

Dynamic Surrogate Modelling for Multistep-Ahead Prediction of Multivariate Nonlinear Chemical Processes

Ahmed Shokry^{1, 2*}, Piero Baraldi³, Enrico Zio^{3,4,5} Antonio Espuña²

¹Center for Applied Mathematics, Ecole Polytechnique, Route de Saclay, 91120 Palaiseau, France.

²Department of Chemical Engineering, Universitat Politècnica de Catalunya, EEBE – Eduard Maristany, 14, 08019 Barcelona, Spain.

³Energy Department, Politecnico di Milano, Via Lambruschini 4, 20156, Milan, Italy.

⁴Eminent Scholar, Department of Nuclear Engineering, College of Engineering, Kyung Hee University, Republic of Korea.

⁵MINES ParisTech, PSL Research University, CRC, Sophia Antipolis, France.

*ahmed.shokry@polytechnique.edu

Abstract

This work proposes a methodology for multivariate dynamic modeling and multistep-ahead prediction of nonlinear systems using surrogate models for the application to nonlinear chemical processes. The methodology provides a systematic and robust procedure for the development of data-driven dynamic models capable of predicting the process outputs over long time horizons. It is based on using surrogate models to construct several Nonlinear AutoRegressive eXogenous models (NARX), each one approximating the future behavior of one process output as a function of the current and previous process inputs and outputs. The developed dynamic models are employed in a recursive schema to predict the process future outputs over several time steps (multistep-ahead prediction). The methodology is able to manage two different scenarios: 1) one in which a set of input-output signals collected from the process is only available for training, and 2) another in which a mathematical model of the process is available and can be used to generate specific datasets for training. With respect to the latter, the proposed methodology includes a specific procedure for the selection of training data in dynamic modeling based on Design Of Computer Experiment (DOCE) techniques. The proposed methodology is applied to case studies from the process industry presented in the literature. The results show very high prediction accuracies over long time horizons. Also, thanks to the flexibility, robustness and computational efficiency of surrogate modeling, the methodology allows dealing with a wide range of situations, which would be difficult to address using first principle models.

Keywords: *Multivariate dynamic modeling- Multistep ahead prediction- Complex Nonlinear Chemical processes- Nonlinear AutoRegressive eXogenous models - Surrogate Models- Artificial Neural Networks - Gaussian models*

Highlights:

- The work presents an efficient methodology for developing data-driven multivariate dynamic models of nonlinear chemical processes.
- The methodology is based on fitting several NARX models, and their use in a recursive scheme for multi-step ahead prediction.
- A new procedure for the selection of training data for dynamic modeling is proposed based on DOCE.
- The methodology is shown able to predict the dynamic multivariate outputs of nonlinear processes over long time horizons in a robust, accurate and computationally efficient way.
- The methodology is shown flexible and can be easily applied in situations where conventional first principle models are difficult to use or not available.

1 Introduction

List of Abbreviations	
ANN	Artificial Neural Network
DOCE	Design Of Computer Experiment
DOE	Design Of Experiment
ESO	Ensemble of Single-Output
FDD	Fault Detection and Diagnosis
FPM	First Principle Model
GP	Gaussian Process
MLP ANN	Multi-Layer Perceptron ANN
MPC	Model Predictive Control
MO	Multi-Output
NARX	Nonlinear AutoRegressive eXogenous
NRMSE	Normalized Root Mean Square Error
OK	Ordinary Kriging
RMSE	Root Mean Square Error
RTO	Real Time Optimization
SVR	Support Vector Regression

List of Main Mathematical Symbols

1) Ordinary Kriging	
x_i	$i - th$ training input sample
w_i	$i - th$ training output sample
n	Number of training samples
k	Number of input variables of the surrogate model
$[X]_{n \times k}$	Matrix of training inputs samples
$[W]_{n \times 1}$	Vector of training output samples
\hat{w}	The estimated output by the surrogate model
μ_{ok}	Mean or main trend of the Ordinary Kriging
$Z(x)$	Stochastic Gaussian Process residual
σ_{ok}^2	Process variance
$R(x_i, x_j)$	Correlation function between the $i - th$ and the $j - th$ training input samples
ξ_l	Ordinary Kriging hyperparameters
δ_{ij}	The Kronecker delta
p_l	Ordinary Kriging smoothing parameters
λ	Regularization constant
Lik	Likelihood function of the observed output
$[R]_{n \times n}$	Correlation matrix between the training input samples
(x^*, w^*)	A new interpolating point, not in the training set
2) Dynamic modelling based on surrogate models	
U	The process control input(s)
D_u	Number of the process control inputs
Y	The process output(s)
D_y	Number of the process outputs
t_0	First time instance
t_f	Final time instance
t_i	$i - th$ time instance
$\Delta t = (t_i - t_{i-1})$	Sampling period of the process variables
U_i	Value(s) of the process control input(s) at the time instance t_i
Y_i	Value(s) of the process output(s) at the time instance t_i
\mathbb{F}	Multi-output emulator/black-box model
$\mathbb{f}_1, \dots, \mathbb{f}_q$	Set of q single-output emulators/black-box models
f	Recursive single-step emulator
q	The required prediction horizon in terms of the number of time steps-ahead
$\hat{Y}_{t+1}, \hat{Y}_{t+2}, \dots, \hat{Y}_{t+q}$	The predicted process outputs q steps-ahead
L	Time lag
$\hat{y}_{j,t+1}$	The predicted value of the j -th process output by the j -th recursive single-step emulator, f_j , at the next time step $t + 1$
$U_{t \min} : U_{t \max}$	Allowable control limits
$[Y_{t \min} : Y_{t \max}]$	Variation domain of the process outputs
3) Bioreactor	
C_m	Concentrations of the microorganisms inside the reactor
C_s	Concentrations of the substrate inside the reactor
U	Reactor outlet flowrate

4) Three-tanks system	
A	Cross section area of the tanks
a_1, a_3 and a_0	Flow coefficients
g	Gravitational acceleration
h_1, h_2 and h_3	Liquid levels in tank 1, tank 2 and tank 3, respectively
Q_1 and Q_2	Inlet flowrates to tank 1 and tank 2, respectively
s_{13}, s_{23} and s_0	Cross section area of connecting pipes
sgn	Sign function
5) Oil Shale Pyrolysis	
C_{Kr}	Concentration of kerogen
C_{pb}	Concentration of pyrolytic bitumen
C_{og}	Concentration of oil and gas
C_{Cr}	Concentration of the organic carbon residue
k_i	The i -th reaction rate
k_{i0}	Initial value of the i -th reaction rate
E_i	The activation energy
R	The gas constant
T	Manipulated temperature

In process engineering area, a reliable dynamic model of the process is necessary for its optimal operation, control and management. In particular, a dynamic model able to accurately predict the future values of the process outputs in reasonable computational times is the base of most online applications, e.g. Real Time Optimization (RTO), Model Predictive Control (MPC), Dynamic Data Reconciliations, Fault Detection and Diagnosis.

Although analytical models (hereafter also called “First Principle Models” – FPMs) are available to describe the dynamics of many chemical processes, practical limitations often hinder their usage, especially in applications, such as RTO and MPC, which require the online repetitive solution of an optimization problem which, in itself, requires the evaluation of the model several times [1, 2]. This may result in an unaffordable computational effort, especially for large-scale or fast dynamic systems [3], due to the complexity of the solution procedure –e.g. iterative schemes and/or integration techniques- used to solve such mathematical models [4, 5].

Furthermore, the available FPMs are often developed under the assumption of favorable (ideal) working conditions, which are typically not encountered at the industrial scale, that is characterized by uncontrolled disturbances, different operating conditions, continuously varying parameters (e.g. heat transfer coefficients) and, possibly, different units/reactors geometries, etc. [6, 7]. Also, since process FPMs typically do not take into account the physical characteristics of mechanical and electrical components, connections and piping, which remarkably influence the real process, the accuracy of the

FPMs predictions are reduced [8, 9]. In other cases, the development of a detailed analytical FPM is conceptually difficult or even unaffordable, due to the limited knowledge about the nonlinear behaviors and complex phenomena characterizing the process, such as reaction kinetics, thermodynamic relationships, heat and mass transfer, etc. [10, 8]. In these situation, on another hand, real data collected from the process are available, but there is no support of a well-founded conceptual/mathematical model for describing the process based on first principles [11, 12, 13].

In all these cases, system identification or data-driven dynamic modeling methods can be used to construct empirical dynamic models for predicting the future values of the process outputs [11]. Many methods have been developed for linear dynamic system modelling, but their application to nonlinear processes provides unsatisfactory results [1]. This is due to the fact that linear approximations severely simplify the nonlinear behavior of the process, resulting in poor prediction accuracy [1, 14]. Advanced data-driven nonlinear modelling techniques, such as Artificial Neural Networks (ANNs) (e.g. Radial Basis-ANNs, Recurrent-ANNs etc.) [15, 16], Fuzzy models [11], Neuro-fuzzy models [17] and recently Gaussian Process (GP) models [18, 19], have been widely proposed to capture nonlinear dynamic relations between the nonlinear process inputs and outputs. These techniques, which are also referred to as metamodels or surrogate models, establish nonlinear relationships between inputs and outputs variables, using input-output training data, which can be either generated from complex FPM simulations or measured from the real process [7].

1.1 Review on Data-driven dynamic modelling in chemical processes

ANNs have become a popular choice for nonlinear dynamic modeling and identification [16, 20, 21], due to their universal approximation abilities [14, 22]. Although they exhibit very powerful capabilities, their usage has two main practical drawbacks: *i*) large effort is required to select a good network structure (numbers of layers and the included neurons) and configurations (type of activation function, training algorithm, cost/error function, etc.) [7], and *ii*) the curse of dimensionality, i.e. the increase of the number of inputs causes the growth of the number of the ANN neurons, and consequently, of the number of parameters (weights and biases) to be set: then, the quantity of data needed for training the ANN grows exponentially with the number of inputs [3].

Although different algorithms have been developed to automatically select ANNs structures and configurations [22], their application requires additional computational effort, since they solve a complex optimization problem, in which the network configuration and its parameters are treated as decision variables to be tuned to minimize an objective associated to the output prediction error [23, 24, 25]. As a result, their application to cases involving high dimensional systems, large-scale databases and/or online fitting and updating has been quite limited.

In spite of these difficulties, a significant number of successful applications of ANNs for dynamic modelling are reported over a wide spectrum of fields [11, 26, 20, 27]. Especially in the process engineering area, ANNs have been extensively used as Nonlinear AutoRegressive eXogenous (NARX) models for dynamic system identification of both univariate (single output) [28, 1, 29, 30, 31, 21] and multivariate (multi-output) problems [16, 32, 17, 33, 14, 34]. In the literature, multivariate systems are usually approximated either using a multi-output ANN model or an ensemble of single-output ANNs models, where, in the latter case, a set of independent single-output ANN models, each approximating one output as a function of the inputs, is built.

On the other hand, Gaussian Process (GP) models have been proposed in the Bayesian inference area by O'Hagan et al. [35, 36] for the approximation of complex static computer codes, representing a generic class of non-parametric probabilistic models. GP models have shown promising accuracy and ability to reduce the previously mentioned problems of ANNs [3, 37, 38]. This is due to their nonparametric nature: they do not approximate the system by fitting the parameters of a selected structure or functional shape but, instead, they search for relationships among the measured data through a correlation function/model. Therefore, the number of the metamodel parameters to be identified is significantly low compared to other parametric models (e.g. ANNs models) and, consequently, the size of the required set of training data is significantly reduced, too [3]. Besides, GP models offer high approximation accuracy, tuning flexibility and ability to estimate a measure of uncertainty about the prediction in the form of prediction error or variance [12, 39].

Thanks to the pioneer works of Murray-Smith, et al. [40], Kocijan, et al. [41] Girard, et al. [42], and Rasmussen & Deisenroth [43], among others, GP models have gained a wide popularity for dynamic modeling and identification of nonlinear systems, and shown performances comparable and competitive to other state-of-art techniques. The main limitation of the GP models is the large computational cost for optimizing/fitting their parameters, especially when considering large amount of training data and/or addressing high dimensional systems [3, 9]. With respect to the problem of performing multi-step ahead predictions, some works have been able to successfully propagate the GP estimated error when it is used in recursive prediction [42]. But, again, the computational cost associated to the uncertainty propagation is still significant.

The Ordinary Kriging (OK) techniques can be considered specific form of GP models [12] and share similar advantages, such as accurate approximation capabilities, required small number of training data, flexible tuning of the model parameters [44] and ability of estimating a prediction error. Also, alike to the GP model, OK suffers from the high computational training effort. Thanks to the works of Davis and Ierapetritou [4] and Caballero and Grossmann [2], the OK surrogate models has been introduced to the chemical process engineering area and, since this time, it is attracting increasing

attention for surrogate-based optimization and analysis of complex nonlinear static processes [7, 45, 46, 47].

Nowadays, the GP and OK models have been proposed for univariate dynamic modeling of nonlinear chemical processes [3, 18], where they are employed as NARX models to estimate the future value - over one step-ahead - of an output of interest, as a function of the process current inputs and output values. The developed model is, then, used to perform multistep-ahead prediction via recursive calculation, where the predicted output at the current time is fed-back to the model as a part of its input for the next prediction step.

To the best of the authors' knowledge, few works have extended the GP and OK capabilities to multivariate dynamic modeling of chemical processes: Hernandez and Grover [48] developed a method for multivariate dynamic modeling based on a set of GP models, each one representing a discrete-time state space model predicting the time evolution of one process output; they also proposed a sequential sampling technique to select the training data to be used for training the GP-based dynamic models; the method was successfully applied to approximate a stochastic zero-input/multi-output dynamic model describing nanoparticle size evolution. In an area more related to process and system engineering, Boukouvala et al. [12] proposed a similar approach based on a set of Kriging metamodels, each one predicting the future values of one process output through recursive prediction over several time steps, and applied it to the simulation of a powder-roller-compaction pharmaceutical process. The approach has shown good accuracy in the identification of the dynamic behavior of the process outputs (ribbons density and roll gap) that are influenced by three control inputs (roll speed, roll pressure, feed speed); they proposed the use of a full factorial design for selecting the initial training dataset, and a sequential procedure to update the trained models during their online usage by adding to the initial training set the predicted instances for which the summation of the OKs estimated variances/errors was lower than a specific threshold.

However, these two works share some common limitation: 1) they have been validated considering processes characterized by very smooth/steady dynamics, without any influencing control/external inputs [48] or with very simple changes in them [12], 2) both works provided simple Markovian state-space models and they have not illustrated the ability of their methodologies to develop dynamic models with delayed/lagged inputs, 3) they presumed that a FPM is always available, which is combined with DOCE methods to produce optimized data for training, and 4) the robustness of their methodologies to handle different cases studies, and their flexibility to integrate different metamodel types are not explored. Finally, the methodology proposed by Boukouvala et al. [12] has not been proven to provide one compact set of models able to simulate the future behavior of the system outputs corresponding to simultaneous changes in the process inputs, since, in this method, a new set of dynamic models should be fitted several times, each time to approximate the system behavior

corresponding to a simple step change in one of the control input variables, keeping the rest of the control input variables fixed.

The aforementioned limitations obstacle the use of these methodologies for the dynamic modelling of real processes or systems, where remarkable challenges are posed: *i*) in real processes, many external inputs exist, which control or disturb the process causing significant changes in its outputs behavior, *ii*) incorporating lags or delays in the model inputs is a basic requirement in data-driven dynamic modeling, in order to capture the possible delayed behavior of the process itself and/or to compensate for missing repressors of the model [49, 50], *iii*) in many practical situations, data collected from the process can be the only source of information available (i.e. no FPM).

Shokry and Espuña [51] and Shokry et al. [52] proposed a multivariate dynamic modeling methodology in which OK models have been used as NARX models. The method was successfully applied to model different batch processes for the purpose of dynamic optimization [51] and continuous processes [52] for the purpose of Fault Detection and Diagnosis (FDD). Shokry et al. [53] successfully applied the same methodology to the univariate dynamic modelling of a real batch process operated under different initial conditions and involving missing measurements. The obtained model is used as a dynamic observer for the online supervision of the process and the detection of possible faults. These works focused on the specific application (e.g., open loop optimal control and FDD) for which these dynamic models were constructed, and lacked of generality and robustness with respect to the considered metamodels type.

More recently, Bradford et al., 2018 presented a similar method for multivariate dynamic modeling that relies on a set of GP-based NARX models. The method was applied to model the multivariate behavior of a real Algal lutein production batch process that involves two control inputs and three process outputs. Although the method provided good prediction accuracy, the addressed case study is characterized by simple dynamics, since one control input is kept constant in all the different batches, while the second is allowed to vary from one batch to another, but its value within the same batch is kept constant. Hence, practically, the control inputs became constant parameters and, consequently, the set of dynamic models are validated by predicting the simple behavior of zero-input batches. Also, when the validated set of GP dynamic models is further used for dynamic optimization, the predicted optimal “offline” profiles of the process outputs are not compared to those of the real batch system.

This work extends and generalizes the works presented in [51, 52, 53] by presenting a generic multivariate dynamic modeling and multi-step ahead prediction methodology. The methodology is based on training a set of OK-based NARX models; each model predicts the upcoming value of one process output over a constant time step as a function of the preceding values of the process inputs and outputs, over a suitable time lag. The obtained models represent discrete state-space models (also

called single-step or one-step ahead simulators) that mimic the incremental evolution of the process outputs. The trained dynamic models interact through a recursive scheme to predict the system outputs over several time steps (multistep-ahead prediction),

The main contributions of this work are:

- 1) the development of a novel, generic and robust methodology for multivariate dynamic modeling and multi-step ahead prediction of complex nonlinear chemical processes using surrogate models. The properties of generality and robustness are fundamental in order to address the main limitations currently attributed to the existing approaches in terms of a) the ability to provide accurate data-driven dynamic models for general multi-input/multi-output processes that may involve complex dynamic behaviors (complex control input profiles, delayed behaviors, etc.), b) the ability to simulate the process future outputs over large time horizons, c) the capability to accommodate different types of data modeling techniques and d) the ability of handling different situations, either when a limited set of input-output data signals are available, or when the training data can be optimally generated using a FPM and design for computer experiment techniques.
- 2) the introduction of the use of OK models for the multivariate dynamic modeling in the chemical process field in a robust and flexible manner, and the comparison of its capabilities with most popular techniques (i.e. ANNs).
- 3) the development of a novel Design Of Computer Experiments procedure for dynamic modeling, considering the purpose of the simplification and complexity reduction of expensive dynamic FPMs.

The rest of the paper is structured as follows. Section 2 gives a general view over the considered DOCE and surrogate modeling techniques (i.e.: OK and ANN), including their mathematical/statistical basis and implementations details. Section 3 presents the proposed dynamic metamodeling method, and the new procedure proposed for the design of computer experiments in the case of dynamic modelling. Section 4 shows the method application to three different case studies (different natures, i.e. continuous and batch and, different areas, i.e. biochemical, industrial and petrochemical) and discusses the obtained results. Finally, Section 5 concludes the work, stresses its advantages and discusses its limitations, which would be further investigated in future works.

2 Surrogate models building techniques

Surrogate models are data-driven techniques which are used to build empirical relations describing the mapping between input and response variable(s) [44, 54]. Although this definition can involve a very

wide range of data-based models, including the simplest types (e.g.: linear or polynomial regressions), the term is usually associated to nonlinear multivariate models, like ANNs, GPs, OK, Support Vector Regression (SVR), etc. [55]. Surrogate models can be trained using real data collected by sensors from the physical systems or using simulation data generated from a complex FPMs, for the purpose of its simplification. The following subsections review most common DOCE techniques used for training data selection in cases where a FPM is available, highlight the basics of the two common nonlinear data-driven modeling techniques, namely OK and ANNs, which have been used in this work, and review basics of common DOCE methods.

2.1 Design Of Computer Experiments (DOCE)

Design Of Computer Experiments (DOCE) techniques [56] aim at selecting the best combinations of the input variables values -within specific domain or bounds- that can be used for the simulation of the complex FPM providing the most representative information/knowledge about the output behavior [57]. The set of combinations of the input variables values is called “sampling plan”, $[X]_n$, where n is the number of sample points or instances. Since the main objective is to collect as much information as possible about the output behavior over all the local sub-regions in the input domain, and most computer simulation models are deterministic, the DOCE techniques consider samples selection criteria different than those of the DOE. These criteria are, mainly, the space-fillingness and the stratification of the sampling plan [44, 57], and both lead to increasing the uniformity of the sampling plan over all the local sub-regions of the input domain to be covered.

Many DOCE techniques have been developed basically for static surrogate modeling. The most common techniques include Latin hypercube sampling [44], low discrepancy sequences as the Hammersley technique [58] and space-filling designs like max-min techniques and space-filling Latin hypercube design [59]. Alternatively, sequential or adaptive sampling are special type of DOCE techniques that are typically related to the use of OK/GP models [7]. In these sequential techniques, the total number of training points are not selected at once: the surrogate model is initially fitted with a relatively small number of training points, and it is iteratively updated by adding new training points of interest (infill or update points) to the initial training dataset so as to enhance a desired criterion of the metamodel performance, which is highly dependent on the eventual use of the metamodel (global approximation, surrogate-based optimization, etc.); then, the metamodel is refitted. The iterative procedure stops when the metamodel performance criterion reaches a satisfactory level or tolerance [44, 56].

Most of these DOCE techniques show both desired and limiting characteristics in terms of the uniformity of the generated sampling plan and demanded computational cost. For example, Latin hypercube and low discrepancy sequence designs provide sampling plans of good uniformity with

very low computational cost [58]. Space-filling designs are able to provide sampling plans with very high uniformity, although the associated computational cost is relatively high [2, 59]. Because, these techniques usually encompass a complex optimization problem [55], in which the locations of the input combinations (i.e., instances) within the k -dimensional input space of the model are manipulated to maximize a certain space-filling criterion [44]. Sequential sampling designs have also shown very high uniformity and high efficiency, since they take maximum advantages of each point in the sampling plan, but the computational cost of such techniques is extremely high, since each iteration involves an optimization problem seeking for the point that optimally enhances the metamodel accuracy, in addition to the subsequent fitting of the metamodel with the updated training set [60]. For the previous reasons, the iterative DOCE procedures are favorable when dealing with very expensive FPM (i.e., computational fluid dynamic models) where the cost of one simulation run using the FPM is much higher than one iteration of the sequential sampling procedure. A more detailed analysis about the different DOCE techniques can be found in [57, 58]

This work considers the Hammersley design technique, due to its ability to provide sampling plans of good uniformity and stratification properties with very low computational cost [57, 58]. In each case, the optimal selection of the number of sample points (n) required to capture the output behavior depends on the input dimensionality of the surrogate model (k), the size of the input space and, also, on the intricacy and nonlinearity of the considered output behavior. In general, as n increases, the effort (time/cost) required not only for executing the experiments, but also to design the sampling plan and for the surrogate model fitting increases. Then, the modeler should carefully balance the trade-offs between the required surrogate model accuracy, the computational cost and the eventual application benefits of the surrogate model.

2.2 Ordinary Kriging (OK)

Given a set of n input-output training data $[x_i, w_i]$, $i=1,2,..,n$, $x \in R^k, w \in R$, the OK assumes the predictor $\hat{w}(x) = \mu_{ok} + Z(x)$, where the constant term μ_{ok} represents the main trend of the system to be approximated, and $Z(x)$ is a deviation/residual from that trend, which accounts for the detailed complex behavior of the system that could not be captured via the main trend μ_{ok} . The residual $Z(x)$ is modeled as a stochastic Gaussian process with expected value $E(Z(x)) = 0$ and covariance between two residuals $cov(Z(x_i), Z(x_j))$ that only depends on their corresponding inputs x_i, x_j . Thus, it can be calculated as: $cov(Z(x_i), Z(x_j)) = \sigma_{ok}^2 R(x_i, x_j)$, being σ_{ok}^2 the process variance and $R(x_i, x_j)$ a correlation function, $R(x_i, x_j) = \exp\left(-\sum_{l=1}^k \xi_l |x_{i,l} - x_{j,l}|^{p_l}\right) + \delta_{ij} \lambda$, where, ξ_l are the model hyper-parameters, δ_{ij} is the Kronecker delta, p_l are smoothing parameters and λ is a regularization constant [61].

The parameters $[\mu_{ok}, \sigma_{ok}^2, \xi_l, p_l, \lambda]$ are estimated via maximizing the likelihood function of the observed output data $[W]_{n \times 1}$. Then, the OK final predictor and estimated error are obtained in Eq. (1)

and in Eq. (2), respectively. Where x^* is a new interpolating point (i.e. not in the training data), $[r]_{n \times 1}$ is the vector of correlations between the new point to be predicted x^* and the original training data points, and calculated as $R(x_i, x^*)$, $[R]_{n \times n}$ is the correlation matrix between the training inputs and $[1]_{n \times 1}$ is the identity vector [60, 2].

$$\widehat{w}(x^*) = \mu_{ok} + r^T R^{-1} (W - \mathbf{1} \mu_{ok}) \quad (1)$$

$$\widehat{s}^2(x^*) = \sigma_{ok}^2 (1 + \lambda - r^T R^{-1} r + (1 - \mathbf{1}^T R^{-1} r)^{-1} / (\mathbf{1}^T R^{-1} \mathbf{1})) \quad (2)$$

The OK implementation proposed by Forrester et al. [44] is considered in this paper due to its high effectiveness, generality and ease of implementation. The “*fmincon*” algorithm of the Matlab optimization toolbox library is used for tuning the OK parameters by maximizing the concentrated likelihood function. The parameters p_i are usually set to value of 2, which guarantees smooth differentiable correlation functions [44].

2.3 Artificial Neural Networks (ANNs)

An ANN is a lattice of nodes, termed as neurons, which are placed in this lattice through a certain number of layers, and are fully interlinked together to be capable of the nonlinear processing of the information. A weight value is assigned to each link connecting each couple of neurons in successive layers; additionally, a bias is considered as an independent input to each neuron. The output of each neuron in a hidden layer is computed as the weighted sum of its inputs received from the neurons in the previous layer plus the bias. The computed value is, then, processed by a transfer function and is sent to the next layer in the network, and so on until the output layer [26, 1]. The training of the ANN is accomplished by solving a nonlinear optimization problem, in which an objective function (related to the errors between the predicted outputs by the network and the target outputs of the training dataset) is minimized by tuning the optimization variables values represented in the weights and the biases of the neurons [26, 1].

Amongst the various kinds of ANNs, we consider the Multi-Layer Perceptron ANN (MLP-ANN), which is the most popular kind used in engineering practice, as it offers high efficiency, accuracy and straightforward applicability [55, 1]. The “*feedforwardnet*” function of the Matlab ANN toolbox is used in this work to build MLP-ANN; a trial and error technique is employed for selecting the suitable number of layers and neurons and the training algorithm achieving a compromise between the structure simplicity and the prediction accuracy.

It deserves highlighting that this paper does not aim at comparing specific techniques for data-driven modelling, but it investigates the robustness and flexibility of the proposed multivariate dynamic and multi-step ahead predictions methodology by handling different data-based modelling techniques.

3 Dynamic modelling based on surrogate models

This part presents *i)* an overview on the most common approaches considered in the literature [62, 3] for the univariate dynamic modeling and multi-step ahead prediction using black box models (Section 3.1), *ii)* the proposed methodology for multivariate dynamic modeling and multi-step ahead prediction of chemical processes based on surrogate models (Section 3.2), *iii)* the proposed DOCE procedure for training data generation in dynamic modeling in cases when the purpose is the simplification and complexity reduction of expensive dynamic FPMs (Section 3.3) and *iv)* the procedure for training data generation that mimics practical situations where a FPM of the process is not available and, only input-output signals, measured and collected from the process by the physical sensors network are available (Section 3.2).

3.1 Univariate dynamic modelling and multi-step ahead prediction

Let us consider a univariate dynamic system or process, characterized by D_u control inputs $\mathbf{U} \in R^{D_u}$ and one process output $\mathbf{Y} \in R$, where both can be real data collected from actual plant or simulated data generated by a FPM over discrete, successive and uniform time intervals or sampling periods $\Delta t = (t_i - t_{i-1})$: $[t_0, t_1, t_2, t_3, \dots, t_i, \dots, t_{f-1}, t_f]$, where t_0 and t_f are the first and the final time instances, respectively. Hence, the measured control input and process output signals become $\mathbf{U} = [U_0, U_1, U_2, \dots, U_i, \dots, U_{f-1}, U_f]$ and $\mathbf{Y} = [Y_1, Y_2, Y_3, \dots, Y_i, \dots, Y_{f-1}, Y_f]$.

Using this input-output training information, it is required to construct a data-driven or black-box model that is able to forecast the future values of the output over q time steps-ahead from the current generic time instance t , i.e., $[\hat{Y}_{t+1}, \hat{Y}_{t+2}, \dots, \hat{Y}_{t+q}]$. For this purpose, three main dynamic modeling approaches have been usually considered [62, 3]:

- i) The first approach is the ‘‘Multi-Output’’ (MO) emulator that considers a q -output data-driven model, where each output of this model corresponds to the process output value at the j -th time step, $j = 1, 2, q$. In this case, the model input, x , must include the previously measured values of the process outputs and the corresponding control inputs over a specific time lag L , i.e. $[\hat{Y}_{t+1}, \hat{Y}_{t+2}, \dots, \hat{Y}_{t+q}] = \mathbb{F}(Y_t, Y_{t-1}, \dots, Y_{t-L}, U_t, U_{t-1}, \dots, U_{t-L})$, where \mathbb{F}

is the multi-output black-box model. In this case, a multi-output surrogate model must be used, e.g. multi-output ANN.

- ii) A second alternative is the “Ensemble of Single-Output” models (ESO) approach in which q single-output black box models are considered: each model predicts the single output at each of the q required times, hence $\hat{Y}_{t+1} = \mathbb{f}_1(Y_t, Y_{t-1}, \dots, Y_{t-L}, U_t, U_{t-1}, \dots, U_{t-L}), \dots, \hat{Y}_{t+q} = \mathbb{f}_q(Y_t, Y_{t-1}, \dots, Y_{t-L}, U_t, U_{t-1}, \dots, U_{t-L})$, where $\mathbb{f}_1, \dots, \mathbb{f}_q$ are single-output black-box models.
- iii) The third approach is the recursive single-step emulator, which employs one black box model to approximate the evolution of the process output over a single time increment or step Δt , such that $\hat{Y}_{t+1} = f(Y_t, Y_{t-1}, \dots, Y_{t-L}, U_t, U_{t-1}, \dots, U_{t-L})$.

The single-step emulator approximates the future value of the process output as a function of the process previous control input and output values, considering a specific time lag L . However, it is used in a recursive way for forecasting the output value along q intervals of time. Hence, at every prediction step, the forecasted value of the process output is sent back to the model acting as a part of its input for the next time step prediction, jointly with the new values of the process control inputs.

The single-step emulator is also known as autoregressive model, and it has proved to be much more efficient than the two previous approaches, because of its capability to predict the output variable values over any number of time steps through a recursive procedure. This capability is not obtainable when using the other two approaches (MO and ESO), because they are designed and trained to predict the output value over a fixed or rigid number of time steps. Thus, if it is required to change the prediction horizon (i.e. number of prediction time steps), a completely new model (MO case) or set of models (ESO case) must be constructed. Additionally, the single step emulator approach is more simple/practical in terms of the computational effort required for its implementation, since only one single-output model is constructed and used, instead of the construction/training of a MO model or ESO models. And, finally, it is worth noting that, when considering a multivariate (i.e. multi-output) process, the effort and time required for the construction of data-driven dynamic models based on the MO or ESO approaches will be dramatically magnified. For all the aforementioned reasons, the single-step emulator scheme is considered in this study.

3.2 Proposed multivariate dynamic modelling and multi-step ahead prediction methodology

Assuming a general multivariate dynamic process involving the inputs $\mathbf{U} \in R^{D_u}$ and outputs $\mathbf{Y} \in R^{D_y}$, and keeping the same assumption that all the process inputs and outputs are either measured (real process) or simulated (computer code) at constant, successive and equal time intervals $[t_0, t_1, t_2, t_3, \dots, t_i, \dots, t_{f-1}, t_f]$, the proposed method is based on the construction/training of a set of D_y NARX models (see Eqs. (3)) in order to capture the incremental evolution of the process outputs, \hat{Y}_{t+1} , over one step-ahead time interval. Thus, each model f_j , $j = 1, 2, \dots, D_y$ approximates the future value of the j -th process output at the next time step $t + 1$, i.e. $\hat{y}_{j,t+1}$, as a function of the previous process inputs and outputs, considering a specific time delay L . In this way, any possible correlation between the upcoming value of a certain output $\hat{y}_{j,t+1}$ and any of the process previous input and output can be captured.

$$\left. \begin{aligned} \hat{y}_{1,t+1} &= f_1 [\hat{Y}_t, \dots, \hat{Y}_{t-L}, U_t, \dots, U_{t-L}], \\ &\vdots \\ \hat{y}_{j,t+1} &= f_j [\hat{Y}_t, \dots, \hat{Y}_{t-L}, U_t, \dots, U_{t-L}], \\ &\vdots \\ \hat{y}_{D_y,t+1} &= f_{D_y} [\hat{Y}_t, \dots, \hat{Y}_{t-L}, U_t, \dots, U_{t-L}], \end{aligned} \right\} \quad (3)$$

After the models group (in Eq. (3)) is trained, they are used to forecast the evolution of the process outputs over longer period of time associated to a “totally new” and known profile of the process control input $U^v = [U_{t_0^v}, U_{t_1^v}, \dots, U_{t_i^v}, \dots, U_{t_{f-1}^v}, U_{t_f^v}]$ that affects the process over the “totally new” time sequence $[t_0^v, t_1^v, \dots, t_i^v, \dots, t_{f-1}^v, t_f^v]$ (the superscript v refers to “validation”), i.e. performing multi-step ahead prediction. The latter goal is achieved through recursive prediction, assuming that the first L values of the outputs are known, $(Y_{t_0^v}, \dots, Y_{t_{nL}^v})$, $nL = L$. The recursive prediction starts using the known inputs $x_1^v = [Y_{t_0^v}, \dots, Y_{t_{nL}^v}, U_{t_0^v}, \dots, U_{t_{nL}^v}]$ to predict the process output values at the next time step, $\hat{Y}_{t_{nL+1}^v}$. These predicted output values are used, jointly with the new control input values, as the new models input, $x_2^v = [Y_{t_1^v}, \dots, Y_{t_{nL}^v}, \hat{Y}_{t_{nL+1}^v}, U_{t_1^v}, \dots, U_{t_{nL}^v}, U_{t_{nL+1}^v}]$, for the next time step, so as to predict the output values, $\hat{Y}_{t_{nL+2}^v}$. The recursive prediction continues until the last time step, at which the prediction input $x_{n_v}^v = [\hat{Y}_{t_{f-1-L}^v}, \dots, \hat{Y}_{t_{f-1}^v}, U_{t_{f-1-L}^v}, \dots, U_{t_{f-1}^v}]$ are used to predict the output $\hat{Y}_{t_f^v}$. Notice that $n_v = t_f^v - nL$ is the number of prediction steps or times recursively performed by the models in order to predict the future outputs behavior of the validation signals.

The dynamic models performance can be assessed considering an accuracy metric (e.g., Normalized Root Mean Square Error – NRMSE - Eq.(5)) that computes the difference between the exact and the

predicted values of each of the D_y output signals, respectively, $y_{j,t_i^v}^v \in Y_{t_i^v}^v$ and $\hat{y}_{j,t_i^v}^v \in \hat{Y}_{t_i^v}^v$, $i = nL + 1, \dots, f$, $j = 1, 2, \dots, D_y$.

$$RMSE_j = \sqrt{\frac{1}{n_v} \sum_{i=nL+1}^f (y_{j,t_i^v}^v - \hat{y}_{j,t_i^v}^v)^2} \quad (4)$$

$$NRMSE_j = 100 \frac{RMSE_j}{(\max(y_{j,t_i^v}^v) - \min(y_{j,t_i^v}^v))} \quad (5)$$

It is worth to highlight that the mathematical structure/design of the proposed modeling approach (Eq.(3)) does not directly or explicitly assume any correlation between the outputs of the single-step emulator $[\hat{y}_{1,t+1}, \dots, \hat{y}_{j,t+1}, \hat{y}_{D_y,t+1}]$, since each dynamic model is constructed and trained independently. However, the information about the eventual correlations among the dynamic model outputs is introduced by two mechanisms: i) the fact that each model output $\hat{y}_{j,t+1}$ is computed as a function of the whole set of former values of the process state (inputs and outputs), and ii) the recursive nature of the prediction scheme (Figure 1), which makes each dynamic model f_j to contribute with its prediction $\hat{y}_{1,t+1}$ to the overall prediction of the process output $\hat{Y}_{t+1} = [\hat{y}_{1,t+1}, \hat{y}_{2,t+1}, \dots, \hat{y}_{j,t+1}, \dots, \hat{y}_{D_y,t+1}]$ which, at the end, will constitute the prediction/model input at the next time step. In other words, the output of each dynamic model at the current time step depends on the delayed outputs predicted by other dynamic metamodels at previous time steps, interacting among them during the recursive calculations, so every sole model benefits from the knowledge supplied by the other models in former time steps.

On another hand, it is unlikely that each process output will be dependent of the complete set of the process input and output variables -including their lagged values-, see Eq. (3). But, since there is no prior knowledge about the process behavior, it is useful to allow for all the possible correlations between the process variables, and to let the training task to extract the knowledge about the strength of the allowed correlations. However, this may be also a limitation when a large-scale process is considered, since this will increase the input dimensionality, complicate the model structure and, consequently, increase the number of model parameters. Therefore, this might pose many challenges to the training task: not only the computational effort will increase, but also a higher number of training data will be required in order to face the tuning of the additional model parameters. In this case, previously to the modeling task, an initial analysis can be carried out in order to reduce the models input dimensionality. Although this is not in the scope of this work, it is worth to mention that this can be achieved either based on the knowledge about the system variables and their relations, or using computational techniques as cross-correlation, sensitivity analysis, feature selection and extraction techniques, etc.

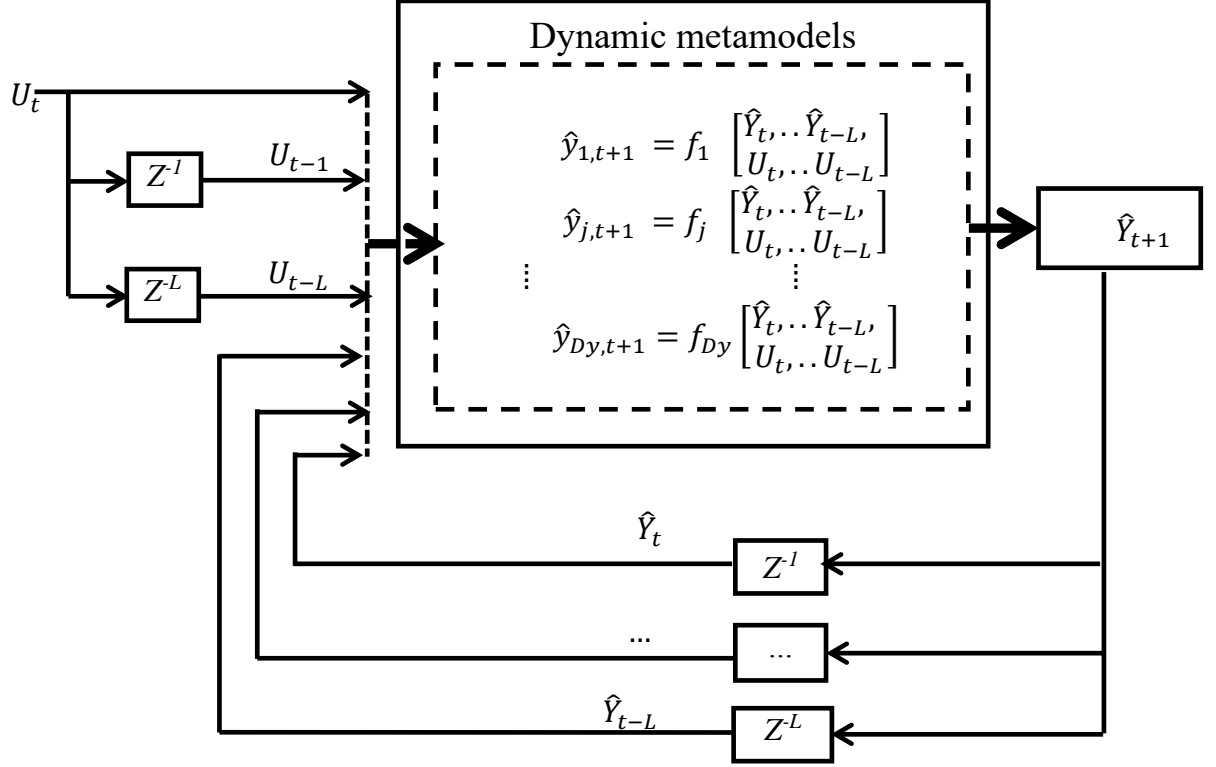


Figure 1. Scheme of the proposed multistep-ahead prediction using the fitted multivariate dynamic models.

Another factor to be selected at this stage is the model lag/order, L , which will affect the resulting model quality and complexity (and, obviously, will also determine the effort required for model training/tuning). Several methods can be found in the literature for making this selection. For linear dynamic models, the cross-correlation between the model output and input has been used [11, 50]. This technique exploits the linear relationships assumed by choosing a linear model. Thus, the cross-correlation between the model output, Y_{t+1} , and the input including different delayed information, $Y_t, Y_{t-1}, \dots, Y_{t-L}, U_t, U_{t-1}, \dots, U_{t-L}$, would give an indication about the delay within which the model input mostly influences its output. Similarly, the correlation between the model inputs and the model prediction error, $e = Y_{t+1} - \hat{Y}_{t+1}$, based on a test set, can reveal the missing regressors, i.e. delayed inputs. Another technique that has been commonly used for the inference/selection of the time lag associated to a linear dynamic model is based on the use of Akanke's information criterion. More details can be found in [49].

For nonlinear dynamic models, a common technique for the estimation of a suitable lag is the calculation of the Lipschitz index from the training data only without any dependence or assumption about the model nature [50, 63, 64]. The method is based on the continuity property of the nonlinear functions that represent input-output models of continuous dynamic systems. The Lipschitz index is

computed considering different lags or delays starting from $L = 0$, and the best embedding dimension is obtained when the index stops decreasing.

Most of the techniques proposed for estimating the data-driven dynamic models order consider only univariate cases. When dealing with multivariate dynamic models, defining a specific different lag for each input with respect to each output is an optimal, but utopic, objective, and to the authors' knowledge, a way for achieving this is not yet available in the literature because it is practically/numerically complicated, mainly due to the eventual combined interactions. A practical and simple approach is to consider a model structure with the same lag for all the input variables [1, 38, 10], see Eq.(3). Although this may seem restrictive, as each process variable, in fact, will present a different physical behavior, the idea is that the importance of the lagged inputs will be adjusted/balanced during the model training according to their significance with respect to the model output, through the manipulation of the values of the weights and biases in the ANN model, or of the parameters ξ_l in the OK model.

In this work, a simple and common try and cut procedure is considered for this selection. So, different sets of multivariate dynamic models are built with different lag values, and the lag that achieves the minimum prediction error of all the D_y models - over a new test set - is selected.

3.3 Design of Computer Experiments for Dynamic Modelling

As mentioned in Section 2.1, different techniques for the DOCE have been commonly used for determining the most convenient training set in the case of data-driven modeling with the purpose of approximating static complex computer models. But these techniques are rarely applied to situations where the purpose is the approximation of “dynamic” computer models.

As indicated before, the few methods already proposed for DOCE in dynamic modeling [48, 12] show different limitations: 1) their capabilities to handle general dynamic processes that often include control inputs and lagged behavior are not illustrated, 2) their robustness to handle different case studies, and their flexibility to integrate different metamodel types are not explored, 3) these sampling procedures are based on the estimated prediction error of the GP/OK metamodels and, therefore, their application with important metamodeling techniques that do not possess this characteristic (e.g. ANN, SVR, etc.) is not feasible, and 4) the sequential nature of these sampling procedure would easily lead to a high computational burden, especially if it is applied to cases characterized by high dimensionality (e.g., several control inputs and process outputs with lagged behavior) and/or include high numbers of training data, see Section 2.1 .

In this section, a DOCE is proposed for data-driven multivariate dynamic modeling of complex processes, assuming the availability of a reliable and accurate FPM. The method is based on the use of

Hammersley sampling design, which is the one selected in this work, as previously justified. However, any efficient alternative can be also used (e.g. optimized Latin hypercube designs, etc.). The proposed procedure is aimed at alleviating the limitations just mentioned at the beginning of this section.

As an important principle of the proposed sampling procedure, it must be taken into account the different nature of the dynamic model (or metamodel) inputs, when compared to steady-state model/metamodel inputs. Since the inputs of a steady-state model are assumed to be independent (e.g. temperature, pressure, volume etc.), the selection of their values combinations $[X]_n$ within their specific bounds is a straightforward task. However, in the case of a dynamic model (Eq. (3)), the model inputs $x = [Y_t, Y_{t-1}, \dots, Y_{t-L}, U_t, U_{t-1}, \dots, U_{t-L}]$ can not be considered independent since, in general, x must include some model inputs ($Y_t, Y_{t-1}, \dots, Y_{t-L}$) which actually correspond to previous outputs (over a certain time lag). Thus, the DOCE technique can freely select any possible combination of values for the process control inputs and their delayed counterparts, $U_t, U_{t-1}, \dots, U_{t-L}$, since these values correspond to external actions applied to the system and, as a consequence, they can be considered neither correlated nor dependent over time (i.e. U_{t-1} does not depend on U_{t-2}). But, in contrast, it is not possible to freely select any arbitrary combination of values for the process outputs and their delayed counterparts, $Y_t, Y_{t-1}, \dots, Y_{t-L}$, because the process outputs may be correlated among others (i.e., y_j depends on $y_{j'}$, j and $j' = 1, 2, \dots, D_y$, $j \neq j'$), they will probably depend on their delayed values (i.e. $y_{j,t-1}$ will probably depend on $y_{j,t-2}$) and, of course, they will depend on the process inputs and their lagged values also (i.e. $Y_{t-i} = f(Y_{t-(i+1)}, U_{t-(i+1)}, \dots)$). Thus, Y_{t-L} are the only output values that can be freely selected, since they are the initial values in the generated profile.

So, the proposed procedure harnesses the Hammersley technique to design a sampling plan that includes n combinations of values of independent models inputs, $[Y_{t-L}, U_t, U_{t-1}, \dots, U_{t-L}]$, over the expected operational domain of the process variables $[U_{t \min} : U_{t \max}, Y_{t \min} : Y_{t \max}]$. Each combination (row of the sampling plan matrix) consists of the D_y initial process output values, Y_{t-L} , besides the $D_u \times (L + 1)$ values of the process control inputs and their lagged counterparts, $[U_t, U_{t-1}, \dots, U_{t-L}]$. The rest of the dynamic metamodel inputs, $[Y_t, Y_{t-1}, \dots, Y_{t-L+1}]$, together with the dynamic metamodel output, Y_{t+1} , are obtained by the simulation of the process model considering the initial process outputs, Y_{t-L} , and the control input profile value, $[U_t, U_{t-1}, \dots, U_{t-L}]$, previously selected by the DOCE technique (Figure 2). Finally, the input–output training data, $[X]_n = [Y_t, Y_{t-1}, \dots, Y_{t-L}, U_t, U_{t-1}, \dots, U_{t-L}]_n$, $[W]_n = [Y_{t+1}]_n$, are used to train the set of D_y dynamic metamodels (Eq. (3)).

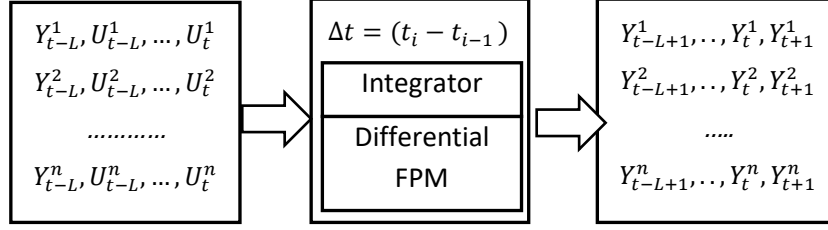


Figure 2. Scheme of the proposed dynamic DOCE.

The sampling procedure becomes simpler when no lag exists (i.e. Markovian process, $L=0$). Hence, the DOCE technique is used to directly select/design a sampling plan $[X]_n = [Y_t, U_t]_n$. After that, n simulation runs are carried out using the process FPM in order to obtain the dynamic model output values $[W]_n = [Y_{t+1}]_n$.

3.4 Input-Output signals generation

In common practical situations, a FPM of the process may not be available and, consequently, the selection of the best training data through the application of the proposed DOCE procedure is not possible. Therefore, this work also considers cases where only input-output signals, measured and collected from the process by the physical sensors network are available. We mimic this situation through considering the process FPM as a real plant that generates these input-output data signals.

The first step in the generation of input-output signals is the synthesis of a piecewise-constant set of the process control inputs $U_t \in R^{D_u}$, which are composed by random step changes of the control input values along the time within the allowable control limits $U_{t \min} : U_{t \max}$. Each step change is expected to hold for some intervals, Δt , to catch the entire dynamic conduct of the process outputs corresponding to this step change. At the same time, the number of sampling periods over which the control input values hold should not be large, in order to avoid gathering redundant information about the steady-state mode of the process. The synthesized control input signals are, then, simulated by the process plant (i.e., model) in order to obtain the corresponding process output signals, to which Gaussian noise is added to emulate the sensors noise. The initial values of the process output signals are selected to be in the middle of their estimated variability domain, in order to maximize the likelihood that during their evolution they could span the sub-regions of the whole domain. These input-output signals, U_t, Y_t , are used to train the system of dynamic surrogate models considering a suitable lag, L .

Usually, the domain within which the process control inputs are allowed to be manipulated $U_{t \min} : U_{t \max}$ is known from the process operational specification. However, the variation domain of the process outputs $[Y_{t \min} : Y_{t \max}]$ should be also checked in front of the recorded process historical data. In the case of the considered simulated case studies, the domain $Y_{t \min} : Y_{t \max}$ has been

estimated through several trial and error simulations, using control profiles within the specified limits of the process control inputs. Also, it is worth to mention that the time step length Δt is conditioned by the subsequent application of the multivariate dynamic models. For example, if these dynamic models are to be employed for monitoring, fault detection and diagnosis, or model predictive control applications, Δt will be the sampling period over which the process must be supervised or controlled. In this work, we have considered the same Δt previously used in the literature for each one of the addressed case studies.

4 Applications

In this section, three benchmark models from the chemical process engineering literature are used to evaluate the proposed modeling methodology, including the sampling procedure, and to compare different metamodels types. These benchmarks are representative examples of nonlinear dynamic systems from three different sub-domains, namely, biochemical, industrial and petrochemical engineering.

The first case involves the model of a continuous bioreactor system that has been considered in different dynamic modeling and control studies, e.g., for data-driven univariate dynamic modeling [38], Quasi-sliding model control [63], and for the design of nonlinear observers [65]. The second application considers the model of a three-tank system that has been commonly used as a benchmark in different monitoring, control and fault detection and diagnosis studies [66, 67, 68, 69]. The third case study involves the model for a shale-oil pyrolysis batch system that has been frequently addressed as example of batch processes dynamic optimization [70, 71].

As previously mentioned, in all these case studies two application scenarios will be considered: the first one would mimic a realistic situation where only input-output signals are available for training the models (see Section 3.4) and, thus, the FPM is used as the process plant from which these signals are collected. The second scenario assumes that the FPM is available for the application of the proposed DOCE procedure in order to optimally select the training data (see Section 3.3). Finally, in both scenarios, the trained dynamic models are tested with a set of totally new data, independently generated in the form of input-output signals. The dynamic models are harnessed for forecasting the process output values, given the values of the validation control inputs, by interacting in a coordinated way through the recursive time integration process proposed (Section 3.2). Finally, the NRMSE (Eq. (5)) is calculated between the predicted outputs and the corresponding known real values.

4.1 Bioreactor

A bioreactor consists in a system inside which microorganisms grow by feeding on the substrate in order to produce the desired product. The difficulties to model the biochemical dynamics associated to the involved processes, usually depending on many factors and conditions not easy to control, convert such systems in challenging situations where to test nonlinear dynamic modeling methods and their applications [65, 63, 38].

A second-order discrete dynamic model of the bioreactor is considered to describe the evolution of the concentrations of the microorganisms, C_m , and the substrate, C_s , inside the reactor, which are affected by the reactor outlet flowrate, U , as detailed by Eqs.(6):

$$\left. \begin{aligned} C_{m(t+1)} &= C_{m(t)} + 0.5 \frac{C_{m(t)} C_{s(t)}}{C_{m(t)} + C_{s(t)}} C - 0.5 U_{(t)} C_{m(t)} \\ C_{s(t+1)} &= C_{s(t)} - 0.5 \frac{C_{m(t)} C_{s(t)}}{C_{m(t)} + C_{s(t)}} C - 0.5 U_{(t)} C_{s(t)} + 0.05 U_{(t)} \end{aligned} \right\} \quad (6)$$

The objective is building a group of data-driven models (Eqs. (7)), which are able to accurately approximate the bioreactor output evolution, $[C_{m(t+1)}, C_{s(t+1)}]$:

$$\left. \begin{aligned} \hat{C}_{m(t+1)} &= f_1(C_{m(t)}, \dots, C_{m(t-L)}, C_{s(t)}, \dots, C_{s(t-L)}, U_{(t)}, \dots, U_{(t-L)}) \\ \hat{C}_{s(t+1)} &= f_2(C_{m(t)}, \dots, C_{m(t-L)}, C_{s(t)}, \dots, C_{s(t-L)}, U_{(t)}, \dots, U_{(t-L)}) \end{aligned} \right\} \quad (7)$$

As previously mentioned, the situation where only signals measured from the process plant are available $[U_{(t)}, C_{m(t)}, C_{s(t)}]$ is first considered for model training. Thus, a flowrate signal, $U_{(t)}$, is synthesized by arbitrary changing its amplitude along the time, where every change lasts over 20 sampling intervals (Figure 3-left). The amplitude values of the step changes are randomly chosen within the known operating range $[0:0.7]$ of the outlet flowrate, $U_{(t)}$.

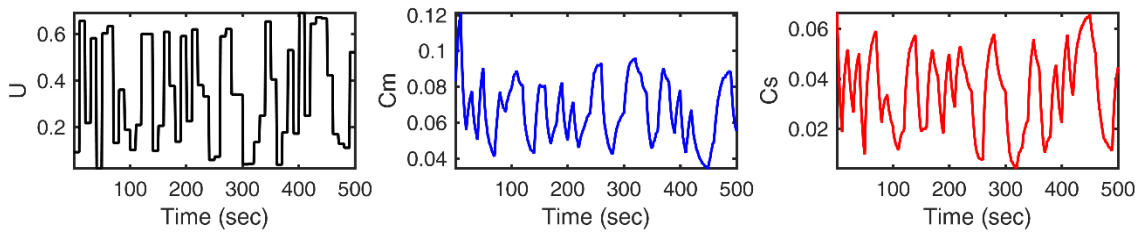


Figure 3. Training signal (bioreactor).

This outlet flowrate signal is introduced to the process FPM (Eq. (6)) in order to obtain the corresponding process output signals: concentrations of microorganisms, $C_{m(t)}$, and substrate, $C_{s(t)}$. To these calculated values, $C_{m(t)}, C_{s(t)}$, a Gaussian noise $\mathcal{N}(\mu = 0, \sigma = 0.0025\%)$ is added to emulate the kind of information which would be available in this case (Figure 3-(middle, right)), where σ is a percentage of the variability domain ($[0: 0.15, 0:0.15]$) of these variables, $C_{m(t)}, C_{s(t)}$. As previously mentioned in Section 3.4, the variation ranges of the output, $C_{m(t)}$ and $C_{s(t)}$ are estimated by carrying out different trial and error simulations using random values of the outlet flowrate, whose variation range is already specified, $[0:0.7]$. Besides, the initial values of the substrate and microorganisms concentrations, $[C_{m(0)}, C_{s(0)}]$, are selected to be in the middle of their variation ranges.

In parallel, a second situation where the training data is generated by means of the proposed sampling procedure for the dynamic modeling has also been considered. Hence, the Hammersley technique is used to sample over the expected variation domain of the dynamic models input, $[0: 0.15, 0: 0.15, 0: 0.7]$, so as to generate a sampling plan which includes 300 sample points (input values combinations), as described in Section 3.3. It should be noted that a different sampling plan is designed for each one of the different lag values considered, since a different lag implies a different number of the dynamic model inputs (i.e. model delayed input).

The procedure application becomes straightforward when no lag is considered ($L = 0$): the DOCE is used to design a sampling plan over the dynamic models input variables $[C_{m(t)}, C_{s(t)}, U_{(t)}]$ and, then, the FPM is used to simulate the model output $[C_{m(t+1)}, C_{s(t+1)}]$; after that, the input-output training data matrices, $[C_{m(t)}, C_{s(t)}, U_{(t)}]_{300} - [C_{m(t+1)}, C_{s(t+1)}]_{300}$, are used to train the models. However, if a lag is considered, just for example, $L = 1$, the Hammersley technique used to design a sampling plan should only consider the independent inputs of the dynamic model, $[C_{m(t-1)}, C_{s(t-1)}, U_{(t-1)}, U_{(t)}]$, and the FPM is employed to simulate the dependent inputs, $[C_{m(t)}, C_{s(t)}]$, of the dynamic model and also the model output, $[C_{m(t+1)}, C_{s(t+1)}]$, as described in Section 3.3. Similarly, in this case, a Gaussian noise with the same mean and standard deviation is added to the all process output data. Finally, the input-output training data matrices, $[C_{m(t)}, C_{m(t-1)}, C_{s(t)}, C_{s(t-1)}, U_{(t)}, U_{(t-1)}]_{300} - [C_{m(t)}, C_{s(t)}]_{300}$, are obtained.

Both training datasets (input-output signals or DOCE) have been used to train different groups of the multivariate models in Eqs.(7), considering the OK and ANNs techniques and various lags ($L = 0, 1, 2$ or 3).

In case of the ANN, its structure has been selected by a search procedure, trying to balance the accuracy and simplicity of the resulting network. Specifically, for any of the models in Eqs.(7), four

different ANNs-based dynamic models, corresponding to four different lags ($L = 0, 1, 2$ or 3), have been fitted. Since in each case the number of the model inputs will be different, a single fixed ANN structure is not likely to be suitable for all these different dynamic models. In this case, a two layer ANN is used, where the number of neurons in each layer equals to double of the number of input variables of the dynamic model. Besides, a log-sigmoid transfer function is used for the hidden layer neurons, whereas a linear transfer function is used for the output layer. The network training is trained by mean of Bayesian regularization backpropagation algorithm, which updates the weights and biases according to Levenberg-Marquardt optimization. This training algorithm usually provides the ANNs with good generalization properties. Again, it is worthy to stress that the selection of the ANN structure and configurations is a time and effort consuming task, even when addressing a low dimensional problem, as the case in hands. This challenge will be magnified as the problem dimensionality and/or the number of training data increases.

Regarding the OK-based models, the “*fmincon*” algorithm for nonlinear optimization of the Matlab optimization toolbox is used to tune the parameters $[\zeta, \lambda]$ (see Section 2.2). Unlike the ANN, all the OK parameters are automatically optimized. However, a main obstacle which complicates the fitting of the OK is the choice of proper initial values necessary for starting the optimization search: a derivative-based optimization algorithm is relatively fast but it can, readily, end up at a local optima, because of the intricacy of the likelihood function. In this work, few optimization runs (each departing from distinct initial values of the parameters) are considered, to ensure effective training of the OK. Although derivative-free optimizers (e.g., genetic algorithms, swarm intelligence-based algorithms) guarantee global search, their search mechanism may demand a huge computational burden considering the expensive evaluation of the likelihood function (see Section 2.2). For assessing the trained models performance, two validation signals have been randomly generated in the same previously mentioned manner (Section 3.4), where the amplitude value of the control scenarios (reactor outlet flowrate, U) has been randomly selected within the specified domain $[0:0.7]$. However, the time length over which each amplitude value holds has been selected differently for each control scenario. The objective is to assess the accuracy and robustness of the multivariate metamodels under different operational conditions and dynamics (Figure 5-top solid black lines) and also to avoid any correspondence with the training conditions.

The dynamic metamodels are harnessed to emulate the evolution of the microorganisms concentration, $C_{m(t+1)}$, and substrate concentration, $C_{s(t+1)}$, along the entire time period (five hundred steps) of each of the two validation scenarios of the output flowrate, U , (Figure 5) through the recursive procedure illustrated in Section 3.2. Table 1 and Figure 4 illustrate the NRMSE of the multivariate dynamic models $\hat{C}_{m(t+1)}$ and $\hat{C}_{s(t+1)}$ when they are trained using the considered techniques (i.e.: ANN, OK), lags ($L=0, 1, 2, 3$) and procedures for training data selection (input-output signals, DOCE).

Table 1. NRMSE (%) of the multivariate dynamic metamodels (bioreactor).

Training data type	Lag	$\hat{C}_{m(t+1)}$		$\hat{C}_{s(t+1)}$		Average ($\hat{C}_{m(t+1)}$ & $\hat{C}_{s(t+1)}$)	
		OK	ANN	OK	ANN	OK	ANN
Signal	0	4.0	2.9	3.1	3.0	3.5	3.0
	1	3.4	4.6	3.0	2.7	3.2	3.6
	2	2.9	3.7	3.0	3.0	2.9	3.4
	3	2.9	4.4	3.0	2.9	3.0	3.6
							$\mu = 3.2,$ $\sigma = 0.2$
DOCE	0	2.3	1.2	0.4	0.3	1.4	0.8
	1	0.7	2.1	0.8	2.3	0.7	2.2
	2	1.6	1.2	0.4	0.3	1.0	0.8
	3	2.3	1.0	0.9	0.4	1.6	0.7
							$\mu = 1.2,$ $\sigma = 0.4$

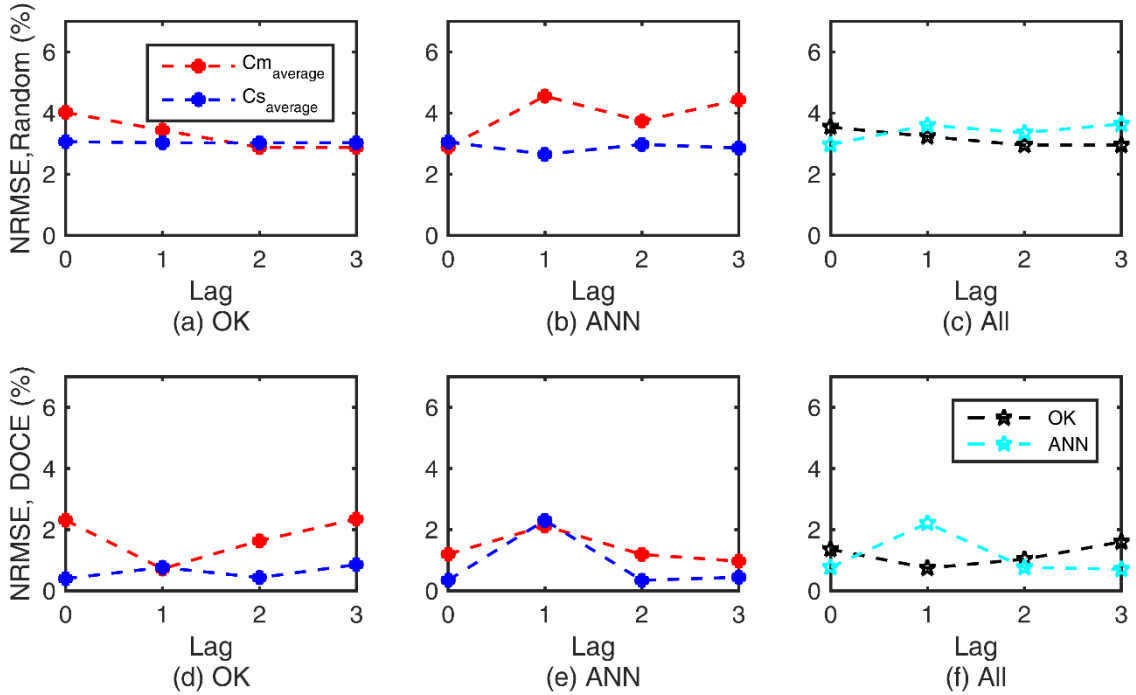


Figure 4. NRMSE of the multi-step-ahead predictions of the output variables (C_m , C_s) of the Bioreactor system versus the considered lag of the dynamic models: (a,b,c) training using signals data and (d,e,f) training using DOCE data.

Notice that, generally, all the models trained with the different training data types (signals, DOCE), techniques (ANN, OK) and lags ($L = 0,1,2,3$) achieved very good performances. In particular, the DOCE further enhances the performance of the multivariate dynamic models, even when only 300 data points have been used for training in these cases, in comparison to the 500 training points used in the cases using input-output signals, (see the overall mean, μ , and standard of deviation, σ , of the different sets of models built with different lags). Also, it is worth to highlight that, regarding the signals-based training procedure, the set of multivariate dynamic models based on ANNs with $L=0$, and OK with $L=2$ achieved the best performances, respectively NRMSE of 3.0 %, and 2.9 %. In relation to the DOCE training procedure, dynamic models based on ANNs with $L= 3$, and OK with $L= 1$ provided the best performance, respectively NRMSE of 0.7 %, and 0.7 %.

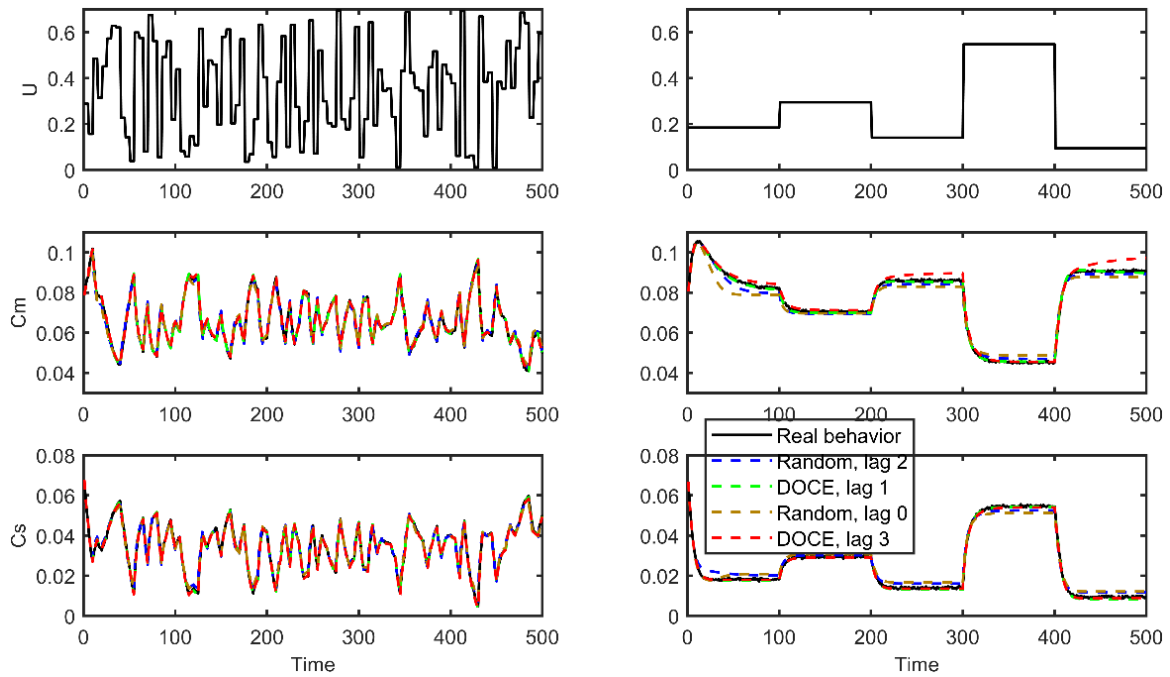


Figure 5. Multi-step ahead prediction of the Bioreactor output variables (C_m , C_s) in two validation scenarios (left and right), predicted by different sets of OK-based dynamic models, trained using different data selection procedures and considering different lags of the dynamic models: solid black line is the exact behavior of the process, blue and brown dashed lines are, respectively, the best and worst predictions of the metamodells set trained using input-output signals and the green and red dashed lines are, respectively, the best and worst predictions of the metamodells set trained using the data selected by the proposed DOCE.

Figure 5 shows the step-ahead predictions of the microorganisms, $C_{m(t+1)}$, and substrate, $C_{s(t+1)}$, concentrations, corresponding to two validation scenarios by means of the multivariate dynamic models set based on the OK technique. The Figure compares -in terms of the prediction accuracy, see

Table 1- the best and the worst models in both training cases: using the input-output signal (blue and brown dashed lines for worst and best respectively) and the DOCE (red and green dashed lines for worst and best, respectively). Similar Figures for the dynamic models based on ANN techniques are illustrated in the Appendix. These Figures not only emphasize the very high prediction accuracy of the best multivariate metamodells, but also show that even in the worst modeling trials (e.g.: blue and red dotted lines) quite satisfactory levels of accuracy are achieved for both the OK and ANN cases. The step-ahead prediction of the multivariate dynamic models set based on the ANN technique are shown in Figure S1.

Azman et al. [38] have used the same case study to illustrate their proposal of univariate dynamic modeling based on GP models, where a single-input-single-output system, $U - C_m$, was considered. They used an input-output training signal of 602 samples with added normal random noise to the C_m data ($\mu = 0, \sigma = 0.0025$), and a random validation scenario that involves 60 time steps. In their work, a dynamic model with a lag $L = 2$ achieved the best prediction accuracy, with a RMSE of 3.44×10^{-3} . Using the methodology proposed in this work, extended prediction capabilities have been achieved with equal (600 samples for the input-output signals training set) or much less (300 samples for the DOCE training set) training data sizes, since all the system outputs (C_m and C_s) have been considered and equivalent RMSEs have been achieved (3% NRMSE that corresponds to a RMSE of 3.1×10^{-3}) over much larger prediction horizons (500 steps-ahead predictions).

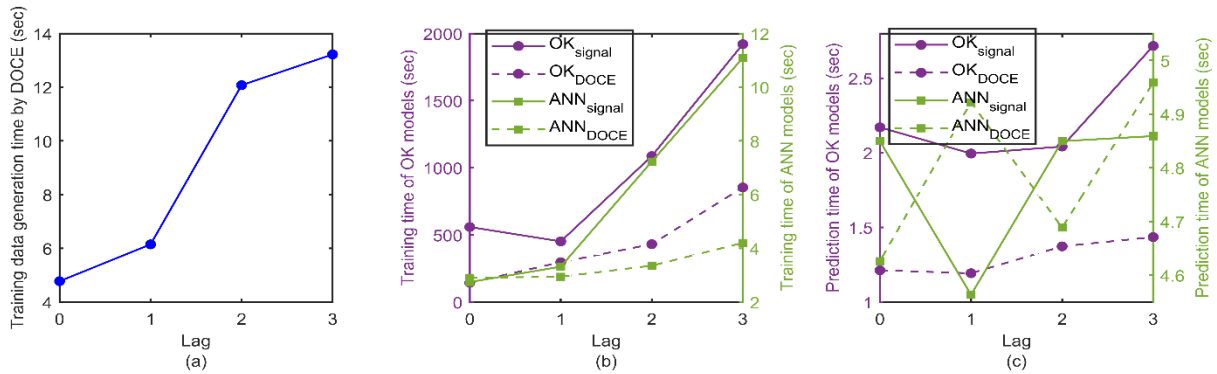


Figure 6. Computational times required for the: (a) generation of the training datasets using the proposed DOCE, (b) training of the multivariate dynamic models sets based on OK and ANN and for (c) the prediction of the testing scenarios of the bioreactor case study. (Intel core i5-6200U CPU@2.3GHz.)

Figure 6-(a) shows that the computational effort required for training data generation using the proposed DOCE procedure increases with the considered lags in the dynamic models: larger considered lags require more integration steps in the analytical model simulation runs (Section 3.3 and Figure 2). Notice that the time required for generating the other type of training data (signals) is not illustrated since it is independent of the model lag (an average of 5.6 sec for generating input-output

signal as in Figure 3). Figure 6-(b) shows that, generally, 1) the increase in the dynamic models lag escalates the training time due to the increase in dynamic model input dimensions and, consequently, the growth of the model parameters to be identified, 2) the training time of the OK-based dynamic models (mauve color) are much larger compared with that of the ANN (green color), because of the very expensive evaluation of the objective function involved in its parameters tuning task (the concentrated likelihood function that implies the expensive calculations of the inverses of the correlation matrix $[R]_{n \times n}$, where n is the number of the training data). Nevertheless, given the fact that the training of the multivariate dynamic models is aimed to be an offline task, the high training computational efforts should be affordable. Figure 6-(c) shows the average prediction time of the entire 500 steps ahead of one testing profile (as in Figure 5) required by the multivariate dynamic models sets with different lags. Notice that the prediction time of the OK-based models are much lower than those of the ANN-based ones, due to the very simple predictor formula associated to OK (see Eq.(1)) compared with the relatively expensive calculations required by the ANN to perform the prediction, which include multiplication of matrices of inputs and weights at each layer besides processing their result by the transfer functions. In general, the prediction time is quite suitable for any online application, as one-step ahead prediction requires an order of magnitude of 10^{-3} sec in a simple Personal Computer.

It is worth noticing that, in this case study, as well as in the next ones, the analysis of the computational time are perturbed by different uncontrolled uncertainties and randomness, which lead to some outliers and noise in the trends of the curves in Figure 6. These uncertainties include the random initial values of the parameters of the metamodels (OK and ANN), the possible change of the behavior of the objective function involved in the parameter tuning task with the increase of the model lags (i.e. increase in the model input dimensions) and, also, the online availability of the processors and RAM of the computer while performing the calculations.

Finally, it should be emphasized that the performance of the proposed methodology in all cases will be affected by the general limitations and criticalities of any data-driven / machine-learning technique, including the one that refers to the size and the quality of the training data: to ensure a satisfactory prediction accuracy level, sufficient number of training data should be available, including enough information about the different dynamic conditions/states/scenarios that the process will face. Also, the quality of the training data in terms of the measuring error/noise (unavoidable in real systems) is an important factor affecting the model performance, as the excess of noisy measurements could lead to poor model performance.

Figure 7 shows two experiments that address the effects of the training dataset size and noise over the model prediction accuracy in this case, based on the OK model, trained with data generated via the DOCE procedure and considering lag =1 (best overall prediction accuracy in this case). Figure 7-(a)

shows how the size of the training dataset (fixing the noise standard deviation to 0.0025%) affects the average prediction accuracies of the model. Considering the overall accuracy (black stars) the initial positive effect of increasing the size of the training dataset achieves an optimum situation and, from this point, an increase of the training data does not necessarily enhance further the accuracy (as usually happens with these techniques). Figure 7-(b) shows how the noise/error in the data also affects the average prediction accuracy of the models (fixing the number of training data to 300, which was the best value for the nominal conditions, with a noise standard deviation of 0.0025%). The Figure also shows that the methodology behaves robust with respect to the change of the training dataset size and the noise.

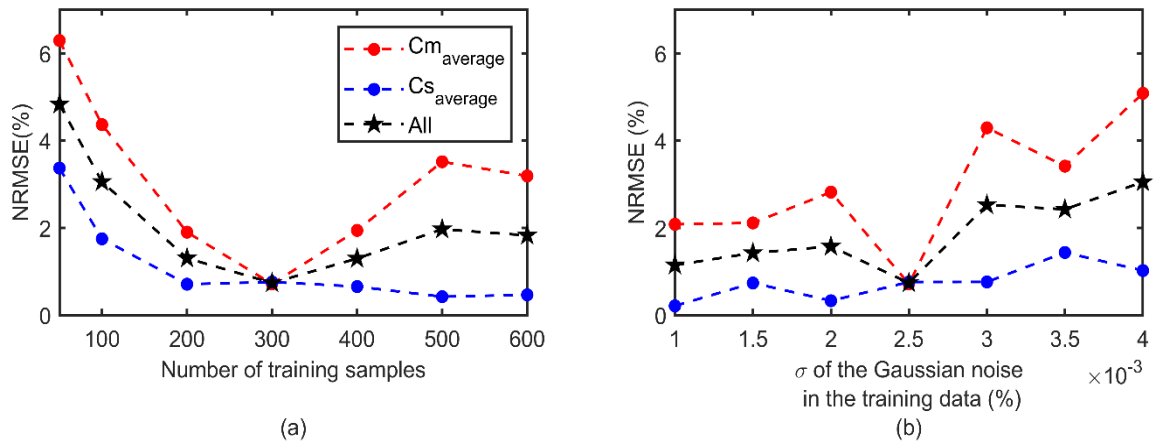


Figure 7. Effect of the training dataset size (a) and the amount of noise (b) on the performance of the multivariate dynamic models set based on the OK technique, trained by data selected via the DOCE procedure and considering lag=1.

4.2 Three-tanks system

The second application is based on the three-tank system illustrated in Figure 8. It is a well-known nonlinear process that has been commonly used as a benchmark in different monitoring, control and fault detection and diagnosis studies [66, 67, 68]. Its popularity stems from the fact that it involves characteristics of fluid distribution systems (tanks, pumps, and pipelines) often encountered in real plants [69, 72], as cooling water circuits of distillation columns and feed water systems in power stations, etc.

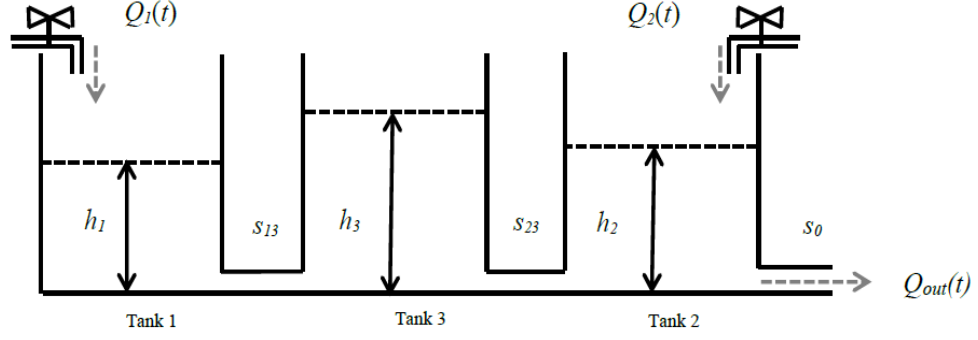


Figure 8. Schematic representation of the three-tanks benchmark system.

$$\left. \begin{aligned}
 A \frac{dh_1}{dt} &= -a_1 s_{13} \operatorname{sgn}(h_1 - h_3) \sqrt{2g|h_1 - h_3|} + Q_1 \\
 A \frac{dh_2}{dt} &= a_3 s_{23} \operatorname{sgn}(h_3 - h_2) \sqrt{2g|h_3 - h_2|} - a_2 s_0 \sqrt{2gh_2} + Q_2 \\
 A \frac{dh_3}{dt} &= a_1 s_{13} \operatorname{sgn}(h_1 - h_3) \sqrt{2g|h_1 - h_3|} - a_3 s_{23} \operatorname{sgn}(h_3 - h_2) \sqrt{2g|h_3 - h_2|}
 \end{aligned} \right\} (8)$$

The system model (in Eqs.(8)) describes the dynamic relations among the levels of the tanks, h_1, h_2, h_3 , (the process outputs) and the inlet flowrates, Q_1, Q_2 , (the control input), whose limiting value is $0.005 \text{ m}^3/\text{s}$. The values of the cross section area of the tanks, A , the cross section areas of the connecting pipes s_{13}, s_{23}, s_0 , and the flow coefficients a_1, a_3, a_0 , can be found in [72].

A set of multivariate dynamic models is to be constructed, which describes the step-ahead evolution of the tanks levels $h_{1(t+1)}, h_{2(t+1)}, h_{3(t+1)}$, see Eqs.(9). The same general procedure described in Section 3 and the application details illustrated in Section 4.1 are systematically followed in this case, too.

$$\left. \begin{aligned}
 \hat{h}_{1(t+1)} &= f_1(h_{i(t)}, h_{i(t-1)}, \dots, h_{i(t-L)}, Q_{j(t)}, Q_{j(t-1)}, \dots, Q_{j(t-L)}) \\
 \hat{h}_{2(t+1)} &= f_2(h_{i(t)}, h_{i(t-1)}, \dots, h_{i(t-L)}, Q_{j(t)}, Q_{j(t-1)}, \dots, Q_{j(t-L)}) \\
 \hat{h}_{3(t+1)} &= f_3(h_{i(t)}, h_{i(t-1)}, \dots, h_{i(t-L)}, Q_{j(t)}, Q_{j(t-1)}, \dots, Q_{j(t-L)})
 \end{aligned} \right\} (9)$$

where $i = 1, 2, 3$, and $j = 1, 2$

The first training set is obtained by means of the generating input-output signals including 750 instances (Figure 9). Thus, piecewise constant signals of the fluid inlet flowrate, Q_1 and Q_2 , are composed, where the signal amplitude values are randomly selected along the time in a constant piecewise manner within the ranges of $[0.0 : 0.005] \text{ m}^3/\text{s}$, and each amplitude change holds for 20 sampling periods. The corresponding output signals, h_1, h_2 and h_3 , are obtained by the process FPM simulation, where Gaussian noise of the same magnitude described in Section 4.1 is added to them.

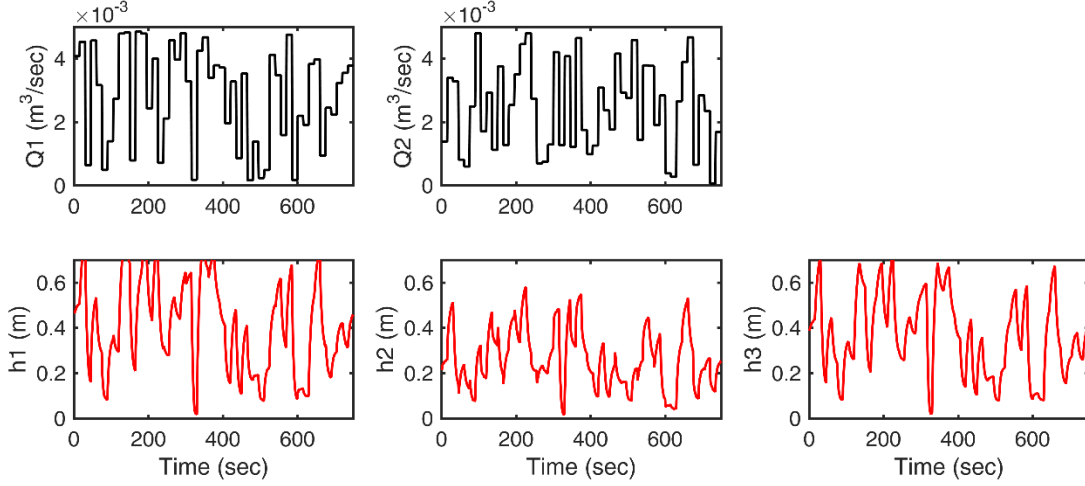


Figure 9. Input-output signal of the three-tanks system used for training the set of multivariate dynamic models.

A second training set is again generated following the proposed dynamic DOCE procedure to include 300 samples over the expected variation domain $[0: 0.8, 0: 0.8, 0: 0.8, 0: 0.005, 0: 0.005]$ of the process variables, respectively, h_1, h_2, h_3, Q_1 and Q_2 . Gaussian noise with the same mean and standard deviation is added to the process output data and, finally, the input-output training matrices are obtained, $[h_{i(t)}, \dots, h_{i(t-1)}, Q_{j(t)}, \dots, Q_{j(t-1)}]_{300} - [h_{i(t+1)}]_{300}, i = 1,2,3$ and $j = 1,2$.

The set of dynamic models in Eq.(9), $[\hat{h}_{1(t+1)}, \hat{h}_{2(t+1)}, \hat{h}_{3(t+1)}]$, is trained using each type of the training datasets, based on the different considered techniques (i.e. OK and ANNs) and different lags. The same setting and guidelines used in Section 4.1 for selecting the ANN structure, for customizing its configurations and for tuning the OK models are also considered here.

Table 2. NRMSE (%) of the multivariate dynamic metamodels (three-tanks).

Training data type	Lag	$\hat{h}_{1(t+1)}$		$\hat{h}_{2(t+1)}$		$\hat{h}_{3(t+1)}$		Average ($\hat{h}_{1(t+1)}, \hat{h}_{2(t+1)}, \hat{h}_{3(t+1)}$)	
		OK	ANN	OK	ANN	OK	ANN	OK	ANN
Signal	0	2.7	1.9	1.8	1.4	2.4	2.0	2.3	1.7
	1	6.6	2.1	2.9	1.9	5.1	2.2	4.9	2.0
	2	5.7	2.8	2.7	2.1	4.4	2.9	4.3	2.6
	3	4.7	2.9	2.5	3.7	3.8	3.0	3.7	3.2
								$\mu = 3.8,$ $\sigma = 1.1$	$\mu = 2.4,$ $\sigma = 0.7$
DOCE	0	1.7	0.4	0.7	0.2	1.6	0.3	1.3	0.3
	1	0.6	0.2	0.3	0.2	0.5	0.2	0.4	0.2
	2	0.9	0.5	0.4	0.4	0.9	0.5	0.7	0.5
	3	0.8	1.6	0.6	1.4	0.7	1.7	0.7	1.6

	$\mu = 0.8,$ $\mu = 0.7,$ $\sigma = 0.4$ $\sigma = 0.6$
--	--

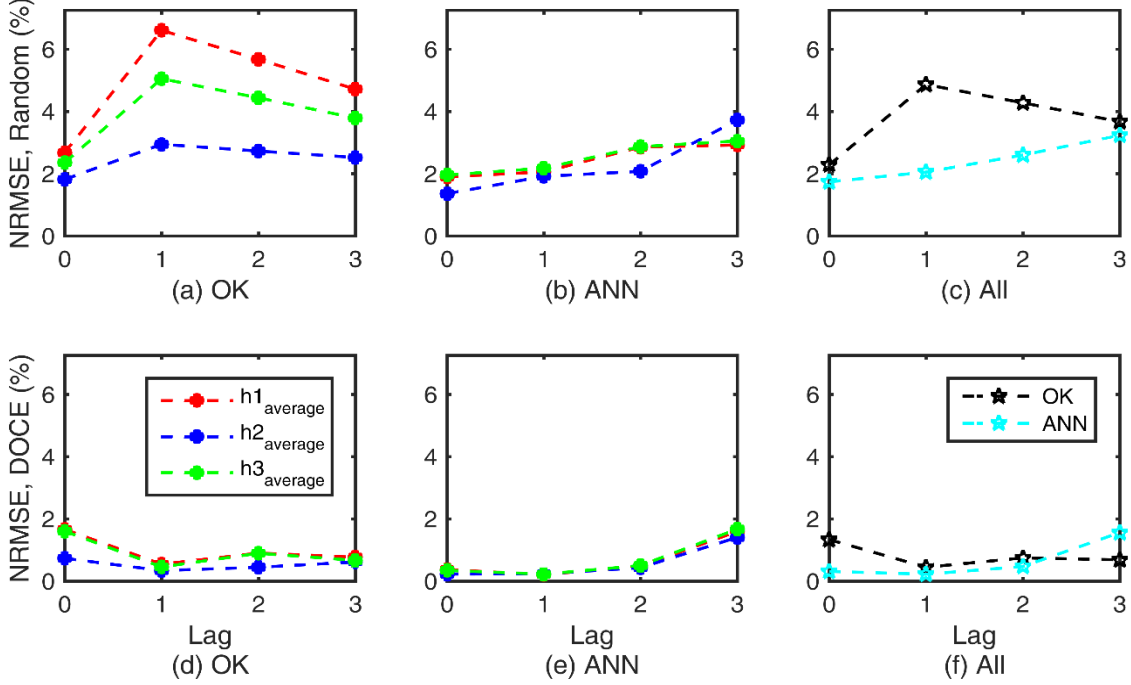


Figure 10. NRMSE of the multi-step-ahead predictions of the output variables (h_1, h_2, h_3) of the three-tanks system versus the considered lag of the dynamic models: (a,b,c) training using input-output signals data and (d,e,f) training using DOCE data.

Again, two validation signals, generated as described in Sections 3.4 and 4.1, are used to assess the fitted dynamic models (Figure 11). It deserves to emphasize that the amplitude values of the validation control scenarios (inlet flowrates, Q_1 and Q_2) have been randomly chosen within the specified domain $[0, 0.005] \text{ m}^3/\text{s}$ and the time length over which amplitude values hold has been selected differently for each scenario (see Figure 11 top four subplots). Table 2 and Figure 10 show the low NRMSE of the multivariate dynamic models $\hat{h}_{1(t+1)}, \hat{h}_{2(t+1)}$ and $\hat{h}_{3(t+1)}$ when they are trained using the considered techniques, lags and procedures for training data selection. Also, the evolution of the tanks levels along the time predicted by the multivariate dynamic models sets based on the OK and the ANN techniques are shown in Figure 11 and Figure S2, respectively.

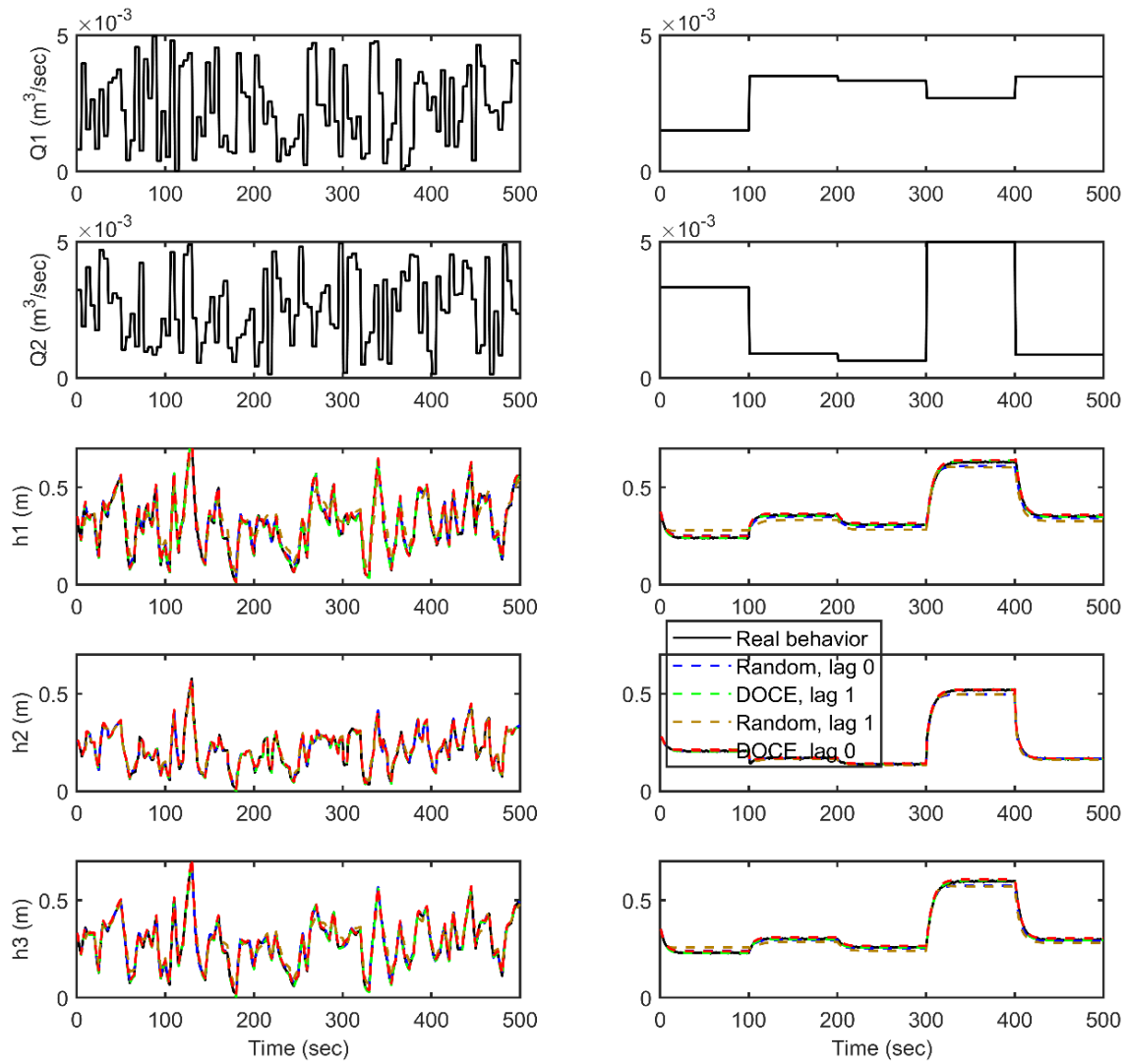


Figure 11. Multi-step ahead prediction of the three-tanks system output variables (h_1 , h_2 , h_3) in two validation scenarios (left and right), predicted by different sets of OK-based dynamic models, trained using different data selection procedures and considering different lags of the dynamic models: solid black line is the exact behavior of the process, blue and brown dashed lines are, respectively, the best and worst predictions of the metamodels set trained using input-output signals and the green and red dashed lines are, respectively, the best and worst predictions of the metamodels set trained using the data selected by the proposed DOCE.

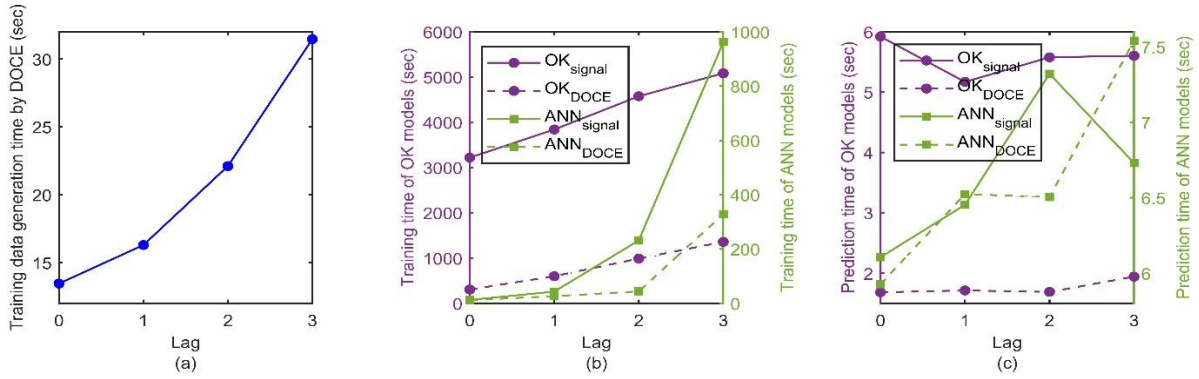
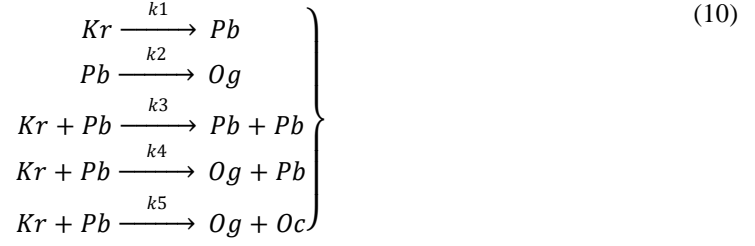


Figure 12. Computational times required for the: (a) generation of the training datasets using the proposed DOCE, (b) training of the multivariate dynamic models sets based on OK and ANN and for (c) the prediction of the testing scenarios of the three tanks case study. (Intel core i5-6200U CPU@2.3GHz.)

Figure 12-(a) shows the computational effort required for the training data generation using the proposed DOCE procedure. As in the previous case, the time required for generating the other type of training data (input-output signal, see Figure 9) is constant (now equals to an average of 9.0 sec.), and the rest of conclusions are also equivalent: Figure 12-(b) illustrates the escalation of the training time with the increase of the dynamic models lag and that the training time of the OK-based dynamic models (mauve color) are much larger relative to the ANN (green color). Figure 12-(c) shows the average prediction time of the entire 500 steps ahead of one testing profile. It emphasizes again the capabilities of the dynamic models for real time predictions, requiring an order of magnitude from 10^{-3} to 10^{-2} sec for one-step ahead prediction.

4.3 Oil Shale Pyrolysis

Oil shale pyrolysis is an industrial process that aims at extracting shale oil through the decomposition of the shale. Pyrolysis approximates the natural processing of the organic material in the shale, i.e. kerogen, using higher temperatures to compensate for the geological time frame [71]. Upon heating, kerogen decomposes by consecutive reactions into a benzene-soluble material (pyrolytic bitumen), which, in turn, decomposes to form the final products of oil, gas, and carbonaceous residue on the spent shale [71]:



The series of reactions taking place during the process are illustrated in Eqs.(10), where Kr is the kerogen, Pb is the pyrolytic bitumen, Og is oil and gas and Oc is the organic carbon residue [71]. The mathematical model in Eqs.(11) describes the evolution of the concentrations, $C_{Kr}, C_{Pb}, C_{Og}, C_{Cr}$, where k_i is the specific reaction rate, k_{i0} is its initial value, E_i is the activation energy, R is the gas constant and T is the temperature that can be manipulated within the range of $[698.15 \leq T \leq 748.15]$ [70]:

$$\left. \begin{array}{l}
\frac{dC_{Kr}}{dt} = -k_1 C_{Kr} - (k_1 + k_4 + k_5) C_{Kr} C_{Pb} \\
\frac{dC_{Pb}}{dt} = k_1 C_{Kr} - k_2 C_{Pb} + k_3 C_{Kr} C_{Pb} \\
\frac{dC_{Og}}{dt} = k_2 C_{Pb} - k_4 C_{Kr} C_{Pb} \\
\frac{dC_{Cr}}{dt} = k_5 C_{Kr} C_{Pb} \\
k_i = k_{i0} \exp\left(\frac{E_i}{RT}\right), \quad i = 1,2,3,4,5
\end{array} \right\} \quad (11)$$

This model has been commonly used for the dynamic optimization of the process [70], aiming at maximizing the pyrolytic bitumen production at the end of the batch, i.e. $C_{Pb}(t_f)$. With this objective, the optimal batch time, t_f , and the optimal temperature profile over the batch time $[t_0: t_f]$ are to be identified, considering the initial conditions $[C_{Kr}(t_0), C_{Pb}(t_0), C_{Og}(t_0), C_{Cr}(t_0)] = [1, 0, 0, 0]$.

In this application, we illustrate the development of a set of dynamic models (Eqs.(12)) which is able to accurately approximate the future behavior of the oil shale pyrolysis process. Six different batch runs are simulated, such that each batch corresponds to a different control profile of the temperature, composed as previously mentioned within the known limits $[698.15 K: 748.15K]$ and, random initial conditions $[C_{Kr}(t_0), C_{Pb}(t_0), C_{Og}(t_0), C_{Cr}(t_0)]$ between the range $[0.95 : 1.05, 0 : 0.05, 0 : 0.05, 0 : 0.05]$. Also, a random Gaussian noise of the aforementioned order of magnitude is added to the output values. It is worthy to mention that the batch time is set to its optimal value identified in the literature [71], i.e. $t_f = 9.3 \text{ min}$, while the sampling period is set to 0.093 min .

$$\left. \begin{aligned}
\hat{C}_{Kr(t+1)} &= f_1(C_{Kr(t)}, \dots, C_{Kr(t-L)}, C_{Pb(t)}, \dots, C_{Pb(t-L)}, C_{Og(t)}, \dots, C_{Og(t-L)}, C_{Cr(t)}, \dots, C_{Cr(t-L)}, T(t), \dots, T(t-L)) \\
\hat{C}_{Pb(t+1)} &= f_2(C_{Kr(t)}, \dots, C_{Kr(t-L)}, C_{Pb(t)}, \dots, C_{Pb(t-L)}, C_{Og(t)}, \dots, C_{Og(t-L)}, C_{Cr(t)}, \dots, C_{Cr(t-L)}, T(t), \dots, T(t-L)) \\
\hat{C}_{Og(t+1)} &= f_3(C_{Kr(t)}, \dots, C_{Kr(t-L)}, C_{Pb(t)}, \dots, C_{Pb(t-L)}, C_{Og(t)}, \dots, C_{Og(t-L)}, C_{Cr(t)}, \dots, C_{Cr(t-L)}, T(t), \dots, T(t-L)) \\
\hat{C}_{Cr(t+1)} &= f_4(C_{Kr(t)}, \dots, C_{Kr(t-L)}, C_{Pb(t)}, \dots, C_{Pb(t-L)}, C_{Og(t)}, \dots, C_{Og(t-L)}, C_{Cr(t)}, \dots, C_{Cr(t-L)}, T(t), \dots, T(t-L))
\end{aligned} \right\} (12)$$

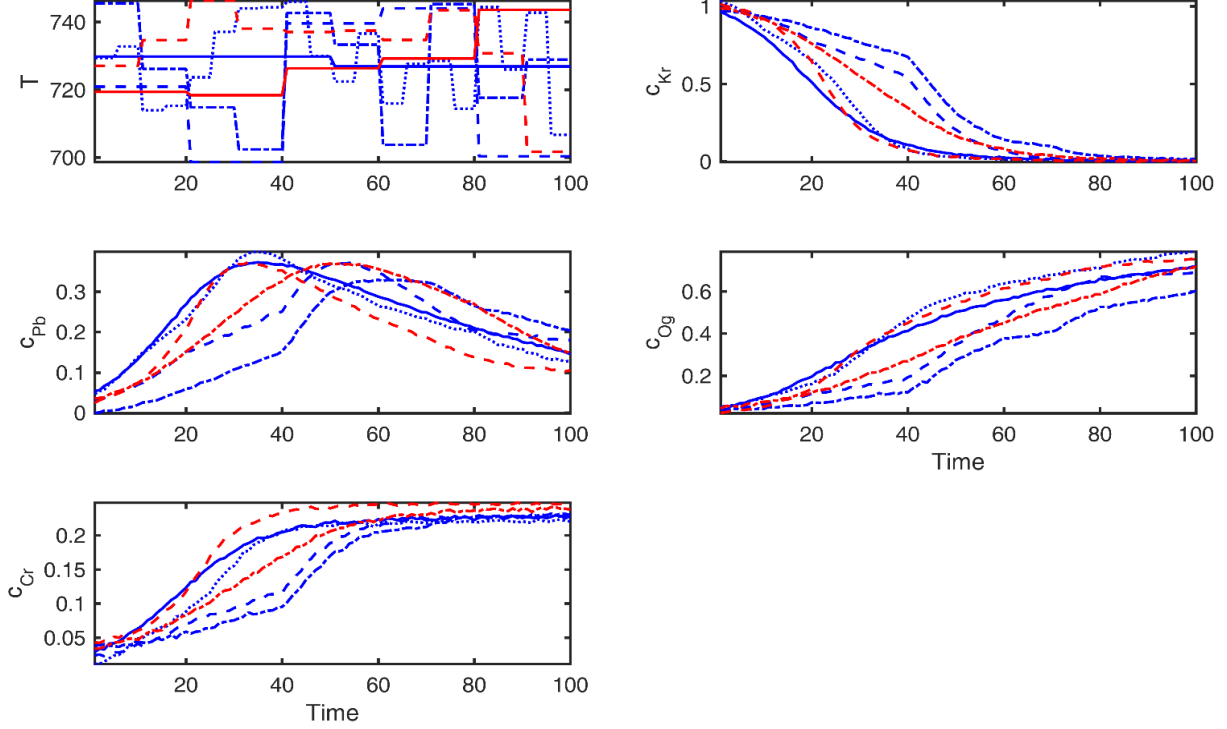


Figure 13. Training (blue) and validation batches (red).

Four batches (blue lines in Figure 13) are considered as the input-output training set, while two batches (red lines in Figure 13) are used for the testing purpose. On the other side, a second training set including 400 samples or instances is generated by the proposed procedure for dynamic DOCE, considering the expected variation domain of the process variables C_{Kr} , C_{Pb} , C_{Og} , C_{Cr} , and T : $[0 : 1.2, 0 : 0.6, 0 : 1.2, 0 : 0.6, 698.15 : 748.15]$

Both types of training data, input-output signals and DOCE, are utilized for fitting the models set in Eqs.(12), considering also the different techniques and lags as in the previous sections. The trained sets of models are used to predict the evolution of the process outputs, C_{Kr} , C_{Pb} , C_{Og} , C_{Cr} , over 100 time steps, corresponding to each validation scenario of the temperature, T (red lines in Figure 13).

The performance of each one of the dynamic models is illustrated in Table 3 and Figure 14, where the prediction NRMSE is shown for each model independently and for the set of dynamic models. It can be noticed that the multivariate dynamic models possess quite satisfactory level of accuracy (Figure 14-(c, f) and the last two column in Table 3), specially taking into account the complex nature of the

considered case. This complexity is expressed by the higher dimensionality of the output, the complex reactions mechanisms (see Eqs.(10)), the high nonlinear relations in the system (see Eqs.(11)) and by the nature of the process as a batch type that often included transient dynamics and sophisticated reaction kinetics and stoichiometry. Besides, the kerogen concentration, C_{Kr} , seems to be the easiest output to be modeled (Figure 15, red lines), however, the organic carbon residue, C_{Cr} , represents the most difficult behavior to be captured (Figure 15, yellow lines).

Table 3. NRMSE (%) of the multivariate dynamic metamodels (oil shale Pyrolysis).

Training data type	Lag	$\hat{c}_{Kr(t+1)}$		$\hat{c}_{Pb(t+1)}$		$\hat{c}_{Og(t+1)}$		$\hat{c}_{Cr(t+1)}$		Average ($\hat{c}_{Kr(t+1)}, \hat{c}_{Pb(t+1)}, \hat{c}_{Og(t+1)}, \hat{c}_{Cr(t+1)}$)	
		OK	ANN	OK	ANN	OK	ANN	OK	ANN	OK	ANN
Signal	0	3.7	2.5	5.4	4.6	3.7	2.5	8.4	6.9	5.3	4.1
	1	2.6	2.6	4.5	8.0	1.6	4.9	8.0	7.4	4.2	5.8
	2	1.7	4.0	3.4	7.4	1.5	4.8	7.9	9.4	3.6	6.4
	3	1.7	1.8	2.9	5.9	1.5	4.0	7.8	6.8	3.5	4.6
											$\mu = 4.2,$ $\sigma = 0.8$
DOCE	0	3.6	1.4	5.6	3.9	4.2	3.0	5.2	5.9	4.6	3.6
	1	4.2	0.4	7.2	0.9	6.0	0.6	8.5	2.9	6.5	1.2
	2	1.4	0.7	3.4	1.1	3.0	0.5	5.2	1.0	3.3	0.8
	3	1.7	0.1	3.0	0.2	3.9	0.2	1.5	0.7	2.5	0.3
											$\mu = 4.2,$ $\sigma = 1.7$

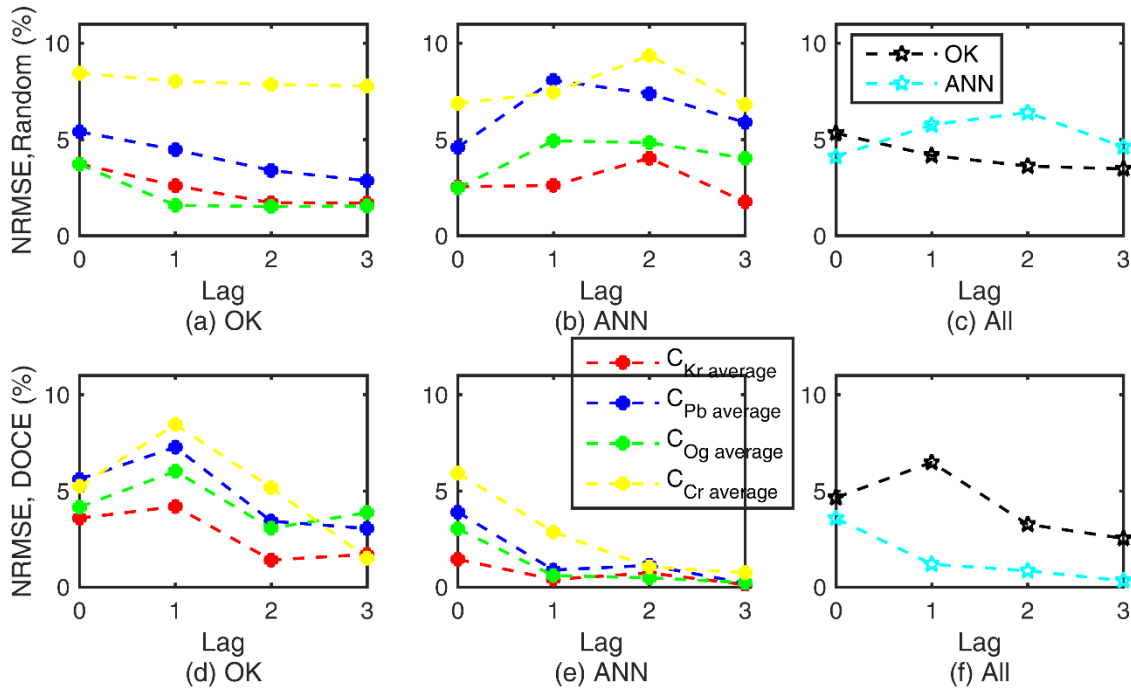


Figure 14. NRMSE of the multi-step-ahead predictions of the output variables (C_{Kr} , C_{Pb} , C_{Og} , C_{Cr}) of the oil shale pyrolysis process versus the considered lag of the dynamic models: (a,b,c) training using input-output signals data and (d,e,f) training using DOCE data.

The best performances (input-output training signal) have been achieved by the sets of dynamic models based on ANNs with $L=0$ and OK with $L=3$, finding NRMSE of 4.1 %, and 3.5 %, respectively. The dynamic models sets (DOCE training) based on ANNs with $L= 3$ and OK with $L= 3$ have provided the best performances, finding NRMSE of 0.3% and 2.5%, respectively. Again, the models trained using data generated by the proposed DOCE procedure exhibit enhanced performance with respect to those trained by the data generated using the input-output signal.

Figure 15 shows the evolutions of the kerogen, $C_{Kr(t+1)}$, pyrolytic bitumen, $C_{Pb(t+1)}$, oil and gas, $C_{Og(t+1)}$, and the organic carbon residue, $C_{Cr(t+1)}$, concentrations in two validation batches, predicted by the set of OK-based dynamic models. Similarly, the worst and best performances with respect to each training data type are highlighted by the aforementioned colors. The Figure shows that even in the worst modeling trials (blue and red dotted lines) quite satisfactory levels of accuracy are achieved, especially for the OK and ANN cases. The step-ahead prediction of the ANN-based dynamic models is showed in Figure S3.

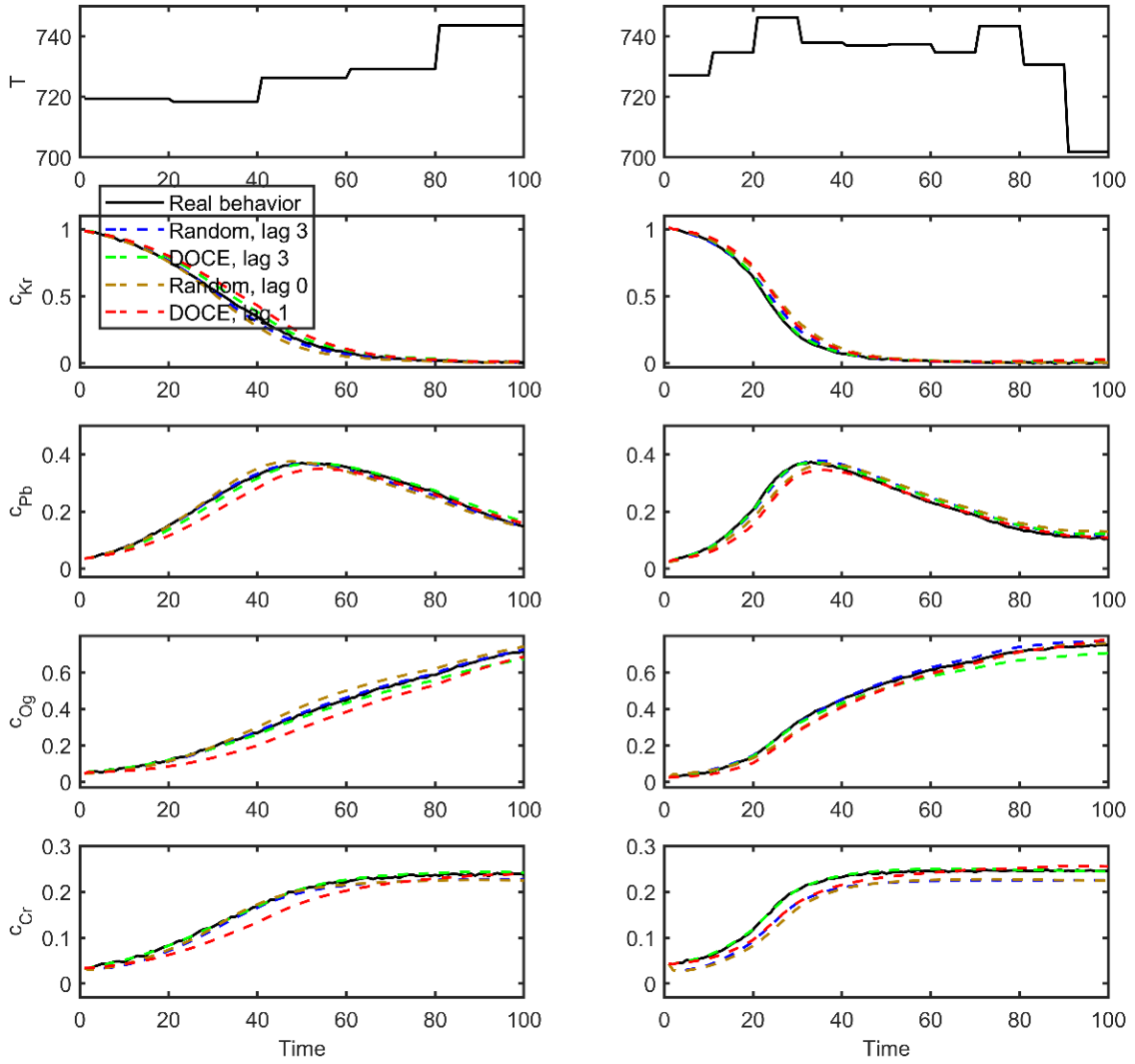


Figure 15. Multi-step ahead prediction of the output variables of the oil shale pyrolysis process (C_{Kr} , C_{Pb} , C_{Og} , C_{Cr}) in two validation batches (left and right), predicted by different sets of OK-based dynamic models, trained using different data selection procedures and considering different lags of the dynamic models: solid black line is the exact behavior of the process, blue and brown dashed lines are, respectively, the best and worst predictions of the metamodels sets trained using arbitrary input-output signals, respectively, and the green and red dashed lines are, respectively, the best and worst predictions of the metamodels sets trained using data selected by the proposed DOCE.

Figure 16 shows the training data collected by the input-output signal generation (red crosses) and the proposed dynamic DOCE (cyan circles) procedures projected onto some of the metamodels input dimensions. The Figure shows that when the methodology is used for approximating a complex FPM, it is capable of efficiently generating all the possible combinations of the process variables values by the proposed DOCE procedure, in order to collect dataset covering the entire domain of the models input and, consequently, to enhance its prediction accuracy. However, when the methodology is meant to be applied to a real process, the FPM model is considered as a process plant, but with only few

input-output datasets available, which have been generated following the procedure in Section 3.4 (one/few signal(s) or “profile(s)” evolving through the complete set of feasible situations). See Figure 16, where the training data in the latter case (red crosses) represent a small local subset within the entire domain of variability of the model input variables. In this case, these “profiles” have been generated in a random way (see Section 3.4) since we do not know the control mechanism (problem) of each specific process and, moreover, this is the typical procedure used in the literature [38, 1].

For real situation, where a database of the process variables measurements history is available, the training data selection should cover as much as possible the dynamic conditions of the process, in order to feed model with sufficient information about the process [53].

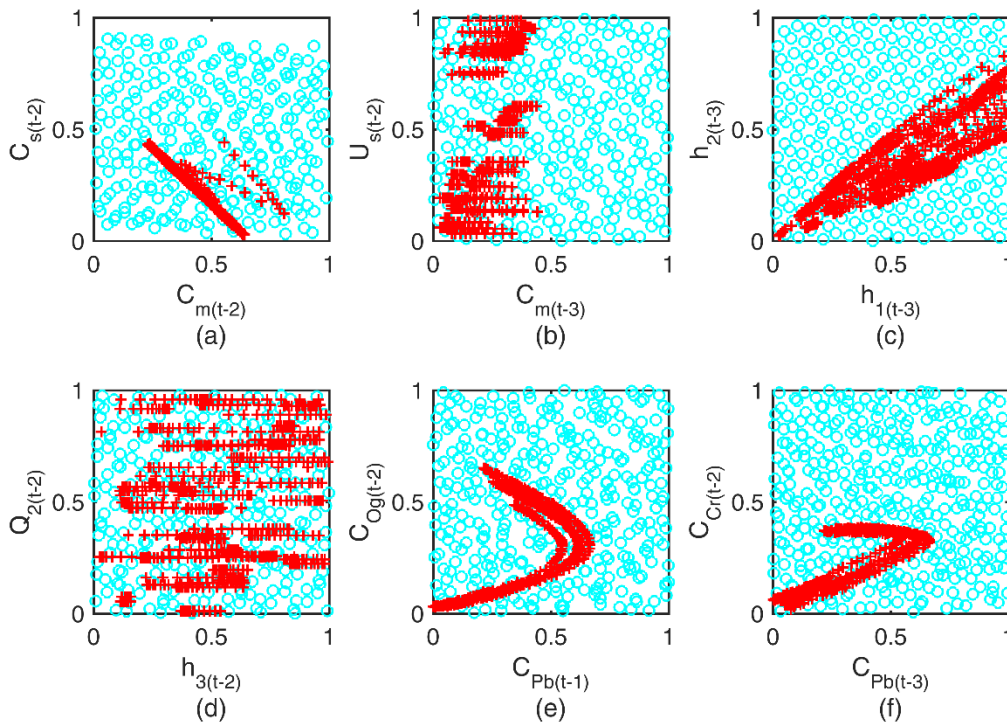


Figure 16. Comparison between the training data selected by the proposed DOCE procedure (cyan circles) and the training data in the case of using input-output signals (red crosses), both projected over arbitrary selected pairs of the dynamic models input dimensions: (a,b) bioreactor,(c,d) three-tanks and (e,f) oil shale pyrolysis.

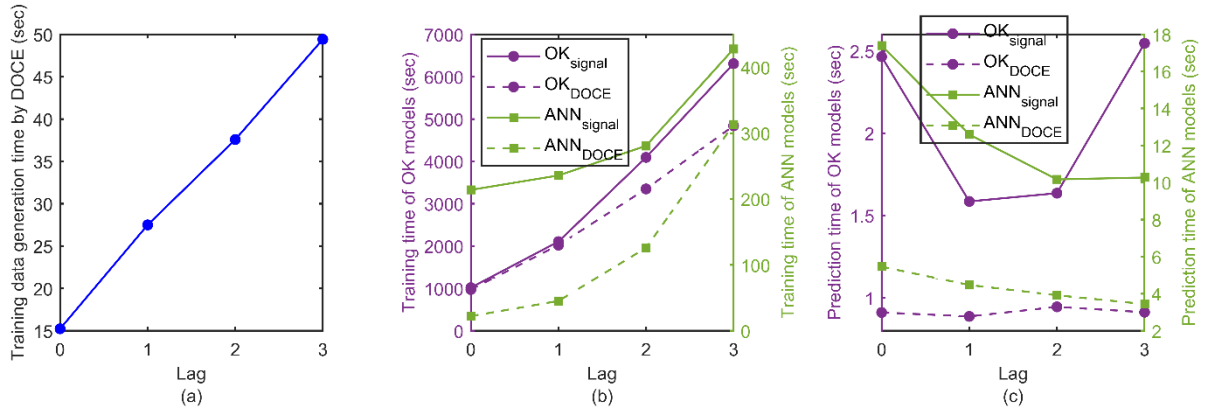


Figure 17. Computational times required for the: (a) generation of the training datasets using the proposed DOCE, (b) training of the multivariate dynamic models sets based on OK and ANN and for (c) the prediction of the testing scenarios of the oil shale pyrolysis case study. (Intel core i5-6200U CPU@2.3GHz.)

Figure 17-(a) shows the computational time required for the training data generation using the proposed DOCE procedure. The time required for generating the other type of training data (input-output signal, see Figure 13) is (again) constant and equals to an average of 9.0 sec. Figure 17-(b) illustrates the escalation of the training time with the increase of the dynamic models lag and that the training times of the OK-based dynamic models (mauve color) are (again) larger relative to the ANN (green color) escalation. Figure 17-(c) shows the average prediction time of the entire 100 steps ahead of one testing profile. It emphasizes again the capabilities of the dynamic models for real time predictions, requiring an order of magnitude from 10^{-2} to 10^{-1} sec for one-step ahead prediction.

Finally, it should be mentioned that in all the analyzed cases but, especially, in situations where only few input-output signals are available for the training and/or they may represent a biased or partial view of the process (as in the last case study, see Figure 16-(e,f), red crosses), the resulting dynamic models may be very sensible to the eventual evolution of the real process behavior through the time, which may drive it to unexpected/unexplored conditions, either due to the natural evolution of the process (e.g.: heat exchanger fouling, process aging, drifting, etc.), or because a wrong/incomplete selection of the training dataset. In such situations, the dynamic models can perform poorly, because they are going to be applied outside the domain of knowledge/information on which they have been trained. An online updating mechanism that continuously feeds/updates the dynamic models with new data (information) collected from the process would be the solution for such problem.

In this sense, the practical application of the proposed methodology needs to account for the uncertainty or confidence about the model prediction, which should be more reliable when the model is to be used for control and optimization (e.g.: in order to assess how the control actions will tolerate the model predictions errors, or in order to detect that the process is evolving into a new or not well

described working area). Ensemble and Monte-Carlo-based methods are suitable for the uncertainty quantification of data-driven models.

5 Conclusions

This work presents a robust and generic methodology for data-driven multivariate dynamic modelling and multi-step ahead prediction of nonlinear chemical processes using surrogate models. The proposed methodology utilizes surrogate models for building a group of NARX models, each of them able to predict the evolution of one output as a function of the other inputs and outputs of the process. The set of multivariate dynamic models are, then, used to forecast the process outputs along larger time intervals, through a recursive and inter-coordinated prediction scheme. The methodology also offers a new procedure for training data selection for dynamic modeling, based on the “design of computer experiments” technique when a FPM of the process is available.

The application of the proposed methodology is illustrated through three case-studies of nonlinear dynamic processes selected from the process engineering literature, including a bioreactor, three-tanks and an oil-shale pyrolysis batch reactor. The results make explicit the promising capabilities of the developed multivariate dynamic models in terms of: 1) a high prediction accuracy, 2) the capability of simulating complex dynamic profiles over large prediction time horizons, and 3) the generality and robustness required to handle cases of different nature (biological, industrial and chemical systems), integrating different metamodel types (ANN and OK), managing situations based on either FPM approximations or where only a limited set of process input-output signals are available, exhibiting very good behavior with respect to the sensitivity against the training data size and the noise in the training data.

The proposal extends the capabilities of the OK techniques (until now only proposed in simpler dynamic situations) and efficiently introduces them to full dynamic scenarios, showing very competitive characteristics with respect to other leading techniques such as ANNs, in terms of accuracy and, more significantly, in terms of flexibility and systematic tuning of parameters. The only disadvantage is the relatively high computational effort required for fitting.

The sets of multivariate dynamic models provided by the methodology fit very well with the requirements and needs of different engineering applications as model predictive control, dynamic optimization, monitoring, etc., where the future values of many process outputs must be accurately and rapidly predicted.

The good results obtained with models trained with a limited quantity of input-output data justify the generalization of the message and the potential applicability of the proposed procedure to situations

when no FPM is available or the conditions from the training data may significantly change, although this is to be further investigated. On the other hand, the main issues which main appear during the application of the proposed methodology, associated to the availability, representativeness and quality of the training data, and common to the application of machine-learning techniques, represent potential lines of future research, such as the development of online updating method to overcome the process evolution, or the development of prediction assessment methods in dynamic environments.

References

- [1] Nagy, Z. K. Model based control of a yeast fermentation bioreactor using optimally designed artificial neural networks. *Chemical Engineering Journal*, vol. 127, pp. 95-109, 2007.
- [2] Caballero, J. A. and Grossmann, I. E. An algorithm for the use of surrogate models in modular flowsheet optimization. *AIChE Journal*, vol. 54, p. 2633–2650, 2008.
- [3] Ažman, K. and Kocijan, J. Dynamical systems identification using Gaussian process models with incorporated local models. *Engineering Applications of Artificial Intelligence*, Vols. 398-408, p. 24, 2011.
- [4] Davis, E. and Ierapetritou, M. A kriging method for the solution of nonlinear programs with black-box functions. *AIChE journal*, vol. 53, pp. 2001-2012, 2007.
- [5] Davis, E. and Ierapetritou, M. A Kriging-Based Approach to MINLP Containing Black-Box Models and Noise. *Ind. Eng. Chem. Res.*, vol. 47, pp. 6101-6125, 2008.
- [6] Qin, S. J. Survey on data-driven industrial process monitoring and diagnosis. *Annual Reviews in Control*, vol. 36, pp. 220-234, 2012.
- [7] Kajero, O.; Chen, T.; Yao, Y.; Chuang, Y.-C. and Wong, D. S. H. Meta-modelling in chemical process system engineering. *Journal of the Taiwan Institute of Chemical Engineers*, vol. 73, pp. 135-145, 2017.
- [8] Ali, J. M.; Hussain, M. A.; Tade, M. O. and Zhang, J. Artificial Intelligence techniques applied as estimator in chemical process systems – A literature survey. *Expert Systems with Applications*, vol. 42, pp. 5915-5931, 2015.
- [9] Shokry, A.; Vicente, P.; Gerard, E.; Pérez-Moya, M.; Graells, M. and Espuña, A. Data-driven soft-sensors for online monitoring of batch processes with different initial conditions. *Computers & Chemical Engineering*, vol. 118, pp. 159-179, 2018.
- [10] Bradford, E.; Schweidtmann, A. M.; Zhang, D.; Jing, K. and Rio-Chanona, E. A. Dynamic modeling and optimization of sustainable algal production with uncertainty using multivariate Gaussian processes. *Computers & Chemical Engineering*, vol. 118, pp. 143-158, 2018.
- [11] Nelles, O. *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*, Berlin: Springer, 2001.
- [12] Boukouvala, F.; Muzzio, F. J. and Ierapetritou, M. G. Dynamic Data-Driven Modeling of Pharmaceutical Processes. *Ind. Eng. Chem. Res.*, vol. 50, p. 6743–6754, 2011.
- [13] Baraldi, P.; Cadini, F.; Mangili, F. and Zio, E. Model-based and data-driven prognostics under different available information. *Probabilistic Engineering Mechanics*, vol. 32, pp. 66-79, 2013.

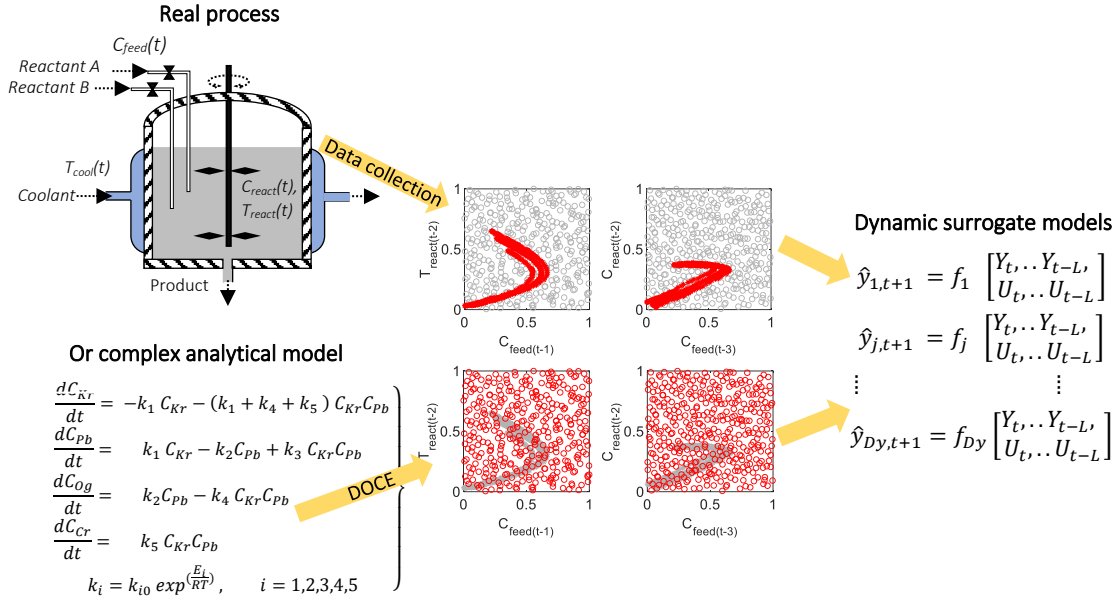
- [14] Amozeghar, M. and Khorasani, K. An ensemble of dynamic neural network identifiers for fault detection and isolation of gas turbine engines. *Neural Networks*, vol. 76, pp. 106 -121, 2016.
- [15] Tsai, C.-S. and Chang, C.-T. Dynamic process diagnosis via integrated neural networks. *Computers & Chemical Engineering*, vol. 19, pp. 747-752, 1995.
- [16] Adebisi, O. A. and Corripio, A. B. Dynamic neural networks partial least squares (DNNPLS) identification of multivariable processes. *Computers and Chemical Engineering*, vol. 27, pp. 143-155, 2003.
- [17] Banu, U. S. and Umab, G. ANFIS based sensor fault detection for continuous stirred tank reactor. *Applied Soft Computing*, vol. 11, pp. 2618-2624, 2011.
- [18] Zhou, L.; Chen, J. and Song, Z. Recursive Gaussian Process Regression Model for Adaptive Quality Monitoring in Batch Processes. *Mathematical Problems in Engineering*, vol. 2015, pp. 1-9, 2015.
- [19] Mattosa, C. L. C.; Dai, Z.; Damianou, A.; Barreto, G. A. and Lawrence, N. D. Deep recurrent Gaussian processes for outlier-robust system identification. *Journal of Process Control*, vol. 60, pp. 82-94, 2017.
- [20] Himmelblau, D. M. Applications of artificial neural networks in chemical engineering. *Korean Journal of Chemical Engineering*, vol. 17, p. 373-392, 2000.
- [21] Poznyak, A.; Chairez, I. and Poznyak, T. A survey on artificial neural networks application for identification and control in environmental engineering: Biological and chemical systems with uncertain models. *Annual Reviews in Control (in press <https://doi.org/10.1016/j.arcontrol.2019.07.003>)*, 2019.
- [22] Dua, V. A mixed-integer programming approach for optimal configuration of artificial neural networks. *Chemical Engineering Research and Design*, vol. 88, pp. 55-60, 2010.
- [23] Ludermir, T. B.; Yamazaki, A. and Zanchettin, C. An Optimization Methodology for Neural Network Weights and Architectures. *IEEE Transactions on Neural Networks*, vol. 17, pp. 1452-1459, 2006.
- [24] Benardos, P. and Vosniako, G.-C. Optimizing feedforward artificial neural network architecture. *Engineering Applications of Artificial Intelligence*, vol. 20, pp. 365-382, 2007.
- [25] Leperi, K. T.; Yancy-Caballero, D.; Snurr, R. Q. and You, F. 110th Anniversary : Surrogate Models Based on Artificial Neural Networks To Simulate and Optimize Pressure Swing Adsorption Cycles for CO₂ Capture. *Ind. Eng. Chem. Res.*, Vols. 58, 39, pp. 18241-18252, 2019.
- [26] Masters, T. Practical Neural Network Recipes in C++, San Diego New York: Academic Press, 1993.
- [27] Rigamonti, M.; Baraldi, P.; Zio, E.; Roychoudhury, I.; Goebel, K. and Poll, S. Ensemble of optimized echo state networks for remaining useful life prediction. *Neurocomputing*, vol. 281, pp. 121-138, 2018.
- [28] Godarzi, A. A.; Amiri, R. M.; Talaei, A. and Jamasb, T. Predicting oil price movements: A dynamic Artificial Neural Network approach. *Energy Policy*, Vols. 371-382, p. 68, 2014.
- [29] Panapakidis, I. P. and Dagoumas, A. S. Day-ahead electricity price forecasting via the application of artificial neural network based models. *Applied Energy*, vol. 172, pp. 132-151, 2016.
- [30] Sadeghassadi, M.; Macnab, C. J. B.; Gopaluni, B. and Westwick, D. Application of neural networks for optimal-setpoint design and MPC control in biological wastewater treatment. *Computers & Chemical Engineering*, vol. 115, pp. 150-160, 2018.
- [31] Xu, D.; Jiang, B. and Shi, P. Adaptive Observer Based Data-Driven Control for Nonlinear Discrete-Time Processes. *IEEE Transactions on Automation Science and Engineering*, vol. 11, pp. 1037-1045, 2014.
- [32] Caccavale, F.; Digiulio, P.; Iamarino, M.; Masi, S. and Pierri, F. A neural network approach for on-line fault detection of nitrogen sensors in alternated active sludge treatment plants. *Water Science & Technology*, vol. 62, pp. 2760-2768, 2010.
- [33] Li, S. and Li, Y. Neural network based nonlinear model predictive control for an intensified continuous reactor. *Chemical Engineering and Processing: Process Intensification*, vol. 96, pp. 14-27, 2015.
- [34] Lee, W. J.; Na, J.; Kim, K.; Lee, C.-J.; Lee, Y. and Lee, J. NARX modeling for real-time optimization of air

- and gas compression systems in chemical processes. *Computers & Chemical Engineering*, vol. 115, pp. 262-274, 2018.
- [35] O'Hagan, A. and Kingman, J. F. C. Curve Fitting and Optimal Design for Prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 40, pp. 1-42, 1978.
- [36] O'Hagan, A.; Kennedy, M. C. and Oakley, J. E. Uncertainty analysis and other inference tools for complex computer codes. In *Bayesian Statistics 6*, (J. M. Bernardo et al (eds.)), Oxford University Press, pp. 503-524, 1999.
- [37] Deisenroth, M. P.; Rasmussen, C. E. and Peters, J. Gaussian process dynamic programming. *Neurocomputing*, vol. 72, pp. 1508-1524, 2009.
- [38] Ažman, K. and Kocijan, J. Application of Gaussian processes for black-box modelling of biosystems. *ISA Transactions*, vol. 46, pp. 443-457, 2007.
- [39] Rasmussen, C. E. and Williams, C. K. I. Gaussian Processes for Machine Learning. Adaptive Computation and Machine Learning series, Cambridge, Massachusetts: MIT Press, 2006.
- [40] Murray-Smith, R.; Sbarbaro, D.; Rasmussen, C. E. and Girard, A. Adaptive, cautious, predictive control with Gaussian process priors. *IFAC Proceedings Volumes*, vol. 36, pp. 1155-1160, 2003.
- [41] Kocijan, J.; Girard, A.; Banko, B. and Murray-Smith, R. Dynamic systems identification with Gaussian processes. *Mathematical and Computer Modelling of Dynamical Systems*, vol. 11, pp. 411-424, 2005.
- [42] Girard, A.; Rasmussen, C. E. R. and Murray-Smith, R. Gaussian Process priors with Uncertain Inputs: Multiple-Step-Ahead Prediction. *Technical Report DCS TR-2002-119*, University of Glasgow, Glasgow, 2002.
- [43] Rasmussen, C. E. and Deisenroth, M. P. Probabilistic Inference for Fast Learning in Control. 229-242, 2008.
- [44] Forrester, A.; Sobester, A. and Keane, A. Engineering Design via Surrogate Modelling: A Practical Guide, Southampton, UK: John Wiley and Sons, 2008.
- [45] Wang, Z. and Ierapetritou, M. A Novel Surrogate-Based Optimization Method for Black-Box Simulation with Heteroscedastic Noise. *Ind. Eng. Chem. Res.*, vol. 56, pp. 10720-10732, 2017.
- [46] Beck, J.; Friedrich, D.; Brandani, S. and Frag, E. S. Multi-objective optimisation using surrogate models for the design of VPSA systems. *Computers & Chemical Engineering*, vol. 82, pp. 318-329, 2015.
- [47] Egea, J. A.; Vries, D.; Alonso, A. A. and Banga, J. R. Global Optimization for Integrated Design and Control of Computationally Expensive Process Models. *Ind. Eng. Chem. Res.*, vol. 46, pp. 9148-9157, 2007.
- [48] Hernandez, A. F. and Grover, M. A. Stochastic dynamic predictions using Gaussian process models for nanoparticle synthesis. *Computers & Chemical Engineering*, vol. 34, pp. 1953-1961, 2010.
- [49] Espinosa, J. J. and Vandewalle, J. Fuzzy Modeling and Identification, A guide for the user. 1998a.
- [50] Espinosa, J. J. and Vandewalle, J. Predictive control using fuzzy models applied to a steam generating unit. *Proceedings of the 3rd International Workshop on Fuzzy Logic and Intelligent Technologies for Nuclear Science and Industry*, 1998b.
- [51] Shokry, A. and Espuña, A. Sequential Dynamic Optimization of Complex Nonlinear Processes based on Kriging Surrogate Models. *Procedia Technology*, vol. 15, p. 376-387, 2014b.
- [52] Shokry, A.; Ardakani, Escudero, M. H.; Graells, M. and Espuña, A. Dynamic Kriging based Fault Detection and Diagnosis Approach for Nonlinear Noisy Dynamic Processes. *Computers & Chemical Engineering*, Vols. 758-776, p. 106, 2017a.
- [53] Shokry, A.; Pérez-Moya, M.; Graells, M. and Espuña, A. Data-Driven Dynamic Modeling of Batch Processes Having Different Initial Conditions and Missing Measurements. *Computer Aided Chemical Engineering*, vol. 40, pp. 433-438, 2017b.

- [54] Wang, S.; Zhou, L.; Ji, X.; Karimi, I. A.; He, G.; Dang, Y. and Xu, X. A Surrogate-Assisted Approach for the Optimal Synthesis of Refinery Hydrogen Networks. *Ind. Eng. Chem. Res.*, vol. 58, pp. 16798-16812, 2019.
- [55] Fang, K.-T.; Li, R. and Sudjianto, A. Design and modelling for computer experiment, New York: Chapman and Hall/CRC, 2005.
- [56] Jurecka, F. Robust Design Optimization Based on Metamodeling Techniques, München: PhD thesis, Technische Universität München, 2007.
- [57] Garud, S. S.; Karimi, I. A. and Kraftb, M. Design of computer experiments: A review. *Computers & Chemical Engineering*, vol. 106, pp. 71-95, 2017.
- [58] Ibrahim, M.; Al-Sobhi, S.; Mukherjee, R. and AlNouss, A. Impact of Sampling Technique on the Performance of Surrogate Models Generated with Artificial Neural Network (ANN): A Case Study for a Natural Gas Stabilization Unit. *Energies*, vol. 12, pp. 1-12, 2019.
- [59] Joseph, V. R. Space-filling designs for computer experiments: A review. *Quality Engineering*, vol. 28, pp. 28-35, 2016.
- [60] Jones, D. R.; Schonlau, M. and Welch, W. J. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, vol. 13, pp. 455-492, 1998.
- [61] Forrester, A. and Keane, A. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, vol. 45, pp. 50-79, 2009.
- [62] Conti, S.; Gosling, J. P.; Oakley, J. and O'Hagan, A. Gaussian process emulation of dynamic computer codes. *Biometrika*, vol. 96, pp. 663-676, 2009.
- [63] Cho, J.; Principe, J. C.; Erdogmus, D. and Motter, M. A. Quasi-sliding mode control strategy based on multiple-linear models. *Neurocomputing*, vol. 10, pp. 960-974, 2007.
- [64] Suykens, J. A.; Vandewalle, J. and De Moor, B. L. Artificial Neural Networks for Modelling and Control of Non-Linear Systems, 1 ed., Springer US, 1996.
- [65] Gauthier, J. P.; Hammouri, H. and Othman, S. A simple observer for nonlinear systems applications to bioreactors. *IEEE Transactions on Automatic Control*, vol. 37, pp. 875-880, 1992.
- [66] Frank, P. and Ding, X. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of Process Control*, vol. 17, pp. 403-424, 1997.
- [67] Kouadri, A.; Aitouche, M. A. and Zelmat, M. Variogram-based fault diagnosis in an interconnected tank system. *ISA Transactions*, vol. 51, p. 471-476, 2012.
- [68] Sarailo, M.; Rahmani, Z. and Rezaie, B. A novel model predictive control scheme based on bees algorithm in A class of nonlinear systems: application to A three tank system. *Neurocomputing*, vol. 152, p. 294-304, 2015.
- [69] Patton, R.; Chen, J. and Siew, T. Fault diagnosis in nonlinear dynamic systems via neural networks. *Proc. of IEE Int. Conf. on Control*, pp. 1346-1351, 1994.
- [70] Carrasco, E. F. and Banga, J. R. Dynamic Optimization of Batch Reactors Using Adaptive Stochastic Algorithms. *Ind. Eng. Chem. Res.*, vol. 36, p. 2252-2261, 1997.
- [71] Wen, C. and Yen, T. Optimization of oil shale pyrolysis. *Chemical Engineering Science*, vol. 32, pp. 346-349, 1977.
- [72] Theilliol, D.; Noura, H. and Ponsart, J.-C. Fault diagnosis and accommodation of a three-tank system based on analytical redundancy. *ISA Transactions*, vol. 41, pp. 365-382, 2002.
- [73] Wang, P.; Zhenzhou, L. and Tang, Z. An application of the Kriging method in global sensitivity analysis with parameter uncertainty. *Applied Mathematical Modelling*, vol. 37, pp. 6543-6555, 2013.
- [74] Simpson, T. W.; Lin, D. K. J. and Chen, W. Sampling strategies for computer experiments: design and

- analysis. *International Journal of Reliability and Applications*, vol. 2, pp. 209-240, 2001.
- [75] Sacks, J.; Schiller, S. B. and Welch, W. J. Design for computer experiments. *Technometrics*, vol. 31, pp. 41-47, 1989.
- [76] Krige, D. G. A statistical approach to some mine valuations and allied problems at the Witwatersrand, Johannesburg: Master thesis, University of the witwatersrand, 1951.
- [77] Fisher, R. A. The Design of Experiments, 9 ed., London: Macmillan Pub. Co., 1971.
- [78] Fisher, B. J. R.A. Fisher and the design of experiments (1922-1929). *The American Statistician*, vol. 34, pp. 1-7, 1980.
- [79] Matheron, G. Principles of geostatistics. *Economic Geology*, vol. 58, p. 1246–1266, 1963.
- [80] Shokry, A. and Espuña, A. Applying Metamodels and Sequential Sampling for Constrained Optimization of Process Operations. *Lecture Notes in Computer Science*, vol. 8468, pp. 396-407, 2014a.
- [81] Cadini, F.; Agliardi, G. L. and Zio, E. Estimation of rare event probabilities in power transmission networks subject to cascading failures. *Reliability Engineering & System Safety*, vol. 158, pp. 9-20, 2017.
- [82] Turati, P.; Pedronia, N. and Zio, E. Simulation-based exploration of high-dimensional system models for identifying unexpected events. *Reliability Engineering & System Safety*, vol. 158, pp. 9-20, 2017.
- [83] Cadini, F.; Gioletta, A. and Zio, E. Improved metamodel-based importance sampling for the performance assessment of radioactive waste repositories. *Reliability Engineering & System Safety*, vol. 134, pp. 188-197, 2015.
- [84] Zio, E.; Di Maio, F. and Stasi, M. A data-driven approach for predicting failure scenarios in nuclear systems. *Annals of Nuclear Energy*, vol. 37, pp. 482-491, 2010.
- [85] Subraveti, S. G.; Li, Z.; Prasad, V. and Rajendran, A. Machine Learning-Based Multiobjective Optimization of Pressure Swing Adsorption. *Ind. Eng. Chem. Res.*, Vols. 58, 44, pp. 20412-20422, 2019.

For Table of Contents Only



Multistep-ahead predictions

