and is not guaranteed to converge.[114] However, MLMC has been shown to converge and perform well in numerous applications.[114,119–121] Another limitation is that the MLMC Theorem requires many constants to

be known or approximated,[114] which adds the need for preliminary sampling when these constants are not available. However, the computational time for MLMC has been shown to be orders of magnitude less than for standard MC sampling,[113,114] which compensates for the time spent on preliminary sampling. Another advantage of MLMC is that it does not rely on *a priori* knowledge about the shapes of probability distributions associated with the uncertain parameters, unlike PCE, which is applicable to a limited number of probability distributions that must be known in advance of constructing the expansion.[92,101] Furthermore, the MLMC Theorem provides information on the difference between the approximated and true values. Thus, if the conditions of the Theorem are met and MLMC converges, the estimate will be within the imposed error tolerance from the true value. By contrast, the standard MC sampling, PSE and PCE make no such claims and may converge to a local minimum or maximum that can differ substantially from the true value.

Since the original publication,[113] the MLMC method has been expanded to estimating the expected values of various types of Ordinary Differential Equations (ODEs) including stochastic differential equations,[114,122–124] continuous time Markov chains,[125] computing mean exit times,[126] central statistical moments of arbitrary order[116] and even probability density functions.[50,115,117,118] Furthermore, MLMC has been applied to uncertainty analysis of flows through highly heterogeneous porous media,[29,30,49,50,121,127–129] estimation of cumulative distribution functions in stochastic oil reservoir simulations, statistical solutions of the Navier-Stokes equations,[120] reliability analysis of engineered systems[119] and surfactant/polymer enhanced-oil-recovery.[49–51] In all cases, improvements in computational costs associated with obtaining accurate results have been reported. However, a review of the available literature appears to show a gap in knowledge. MLMC has not been applied in the field of chemical engineering beyond the topics of flows in porous media and enhanced-oil-recovery, which motivates the work presented in Chapters 4 and

5 of this thesis, where MLMC is applied to standard chemical engineering systems and stochastic multiscale processes, respectively.

2.3 Data-Driven Models

Advanced sampling techniques such as those described in section 2.2 enable the efficient estimation of probability distributions of the observables of stochastic multiscale models. However, the uncertainty in the observables may be time-varying and may also change at different values of the manipulated variables, and hence new PSE and PCE expressions would have to be identified at each of those settings. To circumvent this challenge, PSE/PCE can be identified offline at various settings prior to implementing online control. This approach has been implemented for propagating distributional uncertainty via PCE in a batch crystallization process, a dynamic model represented by a Volterra series, dynamic flux balance metabolic models, a dynamic model of self-powered neutron detector and other dynamic models.[38,130–138] However, in the case of stochastic multiscale models, analytical expressions for PCE expansion coefficients cannot be obtained by intrusive methods (e.g. Galerkin projection) due to the presence of non-closed-form expressions in the models. Furthermore, the computational expense associated with obtaining the observables necessary to identify PCE coefficients by non-intrusive methods (or to calculate the PSE sensitivities) can be quite large. Thus, the high computational cost may impede PSE and PCE from applications in the online control of stochastic multiscale systems, especially if these expressions need to be identified in real time for an unforeseen operating condition (e.g. in response to a disturbance).

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[139–143] 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.[7,22,34–36,54,143–153] 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.[154]

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.[155] 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.[156]

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.[155–173] 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 such as computational fluid dynamics, food chemistry, renewable energy, bioresources. [31,165,174–192]

As briefly mentioned above, artificial neural networks (ANNs) are information processing systems that are data-driven and can identify complex relationships between the controlled and the response variables in a time series.[155,178,184,193,194] 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:

$$\chi_{\ell} = \varphi\left(\Sigma_{\wp}\left(\omega_{\ell,\wp}\chi_{\wp}\right) + \beta_{\ell,ANN}\right) \tag{12}$$

where χ_{ℓ} is the output of neuron ℓ in the layer of interest, φ is the transfer function (this function can also be called the *activation* or *output* function and it acts on $(\Sigma_{\wp}\left(\omega_{\ell,\wp}\chi_{\wp}\right) + \beta_{\ell,ANN})$, 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 $\mathcal{p}$, $\Sigma_{\mathcal{p}}$ signifies the weighted sum of the outputs of all neurons $\mathcal{p}$ in the previous layer, and $\beta_{\ell,ANN}$ 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 $\Sigma_{\mathcal{p}}(\omega_{\ell,\mathcal{p}}\chi_{\mathcal{p}})$. 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.[155,194]

The values of the weights $\omega_{\ell,\mathcal{p}}$ and the biases $\beta_{\ell,ANN}$ 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 $\omega_{\ell,\mathcal{p}}$ and $\beta_{\ell,ANN}$ 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.[154] 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.[154,195–197] 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 $\omega_{\ell,\mathcal{p}}$ and $\beta_{\ell,ANN}$ that work well only for the seen-before data, a technique called *early stopping* is used for ANN training.[154,198] 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.[165] This challenge can be circumvented when a mathematical model of the process is available, which is the case in the present thesis. 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. Typically, this challenge is mitigated by training the ANN in "open-loop" mode, meaning that each point in the training dataset is provided to the ANN as initial condition for predicting the next point in the time series rather than restricting the ANN to only use its own, possibly quite inaccurate, prediction for the current timepoint as the basis for making the forecast (this training regime is referred to as "closed-loop" mode). In addition, the existence of multiple local optima in the non-convex objective function[154,199] 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.[154]

Recently, ANNs have been applied to a stochastic multiscale model that simulated thin film deposition.[31] 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. That work appears to be one of the first applications of ANNs for stochastic multiscale systems. A search in the literature shows a gap in knowledge since 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 at nominal parameter values and under parametric uncertainty.

2.4 Chapter Summary

Stochastic multiscale systems consist of coupled phenomena that occur at multiple distinct temporal and spatial scales. Such systems are abundant in nature and therefore in applied science disciplines such as chemical engineering, where the aim is to use natural processes to design materials and devices. While the macroscale can be represented well with continuous closed-form equations, the microscale exhibits stochastic phenomena associated with the random motion of atoms and molecules. In order to better control stochastic multiscale processes and improve the

quality of their products, multiscale simulations have been devised that couple continuous equations with non-closed-form expressions that represent the microscale. A variety of coupling schemes exists, but the overarching goal is to leverage the accurate but computationally expensive calculations at fine scales to improve the process control procedures implemented at the macroscale. The computational intensity associated with the calculations at the fine scale raises the overall computational costs associated with stochastic multiscale systems.

Since every process is subject to uncertainty, it is also present in stochastic multiscale systems. The stochastic noise that propagates from the microscale calculations into the observables of these systems makes the application of traditional uncertainty quantification techniques (such as Power Series and Polynomial Chaos Expansions) more challenging since these techniques rely on accurate observable values when calculating their sensitivities and expansion coefficients. A new sampling technique called Multilevel Monte Carlo shows great promise in reducing the computational cost while preserving the accuracy, but its applications in the field of chemical engineering and in particular for stochastic multiscale systems appear limited. Furthermore, data-driven models called Artificial Neural Networks are experiencing a resurgence of research interest and are well-suited to efficiently and accurately predict dynamic responses of observables to perturbations of the manipulated variables. This ability makes them attractive for applications in efficient model predictive control of stochastic multiscale systems, but such studies are also limited in the literature. In order to fully leverage stochastic multiscale models for designing high quality products, uncertainty must be accounted for and efficient control schemes must be implemented.

Chapter 3: Power Series and Polynomial Chaos Expansions for Uncertainty Quantification in Stochastic Multiscale Systems

A review of the literature showed that polynomial chaos expansions (PCE) have not been applied for uncertainty propagation in a multiscale model that couples the continuous and discrete stochastic expressions that do not have closed forms, nor has the performance of power series and polynomial chaos expansions (PSE and PCE, respectively) been compared in this context. In order to address the gap in knowledge, this chapter compares the performance of PSE and PCE in the uncertainty quantification of a stochastic multiscale system subject to single- and multi-parameter uncertainty.

3.1 Synopsis

This chapter examines the effect of noise present in stochastic multiscale systems on the performance of PSE and PCE. A multiscale model of thin film formation by chemical vapour deposition[35] was used to study the effects of single parameter and multivariate uncertainty. Although a particular model was used as a case study, the analysis methodology can be applied to other stochastic multiscale models as well.

In this work, probability density functions of observable variables that exhibit linear and nonlinear dependence on the uncertain parameters have been estimated. To this end, MC sampling of the multiscale model, and the PSE and PCE expressions, were used in the cases with single and multiple uncertain parameters. Different factors affecting the accuracy of PSE and PCE were examined and their impact was analyzed. The statistical moments (mean, variance, skewness and kurtosis) of the observable and confidence interval data were computed and used to assess the performance of PSE and PCE. To demonstrate the efficiency of the uncertainty propagation techniques, PSE and PCE expressions were used in the implementation of constrained robust

optimization. The goal of the optimization study was to maintain the probability densities of observables within pre-specified operational targets by manipulating a macroscale parameter (temperature). The accuracy and computational costs of PSE and PCE have been examined in all cases listed above.

3.2 Thin Film Deposition Model

In order to compare the performance of PSE and PCE techniques for uncertainty quantification in multiscale stochastic systems, one such system needs to be selected for case study purposes. One of the most industrially relevant and widely used multiscale processes is thin film formation by chemical vapour deposition.[13,16] A model of this process has been developed previously[1,13,35] and has been selected for this work. An overview of multiscale model is described in the remainder of this section. More details about this model can be found elsewhere.[1,13,35]

The multiscale model consists of the microscale kinetic Monte Carlo (KMC) simulation of thin film formation by chemical vapour deposition coupled with the macroscale gas-phase partial differential equation (PDE) model that captures the mass transfer in the gas boundary layer above the film.[1,13,22,34–36] The gas boundary layer's metrics (concentration, temperature, mole fraction profile) differ from the uniform metrics of the gas in the bulk because the deposition process consumes the precursor molecules from the gaseous phase. A diagram of the model has been presented in Figure 1. The model parameters employed for simulation and testing of the proposed uncertainty quantification techniques are listed in Table 1.

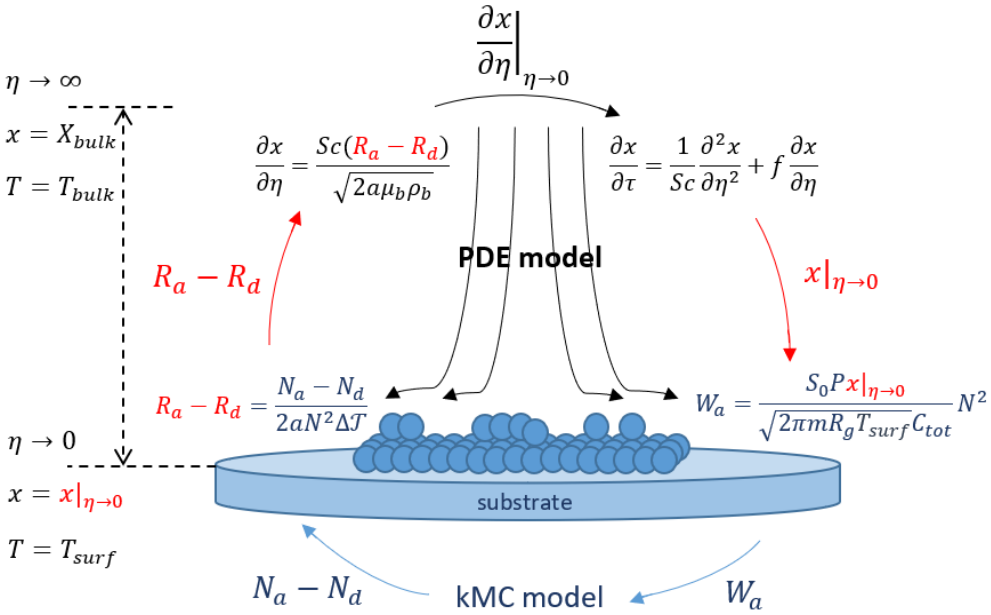

Figure 1: Diagram of the multiscale simulation of thin film formation by chemical vapour deposition

Table 1: Model parameter values (adapted from Rasoulian and Ricardez-Sandoval[35])

Parameter	Symbol	Value and Unit
Density of the bulk divided by gas boundary layer density	ρ_b/ρ	1
Substrate temperature	T_{surf}	800 K
Mole fraction of the chemical species in the bulk	X_{bulk}	2×10^{-6}
Schmidt number of the depositing chemical species	Sc	0.75
Hydrodynamic strain rate	a	5 s^{-1}
Viscosity of the bulk multiplied by its density	$\mu_b\rho_b$	9×10^{11}
Concentration of sites on the film surface	C_{tot}	1.6611×10^{-5} sites · mol/m^2
Sticking coefficient	S_0	0.1
Chamber pressure	P_{CVD}	10^5 Pa
Precursor molecular weight	m	0.028 kg/mol
Event frequency constant	k_{d0}	10^9 s^{-1}
Energy associated with a single bond	E	17000 cal/mol
Energy required for desorption	E_d	17000 cal/mol
Energy required for migration	E_m	10200 cal/mol
Coupling time interval	$\Delta\mathcal{T}$	0.10 s

3.2.1 Continuum (Macroscale) Model

The continuum approach is used to model the gas-phase under the axisymmetric flow assumption by the following momentum, energy and mass transfer equations,[15] respectively:

$$\frac{\partial}{\partial \tau}\left(\frac{\partial f}{\partial \eta}\right) = \frac{\partial^3 f}{\partial \eta^3} + f\frac{\partial^2 f}{\partial \eta^2} + \frac{1}{2}\left[\frac{\rho_b}{\rho} - \left(\frac{\partial f}{\partial \eta}\right)^2\right] \tag{13}$$

$$\frac{\partial T}{\partial \tau} = \frac{1}{Pr}\frac{\partial^2 T}{\partial \eta^2} + f\frac{\partial T}{\partial \eta} \tag{14}$$

$$\frac{\partial x}{\partial \tau} = \frac{1}{Sc}\frac{\partial^2 x}{\partial \eta^2} + f\frac{\partial x}{\partial \eta} \tag{15}$$

where τ is the dimensionless time ($\tau = 2at$ with a as the hydrodynamic strain rate and t as time), f is the dimensionless stream function, η is the dimensionless distance away from thin film surface, Pr is the Prandtl number, and x is the mole fraction of the depositing chemical vapour species. Equations of momentum and energy transfer, (13) and (14), respectively, are solved at steady state at the beginning of the simulation. Thus, only the mass transfer expression shown in equation (15) needs to be solved repeatedly during the simulation until a final batch time is reached. Equations (13)-(15) are subject to the following boundary conditions in the bulk ($\eta \to \infty$):

$$T = T_{bulk} \tag{16}$$

$$\frac{\partial f}{\partial \eta} = 1 \tag{17}$$

$$x = X_{bulk} \tag{18}$$

and at the thin film surface ($\eta \to 0$):

$$T = T_{surf} \tag{19}$$

$$f = 0 \tag{20}$$

$$\frac{\partial f}{\partial \eta} = 0 \tag{21}$$

$$\frac{\partial x}{\partial \eta} = \frac{Sc(R_a - R_d)}{\sqrt{2a\mu_b\rho_b}} \tag{22}$$

where T_{bulk} is the bulk temperature of the chemical vapour, T_{surf} is the temperature of the substrate and the thin film, and R_a and R_d correspond to the rates of adsorption and desorption, respectively. The continuum simulation scale cannot capture the changes in R_a and R_d and thus needs to be informed by the KMC model. Note that the present multiscale model assumes that the changes in gas concentration along the radial direction can be neglected. In practice this uniformity can be achieved using a gas distributor.[35]

3.2.2 Discrete Kinetic Monte Carlo Model

The microscale KMC model employed in this work is an on-lattice solid-on-solid model. A cubic lattice represents the thin film structure. In this model, an atom on the thin film surface can have up to five neighbours: one underneath and up to four in the same layer. Secondary neighbours are not taken into account. As in previous works,[28,29,41] three events are considered to occur on the surface, i.e. adsorption, desorption and migration. The rates of the events, W_a, W_d and W_m, respectively, can be estimated using the kinetic theory of ideal gases and described by the following equations:[15]

$$W_a = \frac{S_0 P_{CVD} x}{\sqrt{2\pi m R_g T} C_{tot}} N^2 \tag{23}$$

$$W_d = \sum_{i=1}^{5} M_i k_{d0} e^{-\frac{iE+E_d}{R_g T}} \tag{24}$$

$$W_m = \sum_{i=1}^{5} M_i k_{d0} e^{-\frac{iE+E_m}{R_g T}} \tag{25}$$

where N is the number of atoms along an edge of the square thin film (and N^2 is the total number of available adsorption sites), R_g is the gas constant, i is the number of neighbours an atom has, and M_i is the number of adsorbed surface atoms with a particular number of neighbours (between one and five). Periodic boundary conditions have been employed to simulate bulk behaviour. The

choice of the event (adsorption, desorption or migration) is determined by the rates of the events (W_a, W_d, W_m) and a uniform random number selected from the range $[0,1)$.[13] Whenever an event is performed, the kMC time is incremented according to equation (26), which relies on another uniform random number ς from the interval (0,1) and the total rate of the system:[13]

$$\Delta t_{kMC} = -\frac{\ln \varsigma}{W_a + W_d + W_m} \tag{26}$$

As shown in Figure 1, the phenomena at the two scales affect each other. The mole fraction of precursor in the gaseous phase (x) will affect the rate of adsorption of atoms onto the film surface. However, the adsorption of atoms onto the surface decreases the concentration of the precursor in the gas boundary layer, while desorption of atoms from the surface does the opposite. During the KMC simulation, the numbers of adsorbed and desorbed atoms, N_a and N_d, are tracked in order to enable the calculation of rates of adsorption and desorption:

$$R_a - R_d = \frac{N_a - N_d}{2aN^2\Delta\mathcal{T}} \tag{27}$$

where $\Delta\mathcal{T}$ is the coupling time interval. Equation (27) allows for the kMC simulation to inform the continuum boundary condition, as shown in equation (22). Once the kMC time reaches $\Delta\mathcal{T}$, the coupling between the microscale and continuum models is carried out through updating the boundary condition, as depicted in Figure 1. Additional details of the implementation can be found in previous studies.[1,13,35]

The roughness R, thickness Th and growth rate Gr of the thin film are considered as the observables of the present multiscale deposition model. The roughness can be calculated in ML (monolayers) as the number of broken bonds on the surface of the film:[4]

$$R = 1 + \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\left|h_{i+1,j} - h_{i,j}\right| + \left|h_{i-1,j} - h_{i,j}\right| + \left|h_{i,j+1} - h_{i,j}\right| + \left|h_{i,j-1} - h_{i,j}\right|\right)}{2N^2} \quad (28)$$

Similarly, thickness of the film can be calculated (also in ML) as the average lattice site height:[2,35]

$$Th = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} h_{i,j} \quad (29)$$

The growth rate can be calculated in ML/s:

$$Gr = \frac{N_a - N_d}{N^2 \Delta\mathcal{T}} \quad (30)$$

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). As shown, the observables depend on the time evolution events occurring on the thin film surface, which also depends on the macroscale behaviour of the reactor.

3.3 Uncertainty Analysis

Since the multiscale model uses a stochastic technique to simulate the molecular scale behaviour, its output will vary even when the same uncertain parameter realisations are used in the simulations. Thus, the results will be subject to stochastic noise, the effect of which is amplified for low order lattices (e.g. 30×30). A larger surface lattice can be used to diminish the noise while increasing the computational costs; e.g. at $T_{surf} = 800\ K$, a CPU time of 37 min was needed to reach the simulation batch time of 100 s for 100×100 lattice, whereas a simulation of 150×150 surface lattice required 2.1 h (on an Intel® Core™ i7-4770 CPU @ 3.40 GHz with 16 GB of RAM). However, Figure 2 illustrates that averaging the results of 6 lattices with $N = 30$

can be used to diminish the noise while maintaining computational efficiency (total CPU time for the six lattices was 5.3 min), which is in agreement with literature.[2,35] Note that previous works have not observed finite size effects for $N = 30$ lattice size.[6,200] As a result, multiples of 6 simulations with 30×30 lattices have been used in this work to identify the PSE and PCE expressions. For every realisation of the uncertain parameters, at least 6 duplicate simulations were conducted. Their final roughness and thickness values were collected and averaged to reduce the stochastic noise in the data.

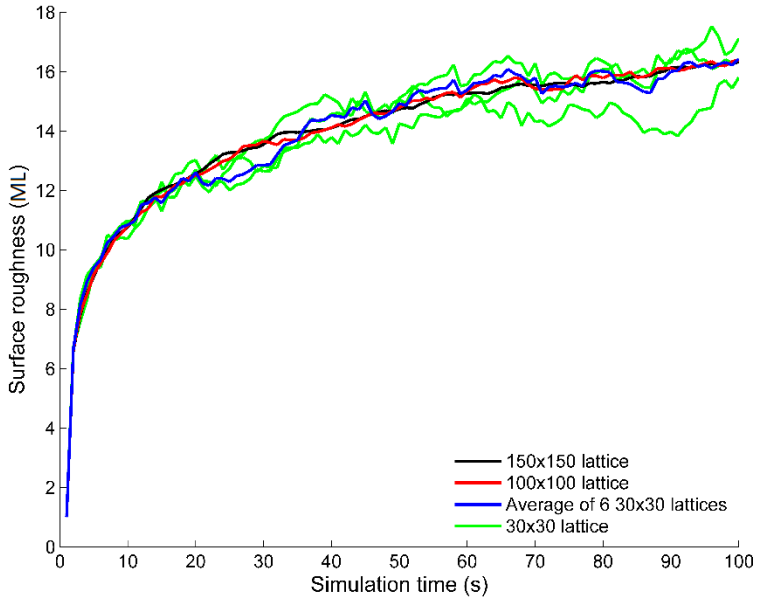

Figure 2: Data from high-order lattices can be approximated by an average of 6 low order lattices (data from each low order lattice has been plotted as green lines)

The uncertain parameters in this study were assumed to be normally distributed, similar to previous works.[34,35,38] Hence, Hermite polynomials (and the corresponding weighting function w as per equation (2)) were used as the orthogonal basis for the PCE method.[92,100,101] For PCE, the uncertain parameter values were transformed to their corresponding random parameter values ξ.

For the PSE approach, the sensitivities were approximated using finite differences, but special care had to be taken in the selection of the size of the perturbations to the nominal values of the uncertain

parameters. The perturbations to the parameters were set to be at least two times the standard deviations of their respective probability distributions in order to reduce the effect of stochastic noise.[109] Single parameter and multivariate uncertainty studies were conducted using the model described in the previous section in order to assess the performance of PSE and PCE.

3.3.1 Single Parameter Uncertainty

The first step in comparing the performance of PSE and PCE in uncertainty quantification for stochastic multiscale systems involves few degrees of freedom to simplify the analysis. The multiscale model used in this work exhibited a nonlinear response of roughness to perturbations in the uncertain model parameters considered in this study. Therefore, the greatest benefit of studying a single uncertain parameter results from examining the effect of uncertainty in a parameter most closely related to roughness. Accordingly, the uncertain parameter under scrutiny was E_m, the magnitude of which affects the migration rate of atoms on the thin film surface. E_m was assumed to have a normal distribution with the mean value reported in Table 1 and standard deviation of 510 cal/mol (i.e. 5% of the mean value).

In order to ensure that a representative distribution for roughness was obtained from the multiscale model, 5,000 MC sampling realisations in E_m were selected from its normal distribution. This number of realisations was found to achieve convergence in the mean and variance of the distribution of thin film roughness. The results from these simulations were used as the benchmark to compare the performance of the PSE and PCE approximations. For the present case study, PSE models were constructed up to 3rd order:

$$R_{PSE} = R(\mu_{E_m}) + \frac{dR}{dE_m}(E_m - \mu_{E_m}) + \frac{1}{2}\frac{d^2R}{dE_m^2}(E_m - \mu_{E_m})^2 + \frac{1}{6}\frac{d^3R}{dE_m^3}(E_m - \mu_{E_m})^3 \qquad (31)$$

where μ_{E_m} represents the expected (mean) value of the uncertain parameter. When constructing the PSE expansions, the sensitivities were approximated using the centered finite difference approach. To test the effect that the magnitude of perturbation from the mean value μ_{E_m} has on PSE accuracy, the perturbation magnitudes of 10%, 25% and 50% of the mean value were used for 1st and 2nd order expansions. For 3rd order, using the 50% perturbation caused an infeasible realisation in E_m (0 cal/mol); therefore, this perturbation size was omitted from the analysis.

PCE formulations were also constructed up to the 3rd order. The 3rd order PCE approximation of roughness is given as follows:

$$R_{PCE} = \beta_0 H_0 + \beta_1 H_1(\xi_{E_m}) + \beta_2 H_2(\xi_{E_m}) + \beta_3 H_3(\xi_{E_m}) \quad (32)$$

where H's represent Hermite polynomials (of up to 3rd order). When the Least Squares regression was used to calculate the β_i expansion coefficients (i.e. PCE LSQ), the number of points selected from the distribution of ξ_{E_m} using MC sampling was 12, 24 and 36. When β_i coefficients were calculated by the NISP method, different numbers of Hermite polynomial roots were used for integrating equation (2) using Gaussian quadrature. For the 1st order expansion, the roots of $H_2(\xi_{E_m})$ polynomial were used as the quadrature points for finding all β_i. The roots of $H_2(\xi_{E_m})$ were also used as ξ_{E_m} realisations that were needed to obtain the corresponding roughness values (R_{PCE}) from the multiscale model (i.e. the $y(\xi)$ values in equation (2)). The procedure was repeated using the roots of $H_3(\xi_{E_m})$ and $H_4(\xi_{E_m})$. For 2nd order PCE, the roots of 3rd, 4th and 5th order polynomials were sampled. For 3rd order expansion, the roots of 4th to 6th order Hermite polynomials were used. For all expansions, 6 and 12 duplicates of every realisation of uncertain parameters were used. Note that duplicates were abbreviated as "dupl" in Table 2, and the number of uncertain parameter realisations for PCE LSQ and roots for PCE NISP were denoted as "pts".

Once the expansions described above were constructed, 10 million realizations selected from the distribution of E_m by MC sampling were used to evaluate the corresponding PSE and PCE models on roughness. Statistical moments (mean, μ_R, variance, σ_R^2, skewness s_R and kurtosis k_R) and the data from two-sided confidence intervals of the constructed expansions were compared to the distribution for roughness obtained by MC sampling of the multiscale model.

Table 2: Roughness data in the case of one uncertain parameter (the roughness values are reported in ML)

	μ_R	σ_R^2	s_R	k_R	Confidence level bounds $\alpha = 0.1\%$		CPU time (hr)
					0.05%	99.95%	
MC sampling	16.35	5.32	0.46	3.49	9.58	25.52	7.64
1st order PSE	16.29	5.44	0.00	3.00	8.61	23.98	0.37
10% step, 12 dupl	-0.35%	2.25%	-100%	-14%	-10%	-6.02%	
1st order PCE LSQ	16.60	5.54	0.00	3.00	8.84	24.36	1.10
12 pts, 6 dupl	1.53%	4.13%	-100%	-14%	-7.65%	-4.54%	
1st order PCE NISP	16.22	5.28	0.00	3.00	8.64	23.79	0.37
4 pts, 6 dupl	-0.82%	-0.76%	-100%	-14%	-9.74%	-6.78%	
2nd order PSE	16.33	5.84	0.32	3.14	9.67	25.55	0.28
50% step, 6 dupl	-0.13%	9.68%	-30%	-10%	0.93%	0.11%	
2nd order PCE LSQ	16.49	4.86	0.27	3.10	10.21	24.71	2.20
12 pts, 12 dupl	0.86%	-8.65%	-42%	-11%	6.58%	-3.16%	
2nd order PCE NISP	16.32	4.54	0.12	3.02	9.71	23.75	0.94
5 pts, 12 dupl	-0.22%	-15%	-74%	-13%	1.44%	-6.93%	
3rd order PSE	16.36	5.35	0.39	3.21	10.21	25.45	0.47
25% step, 6 dupl	0.04%	0.55%	-15%	-7.86%	6.57%	-0.25%	
3rd order PCE LSQ	16.48	5.13	0.21	3.54	8.73	25.65	2.20
12 pts, 12 dupl	0.81%	-3.53%	-54%	1.56%	-8.88%	0.53%	
3rd order PCE NISP 6 pts, 12 dupl	16.44	5.31	0.37	2.93	11.07	24.80	1.10

Table 2 summarizes the results obtained for each expansion type (i.e. PSE, PCE LSQ, PCE NISP) and order (1st, 2nd, 3rd) considered in this work. Similarly, Figure 3 provides a qualitative comparison of select results of PSE and PCE approximations of the probability density of roughness obtained from 5,000 MC samples of E_m applied to the primary multiscale model. As shown in Figure 3, the 2nd and 3rd order expansions estimated the density more accurately than 1st order expansions. The distributions from higher order expansions approximated the right-skewed non-normal shape of the probability density obtained by MC sampling (s_R was 0.46 and k_R was 3.49, see the top row of Table 2), whereas all 1st order expansions produced normal probability densities with s_R of 0.00 and k_R of 3.00 (reported in Table 2). The shape of the density from MC sampling indicated a nonlinear relationship between the uncertain parameter and roughness, since the uncertain parameter was normally distributed but roughness was not.

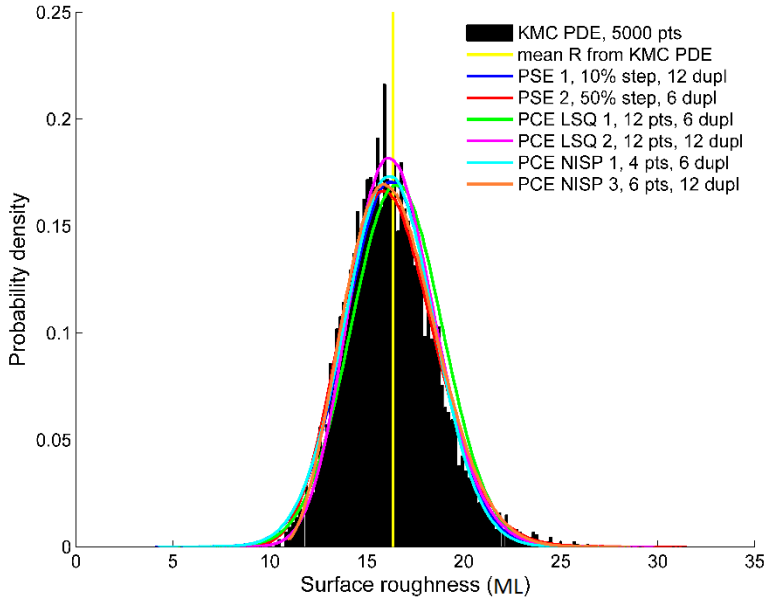

Figure 3: Selected roughness results with one uncertain parameter; "KMC PDE" refers to the multiscale model

The data extracted from the expansions is presented in Table 2 as various statistical moments and confidence interval bounds of roughness, with their corresponding percent deviations from the

results of MC sampling. As mentioned above, different perturbation sizes (referred to as "step" in Table 2) were tested for PSE, and those found to be the most accurate for each PSE order have been presented in Table 2 (the results with inferior accuracy have been omitted for brevity). While the 1st order PSE approximation was more accurate with the smallest step size (10%), higher PSE orders performed best with the largest tested perturbation sizes (50% for 2nd order PSE and 25% for 3rd order PSE). The estimates of confidence level bounds by 1st order PSE exhibited errors greater than 6% for the confidence level $\alpha = 0.1\%$ (see Table 2). Using 2nd order PSE to estimate the bounds on roughness with the same confidence level allowed to reduce the errors to less than 2% (2nd order PSE in Table 2). Increasing the order of PSE from 2nd to 3rd did not improve the estimate. However, the 3rd order PSE errors in estimating the mean and variance of roughness were both lower than those from 2nd order PSE (especially σ_R^2). In addition, the 3rd order PSE decreased the error in the approximation of s_R nearly by a factor of one half compared to 2nd order PSE, and moderately improved upon the estimation of k_R. However, specifying a 3rd order PSE was almost 70% more computationally intensive than identifying a 2nd order PSE; as shown in Table 2, the 3rd order PSE required 0.47 hr of CPU time, while the 2nd order PSE needed only 0.28 hr. If the requirements for the accuracy of estimates of σ_R^2 and s_R are not stringent, then expending the extra CPU time to construct 3rd order PSE may not be necessary.

The percent deviations shown in Table 2 also suggest that 2nd and 3rd order PCE expansions were not able to outperform their corresponding PSE counterparts. Among the 2nd order expansions, PSE demonstrated lower deviations from MC sampling in all cases except when estimating the variance (where 2nd order PCE LSQ was closer to MC sampling result). Among the 3rd order expansions, the only instances when PSE did not provide the most accurate results were in the estimation of σ_R^2 and k_R, for which PCE NISP and PCE LSQ provided more accurate measures,

respectively. Among PSE, PCE LSQ and PCE NISP, only PSE demonstrated a consistent improvement in the estimation of all statistical moments when the expansion order was increased from 2nd to 3rd. However, increasing the order from 2nd to 3rd did not consistently improve the estimates of confidence level bounds for any expansion. In terms of computational costs for 2nd and 3rd order expansions, PSE requirements were the lowest. Using 12 duplicates (instead of 6) did not appear to consistently improve the results of PSE and PCE. The drawback of using 12 duplicates was the doubling of the computational costs, and without a consistent improvement in accuracy it did not appear necessary.

Based on the above, the 2nd order PSE expansion that used 6 duplicates and the 50% perturbation provided the most computationally efficient approximation among all results of Table 2. Among the 1st order expansions, PCE NISP with an adequate number of roots provided similar accuracy to PSE at the same computational expense.

3.3.2 Multivariate Uncertainty

While the single parameter uncertainty analysis provides insight into system behaviour, it is more realistic to consider the case with multivariate uncertainty. The interplay of multiple uncertain parameters adds additional degrees of freedom and complicates the procedure of obtaining the probability distributions of the observables. To study such a scenario and compare the performance of PSE and PCE, two more parameters (in addition to E_m) were assumed to be uncertain: E and X_{bulk}. The parameter E plays a large role in the rate of migration of atoms adsorbed on the film surface, thereby affecting film roughness. The mole fraction of the chemical species in the bulk, X_{bulk}, affects the mass transfer of the depositing species, and is therefore connected to both roughness and thickness. The mean values, μ_E, μ_{E_m} and $\mu_{X_{bulk}}$, of uncertain parameters E, E_m and

X_{bulk}, respectively, were taken from Table 1. The parameters were assumed to be normally distributed with the following variances:

$$\begin{aligned} E &\sim \mathcal{N}(\mu_E, (0.05\mu_E)^2) \\ E_m &\sim \mathcal{N}\left(\mu_{E_m}, \left(0.025\mu_{E_m}\right)^2\right) \\ X_{bulk} &\sim \mathcal{N}\left(\mu_{X_{bulk}}, \left(0.05\mu_{X_{bulk}}\right)^2\right) \end{aligned} \tag{33}$$

A set of 10,000 MC sampling realisations of the uncertain parameters applied to the primary multiscale model was used to obtain a representative distribution for roughness in the multivariate case. In order to estimate the probability distribution for roughness using PSE and PCE, 10^7 realisations of the uncertain parameters were used once the expansions were constructed.

To construct the PSE expansion (as per equation (3)), 25% perturbations ("step sizes") of the nominal values of the uncertain parameters were used for each parameter; centered finite differences were employed to approximate the sensitivities. For PCE LSQ expansion, 50 MC realisations of uncertain parameters were used to construct both 1st and 2nd order expansions. Further, for PCE NISP, the roots of 3rd order Hermite polynomial were used as the quadrature points for calculating the expansion coefficients. As in the case with a single uncertain parameter, these roots were also used as realisations of the uncertain parameters for the multiscale model. In all cases, 6 duplicate simulations were conducted for every realisation of the uncertain parameters.

The estimated probability densities for roughness have been plotted in Figure 4 and the estimated roughness data, along with the computational expenses, are reported in Table 3 (note that the terminology used in Table 3 for its row and column headings is the same as for Table 2).

Table 3: Roughness data in the case of three uncertain parameters (the roughness values are reported in ML)

	μ_R	σ_R^2	s_R	k_R	Confidence level bounds from the probability distributions of roughness								CPU time (hr)
					$\alpha = 0.1\%$		$\alpha = 1\%$		$\alpha = 3\%$		$\alpha = 5\%$		
					0.05%	99.95%	0.5%	99.5%	1.5%	98.5%	2.5%	97.5%	
MC sampling	16.68	17.52	0.84	4.26	7.57	36.85	8.57	30.58	9.48	27.49	10.02	26.21	15.30
1st order PSE	16.36	21.42	0.00	3.00	0.96	31.24	4.14	28.07	6.09	26.12	7.07	25.14	0.64
7 pts, 25% step	-1.94%	22%	-100%	-29%	-87%	-15%	-52%	-8.20%	-36%	-4.98%	-29%	-4.09%	
1st order PCE	17.01	18.86	0.00	3.00	2.45	31.01	5.68	28.02	7.29	26.18	8.21	25.26	4.58
LSQ 50 pts	2.00%	7.68%	-100%	-29%	-68%	-16%	-34%	-8.36%	-23%	-4.76%	-18%	-3.65%	
1st order PCE	17.54	18.14	0.00	3.00	3.34	31.23	6.29	28.28	8.10	26.47	9.01	25.56	0.83
NISP 9 pts	5.14%	3.52%	-100%	-29%	-56%	-15%	-27%	-7.50%	-15%	-3.69%	-10%	-2.48%	
2nd order PSE	16.83	21.85	0.59	3.47	5.70	36.31	7.13	31.05	8.33	28.42	8.81	26.98	1.74
19 pts, 25% step	0.90%	25%	-30%	-19%	-25%	-1.47%	-17%	1.54%	-12%	3.39%	-12%	2.94%	
2nd order PCE	16.76	17.61	0.72	3.70	7.92	35.14	8.82	29.97	9.49	27.49	9.94	26.14	4.58
LSQ 50 pts	0.49%	0.54%	-14%	-13%	4.61%	-4.64%	2.90%	-2.00%	0.15%	0.02%	-0.81%	-0.27%	
2nd order PCE	17.54	18.74	0.64	3.59	7.39	36.07	8.82	31.09	9.76	28.25	10.24	27.06	1.93
NISP 21 pts	5.14%	6.99%	-23%	-16%	-2.31%	-2.13%	2.88%	1.68%	3.01%	2.77%	2.13%	3.24%	

A comparison of MC sampling results in Tables 2 and 3 shows that while the mean roughness values remained somewhat similar (16.35 ML in Table 2 and 16.68 ML in Table 3), all other statistical moments, especially σ_R^2, increased in the multivariate case (Table 3) versus the single parameter scenario (Table 2). This is a clear indication of the effect of multiple uncertain parameter realisations on the multiscale process and the nonlinear relationship between roughness and these parameters.

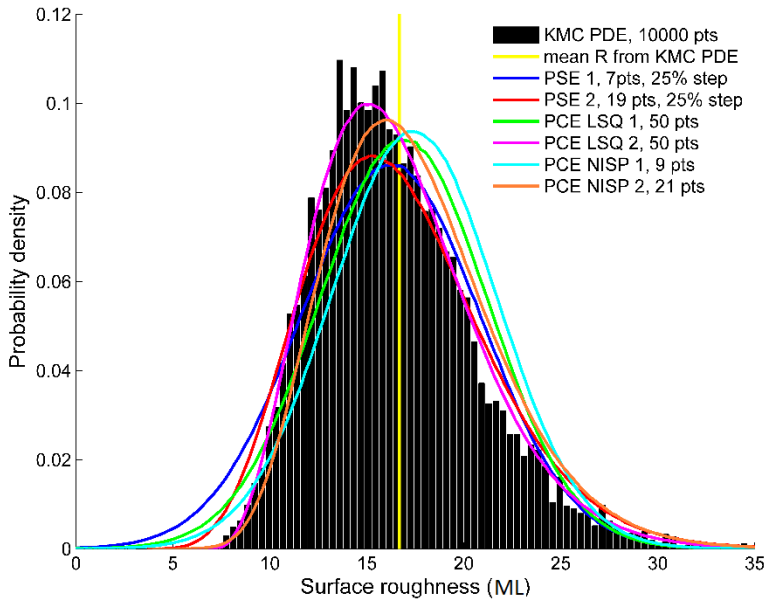

Figure 4: Selected roughness results with three uncertain parameters. The number of duplicates was 6 for all cases.

The qualitative comparison of 1st and 2nd order expansions in Figure 4 shows that for roughness, 2nd order expansions consistently outperformed the 1st order expansions in capturing the shape of the probability density obtained by MC sampling of the multiscale model. The 1st order expansions provided normal probability densities, since all of their predicted s_R and k_R values in Table 3 were 0.00 and 3.00, respectively. The 2nd order expansions yielded non-normal distributions, which can be observed from Figure 4. The s_R and k_R values of 2nd order expansions corroborated the observation of non-normality and also showed that all expansions underestimated skewness and

kurtosis (see Table 3). In addition, it can be observed from Figure 4 that 2nd order PCE estimates exhibited a sharper drop-off at the tails than 2nd order PSE distributions. In Table 3, this was confirmed by more accurate estimates of σ_R^2, s_R and k_R by 2nd order PCE than 2nd order PSE. This was further demonstrated by looking at the confidence level bounds, where 2nd order PSE consistently underestimated the roughness values at the left boundary of every confidence interval (i.e. 0.05%, 0.5%, 1.5% and 2.5% bounds), and the deviations of PSE were significantly larger than 2nd order PCE. As shown in Table 3, the accuracy of the estimations of the right confidence interval bounds (99.95%, 99.5%, 98.5% and 97.5%) was somewhat similar for 2nd order PSE and 2nd order PCE NISP, while 2nd order PCE LSQ approximations changed from being the worst of the three 2nd order expansions at 99.95% and 99.5% confidence level bounds to being the most accurate at α = 3% and α = 5% levels.

A trend that persisted among the estimations of the two-sided confidence interval bounds was that for larger α levels (i.e. closer to the mean of the distribution), the estimations improved. This was especially evident for the 1st order expansions, where the deviations were notably larger at the α = 0.1% level than at α = 3% and α = 5%. It can also be observed from Table 3 that all 2nd order expansions exhibited larger deviations at the 0.05% confidence level bound than at the 2.5% bound. Moreover, 2nd order PCE LSQ underestimated the 99.95% level to a greater extent than the 97.5% level (see Table 3). This, along with the underestimated k_R value, corroborated the assertion made by Lu and coworkers[104] that PCE expansions are not globally accurate (which is also true for PSE), and should therefore exhibit larger errors when the tails of the distribution of the observable are being approximated by PCE and PSE.

Table 3 showed that the 2nd order PCE LSQ exhibited superior accuracy when approximating all statistical moments of roughness, because the percent deviations of 2nd order PCE LSQ were

smaller than from other techniques. While the accuracy exhibited by 2nd order PCE NISP was inferior to PCE LSQ, especially in terms of estimating the statistical moments, the 2nd order PCE NISP method still estimated σ_R^2 significantly better than 2nd order PSE, i.e. the deviations (shown in Table 3) were approximately 7% and 25%, respectively. The lack of accuracy from 2nd order PSE may be because the method of finite differences is not capturing accurately the nonlinear relationships between roughness and the uncertain parameters. Therefore, higher PSE orders might be needed.

Table 3 also summarized the computational requirements. A comparison with Table 2 (single parameter uncertainty) reveals that the computational costs have increased by approximately a factor of 2 for 1st order expansions (except for PCE LSQ, where the increase in CPU time was over 4 times). For the 2nd order expansions, the costs increased by approximately a factor of 4 for PSE and PCE NISP, while for 2nd order PCE LSQ the cost went up by a factor of 2. Thus, a 3-fold increase in the number of uncertain parameters caused the computational costs to rise by 2-4 times for the present analysis.

Among the 2nd order expansions, the most accurate was also the most computationally expensive (PCE LSQ, 4.58 hr). However, 2nd order PCE NISP used the minimum number of uncertain parameter realisations necessary for implementing the expansion and provided an optimal balance between computational efficiency and accuracy. It achieved a level of accuracy similar to 2nd order PCE LSQ, especially when estimating confidence level bounds, using less than half of the time (i.e. NISP required 1.93 hr, as shown in Table 3). Thus, 2nd order PCE NISP can be considered as the most computationally efficient method to arrive at reasonably accurate results in the case of multivariate uncertainty, which was rather different from the conclusions in the previous section. Compared to 2nd order PSE and 2nd order PCE LSQ, the greatest limitation of the PCE NISP

technique was its estimation of the mean roughness value, since the deviation exhibited by NISP was approximately 5% greater than the error observed for PSE and PCE LSQ. The inaccuracy of 2nd order PCE NISP estimation of μ_R is likely due to the stochastic noise, since its estimate of μ_R, i.e. β_0 value (the first term in the expansion), was calculated using only 3 realisations of the uncertain parameters (whereas for 2nd order PCE LSQ, 50 realisations of the uncertain parameters contributed to the calculation of β_0 by Least Squares regression). However, because in PCE NISP the calculation of every coefficient is independent of any other, this issue can be mitigated by using a larger number of quadrature points specifically when calculating β_0. The minimum required number of quadrature points can be used to estimate other expansion coefficients to maintain computational efficiency. Figure 5 demonstrates that the impact of stochastic noise can be diminished by using a larger number of PCE NISP roots when calculating β_0, since the deviation from μ_R obtained by KMC simulations diminishes with more quadrature points. However, increasing the number of NISP roots does not completely eliminate the impact of stochastic noise (see Figure 5). This effect is not often observed in systems comprised entirely of closed-form expressions and is likely due to the absence of closed-form expressions at the discrete scale of the model used in this study.

Thickness distributions have been reported in Figure 6. The film thickness exhibited a linear dependence on the distributions of the uncertain parameters in single and multivariate uncertainty scenarios (Figure 6 shows the latter), and was captured equally well by 1st and 2nd order expansions from all techniques, in harmony with previous works.[34] Quantitative results did not offer additional insight and have been omitted for brevity.

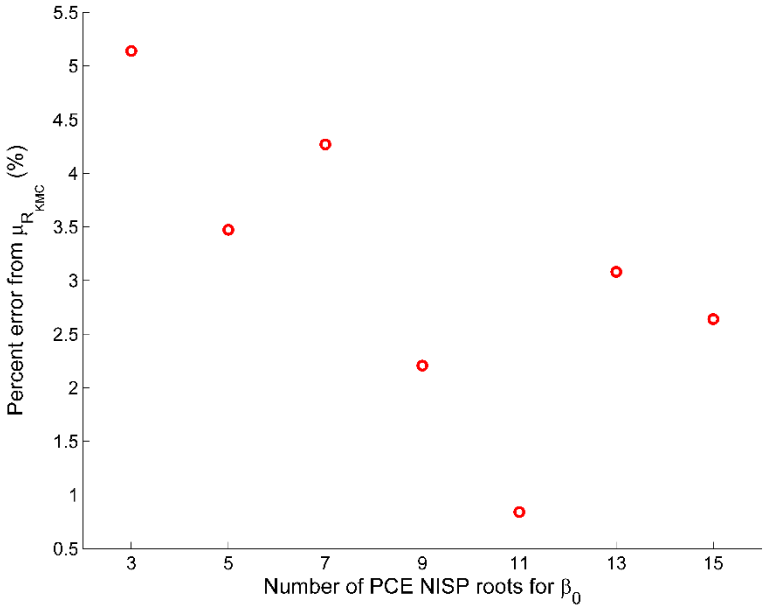

Figure 5: The relationship between the number of PCE NISP roots used to calculate β_0 and percent error from $\mu_{R_{KMC}}$

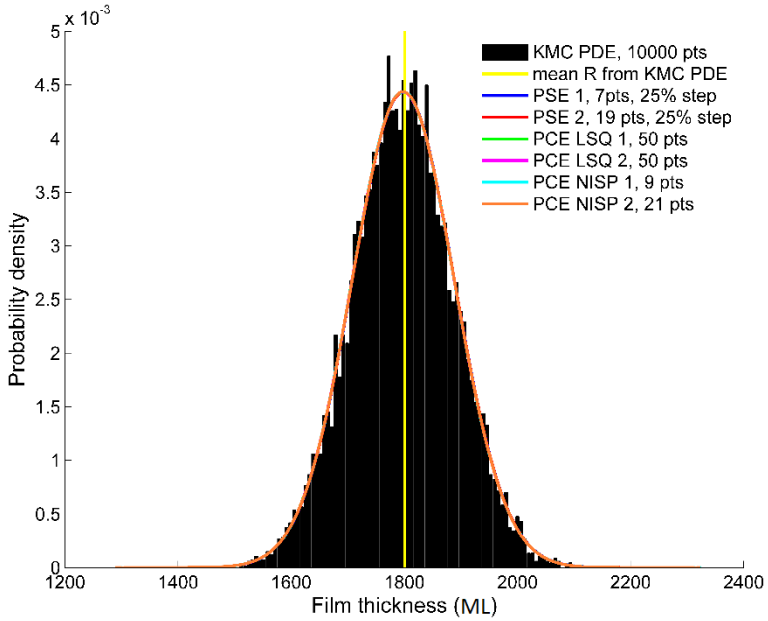

Figure 6: Selected thickness results with three uncertain parameters. The number of duplicates was 6 for all cases

3.4 Robust Optimization

In the case of thin film manufacturing, roughness and thickness strongly affect the device performance; hence, efforts are made to control these observables.[1] A common approach to controlling these parameters is by manipulating the process temperature.[22,34] In the previous section, multivariate PSE and PCE models have been developed at the temperature of 800 K. In this section, PSE and PCE models (with the same uncertain parameters as in the Multivariate Uncertainty section) have been developed at temperature values other than 800 K to establish a relationship between temperature and the coefficients of the low-order approximation models (i.e. PSE and PCE).

Second-order PCE and PSE roughness models and first-order thickness models were identified offline (as per equations (1) and (3), respectively) on the substrate temperature range $T_{surf} \in$ [600 K, 1100 K] with the increments in T_{surf} being 100 K. To identify the models, the same approach was taken as that discussed in the preceding section. The 2nd order PSE model for roughness, constructed at a temperature T, is given by the following expression. Note that in the remainder of this section T and X symbols are used instead of T_{surf} and X_{bulk}, respectively, for brevity:

$$\begin{aligned} R_{PSE}(T,E,E_m,X) = R\left(T,\mu_E,\mu_{E_m},\mu_X\right) + \frac{\partial R}{\partial E}(T)(E-\mu_E) + \frac{\partial R}{\partial E_m}(T)\left(E_m-\mu_{E_m}\right) \\ + \frac{\partial R}{\partial X}(T)(X-\mu_X) + \frac{1}{2}\frac{\partial^2 R}{\partial E^2}(T)(E-\mu_E)^2 + \frac{1}{2}\frac{\partial^2 R}{\partial E_m^2}(T)\left(E_m-\mu_{E_m}\right)^2 \\ + \frac{1}{2}\frac{\partial^2 R}{\partial X^2}(T)(X-\mu_X)^2 + \frac{1}{2}\frac{\partial^2 R}{\partial E\partial E_m}(T)(E-\mu_E)\left(E_m-\mu_{E_m}\right) \\ + \frac{1}{2}\frac{\partial^2 R}{\partial E\partial X}(T)(E-\mu_E)(X-\mu_X) + \frac{1}{2}\frac{\partial^2 R}{\partial E_m\partial X}(T)\left(E_m-\mu_{E_m}\right)(X-\mu_X) \end{aligned} \quad (34)$$

where μ_E, μ_{E_m} and μ_X denote the mean values of the uncertain parameters as shown in equation (33). The 2nd order PCE model for roughness at a temperature T is as follows:

$$R_{PCE}(T,\xi_E,\xi_{E_m},\xi_X) = \beta_0(T)H_0 + \beta_1(T)H_1(\xi_E) + \beta_2(T)H_1(\xi_{E_m}) + \beta_3(T)H_1(\xi_X) + \beta_4(T)H_2(\xi_E) + \beta_5(T)H_2(\xi_{E_m}) + \beta_6(T)H_2(\xi_X) + \beta_7(T)H_1(\xi_E)H_1(\xi_{E_m}) + \beta_8(T)H_1(\xi_E)H_1(\xi_X) + \beta_9(T)H_1(\xi_{E_m})H_1(\xi_X) \quad (35)$$

where ξ_E, ξ_{E_m} and ξ_X are scaled random parameters that correspond to E, E_m and X, respectively. The 1st order PSE and PCE models for thickness, constructed at a temperature T, are given by equations (36) and (37), respectively:

$$Th_{PSE}(T,E,E_m,X) = Th(T,\mu_E,\mu_{E_m},\mu_X) + \frac{\partial Th}{\partial E}(T)(E-\mu_E) + \frac{\partial Th}{\partial E_m}(T)(E_m-\mu_{E_m}) + \frac{\partial Th}{\partial X}(T)(X-\mu_X) \quad (36)$$

$$Th_{PCE}(T,\xi_E,\xi_{E_m},\xi_X) = \beta_0(T)H_0 + \beta_1(T)H_1(\xi_E) + \beta_2(T)H_1(\xi_{E_m}) + \beta_3(T)H_1(\xi_X) \quad (37)$$

After the data for the PSE and PCE models was obtained at all temperatures on the specified range, polynomial models (of up to 3rd order) were used to fit each PSE/PCE model coefficient to temperature, i.e.

$$\omega(T) = \gamma_0 + \gamma_1 T + \gamma_2 T^2 + \gamma_3 T^3 \quad (38)$$

where T is the corresponding temperature value, γ_i are the coefficients of the polynomial model, and $\omega(T)$ represents parameters from the PSE and PCE models that have been fitted to temperature, e.g. $\frac{\partial Th}{\partial E}(T)$, $\beta_i(T)$. As can be seen from equations (34) and (35), the PSE and PCE models for roughness contained 10 coefficients, while the models for thickness (equations (36) and (37)) consider 4 coefficients. A polynomial model (equation (38)) was then fitted for each of the coefficients in the models for roughness and thickness. For brevity, the polynomial coefficients γ_i identified from least squares for each PSE/PCE model coefficient have been presented in the Appendix A. Having access to these models allows the formulation and efficient solution of practical robust optimization problems for thin film deposition.

3.4.1 Case Study 1: Minimizing Temperature Under a Maximum Roughness Constraint

Thin films with lower surface roughness exhibit higher conductivity since the effects of surface scattering of electrons in such films are diminished.[87,88] The first case study in this section aims to find the minimum temperature that will yield a target roughness value at a given confidence level in the presence of multivariate uncertainty. Thus, the temperature is both the decision variable and the objective function in problem (39). The PSE and PCE models together with the polynomial models that related PSE sensitivities and PCE coefficients to temperature were embedded in the optimization. The robust optimization problem involving PSE is as follows:

$$
\begin{aligned}
&\min_{T} \; T \\
&\text{Subject to:} \\
&\quad P(R_{PSE} \leq \delta_R) \geq \delta^* \\
&\quad \text{Uncertain parameter distributions, equation (33)} \\
&\quad \text{Closed-form PSE model for Roughness, equation (34)} \\
&\quad T_l \leq T \leq T_h
\end{aligned}
\tag{39}
$$

In formulation (39), P represents the probability, δ_R is the constraint on roughness (equal to 4.0 ML), δ^* is the probability limit on the constraint ($\delta^* = 95\%$) and R_{PSE} refers to the film surface roughness values predicted by the corresponding closed-form model (PSE and PCE). That is, this constraint ensures that as T is minimized, 95% of the roughness values remain below 4.0 ML in the presence of parametric uncertainty (equation (33)). The values for T_l and T_h were set to 600K and 1100 K, respectively. A similar problem formulation was built for the PCE method and is not shown here for brevity. That is, the corresponding PCE NISP and LSQ models (equation (35)) were used to predict R_{PCE} (instead of R_{PSE}). Three different temperature values were obtained (reported in Table 4), one for each method, i.e. PSE, PCE LSQ and PCE NISP.

Table 4: Case Study 1 - robust optimization to determine the minimum temperature value

Expansion	Optimal T, robust optimization (K)	***R*** value at 95% cumulative probability, validation (ML)	CPU time, robust optimization (s)	CPU time, validation
PSE	1029	4.18 (+4.55%)	113.4	2.00×10^5 s (55.6 hr)
PCE LSQ	1052	3.72 (-7.10%)	202.2	2.68×10^5 s (74.4 hr)
PCE NISP	1034	4.04 (+0.98%)	189.6	2.15×10^5 s (59.7 hr)

Validation of the results was conducted at the three temperature values using the original multiscale model with 3,500 MC realisations in the uncertain parameters. The probability distributions for roughness were examined and values that corresponded to the 95% cumulative probability are reported in Table 4. The results from the validation showed that the robust optimization that employed the PCE NISP model identified a temperature value that satisfied the roughness constraint of problem (39) more closely (with an overestimation by only 0.98%). The temperature value from PSE yielded a roughness value that exceeded the constraint satisfaction by 4.55%, while the resulting temperature of PCE LSQ produced a roughness value that underestimated the desired specification by 7.10%. Figure 7 compares the probability distributions for roughness at 1034 K using the primary multiscale model and 2^{nd} order PCE NISP. The coefficients of PCE NISP have been calculated at 1034 K using the polynomial models employed in robust optimization (see equations (35) and (38)). It can be observed that the estimated probability distribution function captures the shape and the tails of the distribution generated during validation.

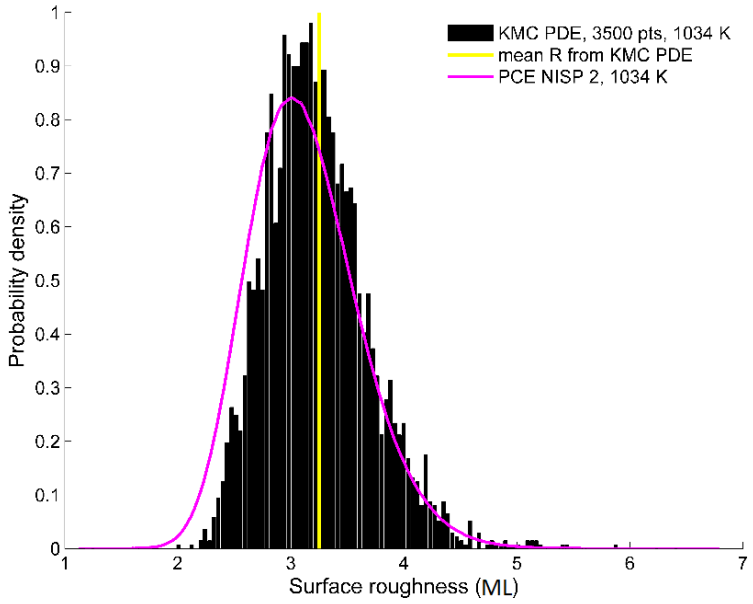

Figure 7: Comparison of the roughness validation results against PCE NISP 2nd order (T=1034 K)

The CPU time requirements have been summarized in Table 4. For each expansion technique, the time for validation using the multiscale model was 3 orders of magnitude greater than the time necessary for the robust optimization to converge to a temperature value. For each PSE and PCE method, the optimization routine converged after 21 to 26 function evaluations. Therefore, if instead of the embedded polynomial models the original multiscale model were used in robust optimization, the time needed for the optimization to converge would become prohibitive; hence the significance of the approach employed in this work.

3.4.2 Case Study 2: Maximizing Thickness Under a Maximum Roughness Constraint

In addition to roughness, thickness also influences thin film conductivity. While excessive roughness is detrimental to conductivity, insufficient thickness has the same effect.[89,90] However, since a lower roughness can be achieved by increasing the temperature, and a higher thickness can be obtained by doing the opposite (for a fixed batch time), selecting a temperature that satisfies both of these constraints in the presence of uncertainty is not a trivial task. Finding the optimal

temperature value that can result in a low roughness film with an adequate thickness will cause the manufacturing process to be less time-consuming and more energy efficient. This robust optimization formulation can be posed as follows:

$$\begin{aligned} &\max_{T} \ \mu_{Th_{PSE}} \\ &\text{Subject to:} \\ &\quad \left(\mu_{R_{PSE}} + \sigma_{R_{PSE}}\right) \leq \delta_R \\ &\quad \text{Uncertain parameter distributions, equation (33)} \\ &\quad \text{Closed-form PSE model for Roughness and Thickness, equations (34) and (36)} \\ &\quad T_l \leq T \leq T_h \end{aligned} \tag{40}$$

In formulation (40), the roughness constraint δ_R has been set to 3.5 ML, $\mu_{Th_{PSE}}$ denotes the mean value of film thickness estimated by the PSE model, while $\mu_{R_{PSE}}$ and $\sigma_{R_{PSE}}$ stand for the estimates of mean and standard deviation of roughness using the PSE model, respectively. The values of T_l and T_h were the same as for case study 1 (section 3.4.1). The same formulation was used for PCE, in which case closed-form PCE models (equations (35) and (37)), combined with the polynomial models (38), were used to provide the estimates for $\mu_{R_{PCE}}$, $\sigma_{R_{PCE}}$ and $\mu_{Th_{PCE}}$.

Three different temperatures were obtained and presented in Table 5 along with the results of optimization and the computational costs. Validation was performed using the primary multiscale model at the PCE NISP temperature (1053 K) with 3,500 MC samples of the uncertain parameters. The resulting mean thickness value obtained from this simulation (1,641.75 ML) was found to be only 0.19% higher than the outcome of robust optimization with PCE NISP (1,638.56 ML), while the validated value of $\mu_R + \sigma_R$ (3.4 ML) was found to be 2.86% lower than the imposed δ_R. Figure 8 shows that roughness probability distribution was skewed and non-normal, and that its shape and tails were captured accurately by PCE NISP. If roughness were normally distributed, at least 84% of the MC sampling results would need to satisfy δ_R. Figure 9 demonstrated that 89.3% of

validation results for roughness satisfied the δ_R constraint. At 1053 K, MC sampling of the primary multiscale model required 94.5 h of CPU time to compute the observables, which was four orders of magnitude greater than the time needed for robust optimization to converge to the optimal temperature using PCE NISP models (13.2 s).

Table 5: Case Study 2 – constrained maximization of thickness

Expansion	Optimal *T*, (K)	$\boldsymbol{\mu_{Th}}$ (ML)	$\boldsymbol{\mu_R}$ (ML)	$\boldsymbol{\sigma_R}$ (ML)	CPU time (s)
PSE	1056	1636.95	3.22	0.28	8.4
PCE LSQ	1064	1630.71	3.06	0.45	15.6
PCE NISP	1053	1638.56	3.06	0.44	13.2

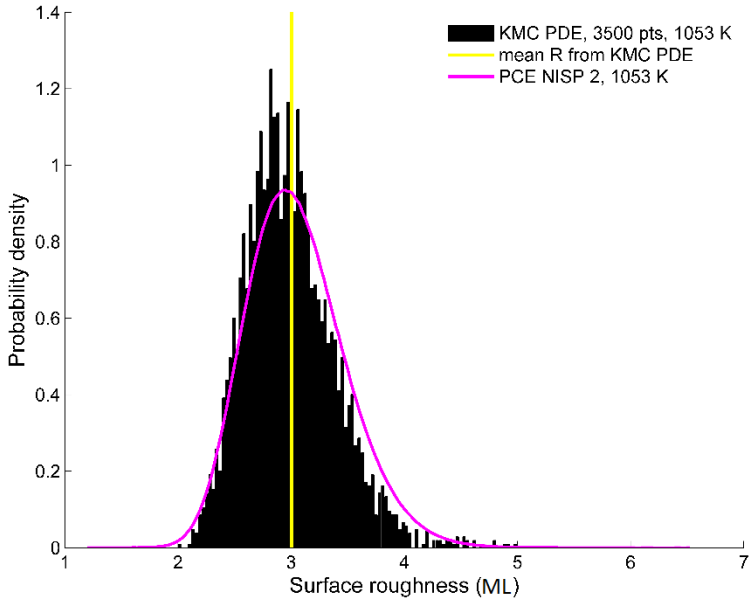

Figure 8: Comparison of the roughness validation results against PCE NISP 2nd order (T=1053 K)

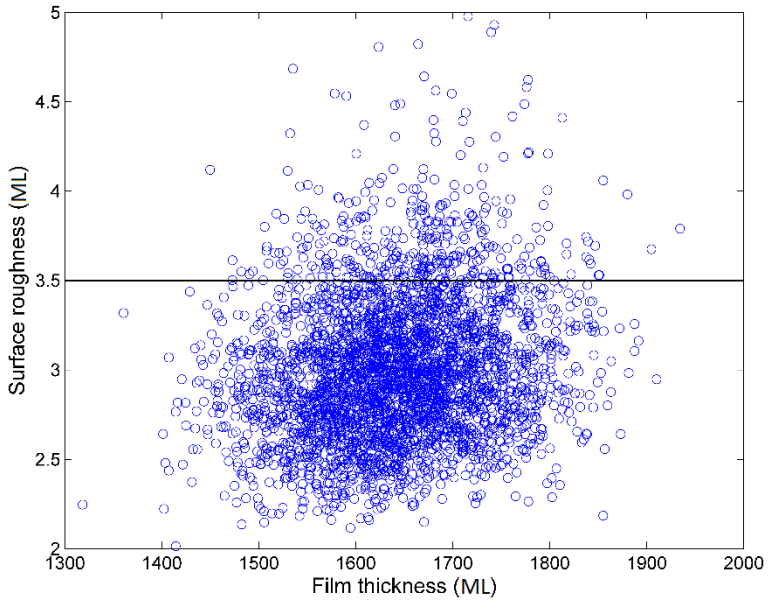

Figure 9: Validated roughness and thickness data satisfied the constraint on roughness (T=1053 K)

3.5 Chapter Summary

This chapter compared the performance of Power Series Expansions and Polynomial Chaos Expansions (calculated via the Least Squares and NISP methods) in the uncertainty analysis of a non-closed-form stochastic multiscale system. While the study used a specific thin film deposition multiscale model, the analysis methodology described herein is general and can be extended to other multiscale systems. In the single uncertain parameter study, the 2nd order PSE was found to be the most accurate and computationally efficient method to predict the probability distribution of an observable. However, in the multivariate case, 2nd order PCE Least Squares provided the most accurate results, while 2nd order PCE NISP provided a similar level of accuracy at half the computational cost. In the multivariate case, the greatest shortcoming of PCE NISP against the other methods considered here was lower accuracy in estimating the mean value of the observable. However, this accuracy was found to improve when more quadrature points (roots) were used for

calculating the expansion coefficient that estimated the mean of the observable value. Using a greater number of roots for this coefficient and the minimum necessary number of roots for the other coefficients in PCE NISP can provide high accuracy while keeping the computational costs low. Nevertheless, the stochastic noise could not be completely eliminated from the estimate of the mean due to the non-closed-form expressions involved in the present multiscale system. Depending on the technique, the computational costs for the present multiscale case study were found to increase by 2-4 times when 3 uncertain parameters were considered in the multivariate uncertainty scenario against the single uncertain parameter case. In order to demonstrate the efficiency of using PSE and PCE for uncertainty propagation instead of MC sampling, the sensitivities and coefficients of multivariate expansions were correlated with temperature and then used in robust optimization case studies. Among the three expansions techniques, PCE NISP was found to be computationally efficient and to yield results which were demonstrated to have satisfied the constraints of the robust optimization problems more closely than the other two methods.

Chapter 4: Multilevel Monte Carlo for the Estimation of Expected Values in Continuous Chemical Engineering Systems

Multilevel Monte Carlo sampling (MLMC) has been demonstrated to provide accurate estimates of the expected values of system observables under uncertainty and save computational time.[113,114] However, as discussed in section 2.2.3, there is a limited number of applications of MLMC for the estimation of expected values in chemical engineering systems subject to uncertainty. To address this gap in knowledge, MLMC has been applied to three standard chemical engineering systems subject to uncertainty and its performance has been compared to traditional uncertainty quantification techniques.

4.1 Synopsis

The aim of the work presented in this chapter is to evaluate the performance of Multilevel Monte Carlo (MLMC) sampling technique for expected value estimation in chemical engineering systems. Initially, the MLMC method is described and a tutorial example is presented, in which MLMC estimated one observable from a mixing tank model subject to two normally distributed uncertain parameters. Afterwards, the performance of MLMC is compared to standard Monte Carlo sampling (MC) and Power Series (PSE) and Polynomial Chaos (PCE) expansions for the estimation of an observable of a wastewater treatment plant subject to multiple uncertain parameters, one of which may be thought of as a model structure error. Following the wastewater treatment plant case study, the performance of MLMC is compared to MC sampling for the simultaneous estimation of four observables of a computationally intensive ternary distillation column model subject to multiple uncertain parameters. Also, the nuances of adapting the MLMC technique to chemical engineering systems and the advantages of using MLMC for uncertainty

quantification are highlighted. Based on a review of the current literature, it appears that this study incorporates the first comparison of MLMC, PSE and PCE techniques.

4.2 Illustrative Case Study: Mixing Tank

For illustration purposes, the MLMC technique was initially implemented on a relatively simple system[201] of a continuously stirred mixing tank that uses a PI controller to maintain the liquid level at a desired setpoint H_{sp}. The model consists of the following equations:[201]

$$A_T \frac{dH}{dt} = F_{in} - F_{out} \tag{41}$$

$$F_{out} = KC_v\sqrt{10^{-3}\rho_w g_w H} \tag{42}$$

where t represents time, A_T is the area of the mixing tank, H is the height of the liquid level, while F_{in} and F_{out} are the inlet and outlet flow rates, respectively. C_v is the outlet valve coefficient, ρ_w is the density of water and g_w is its standard gravity. In this system, it is assumed that the stem position of the outlet valve, K, is manipulated by a PI controller, i.e.

$$K(t) = \overline{K} + K_c\varepsilon + \frac{K_c}{\tau_{i,tank}}\int_0^t \varepsilon dt \tag{43}$$

where $\overline{K}$ is the nominal value of K, K_c is the controller's gain, $\tau_{i,tank}$ is the integral time constant and ε is the error term ($\varepsilon = H_{sp} - H$).

To demonstrate how MLMC can be used for uncertainty propagation in the mixing tank system, C_v and F_{in} were considered as independent normally distributed uncertain parameters. The values for all model parameters have been provided in Table 6.

Table 6: Parameter values for the mixing tank model. Note that $\mathcal{N}$ denotes the normal distribution.

Parameter	Value
A_T	1 m^2
C_v	Uncertain, $\mathcal{N}(25, 5^2)$
F_{in}	Uncertain, $\mathcal{N}(6\ m^3/hr, (0.5\ m^3/hr)^2)$
ρ_w	1000 kg/m^3
g_w	0.981
H_{sp}	8 m
$\bar{K}$	0.1
K_c	-0.01
$\tau_{i,tank}$	0.1 min

The goal was to approximate the expected value of the time-averaged liquid level in the tank, i.e.

$$\mathbb{E}[H^t] = \mathbb{E}\left[\frac{\int_0^{t_f} H(t)dt}{\int_0^{t_f} dt}\right] \tag{44}$$

where t_f was set to 10 hr. The MLMC algorithm for this case study was set up as follows:

Step 1: L was initially set to 5 because this number of levels was observed to produce consistent estimates of α_{MLMC}, β_{MLCM} and γ_{MLMC} when multiple preliminary samplings were conducted, unlike $L = 2$.

The coarse discretization step size, h_0, was set to 0.5 hr. The step sizes from h_1 to h_5 were calculated as $\frac{h_0}{2^l}$. The choice of h_0 magnitude is problem-specific. The guideline is to set h_0 value such that the coarse discretization does not produce excessive variance V_0 in the estimator at $l = 0$ (in this study, $\mathbb{E}[H_0^t]$) through inaccurate numerical integration. When V_0 is significantly large, it causes oversampling of the system by inflating N_0 (see equation (10)). Oversampling diminishes the savings in computational time that MLMC offers. Even though an individual sample is

inexpensive to compute when h_0 is very coarse relative to the entire integration domain, the total number of sample evaluations will be so large as to negate the computational savings offered by MLMC. At the same time, starting MLMC with a small h_0 will also slow down the MLMC algorithm because the computational cost of each sample will be relatively high and will be exacerbated by the total number of samples at the coarser levels, especially at $l = 0$.

Ns_l was set to 100 for all six levels (from $l = 0$ to $l = 5$). This value provides a sufficient representation of H_l^t and V_l values without causing excessive time expenditures on the preliminary sampling. For computationally intensive systems, it may be necessary to use a smaller Ns_l, i.e. 100 may be computationally prohibitive (see the Distillation Column section).

The value of ϵ was chosen to be 10^{-3} m (this magnitude is less than $\exp^{-1}$). For systems that exhibit a large variability of Y_l (i.e. H_l^t in this case study) and/or are computationally expensive, using a very small ϵ may make the approximation of observables computationally prohibitive if their absolute values are orders of magnitude greater than the chosen ϵ.

Step 2: The estimates of $\mathbb{E}[H_l^t]$, V_l and C_l were obtained by drawing MC samples from the distributions of C_v and F_{in} and then simulating the closed-loop mixing tank model (equations (41)-(43)) for each realisation.

Step 3: The α_{MLMC}, β_{MLMC} and γ_{MLMC} values were estimated by linear least-squares regression to be 1.006, 2.837 and 0.176, respectively, using equations (8)-(9). They were all positive and the condition $\alpha_{MLMC} \geq \frac{1}{2}\min(\beta_{MLMC}, \gamma_{MLMC})$ was clearly met. Following the MLMC algorithm described in the previous section, the value of L was reset to 2.

Step 4: Equation (10) was used to calculate the values of $\mathcal{N}_0$, $\mathcal{N}_1$ and $\mathcal{N}_2$ (i.e. in total, there were three currently defined levels). Hence, their corresponding samples were 1,237, 51 and 9, respectively.

Step 5: Since 100 samples have already been performed at levels 0-2, only 1,137 additional samples were done at $l = 0$. Hence, Ns_0 was updated to 1,237, while Ns_1 and Ns_2 were kept at 100.

Step 6: Having updated V_l using the results of Step 5, the new values for $\mathcal{N}_0$, $\mathcal{N}_1$ and $\mathcal{N}_2$ were 1,147, 49 and 9, respectively.

Step 7: The $\mathcal{N}_l$ values from Step 6 did not exceed $\mathcal{N}_l$ from Step 4. Therefore, proceed to Step 8. Had the updated $\mathcal{N}_l$ values exceeded those from Step 4 for any l, it would have been necessary to return to Step 5.

Step 8: The current approximations of $\mathbb{E}[H_0^t]$, $\mathbb{E}[H_1^t]$ and $\mathbb{E}[H_2^t]$ were 7.9157 m, 0.0377 m and 0.0188 m, respectively. Condition (11) did not hold for any of the values. Therefore, set $V_3 = V_2/2^{2.837}$, set Ns_3 to 0, calculate h_3 as $\frac{h_0}{2^3}$, set $L = 3$, and go back to Step 4.

Steps 4-8 were repeated until $L = 7$ was reached. During the procedure, an additional 20 samples were drawn from the distributions of the uncertain parameters for $l = 0$ (1,257 total samples at $l = 0$, $\mathbb{E}[H_0^t]$ changed to 7.9153 m). The values of $\mathcal{N}_l$ and $\mathbb{E}[H_l^t]$ were reported in Table 7. The multilevel estimator of $\mathbb{E}[H^t]$ was 7.9906 m, calculated by adding the individual $\mathbb{E}[H_l^t]$ values in Table 7. The entire MLMC approximation was accomplished in 4.01 seconds on an Intel® Core™ i7-4770 CPU @ 3.40 GHz with 16 GB of RAM (the same computer was used for all the case studies presented in this chapter).

Table 7: The results of MLMC approximation for the mixing tank model. Note that the values of $\mathbb{E}[H_l^t]$ are in metres.

l	$\mathcal{N}_l$	$\mathbb{E}[H_l^t]$ (m)
0	1,257	7.9153
1	100	0.0377
2	100	0.0188
3	2	0.0094
4	1	0.0052
5	1	0.0024
6	1	0.0012
7	1	0.0006

For comparison purposes, standard MC sampling was used to determine $\mathbb{E}[H^t]$ using 10,000 realisations of uncertain parameters with the integration time-step of 0.01 hr (roughly equal to $\frac{h_0}{2^2}$). The approximation for $\mathbb{E}[H^t]$ was 7.9881 m, obtained in 26.17 seconds. The results of MC and MLMC have been reported in Figure 10, which shows the distributions and independence of the two uncertain parameters and compares the mean values obtained by MC and MLMC. There is a close agreement of $\mathbb{E}[H^t]$ approximations (with a difference of 0.0025 m), obtained 6.5 times faster using MLMC.

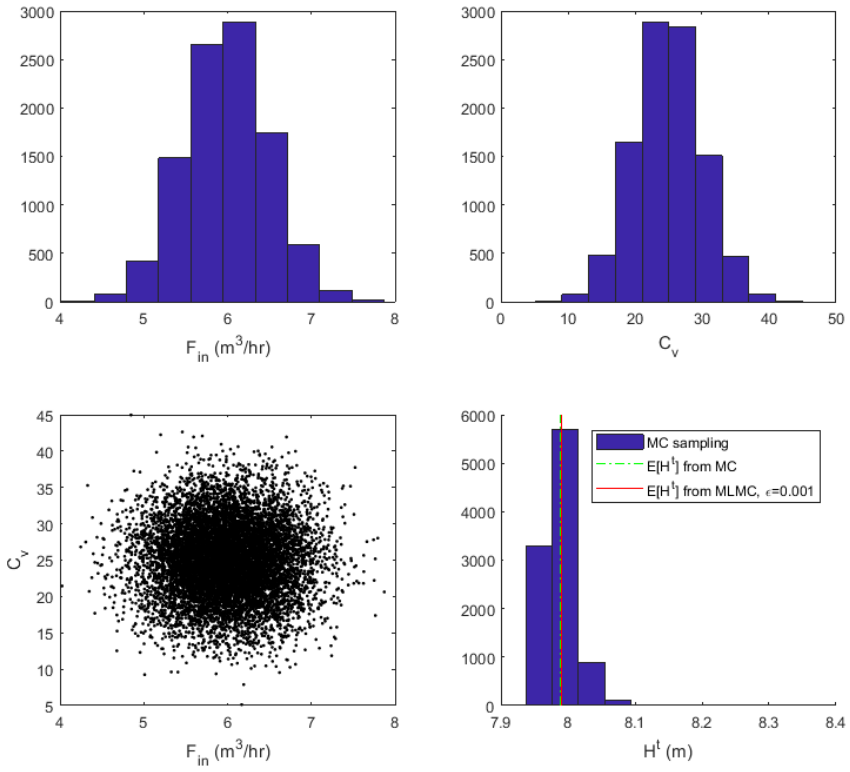

Figure 10: The uncertain parameters and the results of applying MLMC to the mixing tank model.

4.3 Wastewater Treatment Plant

In order to examine the potential benefits of applying the MLMC technique to more computationally demanding chemical engineering systems, the method was applied to a wastewater treatment plant described elsewhere[202,203] (see Figure 11 for a diagram). The aim of this process is to remove organic substrates from municipal wastewater by using a population of microorganisms (referred to as *biomass*). The microorganisms consume the substrates, grow in the bioreactor and form an activated sludge. The effluent from the reactor is passed to the settler, where the sludge is separated by sedimentation and then recycled back to the bioreactor along with organic substrates that were not consumed by the biomass.[203]

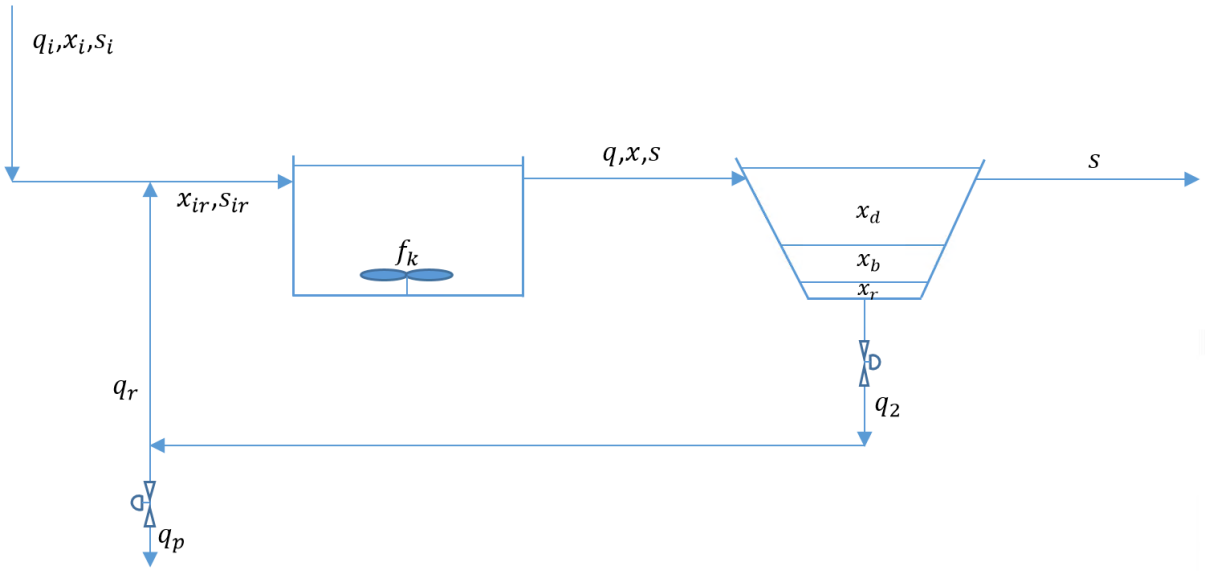

Figure 11: A diagram of the wastewater treatment plant model.

The wastewater treatment plant is represented by the following set of ordinary differential equations:

$$\frac{dx_{bio}}{dt} = \mu_{gr} y_{bio} \frac{x_{bio} s}{k_s + s} - k_d \frac{x^{\varphi}}{s} - k_c x_{bio} + \frac{q}{V_r}(x_{ir,bio} - x_{bio}) \tag{45}$$

$$\frac{ds}{dt} = -\mu_{gr} \frac{x_{bio} s}{k_s + s} + f_d k_d \frac{x^{\varphi}}{s} + f_d k_c x_{bio} + \frac{q}{V_r}(s_{ir} - s) \tag{46}$$

$$\frac{dx_{b,bio}}{dt} = \frac{1}{A_d l_b}(q_i + q_2 - q_p)(x_{bio} - x_{b,bio}) + \frac{1}{l_b}(v s_d - v s_b) \tag{47}$$

$$\frac{dx_{d,bio}}{dt} = \frac{1}{A_d l_d}(q_i - q_p)(x_{b,bio} - x_{d,bio}) - \frac{1}{l_d} v s_d \tag{48}$$

$$\frac{dx_{r,bio}}{dt} = \frac{1}{A_d l_r} q_2 (x_{b,bio} - x_{r,bio}) + \frac{1}{l_r} v s_b \tag{49}$$

$$\frac{dc_w}{dt} = k_{la} f_k (c_s - c_w) - OUR - \frac{q}{V_r} c_w \tag{50}$$

$$OUR = k_{01} \mu_{gr} x_{bio} \frac{x_{bio}}{k_s + s} \tag{51}$$

$$q_r = q_2 - q_p \tag{52}$$

$$q = q_i + q_r \tag{53}$$

$$q_i = 100\,sin(\omega t) + 500 \tag{54}$$

where t is time ($t \in$ [0 hr, 2000 hr]), x_{bio} is the time-dependent biomass concentration and s is the time-dependent concentration of organic substrates in the bioreactor (all concentrations are in mg/L); q_i, q_r and q stand for flow rates (in m^3hr^{-1}) of inlet, recycled and outlet streams, respectively, q_p is the purge flow rate, $x_{ir,bio}$ represents the combined inlet and recycled concentration of biomass, while $x_{d,bio}$ and $x_{b,bio}$ stand for biomass concentrations in the surface and middle layers in the settler, and s_{ir} denotes the combined inlet and recycled concentration of organic substrates. The surface area of the top layer of the settler is represented by A_d, and vs_d and vs_b are the rates of settling of the activated sludge in the top and middle layers, respectively. The dissolved oxygen concentration is designated by c_w and the speed of aeration turbines is represented by f_k. Note that "OUR" is an acronym for "Oxygen Uptake Rate". The parameters of equations (45)-(54) are listed in Table 8.

The purpose of applying the MLMC technique was to estimate the expected value of the time-averaged substrate concentration at the outlet, $\mathbb{E}[s^t]$, subject to uncertainty. The observable was formulated as follows:

$$\mathbb{E}[s^t] = \mathbb{E}\left[\frac{\int_0^{t_f} s(t)dt}{\int_0^{t_f} dt}\right] \tag{55}$$

where t_f was set to 2,000 hr.

Table 8: Parameters for the simplified model of a wastewater treatment plant.[202,203] *Note that $\mathcal{U}$ denotes the uniform distribution.*

Parameter	Value	Description
μ_{gr}	0.1824 hr^{-1}	Specific growth rate of the microorganisms
y_{bio}	0.5948	Fraction of substrate converted to biomass
k_s	300 hr^{-1}	Saturation constant
k_d	5.0×10^{-5} hr^{-1}	Rate of biomass death by endogenous metabolism
k_c	1.333×10^{-4} hr^{-1}	Rate of biomass death by biological waste
k_{la}	0.7 hr^{-1}	Rate of oxygen transfer into the water
k_{01}	1.0×10^{-4} hr^{-1}	Rate of oxygen demand
V_r	2500 m^3	Volume of the bioreactor
f_d	0.2	Fraction of dead biomass that contributes to the substrate s
q_r	403 m^3hr^{-1}	Flow rate of the recycled stream
$x_{i,bio}$	$\text{Lognormal}\left(80\frac{\text{mg}}{\text{L}}, \left(15\frac{\text{mg}}{\text{L}}\right)^2\right)$	Biomass concentration in the inlet stream
$x_{r,bio}$	5975.82 mg/L	Biomass concentration in the recycled stream
s_i	$\text{Lognormal}\left(366\frac{\text{mg}}{\text{L}}, \left(40\frac{\text{mg}}{\text{L}}\right)^2\right)$	Substrate concentration in the inlet stream
c_s	8.0 hr^{-1}	Oxygen specific saturation
l_d	2 m	Depth of the top layer of the settler
l_b	1 m	Depth of the middle layer of the settler
l_r	0.5 m	Depth of the bottom layer of the settler
ω	$\text{Lognormal}(0.005\ \text{hr}^{-1}, (0.005\ \text{hr}^{-1})^2)$	The frequency of the disturbance in the inlet stream
φ	$\mathcal{U}(1.3,2.7)$	Uncertain exponent of biomass concentration in equations (45) and (46)

Uncertainty was assumed to arise from four uncertain parameters. The value of exponent φ in equations (45) and (46) was taken to be uniformly distributed on the interval [1.3, 2.7] (i.e. $\varphi\sim\mathcal{U}(1.3,2.7)$). This uncertainty alone caused the observable to deviate by 18% from the

observable value obtained when all uncertain parameters were set to their mean values (not shown for brevity). Furthermore, this uncertain parameter may be accounted for as a model structure error because it directly affects the dynamic behaviour of the dependent variable x_{bio}. The other three uncertain parameters considered in the study were $x_{i,bio}$, s_i, and ω. Each of these three parameters was assumed to be lognormally distributed with the mean values and standard deviations as reported in Table 8. Note that although each uncertain parameter was time-invariant, ω appears in equation (54), which is a time-dependent function that determines the variability in the inlet stream's flowrate at any time *t*. Therefore, this variation can be considered as a time-varying disturbance affecting this process.

The root mean square error criterion (ϵ) for MLMC was set to 0.1 mg/L. Levels 0-5, inclusive, with 100 uncertain parameter realisations at each level, were used to estimate V_l for equation (10). The coarsest time domain discretization of MLMC, h_0, was set to 31.25 hr because it provided a reasonable balance between accuracy in the estimates and computational time at $l = 0$ during the MLMC's preliminary sampling phase. The MLMC procedure was performed 20 times and the results have been presented in Figures 12 and 13 as solid horizontal blue lines spanning the integration time-steps used by MLMC, while MC results were presented as circles at the appropriate time-steps. For comparison purposes, MC sampling was conducted with various integration time-steps (see the green circles in Figures 12 and 13). MC sampled 10,000 realisations of the uncertain parameters per batch until the relative error between the estimate of $\mathbb{E}[s^t]$ based on all previous batches and the estimate based on all batches (i.e. all previous plus the current batch) was less than 0.1 mg/L. The "Benchmark" values were obtained using MC sampling with small integration time-steps (see Figure 12, black circles) with the relative error convergence criterion of 0.001 mg/L, resulting in 170,000 samples (17 batches of 10,000 samples each) at the

integration time-step of 0.1 hr and 390,000 samples (39 batches of 10,000 samples each) at the integration time-step of 0.01 hr.

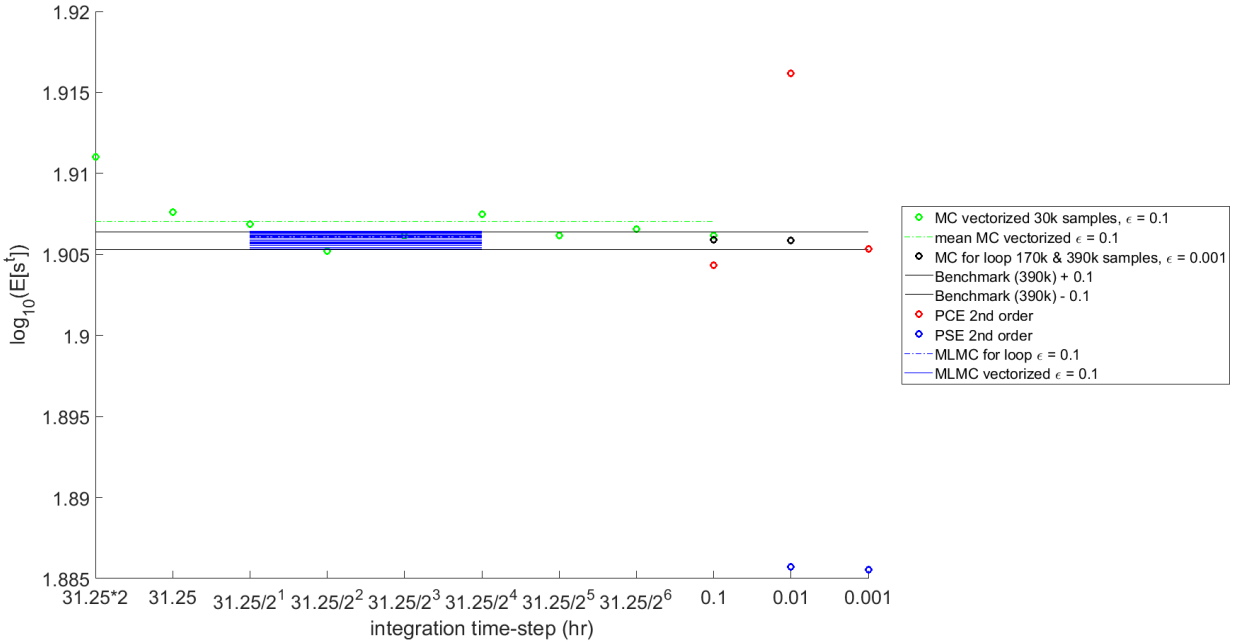

Figure 12: A comparison of the accuracy of MC and MLMC estimates for the wastewater treatment plant model. Note that the vertical axis is expressed as a logarithm (for clarity).

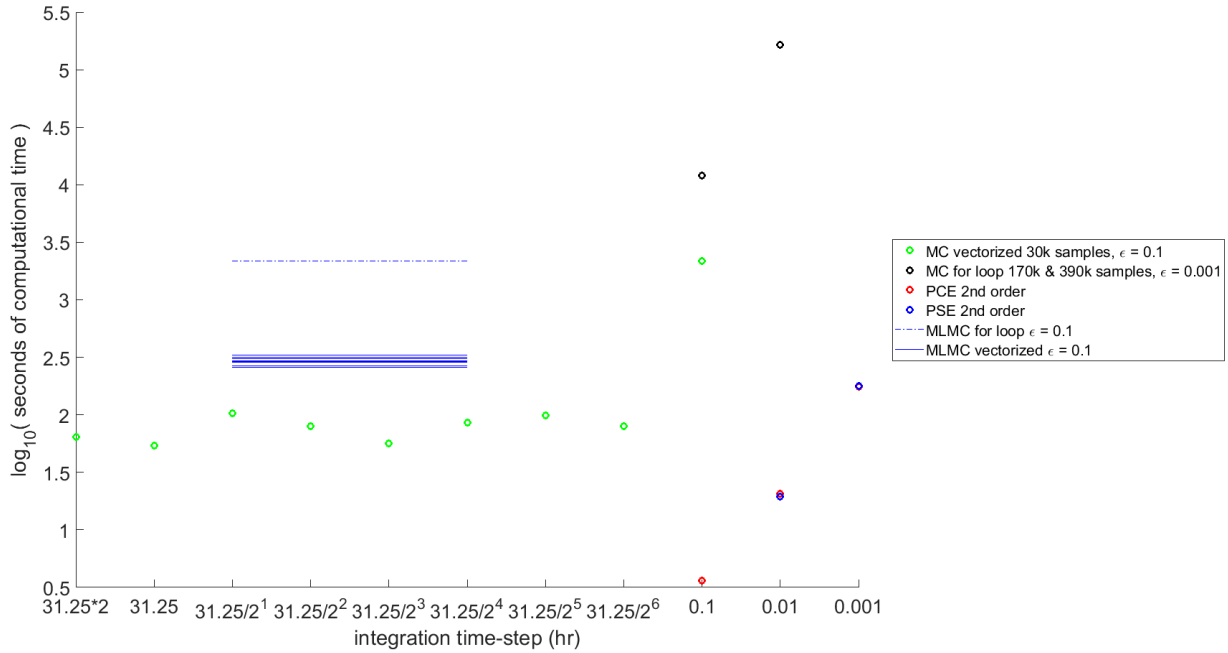

Figure 13: Comparison of the computational requirements of MC and MLMC sampling for the wastewater treatment plant.

In order to minimize the computational expenses for both MC and MLMC techniques, vectorized (element-wise) operations were implemented wherever possible for the wastewater treatment plant model. However, the capacity of RAM memory became a limiting factor for vectorized operations when very small time-steps were used for MC sampling. This limitation is evident from the vectorized MC data of Figure 13 at the time-step of 0.1 hr, where the vectorized MC operations required over 1.5 orders of magnitude more CPU time than what was needed at larger time-steps. Calculations using vectorized MC operations became computationally intractable for the time-step of 0.01 hr while using the computer used in this study; hence the absence of the result from Figure 13. Consequently, simulations with conventional "for" loops were used for the time-steps of 0.1 hr and 0.01 hr, resulting in significantly slower calculations (represented by the black circles in Figure 13) despite running the "for" loop operations in parallel on the quad-core CPU computer used in this study. MLMC code was also implemented with "for" loops to ensure that the way the algorithm was simulated did not affect the accuracy of results; the data was presented in Figures 12 and 13 using dashed blue lines.

By inspecting Figure 12, one can observe that the results of MLMC were within ±0.1 mg/L of the Benchmark values. Moreover, the difference between the maximum and minimum MLMC results was approximately 0.20 mg/L, which divided by 2 was 0.10 mg/L, in harmony with the ϵ setting of 0.1 mg/L. The computational requirements of vectorized MLMC were up to 2.8 orders of magnitude less than MC sampling that yielded the Benchmark value after 390,000 samples (see Figure 13). The vectorized MC sampling required 0.5 to 0.75 orders of magnitude less CPU time than vectorized MLMC, except for the case where integration time-step of 0.1 hr was used for vectorized MC operations, where the CPU time for MC was at least 0.7 orders of magnitude greater than MLMC. However, Figure 12 showed that the results of this MC sampling were not guaranteed

to land within the ±0.1 mg/L band (black solid lines in Figure 12) from the Benchmark values. In fact, even the mean value of different MC sampling outcomes, represented by the green dashed line in Figure 12, was outside the ±0.1 mg/L band. On the other hand, the MLMC results were within the band since ϵ was set to 0.1 mg/L. The large variability of MC results is due to the fact that MC sampling convergence is based on monitoring the change in estimates when additional samples were added to the pool of sampled data. Therefore, MC sampling is much more likely to converge to a local critical point (minimum/maximum) than to the true value. Although MLMC is not guaranteed to converge,[114] its results were observed to be more precise and more accurate than standard MC. The final estimates of MLMC are informed by the progressively more accurate calculations done at the increasingly finer discretization levels. The incorporation of accurate calculations into the final outcome, combined with checking for robust convergence,[114] improves the precision of MLMC results and increases the chances of those results being more accurate than standard MC.

In order to make a thorough comparison of MLMC to other common uncertainty propagation techniques, 2nd order Power Series (PSE) and Polynomial Chaos (PCE) expansions were constructed, parametric uncertainty was propagated, and the resulting mean values and the necessary computational time were plotted in Figures 12 and 13, respectively, as red (PCE) and blue (PSE) circles at the time-steps of 0.1 hr, 0.01 hr and 0.001 hr. The PSE formulation relied on 33 realizations of the uncertain parameters and used centered finite differences for the sensitivities. The PCE was constructed with Lagrange and Hermite polynomials (for the uniform and lognormal distributions of the uncertain parameters, respectively) and the PCE coefficients were calculated using the Least Squares method with 33 realisations of the parameters. Further details on how to construct PSE and PCE expressions can be found elsewhere.[54] While PSE and PCE required 0.2

to 1.8 orders of magnitude less CPU time than MLMC (see Figure 13), their accuracy was substantially inferior to the results of both MLMC and MC sampling, as indicated in Figure 12. The 2nd order PSE produced values that underestimated the Benchmark by 20.5%, while the 2nd order PCE results were inconsistent, from 2.4% lower to 2.1% higher than the Benchmark. Only one PCE approximation was within the ±0.1 mg/L band (see Figure 12, integration time-step of 0.001 hr). Thus, MLMC produced the most accurate results at a computational cost that was 0.7 to 2.8 orders of magnitude lower than what was required for MC to achieve a similar accuracy.

4.4 Distillation Column

To examine the performance of MLMC in highly computationally intensive chemical engineering systems, this uncertainty propagation technique was applied to a ternary distillation column.[204,205] This separation system consists of 28 stages, with stage 1 at the top of the column. Stage 1 is comprised of a condenser/reflux drum, and stage 28 contains a partial reboiler and a cooler. The feed stream, which is a mixture of toluene, n-hexane and heptane, is provided at the 22nd stage for the purpose of separating toluene from the other two chemicals. The bottom outlet stream is required to contain at least 90% toluene by mole fraction, while the distillate (top) stream should predominantly consist of n-hexane and heptane. The model assumes that the mole fractions of toluene in the top stream and heptane in the bottom stream can be measured online and contains PI controllers that regulate variability in these variables. The toluene mole fraction in the top stream is controlled by changing the reflux ratio of the condenser, while the heptane mole fraction in the bottom stream is maintained by adjusting the partial reboiler heat duty. Table 9 contains the values for various model parameters. The model was simulated on a time domain $t \in [0\text{ hr}, 25\text{ hr}]$.

Table 9: Parameters of the ternary distillation column model.

Parameter	Value	Description
N_{stages}	28	Number of stages in the column (the top stage is #1)
N_{feed}	22	Feed stage number
D_{col}	0.722 m	Column diameter
A_{cond}	0.1795 m^2	Area of the condenser
A_{reb}	0.1014 m^2	Area of the reboiler
A_{cool}	0.0250 m^2	Area of the cooler
F_{22}	600 mol/hr	Feed flowrate
T	60 ^{0}C	Temperature of inlet and outlet streams
P_{cond}	0.9334 atm	Pressure in the condenser
sp_1	5.83×10^{-3}	Set-point for the PI controller of the of toluene mole fraction in the top stream
K_{c1}	-16.67	Gain of the toluene PI controller
τ_{I1}	0.505 min	Integral time constant of the toluene controller
sp_2	8.00×10^{-2}	Set-point for the PI controller of heptane mole fraction in the bottom stream
K_{c2}	43.33	Gain of the heptane PI controller
τ_{I2}	0.175 min	Integral time constant of the heptane controller

The aim of this case study was to use MC and MLMC sampling to characterize product variability due to uncertainty in the feed composition. The mole fractions of n-hexane and heptane in the inlet stream were assumed to be uncertain and uniformly distributed on the intervals [0.06, 0.10] and [0.10, 0.14], respectively. Time trajectories of the observables at various uncertain parameter realisations were plotted in Figure 14. For this case study, the MLMC technique was adjusted to simultaneously estimate the expected values of the time-averaged mole fractions of toluene in the bottom stream ($\mathbb{E}[t\ell_b^t]$), n-hexane in the distillate stream ($\mathbb{E}[\hbar x_d^t]$), n-hexane in the bottom stream ($\mathbb{E}[\hbar x_b^t]$) and heptane in the distillate stream ($\mathbb{E}[\hbar p_d^t]$). This is formulated as follows:

$$\mathbb{E}[t\ell_b^t] = \mathbb{E}\left[\frac{\int_0^{t_f} t\ell_b(t)dt}{\int_0^{t_f} dt}\right] \tag{56}$$

$$\mathbb{E}[\hbar x_d^t] = \mathbb{E}\left[\frac{\int_0^{t_f} \hbar x_d(t)dt}{\int_0^{t_f} dt}\right] \tag{57}$$

$$\mathbb{E}[\hbar x_b^t] = \mathbb{E}\left[\frac{\int_0^{t_f} \hbar x_b(t)dt}{\int_0^{t_f} dt}\right] \tag{58}$$

$$\mathbb{E}[\hbar p_d^t] = \mathbb{E}\left[\frac{\int_0^{t_f} \hbar p_d(t)dt}{\int_0^{t_f} dt}\right] \tag{59}$$

where t_f was 25 hr. Note that these quantities are not directly under closed-loop control, but they are important for assessing the product quality of the distillation column.

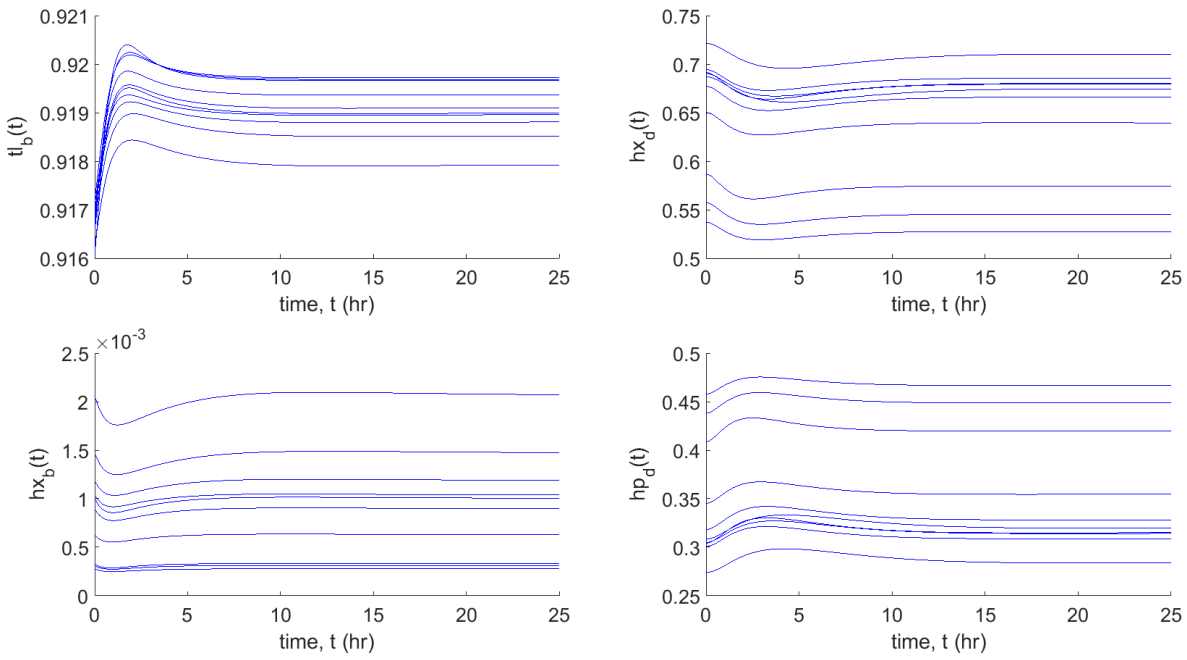

Figure 14: Time trajectories of the mole fractions of the distillation column's observables at various realisations of uncertain parameters.

The ϵ value for MLMC was set to 0.01 (the same for all observables). For preliminary sampling, the total number of levels, *L*, was set to 3, and the number of uncertain parameter realisations to

use at each level, Ns_l, was set to 8. The coarsest MLMC time-step (h_0) was set to 0.05 hr because it was the largest time-step that did not produce numerical instabilities during the integration of model equations. Compared to the mixing tank and the wastewater treatment plant, the number of levels L used in preliminary sampling of the distillation column model and the number of samples Ns_l at each level were much smaller. This reduction is due to a significantly higher computational complexity of the distillation column model: the time to conduct one simulation is two orders of magnitude greater than for the mixing tank and wastewater treatment plant models, and can range from 120 to 620 seconds, depending on the realisation of uncertain parameters.

The MLMC procedure for the simultaneous estimation of multiple observables was carried out as described in the Multilevel Monte Carlo section of this study. Four sets of optimal numbers of samples were generated, one for each observable. For each level, the maximum among the sets was used to conduct MLMC calculations. The MLMC sampling converged upon reaching $L = 3$, and the entire procedure required an average of 11,673 seconds (in total, there were 5 implementations of MLMC). The computational budget for MC sampling was set to 13,500 seconds, 360 seconds higher than the maximum time required by MLMC (13,140 seconds), to observe the quality of results MC sampling was capable of producing with a slightly higher timeframe than MLMC. Like MLMC, MC was also implemented 5 times. Note that the integration of model equations during MC sampling was performed using the time-step h_3 (the finest discretization) of MLMC to allow MC sampling to produce reasonably accurate results.

Table 10 summarizes the results from the present case study. To compare the performance of MC and MLMC sampling within the same computational budget, the outcomes of the sampling techniques have been plotted against CPU time in Figure 6. Qualitatively, it can be seen in Figure 15 that the variability of final MC results exceeded the variability of MLMC approximations for

all observables since the differences between the largest and smallest MC approximations at the final CPU times were greater than the differences between the largest and smallest results of MLMC. Table 10 shows that the variability of MC sampling outcomes exceeded the variability of MLMC results by 3.71 times for $\mathbb{E}[t\ell_b^t]$, 1.55 times for $\mathbb{E}[\hbar x_d^t]$, 3.55 times for $\mathbb{E}[\hbar x_b^t]$, and 1.58 times for $\mathbb{E}[\hbar p_d^t]$. While additional MC sampling would be needed to establish the benchmark values for the expected time averages and their variability, for identical CPU time constraints MLMC demonstrated higher precision than MC. In order for the standard MC sampling to obtain results of similar precision to MLMC, it would likely require at least one order of magnitude greater computational budget (i.e. in excess of 37.5 hours). Therefore, MLMC is the preferred method for computationally demanding problems.

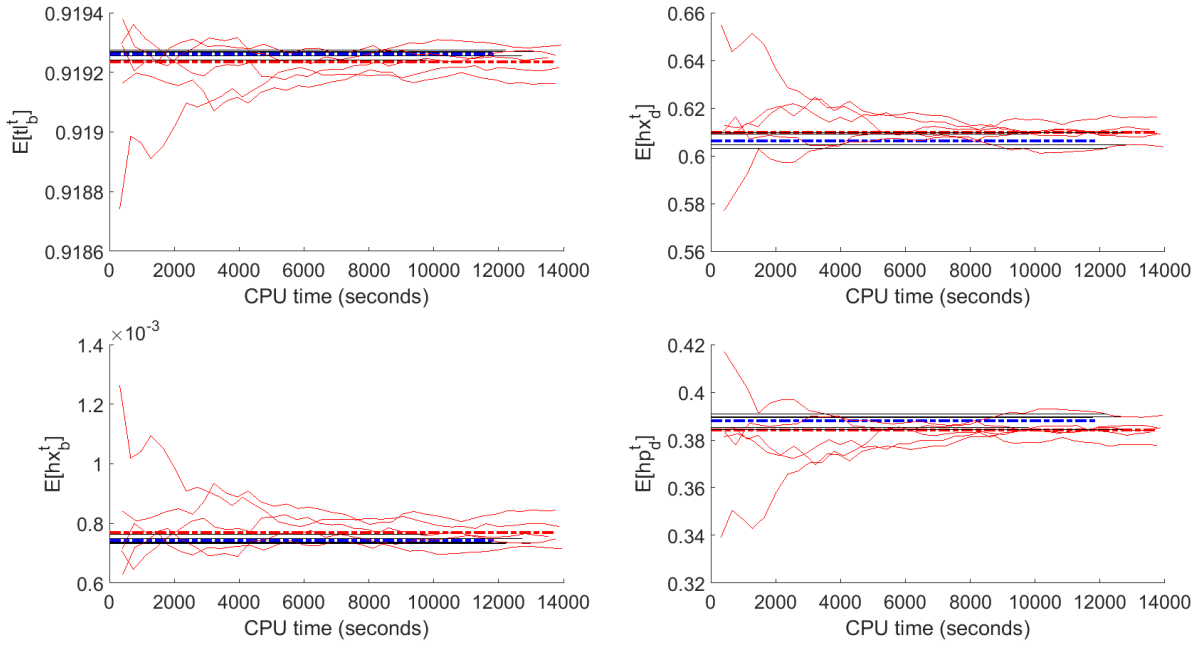

Figure 15: A comparison of MC and MLMC estimates of the observables of the distillation column model. Red solid lines represent MC results, black solid lines represent MLMC results and span the time required for a particular MLMC run. Red dashed lines represent the mean MC results, and the blue dashed lines represent the mean MLMC results. MC sampling used the smallest time-step employed by MLMC (i.e. h_3).

Table 10: A comparison of MC and MLMC sampling results. Note that μ_{MC} stands for the mean of MC sampling results, σ_{MC} stands for the standard deviation of MC sampling results, μ_{MLMC} represents the mean of MLMC sampling runs, and σ_{MLMC} is the standard deviation of MLMC sampling runs.

Observable	μ_{MC}	$\frac{\sigma_{MC}}{\mu_{MC}}$	μ_{MLMC}	$\frac{\sigma_{MLMC}}{\mu_{MLMC}}$	$\frac{\sigma_{MC}}{\mu_{MC}} / \frac{\sigma_{MLMC}}{\mu_{MLMC}}$
$\mathbb{E}[t\ell_b^t]$	0.9192	0.0053%	0.9193	0.0014%	3.71
$\mathbb{E}[\hbar x_d^t]$	0.6099	0.74%	0.6063	0.48%	1.55
$\mathbb{E}[\hbar x_b^t]$	7.707×10^{-4}	6.35%	7.436×10^{-4}	1.79%	3.55
$\mathbb{E}[\hbar p_d^t]$	0.3843	1.18%	0.3879	0.75%	1.58

4.5 Potential Shortcomings of MLMC

The results presented in this work show that MLMC is an attractive alternative to the standard MC sampling. However, it should be emphasized that MLMC is not free of potential shortcomings. The convergence in the estimates is not guaranteed since there may be a limit on the minimum achievable ϵ. Oversampling is possible in systems with a high variability of the estimated quantities. As already noted in the Illustrative Case Study section, only the optimal domain discretization at l = 0 provides the maximum time savings offered by MLMC over standard MC. Several preliminary sampling runs may be required to find the optimal domain discretization at l = 0. Furthermore, for the preliminary sampling, the choice of the initial number of levels and samples per level has to be guided by the user's experience with the system of interest, rather than a universal rule, making the automation of preliminary sampling problem-specific. The total number of levels that will be necessary for the MLMC sampling procedure is difficult to determine beforehand; the relationship between the discretization level and the variance of the estimate needs to be known or observed through preliminary sampling. As discussed in the Multilevel Monte Carlo section, the constants α_{MLMC}, β_{MLMC}, and γ_{MLMC} have to be known, assumed a priori, or estimated through preliminary sampling. If the conditions of the MLMC Theorem cannot be met

(e.g. if the error in the estimates increases as the domain discretization becomes finer), MLMC is not applicable to the problem and standard MC sampling should be used for estimating the observables of interest.

4.6 Chapter Summary

In this work, MLMC has been applied to three chemical engineering systems and its accuracy and computational costs have been compared to MC sampling, and, for the first time, to PSE and PCE. A mixing storage tank was used in this study to illustrate the step-by-step implementation of the MLMC algorithm. For the case of the wastewater treatment plant, the most notable feature of MLMC results was that they were within the specified error bound from the Benchmark value, not just from each other. Thus, the MLMC accuracy significantly exceeded that of MC, PSE and PCE at a lower computational cost than MC. For the case of the ternary distillation column with a fixed computational budget, MLMC estimated four observables simultaneously and produced results with a lower variability than MC sampling for all observables. The MLMC technique is a very promising and computationally attractive option when a certain degree of accuracy in the results needs to be ensured, since it is more likely to converge to the true value than standard MC sampling.

Chapter 5: Multilevel Monte Carlo Applied to Stochastic Multiscale Systems

Building upon the results of Chapter 4 and the favourable performance of Multilevel Monte Carlo Sampling (MLMC) for traditional chemical engineering systems, this chapter presents the applications of MLMC to stochastic multiscale systems in chemical engineering. First, MLMC is used to estimate and reduce the noise in the observables of the chemical vapour deposition (CVD) process discussed in Chapter 3. Next, MLMC is used to improve uncertainty quantification in the CVD model. Following that, the same methodology is applied to improve uncertainty quantification in a catalytic pore reactor model.

The purpose of the work presented in this chapter is to adapt the MLMC sampling technique (described in sections 2.2.3 and 4.2) for the estimation and reduction of random noise in the observables of stochastic multiscale systems. The systems used as case studies of industrially-relevant stochastic multiscale chemical engineering processes were the simulation of thin film formation by chemical vapour deposition discussed in section 3.2 and a stochastic multiscale model of a catalytic pore reactor (discussed in section 5.4). First, the noise in the expected value of the CVD system's observable was estimated using MLMC. The performance of this technique was compared to the standard Monte Carlo (MC) sampling. Next, the MLMC method was used to increase the precision of the estimated expected values of the observables of the two stochastic multiscale systems (the CVD and the catalytic pore reactor models) and thereby improve the quality of the traditional low-order uncertainty quantification expressions (PCE has been used in this chapter). The performance of MLMC for improving uncertainty quantification was compared to the heuristic method of constructing PCE and to standard MC sampling.

The work presented in this chapter highlights the nuances of adapting the MLMC technique to stochastic multiscale systems and provides insight on the benefits and challenges of using MLMC with stochastic multiscale systems. Based on a review of the current published literature (presented in section 2.2.3), the present work is the first application of MLMC for noise estimation and uncertainty quantification in multiscale systems that incorporate non-closed-form equations.

5.1 MLMC sampling applied to stochastic multiscale models

When applying the MLMC sampling technique to a system, it is necessary to identify the key discretization domains that affect the variability of the observable values generated by the system, as well as the computational cost associated with each level of accuracy. The majority of current MLMC implementations discretize the time domains of their respective simulation systems because the systems are represented by time-dependent differential equations and smaller integration timesteps yield results of lower variability. In the case of stochastic multiscale systems, domains other than temporal (e.g. spatial) may have a greater influence on the accuracy of simulations' outcomes. For example, in systems that combine discrete particle simulations (e.g. kMC) with continuous differential equations, the number of particles considered in the discrete simulation domain may have a stronger impact on the noise in the observables than the integration timestep used to integrate the continuous differential equations.

Once the discretization domain is chosen, the sequence for discretization level refinement needs to be selected. The levels should be chosen such that the variability of the observables is reduced in an optimal way. On the one hand, a slow refinement of the discretization would reduce the variance, but the overall computational cost would be unnecessarily high since too many levels would be sampled by the MLMC scheme. On the other hand, a drastic refinement of discretization could quickly make higher discretization levels (and the total cost) computationally intractable;

for example, decreasing the integration timestep of a large system of differential equations by an order of magnitude with every additional level could make the costs of sampling prohibitive after a few levels have been added by the MLMC scheme. For efficient MLMC sampling, a geometric sequence is commonly used for discretization level refinement.[114] The optimal value of the refinement factor is system-specific:[113,114]

$$\mathcal{M} = \frac{\Delta t_{l+1}}{\Delta t_l} \tag{60}$$

where $\mathcal{M}$ is the optimal refinement factor (e.g. ½, ¼), and Δt_l and Δt_{l+1} represent the integration timesteps used by MLMC while sampling a time-dependent system at the successive discretization levels l and $l + 1$, respectively.

5.1.1 Applying MLMC to the Chemical Vapour Deposition Model

In the stochastic multiscale kMC-PDE model described in section 3.2, the size of the lattice of the kMC thin film model (represented by N in section 3.2.2) directly affects the variance of the roughness observable at the end of the batch, as demonstrated in Figure 16. As discussed in section 3.3, using larger lattices (e.g. $N = 100$ or 150) substantially decreases the noise in the observable (i.e. surface roughness) at the expense of longer simulation times. Therefore, the relationships between lattice size, observable's variability and computational cost indicate that MLMC sampling is indeed applicable to this stochastic multiscale system.

As discussed in section 3.3, the current heuristic approach[2] to approximating roughness is to use six $N = 30$ lattices, find the roughness time trajectories from each lattice, find the average trajectory, and use it as an adequate representation of results that would be obtained from $N = 100$ and $N = 150$ simulations, as illustrated in Figure 2 (see section 3.3). Substantial CPU time savings can be achieved when the simulations are run in parallel. MLMC sampling can also take advantage

of parallel computing, but one can approach the selection of the number and size of lattices in a more objective manner because MLMC tracks the decay in the variability of the estimates of the observable with each discretization level and adjusts the number of samples accordingly, as per equations (4) and (10).

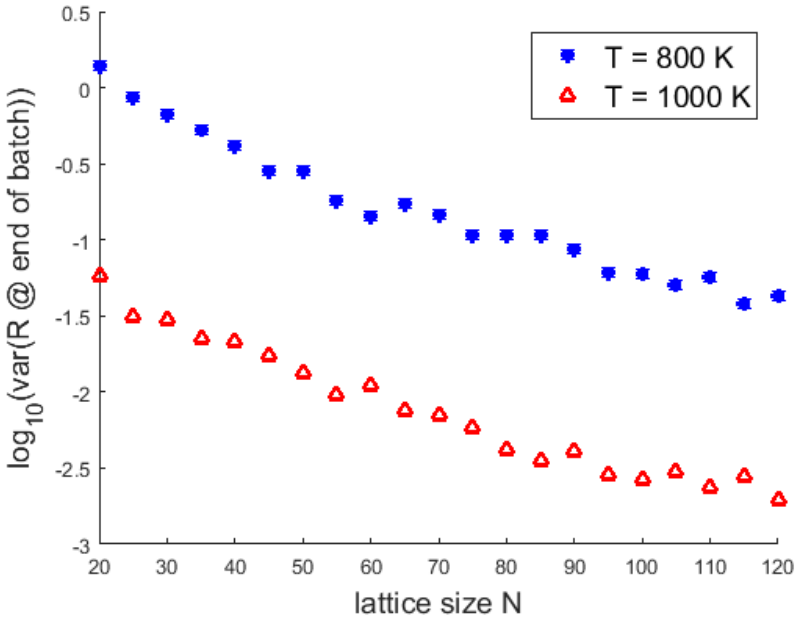

Figure 16: The decay of variance in roughness at the end of the batch due to increasing thin film lattice size. For brevity, results at only two temperature values have been shown.

Since larger thin film lattice sizes impact both the observable's estimates and the computational cost, the N values can be considered as discretization levels in the spatial domain. To adapt the MLMC sampling algorithm to the present multiscale model, an empirical relationship was established using least squares regression between the lattice size N, temperature T, and the kMC timestep Δt_{kMC}, since they are already related through the kMC rate of adsorption W_a. This rate is calculated based on $\sqrt{T}$ and N^2 in equation (23) and is inversely related to Δt_{kMC} through equation (26). In this work, only N and T were manipulated. Thus, an empirical relationship was established using N and T as the independent variables. Note that since ς of equation (26) is a uniform random

number, variability is present in the kMC timestep. Hence, N and T are related to the mean value of the kMC timestep ($\widehat{\Delta t}_{kMC}$) as follows:

$$\widehat{\Delta t}_{kMC} = (\alpha T + \beta)N^{-2} \tag{61}$$

where the α and β model coefficients were found by least squares regression. This model was validated by examining its quality of fit to the mean kMC timestep values collected from the multiscale model at various N and T values (discussed in detail in section 5.1.2 below). For the purposes of this study, equation (26) of the kMC-PDE model was replaced with equation (61). Note that if another stochastic multiscale model were considered for MLMC sampling or if the kMC-PDE model is examined at a temperature outside the valid range of temperatures where equation (61) is valid, an empirical relationship different from equation (61) will need to be established.

Next, it was necessary to establish the rule for selecting the kMC lattice sizes to enable MLMC sampling to systematically define each discretization level. Substituting equation (61) into equation (60) yields:

$$\mathcal{M} = \frac{(\alpha T + \beta)N_{l+1}^{-2}}{(\alpha T + \beta)N_l^{-2}} \tag{62}$$

which, upon simplification, results in:

$$\frac{N_{l+1}}{N_l} = \sqrt{\mathcal{M}^{-1}} \tag{63}$$

Equation (63) can be further rearranged to yield the rule for selecting the lattice sizes N_l during MLMC sampling of the modified kMC-PDE model:

$$N_l = \left\lceil N_0 \times \left(\sqrt{\mathcal{M}^{-1}}\right)^l \right\rceil \tag{64}$$

where N_0 is the user-defined minimum lattice size used by MLMC sampling. Equation (64) ensures a systematic selection of discretization levels of the spatial domain of the kMC simulation: the lattice size at an MLMC discretization level $l > 0$ will correspond to the $\widehat{\Delta t}_{kMC}$ value that is an $\mathcal{M}$ factor refinement of the mean kMC timestep at the preceding discretization level, in harmony with the existing MLMC implementations.[113–115,206]

5.1.2 Establishing the α and β constants

To estimate the α and β coefficients of equation (61), Δt_{kMC} values of equation (26) were collected from various simulations of the original, unmodified kMC-PDE model. In total, 105 simulations were conducted, where the lattice sizes ranged from 20 to 120, inclusive, with increments of 5, at temperatures from $800\ K$ to $1200\ K$, inclusive, with increments of $100\ K$. The temperature range was selected based on the expected operating conditions for an actual thin film deposition system. The mean kMC timestep values, $\widehat{\Delta t}_{kMC}$, were calculated for each of the 105 simulations and the results have been plotted in Figure 17 (red markers). It can be seen that all 105 $\widehat{\Delta t}_{kMC}$ values were distinct from each other. Their uniqueness was further verified from their statistical distributions. Since each simulation was run from $0\ s$ to $100\ s$ with the coupling interval $\Delta\mathcal{T} = 0.1\ s$, the kMC and PDE scales were coupled 1000 times in every simulation. The mean kMC timestep values were calculated for every coupling interval of every simulation, and their histograms were plotted (one histogram per simulation). The statistical distributions from all simulations were normal or close to normal, with sharp peaks that coincided with the data (red dots) presented in Figure 17 and very short tails that did not overlap with distribution tails from other simulations (the distributions were not shown for brevity). Thus, it was concluded that the variability of the mean kMC timesteps was sufficiently low to be neglected.

Next, the α and β coefficient values of equation (61) were obtained from least squares regression ($-7.594 \times 10^{-4}\ s \cdot K^{-1}$ and $1.106\ s$, respectively). The resulting surface plot, where $\widehat{\Delta t}_{kMC}$ was calculated according to equation (61), has also been presented in Figure 17. A very close agreement between the two datasets was demonstrated: the average Δt_{kMC} of the original model (red markers in Figure 17) and $\widehat{\Delta t}_{kMC}$ from equation (61) (surface plot in Figure 17) have negligible residual errors (between -4.4% and 1.7%, not shown for brevity). Therefore, the dataset considered for N and T was sufficient to obtain an accurate empirical relationship.

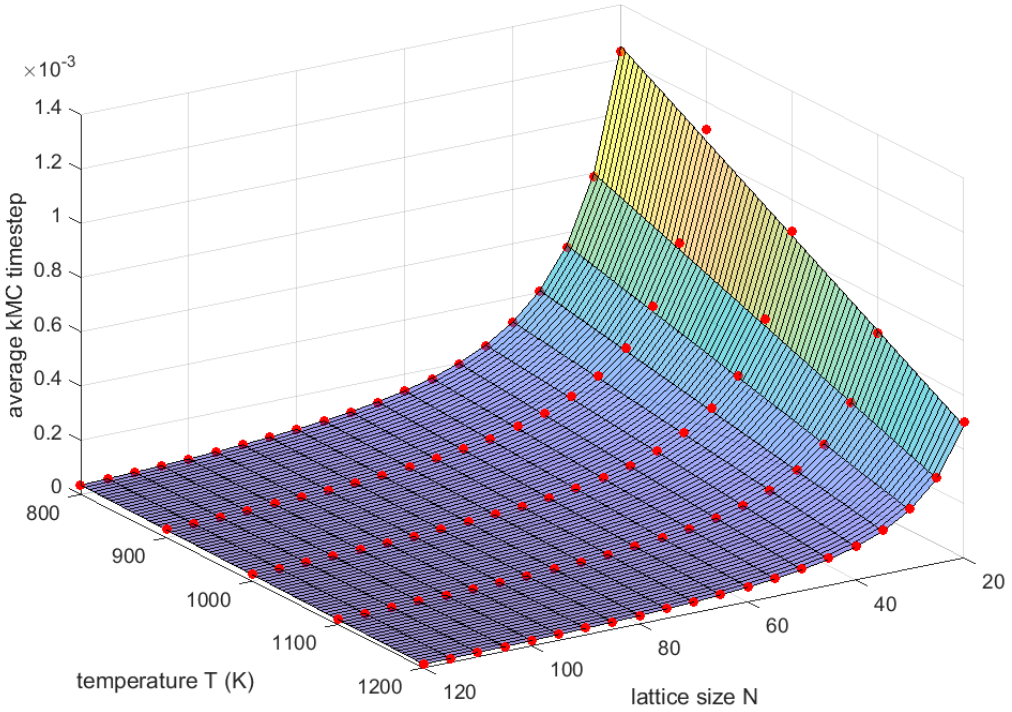

Figure 17: A comparison between the mean kMC timesteps obtained from simulations (red markers) and the kMC timesteps calculated from equation (61) (surface plot). Note that α was found to be $-7.594 \times 10^{-4}\ s \cdot K^{-1}$ and β was 1.106 s.

Next, the method of calculating the kMC timestep in the multiscale model was modified: equation (26) was replaced with equation (61). To verify that the modified kMC model can produce sufficiently accurate results, multiscale simulations were conducted with lattice sizes $N = 50$ and $N = 100$ at the temperatures of $940\ K$ and $1140\ K$. These temperature values were chosen

because they were not used to identify the coefficients of equation (61) and because they represent the two regimes of the multiscale model, the lower-temperature adsorption-dominated regime that produces thin films with higher roughness ($940\ K$) and the higher-temperature migration-dominated regime that results in smoother thin films ($1140\ K$). For each temperature, two kinds of simulations were carried out: with the original multiscale model where Δt_{kMC} was calculated by equation (26) after each kMC event, and with the modified multiscale model where the fixed $\widehat{\Delta t_{kMC}}$ was calculated from equation (61) once at the beginning of the simulation and used throughout without alterations. The comparison is presented in Figure 18, with 8 roughness time trajectories from the original model and 1 representative roughness trajectory (no averaging) from the modified model in each subplot. Clearly, the replacement of equation (26) by equation (61) preserved the accuracy of predictions. The predictions from the modified model approached the average of the time trajectories obtained from the original model, especially for larger lattice sizes ($N = 100$), as shown in Figure 18. Note that the use of a fixed kMC timestep did not affect the computational costs. Additional testing has been performed to corroborate this observation (omitted for brevity). A review of the current literature indicates that such a result has not been reported for this stochastic multiscale system. Also, note that fixing the kMC timestep did not eliminate the noise from the roughness time trajectories, since the kMC portion of the multiscale model continued to use uniform random numbers to choose among the adsorption, desorption and migration surface events, as discussed in section 3.2.2.

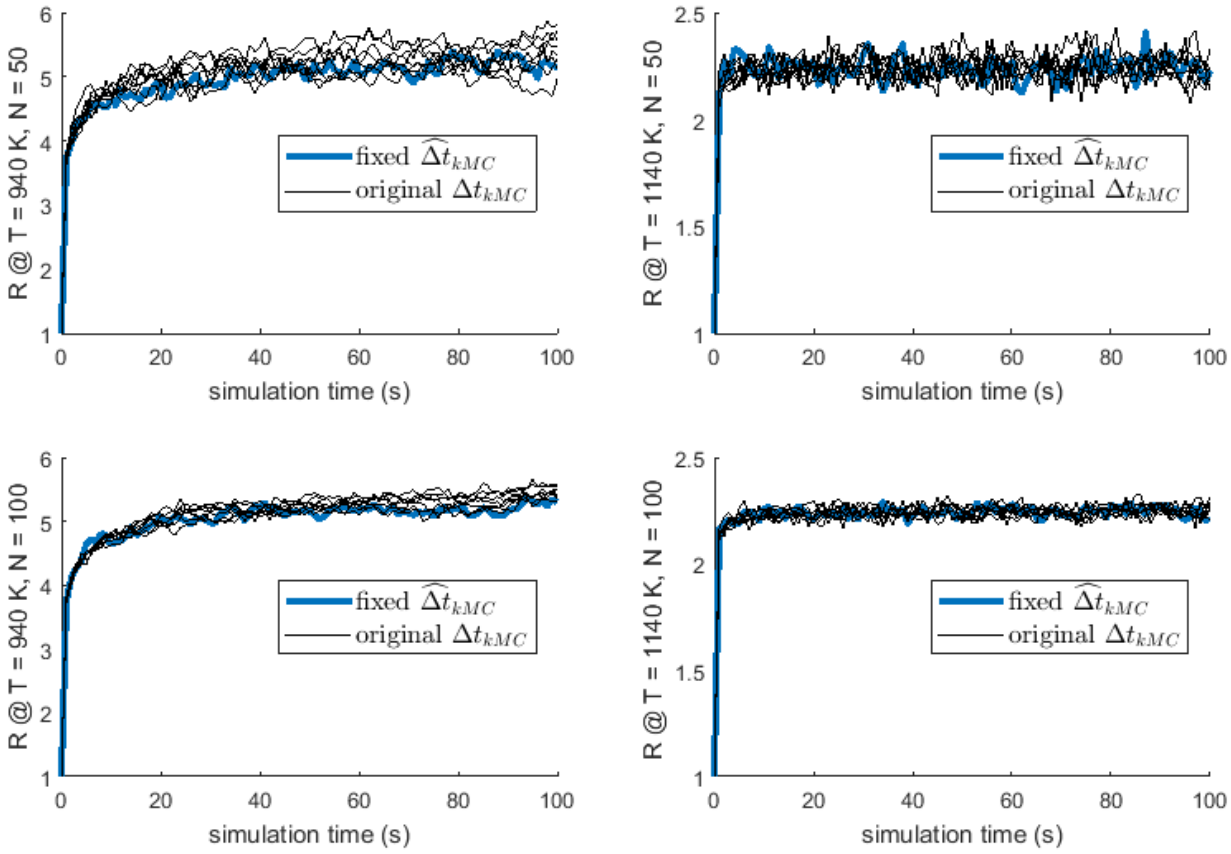

Figure 18: A comparison between the time trajectories of roughness produced by the original multiscale simulation and the modified model with a fixed kMC timestep.

5.1.3 Estimating the noise via MC and MLMC sampling

To compare the computational cost and accuracy of estimating the noise in the roughness observable of the multiscale model by the two sampling techniques, Monte Carlo and Multilevel Monte Carlo, multiscale simulations were conducted with the original model and the modified model (with equation (61) substituted for equation (26)) at three temperature settings: $940\ K$, $1040\ K$ and $1140\ K$. The focus of this work was to estimate the maximum noise in the average of the last five roughness values (i.e. at seconds 96-100, inclusive, of each simulation). The average is denoted from henceforth by R_f.

While MC sampling relied on the original model, the MLMC scheme used the modified version, i.e. the model that uses a fixed kMC timestep. MC sampling was performed using $N = 100$. Larger lattices were not sampled because the substantial increase in computational cost was not justified by the observed changes in the roughness trajectory. For example, a negligible difference can be observed between $N = 100$ and $N = 150$ in Figure 2, yet the computational cost is more than 3 times higher for $N = 150$ (see section 3.3). Throughout MC sampling, the maximum and minimum observed R_f values, as well as the difference between them, were updated whenever a new batch of 80 multiscale simulations finished running. In total, 6,000 simulations were performed at each temperature.

MLMC sampling used equation (64) as the rule for setting the spatial domain discretization at each MLMC level. The minimum lattice size used in equation (64), i.e. N_0, was set to 20, since that was the smallest N value that did not cause extreme noise in the predictions made by the present kMC-PDE model. The optimal refinement factor ($\mathcal{M}$) was set to ½ because it was the smallest value that did not cause MLMC to sample computationally intractable lattice sizes at higher discretization levels.

MLMC sampling was performed for each of the three temperatures; the root mean square error tolerance ϵ was set to 0.3 and 0.2 at each temperature. To ensure adequate sampling, 64 MLMC runs were performed at each combination of the temperature and ϵ values. Once all MLMC sampling runs were completed, the differences between the maximum and minimum estimated R_f values were calculated at each setting. Figure 19 and Table 11 compare the R_f noise bounds obtained by MC and MLMC sampling.

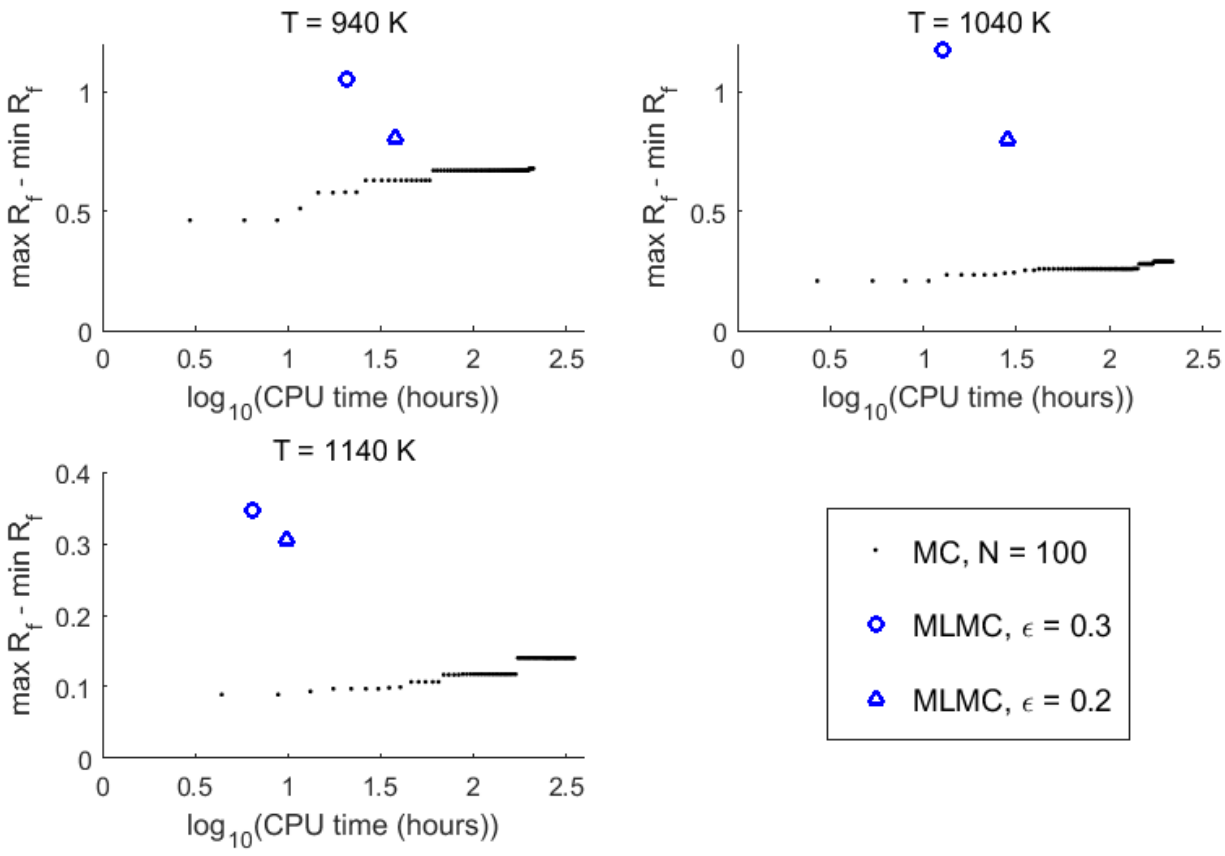

Figure 19: A comparison of the estimates of noise in R_f from MC and MLMC sampling.

Table 11: Summary of MC and MLMC results.

T (K)	Sampling method	CPU time (hours)	$max\ R_f - min\ R_f$
940	MC, $N = 100$	208.3	0.679
	MLMC, $\epsilon = 0.3$	21.0	1.054
	MLMC, $\epsilon = 0.2$	38.2	0.809
1040	MC, $N = 100$	217.6	0.289
	MLMC, $\epsilon = 0.3$	12.7	1.175
	MLMC, $\epsilon = 0.2$	28.5	0.803
1140	MC, $N = 100$	349.8	0.140
	MLMC, $\epsilon = 0.3$	6.5	0.347
	MLMC, $\epsilon = 0.2$	9.8	0.307

Based on the results summarized in Figure 19, it was observed that MC sampling exhibited creeping noise in roughness values due to the asymptotic behaviour of $\max R_f - \min R_f$ values. MC sampling achieved noise bounds of 0.679 for $T = 940\ K$, 0.289 for $1040\ K$ and 0.140 for $1140\ K$. It should be noted that these noise estimates could increase with additional MC sampling. In this study, the computational cost of MC became prohibitive after such a long sampling time (i.e. over 200 hours).

From Figure 19 and Table 11, it was apparent that MLMC could produce conservative estimates of noise in R_f values and offered an order of magnitude time savings compared to MC sampling: the computational requirements of MLMC were between 6.5 hours (at $1140\ K$, $\epsilon = 0.3$) and 38.2 hours (at $940\ K$, $\epsilon = 0.2$), whereas the MC sampling time ranged from 208.3 hours (at $T = 940\ K$) to 349.8 hours (at $T = 1140\ K$), as per Table 11. It could also be observed that the MLMC and MC results at $940\ K$ and $1140\ K$ showed consistent trends: all MLMC and MC noise estimates at $1140\ K$ were smaller than at $940\ K$ (e.g. for $\epsilon = 0.2$, MLMC results were 0.809 at $940\ K$ and 0.307 at $1140\ K$, while MC results were 0.679 at $940\ K$ and 0.140 at $1140\ K$). However, at $1040\ K$, the MC estimate (i.e. 0.289) was between the MC noise values from $940\ K$ and $1140\ K$ (0.679 and 0.140, respectively), while MLMC produced a result higher than at $940\ K$ (for $\epsilon = 0.3$, at $1040\ K$ the result was 1.175, higher than 1.054 at $940\ K$). Thus, at 1040 K, MLMC diverged from the trends in the noise bounds observed at 940 K and 1140 K. One possible explanation to this behaviour is that the multiscale system started to transition into the lower-noise migration-dominated regime and caused MLMC R_f noise estimate to become overly conservative. This issue could be mitigated to some extent by decreasing ϵ from 0.3 to 0.2 at $T = 1040\ K$, but for $\epsilon = 0.2$ the MLMC result at $1040\ K$ was still very close to the value at $940\ K$ (0.803 versus 0.809, respectively). The obvious shortcoming of this approach to tuning the ϵ value is that to

obtain a more accurate noise estimate, the selection of ϵ has to be guided by some *a priori* knowledge of the system behaviour. However, as a general guideline, it is recommended to initially set ϵ to 5%-10% of the magnitude of the expected value and subsequently tune the ϵ tolerance as necessary. Thus, despite the computational time savings offered by MLMC sampling, MC sampling may still need to be conducted for a short amount of time to obtain an approximation of the noise that can validate the estimates from MLMC. The greatest computational savings from MLMC were observed in the low-noise regime, i.e. at $1140\ K$, where MLMC required at most 9.8 hours of CPU time (for $\epsilon = 0.2$), whereas MC sampling took 349.8 hours. Note that the reported MLMC CPU time does not include the time needed to identify the MLMC Theorem constants via preliminary sampling (discussed in Chapter 3). The computational cost of preliminary sampling may reach the CPU time needed for one MLMC run (e.g. approximately 4 hours at $T = 1140\ K$), depending on the number of levels and samples per level (Ns_l) that the user chooses to employ in the estimation of the constants α_{MLMC}, β_{MLMC} and γ_{MLMC}. Also, since the CPU times required by MLMC were on the order of hours, online applications of MLMC for stochastic systems would still be challenging. However, the usage of high-performance computing clusters could enable online applications by leveraging the parallelization of MLMC sampling. Furthermore, using larger ϵ values could also reduce the computational costs.

Note that smaller ϵ were considered (e.g. 0.1). At $T = 940\ K$ and $1040\ K$, setting ϵ to 0.1 caused MLMC to sample large, computationally intractable thin film lattices. Such MLMC sampling runs were terminated before completion and their results could not be obtained. At $1140\ K$, MLMC runs with $\epsilon = 0.1$ were able to achieve roughness estimates because the multiscale system was in the low-noise regime at that temperature, but the approximations did not improve upon the presented results. This behaviour of MLMC is different from what was observed previously for

closed-form systems in Chapter 4. This result suggests that there is a lower bound on the ϵ error tolerance that can be achieved by MLMC for the case of stochastic systems.

Furthermore, it is imperative for MLMC to use the same Brownian path[113] when the samples of the observable are obtained at two subsequent discretization levels, as discussed in section 2.2.3. When MLMC is applied to systems represented by continuous equations, the discretization level must be the only source of discrepancies between the samples. Due to the application of MLMC sampling to a stochastic multiscale system, an additional source of variability arises: the inherent noise associated with the non-closed-form expressions will cause deviations in the results of different simulation runs even when all parameter values and the discretization level are the same. To ensure that MLMC uses the same Brownian path when sampling the observables of a stochastic multiscale system for a particular uncertain parameter realization, the same seed must be provided to the random number generator in the simulations performed at two subsequent discretization levels.

Note that multiple approaches for adapting the multiscale model for MLMC sampling were explored in this study. The coupling timestep $\Delta\mathcal{T}$ between the kMC and PDE scales was considered as the discretization level in MLMC sampling. However, when $\Delta\mathcal{T}$ was small, it resulted in longer simulation times and produced the same roughness trajectories. However, when $\Delta\mathcal{T}$ was large, it caused discontinuities in the roughness time trajectory, indicating numerical instability encountered at the coupling of the two scales. Consequently, MLMC was applied to use the lattice sizes as discretization levels.

5.2 Applying MLMC sampling to the identification of PCE

As shown in equation (2), the calculation of PCE coefficients β_i by the NISP method directly depends on $y(\boldsymbol{\xi})$, which are the observable values obtained at specific realisations of the corresponding uncertain parameters. The noise inherent in stochastic multiscale systems causes the sampled observables to vary even when all process/model parameters are kept the same between batch runs. Hence, the quality of PCE approximation can be improved if greater care is exercised when estimating the values that are needed for the NISP method to calculate β_i for PCE.

It was demonstrated in section 5.1.3 that MLMC sampling can reduce the noise in the expected values of the observables of stochastic multiscale systems by tuning the ϵ parameter (i.e. the root mean square error tolerance). Therefore, MLMC has been applied to assist in the calculation of β_i PCE coefficients obtained via the NISP method. MLMC was used to decrease the variability in the corresponding observables obtained at appropriate realizations of the uncertain parameters of two stochastic multiscale systems: the CVD model and a catalytic pore reactor model (described in section 5.4). Since MLMC makes the obtained observables more precise, the quality of uncertainty quantification is improved. The framework for applying MLMC sampling to uncertainty quantification via PCE NISP is summarized below.

MLMC Preliminaries

Step 1: Select an appropriate discretization domain for MLMC to control the accuracy of the estimates of the observable of the stochastic multiscale model, i.e. P_l of equation (4).

Step 2: Select a sequence for refining the discretization levels l for the domain chosen in Step 1.

Step 3: Conduct preliminary sampling of the stochastic multiscale model at several discretization levels (e.g. l = 0, 1…4) to estimate the variability of the estimates of the observable, i.e. the V_l values in equation (10).

Step 4: Based on the knowledge from Step 3, select the magnitude of the root mean square error tolerance ϵ that will be appropriate for the magnitude of the estimated observable of the stochastic multiscale model and computationally feasible considering the amount of stochastic noise present in the system. The minimum feasible ϵ can be approximated from preliminary simulations by first setting it to 10% of the expected observable value obtained at l = 0 and then adjusting the ϵ value through trial and error until MLMC cannot achieve the imposed error tolerance.

PCE NISP Settings

Step 5: Select the order of the PCE expression and the type of orthogonal polynomial basis based on the *a priori* knowledge about the distributions of the uncertain parameters $\boldsymbol{u}$.

Step 6: Set the number of polynomials' roots that will be used to calculate β_i in equation (2).

Step 7: Generate the combinations of uncertain parameters' realizations that correspond to the roots in Step 6.[54,100]

MLMC applied to identifying the PCE

Step 8: Use MLMC sampling to estimate the observable values of the stochastic multiscale model (y in equations (2), or, equivalently, $\mathbb{E}[P]$ in equation (4)) at each combination of the uncertain parameter realizations generated in Step 7. MLMC will continue to sample the system at each combination of uncertain parameters and add discretization levels until the variability of the observable's estimates at the last three discretization levels satisfies the ϵ chosen in Step 4.[55,56] If

the ϵ tolerance is too strict (i.e. smaller than the noise inherent in the stochastic multiscale system), MLMC will attempt computationally infeasible discretization levels and not converge. In this situation, the ϵ criterion in Step 4 should be relaxed to enable MLMC to converge.

Identify β_i via NISP

Step 9: Using the data collected in Step 8 with MLMC, calculate the β_i coefficients via the NISP method according to equation (2).

Uncertainty quantification via PCE

Step 10: Estimate the probability distribution of the observable of interest using the PCE expression identified in Step 9, respectively.

The above procedure was followed for both stochastic multiscale models presented in this work. The particulars of Steps 1-4 had to be adjusted to each model since different discretization domains played prominent roles in each simulation code. Furthermore, the models produced observables of different magnitudes. The specific adjustments that were done to Steps 1-4 are described further.

5.3 MLMC for uncertainty quantification in the CVD model

As discussed in sections 3.3 and 5.1.1, the current heuristic for estimating the key observable of the CVD stochastic multiscale system, i.e. roughness at the end of the batch (R_{final}), without incurring excessive computational expense is to simulate six batches (in parallel) at identical conditions with the lattice size of N = 30 and use the average of the six roughness values as an estimate equivalent to the result that would be obtained from a large lattice with low levels of noise (e.g. N = 100, 150),[2,35,37,54,56] as illustrated in Figure 2.

The values obtained in this fashion can subsequently be used to construct closed-form expressions such as PCE. However, the usage of this heuristic does not guarantee that adequate sampling has been conducted during the estimation and the results of PCE based on this heuristic may still vary significantly depending on the drawn samples, affecting the reliability of uncertainty quantification that the PCE would be used for.

In order to compare the performance of the heuristic and MLMC when obtaining the observables necessary to construct PCE, the following three parameters of the CVD model were assumed to be uncertain: the energy associated with a single bond between particles in the thin film, i.e. E in equations (24) and (25), the energy associated with particle migration on the surface of the thin film, i.e. E_m in equation (25), and the gas-phase precursor mole fraction, i.e. X_{bulk} in equation (18). Hence, uncertainty is present at both scales in this multiscale model, similar to section 3.3.2. The parameters were assumed to have normal uncorrelated distributions, i.e. [54]

$$\begin{gathered} E \sim \mathcal{N}(17000\ cal/mol, (850\ cal/mol)^2) \\ E_m \sim \mathcal{N}(10200\ cal/mol, (255\ cal/mol)^2) \\ X_{bulk} \sim \mathcal{N}(2 \times 10^{-6}, (1 \times 10^{-7})^2) \end{gathered} \tag{65}$$

Three methods were employed to approximate the distribution of R_f subject to uncertainty: standard Monte Carlo (MC) sampling and 2nd order PCE NISP[54] with Hermite polynomials and with 3 roots per coefficient, where the pre-requisite R_f values were estimated using the traditional heuristic rule (i.e. 6 duplicates of N = 30 lattices at each combination of the necessary uncertain parameter realizations) and MLMC sampling with the root mean square error tolerance ϵ = 0.1. The ϵ tolerance was arrived at through a trial and error process. As a rule of thumb, we recommend setting the ϵ to 5% of the observable at nominal uncertain parameter values on the first try in the trial and error procedure and then adjusting it until MLMC produces PCE expressions whose predictions are consistent with each other.

During the preliminary sampling stage of MLMC, 100 realizations of the uncertain parameters were sampled at five levels (from l = 0 to l = 4) to estimate the variability of R_f at each of the levels. The 2nd order was chosen for PCE because it could capture asymmetric distributions without incurring the computational costs associated with constructing higher order expansions, as demonstrated in our previous work.[54] Note that the R_f distribution was estimated at T = 940 K since at this industrially relevant temperature the system is in the high-noise regime.[54,56] For MC, sampling was conducted with the lattice size N = 100 in batches of 1,000 until the distribution was observed to converge with 10,000 samples. To assess the performance of the heuristic rule, 200 2nd order PCE expressions were constructed to observe the variability in the resulting R_f probability density approximations. For MLMC, the 2nd order PCE was estimated 10 times to observe the variability in the predictions. When propagating the uncertainty, 10 million realizations of the uncertain parameters were generated and used in the PCE expressions. Note that these calculations are performed very fast (i.e. within 2 seconds) since the PCE expressions are low-order models.

The results are summarized in Table 12, Figure 20 (heuristic-based PCE) and Figure 21 (MLMC-based PCE). Since preliminary MLMC sampling was a one-time procedure and its computational cost was insignificant compared to the time needed to obtain all the necessary data for one MLMC-based PCE, the CPU time for MLMC in Table 12 does not include that cost. For the CVD model at T = 940 K, ϵ = 0.1 was approximately 1.9% of 5.25 ML, which was the R_f value at nominal uncertain parameter realizations. The ϵ = 0.1 value was close to the minimum computationally feasible error tolerance that could be imposed at T = 940 K because of the inherent stochastic noise present in the CVD model, and the usage of ϵ = 0.1 resulted in PCE expressions that could capture the benchmark shape, as shown in Figure 21.

Qualitatively, it can also be seen from Figures 20 and 21 that the PCE expressions obtained by the heuristic method failed to capture the distribution of the benchmark (Figure 20) while the PCEs based on MLMC sampling with the root mean square error tolerance ϵ = 0.1 were able to match the benchmark distribution shape more closely and accurately (Figure 21). Also, the MLMC-based PCE distributions in Figure 21 were quite close to each other. Therefore, observing convergence between multiple MLMC-based PCE can serve as another way to confirm that the chosen ϵ tolerance is adequate.

Table 12 corroborates the observation that the usage of MLMC can help achieve a more accurate approximation of the benchmark because compared to the heuristic, MLMC could significantly reduce the percent deviations in the reported statistics of the final roughness value R_f, i.e. the minimum value $\min R_f$, the maximum value $\max R_f$, the mean value μ_{R_f}, standard deviation σ_{R_f}, skewness s_{R_f} and kurtosis k_{R_f}. For example, consider the improvement in accuracy when estimating the $\min R_f$ value: the prediction could vary by 188 % based on the heuristic method (from underestimation by 148 % to overestimation by 40 %), while the variability from MLMC-based PCE decreased to 36 % (from underestimation by 13 % to overestimation by 23 %). Other significant improvements in variability due to the usage of MLMC were demonstrated for the skewness of final roughness distributions (s_{R_f}) and the maximum predicted final roughness ($\max R_f$). The variability shown by the heuristic method was 222 % for s_{R_f} predictions and 71% for $\max R_f$, while the MLMC technique achieved 41 % variability for s_{R_f} and 12 % for $\max R_f$ (see Table 12).

However, the improvement offered by MLMC comes at a significant computational cost increase compared to the heuristic method; the CPU time to obtain one MLMC-based PCE was two orders

of magnitude greater than the cost to obtain one heuristic-based PCE, as stated in Table 12. However, it should also be emphasized that constructing one heuristic-based PCE does not guarantee that the shape of the benchmark will be adequately captured: out of 200 heuristic-based PCE that were constructed for this scenario, approximately 50 failed to capture the shape of the benchmark by overestimating the R_f values at the left tail and underestimating the skewness of the benchmark distribution (such distributions were shown in Figure 20 with red, green and black lines). Furthermore, constructing additional heuristic-based PCE, even hundreds of them as in this work, can only provide an upper bound on the variability of the statistics of R_f rather than decrease that variability (which MLMC can do by tuning ϵ). Thus, the CPU time for the heuristic method can grow by orders of magnitude without a true improvement in precision and accuracy of predictions. Therefore, the advantage of MLMC compared to the heuristic is the ability to decrease the variability of the predictions by imposing a smaller error tolerance (ϵ) at the expense of additional computational effort. Note that the MLMC method was one order of magnitude (up to 5.3 times) faster than the benchmark's CPU time, as reported in Table 12.

Table 12: Estimating R_{final} in the CVD model using MC sampling and 2nd order PCE NISP constructed with the heuristic rule and MLMC. The columns that correspond to heuristic and MLMC methods contain percentage deviations from the benchmark observed from probability distributions constructed with PCE NISP based on those methods.

Method	MC sampling N = 100 10,000 realizations	Heuristic 6 dupl. N = 30	MLMC $\boldsymbol{\epsilon}$ = 0.1
min R_f	2.82 ML	-148 % to 40 %	-13 % to 23 %
max R_f	11.42 ML	-13 % to 58 %	7 % to 19 %
$\boldsymbol{\mu_{R_f}}$	5.46 ML	-2 % to 7 %	-1 % to 2 %
$\boldsymbol{\sigma_{R_f}}$	0.99 ML	-22 % to 15 %	-15 % to -4 %
$\boldsymbol{s_{R_f}}$	0.68	-140 % to 82 %	-28 % to 13 %
$\boldsymbol{k_{R_f}}$	3.81	-21 % to 34 %	-13 % to 0 %
CPU time	209.3 hours	0.43 hours	39.2 hours

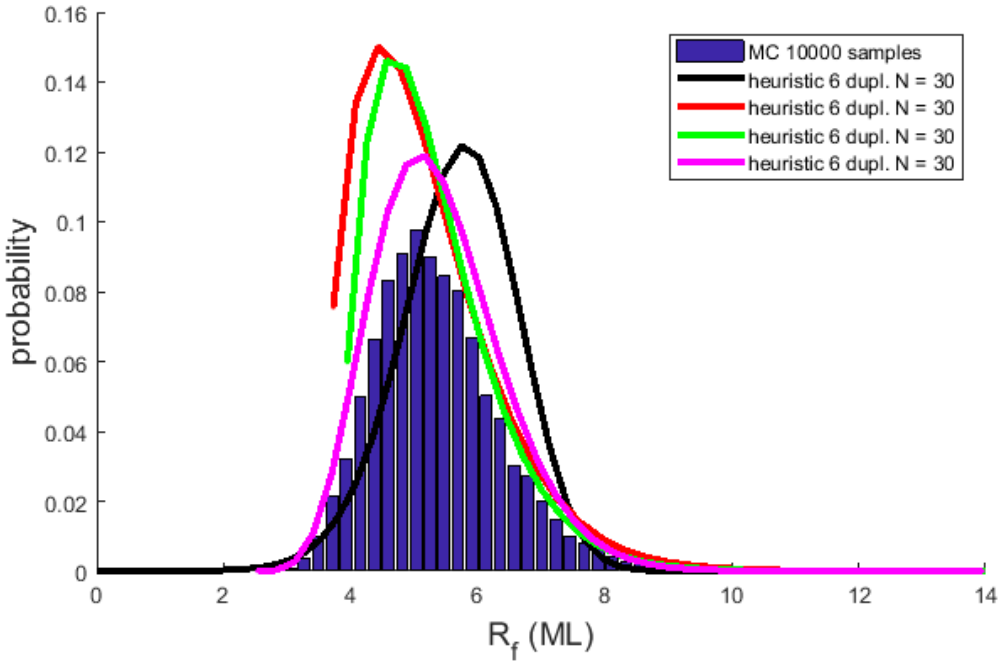

Figure 20: Approximations of the probability distribution of R at the end of the batch using MC sampling and the heuristic method of constructing 2nd order PCE NISP under multiple parameter uncertainty. The heuristic method can cause the PCE-based distribution to not capture the shape of the MC-based benchmark. Four representative PCE distributions were plotted.

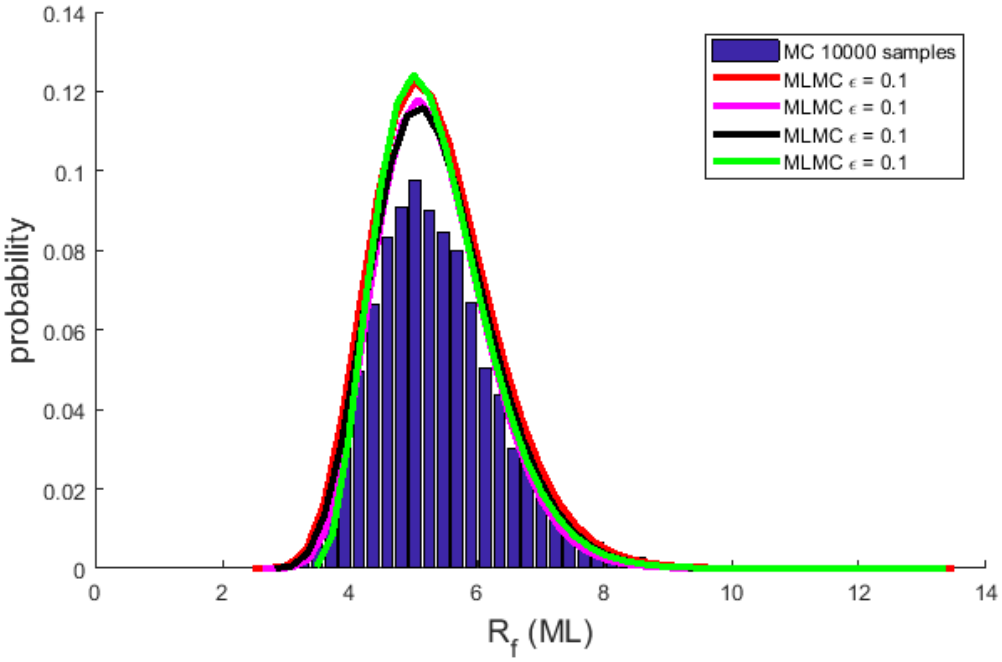

Figure 21: Approximating the distribution of roughness using MLMC-based PCE NISP with the root mean square error tolerance $\epsilon = 0.1$. Four representative PCE distributions were plotted.

5.4 MLMC for uncertainty quantification in a catalytic pore reactor model

The second multiscale model employed in this work is a steady-state model of a plug flow catalytic pore reactor that facilitates a bimolecular reaction,[9,37] as illustrated in Figure 22. Two gaseous species adsorb to vacant sites on the catalyst surface inside the pore to either irreversibly react (which is possible only if they adsorb to adjacent vacant sites) and produce a gaseous product, or desorb without reacting, as per the reaction mechanism (66):

$$\begin{gathered} A + * \rightleftharpoons A^* \\ B + * \rightleftharpoons B^* \\ A^* + B^* \rightarrow AB + ** \end{gathered} \tag{66}$$

where A and B represent the gaseous reactants, $*$ represents one vacant site on the catalyst surface, A^* and B^* represent the adsorbed reactants, and AB represents the gaseous product of the irreversible reaction. The stochastic multiscale model for this system consists of three PDEs coupled with kMC lattices which are assumed to be uniformly distributed along the length of the pore. Each PDE describes the concentration of the corresponding species (A, B or AB) within the catalytic pore, while the kMC lattices simulate the atomistic-scale events (adsorption, desorption, reaction). The two scales are coupled through boundary conditions because the microscale event rates depend on the concentrations of the species, while the concentrations are in turn affected by the number of irreversible reactions that have taken place along the length of the pore. The fundamental equations of the multiscale model are presented below; a detailed description of the model, including parameter definitions and values, can be found in Appendix B and in previous reports.[9,37]

5.4.1 Continuum (macroscale) model

The catalytic pore is assumed to have length $\mathcal{L}$ = 1 μm and radius ρ = 50 nm, as illustrated in Figure 22. The reactants A and B enter the pore and react to produce AB as they flow along the

pore length. The reaction is assumed to be isothermal, and plug flow is assumed to adequately explain the mass transfer inside the pore. Furthermore, the reaction is assumed to occur at steady state. The pore is assumed to have the shape of a perfect cylinder, the radial component of the flow velocity is treated as negligible and the axial component is considered constant.[9,37] Hence, only the mass transfer PDEs are needed to describe the concentration of each of the three species in the pore.

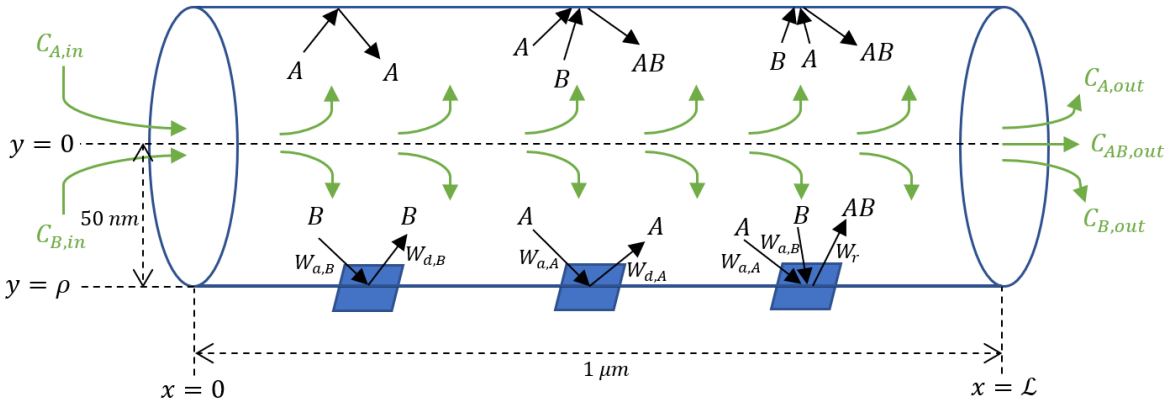

Figure 22: Schematic of the catalytic pore reactor model with reactions happening on the inner surface of the pore. Note that the symbol W refers to the rates of kMC events (adsorption, desorption, reaction) happening to the appropriate species.

Symmetry arguments enable the simplification of the PDEs to 2 dimensions, x and y:

$$D_i \frac{\partial^2 C_i(x,y)}{\partial y^2} - v \frac{\partial C_i(x,y)}{\partial x} = 0 \tag{67}$$

where C_i corresponds to the concentration of species i (i.e. A, B, AB), D_i is the corresponding diffusion coefficient and v is the fluid velocity. For each species the following boundary conditions apply:

$$C_i(0,y) = C_{i,in} \tag{68}$$

$$\frac{\partial C_i(x,0)}{\partial y} = 0 \tag{69}$$

$$D_i \frac{\partial C_i(x,\rho)}{\partial y} = -\omega_{i,ads}(x,\rho) + \omega_{i,des}(x,\rho) + \omega_{i,prod}(x,\rho) - \omega_{i,cons}(x,\rho) \tag{70}$$

where $C_{i,in}$ is the inlet concentration of species i, and $\omega_{i,ads}$, $\omega_{i,des}$, $\omega_{i,prod}$ and $\omega_{i,cons}$ are the rates of adsorption, desorption, production and consumption of species i, respectively. The coupling of the PDEs to kMC is accomplished through boundary condition (70) because the rates cannot be calculated accurately using the continuum assumption.[9,37]

5.4.2 Microscale kinetic Monte Carlo model

As mentioned above, the on-lattice kMC model with periodic boundary conditions simulates reaction mechanism (66) occurring on the catalyst surface (i.e. at $y = \rho$). The kMC rates of adsorption, desorption and reaction events, respectively, are calculated as follows:

$$W_{a,i}(x,\rho) = C_i(x,\rho) N_e(x,\rho) k_{a,i} \tag{71}$$

$$W_{d,i}(x,\rho) = N_i(x,\rho) k_{d,i} \mathrm{e}^{-\frac{E_{d,i}}{RT}} \tag{72}$$

$$W_r(x,\rho) = N_r(x,\rho) k_r \mathrm{e}^{-\frac{E_r}{RT}} \tag{73}$$

where N_e denotes the number of empty catalytic surface sites, N_i represents the quantity of adsorbed molecules of species i, and N_r is the number of A^* and B^* pairs adsorbed at adjacent catalyst surface sites ready to react to produce AB. The definitions of other parameters can be found in Appendix B and in previous works.[9,37] Note that the kMC rate of adsorption, i.e. equation (71), depends upon the solution of PDE (67), thereby necessitating the coupling between the macro and micro scales. The kMC code tracks the values of the parameters N_e, N_i, and N_r. Whenever an event is executed, the kMC time is incremented as follows:

$$\tau_\iota = -\frac{\ln(\zeta)}{\sum_s^{ns}\left[W_{a,s}(x,\rho) + W_{d,s}(x,\rho)\right] + W_r(x,\rho)} \tag{74}$$

where ζ is a uniform random number from the interval (0,1). The coupling between the kMC and PDE scales is carried out when the kMC simulation reaches steady state in the parameters N_e, N_i, and N_r, at which point the elapsed kMC time exceeds the coupling interval duration.

Since kMC simulations are computationally intensive, it is not possible to use kMC at every possible value of x along the length of the catalytic pore. Instead, the kMC lattices, also referred to as "teeth", are placed at equidistant intervals, referred to as "gaps", along the length of the pore; hence the name "gap-tooth" model. The results from the individual teeth are interpolated over the gaps prior to being provided to the boundary condition (70). Hence, this model uses the multigrid approach (mentioned in section 2.1) to implement the coupling between macro and micro scales. The rates for this boundary condition are determined using the interpolated data from the kMC simulation as follows:

$$\omega_{i,ads}(x,\rho) = \kappa k_{a,i} C_i(x,\rho) \frac{N_e(x,\rho)}{N_{total}} \tag{75}$$

$$\omega_{i,des}(x,\rho) = \kappa k_{d,i} \mathrm{e}^{-\frac{E_{d,i}}{RT}} \frac{N_i(x,\rho)}{N_{total}} \tag{76}$$

$$\omega_{i,prod}(x,\rho) = \kappa \frac{M_{i,prod}(x,\rho)}{N_{total} \sum_{\iota=1}^{n_\tau(x,\rho)} \tau_\iota(x,\rho)} \tag{77}$$

$$\omega_{i,cons}(x,\rho) = 0 \tag{78}$$

where N_{total} is the total number of sites on one kMC lattice. In this work, N_{total} = 900 since N = 30 lattice sizes were used at each tooth. Note that the adsorption and desorption rates for each species, equations (75) and (76), respectively, have an explicit dependence on N_e and N_i provided by kMC. Equation (77) indirectly depends on N_r because $M_{i,prod}$ is the number of gas-phase species i being produced. Also, note that equation (78) is equal to zero because it pertains to the consumption of gas-phase species, but in this study the adsorbed species are the only ones being consumed. The definitions of other parameters in equations (75)-(78) can be found in Appendix B

and in the literature.[9,37] A complete cycle of the multiscale model can be described as follows: the fluid phase is analyzed using the continuum models, and the fluid species concentrations $C_i(x,\rho)$ are extracted along the length of the catalyst surface in order to calculate the catalyst adsorption rates in equation (71) at each tooth. Subsequently, the kMC model is operated at each tooth until it reaches steady state as described above. The parameters N_e, N_i, and $M_{i,prod}$ are extracted at each tooth and interpolated across each gap, and then used to update the surface boundary condition (70) using equations (75)-(78). The feedback loop between the continuous and discrete scales is repeated until the steady state solution to PDE (67) is obtained for each of the concentrations of the three species in reaction (66). For the purposes of this work, the key observable of interest was the concentration of the product of reaction (66) at the outlet of the catalytic pore, i.e. $C_{AB}(\mathcal{L},y)$ or $C_{AB,out}$ for short, since this outlet concentration exhibited the largest variability among the three species A, B, and AB (plots not shown for brevity).

Note that the number of teeth, i.e. n, affects both the accuracy of the calculated concentrations and the computational cost of the multiscale model. Also, the uncertainty exhibited in the pore may depend on the spatial coordinates. Furthermore, the usage of kMC at each tooth increases the noise present in this system; thus, great care is required in the selection of coupling intervals between the discrete and continuous scales in order to avoid numerical instability. Consequently, the complexity of the catalytic pore model exceeds that of the CVD model and adds more degrees of freedom to the uncertainty quantification procedure.

5.4.3 MLMC sampling applied to the catalytic reactor model

For the catalytic pore reactor model, MLMC was used to control the spatial discretization through defining the number of teeth along the length of the pore (n). As the number of teeth grows, the noise in the estimate of $C_{AB,out}$ diminishes. Hence, the MLMC technique can enable efficient and

accurate calculation of $C_{AB,out}$ values which can be used to construct the PCE expressions via NISP method for uncertainty quantification, as discussed previously in sections 5.2 and 5.3.

It was assumed that the two halves of the catalytic pore were subject to different uncertainties. The allowed total numbers of teeth at a level were restricted to positive even values in order to ensure that an equal number of teeth was used for each of the two halves of the reactor:

$$n_l = 2^{l+1} \tag{79}$$

Note that in this scenario the minimum number of teeth (i.e. n_0 at l = 0) is 2 and that the number of teeth grows exponentially with the level number. The choice n_0 = 2 ensured a low computational expense at MLMC level 0 while also avoiding the excessive noise that would be exhibited due to an insufficient number of teeth (i.e. 1) that would ignore the effect of spatially varying uncertainty. The exponential growth scheme presented in equation (79) was chosen since it limited the growth of the number of teeth and kept the higher MLMC discretization levels computationally feasible as opposed to using 4^{l+1} or even higher spatial discretization refinement factors.

5.4.4 The results of MLMC-based PCE approximations (Catalytic Pore model)

To consider the impact of catalyst deactivation/catalytic surface fouling on the final product, the activation energy of desorption for species A, i.e. $E_{d,A}$ in equations (72) and (76) with $i = A$, was assumed to be uncertain and the distribution of the concentration $C_{AB,out}$ was estimated. Spatially varying uncertainty was assumed at $y = \rho$ along the length of the reactor such that the first half of the reactor (inlet to midpoint) was subject to lower variance than the second half:

$$\begin{aligned} E_{d,A}^{(1)} &\sim \mathcal{N}(95000\ kJ/mol\ , (1900\ kJ/mol)^2) \text{ for } x \in \left[0, \tfrac{L}{2}\right), y = \rho \\ E_{d,A}^{(2)} &\sim \mathcal{N}(95000\ kJ/mol\ , (4750\ kJ/mol)^2) \text{ for } x \in [\tfrac{L}{2}, L], y = \rho \end{aligned} \tag{80}$$

where $E_{d,A}^{(1)}$ and $E_{d,A}^{(2)}$ represent the uncertain parameters in the first and second halves of the catalytic pore, respectively. Consequently, the PCE expression was formulated as a function of two uncertain parameters.

Two methods were employed to approximate the distribution of $C_{AB,out}$ subject to uncertainty: standard Monte Carlo (MC) sampling and 2nd order PCE NISP with Hermite polynomials with 3 NISP roots per coefficient, where the pre-requisite $C_{AB,out}$ values were estimated using MLMC sampling with the root mean square error tolerances $\epsilon = 1\times10^{-5}$ and $\epsilon = 5\times10^{-6}$. Similar to the CVD model scenario, the 2nd order for PCE was chosen to keep the identification costs low while enabling the resulting PCE expression to capture asymmetric probability distributions. Also similar to the CVD scenario, 100 realizations of the uncertain parameter were sampled at the first five levels (from $l = 0$ to $l = 4$) during the preliminary sampling of MLMC (the associated computational cost was not tracked in the total CPU time for the MLMC method for the same reasons as in the CVD model case study). When obtaining the MC benchmark, sampling was conducted using 12 teeth along the length of the catalytic pore in batches of 1,000 samples until the distribution converged; 10,000 samples were collected. To test the precision of MLMC calculations, four PCE expressions were identified for each ϵ value. Note that more PCE expressions could be obtained, but they were outside the allocated computational budget. When propagating the uncertainty, 10 million realizations of $E_{d,A}^{(1)}$ and $E_{d,A}^{(2)}$ were generated and used in the PCE expressions.

The resulting distributions of $C_{AB,out}$ from both cases are presented in Figures 23 ($\epsilon = 1\times10^{-5}$) and 24 ($\epsilon = 5\times10^{-6}$). Note that the ϵ values of 1×10^{-5} and 5×10^{-6} were 3.0% and 1.5%, respectively, of 3.35×10^{-4} mol/L, which was the $C_{AB,out}$ value observed at nominal uncertain parameter

realizations. Both of the utilized ϵ tolerances resulted in adequate PCE performance (as shown in Figures 23 and 24), which corroborates our recommendation that ϵ should initially be set to less than 5% of the observable value obtained at nominal uncertain parameters (and subsequently tuned based on the available computational budget and precision requirements).

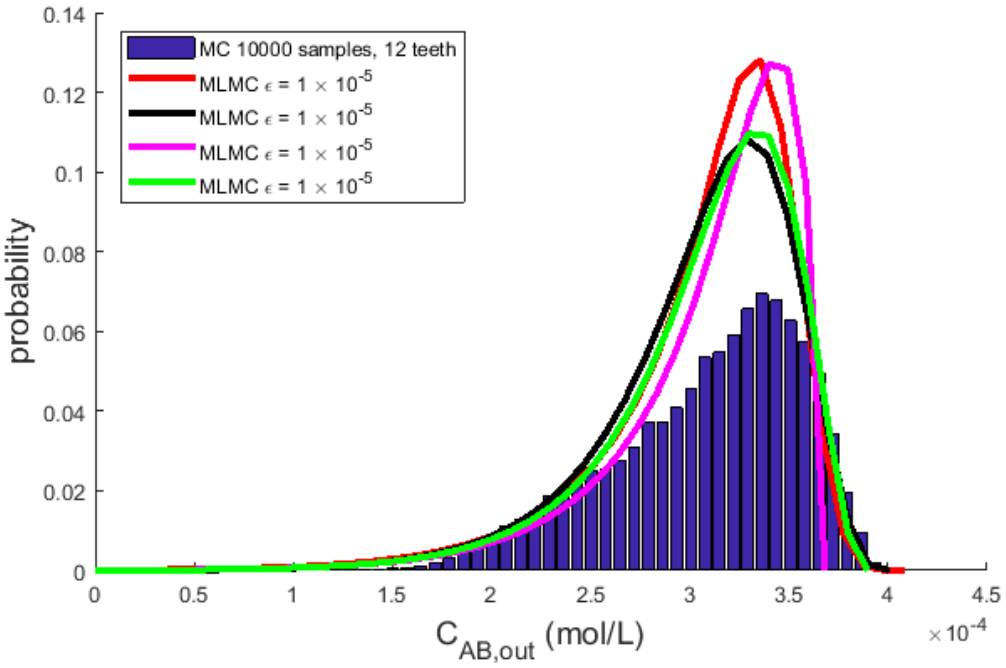

Figure 23: Approximations of the probability distribution of $C_{AB,out}$ with MC sampling and 2nd order PCE NISP constructed using MLMC with $\epsilon = 1{\times}10^{-5}$.

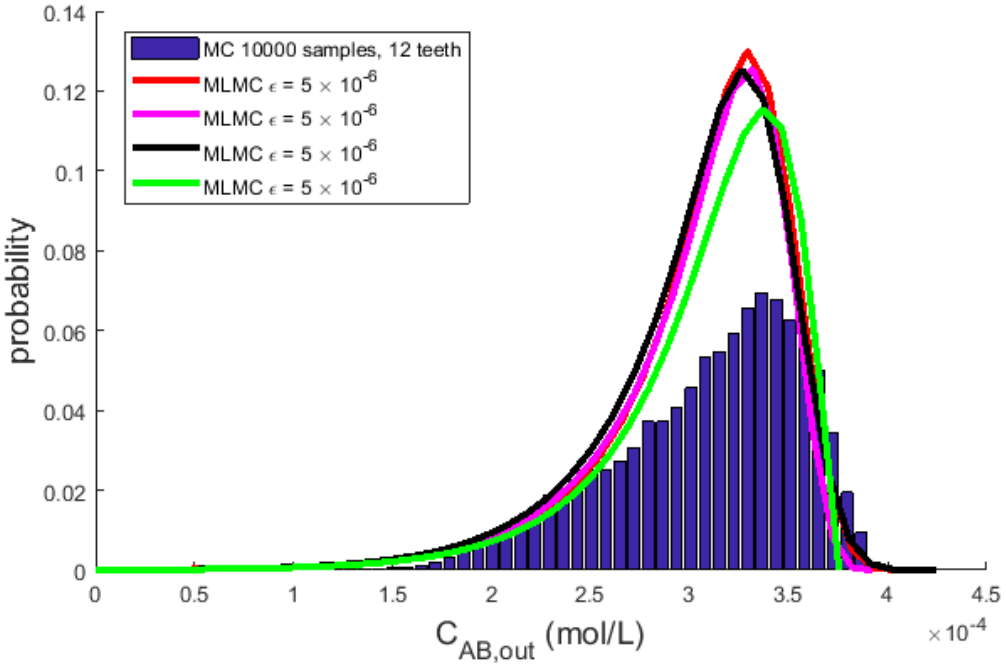

Figure 24: Approximations of the probability distribution of $C_{AB,out}$ with MC sampling and 2nd order PCE NISP constructed using MLMC with $\epsilon = 5\times10^{-6}$.

Furthermore, the results are summarized in Tables 13 and 14. Table 13 shows the variability in the $C_{AB,out}$ values that were used to calculate the four sets of PCE coefficients by the NISP method at each ϵ value of MLMC. The variability in $C_{AB,out}$ (denoted in Table 13 as max $C_{AB,out}$ - min $C_{AB,out}$) was due to the stochastic noise in the catalytic pore model. The noise-induced variability was the reason why the PCE-based distributions in Figures 23 and 24 differed from each other (i.e. the noise in $C_{AB,out}$ propagated into the PCE coefficients calculated by the NISP method). The variability was assessed based on the four $C_{AB,out}$ values obtained from the catalytic pore model by MLMC sampling at every combination of the realizations of $E_{d,A}^{(1)}$ and $E_{d,A}^{(2)}$ that corresponded to the NISP roots. Table 14 summarizes the deviations of MLMC-based PCEs from the MC-based benchmark in terms of the minimum, maximum, mean, standard deviation, skewness and kurtosis of $C_{AB,out}$ (i.e. min $C_{AB,out}$, max $C_{AB,out}$, $\mu_{C_{AB,out}}$, $\sigma_{C_{AB,out}}$, $s_{C_{AB,out}}$, and $k_{C_{AB,out}}$, respectively).

Table 13: The precision in the estimates of $C_{AB,out}$ was improved by decreasing the ϵ value

$E_{d,A}^{(1)}$ realization (J/mol)	$E_{d,A}^{(2)}$ realization (J/mol)	$\max C_{AB,out}$ - $\min C_{AB,out}$ (mol/L)		Decrease in the variability of $C_{AB,out}$ due to ϵ tuning
		MLMC $\epsilon = 1\times10^{-5}$	MLMC $\epsilon = 5\times10^{-6}$	
91709	86773	1.36×10^{-5}	2.80×10^{-6}	79 %
91709	95000	1.68×10^{-5}	7.50×10^{-6}	55 %
91709	103230	1.74×10^{-5}	4.40×10^{-6}	75 %
95000	86773	1.13×10^{-5}	3.70×10^{-6}	67 %
95000	95000	1.06×10^{-5}	4.40×10^{-6}	58 %
95000	103230	1.47×10^{-5}	9.40×10^{-6}	36 %
98291	86773	1.28×10^{-5}	1.02×10^{-5}	20 %
98291	95000	1.43×10^{-5}	9.50×10^{-6}	34 %
98291	103230	2.77×10^{-5}	1.06×10^{-5}	62 %

Table 14: Estimating $C_{AB,out}$ using MC sampling and 2nd order PCE NISP constructed with MLMC. The last two columns contain percent deviations of MLMC-based PCEs from the MC-based benchmark in terms of the minimum, maximum, mean, standard deviation, skewness and kurtosis of $C_{AB,out}$.

Method	MC sampling, 12 teeth 10,000 realizations	MLMC $\boldsymbol{\epsilon} = 1\times10^{-5}$	MLMC $\boldsymbol{\epsilon} = 5\times10^{-6}$
$\boldsymbol{\min C_{AB,out}}$	1.21×10^{-4} mol/L	-100 %	-100 %
$\boldsymbol{\max C_{AB,out}}$	4.01×10^{-4} mol/L	-8 % to 2 %	-6 % to 6 %
$\boldsymbol{\mu_{C_{AB,out}}}$	3.12×10^{-4} mol/L	-0.4 % to 0.8 %	-1 % to 0.5 %
$\boldsymbol{\sigma_{C_{AB,out}}}$	4.77×10^{-5} mol/L	-5 % to -1 %	-6 % to -4 %
$\boldsymbol{s_{C_{AB,out}}}$	-0.70	75 % to 142 %	93 % to 119 %
$\boldsymbol{k_{C_{AB,out}}}$	2.89	86 % to 143 %	103 % to 129 %
CPU time	190.6 hours	4.25 hours	22.4 hours

A qualitative examination of Figures 23 and 24 shows that the decrease of ϵ from 1×10^{-5} to 5×10^{-6} caused the MLMC-based PCE probability distributions to be more consistent with each other, although at both ϵ values the stochastic noise in $C_{AB,out}$ estimates prevented some PCE from capturing the right tail of the benchmark (represented by the magenta line in Figure 23 and the green line in Figure 24). An examination of Table 13 explains the underlying reason for this

qualitative observation. For each realization of the uncertain parameters, the variability in the obtained $C_{AB,out}$ values diminished as ϵ was decreased. Consequently, the variability in the PCE coefficients calculated by the NISP method using the obtained $C_{AB,out}$ values also decreased. The improvements in the variability of $C_{AB,out}$ ranged from 20 % (a decrease from 1.28×10^{-5} mol/L to 1.02×10^{-5} mol/L for the uncertain parameter realizations 98291 J/mol and 86773 J/mol) to 79 % (where the decrease was from 1.36×10^{-5} mol/L to 2.80×10^{-6} mol/L for realizations 91709 J/mol and 86773 J/mol). On average, the variability in $C_{AB,out}$ decreased by 54 %. While it is possible for the variability to increase in either scenario with additional sampling of $C_{AB,out}$ values by MLMC, the results indicate that tuning the ϵ value helps to increase the precision of the estimates of the observables and thereby improve the consistency of PCE predictions (which is not a feature of other heuristic rules for sampling stochastic multiscale systems to construct uncertainty quantification expressions such as PCE).

Table 14 indicates that tuning the ϵ value decreased the variability in the predictions of such statistical moments as standard deviation $\sigma_{C_{AB,out}}$, skewness $s_{C_{AB,out}}$, and kurtosis $k_{C_{AB,out}}$. The variability in $s_{C_{AB,out}}$ and $k_{C_{AB,out}}$ demonstrated the most notable improvements. For example, at $\epsilon = 1\times10^{-5}$, $s_{C_{AB,out}}$ varied by 67 % (overestimation by 75 % to 142% of the benchmark value) while at $\epsilon = 5\times10^{-6}$ the variability in $s_{C_{AB,out}}$ decreased to 26 % (overestimation by 93 % to 119 %). The variability in $k_{C_{AB,out}}$ decreased from 57 % ($\epsilon = 1\times10^{-5}$) to 26 % ($\epsilon = 5\times10^{-6}$). Note that tuning the ϵ did not improve the accuracy in the estimations of skewness and kurtosis because for both ϵ the $\min C_{AB,out}$ value obtained from the benchmark was strongly underestimated by the PCE-based expressions (see Table 14). However, this is a shortcoming of the PCE technique (more NISP roots and/or higher PCE order may be needed) rather than the MLMC method. MLMC could

control the variability in the observable values obtained from the catalytic pore model, as per Table 13. Also, while there were PCE expressions that did not adequately capture the right tail of the benchmark in Table 14 and Figures 23 and 24, this indicated the sensitivity of PCE to the noise in $c_{AB,out}$ rather than MLMC's inability to estimate the observables according to the specified mean square error tolerances. Despite these shortcomings, the PCE-based estimates of the observable's distributions provided a reasonable approximation of the benchmark distribution's shape at a computational cost that was 1-2 orders of magnitude lower than MC sampling thanks to the usage of MLMC method, as shown in Table 14.

5.5 Chapter Summary

In this study, the Multilevel Monte Carlo (MLMC) sampling technique was adapted to stochastic multiscale systems. Unlike in most other MLMC implementations, which discretize the time domain, in this work the spatial discretization domain was used in the MLMC scheme. As a case study, MLMC was applied to models that represented thin film formation by chemical vapour deposition (CVD) and a catalytic pore reactor. First, MLMC and Monte Carlo (MC) sampling techniques were used to estimate the noise in the observable of the CVD model, i.e. the roughness of the thin film. The greatest computational efficiency of MLMC was observed when the system was in the low-noise regime. However, while the potential computational time savings offered by MLMC were an order of magnitude compared to MC sampling, MLMC overestimated the noise values obtained from MC. The tuning of the root mean square error tolerance (ϵ) of the MLMC technique improved the accuracy of the noise estimates, but it relied on *a priori* knowledge of the system's behaviour. It was also found that unlike for the case of closed-form continuous systems of equations, in stochastic multiscale systems MLMC could not satisfy every imposed ϵ at every set of simulation conditions, which suggests the existence of a lower bound on the error tolerance.

Next, MLMC was applied to improve uncertainty quantification via polynomial chaos expansions (PCE) in the CVD model with parametric uncertainty at both simulation scales and the catalytic pore reactor model with spatially varying uncertainty. The usage of MLMC and the tuning of ϵ allowed to improve the precision with which the systems' observables were identified. Since the observables were used to calculate the coefficients of the PCE using MLMC, the predictions made by PCE also became more consistent with each other. Despite the greater computational expense associated with MLMC, its ability to control the precision of PCE identification was a substantial improvement over the commonly used heuristic rule for obtaining low order models (e.g. PCE) for uncertainty quantification. For the CVD model, MLMC-based PCE was notably more accurate than the heuristic method of constructing PCE. However, while for the catalytic pore model the tuning of the ϵ tolerance caused the identified PCEs to be more consistent with each other, they strongly underestimated the values in the left tail of the benchmark distribution. Thus, it is recommended to select an uncertainty quantification technique that would be most appropriate and accurate for a given stochastic multiscale system and use MLMC to improve the precision in the identification of the uncertainty quantification expression (e.g. PCE, power series expansion (PSE) or others).

Chapter 6: Artificial Neural Networks (ANNs) for the Control of Stochastic Multiscale Systems

Efficient sampling and uncertainty quantification techniques such as PSE, PCE and MLMC can estimate the statistical moments and distributions of the observables, but they do not have the ability to predict the dynamic responses of the approximated quantities. Once the process settings change, the identification of PSE sensitivities and PCE coefficients, as well as the MLMC procedure, would have to be repeated at the new settings, which can be computationally expensive. The computational cost makes the applications of these techniques challenging, especially for online optimization and control under uncertainty.

This challenge can be overcome using data-driven models such as Artificial Neural Networks (ANNs). These modelling tools are able to provide efficient and accurate predictions of time series data by identifying the latent relationships between the manipulated variables and the observables. Although substantial computational resources are required to generate large datasets necessary to identify the ANNs with good interpolation (and extrapolation) abilities, the computational efficiency of ANNs enables efficient control of stochastic multiscale systems.

This chapter presents studies that employ ANNs in the shrinking horizon nonlinear model predictive control (NMPC) of the thin film deposition system discussed in section 3.2. First, two ANNs were trained to predict the film's roughness and growth rate under nominal conditions (no parametric uncertainty was considered). The ANNs were subsequently employed in a shrinking horizon optimization scheme to obtain the optimal time-varying profiles of the manipulated variables (temperature and precursor mole fraction) that would meet the desired thin film properties at the end of the batch. The resulting profiles were implemented on the thin film deposition system and a good agreement with the predictions of the ANNs was observed.

Next, uncertainty in the energy associated with a single bond between particles was considered. Additional ANN was identified using a new dataset to predict the standard deviation of roughness that arose due to the parametric uncertainty, and all three identified ANNs were employed in a reformulated shrinking horizon NMPC for open-loop control of the thin film batch process. Furthermore, an alternative shrinking horizon NMPC for closed-loop control was developed. The same datasets that were used to identify the ANNs listed above were instead used to train ANNs for specific uncertain parameter realizations. Next, maximum likelihood estimation was used for uncertain parameter estimation to identify the best-fit ANN for predicting the roughness trajectory under time-invariant uncertainty. In all studies, the robustness of ANNs to disturbances unseen in the training data was also tested.

6.1 Nonlinear model predictive control (NMPC) framework

The production of smooth and sufficiently thick films is critical for the semiconductor industry, since both properties increase electrical conductivity.[87,88,207] The roughness and thickness properties are likewise significant for biomedical and optical applications of thin films.[1] Furthermore, a high growth rate Gr would increase the throughput of the manufacturer and contribute to their profits by satisfying customer demand in a timely manner. However, achieving low roughness R and high Gr values under uncertainty requires optimal control actions in the manipulated variables (the inlet precursor mole fraction X_{bulk} and the substrate temperature T_{surf}).

The relationships between the variables of the batch process for thin film manufacturing are nonlinear. Therefore, the process may be amenable to nonlinear model predictive control (NMPC).[95,141] NMPC relies on a nonlinear dynamic model to predict the responses of the observables to changes in the manipulated variables. The dynamic model needs to be

computationally efficient in online closed-loop NMPC applications, where the observable variables can be monitored and adjusted online based on the incoming measurements until the end of the batch. Hence, the shrinking horizon optimization scheme is best suited for finding the optimal profiles in T_{surf} and X_{bulk} that would satisfy the final product quality and manufacturing rate requirements. In the shrinking horizon optimization approach, the goal is to drive the process to some desired target while adjusting the control actions online in the presence of uncertainty. Equation (81) describes the formulation of the shrinking horizon optimization problem that can be used to find the optimal time-dependent T_{surf} and X_{bulk} profiles that would maximize the expected Gr during the production batch (and, consequently, the total Th) while satisfying the desirable roughness R_{kMC} under uncertainty at the end of the batch. Note that this formulation, as well as other formulations presented later in this chapter, can also be considered as economic model predictive control:

$$\max_{T_{surf}(t_i),\, X_{bulk}(t_i)} \quad \frac{1}{n}\sum_{i=1}^{n} Gr\left(T_{surf}(t_i), X_{bulk}(t_i)\right)$$

Subject to:

$$\begin{aligned}
& f_{kMC-PDE}\left(T_{surf}(t_i), X_{bulk}(t_i), E, R_{kMC}(t_i), Gr_{kMC}(t_i), Th_{kMC}(t_i)\right) = 0, \quad \text{Eqs. (13)-(30)} \\
& t_i = i\Delta t; \quad \forall i = 0,1,2,\dots n \\
& R\left(T_{surf}(t_n), X_{bulk}(t_n)\right) \le R^* \\
& T_{min} \le T_{surf}(t_i) \le T_{max} \\
& X_{min} \le X_{bulk}(t_i) \le X_{max} \\
& T_{surf}(0) = T_0 \\
& X_{bulk}(0) = X_0 \\
& -\Delta T \le T_{surf}(t_{i+1}) - T_{surf}(t_i) \le \Delta T \\
& -\Delta X \le X_{bulk}(t_{i+1}) - X_{bulk}(t_i) \le \Delta X
\end{aligned} \tag{81}$$

where t_i represents the timepoints at which the process' observables are sampled or measured disturbances can be detected (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; $f_{kMC-PDE}$ is the stochastic multiscale model subject to time-invariant uncertainty in E (i.e. the energy associated with a single bond between particles in the thin film), Gr is the growth rate under uncertainty, R is the roughness under uncertainty, and R^* is the maximum allowed roughness constraint at the end of the batch. T_{min} and T_{max} are the minimum and maximum allowed temperature values, respectively, and X_{min} and X_{max} are the minimum and maximum allowed precursor mole fraction values, respectively. The settings T_0 and X_0 are the initial conditions for the manipulated temperature and precursor mole fraction, respectively, while ΔT and ΔX are the maximum allowed changes in the respective manipulated variables between adjacent sampling intervals. Note that the uncertainty in E affects W_d and W_m, i.e. the rates of desorption and migration (equations (24) and (25), respectively), and consequently it can impact the thin film's growth rate and roughness. The usage of Arrhenius-type expressions in equations (24) and (25) indicates that larger magnitudes of E would slow down both W_d and W_m rates, thereby causing slower growth rate and higher roughness. Also, E is expected to change from batch to batch due to changes in the equipment specifications and process variability (e.g. changes in the property of the raw materials).

In each iteration of the shrinking horizon optimization problem presented above, only the first control action from the optimal profile is implemented on the process. When new information becomes available during the batch run (e.g. a new measurement from the process or a detected disturbance to the operating conditions), it is used as the initial condition for the next iteration of shrinking horizon optimization, where a new set of optimal control actions are obtained for the shorter prediction horizon. The optimization scheme proceeds until the end of the production batch is reached.

The shrinking horizon NMPC framework seeks for the optimal control actions during the operation of the process (i.e. online) in the presence of time-invariant uncertainty in E. Hence, having a computationally efficient method of identifying the optimal control action profiles within a time interval Δt is critical for the optimal design of thin films. However, due to the presence of non-closed-form calculations at the discrete scale, the full kMC-PDE model comprised of equations (13)-(30), i.e. $f_{kMC-PDE}$, is computationally intensive, and its application in equation (81) for online control is computationally intractable, especially in the presence of parametric uncertainty. Thus, to make the problem of equation (81) computationally feasible, we trained ANN models to predict the dynamic responses of $R\left(T_{surf}(t_i), X_{bulk}(t_i)\right)$ and $Gr\left(T_{surf}(t_i), X_{bulk}(t_i)\right)$ under uncertainty in E and replaced the full kMC-PDE model with the ANNs. The training procedure and the structure of the ANNs are discussed in the next section.

6.2 Training ANNs for stochastic multiscale systems

In order to train the ANNs, we used the full multiscale model to generate two datasets under uncertainty. Uncertainty in the energy of a single bond, i.e. the *E* parameter in equations (24) and (25) which describe the rates of desorption and migration, respectively, is assumed to be an uncertain parameter that follows a normal probability distribution function, i.e.

$$E \sim \mathcal{N}\left(17{,}000\frac{cal}{mol},\ \left(850\frac{cal}{mol}\right)^2\right) \tag{82}$$

The value of 850 cal/mol was chosen for testing our approach and its ability to control thin film product properties in the presence of parameter uncertainty. In principle, this value could be obtained from process heuristics through the implementation of a parameter estimation technique, which is beyond the scope of this work. Note that other probability distributions for the uncertain parameters can be used for the design of ANNs. In this work however, we decided to use a

Gaussian distribution because it is the most practical assumption considered in an industrial setting. Also, a single uncertain parameter was considered in this work for simplicity. However, the present approach can be readily extended to consider multiple uncertain parameters in the analysis.

The first dataset was generated using only the nominal uncertain parameter value (E = 17,000 cal/mol) and it contained 400,000 points (one point per second), where each point consisted of the manipulated variables (the substrate temperature T_{surf} and the inlet precursor mole fraction X_{bulk}), and the corresponding outputs that are needed to evaluate the objective function and constraints considered in the shrinking horizon optimization formulation (81), i.e. roughness R and growth rate Gr. Since the uncertain parameter was set to its nominal value, this dataset was used to train the ANNs for predicting μ_R and μ_{Gr}, i.e. the mean values of roughness and growth rate, respectively.

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 T_{surf} was 50 K (to avoid unrealistically high temperature transitions/thermal shock). For similar reasons, the precursor mole fraction was constrained to the range between 2.5×10^{-6} and 4×10^{-6} with the maximum allowed change in X_{bulk} being 5×10^{-7}. The values of T_{surf} and X_{bulk} 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 (400,000 points) 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, as discussed in

section 2.3). 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 for 400,000 seconds was approximately 72 hours. Note that all calculations in this work were performed on a computer with 96 GB of RAM and two Intel® Xeon® E5-2620 v4 processors running at 2.10 GHz. To circumvent the previously mentioned challenges that arise from using stochastic multiscale models to generate ANN training data, we controlled the lattice size N (spatial domain discretization) of our multiscale model. In order to decrease the noise in the training data while maintaining feasible computational times, N was set to 50. Also, we resorted to parallel computing to speed up the generation of ANN training data under uncertainty (described below). Note that the noise could be diminished further with larger lattices such as $N = 100$ or $N = 150$, but at a disproportionately large increase in the computational time. For example, when the value of N is increased by a factor of 1.5, the computational cost grows by approximately a factor of 4, as discussed in our previous works.[54,56]

The second dataset was generated using 300 realizations from the probability distribution of the uncertain parameter E shown in equation (82). For each of 300 realizations, the full multiscale model was simulated for 10,000 s using the same time-varying signal in T_{surf} and X_{bulk} subject to the restrictions described above, thereby generating the values of R and Gr under various realizations in E. Hence, the raw data used in the ANN identification contained 3,000,000 points (300 realizations × 10,000 s). The data recorded from these experiments was used to calculate standard deviations in R and Gr, i.e. σ_R and σ_{Gr}, respectively, for each second in the 10,000 s signal. Consequently, the second dataset consisted of 10,000 points, where each point consisted of the values for T_{surf}, X_{bulk}, σ_R and σ_{Gr}. A representative plot of the sensitivity of R and Gr to uncertainty is presented in Figure 25.

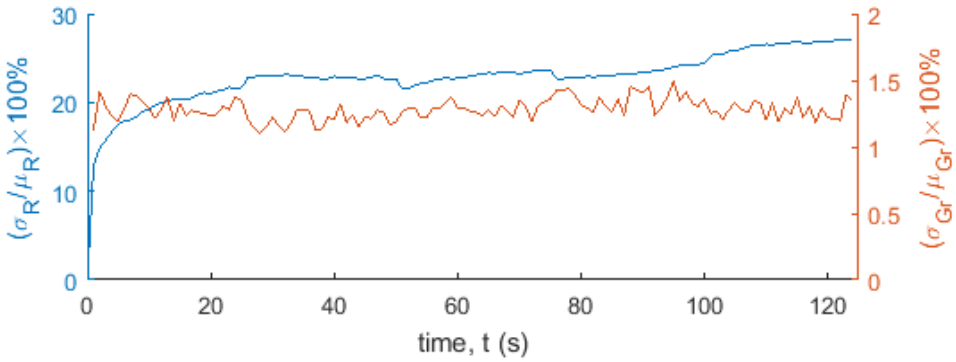

Figure 25: The sensitivity of roughness and growth rate observables to parametric uncertainty, presented as the ratio of the standard deviation of the observables to their respective mean values.

As shown in Figure 25, the growth rate observable is relatively insensitive to the parametric uncertainty in E, since the standard deviation of growth rate σ_{Gr} tends to be less than 1.5% of its mean value μ_{Gr}. On the other hand, the standard deviation in roughness, i.e. σ_R, can reach up to 30% of its mean value μ_R. Thus, we decided to treat the standard deviation in growth rate as negligible. Note that the usage of the dataset obtained under 300 realizations of E and the ANNs trained on it are described in sections 6.4 and 6.5.

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 T_{surf} and X_{bulk}) and their own past predictions (i.e. the past predictions of μ_R, σ_R and μ_{Gr} were supplied to the respective ANN via closed-loop feedback). The autoregressive property of our ANNs allowed them to generate predictions in the absence of online measurements from the plant (which would not be the case for feedforward ANNs). The two datasets of 400,000 and 10,000 points were randomly subdivided into training, validation and test subsets: 70%, 15% and 15% of each dataset 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 (discussed in section 2.3), 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. That is, 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 T_{surf} and X_{bulk}, 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 R and Gr 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 T_{surf} and X_{bulk} 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 15 and 16, respectively. Similarly, the probability distributions of the errors have been plotted in Figures 26 and 27, 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 15: 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 16: 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

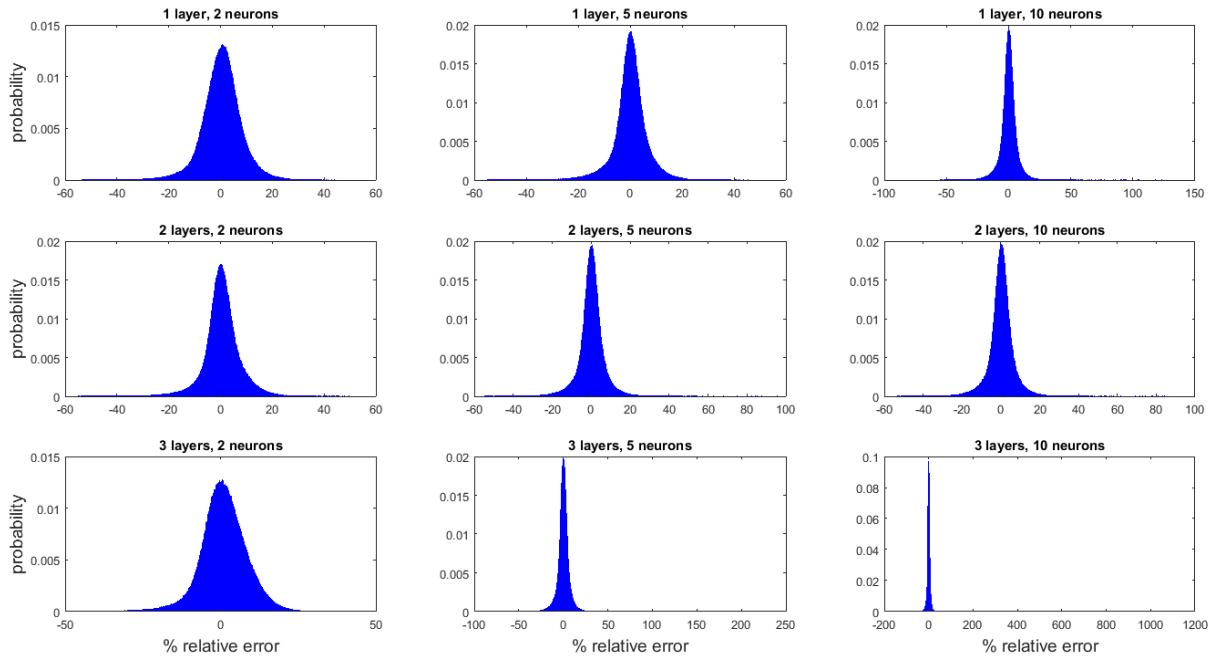

Figure 26: Probability distributions of the percent relative errors between 1,600,000 predicted and actual roughness (R) responses

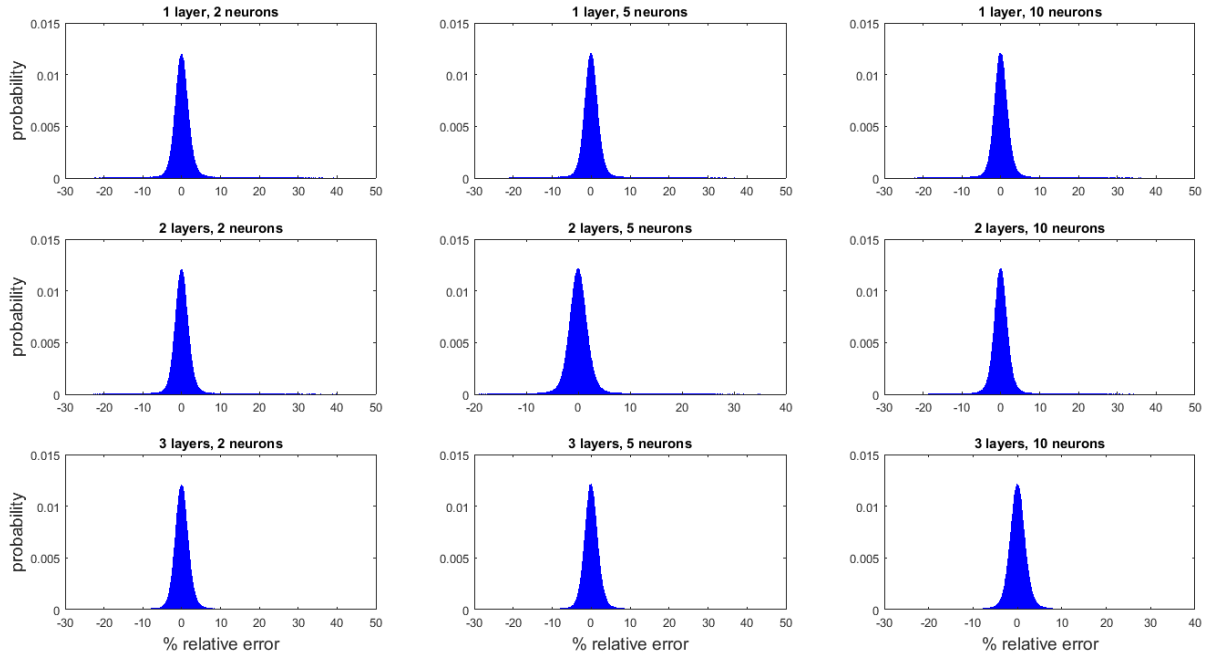

Figure 27: Probability distributions of the percent relative errors between 1,600,000 predicted and actual growth rate (Gr) responses

Based on the data presented in Table 155 and Figure 266, the ANN with 1 hidden layer and 5 neurons was chosen to predict roughness (R). It can be seen in Table 15 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 26 appeared narrow and sharp.

While no significant differences could be observed from Figure 27 among the error distributions from the ANNs that predicted the growth rate (Gr) responses, Table 16 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 28 and 29, respectively.

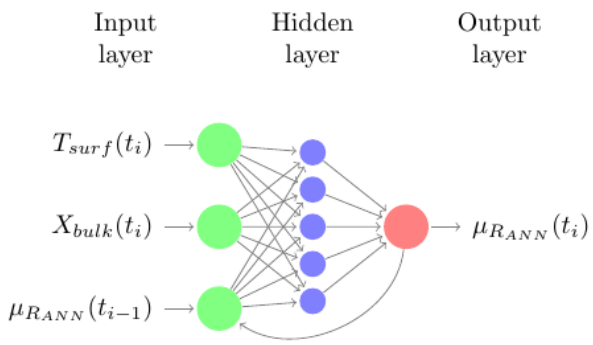

Figure 28: A schematic of the chosen "1 layer, 5 neurons" ANN structure for predicting roughness (μ_R). Note that the output of the ANN is used for autoregression via the closed-loop feedback.

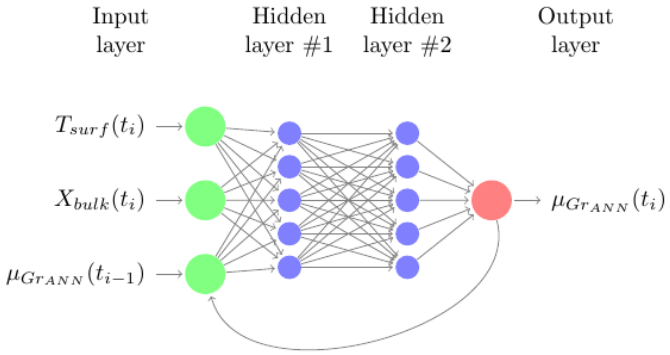

Figure 29: A schematic of the chosen "2 layers, 5 neurons" ANN structure for predicting mean growth rate (μ_{Gr}). Note that the output of the ANN is used for autoregression via the closed-loop feedback.

6.3: ANNs for NMPC of stochastic multiscale systems under nominal process conditions

In this section, we present the results of applying the ANNs chosen in section 6.2 for the shrinking horizon optimization problem of equation (81) in section 6.1 under nominal conditions (i.e. no uncertainty in E). 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 (81), 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 6.2, the temperature parameters were set as follows: T_{min} = 800 K, T_{max} = 1,200 K and ΔT = 50 K. The precursor mole fraction parameters were also identical to the characteristics of the training dataset: $X_{min} = 1\times10^{-6}$, $X_{max} = 4\times10^{-6}$ and $\Delta X = 5\times10^{-7}$. Note that the initial conditions for the temperature and precursor mole fraction were set to $T_0 = T_{min}$ and $X_0 = X_{max}$, respectively; this was done to maximize the value of W_a in equation (23), 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).

6.3.1 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 (81) 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 30a, 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 T_{surf} and X_{bulk} 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 17 and in Figures 31-33. Note that in Table 17, as well as in Tables 18-20 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.

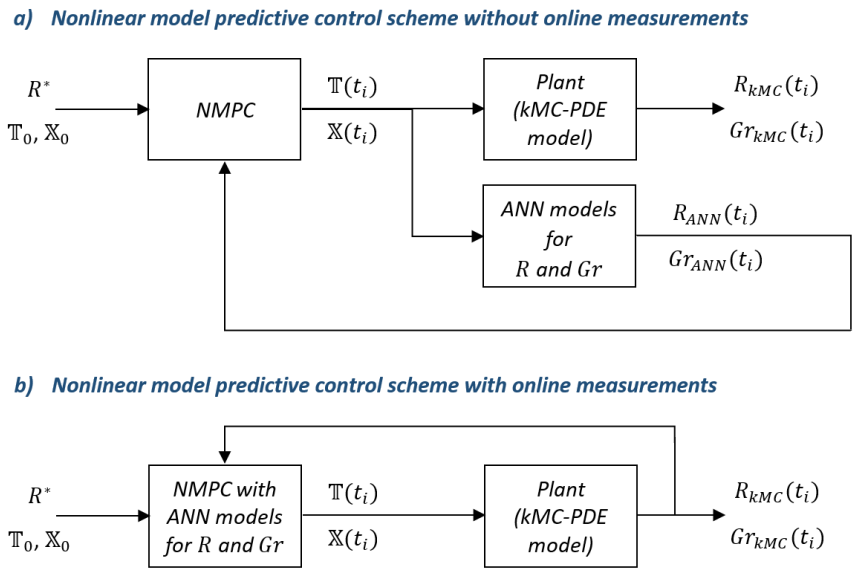

Figure 30: The open-loop (a) and closed-loop (b) shrinking horizon NMPC schemes employed in Scenarios A-D. Scenario A used the open-loop scheme (a), while Scenarios B-D used the closed-loop scheme (b).

Table 17: The performance of ANN models for the case of no online measurements and the sampling interval Δt = 25 s.

	$T_{surf}(t_n)$ (K)	$X_{bulk}(t_n)$	$\overline{Gr_{ANN}}$ (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

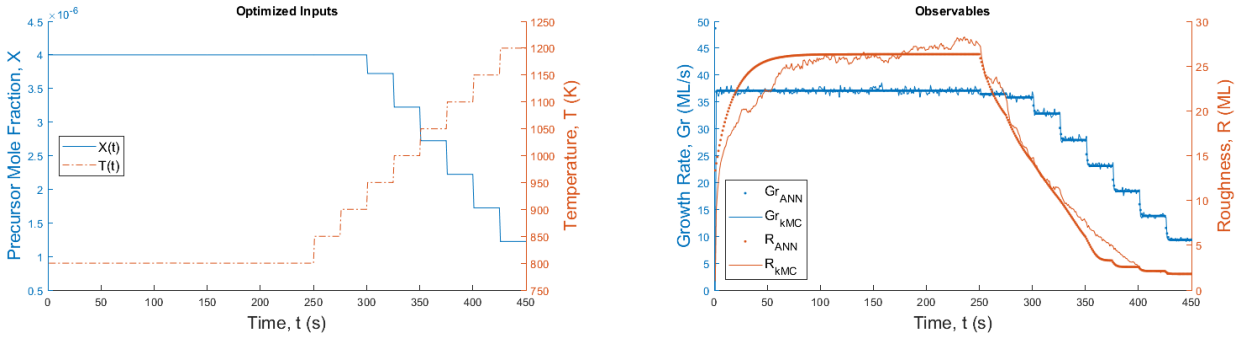

Figure 31: Optimal profiles of T_{surf} and X_{bulk} and the responses in R and Gr, no online measurements, Δt = 25 s, R^* = 1.8 ML

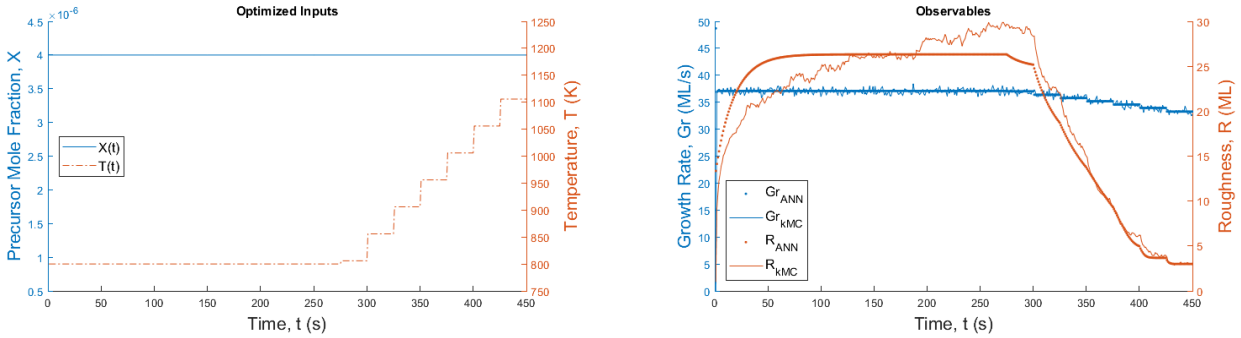

Figure 32: Optimal profiles of T_{surf} and X_{bulk} and the responses in R and Gr, no online measurements, Δt = 25 s, R^* = 3.0 ML

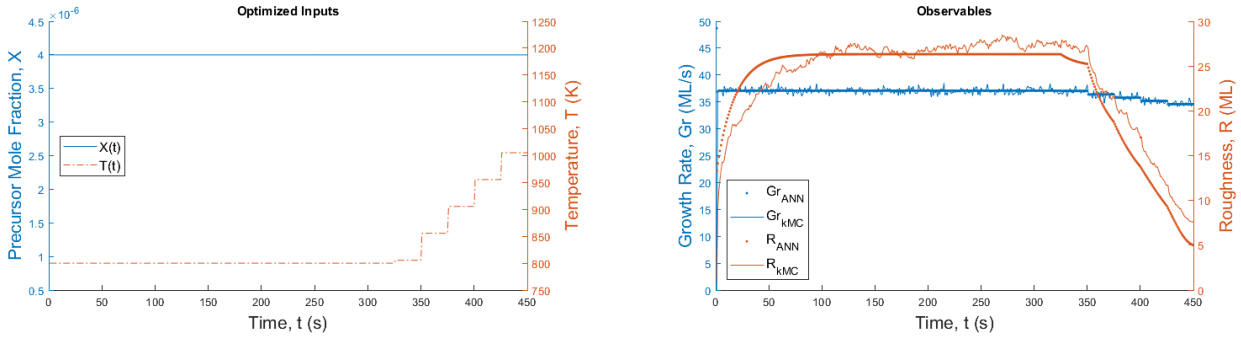

Figure 33: Optimal profiles of T_{surf} and X_{bulk} and the responses in R and Gr, no online measurements, Δt = 25 s, R^* = 5.0 ML

Figures 31-33 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 31 (R^* = 1.8 ML) and 32 (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 17). 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 33, where the ANN predictions of roughness reached steady state at 800 K before the multiscale model did (a similar effect is observed in Figures 31 and 32), 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[154] 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 17, compare 36.37 ML/s from the ANN to 36.24 ML/s from the multiscale model).

Note that in Table 17, 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 16).

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 31-33 to the corresponding time-dependent profiles of the observables. In Figures 32 and 33, only the substrate temperature T_{surf} 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 (X_{bulk}) and the observables: comparing Figures 31 and 32, it can be observed that the major difference is the manipulation of X_{bulk} in Figure 31 (R^* = 1.8 ML). In the scenario of Figure 31, both X_{bulk} and T_{surf} 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 31, which was 3.7 times lower than 33 ML/s in Figure 32). Comparing Figure 32 to Figure 31, 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.

6.3.2 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 30b and 34-36, as well as in Table 18) 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 30b, 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. Note that in practice, such measurements can be collected using laser scattering combined with adaptive optics, as well as X-ray reflectivity.[208,209] Figures 34-36 and Table 18 summarize the results of the study.

Table 18: The performance of ANN models for the case with online measurements and Δt = 25 s.

	$T_{surf}(t_n)$ *(K)*	$X_{bulk}(t_n)$	$\overline{Gr_{ANN}}$ *(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

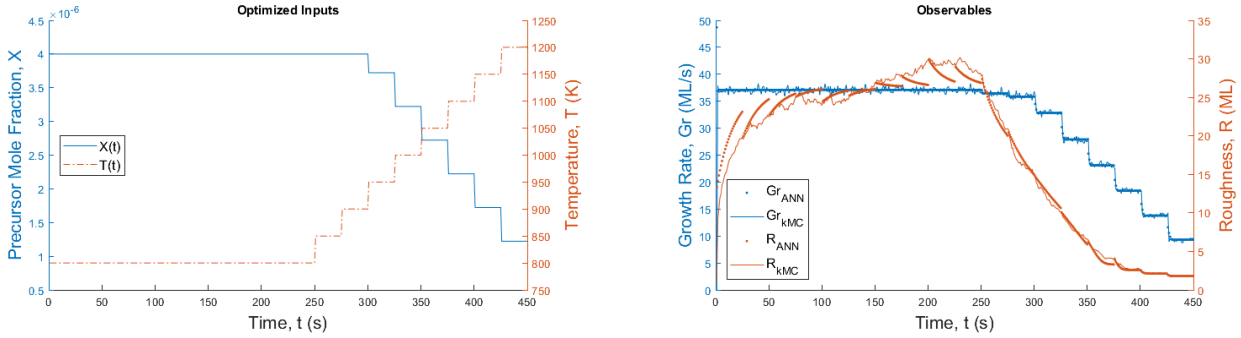

Figure 34: Optimal profiles of T_{surf} and X_{bulk} and the responses in R and Gr, with online measurements, Δt = 25 s, R^* = 1.8 ML

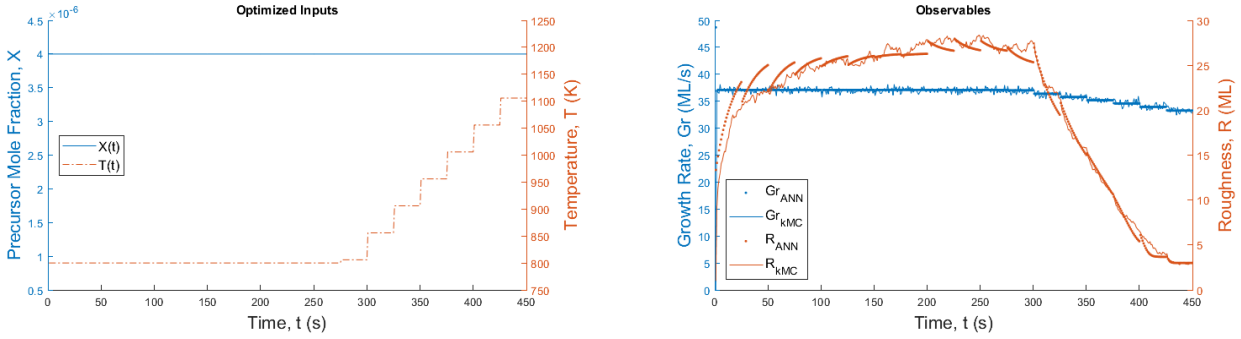

Figure 35: Optimal profiles of T_{surf} and X_{bulk} and the responses in R and Gr, with online measurements, Δt = 25 s, R^* = 3.0 ML

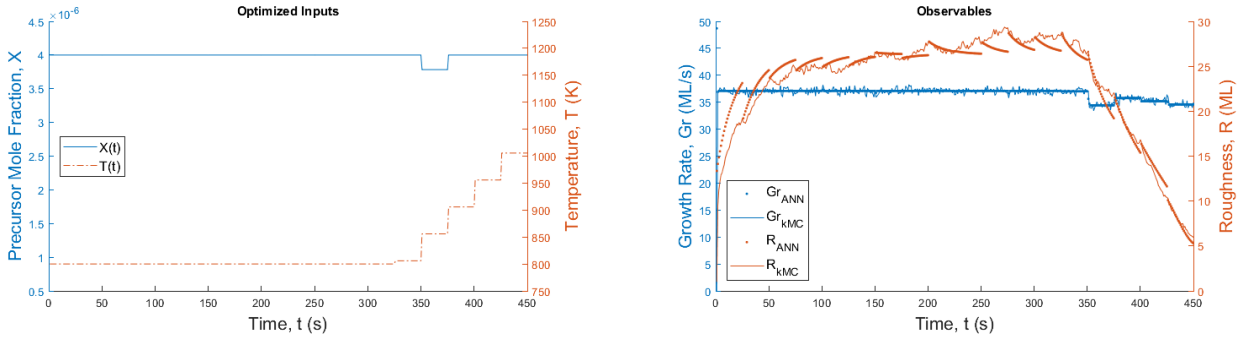

Figure 36: Optimal profiles of T_{surf} and X_{bulk} and the responses in R and Gr, with online measurements, Δt = 25 s, R^* = 5.0 ML

Unlike in Figures 31-33, 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 18 and Figure 34) and R^* = 3.0 ML

(3.00 ML vs 2.97 ML, respectively, see Table 18 and Figure 35). The growth rate predictions obtained using ANN and the stochastic multiscale model also exhibited good agreement. Furthermore, the final temperatures $T_{surf}(t_n)$ 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 36) 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 18. 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 36, 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. ΔT = 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® *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.

6.3.3 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

T_{surf} was 150 K and the change in X_{bulk} 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 (81). While accounting for the measured R and Gr through the initial conditions. The results have been summarized in Table 19 and Figures 37-39.

Table 19: The performance of ANN models for the case with online measurements, disturbance at 76 s and Δt = 25 s.

	$T_{surf}(t_n)$ (K)	$X_{bulk}(t_n)$	$\overline{Gr_{ANN}}$ (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

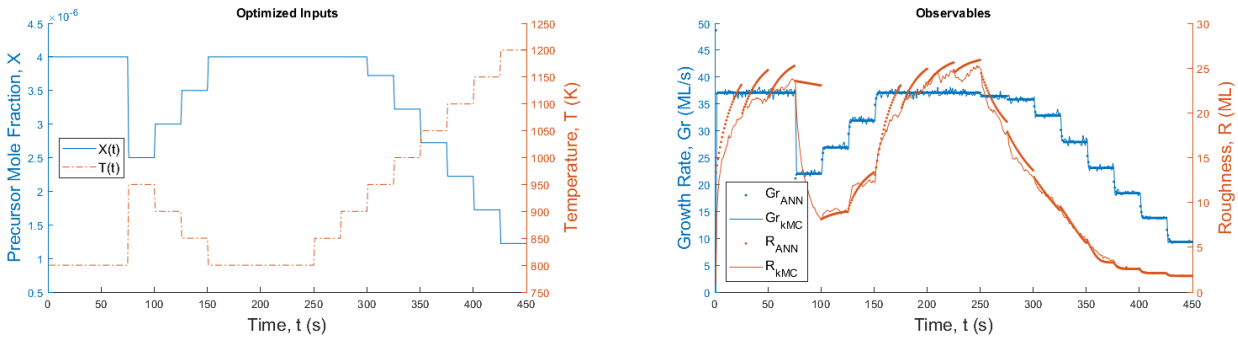

Figure 37: Optimal profiles and the responses, with online measurements and a disturbance, Δt = 25 s, R^* = 1.8 ML

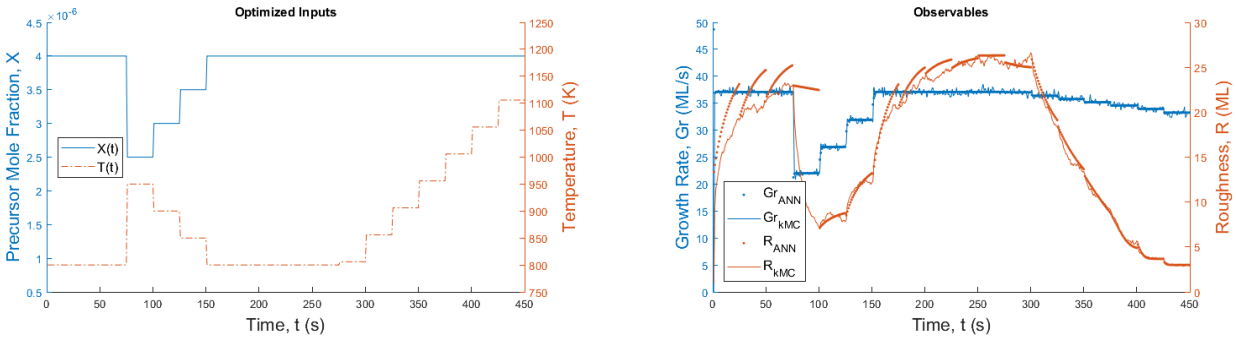

Figure 38: Optimal profiles and the responses, with online measurements and a disturbance, Δt = 25 s, R^* = 3.0 ML

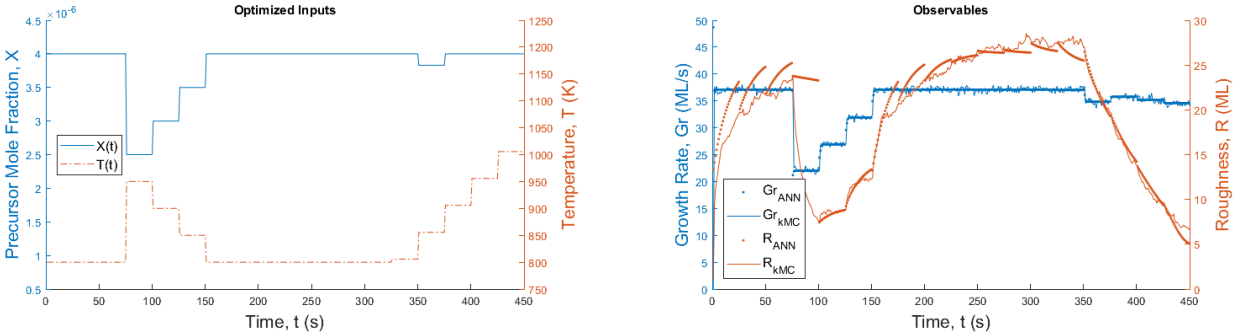

Figure 39: Optimal profiles and the responses, with online measurements and a disturbance, Δt = 25 s, R^* = 5.0 ML

It can be seen from the optimal profiles in Figures 37-39 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 39) was not met neither by the ANN model for roughness (5.06 ML was the value of $R_{ANN}(t_n)$, as per Table 19), nor by the plant (6.58 ML was the measured $R_{kMC}(t_n)$, per Table 19).

6.3.4 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 20 and Figure 40.

Table 20: 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.

	$T_{surf}(t_n)$ (K)	$X_{bulk}(t_n)$	$\overline{Gr_{ANN}}$ (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

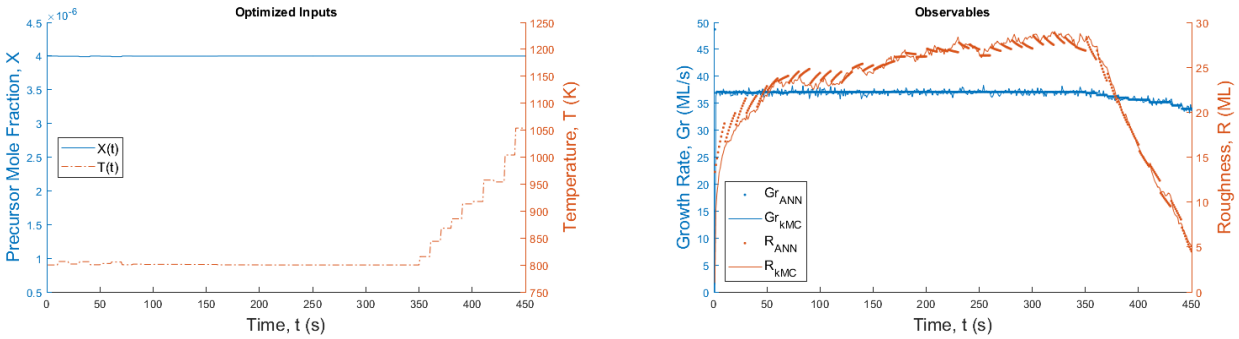

Figure 40: Optimal profiles and the responses, with online measurements, Δt = 10 s, R^* = 5.0 ML

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, $T_{surf}(t_n)$ was 1,054 K (as per Table 20), 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 20 and Figure 40), and $R_{ANN}(t_n)$ was even lower (4.53 ML, see Table 20). 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 6.3.1-6.3.3). 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.

6.4: ANNs for open-loop control NMPC of stochastic multiscale systems under uncertainty

The study in section 6.3 did not consider the effect of parametric uncertainty on the stochastic multiscale system. Hence, the work described in that section was expanded upon by considering the effect of parametric uncertainty on the roughness observable while the effect on growth rate was treated as negligible. Another ANN was trained to estimate the dynamic responses of the standard deviation of roughness and subsequently all three ANNs (the two from section 6.3 for the nominal values of the observables and the new ANN obtained in this section) were employed in a dynamic optimization scheme to identify the optimal profiles of the manipulated variables that would attain the desired thin film properties at the end of the batch. The resulting profiles were validated using the stochastic multiscale system.

6.4.1 Training of the ANN for standard deviation of roughness

To identify the most accurate ANN for modeling the standard deviation of roughness, i.e. σ_R, four ANN structures were considered, listed in Table 21 and Figure 41. Note that larger ANNs with more layers and neurons per layer were also proposed but failed to achieve a reasonable fit to the data and have therefore been omitted for brevity. Each ANN was a multiple-input single-output

autoregressive network trained in open-loop on the 10,000 s dataset comprised of T_{surf} and X_{bulk} manipulated variables and σ_R target training data (described above). In order to assess the accuracy of ANN predictions, the mean square error (MSE) was calculated between the ANNs' predictions for σ_R and the output of the kMC-PDE model based on the T_{surf} and X_{bulk} signals shown in Figure 41. The kMC-PDE model was simulated using the signals in Figure 41 for each of 500 realizations in E (none of which overlapped with the 300 realizations used to generate the 10,000 s dataset for training the σ_R ANNs). Thus, the raw dataset from the kMC-PDE model for selecting the σ_R ANN structure consisted of 187,500 points (500 realizations × 375 s), processed into 375 points by calculating the standard deviation of roughness at each second in the signal. Since the dataset for σ_R ANN structure selection was based on a greater number of realizations of E, it allowed to test the ANNs' abilities to generalise and predict σ_R behaviour based on not-seen-before data.

Table 21: The mean square errors in the predictions of the standard deviation of roughness generated by different ANN structures

ANN structure for predicting σ_R	MSE (ANN vs kMC-PDE model)
1 layer, 2 neurons	0.5166
1 layer, 5 neurons	0.4902
2 layers, 2 neurons each (2 × 2 in Figure 41)	0.6414
2 layers, 5 neurons each (5 × 5 in Figure 41)	0.0678

It can be seen from Table 21 that the ANN which produced the lowest MSE consists of 2 hidden layers with 5 neurons per layer. Note that its corresponding MSE was smaller than all others by one order of magnitude. This observation is qualitatively corroborated by Figure 41, where the output of the ANN with 2 layers and 5 neurons per layer (labelled as "ANN 5 × 5 neurons" in Figure 41), represented by the red dotted line, matched the kMC-based output (solid black line) most closely. Because of its superior performance, the ANN with 2 hidden layers and 5 neurons per layer was chosen to be incorporated into the NMPC framework for modelling σ_R in this work. The structure of this ANN is depicted in Figure 42.

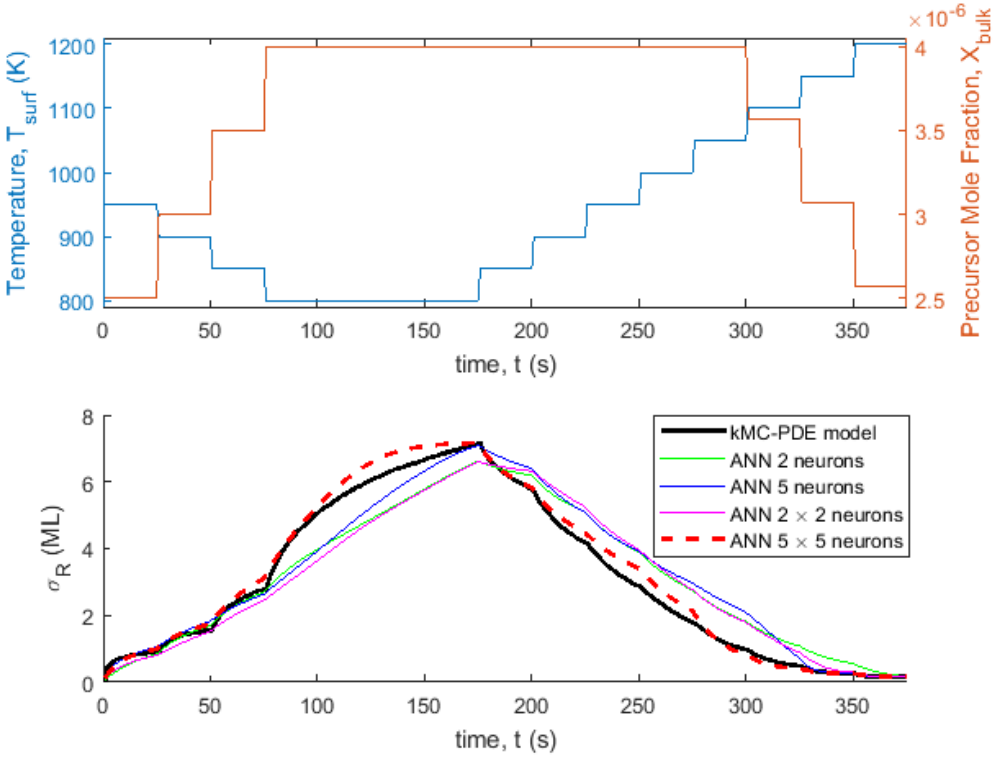

Figure 41: A comparison of kMC-PDE model output and different ANNs for the same temperature and precursor mole fraction profiles. Note that ANNs were specified by the number of neurons per hidden layer (e.g. "ANN 2 neurons" corresponds to 1 hidden layer with 2 neurons, "ANN 5 × 5 neurons" corresponds to 2 hidden layers with 5 neurons per layer, etc.).

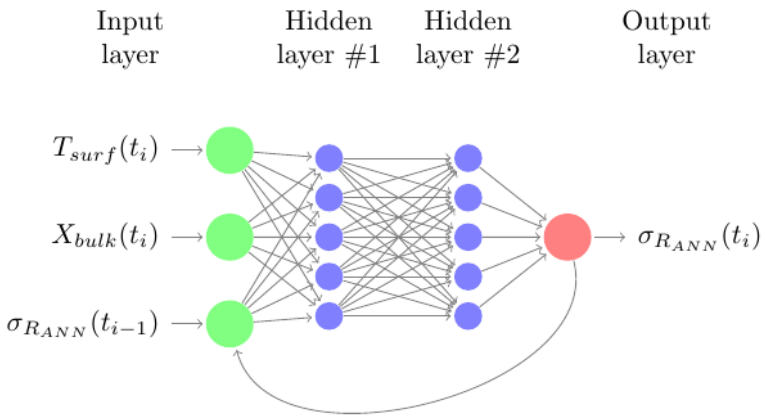

Figure 42: The chosen ANN structures for predicting the standard deviation of roughness. Note that the output of the ANN is used for autoregression via the closed-loop feedback.

6.4.2 Open-loop NMPC under uncertainty

In this section, we present the results of applying the ANNs chosen in sections 6.2 and 6.4.1 for the shrinking horizon optimization problem of equation (81). As mentioned in section 6.1, problem (81) is intractable if the full kMC-PDE model under certainty is used in the optimization scheme. However, the presence of the statistical moments μ_{Gr}, μ_R, and σ_R (mean growth rate, mean roughness and standard deviation of roughness, respectively) in the formulation allowed problem (81) to be reformulated using ANNs as follows:

$$\max_{T_{surf}(t_i),\, X_{bulk}(t_i)} \quad \frac{1}{n}\sum_{i=1}^{n} \mu_{Gr_{ANN}}\left(T_{surf}(t_i), X_{bulk}(t_i)\right)$$

Subject to:

$$\begin{aligned}
&\text{ANN models } \mu_{R_{ANN}},\ \sigma_{R_{ANN}} \text{ and } \mu_{Gr_{ANN}} \\
&t_i = i\Delta t; \quad \Delta t = 25\ s; \quad \forall i = 0,1,2,\dots n; \quad n = 18 \\
&\mu_{R_{ANN}}\left(T_{surf}(t_n), X_{bulk}(t_n)\right) + 3\sigma_{R_{ANN}}\left(T_{surf}(t_n), X_{bulk}(t_n)\right) \le R^* \\
&800\ \text{K} \le T_{surf}(t_i) \le 1200\ \text{K} \\
&2.5 \times 10^{-6} \le X_{bulk}(t_i) \le 4 \times 10^{-6} \\
&T_{surf}(0) = 800\ \text{K} \\
&X_{bulk}(0) = 4 \times 10^{-6} \\
&-50\ \text{K} \le T_{surf}(t_{i+1}) - T_{surf}(t_i) \le 50\ \text{K} \\
&-5 \times 10^{-7} \le X_{bulk}(t_{i+1}) - X_{bulk}(t_i) \le 5 \times 10^{-7}
\end{aligned} \tag{83}$$

Problem (83) was solved using the second order gradient-based active-set constrained optimization algorithm implemented in the *fmincon* function in MATLAB® with the same settings as described in section 6.3. However, the parameter λ in Problem (83) was set to 3 in order to approximate the probability limit on roughness at the end of the batch (i.e. $P(R(t_n) \le R^*)$) to 99.9 %. Also, to expedite the calculations, active set algorithm was used in *fmincon* instead of interior point. That probability limit is considered as an industrially relevant criterion for smooth high-performance thin films since it would capture most thin film roughness values.

The NMPC optimization framework is presented in Figure 43. For both scenarios A and B (sections 6.4.3 and 6.4.4, respectively), it has been assumed that the online measurements are not available from the plant, i.e. the present work only considers the optimal control of the thin film deposition process in open-loop. However, it has also been assumed that any disturbances $d(t_i)$ affecting the operation of the plant (such as in scenario B, see section 6.4.4) can be measured online during a sampling interval and compensated for by obtaining new optimal profiles of the manipulated variables T_{surf} and X_{bulk} using shrinking horizon NMPC in real time. This can be accomplished due to the computational efficiency of the ANNs for μ_R, σ_R and μ_{Gr}. Note that in Figure 43, $\mu_{R_{kMC}}(t_i)$, $\sigma_{R_{kMC}}(t_i)$ and $\mu_{Gr_{kMC}}(t_i)$ denote the values of the statistical moments of the observables produced by the plant (represented here as the kMC-PDE model shown in section 3.2) while $\mu_{R_{ANN}}(t_i)$, $\sigma_{R_{ANN}}(t_i)$ and $\mu_{Gr_{ANN}}(t_i)$ represent the predictions of the ANN models under uncertainty in E.

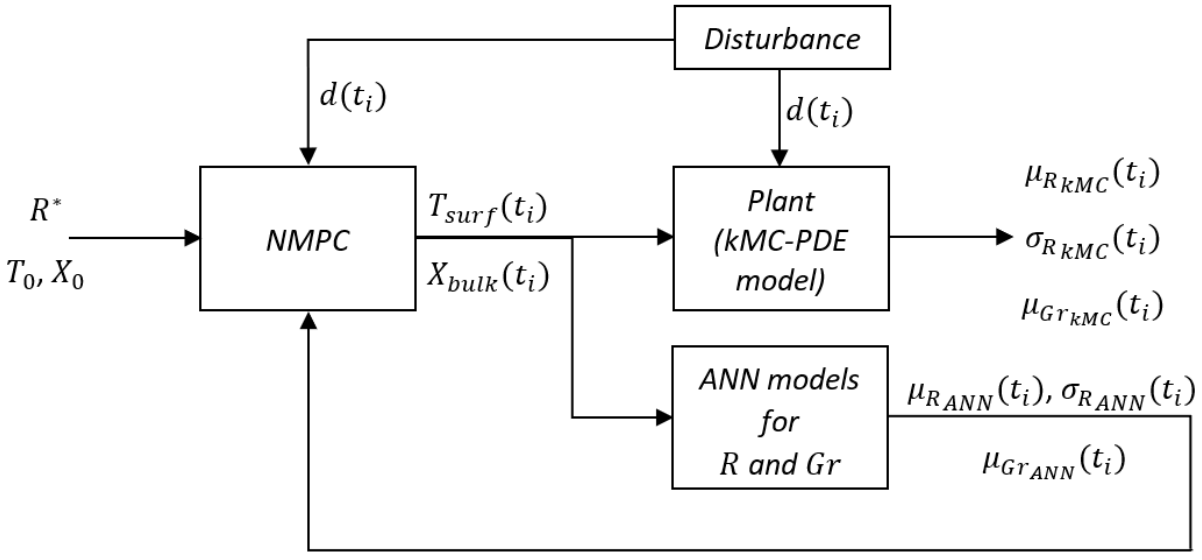

Figure 43: Open-loop ANN-based NMPC framework with measured disturbance $d(t_i)$.

6.4.3 Scenario A: dynamic optimization

In this scenario, the ANNs were used within the gradient-based constrained optimization algorithm presented in Problem (83) to find the optimal time-dependent control action profiles that would, according to the ANN predictions, satisfy the final roughness targets R^* of 3.0 ML and 2.52 ML. Once obtained, the profiles were used to run the plant until the end of the batch (450 s) in the presence of uncertainty in the parameter E (see equation (82)). This control scheme has been presented in Figure 43, but in this scenario there was no disturbance affecting the plant. To test the performance of the optimal control profiles under uncertainty, the plant was simulated with the NMPC-generated T_{surf} and X_{bulk} profiles using 500 different realizations of the uncertain parameter E that were not used for ANN training. That is, the profiles obtained from Problem (83) for this scenario were used to run the batch operation 500 times. The results are presented in Table 22 and in Figures 44 and 45. Note that in Table 22, $\overline{\mu_{Gr}}$ is the average nominal growth rate value that the optimization framework of Problem (81) sought to maximize. The mean square error (MSE) was used as metric to estimate process performance and has been calculated over the entire 450 s batch time. Table 22 compares the predictions made by the ANNs embedded in the NMPC framework to the output generated by the kMC-PDE model.

Table 22: Comparison of ANN-based predictions and plant outputs

	R^* = 3.00 ML		R^* = 2.52 ML	
	ANN	kMC	ANN	kMC
$\overline{\mu_{Gr}}$	35.91	35.80	34.53	34.42
μ_R(t = 450 s)	2.31	2.33	2.03	2.06
MSE μ_R	3.94		3.61	
σ_R(t = 450 s)	0.23	0.22	0.16	0.17
MSE σ_R	0.52		0.49	
μ_R(t = 450 s) + 3σ_R(t = 450 s)	3.00	2.99	2.52	2.57
CPU time (s)	2.28×10^{1}	3.06×10^{4}	1.68×10^{1}	3.04×10^{4}

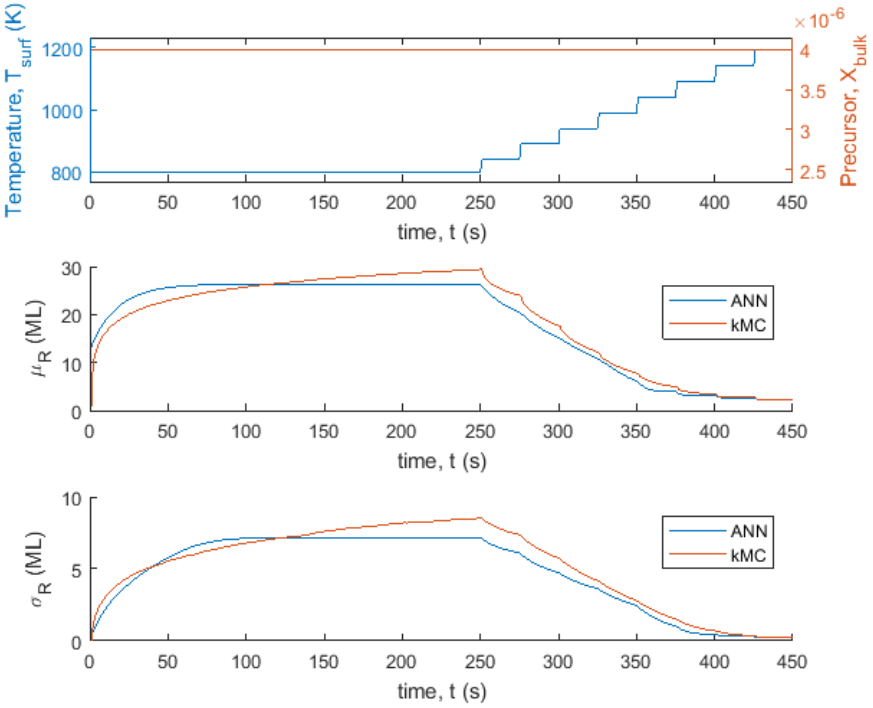

Figure 44: Time-dependent profiles of the manipulated variables and the statistical moments of the roughness observable for R = 3.00 ML.*

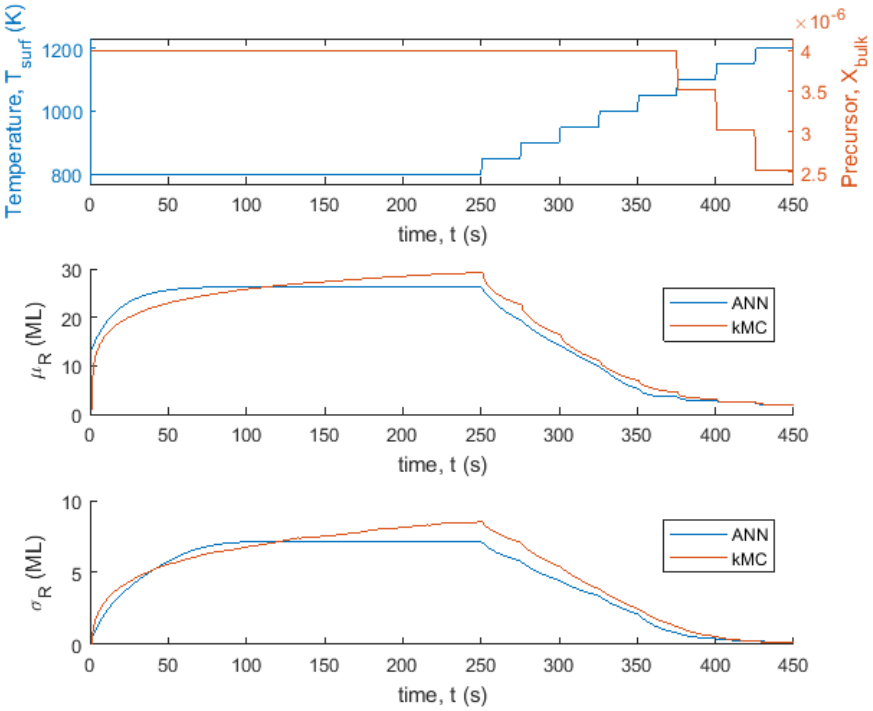

Figure 45: Time-dependent profiles of the manipulated variables and the statistical moments of the roughness observable for R = 2.52 ML.*

As shown in Table 22 and Figures 44 and 45, the ANN-based predictions agree closely with $\overline{\mu_{Gr}}$, μ_R and σ_R for the overall plant response obtained from the 500 batch runs. For example, the average nominal growth rate values $\overline{\mu_{Gr}}$ predicted by ANNs were only 0.3% greater than their corresponding kMC values (see Table 22). The discrepancies between ANN and kMC-based values of μ_R and σ_R at t = 450 s (i.e. final batch time) became slightly larger for R^* = 2.52 ML than for R^* = 3.00 ML, but even for R^* = 2.52 ML the differences were not larger than 6% (e.g. see the predictions for σ_R at t = 450 s in Table 22). However, the MSE (mean square error) values were smaller for R^* = 2.52 ML than for 3.00 ML, e.g. in the case of μ_R the MSE over the 450 s batch time was 3.94 for R^* = 3.00 ML, as opposed to 3.61 for R^* = 2.52 ML, a 9.1 % decrease.

By comparing the Figures 44 and 45, it can be seen that the profiles of T_{surf} and X_{bulk} changed according to the R^* value. For R^* = 3.00 ML (Figure 44), X_{bulk} remained at its maximum value during the entire batch and the final value of T_{surf} was 1190 K. For R^* = 2.52 ML (Figure 45), X_{bulk} was decreased almost to its minimum allowed value (2.52×10^{-6}) during the last three sampling intervals (376 s – 450 s) and the final T_{surf} was 1200 K (higher than for R^* = 3.00 ML). Therefore, the ANN-based NMPC scheme was able to correctly adjust the manipulated variables based on the specified R^* value and drive the plant to the optimal desired target.

For both R^* targets, the ANNs required 3 orders of magnitude less CPU time than the corresponding kMC-PDE implementation of the optimal profiles under uncertainty (e.g. for R^* = 3.00 ML, 22.8 s was needed for ANNs to identify the optimal profiles for $T_{surf}(t_i)$ and $X_{bulk}(t_i)$, but 3.06×10^4 s was needed to implement them on the kMC-PDE model under uncertainty, see Table 22). Also, for both R^* values, the CPU time for optimization was shorter than the duration of the sampling interval (25 s). Since the ANNs were able to efficiently provide accurate estimates,

they are a promising approach to perform optimal control of stochastic multiscale systems under uncertainty.

6.4.4 Scenario B: rejection of measured disturbance

To further test the performance of the optimal control profiles obtained from the ANNs developed in this framework, we present a variation of scenario A (R^* = 2.52 ML), where we studied the effect of large measured disturbances ($d(t_i)$ in Figure 5) in both manipulated variables. Using the same optimization procedure as in section 5.1, we added a disturbance that was not considered during ANN training. At 76 s (sampling interval i = 4), disturbances to both the temperature and the precursor mole fraction were introduced. The disturbances were 3 times greater than the maximum allowed amount during data generation and training, i.e. the change in T_{surf} was 150 K and the change in X_{bulk} was -1.5×10^{-6}. The disturbance in X_{bulk} was held constant for one sampling interval (until 100 s), but the disturbance in T_{surf} was held constant for a larger period of time, i.e. five sampling intervals (until 200 s). Since it was assumed that the NMPC framework can detect measured disturbances (as per Figure 43), shrinking horizon optimization was conducted once at the beginning of the batch horizon and every sampling interval whenever a disturbance in either of the two manipulated variables was detected by NMPC. The results have been summarized in Table 23 and Figures 8 and 9. Note that the definitions of quantities listed in Table 3 are the same as for Table 2. As in scenario A, 500 batch runs were performed using the optimal control profiles obtained from this scenario; each batch run corresponded to a particular realization in the uncertain parameter E.

Table 23: Comparison of ANN-based predictions and the results from the full kMC-PDE model for the case with disturbances in T_{surf} and X_{bulk}

	R^* = 2.52 ML	
	ANN	kMC
$\overline{\mu_{Gr}}$	32.42	32.35
μ_R(t = 450 s)	2.03	2.06
MSE μ_R	36.7	
σ_R(t = 450 s)	0.16	0.17
MSE σ_R	0.97	
μ_R(t = 450 s) + 3σ_R(t = 450 s)	2.52	2.57
CPU time (s)	9.83×10^{1}	3.08×10^{4}

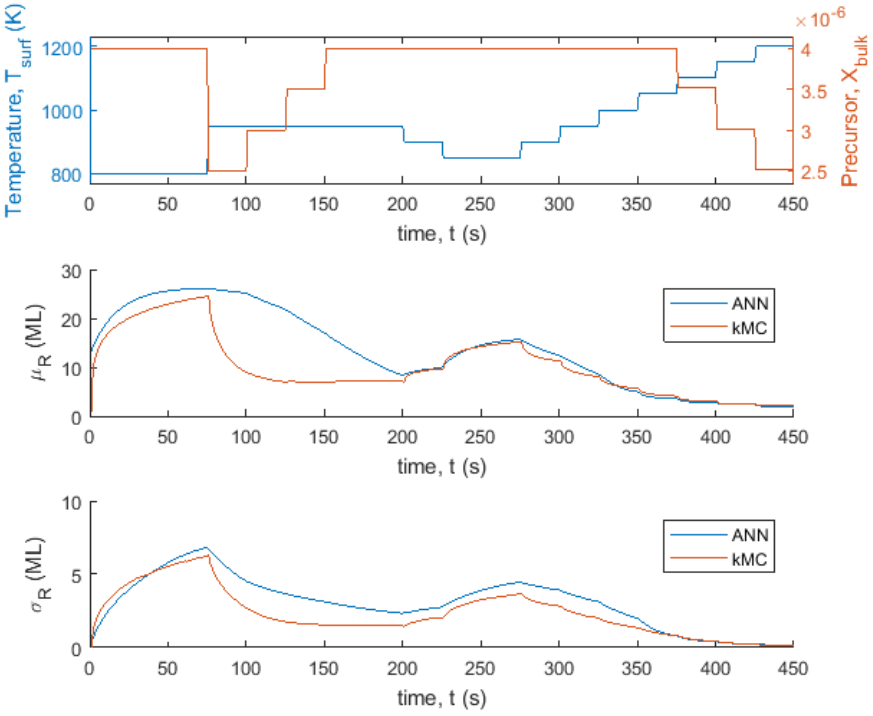

Figure 46: Time-dependent profiles of the manipulated variables (with disturbances) and the statistical moments of the roughness observable for R = 2.52 ML.*

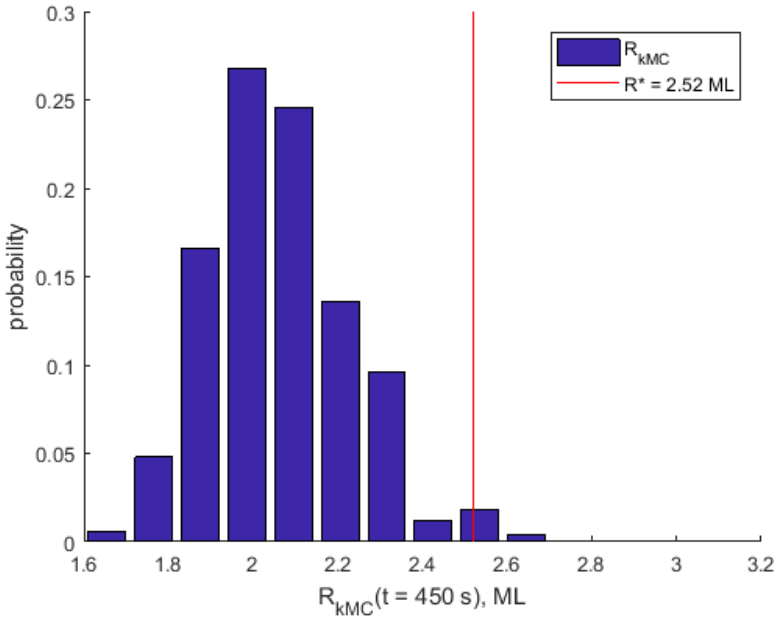

Figure 47: The distribution of roughness values at the end of the batch produced after validating the T_{surf} and X_{bulk} profiles with disturbances using the full kMC-PDE model (R^* = 2.52 ML).

As shown in Table 3 and Figure 8, the measured disturbances to the manipulated variables caused the predictions of ANNs to deviate from the results of kMC. For example, the MSE values in Table 3 were substantially greater than those reported in Table 2. For μ_R, the MSE in Table 3 was an order of magnitude greater than the values reported in Table 2 (R^* = 2.52 ML). Figure 8 illustrates the significant increase in MSE since the ANN for μ_R was not able to accurately capture the dynamic response of the plant to the disturbance between t = 76 s and t = 200 s (i.e. the time interval at which the disturbance entered the plant). For σ_R, the MSE in Table 3 was approximately 2 times greater than the corresponding values in Table 2 (R^* = 2.52 ML), and a deviation due to the disturbances is also observed from Figure 8 (i.e. within the 76 s – 200 s interval). Nevertheless, the ANN-based NMPC was able to reject the disturbances and drive the system to the specified R^* target while also attempting to maximize the average nominal growth rate $\overline{\mu_{Gr}}$ by temporarily lowering T_{surf} and raising X_{bulk} once the disturbances stopped acting on the plant. Furthermore, the total CPU times required by shrinking horizon optimization and the implementation of the

optimal profiles on the plant under uncertainty indicate the efficiency of the ANN models. As shown in Table 3, the total CPU time for ANN-based optimization was still 3 orders of magnitude lower than the CPU time for implementing the optimal profiles on the plant.

At t = 450 s, 98.4% of the roughness values generated through implementing the optimal profiles using kMC-PDE model under uncertainty were less than R^* = 2.52 ML, as shown in Figure 9. This illustrates that the ANN-based approach used in Problem (20) for approximating the probability limit $P(R(t_n) \leq R^*)$ produces adequate results, since the distribution in Figure 9 is nonlinear. In the cases where this approach would not be adequate, more data from a first-principles model (or actual plant data) would be required to have an accurate representation of the shape of the observables in the ANN training data. Additional ANNs could then be trained to capture higher-order moments if needed (e.g. skewness, kurtosis, different modes for multi-modal distributions, etc.). Such moments can be obtained from various density estimation methods, e.g. Bayesian approach, Metropolis-Hastings algorithm, density-based statistical data models, particle filtering, and other methods.[210–212]

6.5: ANN model discrimination via parameter estimation for closed-loop NMPC of stochastic multiscale systems under uncertainty

While the approach described in section 6.4 considered parametric uncertainty, it could not accommodate for any online measurements that could be collected from the plant. In order to enable the closed-loop NMPC of the thin film deposition system, an ANN was trained on each of the 300 signals of the second dataset described in section 6.2 to predict R using the ANN for nominal roughness identified on the 400,000-point dataset (described in section 6.2) as the initial condition for each new ANN. Consequently, the 300 ANNs trained to predict R under different realizations of E all used the structure shown in Figure 24. The same training methodology was

employed for developing these ANNs as the one discussed above regarding ANN training on the 400,000-point dataset. Once trained, the set of 300 ANNs (which hereafter shall be referred to as set *J*) was used in the estimation of the true value of E in the thin film production process.

6.5.1 Parameter Estimation

The estimation of the uncertain parameter realization was accomplished by comparing the measurements of roughness obtained from the process, i.e. R_{kMC}, to the predictions of roughness generated by the set *J* comprised of 300 ANNs, i.e. R_{ANN_J}. The same signal in the input variables T_{surf} and X_{bulk} was provided to the process and to the ANNs, and the results were compared according to the criteria discussed further in this work.

In order to extract the information from the process that would allow for an accurate estimation of the realization in E, and, consequently, accurate ANN-based predictions of the observables, the following considerations were taken into account during input signal design. The signal needs to be time-varying and with sufficient frequency content, i.e. slow and fast changes in the input (manipulated) variables. This is required to capture the dynamic responses of the observables under different combinations in the manipulated variables. Furthermore, the signal should include positive and negative changes in the input variables so that R_{kMC} would contain information about the roughness observable's responses to both types of changes in the inputs. In addition, to mimic the industrial setting, the magnitudes of the manipulated variables in the signal were constrained to the range that promoted thin film growth, i.e. relatively low T_{surf} and relatively high X_{bulk} values. Furthermore, the length of the input signals employed in parameter estimation needs to be relatively short because the industrial process needed to spend the maximum possible amount of time manufacturing the product. Based on these considerations, several input signals for T_{surf} and

X_{bulk} were designed and studied in this work, as shown in the Results and Discussion section below.

Two standard statistical modeling techniques were used for estimating E: maximum likelihood estimation (MLE) and mean square error (MSE). MLE is used to estimate parameter values that maximize the probability of obtaining the observed data from an assumed statistical model [213–215]. On the other hand, MSE is a traditional method in regression analysis for assessing the goodness-of-fit of models proposed to explain the observed data [216]. Therefore, these two criteria were used to assess the magnitude of the differences between the measured R_{kMC} and the predictions generated by the R_{ANN_J} set of neural networks, where the same input signal was provided to the process and to the J set of ANNs. The ANN that produced the smallest deviations from R_{kMC} according to the MLE and MSE criteria was chosen as the data-driven model for predicting the correct input values for satisfying the thin film product requirements. The corresponding realization of E used in ANN training was considered to be an estimate of the true uncertain parameter value E in the thin film production process and can be used to control the properties of the thin film for the rest of the batch.

The Maximum Likelihood Estimation (MLE) criterion considered the conditional probability of obtaining ANN-based predictions of the roughness observable given the observed measurements from the process represented by the kMC-PDE model:

$$\max_{J} \; p_{R_{ANN}|R_{kMC}}\left(R_{ANN_J}(T_{surf}(t_i), X_{bulk}(t_i))\,\middle|\,R_{kMC}\left(T_{surf}(t_i), X_{bulk}(t_i)\right)\right) \tag{84}$$

The solution to this problem yields j^*, i.e. the index of the best-fit ANN (ranging from 1 to 300 for the set J) and the conditional probability $p_{R_{ANN}|R_{kMC}}$ is given by:

$$p_{R_{ANN}|R_{kMC}} = \prod_{i=1}^{K} p_i \left(R_{ANN_J} \left(T_{surf}(t_i), X_{bulk}(t_i) \right) \right) \tag{85}$$

where K is the number of measurements collected during the identification from the process and p_i is the probability of obtaining $R_{ANN_J}\left(T_{surf}(t_i), X_{bulk}(t_i)\right)$, i.e. the roughness value predicted by each ANN in the set J, from the normal probability density function with its mean located at the roughness measurement $R_{kMC}\left(T_{surf}(t_i), X_{bulk}(t_i)\right)$, denoted as $R_{kMC,i}$ for short, and unit standard deviation, i.e. $\mathcal{N}\left(R_{kMC,i}, 1\right)$. Note that the normal distribution shape is assumed to adequately approximate the true nonlinear distribution of the roughness measurements that occurs due to the stochastic noise present in the multiscale process.

In addition to MLE, the Mean Square Error (MSE) criterion has also been considered for estimating the true E realization and identifying the corresponding ANN that is most likely to predict the correct responses in roughness:

$$\min_J \frac{1}{K} \sum_{i=1}^{K} \left(R_{ANN_J} \left(T_{surf}(t_i), X_{bulk}(t_i) \right) - R_{kMC} \left(T_{surf}(t_i), X_{bulk}(t_i) \right) \right)^2 \tag{86}$$

which also produces j^*, i.e. the index of the best-fit ANN and of the corresponding estimate of the true E value.

6.5.2 ANN discrimination

In order to perform parameter estimation and ANN discrimination, several input signals (depicted in Figure 48) were designed according to the considerations mentioned in the previous section. For all signals, the number of measurements, i.e. K in equations (21) and (22), was set to 5. Consequently, the signal length was 125 s, which was sufficient to implement several changes in the input (manipulated) variables T_{surf} and X_{bulk} and observe the appropriate responses in the

roughness observable. The R_{kMC} values were collected at the end of every sampling interval Δt, i.e. every 25 s. In order to promote thin film growth while observing process responses to the provided signals, the range of T_{surf} values has been limited to [800 K, 900 K] and the range of X_{bulk} values was limited to [3×10^{-6}, 4×10^{-6}].

Figure 48 illustrates that the MLE and MSE methods produce equivalent estimates of the uncertain parameter regardless of the input signal. For each of the input signals 1-4 shown in Figure 48, the thin film multiscale model was simulated using 500 realizations in E that did not share any datapoints with the 300 E realizations used to train R_{ANN_J}. Using the MLE and MSE techniques, for each of the 500 realizations (i.e. the true values of E) an estimate was identified among the 300 E realizations that corresponded to R_{ANN_J}. The percent relative errors between the MLE/MSE estimates and the true realizations of E have been calculated for each input signal and their probability distributions have been presented in Figure 48. Despite the different shapes of input signals 1-4, the percent relative errors in the estimates of E were between -5% and 2% in all cases without any significant differences in the shapes of the probability distributions. Furthermore, the estimates of E generated by MSE and MLE were similar, as can be seen from their respective probability distributions in Figure 48 (represented by blue and yellow bars, respectively).

The mean percent relative error in the estimation of E was approximately -2% for all signals (as per the probability distributions in Figure 48), which did not adversely affect the ANN-based predictions of the observables since Gr_{kMC} was insensitive to parametric uncertainty while the stochastic noise present in R_{kMC} made the effect of the -2% relative error on ANN selection and R_{kMC} prediction accuracy negligible. In addition, a deviation is expected since the set ANN models were developed using realizations in E that are likely to be different from the true realization in E.

Since the shapes of the signals and the choice between MLE and MSE did not affect the quality of E estimation, signal 1 and the MLE method were employed in the scenarios considered in the next section. Note that other input signals were tested presenting the same results, but these are not shown here for brevity.

Figure 49 further illustrates the validity of the assumption used in our application of the MLE method regarding the ability of the normal distribution to adequately approximate the actual shape of the distribution of roughness measurements at different time intervals. The thin film deposition model was run under three different realizations in E (at the nominal value and at one standard deviation away from the mean) using input signal 1 (shown in Figure 48). At each realization of E, the model was simulated 300 times to assess the noise in the R_{kMC} measurements. This noise seems to follow a Gaussian distribution under various E realizations and different measurement timepoints, as per Figure 49, thereby corroborating the validity of the MLE approach to ANN discrimination.

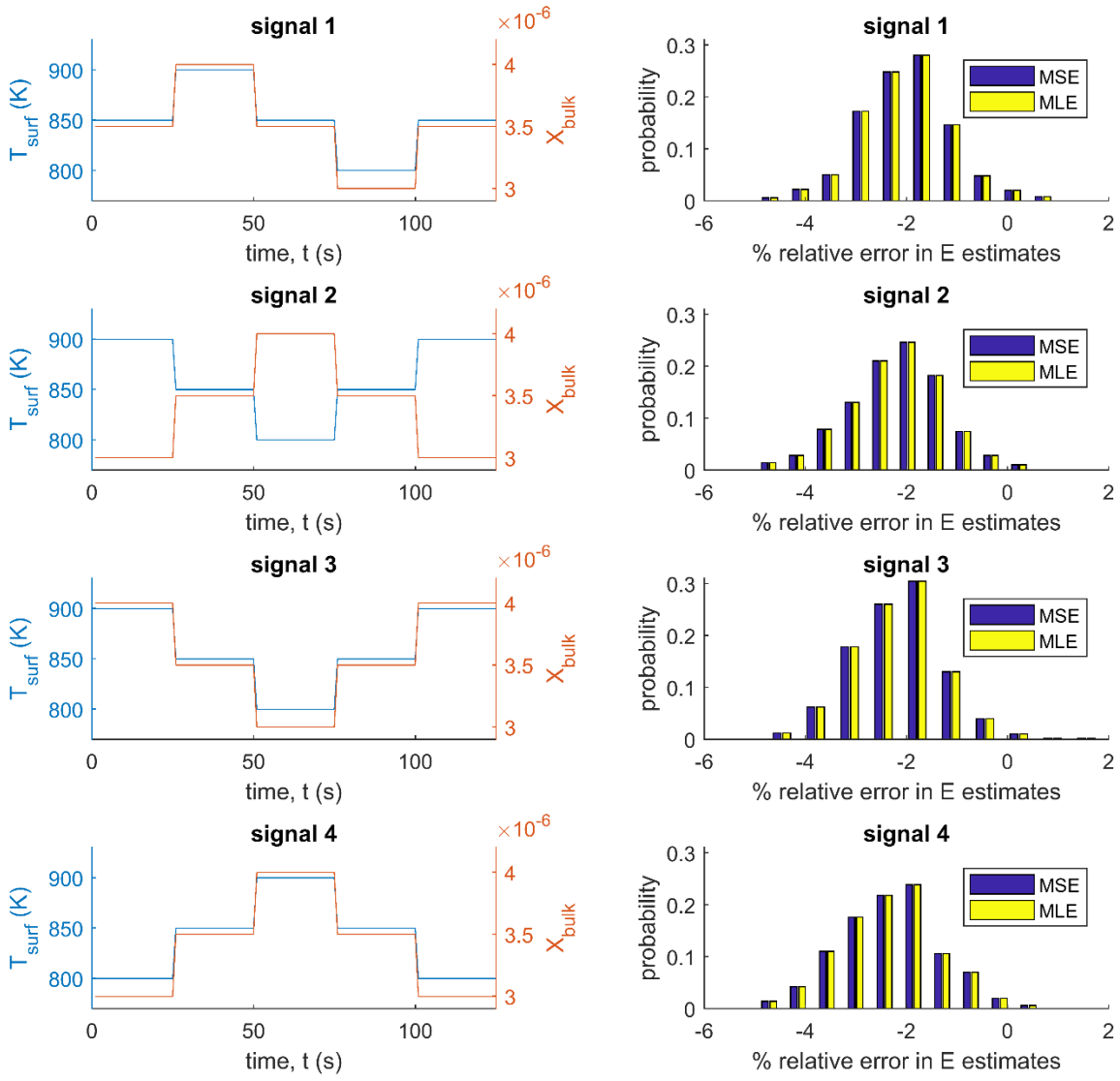

Figure 48: Different shapes of the input signals do not appear to significantly impact the accuracy of uncertain parameter estimates. Also, ANN discrimination by the MSE and MLE methods produce equivalent results.

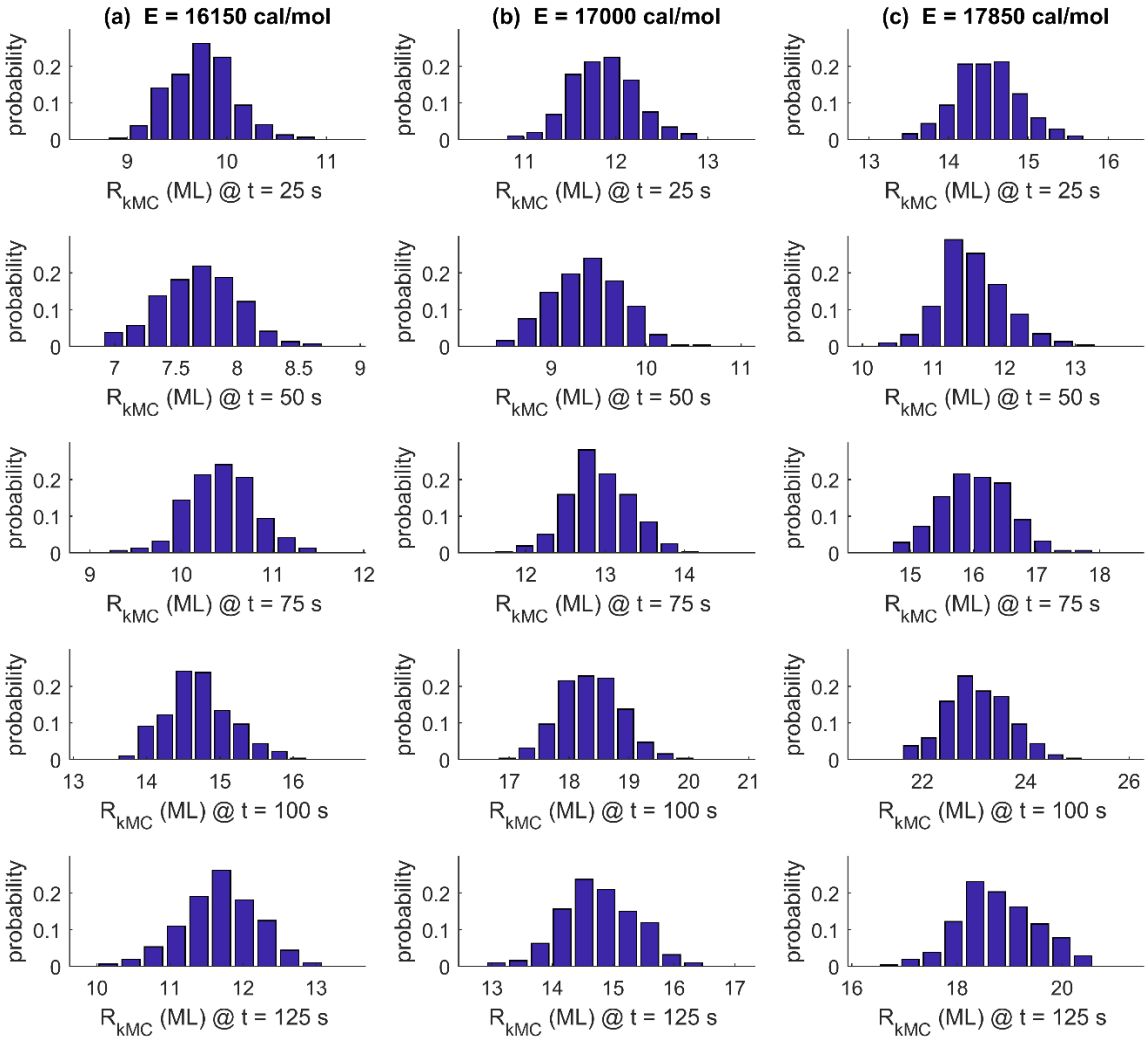

Figure 49: The measurements of roughness, reported in monolayers (ML), approximate to be normally distributed at different realizations of uncertainty and at different sampling times.

6.5.3 Closed-loop NMPC under uncertainty

In this section, we present a framework to manufacture thin films that user-defined properties under uncertainty in the production process. As presented above, the parameter estimation method can be used to identify an ANN model that can predict with sufficient accuracy the response in the observables due to changes in the input signals (i.e. T_{surf} and X_{bulk}). The resulting ANN model can be embedded within a nonlinear shrinking horizon optimization problem like that shown in

equation (81) to seek for the optimal control actions in the manipulated variables that can produce a thin film with specific (desirable) properties at the end of the batch. As already mentioned, problem (81) is intractable if the full kMC-PDE model under certainty is used in the gradient-based optimization scheme because of its associated computational cost and the inherent stochastic noise in its observables. However, the reliance on the dynamic behavior of the observables R and Gr in the formulation allowed problem (81) to be reformulated using ANNs as follows:

$$\max_{T_{surf}(t_i),\, X_{bulk}(t_i)} \quad \frac{1}{n}\sum_{i=1}^{n} Gr_{ANN}\left(T_{surf}(t_i), X_{bulk}(t_i)\right)$$

Subject to:

$$\begin{aligned}
&\text{ANN model for roughness chosen via MLE, } R_{ANN_{j^*}}\\
&\text{ANN model for the nominal growth rate, } Gr_{ANN}\\
&t_i = i\Delta t; \quad \Delta t = 25\ s; \quad \forall i = 0,1,2,\dots n; \quad n = 18\\
&R_{ANN}\left(T_{surf}(t_n), X_{bulk}(t_n)\right) \le R^*\\
&800\text{ K} \le T_{surf}(t_i) \le 1200\text{ K}\\
&1 \times 10^{-6} \le X_{bulk}(t_i) \le 4 \times 10^{-6}\\
&T_{surf}(0) = 800\text{ K}\\
&X_{bulk}(0) = 4 \times 10^{-6}\\
&-50\text{ K} \le T_{surf}(t_{i+1}) - T_{surf}(t_i) \le 50\text{ K}\\
&-5 \times 10^{-7} \le X_{bulk}(t_{i+1}) - X_{bulk}(t_i) \le 5 \times 10^{-7}
\end{aligned} \tag{87}$$

Problem (87) was solved using the same methods as described in section 6.4.2 with the active set algorithm used in the *fmincon* function. However, in the scenarios presented below the final roughness target (R^*) was set to 2.18 ML as an industrially relevant criterion for smooth high-performance thin films since the ideal thin film roughness is 1 ML.[2]

The NMPC optimization framework is presented in Figure 50. It has been assumed that the online measurements of R and Gr are available from the process. It has been assumed that any disturbances $d(t_i)$ affecting the operation of the thin film deposition process can be detected online during a sampling interval through the measurements and compensated for by obtaining new

optimal profiles of the manipulated variables T_{surf} and X_{bulk} using the shrinking horizon NMPC framework in real time. This can be accomplished due to the computational efficiency of the ANNs for R_{ANN} and Gr_{ANN}. Note that in Figure 50, $R_{kMC}(t_i)$ and $Gr_{kMC}(t_i)$ denote the values of the of the observables produced by the process (represented here as the kMC-PDE model shown earlier) while $R_{ANN}(t_i)$ and $Gr_{ANN}(t_i)$ represent the predictions of the ANN models under time-invariant uncertainty in E.

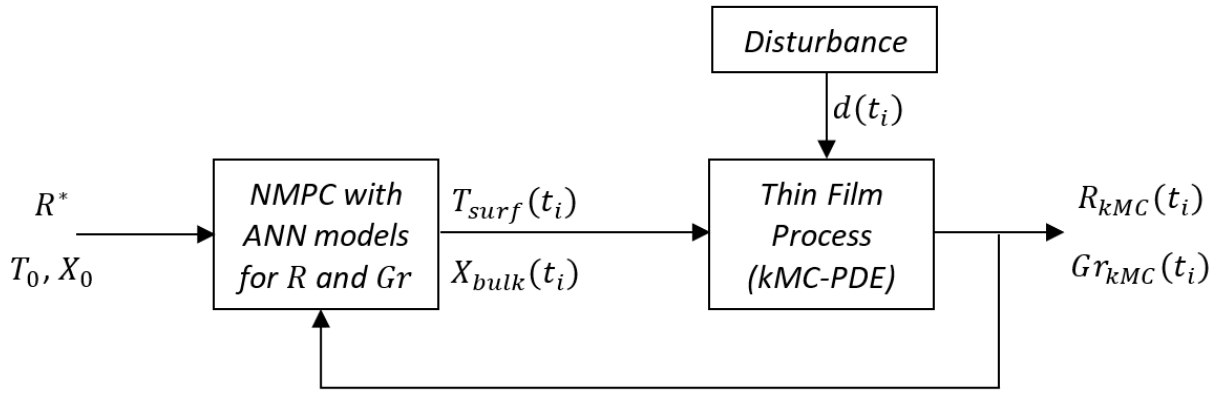

Figure 50: Closed-loop ANN-based NMPC framework with disturbance $d(t_i)$ acting on the process.

6.5.4 Scenario A: Online closed-loop optimization

In this scenario, the ANNs were used within the gradient-based constrained optimization algorithm presented in problem (87) to find the optimal time-dependent control action profiles that would, according to the ANN predictions, cause the produced thin film to satisfy the final roughness target R^* of 2.18 ML. Three different realizations of the uncertain parameter E were used in the thin film production process: 16,151, 16,988 and 17,852 cal/mol. Note that these realizations are approximately equal to the nominal E from equation (82) and to the E values one standard deviation away from the nominal value. Thus, the ability of the ANN discrimination method to

identify the best-fit ANN via equation (84) was tested under the most likely realizations of E as per equation (82).

Since different realizations of E were employed, ANN discrimination had to be performed for each realization. Once the appropriate ANN was identified, closed-loop shrinking horizon NMPC shown in (23) was solved to determine the control actions that enable thin films to satisfy R^* at the final batch time of 450 s. This workflow has been schematically shown in Figure 51 and the results are summarized in Table 24 and Figure 52. Note that in this scenario it was assumed that the disturbance shown in Figure 43 is not acting on the process. Also, note that in Table 24, $\overline{Gr}$ is the average growth rate value achieved by the process that the optimization framework of Problem (87) sought to maximize.

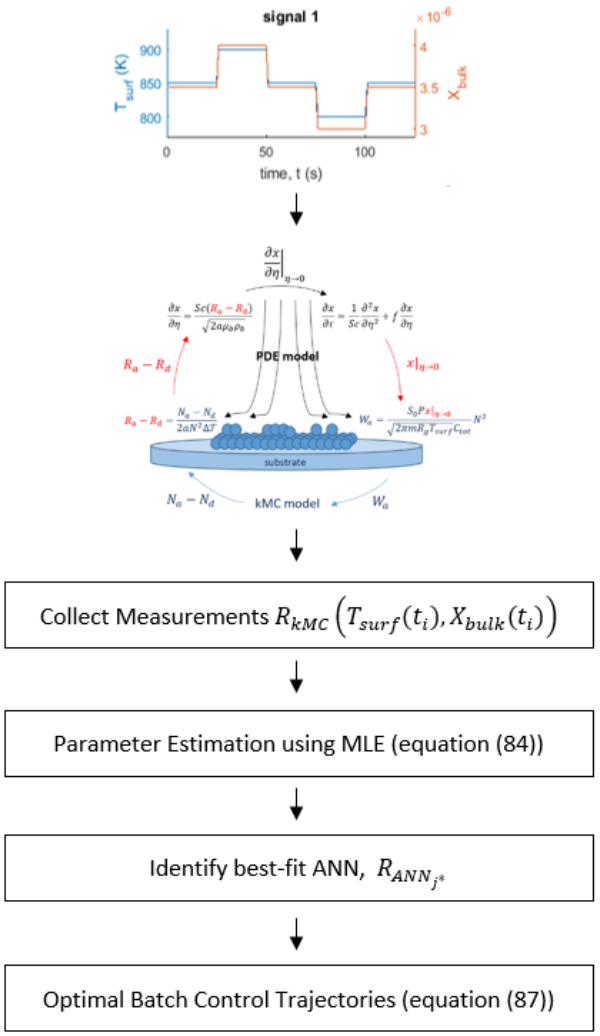

Figure 51: The workflow for performing NMPC under uncertainty in the E parameter.

Table 24: Thin film properties that resulted from ANN-based NMPC

	(a) E = 16,151 cal/mol	*(b) E = 16,988 cal/mol*	*(c) E = 17,852 cal/mol*
$\overline{Gr}$	*35.88 ML/s*	*35.63 ML/s*	*35.56 ML/s*
R_{kMC}(t = 450 s)	*2.15 ML*	*2.28 ML*	*2.38 ML*
E estimate (% relative error)	*15,974 cal/mol (-1.1 %)*	*16,589 cal/mol (-2.4 %)*	*17,362 cal/mol (-2.74 %)*

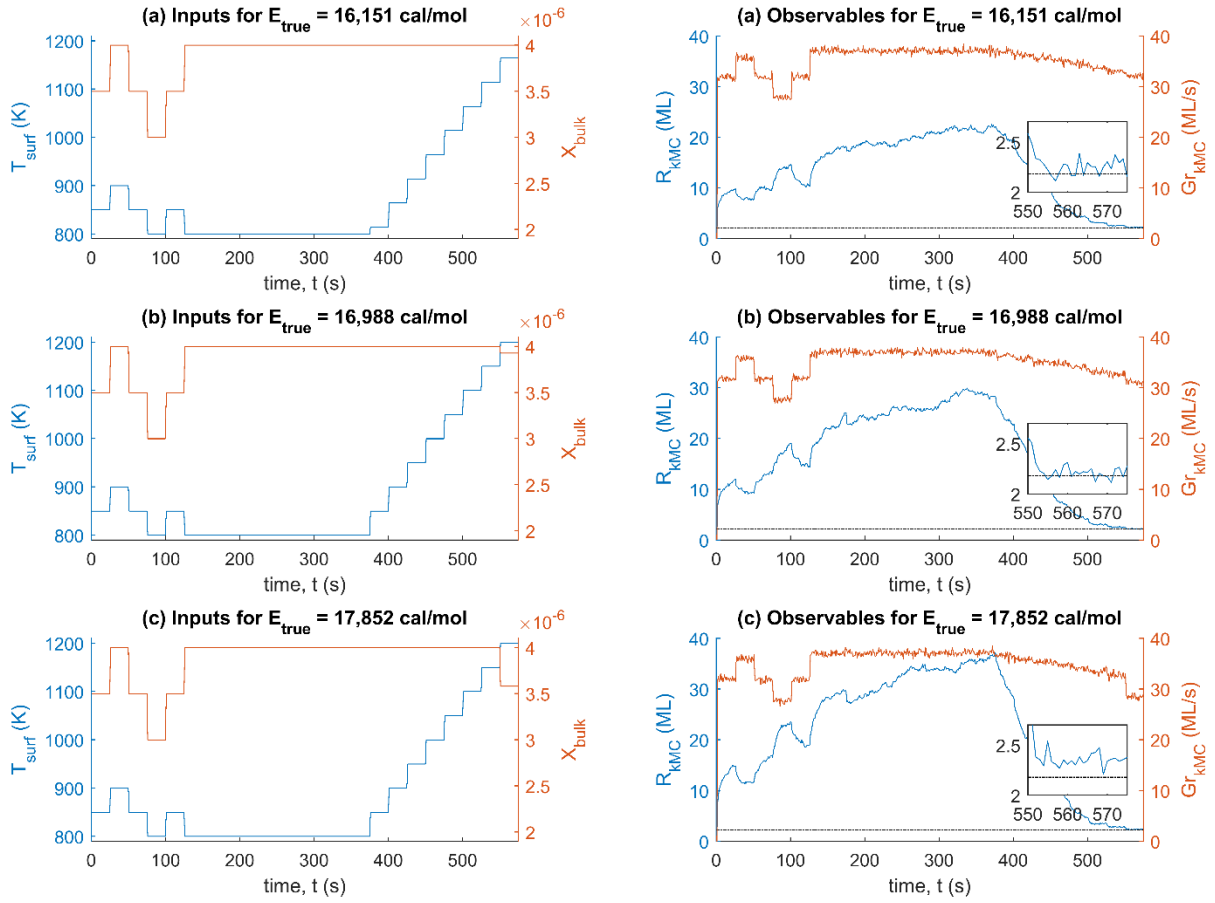

Figure 52: The ANN-based NMPC was able to drive the plant towards the imposed roughness target under various uncertain parameter realizations. The roughness target is shown by the horizontal dotted line in the right-hand plots. Note that the insets in the right-hand plots show the behaviour of roughness achieved by the process during the final sampling interval.

As shown in Table 24, the estimates of the uncertain parameter E agreed closely with the true E realizations, with the largest relative error of only -2.74 % for the case with true E being 17,852 cal/mol. Note that in the present approach, ANNs were not trained for the E realizations that were used to run the thin film process. Thus, a deviation between the true E and the ANN model that considers a realization in E (closer to the true E) is expected. Furthermore, the final thin film roughness was close to the imposed target of 2.18 ML in all cases. However, it was satisfied only for the lowest realization of the uncertain parameter (case (a) in Table 24 and Figure 52a), while the larger uncertain parameter realizations caused the final film roughness to exceed ANN

predictions, likely due to greater stochastic noise present at those E values. The average growth rate values $\overline{Gr}$ were similar regardless of the uncertain parameter realization since the growth rate is insensitive to this kind of parametric uncertainty, as discussed earlier. However, the $\overline{Gr}$ values did decrease slightly at larger uncertain parameter realizations because the ANN-based NMPC adjusted T_{surf} and X_{bulk} according to the uncertainty realizations in order to meet the imposed thin film roughness target.

Figure 52 summarizes the time trajectories of the input variables T_{surf} and X_{bulk}, as well as the observables of the thin films, i.e. R and Gr. The time trajectories of the inputs include signal 1 (see Figure 48) used for ANN discrimination (from 0 s to 125 s in Figure 52), and the subsequent T_{surf} and X_{bulk} values determined by the ANN-based online NMPC framework. It is noteworthy that the time trajectories of the inputs changed depending on the realization of E. The greater the uncertain parameter realization, the higher the final T_{surf} and the lower the final X_{bulk}. Therefore, if uncertainty were to be ignored and instead the ANN for nominal E were to be used, the input variables' profiles generated by NMPC would be suboptimal – for E realizations below the nominal, $\overline{Gr}$ achieved by the process would not be at its maximum possible value, while for E realizations above the nominal, the target R^* would be substantially overestimated by NMPC and, consequently, not met by the thin film process.

In all cases, the thin film roughness was decreased to match or closely approach the imposed roughness target. This target is indicated in the plots of Figure 52 by the horizontal dotted line. The insets in the right-hand plots of Figure 52 demonstrate that during the final sampling interval, the roughness observable was at the target or close to it but exhibited noise due to the stochastic nature of the process. Thus, even though Table 24 indicated that the final roughness target was

satisfied only in scenario (a), Figure 52 showed that in scenario (b) the thin film roughness in the last sampling interval was also at the imposed target during that interval (see the inset plot for scenario (b)). In scenario (c), during the last sampling interval the thin film roughness tended to be above the imposed target while exhibiting noise. This is likely due to larger noise present at higher realizations in E, which caused the ANNs trained on the data generated at such E values to average out the stochastic noise, creating a slight mismatch between the process and the ANN predictions. However, this issue can likely be resolved by training the ANNs on larger datasets. In all cases, ANN-based NMPC required less CPU time than the sampling interval duration (25 s). Thus, the ANNs were able to efficiently provide accurate estimates of the observables online, and hence they are a promising approach to perform online control of stochastic multiscale systems under uncertainty.

6.5.5 Scenario B: Disturbance rejection

To further test the performance of the ANN-based NMPC framework, we studied the effect of large unexpected changes in the input variables T_{surf} and X_{bulk} (portrayed as $d(t_i)$ in Figure 43) on the ability of ANN-based NMPC to achieve the specified targets in the observables despite the unexpected behaviour of the inputs during operation (i.e. reject the disturbances). Using the same optimization procedure and roughness target as described earlier, we added large asynchronous disturbances of different durations early and late in the batch process. Note that the magnitudes of the disturbances performed for this scenario were not considered during ANN training; T_{surf} was perturbed by 200 K and X_{bulk} was changed by 2×10^{-6} (i.e. both disturbances were four times larger than what was used for ANN training).

Disturbance 1 occurred shortly after the best-fit ANN was identified by MLE, i.e. early in the shrinking horizon optimization which commenced at 126 s: T_{surf} was perturbed at 175 s for 50 s

(i.e. for two sampling intervals), and X_{bulk} was perturbed at 200 s for 25 s (for one sampling interval). Disturbance 2 occurred late in the batch: T_{surf} was perturbed at 426 s for 25 s, and X_{bulk} was perturbed at 451 s for 50 s. The time trajectories of the inputs as well as the responses in the observables are shown in Figure 53. Note that the results in Figure 53 were obtained using E realization of 16,151 cal/mol. Other uncertain parameter realizations were considered, but omitted for brevity, since the ANN performance is somewhat similar.

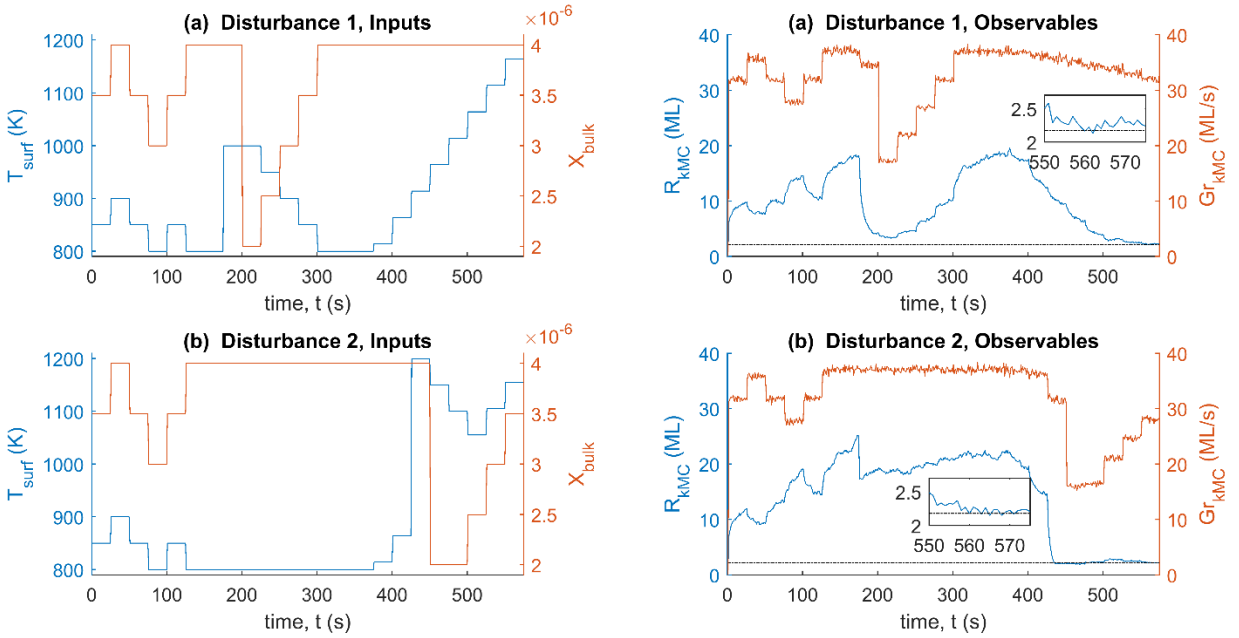

Figure 53: The ANN-based NMPC was able to reject disturbances in the manipulated variables that were 4 times larger than what was seen in the training dataset. Note that the insets in the right-hand plots show the behaviour of roughness achieved by the process during the final sampling interval.

It can be seen from Figure 53 that in both disturbance cases the asynchronous disturbances of different durations in both input variables were successfully rejected by the ANN-based NMPC. Disturbance 1 occurred before the ramp up in T_{surf} would normally begin, and the NMPC adjusted both manipulated inputs back to the values that would promote the maximum growth rate (i.e. 800 K and 4×10^{-6}) before proceeding to increase T_{surf} to meet the thin film roughness requirement. Disturbance 2 took place after the normal ramp up in T_{surf}, but the NMPC successfully adjusted

both manipulated variables to promote growth rate of the thin film as much as possible within the limited remaining time in the batch. As depicted by the insets in the right-hand plots of Figure 53, the thin film roughness exhibited stochastic noise in the last sampling interval but it satisfied the imposed R^* target.

6.6 Chapter Summary

In these studies, 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.

First, the ANNs were applied to the process at nominal conditions (no parametric uncertainty). Next, two approaches were considered when the process was subject to uncertainty. In the first approach, ANNs were used to predict the responses of statistical moments of the observables and subsequently conduct open-loop NMPC. In the second approach, ANNs were trained under various time-invariant realizations of the uncertain parameter to predict the dynamic responses of the observables that corresponded to the realizations. Maximum likelihood and minimum mean square error were used to perform parameter estimation and the identification of the corresponding best-fit ANN. The ANN was then used in closed-loop shrinking horizon NMPC.

The ANN models captured system behaviour under uncertainty accurately and provided orders-of-magnitude computational savings over implementing the optimal profiles on the full stochastic multiscale model. Furthermore, the ANN models were able to reject and compensate for the

disturbances (also under uncertainty) to the manipulated variables that were three to four times greater than what was seen in the ANN training data. The results show that ANNs are an effective technique to accurately capture the behaviour of stochastic multiscale systems under uncertainty emerging in relevant industrial applications; therefore, they can be used for online control and optimization applications.

Chapter 7: Conclusions & Recommendations

In order to improve industrial product quality, stochastic multiscale models attempt to explicitly account for the stochastic phenomena that occur at the microscale and use this information to inform device-relevant domain sizes (i.e. macroscale calculations). Since these models couple micro and macro scales and incorporate non-closed-form expressions to represent the microscale, their computational cost is quite high, and their observables exhibit stochastic noise that propagated from the microscale calculations.

The noise present in the observables complicates the application of standard uncertainty quantification methods such as Power Series and Polynomial Chaos Expansions (PSE and PCE, respectively) which rely on accurate observable values to identify the PSE and PCE expressions. However, uncertainty must be accounted for in order to make sure that the products of stochastic multiscale systems meet imposed quality requirements. Furthermore, the noise levels in the observables need to be estimated and controlled in order to make PSE and PCE predictions repeatable and trustworthy. In addition, the high computational cost of stochastic multiscale models impedes the usefulness of their accurate results because these models cannot be used directly for online optimization and model predictive control of the physical systems they represent.

The work presented in this thesis sought to contribute to the solution of these challenges by comparing the performance of PSE and PCE for uncertainty quantification in stochastic multiscale systems, applying Multilevel Monte Carlo (MLMC) sampling to estimate the noise in the observables and improve uncertainty quantification, and training Artificial Neural Networks (ANNs) to conduct uncertain parameter estimation and robust control of stochastic multiscale systems in the presence of uncertainty.

7.1: Summary of Contributions

The comparison of the performance of PCE and PSE for stochastic multiscale systems showed that PCE with coefficients calculated by the nonintrusive spectral projection (NISP) method appears to be more vulnerable to noise in stochastic multiscale systems than the least squares technique. Increasing the number of NISP roots could improve the accuracy with which PCE could capture the mean of the distribution. Also, PSE appeared to be the most robust to noise in stochastic multiscale systems.

PSE that used centered order finite differences calculated based on 3 realizations of uncertain parameters was more accurate than PCE NISP with 3 roots, which can indicate greater computational efficiency of PSE for achieving the same level of accuracy. However, a 50% step in the uncertain parameter realizations was necessary to achieve accurate PSE performance. Such a large perturbation may not be physically realizable in all stochastic multiscale systems. PSE also appears to be more efficient than PCE with coefficients calculated by the least squares method because the latter needs many realizations of the uncertain parameters to do a proper fit of the coefficients.

The noise inherent in stochastic multiscale systems requires duplicate simulations when gathering the data for PSE and PCE. Heuristic rules are used to determine the number of duplicates. However, MLMC sampling can be used with stochastic multiscale systems to help control the precision in the identification of uncertainty quantification expressions by tuning its root mean square error tolerance ϵ. It is recommended to initially set ϵ to 5% of the expected value of the observable and adjust the value as necessary. However, it should be noted that MLMC capabilities differ for continuous systems (e.g. classical chemical engineering models comprised of closed-form equations such as simulations of a wastewater treatment plant or a distillation column) and

for stochastic multiscale systems (e.g. chemical vapour deposition and catalytic pore models). In the latter case, a lower bound on the achievable ϵ value exists because of the inherent stochastic noise. Also, the accuracy with which the actual distribution of the observable will be captured ultimately depends on the choice of the uncertainty quantification method (e.g. PSE with centered order finite differences and sensitivities obtained using 50% perturbations in the uncertain parameter, PCE with coefficients calculated using least squares approximation with a large number of realizations/points, PCE NISP with a varying number of roots, etc.).

The advantage of data-driven models such as ANNs over PSE, PCE and MLMC is their ability to predict dynamic responses of the observables. In the presented work, ANN models captured the behaviour of the stochastic multiscale system under uncertainty accurately and provided orders-of-magnitude computational savings over implementing the optimal profiles on the full stochastic multiscale model. ANNs were also successfully used to approximate uncertain parameter realizations via maximum likelihood estimation. Furthermore, the ANN models were able to reject and compensate for the disturbances (also under uncertainty) that were three to four times greater than what was seen in the ANN training data. These results show that ANNs are an effective technique to accurately capture the behaviour of stochastic multiscale systems under uncertainty; therefore, they can be used for online control and optimization applications.

7.2: Recommendations for Further Research

The findings and conclusions from the presented studies imply that the following work can potentially be conducted to further extend this research and fill additional knowledge gaps.

- Unlike PSE and PCE, MLMC does not require *a priori* knowledge or assumption of the shape of the parametric uncertainty. However, only Gaussian uncertain parameter

distributions have been predominantly considered in this work for the application of MLMC sampling in stochastic multiscale systems. Thus, it would be of interest to compare the performance of the uncertainty quantification techniques for non-Gaussian uncertain parameters.

- Furthermore, one of the conclusions indicated that the choice of the uncertainty quantification technique ultimately determines the accuracy with which the shape of the distribution of the observables is captured. MLMC has been applied to improve the precision with which PCE NISP expansions for stochastic multiscale systems are identified and should therefore be applied to construct other uncertainty quantification expressions (e.g. PSE) for uncertainty analysis in stochastic multiscale systems to extend the work of Chapter 5.
- In addition, it is recommended to use MLMC directly to calculate higher order statistical moments of the observables (e.g. standard deviation, skewness, kurtosis) and compare the results to the predictions generated by PSE/PCE.
- The stochastic multiscale models considered in this research relied on a number of simplifying assumptions. For example, the chemical vapour deposition model assumed that the gas concentration in the boundary layer above the substrate (i.e. in the macroscale portion of the multiscale model) was spatially invariant. The catalytic pore reactor model relaxed this assumption by using the multigrid ("gap-tooth") approach towards coupling the two scales (continuum and discrete), but it nevertheless assumed that the concentration of macroscale components varies only with respect to the length of the catalytic pore. Furthermore, on-lattice kMC was used in both models to represent the microscale. This research showed that when MLMC is applied to stochastic multiscale models, the

implementation of the application differs depending on the underlying stochastic multiscale model. Thus, it is recommended to consider multiscale models that relax the mentioned simplifying assumptions, apply MLMC to them and show how multiscale model complexity affects the implementation of MLMC sampling. For instance, models that have the following attributes could be examined in subsequent research: multidimensional spatial gradients in the coupling boundary condition, off-lattice kMC or MD represents the microscale with more simultaneous events and reactions, more than two scales are coupled, coarse-graining is used to access larger temporal and spatial domains rather than only coupling through the boundary conditions.

- ANNs showed great promise in their application to stochastic multiscale systems. However, their major drawback is the requirement for generating large training datasets from stochastic multiscale systems, which can be computationally expensive, especially under uncertainty. To lessen the computational burden, it is recommended to apply ANNs to predict the dynamic behaviour of PSE sensitivities and PCE coefficients. This approach could yield predictions of observables' distributions and the ability to calculate statistical moments of the observables of arbitrary order while substantially diminishing the computational cost associated with generating ANN training data. However, one potential area of difficulty with this research will be the determination of how many duplicate simulations should be conducted at each realization of the uncertain parameters to make sure that the effects of stochastic noise in the observables are averaged out.
- One of the approaches that considered ANNs for controlling stochastic multiscale systems under uncertainty relied on open-loop control and used ANNs to predict the statistical moments of the distributions of the observables to account for uncertainty. Online

measurements from the plant were not used to improve the ANN-based estimations in real time because a single datapoint collected at a time could not improve the accuracy with which the statistical moments were predicted. It is therefore recommended to devise a scheme where ANNs can approximate the statistical moments of the observables and inform those estimates using measurements from the plant. For example, the measurements could be incorporated into a kernel density estimation scheme, the statistical moments could be estimated from the assumed kernel' shape and then serve as initial conditions for ANN-based estimations of the mean (as well as standard deviation, skewness, etc.) of the observables.

- In addition, the presented work with ANNs for control under uncertainty considered only time-invariant Gaussian parametric uncertainty. It is recommended to relax these assumptions to further improve the realism of the considered scenarios. For instance, time-varying parametric uncertainty could be considered, in which case maximum likelihood estimation could be carried out several times in the shrinking horizon optimization scheme to make sure that the ANN used for online control remains the most appropriate choice for the current uncertain parameter realization. Multiparameter uncertainty and non-Gaussian distributions should also be considered. Furthermore, other sources of uncertainty (e.g. model structure error) should be considered in addition to parametric uncertainty.
- In this work, closed-loop ANN-based control showed that the ANNs can benefit from real-time measurements provided as initial conditions for the next iteration of shrinking horizon optimization. However, the ANNs remained oblivious to any discrepancies between their forecasts and the measurements, particularly in the high-noise regimes. Thus, one area of further research is to consider using active learning (i.e. real-time retraining of ANNs) to

improve ANN-based predictions of any measurements of the observables obtained from the process in real-time. This approach would be similar in spirit to using real-time user behaviour data that modern recommendation engines (e.g. YouTube, Netflix, etc.) use to improve their accuracy.

- In this research, ANNs were trained using synthetic data obtained from mathematical models. It would be beneficial to develop ANNs using experimental data or compare the predictions of the trained ANNs to experiments to assess the accuracy of the synthetic training data approach. Alternatively, synthetic and experimental data could be combined prior to training the ANNs.
- Interpretable ANNs remain an area of research in many fields (e.g. image recognition, natural language understanding, etc.). Technical disciplines such as chemical engineering can have an advantage over the more widely studied and commercialized applications of ANNs because many first-principles models exist that could inform the ANNs regarding the significance of correlations between the manipulated variables and the observables. These first-principles models could be leveraged to develop hybrid approaches that integrate first-principles models with ANNs and yield interpretable ANNs where different weights and biases have clear physical significance.
- This work attempted to capture uncertainty in the observables by using ANNs to predict statistical moments. Other approaches for considering uncertainty in the output of ANNs exist, but they tend to involve manipulations of the output data (e.g. bootstrapping). An unexplored area of research (which is related to interpretable ANNs) is the assessment of the impact of the uncertainty of ANN weights and biases on the variability of ANN predictions. Thus, it is recommended to attempt to capture uncertainty in the

observables/ANN predictions through the uncertainty/confidence intervals in the ANN weights.

- Similar to the recommendation for MLMC sampling, more sophisticated stochastic multiscale models should be considered in further research involving ANN-based control schemes. In fact, while those models would be even more computationally expensive than those considered in this research, their most computationally intensive parts could in principle be replaced with ANN-based predictors. Thus, hybrid multiscale models could be developed where ANN substitute the most computationally intensive scales. Such models could be sufficiently accurate and fast to be directly incorporated into online optimization and model predictive control under uncertainty, and therefore assist in devising approaches for optimal design and control of stochastic multiscale systems and their products.

Bibliography

1. Christofides PD, Armaou A. Control and optimization of multiscale process systems. *Comput Chem Eng*. 2006;30(10-12):1670-1686.

2. Christofides PD, Armaou A, Lou Y, Varshney A. *Control and Optimization of Multiscale Process Systems*. 1st ed. (Levine WS, ed.). New York, NY: Birkhäuser Boston; 2009.

3. Salciccioli M, Stamatakis M, Caratzoulas S, Vlachos DG. A review of multiscale modeling of metal-catalyzed reactions: Mechanism development for complexity and emergent behavior. *Chem Eng Sci*. 2011;66(19):4319-4355.

4. Raimondeau S, Vlachos DG. Low-Dimensional Approximations of Multiscale Epitaxial Growth Models for Microstructure Control of Materials. *J Comput Phys*. 2000;160(2):564-576.

5. Crose M, Sang-Il Kwon J, Nayhouse M, Ni D, Christofides PD. Multiscale modeling and operation of PECVD of thin film solar cells. *Chem Eng Sci*. 2015;136:50-61. doi:10.1016/j.ces.2015.02.027

6. Kwon JS-I, Nayhouse M, Christofides PD, Orkoulas G. Modeling and control of crystal shape in continuous protein crystallization. *Chem Eng Sci*. 2014;107:47-57. doi:10.1016/j.ces.2013.12.005

7. Kwon JS-I, Nayhouse M, Orkoulas G, Christofides PD. Crystal shape and size control using a plug flow crystallization configuration. *Chem Eng Sci*. 2014;119:30-39. doi:10.1016/j.ces.2014.07.058

8. Kwon JS-I, Nayhouse M, Christofides PD, Orkoulas G. Modeling and Control of Protein Crystal Shape and Size in Batch Crystallization. *AIChE J*. 2013;59(7):2317-2327. doi:10.1002/aic.14039

9. Chaffart D, Ricardez-Sandoval LA. Robust Optimization of a Multiscale Heterogeneous Catalytic Reactor System with Spatially-Varying Uncertainty Descriptions using Polynomial Chaos Expansions. *Can J Chem Eng*. 2017;96(1):113-131. doi:10.1002/cjce.22912

10. Chaffart D, Ricardez-Sandoval LA. Robust dynamic optimization in heterogeneous multiscale catalytic flow reactors using polynomial chaos expansion. *J Process Control*. 2017;60:128-140. doi:10.1016/j.jprocont.2017.07.002

11. Huang J, Zhang X, Orkoulas G, Christofides PD. Dynamics and control of aggregate thin film surface morphology for improved light trapping: Implementation on a large-lattice kinetic Monte Carlo model. *Chem Eng Sci*. 2011;66(23):5955-5967. doi:10.1016/j.ces.2011.08.020

12. Braatz RD, Alkire RC, Seebauer E, et al. Perspectives on the design and control of multiscale systems. *J Process Control*. 2006;16(3):193-204.

13. Vlachos DG. Multiscale Integration Hybrid Algorithms for Homogeneous–Heterogeneous Reactors. *AIChE J*. 1997;43(11):3031-3041.

14. Bevan MA, Ford DM, Grover MA, et al. Controlling assembly of colloidal particles into structured objects: Basic strategy and a case study. *J Process Control*. 2015;27:64-75. doi:10.1016/j.jprocont.2014.11.011

15. Lam R, Vlachos DG. Multiscale model for epitaxial growth of films: Growth mode transition. *Phys Rev B*. 2001;64(3):035401.

16. Vlachos DG. A Review of Multiscale Analysis: Examples from Systems Biology, Materials Engineering, and Other Fluid–Surface Interacting Systems. *Adv Chem Eng*. 2005;30:1-61.

17. Kwon JS-I, Nayhouse M, Christofides PD. Multiscale, Multidomain Modeling and Parallel Computation: Application to Crystal Shape Evolution in Crystallization. *Ind Eng Chem Res*. 2015;54(47):11903–11914. doi:10.1021/acs.iecr.5b02942

18. Crose M, Zhang W, Tran A, Christofides PD. Multiscale three-dimensional CFD modeling for PECVD of amorphous silicon thin films. *Comput Chem Eng*. 2018;113:184-195. doi:10.1016/j.compchemeng.2018.03.011

19. Tang X, Rupp B, Yang Y, Edwards TD, Grover MA, Bevan MA. Optimal Feedback Controlled Assembly of Perfect Crystals. *ACS Nano*. 2016;10(7):6791-6798. doi:10.1021/acsnano.6b02400

20. Tang X, Xue Y, Grover MA. Colloidal Self-Assembly with Model Predictive Control. In: *2013 American Control Conference*. IEEE; 2013:4228-4233. doi:10.1109/ACC.2013.6580489

21. Ricardez-Sandoval LA. Current Challenges in the Design and Control of Multiscale Systems. *Can J Chem Eng*. 2011;89(6):1324-1341.

22. Rasoulian S, Ricardez-Sandoval LA. Stochastic nonlinear model predictive control applied to a thin film deposition process under uncertainty. *Chem Eng Sci*. 2016;140:90-103.

23. Miller DC, Syamlal M, Mebane DS, et al. Carbon Capture Simulation Initiative : A Case Study in Multiscale Modeling and New Challenges. 2014. doi:10.1146/annurev-chembioeng-060713-040321

24. Adomaitis RA. Multiscale modeling and optimization of an atomic layer deposition process for nanomanufacturing applications. In: *IFAC Proceedings Volumes*. Vol 43. Elsevier; 2010:859-864. doi:10.3182/20100705-3-BE-2011.00142

25. Snyder MA, Chatterjee A, Vlachos DG. Net-event kinetic Monte Carlo for overcoming stiffness in spatially homogeneous and distributed systems. *Comput Chem Eng*. 2005;29(4):701-712.

26. Cheimarios N, Kokkoris G, Boudouvis AG. Multiscale modeling in chemical vapor deposition processes: Coupling reactor scale with feature scale computations. *Chem Eng Sci*. 2010;65(17):5018-5028. doi:10.1016/j.ces.2010.06.004

27. Zhang Y, Ding Y, Christofides PD. Multiscale computational fluid dynamics modeling of thermal atomic layer deposition with application to chamber design. *Chem Eng Res Des*. 2019;147:529-544. doi:10.1016/j.cherd.2019.05.049

28. Adomaitis RA. Development of a multiscale model for an atomic layer deposition process. *J Cryst Growth*. 2010;312(8):1449-1452. doi:10.1016/j.jcrysgro.2009.12.041

29. Müller F, Jenny P, Meyer DW. Multilevel Monte Carlo for two phase flow and Buckley-Leverett transport in random heterogeneous porous media. *J Comput Phys*. 2013;250:685-702.

30. Icardi M, Boccardo G, Tempone R. On the predictivity of pore-scale simulations: Estimating uncertainties with multilevel Monte Carlo. *Adv Water Resour*. 2016;95:46-60.

31. Chaffart D, Ricardez-Sandoval LA. Optimization and Control of a Thin Film Growth Process: A Hybrid First Principles/Artificial Neural Network Based Multiscale Modelling Approach. *Comput Chem Eng*. 2018;119:465-479. doi:10.1016/j.compchemeng.2018.08.029

32. Choi H-K, Kwon JS-I. Multiscale modeling and multiobjective control of wood fiber morphology in batch pulp digester. *AIChE J*. March 2020:e16972. doi:10.1002/aic.16972

33. Choi H-K, Kwon JS-I. Multiscale modeling and control of Kappa number and porosity in a batch-type pulp digester. *AIChE J*. 2019;65(6):e16589. doi:10.1002/aic.16589

34. Rasoulian S, Ricardez-Sandoval LA. A robust nonlinear model predictive controller for a multiscale thin film deposition process. *Chem Eng Sci*. 2015;136:38-49.

35. Rasoulian S, Ricardez-Sandoval LA. Uncertainty analysis and robust optimization of multiscale process systems with application to epitaxial thin film growth. *Chem Eng Sci*. 2014;116:590-600.

36. Rasoulian S, Ricardez-Sandoval LA. Robust multivariable estimation and control in an epitaxial thin film growth process under uncertainty. *J Process Control*. 2015;34:70-81.

37. Chaffart D, Rasoulian S, Ricardez-Sandoval LA. Distributional Uncertainty Analysis and Robust Optimization in Spatially Heterogeneous Multiscale Process Systems. *AIChE J*. 2016;62(7):2374-2390. doi:10.1002/aic.15215

38. Nagy ZK, Braatz RD. Distributional uncertainty analysis using power series and polynomial chaos expansions. *J Process Control*. 2007;17(3):229-240.

39. Rasoulian S, Ricardez-Sandoval LA. Worst-case and Distributional Robustness Analysis of a Thin Film Deposition Process. In: *9th International Symposium on Advanced Control of Chemical Processes*. ; 2015:1127-1132.

40. Bahakim SS, Rasoulian S, Ricardez-Sandoval LA. Optimal Design of Large-Scale Chemical Processes Under Uncertainty: A Ranking-Based Approach. *AIChE J*. 2014;60(9):3243-3257.

41. Helton JC, Davis FJ. Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliab Eng Syst Saf*. 2003;81(1):23-69.

42. Helton JC, Davis FJ, Johnson JD. A comparison of uncertainty and sensitivity analysis results obtained with random and Latin hypercube sampling. *Reliab Eng Syst Saf*. 2005;89(3):305-330.

43. Tran A, Sun J, Liu D, Wildey T, Wang Y. Multiscale stochastic reduced-order model for uncertainty propagation using Fokker-Planck equation with microstructure evolution applications. 2020. https://arxiv.org/abs/2004.06487.

44. Braatz RD, Alkire RC, Seebauer EG, et al. A multiscale systems approach to microelectronic processes. *Comput Chem Eng*. 2006;30:1643-1656.

45. Braatz RD, Alkire RC, Rusli E, Drews TO. Multiscale systems engineering with applications to chemical reaction processes. *Chem Eng Sci*. 2004;59(22-23):5623-5628. doi:10.1016/j.ces.2004.09.022

46. Bahramian A. Study on growth rate of TiO2 nanostructured thin films: Simulation by molecular dynamics approach and modeling by artificial neural network. *Surf Interface Anal*. 2013;45(11):1727-1736. doi:10.1002/sia.5314

47. Kim CE, Moon P, Yun I, et al. Process estimation and optimized recipes of ZnO:Ga thin film characteristics for transparent electrode applications. *Expert Syst Appl*. 2011;38(3):2823-2827. doi:10.1016/j.eswa.2010.08.074

48. Ho W, Tsai J, Hsu G, Chou J. Process Parameters Optimization: A Design Study for TiO2 Thin Film of Vacuum Sputtering Process. *IEEE Trans Autom Sci Eng*. 2010;7(1):143-146. doi:10.1109/TASE.2009.2023673

49. Guha N, Tan X. Multilevel approximate Bayesian approaches for flows in highly heterogeneous porous media and their applications. *J Comput Appl Math*. 2017;317:700-717.

50. Lu D, Zhang G, Webster C, Barbier C. An improved multilevel Monte Carlo method for estimating probability distribution functions in stochastic oil reservoir simulations. *Water Resour*

Res. 2016;52(12):9642-9660.

51. Alkhatib A, Babaei M. Applying the Multilevel Monte Carlo Method for Heterogeneity-Induced Uncertainty Quantification of Surfactant / Polymer Flooding. *SPE J*. 2016;21(4):1192-1203.

52. Alkhatib AM. Robust Quantification of Uncertainty in Heterogeneity for Chemical EOR Processes: Applying the Multi-Level Monte Carlo Method. In: *SPE Middle East Oil & Gas Show and Conference*. Manama, Bahrain: Society of Petroleum Engineers; 2015. doi:https://doi.org/10.2118/172635-MS

53. Alsaif SF, Alkhatib AM, Marsala AF. Reservoir Saturation Mapping from Electromagnetic Geophysical Surveys using Advanced Uncertainty Quantification Methods. In: *SPE Middle East Oil & Gas Show and Conference*. Manama, Bahrain: Society of Petroleum Engineers; 2017.

54. Kimaev G, Ricardez-Sandoval LA. A comparison of efficient uncertainty quantification techniques for stochastic multiscale systems. *AIChE J*. 2017;63(8):3361-3373. doi:10.1002/aic.15702

55. Kimaev G, Ricardez-Sandoval LA. Multilevel Monte Carlo applied to chemical engineering systems subject to uncertainty. *AIChE J*. 2018;64(5):1651-1661. doi:10.1002/aic.16045

56. Kimaev G, Ricardez-Sandoval LA. Multilevel Monte Carlo for noise estimation in stochastic multiscale systems. *Chem Eng Res Des*. 2018;140(Special Issue on Dynamics and Control):33-43. doi:10.1016/j.cherd.2018.10.006

57. Kimaev G, Chaffart D, Ricardez-Sandoval LA. Multilevel Monte Carlo Applied for Uncertainty Quantification in Stochastic Multiscale Systems. *AIChE J*. May 2020:e16262. doi:10.1002/aic.16262

58. Kimaev G, Ricardez-Sandoval LA. Nonlinear model predictive control of a multiscale thin film deposition process using artificial neural networks. *Chem Eng Sci*. 2019;207:1230-1245. doi:10.1016/j.ces.2019.07.044

59. Li J, Croiset E, Ricardez-Sandoval LA. Carbon nanotube growth: First-principles-based kinetic Monte Carlo model. *J Catal*. 2015;326:15-25. doi:10.1016/j.jcat.2015.03.010

60. Crose M, Kwon JS-I, Tran A, Christofides PD. Multiscale modeling and run-to-run control of PECVD of thin film solar cells. *Renew Energy*. 2017;100:129-140. doi:10.1016/j.renene.2016.06.065

61. Adomaitis RA. A Reduced-Basis Discretization Method for Chemical Vapor Deposition Reactor Simulation. *Math Comput Model*. 2003;38(1-2):159-175.

62. Lee D, Mohr A, Kwon JS-I, Wu H-J. Kinetic Monte Carlo modeling of multivalent binding of CTB proteins with GM1 receptors. *Comput Chem Eng*. 2018;118:283-295. doi:10.1016/j.compchemeng.2018.08.011

63. Yang A, Marquardt W. An ontological conceptualization of multiscale models. *Comput Chem Eng*. 2009;33(4):822-837. doi:10.1016/j.compchemeng.2008.11.015

64. Gubbins KE, Moore JD. Molecular modeling of matter: Impact and prospects in engineering. *Ind Eng Chem Res*. 2010;49(7):3026-3046. doi:10.1021/ie901909c

65. Sholl DS, Steckel JA. *Density Functional Theory : A Practical Introduction*. (Steckel JA, ed.). Hoboken, N.J: Wiley; 2009.

66. Leach AR. *Molecular Modelling : Principles and Applications* . 2nd ed. Harlow, England ;

Prentice Hall; 2001.

67. Schroeder D V. *An Introduction to Thermal Physics*. Pearson Education; 2007.

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

69. Gillespie DT. A rigorous derivation of the chemical master equation. *Phys A Stat Mech its Appl*. 1992;188(1-3):404-425.

70. Nielsen SO, Lopez CF, Srinivas G, Klein ML. Coarse grain models and the computer simulation of soft materials. *J Phys Condens Matter*. 2004;16(15):R481-R512. doi:10.1088/0953-8984/16/15/r03

71. Bird RB, Stewart WE, Lightfoot EN. *Transport Phenomena*. 2nd ed. John Wiley & Sons; 2002.

72. Kimaev G, Ricardez-Sandoval LA. A Basic Case Study Addressing Multi-Scale Modelling of Drug Delivery Using COMSOL. Computer Aids for Chemical Engineering. https://cache.org/drug_delivery_case_study. Published 2019.

73. Weinan E, Engquist B, Huang Z. Heterogeneous multiscale method: A general methodology for multiscale modeling. *Phys Rev B - Condens Matter Mater Phys*. 2003;67(9). doi:10.1103/PhysRevB.67.092101

74. Abdulle A, Weinan E, Engquist B, Vanden-Eijnden E. The heterogeneous multiscale method. *Acta Numer*. 2012;21:1-87. doi:10.1017/S0962492912000025

75. Gear CW, Li J, Kevrekidis IG. The gap-tooth method in particle simulations. *Phys Lett A*. 2003;316(3-4):190-195. doi:10.1016/j.physleta.2003.07.004

76. Chatterjee A, Vlachos DG. An overview of spatial microscopic and accelerated kinetic Monte Carlo methods. *J Comput Mater Des*. 2007;14(2):253-308. doi:10.1007/s10820-006-9042-9

77. Chatterjee A, Vlachos DG. Multiscale spatial Monte Carlo simulations: Multigriding, computational singular perturbation, and hierarchical stochastic closures. *J Chem Phys*. 2006;124(6). doi:10.1063/1.2166380

78. Tian T, Burrage K. Binomial leap methods for simulating stochastic chemical kinetics. *J Chem Phys*. 2004;121(21):10356-10364. doi:10.1063/1.1810475

79. Gillespie DT. Approximate accelerated stochastic simulation of chemically reacting systems. *J Chem Phys*. 2001;115(4):1716-1733. doi:10.1063/1.1378322

80. Cao Y, Petzold LR, Rathinam M, Gillespie DT. The numerical stability of leaping methods for stochastic simulation of chemically reacting systems. *J Chem Phys*. 2004;121(24):12169-12178. doi:10.1063/1.1823412

81. Cao Y, Petzold L. Slow-scale tau-leaping method. *Comput Methods Appl Mech Eng*. 2008;197(43-44):3472-3479. doi:10.1016/j.cma.2008.02.024

82. Rathinam M, Petzold LR, Cao Y, Gillespie DT. Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *J Chem Phys*. 2003;119(24):12784-12794. doi:10.1063/1.1627296

83. Cao Y, Gillespie DT, Petzold LR. Efficient step size selection for the tau-leaping simulation method. *J Chem Phys*. 2006;124(4). doi:10.1063/1.2159468

84. Cao Y, Gillespie DT, Petzold LR. Adaptive explicit-implicit tau-leaping method with automatic tau selection. *J Chem Phys*. 2007;126(22). doi:10.1063/1.2745299

85. Dollet A. Multiscale modeling of CVD film growth—a review of recent works. *Surf Coatings Technol*. 2004;177-178:245-251. doi:10.1016/j.surfcoat.2003.09.040

86. Luo EZ, Heun S, Kennedy M, Wollschläger J, Henzler M. Surface roughness and conductivity of thin Ag films. *Phys Rev B*. 1994;49(7):4858-4865.

87. Palasantzas G, Barnaś J. Surface-roughness fractality effects in electrical conductivity of single metallic and semiconducting films. *Phys Rev B*. 1997;56(12):7726-7731.

88. Palasantzas G. Surface roughness and grain boundary scattering effects on the electrical conductivity of thin films. *Phys Rev B*. 1998;58(15):9685-9688.

89. Fuchs K. The conductivity of thin metallic films according to the electron theory of metals. *Math Proc Cambridge Philos Soc*. 1938;34(1):100-108.

90. Fishman G, Calecki D. Surface-induced resistivity of ultrathin metallic films: A limit law. *Phys Rev Lett*. 1989;62(11):1302-1305.

91. Oberkampf WL, Trucano TG, Hirsch C. Verification, validation, and predictive capability in computational engineering and physics. *Appl Mech Rev*. 2004;57(5):345-384.

92. Eldred MS. Recent Advances in Non-Intrusive Polynomial Chaos and Stochastic Collocation Methods for Uncertainty Analysis and Design. In: *AIAA Structures, Structural Dynamics and Materials Conference*. Palm Springs, California; 2009:1-37.

93. Refsgaard JC, van der Sluijs JP, Brown J, van der Keur P. A framework for dealing with uncertainty due to model structure error. *Adv Water Resour*. 2006;29(11):1586-1597.

94. Xiu D. Fast Numerical Methods for Stochastic Computations: A Review. *Commun Comput Phys*. 2009;5(2-4):242-272.

95. Nagy ZK, Allgower F. Nonlinear Model Predictive Control: From Chemical Industry to Microelectronics. In: *43rd IEEE Conference on Decision and Control (CDC)*. ; 2004:4249-4254.

96. Nagy ZK, Braatz RD. Worst-Case and Distributional Robustness Analysis of Finite-Time Control Trajectories for Nonlinear Distributed Parameter Systems. *IEEE Trans Control Syst Technol*. 2003;11(5):694-704.

97. Ma DL, Chung SH, Braatz RD. Worst-Case Performance Analysis of Optimal Batch Control Trajectories. *AIChE J*. 1999;45(7):1469-1476.

98. Nagy ZK, Braatz RD. Open-loop and closed-loop robust optimal control of batch processes using distributional and worst-case analysis. *J Process Control*. 2004;14(4):411-422.

99. Ghanem R, Spanos PD. Polynomial Chaos in Stochastic Finite Elements. *J Appl Mech*. 1990;57:197-202.

100. Xiu D. *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton, New Jersey: Princeton University Press; 2010.

101. Xiu D, Karniadakis GE. The Wiener-Askey Polynomial Chaos For Stochastic Differential Equations. *SIAM J Sci Comput*. 2002;24(2):619-644.

102. Wille H, Ruess M, Rank E, Yosibash Z. Uncertainty quantification for personalized analyses of

human proximal femurs. *J Biomech*. 2016;49(4):520-527.

103. Garcia-Cabrejo O, Valocchi A. Global Sensitivity Analysis for multivariate output using Polynomial Chaos Expansion. *Reliab Eng Syst Saf*. 2014;126:25-36.

104. Lu F, Morzfeld M, Tu X, Chorin AJ. Limitations of polynomial chaos expansions in the Bayesian solution of inverse problems. *J Comput Phys*. 2015;282:138-147.

105. Bilionis I, Zabaras NJ. Solution of inverse problems with limited forward solver evaluations: a Bayesian perspective. *Inverse Probl*. 2014;30(1):015004.

106. Branicki M, Majda AJ. Fundamental limitations of polynomial chaos for uncertainty quantification in systems with intermittent instabilities. *Commun Math Sci*. 2013;11(1):55-103.

107. Hou TY, Luo W, Rozovskii B, Zhou HM. Wiener Chaos expansions and numerical solutions of randomly forced equations of fluid mechanics. *J Comput Phys*. 2006;216(2):687-706.

108. Rice RG, Do DD. *Applied Mathematics and Modeling for Chemical Engineers*. 2nd ed. Hoboken, New Jersey: John Wiley & Sons, Inc.; 2012.

109. Raimondeau S, Aghalayam P, Mhadeshwar AB, Vlachos DG. Parameter optimization in molecular models: Application to surface kinetics. *Ind Eng Chem Res*. 2003;42:1174-1183.

110. McKay MD, Beckman RJ, Conover WJ. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code. *Technometrics*. 1979;21(2):239-245.

111. Box G. E. P., Wilson K. B. On the Experimental Attainment of Optimum Conditions. *J R Stat Soc Ser B*. 1951;13(1):1-45.

112. Myers RH. Response surface methodology--current status and future directions. *J Qual Technol*. 1999;31(1):30-44.

113. Giles M. Multi-level Monte Carlo path simulation. *Oper Res*. 2008;56(3):607-617.

114. Giles M. Multilevel Monte Carlo methods. *Acta Numer*. 2015;24:259-328.

115. Giles M, Nagapetyan T, Ritter K. Multilevel Monte Carlo Approximation of Distribution Functions and Densities. *SIAM/ASA J Uncertain Quantif*. 2015;3:267-295.

116. Bierig C, Chernov A. Estimation of arbitrary order central statistical moments by the Multilevel Monte Carlo Method. *Stoch Partial Differ Equations Anal Comput*. 2016;4:3-40.

117. Bierig C, Chernov A. Approximation of probability density functions by the Multilevel Monte Carlo Maximum Entropy method. *J Comput Phys*. 2016;314:661-681.

118. Wilson D, Baker RE. Multi-level methods and approximating distribution functions. *AIP Adv*. 2016;6(7):075020.

119. Aslett LJM, Nagapetyan T, Vollmer SJ. Multilevel Monte Carlo for Reliability Theory. *Reliab Eng Syst Saf*. 2017;165(March):188-196.

120. Barth A, Schwab C, Šukys J. Multilevel Monte Carlo Simulation of Statistical Solutions to the Navier-Stokes Equations. In: Cools R, Nuyens D, eds. *Monte Carlo and Quasi-Monte Carlo Methods, Springer Proceedings in Mathematics & Statistics*. Leuven, Belgium: Springer; 2016:209-227.

121. Fagerlund F, Hellman F, Malqvist A, Niemi A. Multilevel Monte Carlo methods for computing

failure probability of porous media flow systems. *Adv Water Resour*. 2016;94:498-509.

122. Mishra S, Schwab C, Šukys J. Multi-level Monte Carlo finite volume methods for nonlinear systems of conservation laws in multi-dimensions. *J Comput Phys*. 2012;231(8):3365-3388.

123. Sauer T. Computational solution of stochastic differential equations. *Wiley Interdiscip Rev Comput Stat*. 2013;5(5):362-371. doi:10.1002/wics.1272

124. Abdulle A, Blumenthal A. Stabilized multilevel Monte Carlo method for stiff stochastic differential equations. *J Comput Phys*. 2013;251:445-460.

125. Anderson DF, Higham DJ. Multilevel Monte Carlo for Continuous Time Markov Chains, with Applications in Biochemical Kinetics. *Multiscale Model Simul*. 2012;10(1):146-179.

126. Higham DJ, Mao X, Roj M, Song Q, Yin G. Mean Exit Times and the Multilevel Monte Carlo Method. *SIAM/ASA J Uncertain Quantif*. 2013;1:2-18.

127. Müller F, Meyer DW, Jenny P. Solver-based vs. grid-based multilevel Monte Carlo for two phase flow and transport in random heterogeneous porous media. *J Comput Phys*. 2014;268:39-50.

128. Müller F, Jenny P, Meyer DW. Parallel Multilevel Monte Carlo for Two-Phase Flow and Transport in Random Heterogeneous Porous Media with Sampling Error and Discretization Error Balancing. *SPE J*. 2016;21(6):2027-2037.

129. Efendiev Y, Iliev O, Kronsbein C. Multilevel Monte Carlo methods using ensemble level mixed MsFEM for two-phase flow and transport simulations. *Comput Geosci*. 2013;17(5):833-850.

130. Kumar D, Budman H. Robust nonlinear MPC based on Volterra series and polynomial chaos expansions. *J Process Control*. 2014;24(1):304-317. doi:10.1016/j.jprocont.2013.03.003

131. Kumar D, Budman H. Robust nonlinear predictive control for a bioreactor based on a Dynamic Metabolic Flux Balance model. *IFAC-PapersOnLine*. 2015;48(8):930-935. doi:10.1016/j.ifacol.2015.09.089

132. Mandur J, Budman H. Robust optimization of chemical processes using Bayesian description of parametric uncertainty. *J Process Control*. 2014;24:422-430.

133. Hille R, Mandur J, Budman HM. Robust batch-to-batch optimization in the presence of model-plant mismatch and input uncertainty. *AIChE J*. 2017;63(7):2660-2670. doi:10.1002/aic.15645

134. Mandur JS, Budman HM. Robust Algorithms for Simultaneous Model Identification and Optimization in the Presence of Model-Plant Mismatch. *Ind Eng Chem Res*. 2015;54(38):9382-9393. doi:10.1021/acs.iecr.5b01560

135. Mandur J, Budman H. A Polynomial-Chaos based Algorithm for Robust optimization in the presence of Bayesian Uncertainty. *IFAC Proc Vol*. 2012;45(15):549-554. doi:https://doi.org/10.3182/20120710-4-SG-2026.00041

136. Paulson JA, Martin-Casas M, Mesbah A. Fast uncertainty quantification for dynamic flux balance analysis using non-smooth polynomial chaos expansions. *PLOS Comput Biol*. 2019;15(8):e1007308. https://doi.org/10.1371/journal.pcbi.1007308.

137. Peng X, Lou L, Cai Y, et al. Uncertainty quantification of the SPND dynamic model based on non-intrusive polynomial chaos method. *Ann Nucl Energy*. 2019;133:73-83. doi:https://doi.org/10.1016/j.anucene.2019.04.014

138. Xiong F, Chen S, Xiong Y. Dynamic system uncertainty propagation using polynomial chaos.

Chinese J Aeronaut. 2014;27(5):1156-1170. doi:https://doi.org/10.1016/j.cja.2014.08.010

139. Jalali H, Nieuwenhuyse I Van, Picheny V. Comparison of Kriging-based algorithms for simulation optimization with heterogeneous noise. *Eur J Oper Res*. 2017;261(1):279-301. doi:10.1016/j.ejor.2017.01.035

140. Kleijnen JPC. Regression and Kriging metamodels with their experimental designs in simulation: A review. *Eur J Oper Res*. 2017;256(1):1-16. doi:10.1016/j.ejor.2016.06.041

141. Lawrynczuk M. Nonlinear predictive control of dynamic systems represented by Wiener–Hammerstein models. *Nonlinear Dyn*. 2016;86:1193-1214. doi:10.1007/s11071-016-2957-0

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

143. Narasingam A, Kwon JS-I. Data-driven identification of interpretable reduced-order models using sparse regression. *Comput Chem Eng*. 2018;119:101-111. doi:10.1016/j.compchemeng.2018.08.010

144. Meidanshahi V, Corbett B, Adams II TA, Mhaskar P. Subspace model identification and model predictive control based cost analysis of a semicontinuous distillation process. *Comput Chem Eng*. 2017;103:39-57. doi:10.1016/j.compchemeng.2017.03.011

145. Solvason CC, Chemmangattuvalappil NG, Eden MR. Multi-Scale Chemical Product Design using the Reverse Problem Formulation. *Comput Aided Chem Eng*. 2010;28:1285-1290.

146. Solvason CC. Integrated Multiscale Chemical Product Design using Property Clustering and Decomposition Techniques in a Reverse Problem Formulation. 2011.

147. Oladyshkin S, Nowak W. Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion. *Reliab Eng Syst Saf*. 2012;106:179-190. doi:10.1016/j.ress.2012.05.002

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

149. Siddhamshetty P, Wu K, Kwon JS-I. Optimization of simultaneously propagating multiple fractures in hydraulic fracturing to achieve uniform growth using data-based model reduction. *Chem Eng Res Des*. 2018;136:675-686. doi:10.1016/j.cherd.2018.06.015

150. Yuan S, Jiao Z, Quddus N, Kwon JS-I, Mashuga C V. Developing Quantitative Structure–Property Relationship Models To Predict the Upper Flammability Limit Using Machine Learning. *Ind Eng Chem Res*. 2019;58(8):3531-3537. doi:10.1021/acs.iecr.8b05938

151. Hasenauer J, Jagiella N, Hross S, Theis FJ. Data-Driven Modelling of Biological Multi-Scale Processes. *J Coupled Syst Multiscale Dyn*. 2015;3(2):101-121. doi:10.1166/jcsmd.2015.1069

152. Garg A, Corbett B, Mhaskar P, Hu G, Flores-Cerrillo J. Subspace-based model identification of a hydrogen plant startup dynamics. *Comput Chem Eng*. 2017;106:183-190. doi:10.1016/j.compchemeng.2017.05.020

153. Garg A, Mhaskar P. Subspace Identification-Based Modeling and Control of Batch Particulate Processes. *Ind Eng Chem Res*. 2017;56(26):7491-7502. doi:10.1021/acs.iecr.7b00682

154. Svozil D, Kvasnicka V, Pospichal J. Introduction to multi-layer feed-forward neural networks. *Chemom Intell Lab Syst*. 1997;39(1):43-62. doi:10.1016/S0169-7439(97)00061-0

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

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

157. Du Tao, Wang Xiuli, Wang Xifan. A combined model of wavelet and neural network for short term load forecasting. In: *Proceedings. International Conference on Power System Technology*. Vol 4. IEEE; :2331-2335. doi:10.1109/ICPST.2002.1047201

158. Chow TWS, Leung CT. Neural network based short-term load forecasting using weather compensation. *IEEE Trans Power Syst*. 1996;11(4):1736-1742. doi:10.1109/59.544636

159. Oliveira R. Combining first principles modelling and artificial neural networks: A general framework. *Comput Chem Eng*. 2004;28(5):755-766. doi:10.1016/j.compchemeng.2004.02.014

160. Lapedes A, Farber R. Nonlinear signal processing using neural networks: Prediction and system modelling. In: *IEEE International Conference on Neural Networks, San Diego, CA, USA, 21 Jun 1987*. https://www.osti.gov/servlets/purl/5470451.

161. Dayhoff JE, DeLeo JM. Artificial neural networks: Opening the black box. *Cancer*. 2001;91(8 SUPPL.):1615-1635.

162. Akkisetty PK, Lee U, Reklaitis GV, Venkatasubramanian V. Population balance model-based hybrid neural network for a pharmaceutical milling process. *J Pharm Innov*. 2010;5(4):161-168. doi:10.1007/s12247-010-9090-2

163. Lang KJ, Waibel AH, Hinton GE. A time-delay neural network architecture for isolated word recognition. *Neural Networks*. 1990;3(1):23-43. doi:10.1016/0893-6080(90)90044-L

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

165. Venkatasubramanian V. The promise of artificial intelligence in chemical engineering: Is it here, finally? *AIChE J*. 2019;65(2):466-478. doi:10.1002/aic.16489

166. Lian C, Zeng Z, Yao W, Tang H, Chen CLP. Landslide Displacement Prediction With Uncertainty Based on Neural Networks With Random Hidden Weights. *IEEE Trans Neural Networks Learn Syst*. 2016;27(12):2683-2695.

167. Khanmohammadi S, Tutun S, Kucuk Y. A New Multilevel Input Layer Artificial Neural Network for Predicting Flight Delays at JFK Airport. *Procedia - Procedia Comput Sci*. 2016;95:237-244. doi:10.1016/j.procs.2016.09.321

168. Chitsazan N, Nadiri AA, Tsai FT-C. Prediction and structural uncertainty analyses of artificial neural networks using hierarchical Bayesian model averaging. *J Hydrol*. 2015;528:52-62. doi:10.1016/j.jhydrol.2015.06.007

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

170. Tiwari MK, Chatterjee C. Uncertainty assessment and ensemble flood forecasting using bootstrap based artificial neural networks (BANNs). *J Hydrol*. 2010;382(1-4):20-33. doi:10.1016/j.jhydrol.2009.12.013

171. Bashir ZA, El-Hawary ME. Applying wavelets to short-term load forecasting using PSO-based

neural networks. *IEEE Trans Power Syst*. 2009;24(1):20-27. doi:10.1109/TPWRS.2008.2008606

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

173. Huang C-M, Yang H-T. Evolving wavelet-based networks for short-term load forecasting. *IEE Proc - Gener Transm Distrib*. 2001;148(3):222. doi:10.1049/ip-gtd:20010286

174. Sarkar D, Modak JM. ANNSA: a hybrid artificial neural network/simulated annealing algorithm for optimal control problems. *Chem Eng Sci*. 2003;58(14):3131-3142. doi:10.1016/S0009-2509(03)00168-4

175. Krishnapura VG, Jutan A. A neural adaptive controller. *Chem Eng Sci*. 2000;55(18):3803-3812. doi:10.1016/S0009-2509(00)00034-8

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

177. Watanabe K, Matsuura I, Abe M, Kubota M, Himmelblau DM. Incipient fault diagnosis of chemical processes via artificial neural networks. *AIChE J*. 1989;35(11):1803-1812. doi:10.1002/aic.690351106

178. Venkatasubramanian V, Chan K. A neural network methodology for process fault diagnosis. *AIChE J*. 1989;35(12):1993-2002. doi:10.1002/aic.690351210

179. Nandi S, Ghosh S, Tambe SS, Kulkarni BD. Artificial neural-network-assisted stochastic process optimization strategies. *AIChE J*. 2001;47(1):126-141. doi:10.1002/aic.690470113

180. Sharma H, Das G, Samanta AN. ANN–based prediction of two-phase gas– liquid flow patterns in a circular conduit. *AIChE J*. 2006;52(9):3018-3028. doi:10.1002/aic.10922

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

182. Venkatasubramanian V. DROWNING IN DATA: Informatics and modeling challenges in a data-rich networked world. *AIChE J*. 2009;55(1):2-8. doi:10.1002/aic.11756

183. Wu Z, Tran A, Ren YM, Barnes CS, Chen S, Christofides PD. Model predictive control of phthalic anhydride synthesis in a fixed-bed catalytic reactor via machine learning modeling. *Chem Eng Res Des*. 2019;145:173-183. doi:10.1016/j.cherd.2019.02.016

184. Venkatasubramanian V, Vaidyanathan R. Diagnosing noisy process data using neural networks. In: *IFAC Symposia Series*. ; 1992:547-552.

185. Ding Y, Zhang Y, Kim K, Tran A, Wu Z, Christofides PD. Microscopic modeling and optimal operation of thermal atomic layer deposition. *Chem Eng Res Des*. 2019;145:159-172. doi:10.1016/j.cherd.2019.03.004

186. Sundaram A, Ghosh P, Caruthers JM, Venkatasubramanian V. Design of fuel additives using neural networks and evolutionary algorithms. *AIChE J*. 2001;47(6):1387-1406. doi:10.1002/aic.690470615

187. Li G, Liu Z, Li J, et al. Application of general regression neural network to model a novel integrated fluidized bed gasifier. *Int J Hydrogen Energy*. 2018;43(11):5512-5521. doi:10.1016/j.ijhydene.2018.01.130

188. Ozkaya B, Demir A, Bilgili MS. Neural network prediction model for the methane fraction in biogas from field-scale landfill bioreactors. *Environ Model Softw*. 2007;22(6):815-822. doi:10.1016/j.envsoft.2006.03.004

189. Yıldız Z, Uzun H, Ceylan S, Topcu Y. Application of artificial neural networks to co-combustion of hazelnut husk–lignite coal blends. *Bioresour Technol*. 2016;200:42-47. doi:10.1016/j.biortech.2015.09.114

190. Kumar BS, Venkateswarlu C. Estimating biofilm reaction kinetics using hybrid mechanistic-neural network rate function model. *Bioresour Technol*. 2012;103(1):300-308. doi:10.1016/j.biortech.2011.10.006

191. Shaikh A, Al-Dahhan M. Development of an artificial neural network correlation for prediction of overall gas holdup in bubble column reactors. *Chem Eng Process Process Intensif*. 2003;42(8-9):599-610. doi:10.1016/S0255-2701(02)00209-X

192. Huang K, Zhan X-L, Chen F-Q, Lü D-W. Catalyst design for methane oxidative coupling by using artificial neural network and hybrid genetic algorithm. *Chem Eng Sci*. 2003;58(1):81-87. doi:10.1016/S0009-2509(02)00432-3

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

194. McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys*. 1943;5(4):115-133. doi:10.1007/BF02478259

195. Marquardt DW. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *J Soc Ind Appl Math*. 1963;11(2):431-441. https://doi.org/10.1137/0111030.

196. Levenberg K. A method for the solution of certain non-linear problems in least squares. *Q Appl Math*. 1944;2(2):164-168. https://doi.org/10.1090/qam/10666.

197. Hagan MT, Menhaj MB. Training feedforward networks with the Marquardt algorithm. *IEEE Trans Neural Networks*. 1994;5(6):989-993. doi:10.1109/72.329697

198. Borggaard C, Thodberg HH. Optimal minimal neural interpretation of spectra. *Anal Chem*. 1992;64(5):545-551. doi:10.1021/ac00029a018

199. Gandhi AB, Joshi JB, Jayaraman VK, Kulkarni BD. Development of support vector regression (SVR)-based correlation for prediction of overall gas hold-up in bubble column reactors for various gas–liquid systems. *Chem Eng Sci*. 2007;62(24):7078-7089. doi:10.1016/J.CES.2007.07.071

200. Ke SC, DeLucas LJ, Harrison JG. Computer simulation of protein crystal growth using aggregates as the growth unit. *J Phys D Appl Phys*. 1998;31(9):1064-1070.

201. Rafiei-Shishavan M, Mehta S, Ricardez-Sandoval LA. Simultaneous design and control under uncertainty : A back-off approach using power series expansions. *Comput Chem Eng*. 2017;99:66-81.

202. Vega P, Lamanna R, Revollar S, Francisco M. Integrated design and control of chemical processes - Part II: An illustrative example. *Comput Chem Eng*. 2014;71:618-635.

203. Bahakim SS, Ricardez-Sandoval LA. Simultaneous design and MPC-based control for dynamic systems under uncertainty: A stochastic approach. *Comput Chem Eng*. 2014;63:66-81.

204. Mohideen MJ, Perkins JD, Pistikopoulos EN. Optimal Design of Dynamic Systems under

Uncertainty. *AIChE J*. 1996;42(8):2251-2272.

205. Sanchez-Sanchez K, Ricardez-Sandoval LA. Simultaneous Process Synthesis and Control Design under Uncertainty: A Worst-Case Performance Approach. *AIChE J*. 2013;59(7):2497-2514.

206. Higham DJ. An introduction to multilevel Monte Carlo for option valuation. *Int J Comput Math*. 2015;92(12):2347-2360. doi:10.1080/00207160.2015.1077236

207. Ketenoğlu D, Ünal B. Influence of surface roughness on the electrical conductivity of semiconducting thin films. *Phys A Stat Mech Its Appl*. 2013;392(14):3008-3017.

208. Fuh YK, Wang CH. In situ roughness monitoring of sputtered Pt thin film under dynamic turbulence using adaptive optics. *Optik (Stuttg)*. 2014;125(9):2086-2089. doi:10.1016/j.ijleo.2013.10.056

209. Kaufholz M, Krause B, Kotapati S, et al. Monitoring the thin film formation during sputter deposition of vanadium carbide. *J Synchrotron Radiat*. 2015;22(1):76-85. doi:10.1107/S1600577514024412

210. Hasenauer J, Waldherr S, Doszczak M, Radde N, Scheurich P, Allgöwer F. Identification of models of heterogeneous cell populations from population snapshot data. *BMC Bioinformatics*. 2011;12(1):125. doi:10.1186/1471-2105-12-125

211. Hasenauer J, Waldherr S, Doszczak M, Scheurich P, Radde N, Allgöwer F. Analysis of heterogeneous cell populations: A density-based modeling and identification framework. *J Process Control*. 2011;21(10):1417-1425. doi:10.1016/j.jprocont.2011.06.020

212. Lee D, Jayaraman A, Kwon JS-I. Identification of cell-to-cell heterogeneity through systems engineering approaches. *AIChE J*. 2020;66(5):e16925. doi:10.1002/aic.16925

213. Kannan A, Ostendorf M, Karl WC, Castanon DA, Fish RK. ML parameter estimation of a multiscale stochastic process using the EM algorithm. *IEEE Trans Signal Process*. 2000;48(6):1836-1840. doi:10.1109/78.845950

214. Chou KC. Maximum-likelihood estimation of multiscale stochastic model parameters. In: *IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*. Paris, France: IEEE, 345 E 47TH ST, NEW YORK, NY 10017; 1996:17-20. doi:10.1109/TFSA.1996.546675

215. Wen C, Wang S, Wen C, Chen Z. The Multiscale Parameter Estimation Methods for a Sort of Time Series. *Chinese J Electron*. 2009;18(4):660-664.

216. Fukunaga K. Chapter 6 - Nonparametric Density Estimation. In: Fukunaga KBT-I to SPR (Second E, ed. *Introduction to Statistical Pattern Recognition (Second Edition)*. Boston: Academic Press; 1990:254-299. doi:10.1016/B978-0-08-047865-4.50012-0

Appendix A. Supplementary Material for the Robust Optimization Study in Section 3.4

Table A1: Data for the R_{PSE} model (equation (34)): roughness at the nominal values of uncertain parameters, and the sensitivities, calculated at various temperatures.

	Temperature, $\boldsymbol{T}$ (K)					
	600 K	700 K	800 K	900 K	1000 K	1100 K
$\boldsymbol{R(\mu_E, \mu_{E_m}, \mu_X)}$	92.5102	47.7179	17.5399	7.4081	3.9160	2.5437
$\boldsymbol{\partial R/\partial E}$	5.0617	9.7125	4.1002	1.4541	0.5562	0.2550
$\boldsymbol{\partial R/\partial E_m}$	1.6280	3.0911	0.8754	0.3331	0.0969	0.0626
$\boldsymbol{\partial R/\partial X}$	2.6318	1.7086	0.7459	0.0876	0.0988	0.0445
$\boldsymbol{\partial^2 R/\partial E^2}$	-1.5096	-0.0785	0.4598	0.1854	0.0440	0.0212
$\boldsymbol{\partial^2 R/\partial E_m^2}$	-0.3099	-0.0514	-0.1972	-0.0008	0.0070	0.0090
$\boldsymbol{\partial^2 R/\partial X^2}$	-0.4319	-0.6350	-0.1025	0.0035	-0.0041	0.0001
$\boldsymbol{\partial^2 R/\partial E \partial E_m}$	-0.5477	0.1933	0.2915	0.0241	0.0379	0.0131
$\boldsymbol{\partial^2 R/\partial E \partial X}$	-0.1301	0.1825	0.0058	0.0341	0.0404	0.0082
$\boldsymbol{\partial^2 R/\partial E_m \partial X}$	0.0352	-0.0455	-0.0571	-0.0097	-0.0131	0.0085

Table A2: Expansion coefficients for the R_{PCE} model (equation (35)), calculated by the Least Squares method at various temperatures.

	Temperature, $\boldsymbol{T}$ (K)					
	600 K	700 K	800 K	900 K	1000 K	1100 K
$\boldsymbol{\beta_0}$	92.2250	47.0025	16.7636	7.1764	3.8494	2.5277
$\boldsymbol{\beta_1}$	4.9884	9.8531	3.9404	1.3764	0.5807	0.2624
$\boldsymbol{\beta_2}$	1.7706	2.8549	1.1212	0.3035	0.1018	0.0514
$\boldsymbol{\beta_3}$	2.4158	1.3973	0.5136	0.1255	0.0732	0.0325
$\boldsymbol{\beta_4}$	-0.9987	0.4040	0.4479	0.1126	0.0638	0.0275
$\boldsymbol{\beta_5}$	-0.3482	0.2189	-0.0681	0.0143	-0.0007	-0.0002
$\boldsymbol{\beta_6}$	-0.1118	-0.1178	0.0367	0.0024	0.0036	-0.0015
$\boldsymbol{\beta_7}$	-0.9178	0.2940	0.3257	0.0380	0.0166	0.0025
$\boldsymbol{\beta_8}$	0.0180	0.3129	0.1354	-0.0203	0.0575	0.0102
$\boldsymbol{\beta_9}$	-0.0575	0.0635	0.1780	-0.0317	0.0093	-0.0064

Table A3: Expansion coefficients for the R_{PCE} model (equation (35)), calculated by the NISP method at various temperatures.

	Temperature, $\boldsymbol{T}$ (K)					
	600 K	700 K	800 K	900 K	1000 K	1100 K
$\boldsymbol{\beta_0}$	92.5102	47.7179	17.5399	7.4081	3.9160	2.5437
$\boldsymbol{\beta_1}$	5.0617	9.7125	4.1002	1.4541	0.5562	0.2550
$\boldsymbol{\beta_2}$	1.6280	3.0911	0.8754	0.3331	0.0969	0.0626
$\boldsymbol{\beta_3}$	2.6318	1.7086	0.7459	0.0876	0.0988	0.0445
$\boldsymbol{\beta_4}$	-1.5096	-0.0785	0.4598	0.1854	0.0440	0.0212
$\boldsymbol{\beta_5}$	-0.3099	-0.0514	-0.1972	-0.0008	0.0070	0.0090
$\boldsymbol{\beta_6}$	-0.4319	-0.6350	-0.1025	0.0035	-0.0041	0.0001
$\boldsymbol{\beta_7}$	-0.5477	0.1933	0.2915	0.0241	0.0379	0.0131
$\boldsymbol{\beta_8}$	-0.1301	0.1825	0.0058	0.0341	0.0404	0.0082
$\boldsymbol{\beta_9}$	0.0352	-0.0455	-0.0571	-0.0097	-0.0131	0.0085

Table A4: Data for the Th_{PSE} model (equation (36)): thickness at the nominal values of uncertain parameters, and the sensitivities, calculated at various temperatures.

	Temperature, $\boldsymbol{T}$ (K)					
	600 K	700 K	800 K	900 K	1000 K	1100 K
$\boldsymbol{Th(\mu_E, \mu_{E_m}, \mu_X)}$	1938.9827	1865.1597	1800.8498	1740.3520	1678.6961	1605.4975
$\boldsymbol{10^4(\partial Th/\partial E)}$	0.8677	0.9574	6.0272	20.1923	58.2664	151.2352
$\boldsymbol{10^4(\partial Th/\partial E_m)}$	0.7155	-2.2469	-4.4035	-17.4068	-50.3718	-112.2404
$\boldsymbol{10^{-7}(\partial Th/\partial X)}$	96.8234	93.1537	89.9285	87.0212	84.3023	81.2821

Table A5: Expansion coefficients for the Th_{PCE} model (equation (37)), calculated by the Least Squares method at various temperatures.

	Temperature, $\boldsymbol{T}$ (K)					
	600 K	700 K	800 K	900 K	1000 K	1100 K
$\boldsymbol{\beta_0}$	1938.4198	1864.9606	1800.3728	1740.2908	1678.1673	1604.1333
$\boldsymbol{\beta_1}$	-0.0031	0.0972	0.2916	1.4520	3.9306	8.9868
$\boldsymbol{\beta_2}$	0.0292	-0.0437	-0.0487	-0.4156	-1.2233	-2.4541
$\boldsymbol{\beta_3}$	96.8002	93.0870	89.9020	87.1028	84.3194	81.3022

Table A6: Expansion coefficients for the Th_{PCE} model (equation (37)), calculated by the NISP method at various temperatures.

	Temperature, $\boldsymbol{T}$ (K)					
	600 K	700 K	800 K	900 K	1000 K	1100 K
$\boldsymbol{\beta_0}$	1938.3899	1864.9083	1800.4599	1740.3914	1678.4797	1605.2348
$\boldsymbol{\beta_1}$	0.2607	-0.0013	0.3555	1.5475	3.7608	9.4523
$\boldsymbol{\beta_2}$	-0.0523	-0.1731	-0.1319	-0.2733	-1.3659	-2.8026
$\boldsymbol{\beta_3}$	96.9351	93.2996	89.8683	87.1507	84.3482	81.1791

Table A7: Polynomial models that were used to fit R_{PSE} model coefficients (equation (34)) to temperature.

$\omega(T)$	γ_0	γ_1	γ_2	γ_3
$R(\mu_E, \mu_{E_m}, \mu_X)$	1.2010E+03	-3.4574E+00	3.3304E-03	-1.0706E-06
$\partial R/\partial E$	4.6214E-02	-4.6256E-05	-3.4773E-08	3.5411E-11
$\partial R/\partial E_m$	2.5767E-01	-7.2410E-04	6.7986E-07	-2.1305E-10
$\partial R/\partial X$	6.8642E+08	-2.1143E+06	2.1686E+03	-7.3960E-01
$\partial^2 R/\partial E^2$	6.4850E-05	-1.8156E-07	1.7049E-10	-5.3614E-14
$\partial^2 R/\partial E_m^2$	1.1660E-04	-3.4570E-07	3.4126E-10	-1.1214E-13
$\partial^2 R/\partial X^2$	3.1455E+14	-1.0869E+12	1.2189E+09	-4.4632E+05
$\partial^2 R/\partial E \partial E_m$	-4.6273E-06	3.1129E-08	-4.5408E-11	1.9064E-14
$\partial^2 R/\partial E \partial X$	9.8003E+04	-2.7842E+02	2.6515E-01	-8.4488E-05
$\partial^2 R/\partial E_m \partial X$	1.5682E+05	-4.5883E+02	4.4764E-01	-1.4554E-04

Table A8: Polynomial models that were used to fit R_{PCE} model coefficients (equation (35)) to temperature. The PCE coefficients were calculated by the Least Squares method.

$\omega(T)$	γ_0	γ_1	γ_2	γ_3
β_0	1.1475E+03	-3.2793E+00	3.1358E-03	-1.0006E-06
β_1	2.6371E+02	-7.6244E-01	7.3860E-04	-2.3928E-07
β_2	7.3837E+01	-2.1054E-01	2.0054E-04	-6.3738E-08
β_3	4.2851E+01	-1.2578E-01	1.2331E-04	-4.0324E-08
β_4	4.6319E+01	-1.3820E-01	1.3760E-04	-4.5657E-08
β_5	-1.7766E+01	5.4587E-02	-5.5617E-05	1.8796E-08
β_6	6.5862E+00	-2.0160E-02	2.0522E-05	-6.9447E-09
β_7	4.3286E+01	-1.2995E-01	1.2984E-04	-4.3156E-08
β_8	5.2842E+01	-1.6611E-01	1.7312E-04	-5.9792E-08
β_9	4.7772E+01	-1.4740E-01	1.5088E-04	-5.1238E-08

Table A9: Polynomial models that were used to fit R_{PCE} model coefficients (equation (35)) to temperature. The PCE coefficients were calculated by the NISP method.

$\omega(T)$	γ_0	γ_1	γ_2	γ_3
β_0	1.1163E+03	-3.1650E+00	3.0030E-03	-9.5086E-07
β_1	2.3114E+02	-6.5464E-01	6.2152E-04	-1.9747E-07
β_2	1.1904E+02	-3.6069E-01	3.6446E-04	-1.2263E-07
β_3	3.9673E+01	-1.0980E-01	1.0099E-04	-3.0826E-08
β_4	9.1414E+00	-1.9762E-02	1.3028E-05	-2.3642E-09
β_5	-3.0509E+01	9.1772E-02	-9.1740E-05	3.0484E-08
β_6	-2.0067E+01	6.1225E-02	-6.2037E-05	2.0874E-08
β_7	5.0958E+01	-1.5565E-01	1.5807E-04	-5.3333E-08
β_8	9.6069E-01	-4.4839E-03	6.3096E-06	-2.7459E-09
β_9	-1.1345E+01	3.5312E-02	-3.6593E-05	1.2613E-08

Table A10: Polynomial models that were used to fit Th_{PSE} model coefficients (equation (36)) to temperature.

$\omega(T)$	γ_0	γ_1	γ_2	γ_3
$Th(\mu_E, \mu_{E_m}, \mu_X)$	2.3271E+03	-6.5352E-01	0	0
$\partial Th/\partial E$	-1.0048E-01	4.2207E-04	-5.8770E-07	2.7224E-10
$\partial Th/\partial E_m$	5.5743E-02	-2.3859E-04	3.4070E-07	-1.6286E-10
$\partial Th/\partial X$	1.1478E+09	-3.0619E+05	0	0

Table A11: Polynomial models that were used to fit Th_{PCE} model coefficients (equation (37)) to temperature. The PCE coefficients were calculated by the Least Squares method.

$\omega(T)$	γ_0	γ_1	γ_2	γ_3
β_0	2.3277E+03	-6.5483E-01	0	0
β_1	-4.0804E+01	1.7762E-01	-2.5758E-04	1.2476E-07
β_2	6.3642E+00	-3.0104E-02	4.7482E-05	-2.4923E-08
β_3	1.1464E+02	-3.0455E-02	0	0

Table A12: Polynomial models that were used to fit Th_{PCE} model coefficients (equation (37)) to temperature. The PCE coefficients were calculated by the NISP method.

$\omega(T)$	γ_0	γ_1	γ_2	γ_3
β_0	2.3263E+03	-6.5289E-01	0	0
β_1	-4.4144E+01	1.9497E-01	-2.8432E-04	1.3754E-07
β_2	1.5260E+01	-6.6049E-02	9.4340E-05	-4.4779E-08
β_3	1.1511E+02	-3.0958E-02	0	0

Appendix B. Parameter Definitions for the Catalytic Pore Model

Table B1: Catalytic pore model parameter values (adapted from Chaffart, D., & Ricardez-Sandoval, L. A. (2017). Robust dynamic optimization in heterogeneous multiscale catalytic flow reactors using polynomial chaos expansion. Journal of Process Control, 60, 128–140. https://doi.org/10.1016/j.jprocont.2017.07.002

¹⁰¹⁰¹⁰¹⁰¹⁰

Parameter	Symbol	Value and Unit
Pore length	$\mathcal{L}$	1 µm
Pore radius	ρ	50 nm
Fluid velocity	v	0.01 m/s
Diffusion coefficient for species *i*	D_i	5.3×10^{-9} m^2/s
Initial concentration, species *A*	$C_{A,in}$	241 mmol/L
Initial concentration, species *B*	$C_{B,in}$	241 mmol/L
Initial concentration, species *C*	$C_{C,in}$	0 M
Surface model adsorption constant, species *A*	$k_{a,A}$	1 molecule/(M·site·s)
Surface model adsorption constant, species *B*	$k_{a,B}$	1 molecule/(M·site·s)
Surface model desorption constant, species *A*	$k_{d,A}$	1×10^{10} molecules/(site·s)
Surface model desorption constant, species *B*	$k_{d,B}$	1×10^{10} molecules/(site·s)
Activation energy of desorption, species *A*	$E_{d,A}$	95,000 J/mol
Activation energy of desorption, species *B*	$E_{d,B}$	100,000 J/mol
Surface model reaction constant	κ	2×10^{6} molecules/(site·s)
Activation energy of reaction	E_r	57,000 J/mol
Temperature	T	460 K
Number of catalyst sites per lattice	N_{total}	900 sites
Number of chemical species	$n_\tau(x,\rho)$	3 species