ORIGINAL ARTICLE

Model reduction, machine learning based global optimisation for large-scale steady state nonlinear systems

Min Tao | Panagiotis Petsagkourakis | Jie Li | Constantinos Theodoropoulos

Department of Chemical Engineering, The University of Manchester, M13 9PL, UK

[1]Department of Chemical Engineering, The University of Manchester, M13 9PL, UK

Correspondence
Constantinos Theodoropoulos, Department of Chemical Engineering, The University of Manchester, M13 9PL, UK
Email: k.theodoropoulos@manchester.ac.uk

Funding information
The University of Manchester and China Scholarship Council joint scholarship (file no. 201706250031)

Many engineering processes can be accurately modelled using partial differential equations (PDEs), but high dimensionality and non-convexity of the resulting systems pose limitations on their efficient optimisation. In this work, a model reduction, machine-learning methodology combining principal component analysis (PCA) and artificial neural networks (ANNs) is employed to construct a reduced surrogate model, which can then be utilised by advanced deterministic global optimisation algorithms to compute global optimal solutions with theoretical guarantees. However, such optimisation would still be time-consuming due to the high non-convexity of the activation functions inside the reduced ANN structures. To develop a computationally-efficient optimisation framework, we propose two alternative strategies: The first one is a piecewise-affine reformulation of the nonlinear ANN activation functions, while the second one is based on deep rectifier neural networks with ReLU activation function. The performance of the proposed framework is demonstrated through three illustrative case studies.

KEYWORDS
Model reduction, Distributed parameter systems, Piecewise

affine reformulation, Deep rectifier neural networks, Data-driven optimisation, Machine learning, Global optimisation

1 | INTRODUCTION

Partial differential equation (PDE)-based process models, also termed distributed-parameter systems, have wide applicability in industrial engineering areas [1], such as chemical [2], biochemical [3], and mechanical engineering [4] and aerodynamics [5]. However, complex PDEs are inherently high-dimensional and non-convex, including multiple local optima, hence resulting in intensive computational costs when the computation of global optima is sought. Moreover, most of the generic commercial PDE simulators [6, 7] are essentially black-box and offer no optimisation options. Even if complex model codes are accessible in open-source software (e.g. [8]), the cost of direct optimisation is often unacceptable. To date, performing optimisation tasks efficiently for large-scale complex systems, is still a challenge in engineering design.

A promising way to deal with high dimensionality is to use projective model order reduction methods, which reduce the complexity of detailed models but preserve their main input-output features [9]. The popular principal component analysis (PCA) strategy, an efficient dimensionality reduction technique in data science [10], also termed as Karhunen-Loeve decomposition or proper orthogonal decomposition (POD), is usually combined with projection and/or surrogate model approaches to construct reduced models. POD together with Galerkin projection is capable of producing high-fidelity low-dimensional models for optimisation tasks [11]. Similarly, the combination of POD with machine-learing Artifical Neural Networks (ANN) has been used to construct reduced surrogate models for black-box large-scale dynamic systems, resulting in efficient optimisation and control strategies [12]. In addition, PCA and Kriging models have been utilised to replace complex process models [13].

Furthermore, *equation-free* methodologies offer another effective model reduction approach for large-scale black-box systems, for optimisation and control purposes. Exploiting the dominant eigendirections of the outputs of complex black-box system models, or direct historical system data, low-dimensional reduced Jacobian and Hessian matrixes can be computed. An equation-free based reduced SQP method was proposed exploiting the computation of low-dimensional Jacobians and Hessians, to accelerate the optimisation procedure for large-scale steady state nonlinear systems [14]. An aggregation function was subsequently applied to address general nonlinear inequality constraints, extending the scope and capability of equation-free reduced SQP methods [15]. Furthermore, equation-free based dynamic optimisation and control methods have also been constructed [16, 17]. An extensive discussion about model reduction based optimisation methodologies can be found in [18].

To address non-convexity in complex nonlinear optimisation problems, both stochastic and deterministic algorithms can be utilised. Stochastic search methods, such as simulated annealing [19] and genetic algorithms [20], can globally explore the feasible solution space avoiding local optima. However, such stochastic search algorithms are slow for large-scale problems and offer no theoretical guarantees on the global optimality of the computed solutions. Deterministic global optimisation methods are capable of computing global optima utilising branch-and-bound techniques [21], but they are often computationally intensive for large-scale systems due to the need for multiple evaluations of the lower bounds of the optimisation problems.

The aim of this work is to construct an efficient deterministic global optimisation framework for large-scale steady-state input/output (black-box) systems combining model reduction with machine-learning methodologies. Often a single model reduction technique cannot easily deal with the complexities of large-scale nonlinear systems. For example, although optimal principal component regressions (PCRs) [22] are popular to deal with high dimensional input-output

data, the linear or low-complex models are not accurate enough to replace high nonlinear complex system models. POD on the other hand, is a very powerful method, but projecting the original system onto the global POD modes is not always easy and requires full knowledge of the full-scale system model. Meanwhile, ANN models use machine learning to capture highly nonlinear behaviours but usually require large-scale ANN structures (increasing number of neurons and layers) due to the high dimensionality of the original systems. Combining model reduction techniques, e.g. principal component analysis (PCA) with artificial neural networks (ANNs)[23], can produce accurate reduced surrogate models. Then such reduced ANN models can be explicitly utilised by global general-purpose optimisation solvers.

Nevertheless, performing global optimisation tasks with general ANN models is still time consuming (even for reduced ANNs), hence most existing research focuses on local optimisation and/or small-scale problems. Surrogate ANN models have been used to replace superstructure process models and have been optimised locally [24, 25]. Small-scale ANN models (1 hidden layer, 3 neurons) were constructed and optimised globally by the advanced global solver BARON [26]. Larger ANN models are more expensive to optimize as high non-convexity often requires the repeated use of branch-and-bound algorithms. A reduced space-based global optimisation method, recently proposed by Schweidtmann and Mitsos [27], projected the iteration space of non-convex variables onto the subspace of dependent variables, resulting in small-size sub-problems and, consequently, in significant computational savings.

In this work, two strategies are adopted to construct efficient reduced models in a PCA-ANN global optimisation framework. The first is a piecewise affine (PWA) reformulation technique of the nonlinear ANN activation function, while the second is the use of a deep rectifier neural network. It should be noted that this work extends previous preliminary findings of the authors [28].

The rest of the paper is organized as follows. In Section 2, the basic PCA-ANN global optimisation framework is proposed and the detailed theoretical basis and implementation are provided. In Section 3, the PWA-based reformulation is outlined and illustrated with an example. In Section 4, the deep rectifier ANN-based improvement is employed in the optimisation framework and validated using a large-scale combustion case study. In Section 5, conclusions and further applications are discussed.

2 | PROBLEM FORMULATION

In this work, a model reduction-based optimisation framework is presented to deal with large-scale nonlinear steady-state systems focusing on the optimisation of spatially distributed processes, described by sets of dissipative PDEs of the form:

$$\frac{\partial \boldsymbol{y}}{\partial t} = D\{\frac{\partial \boldsymbol{y}}{\partial x}, \frac{\partial^2 \boldsymbol{y}}{\partial x^2}, ..., \frac{\partial^n \boldsymbol{y}}{\partial x^n}, \boldsymbol{d}\} + R(\boldsymbol{d}, \boldsymbol{y}) \tag{1}$$

Here $t \in \mathbb{R}$ denotes time, $x \in \mathbb{R}^{N_x}$, N_x the spatial dimensions, N_x=1,2, or 3. $D \in \mathbb{R}$ is the dissipative spatial differential operator, $\boldsymbol{d} \in \mathbb{R}^{N_d}$ the parameter variables and $\boldsymbol{y} \in \mathbb{R}^{N_y}$ a set of state variables, $R(\boldsymbol{d}, \boldsymbol{y}) : \mathbb{R}^{N_d} \times \mathbb{R}^{N_y} \rightarrow \mathbb{R}^{N_y}$ are the nonlinear terms. Considering steady state systems and assuming that $\boldsymbol{y}(t, x) \rightarrow \boldsymbol{y}(x)$, and $\partial \boldsymbol{y}/\partial t = 0$, the above equations become:

$$0 = D\{\frac{\partial \boldsymbol{y}}{\partial x}, \frac{\partial^2 \boldsymbol{y}}{\partial x^2}, ..., \frac{\partial^n \boldsymbol{y}}{\partial x^n}, \boldsymbol{d}\} + R(\boldsymbol{d}, \boldsymbol{y}) \tag{2}$$

Therefore, the general optimisation problems for steady state PDE-based systems can be formulated as the fol-

lowing problem $\boldsymbol{P1}$:

$$
\begin{aligned}
(\boldsymbol{P1}) \quad & \min_{\boldsymbol{d}} \quad G(\boldsymbol{d}, \boldsymbol{y}) \\
& s.t. 0 = D\{\frac{\partial \boldsymbol{y}}{\partial x}, \frac{\partial^2 \boldsymbol{y}}{\partial x^2}, ..., \frac{\partial^n \boldsymbol{y}}{\partial x^n}, \boldsymbol{d}\} + R(\boldsymbol{d}, \boldsymbol{y}) \\
& \quad A\{\frac{\partial \boldsymbol{y}}{\partial x}, \frac{\partial^2 \boldsymbol{y}}{\partial x^2}, ..., \frac{\partial^n \boldsymbol{y}}{\partial x^n}\}\Big|_{x=\Omega} = h_{bds}(\boldsymbol{d}, \boldsymbol{y}) \\
& \quad \boldsymbol{g}_{cons}(\boldsymbol{d}, \boldsymbol{y}) \leq 0
\end{aligned}
\tag{3}
$$

where $G(\boldsymbol{d}, \boldsymbol{y}) : \mathbb{R}^{N_d} \times \mathbb{R}^{N_y} \rightarrow \mathbb{R}$ is the objective function. The equality constraints are the system PDEs, which in the case of a black-box system are not explicitly available, with corresponding boundary conditions. $h_{bds}(\boldsymbol{d}, \boldsymbol{y}) : \mathbb{R}^{N_d} \times \mathbb{R}^{N_y} \rightarrow \mathbb{R}^{N_y}$ are the right hand sides of the boundary conditions, A is the operator of the boundary condition equations, Ω are the boundaries and $\boldsymbol{g}_{cons}(\boldsymbol{d}, \boldsymbol{y}) : \mathbb{R}^{N_d} \times \mathbb{R}^{N_y} \rightarrow \mathbb{R}^{N_y}$ denote other general constraints, e.g. bounds and other constraints, for the state variables $\boldsymbol{y}$ and the design parameter variables $\boldsymbol{d}$.

In general, the unavailability of system equations inside commercial software prohibits the use of direct model-based optimisation techniques. Even in the case that large-scale system equations are available, the optimisation problem $\boldsymbol{P1}$ can not be efficiently handled by global optimisation algorithms [29]. In this work, this barrier is overcome by employing accurate surrogate models to formulate a highly accurate approximate problem $\boldsymbol{P2}$, which is then utilised by a general purpose global optimisation solver.

If we use explicit surrogate inputs-outputs to replace the black-box system equations in the above formulation, then problem $\boldsymbol{P1}$ can be transformed into the following problem $\boldsymbol{P2}$:

$$
\begin{aligned}
(\boldsymbol{P2}) \quad & \min_{\boldsymbol{d}} \quad G(\boldsymbol{d}, \boldsymbol{y'}) \\
& s.t. \boldsymbol{y'} = F(\boldsymbol{d}) \\
& \quad \boldsymbol{g}_{cons}(\boldsymbol{d}, \boldsymbol{y'}) \leq 0
\end{aligned}
\tag{4}
$$

where $\boldsymbol{y'}$ are the outputs of the surrogate model and F the black-box nonlinear operator.

The gap between problems $\boldsymbol{P1}$ and $\boldsymbol{P2}$ can be measured by the output errors $(\boldsymbol{y} - \boldsymbol{y'})$. If these errors are small enough, the global optima of the reduced surrogate model-based problems $\boldsymbol{P2}$ will be close to the global solutions of the original problem $\boldsymbol{P1}$. Thus, accurate surrogate models are key to guarantee small errors across the design domain. The accuracy of surrogate models depends on sampling quality, machine-learning techniques employed and quantity and model building techniques. This work assumes that enough representative/informative samples are available for building accurate models. Details about efficient sampling approaches are discussed in Section 2.1.

Then model building procedures directly affect the accuracy and the complexity of constructed surrogate models, which in turn have a significant effect on the computational accuracy and speed of the subsequent deterministic global optimisation.

This work employs a double model reduction process in conjunction with machine learning, through a combination of PCA, ANN and reformulation techniques to generate an accurate reduced model, which is subsequently used to compute near global solutions for the original problem $\boldsymbol{P1}$. In the following sections, we are discussing the basic components of our PCA-ANN-global optimisation methodology.

2.1 | Sampling and data collection

To build accurate surrogate models, suitable sampling methods are needed to collect highly representative samples for a range of design variables. Inefficient sampling strategies, including too few samples and/or unrepresentative sampling, would result in inaccurate reduced models, in turn producing inaccurate optimal solutions. While provably representative sampling is still an open problem, there are several popular sampling techniques such as Hammersley sequences [30], D-optimal designs [31] and Latin Hypercube (LHC)[32] that can produce good quality results. Hammersley sequences employ quasi-deterministic sequences to convergence to a set of *informative* samples. The D-optimal design approach aims to reduce the number of experimental runs and maximise sample variances. The LHC method can produce samples covering the whole design space and maximize the difference among the generated samples. Specifically, the sample domain is divided into many sub-domains where sample points are generated randomly in order to represent the specific sub-domain. In this work, we choose LHC because it has been shown to be able to fill the design space and to capture input/output relationships given an *adequate* number of samples. The number of LHC samples is decided by testing the model accuracy. In general, more LHC samples are more likely to contain the information needed to capture complex input/output relationships. LHC sampling for complex systems often requires a relatively large number of samples, which is also the pre-condition to perform successful PCA reduction and ANN-based surrogate model construction.

In the presence of constraints (such as $\boldsymbol{g}_{cons}$ here), it is hard for the LHC algorithm to directly capture the complex design space. Previous work [33] employed constraints to filter the LHC samples in order to reduce function evaluations for expensive systems. In addition, a complex strategy was used to first decompose the design space into many subdomains, where system features were represented through multiple low-fidelity models. Nevertheless the adaptive optimisation performed within the constrained sampling strategy can be computationally intensive for high-dimensional outputs $\boldsymbol{y}$ and/or inequality constraints for $\boldsymbol{y}$.

In this work, we utilised universal ANN surrogate models to capture the nonlinear behaviour of the black-box PDE equality in its entirety. The constrained sampling strategy may possibly lead to a discontinuous design space, requiring much larger ANN structures to capture it [34]. Hence, we separated the expensive black-box PDE-based equality constraints from the "known" inequality constraints. Our aim is to construct accurate but simple ANN models to replace the PDE-based equality constraints, which together with the known inequality constraints $\boldsymbol{g}_{cons}$, provide a highly accurate explicit model formulation $\boldsymbol{P2}$ to the general-purpose global optimisation solvers.

While building accurate surrogate models requires enough representative/informative samples, too many samples would lead to intensive computations for high-dimensional systems. Improving sampling efficiency can significantly reduce computational times. Exploiting process knowledge or advanced adaptive sampling approaches may help to achieve this goal. Prior knowledge about the processes can provide useful information to collect representative samples with higher probability. However, this requires case-by-case detailed experience about the black-box systems. Adaptive sampling uses a few prior input/output samples to subsequently generate representative samples through solving a set of optimisation problems. Most relevant previous studies in literature [35, 36] deal with low-dimensional inputs/outputs, leading to small-size optimisation problems. This work, however, deals with high-dimensional outputs $\boldsymbol{y}$, possibly together with large numbers of inequality constraints $\boldsymbol{g}_{cons}$, hence the computation costs for performing adaptive sampling could be high. Nevertheless adaptive sampling procedures are fully compatible with the algorithms developed here provided that the relevant computations can be appropriately reduced.

Here we consider a more general approach without exploiting process knowledge and adaptive sampling techniques. The assumption is that the sampling process takes place offline and does not directly affect the computational efficiency of the online optimisation computations. Nevertheless, the proposed model reduction based global opti-

misation framework can be easily combined with prior knowledge and adaptive sampling approaches, as mentioned above to speed-up the offline parts of the computations.

We collect samples across the space of design parameters $\boldsymbol{d}$ and corresponding input-output data sets ($\boldsymbol{D} \in \mathbb{R}^{N_d \times N}$, $\boldsymbol{Y} \in \mathbb{R}^{m \times N}$), where $m \in \mathbb{N}$ is the number of discrete interval points, which for distributed parameter systems tends to be be a large number, and $N \in \mathbb{N}$ is the number of samples. The obtained data sets ($\boldsymbol{D}$, $\boldsymbol{Y}$) are then used to construct accurate reduced surrogate models through the combination of PCA and ANN.

2.2 | Principal Component Analysis (PCA)

Due to the high dimensionality of spatially discrete output data $\boldsymbol{Y}$, directly constructing surrogate ANN models would result in large ANN structures. Here, the popular PCA method is first employed to build a reduced model from output data $\boldsymbol{Y}$.

A sampling method (here LHC as discussed in the previous section) is firstly employed to construct a data ensemble $\boldsymbol{Y}$ over a finite spatial interval $\Omega' \in \mathbb{R}$. PCA then calculates a reduced output $\boldsymbol{U}$, by projecting the data sample **Y** onto the subspace of the "small" set of principal components (PCs) $P = (p_1, p_2, ..., p_k)$, $k \in \mathbb{N}$ being the number of PCs.

$$\boldsymbol{U} = \boldsymbol{P}\boldsymbol{Y} \tag{5}$$

Here $\boldsymbol{U} \in \mathbb{R}^{k \times N}$ is the projection of the original data $\boldsymbol{Y}$ onto the reduced subspace $\mathbb{P}$ and $\boldsymbol{P} \in \mathbb{R}^{k \times m}$ is the corresponding orthogonal projector. In the PCA method the matric $\boldsymbol{P}$ is constructed through the covariance matrix, $\boldsymbol{C_y} \in \mathbb{R}^{m \times m}$ of the output data $\boldsymbol{Y}$:

$$\boldsymbol{C_y} = \frac{1}{m-1}\boldsymbol{Y}\boldsymbol{Y^T} \tag{6}$$

Here we seek to minimise covariance between data and maximise variance i.e. minimise the off-diagonal elements of $\boldsymbol{C_y}$, while maximising its diagonal elements. This is equivalent to performing singular value decomposition (SVD) on $\boldsymbol{C_y}$:

$$\boldsymbol{C_y} = \boldsymbol{Z^T Z} = (\frac{1}{\sqrt{m-1}}\boldsymbol{Y}^T)^T(\frac{1}{\sqrt{m-1}}\boldsymbol{Y}^T) = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V^T} \tag{7}$$

where $D \in \mathbb{R}^{m \times m}$ is a diagonal matrix whose diagonal elements are the eigenvalues of $Z^T Z$ and V is the orthogonal matrix whose columns are the eigenvectors of $Z^T Z$, which as can be easily shown are equivalent to the principal components of $\boldsymbol{Y}$. In fact we can keep the first k PCs corresponding to the k dominant eigenvalues of $\boldsymbol{C_y}$, where usually $k << m$, hence $\boldsymbol{V} \in \mathbb{R}^{m \times k}$ and D now contains only the k most dominant eigenvalues of the system, $D \in \mathbb{R}^{k \times k}$. We can then set $\boldsymbol{P} = \boldsymbol{V}^T$ and perform data reduction through the projection in Eq. 5. The original data sample, $\boldsymbol{Y}$ can be reconstructed from the projected data:

$$\boldsymbol{Y} = \boldsymbol{P^T U} \tag{8}$$

More details about the theory and application of PCA can be found in [37, 38, 39, 40, 41].

The PCA step aims to project the high dimensional output states arising from the discretisation of the PDE system,

onto a small number of dominant variables. Then the resulting low-dimensional relationship between inputs and projected outputs can be captured through small-size ANN structures, hence producing efficient doubly reduced models that will significantly reduce deterministic global optimisation computations. Implementing the PCA step is not always easy, as it is very sensitive to the quantity and quality of samples. Only with enough representative/informative samples can PCA be efficiently performed and globally capture the accurate dominant PCs.

2.3 | Artificial Neural Networks (ANNs)

We employ ANNs on the reduced models (Eq.5) from the PCA step. ANN-based models are chosen due to both successful practices and proven theoretical support that a *shallow* feed-forward neural network with one single layer is sufficient to represent any smooth function [42]. Furthermore, advanced optimisation algorithms have been developed to handle the machine-learning component of the process, i.e. handling the the manipulated variables for ANN structures, such as Levenberg-Marquardt backpropagation [43] and Bayesian regularization backpropagation [44]. Fig.1 shows a conventional feed forward neural network with a hyperbolic tangent activation function $tanh(\cdot)$.

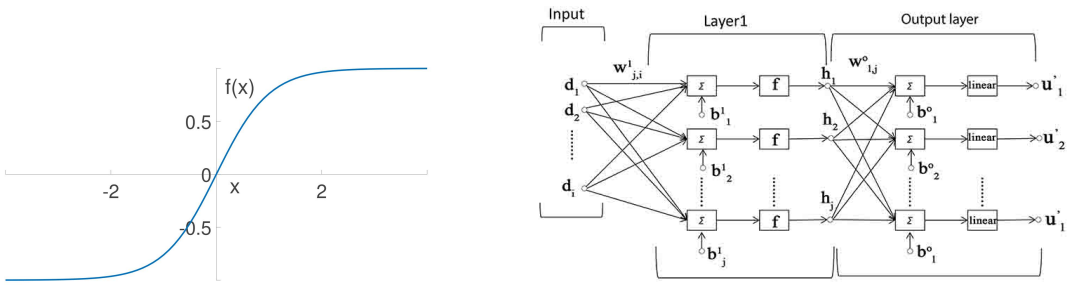


FIGURE 1 Feed-forward neural network with hyperbolic tangent activation function

Shallow ANNs, as the one displayed in Fig.1, are implemented in our basic PCA-ANN global optimisation framework. The feed-forward ANN contains three main components: The input layer, the hidden layer (only one in a shallow ANN) and the output layer, which sequentially perform transformations on the input variables. The input variables, $\boldsymbol{d} = (d_1, d_2, ..., d_{N_d})$, are first linearly transformed and then non-linearly activated through the hidden layer, and further forced by linear transformation and sequential activation in the output layer, to finally formulate the output variables $\boldsymbol{u'} = (u'_1, u'_2, ..., u'_k), \in \mathbb{R}^k$. The mathematical description is given in eq. (9) :

$$
\begin{aligned}
h_j &= f\Big(\sum_{i=1}^{N_d} w^1_{j,i} d_i + b^1_j\Big), \qquad \forall j \in \{1, 2, ..., n\} \\
u'_l &= \sum_{j=1}^{n} w^o_{l,j} h_j + b^o_l, \qquad\quad \forall l \in \{1, 2, ..., k\}
\end{aligned}
\tag{9}
$$

Here $h_j \in \mathbb{R}$ is the output value from the hidden layer with $n \in \mathbb{N}$ neurons, $j = 1, \ldots, n$ and $f \in \mathbb{R}$ is the activation function. Each neuron j contains two parameters: weights $w^1_{j,i} \in \mathbb{R}$ and biases $b^1_j \in \mathbb{R}$ which perform linear transformations. Similarly, $u'_l \in \mathbb{R}$ is the final value from the output layer with k neurons, $l = 1, \ldots, k$, including weights $w^o_{l,j} \in \mathbb{R}$ and biases $b^o_l \in \mathbb{R}$. Three activation functions, the sigmoidal, the hyperbolic tangent and the linear function,

are widely used in neural networks. In this work, the hyperbolic tangent function f was utilised to convert the output value into the range [0,1] in the hidden layer while the linear function was applied in the output layer. The configured feed-forward neural network was subsequently trained through the back-propagation algorithm using the reduced low-dimensional data sets ($\boldsymbol{D}$, $\boldsymbol{U}$) from the PCA step. Similar to the PCA technique, the performance of surrogate ANN models is significantly affected by quantity and quality of collected samples.

The above ANN training can be viewed as an application of supervised machine learning methodology [45], which aims to learn a function that maps the input variables $\boldsymbol{d}$ to the output variables $\boldsymbol{u'}$ based on the collected input-output samples ($\boldsymbol{D}$, $\boldsymbol{U}$). Here the structure of the learned function is the chosen ANN model with the corresponding learning algorithm being the training method, the Levenberg-Marquardt algorithm with an early stopping procedure. Performing this machine learning task will generate a predictive ANN model, which together with the previous PCA projection will formulate a low-dimensional surrogate model for the original high-dimensional system.

2.4 | PCA-ANN global optimisation framework

To cope with the non-convexity of highly non-linear systems, deterministic optimisation methods are considered for the reduced surrogate model from the PCA-ANN reduction. The black-box or grey-box global optimisation problem can be transformed into the general explicit NLP optimisation problem as follows combining Eqs.(4,5,9):

$$
\begin{aligned}
\min_{\boldsymbol{d}=[d_1,d_2,\ldots,d_{N_d}]} \quad & G(\boldsymbol{d},\boldsymbol{u'}) \\
s.t. h_j &= f\Big(\sum_{i=1}^{N_d} w^1_{j,i} d_i + b^1_j\Big), \quad \forall j \in \{1,2,\ldots,n\} \\
u'_l &= \sum_{j=1}^{n} w^o_{l,j} h_j + b^o_l, \qquad \forall l \in \{1,2,\ldots,k\} \\
\boldsymbol{u'} &= (u'_1, u'_2, \ldots, u'_k), \\
& g_{cons}(\boldsymbol{d}, \boldsymbol{P^T u'}) \le 0
\end{aligned}
\tag{10}
$$

In this work, $g_{cons}(\boldsymbol{d}, \boldsymbol{P^T u'})$ includes the box bound constraints and possible inequality constraints for design variables $\boldsymbol{d}$ and discretised state variables $\boldsymbol{y'}$. The ANN-based nonlinear objective function G, can be reformulated into the constraints. The main non-convexity of the optimisation problems lies on the surrogate model constraints $h_j = f(\cdot)$ due to the highly non-convex activation function, here the hyperbolic tangent function $tanh(\cdot)$ in the feed-forward ANN structure. General-purpose global optimisation commercial software, including ANTIGONE [46], BARON [26] and SCIP [47], are efficient tools for the above problems due to the advanced bound tightening and branching techniques. Nevertheless, these general global solvers can not handle the $tanh(\cdot)$ formulation directly, as high performance algorithms need the explicit model equations. Therefore the explicit algebraic form $tanh(z) = (e^z - 1)/(e^z + 1)$ is required [48]. The basic formulation is further transformed into $tanh(z) = -2/(e^z + 1) + 1$ in order to produce a tighter under-estimator for the global solver [27]. The flow chart of the basic PCA-ANN global optimisation framework is shown in Fig.2.

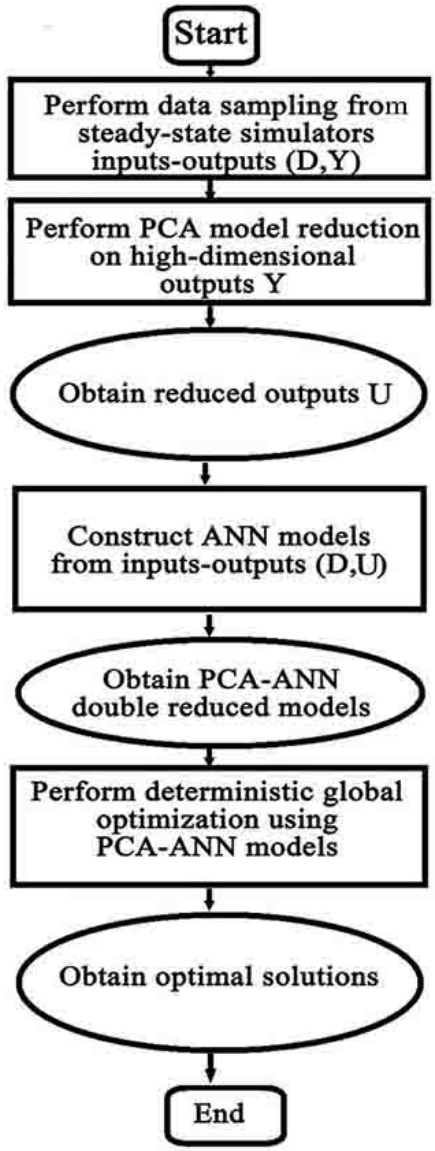


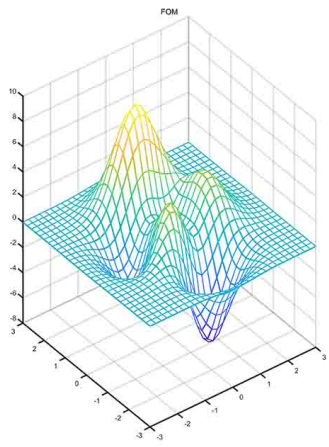
FIGURE 2 Flow chart of the basic PCA-ANN global optimisation framework

Nevertheless, constrained global optimisation is in general expensive even when reduced models are used.

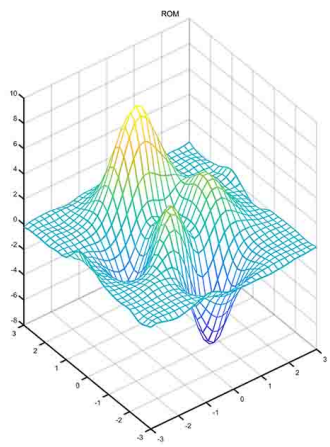
Hence, here we first test our proposed algorithm through a small illustrative example, a two-dimensional multi-modal peaks function, with the following mathematical formulation:

$$g_{peaks} = 3(1-x)^2 exp(-x^2-(y+1)^2) - 10(\frac{x}{5} - x^3 - y^5)exp(-x^2-y^2) - \frac{1}{3}exp(-(x+1)^2 - y^2) \qquad x, y \in [-3,3] \tag{11}$$

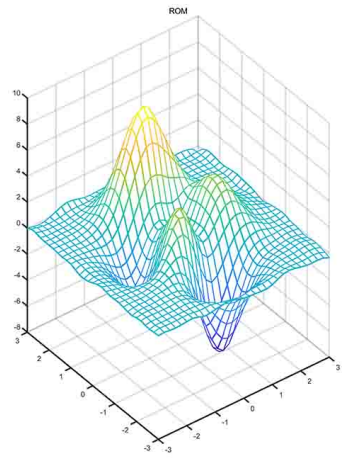
The peaks function is simulated using using 961 grid points (31 uniform grid points in each direction). We subsequently treated the model outputs as a black-box system and we employed the LHC sampling method to collect snapshots. Four groups of LHC samples were collected, 300, 500, 1000, and 1600 respectively, to construct the ANN reduced order models (ROMs). We then compared the 4 different ROMs with the original full order model (FOM). It should be noted that for this relatively problem we did not apply PCA first, and we used the LHC samples to directly train our ANN network.



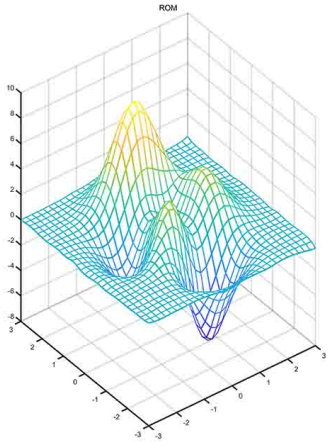
(a) FOM peaks function. Global optimum being -6.551 at (0.228, -1.626)



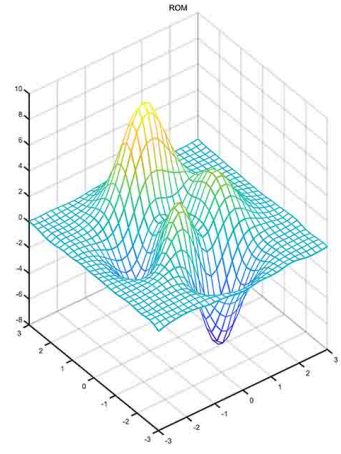
(b) ROM generated with 300 LHC samples and 50-neuron shallow ANN. Global optimum being -6.461 at (0.237, -1.625)



(c) ROM generated with 500 LHC samples and 55-neuron shallow ANN. Global optimum being -6.536 at (0.222, -1.632)



(d) ROM generated with 1000 LHC samples and 45-neuron shallow ANN. Global optimum being -6.540 at (0.220, -1.630)



(e) 1600 LHC samples, 52-neuron shallow ANN. Global optimum being -6.555 at (0.228, -1.625)

FIGURE 3 Comparison of the original peaks function (FOM) and different surrogate models (ROMs)

Fig.3 shows that all ANN ROMs constructed via the four sample groups could represent well the general nonlinear behaviour as well as the position of the global optimum. However, the prediction accuracy of the four ANN models differs. The ANN models constructed through relatively more samples can provide more accurate surrogates for the original peaks function across the whole design space. This is really important for constrained optimisation problems where nonlinear equality constraints need to accurately represent the system at hand. Thus more samples should be utilised to reduce the model errors across the whole design space for complex systems, especially for problems with highly nonlinear equality and inequality constraints.

The nonlinear optimisation problem for the peaks function, including the model equations as equality constraints is given by eq. 12

$$\begin{aligned} \min_{x_1,x_2} \quad & f \\ s.t. \quad & f = g_{peaks}(x_1, x_2) \\ & x_1, x_2 \in [-3, 3] \end{aligned} \tag{12}$$

The computed global optimum is located at point (0.228, -1.626) has value of -6.551. Then the a 52-neuron shallow ANN trained through 1600 collected samples as shown in Fig.3(e) was employed as a ROM to replace the peaks function. While 1600 samples may be excessive they can adequately be used to train the ANN to accurately capture the nonlinear features of the peaks function. Here we focus on developing a ROM-based optimisation algorithm assuming enough informative samples. For practical large-scale problems, sampling reduction approaches could be combined to improve computational efficiency. We utilised BARON 17.4.1 as the global solver with both relative and absolute tolerances being 0.002 and a limit of 36000 seconds (10 hours). A near global solution was computed after 30294.26 CPU seconds, at the point (0.228,-1.625) with the objective function being equal to -6.555, which is really close to the known global solution of the original problem.

Furthermore, we tested the effect of nonlinear inequality constraints on our surrogate model based global optimisation framework. In general, the feasibility and global optima issues caused by inequality constraints (g_{cons} in eq. 4), would be hard for black-box optimisation algorithms to cope with. Nevertheless, surrogate model based global optimisation approaches can efficiently deal with such problems.

If simple but accurate surrogate models are built, highly accurate global optima approximations can be obtained. Here the peaks function comprising the black-box equality constraints, together with inequality constraints formulate the following optimisation problem:

$$\begin{aligned} \min_{x_1,x_2} \quad & f \\ & s.t. f = g_{peaks}(x_1, x_2) \\ & x_1 f \geq 1.1 \\ & x_1 x_2 \geq 0 \\ & x_1, x_2 \in [-3, 3] \end{aligned} \tag{13}$$

The global optima was computed at the optimal point (-0.226,-1.710) with a value of -4.867 through BARON 17.4.1 with both relative and absolute tolerances being 0.002. Then the "black-box" 1600 LHC samples -based 52-neuron shallow ANN was utilised to represent the peaks function. Only 9.23 CPU seconds were required to produce a near global optimal (-0.226,-1.709) with an objective function value of -4.862, which is extremely close to the

true global solution of the original problem. The possible reason for this significantly faster computation is that the inequality constraints perform excellent cuts of the design space, hence enhancing the branch and reduce procedures within BARON. Next, we tested the same problem Eq.(13) using the sample-based search solver COBYLA [49] with the convergence tolerance being 0.002 and maximum function evaluations of 2000. Since computational results highly depend on initial points due to global optima issues, 50 runs were implemented with different LHC samples as initialisation points. Only 20 feasible solutions were found for the 50 LHC samples. The 50 runs took 0.081 seconds to compute the best solution at the point (-1.387,0) with the objective function value being -2.861, which was far from the true global solutions. The computational results imply that sample-based optimisation procedure may suffer serious feasibility issues in the computation of global optima. The feasibility issues become more prominent if the inequality constraints involve high-dimensional outputs. Moreover, multiple initialisations requires higher sample times to avoid local optima but cannot guarantee the accuracy of global optima solutions.

Although the above two illustrative examples show that the ANNs ROM-based global optimisation framework with/without inequality constraints can produce near-global optima solutions, the first one that does not include non-linear inequality constraints still requires intensive computations due to the complex ANN structures. This drawback drives our further improvements for our PCA-ANN global optimisation framework, through the reduction of the complexity of the constructed ANN models, leading to significantly faster computations for the general ANN constrained problems. It should be noted that, as mentioned above, the PCA step is not necessary for the small peaks function example Eq.(13) with the low-dimensional input-output variables, but is utilised for the two large-scale case studies, the tubular reactor and the combustion process, which are also used to illustrate the two improvements of our PCA-ANN global optimization framework.

3 | PIECE-WISE AFFINE BASED FORMULATION

In this section, a piecewise affine (PWA) reformulation is introduced to deal with the non-convex hyperbolic tangent activation function in the reduced surrogate ANN-based model. Previous research has suggested the PWA technique for ANN models [34], which has been verified to be efficient [50]. Although these studies provided some computational results, further detailed implementation schemes and analysis have not been reported. In this work, the PWA reformulation was utilised to approximate the highly non-convex NLP problem with a MILP problem. The global optimisation algorithms for both NLP and MILP problems are based on the branch and bound framework. However, the branching step is performed on continuous variables for the NLP problems and on auxiliary binary variables for the MILP problems through the use of CPLEX 12.7.1. An adaptive procedure to construct PWA models is presented below.

3.1 | Adaptive procedure

The hyperbolic tangent activation function $f(z) = tanh(z)$ is an odd function with central symmetry, which is concave on $(0, +\infty]$ and convex on $[-\infty, 0)$. Therefore the PWA approximation on $[-\infty, 0)$ can be directly computed from the PWA formulation on $(0, +\infty]$. Within the range of $(0, +\infty]$, $tanh(z)$ function first increases and then tends to level off with a slight increase towards the limit value of 1. The adaptive PWA procedure starts from the interval $(0, +\infty]$ and two points, the point of symmetry and one point close to the maximum value (equal to 1). Then a new point is chosen between the two original points so that the error E_{ero} between $f(z)$ and its PWA approximation $f_{PWA}(z)$ (currently

consisting of two intervals) is minimised.

$$E_{ero} = \int abs(f(z) - f_{PWA}(z))\mathrm{d}z, \tag{14}$$

Then the segment with the largest error is chosen and a new point is added within to minimise E_{ero} in this segment. This procedure continues iteratively until the error in eq. 14 becomes less than a pre-defined tolerance. Finally the points chosen for the $(0, +\infty]$ interval are mirrored to the $[-\infty, 0)$ interval.

The iteration procedure efficiently produces a tight PWA representation of the $tanh(z)$ function. Fig.4 shows the adaptive process, narrowing the interval sizes and reducing the error (red shade) between piecewise affine models and $tanh(z)$.

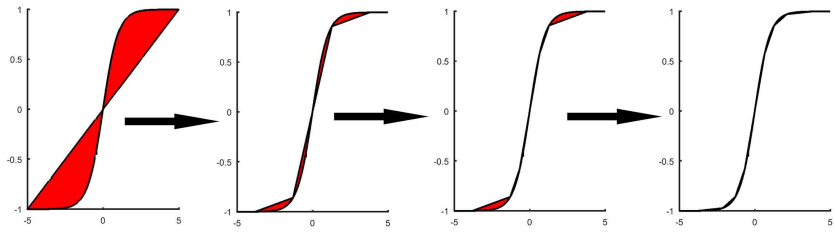


FIGURE 4 Adaptive PWA procedure for the hyperbolic activation function

There are different approaches to formulate the PWA models, such as the classic method, the linear segmentation method,the convex hull method and special structure methods. The classic method is the basic step of the other three methods and may work less efficiently [51]. Here we employed the special order sets technique since the advanced MILP solvers including CPLEX could smartly exploit the structures of special order sets and speed up computations [52]. For $N' + 1$ generated grid points $z_1, z_2, ..., z_{N'+1} \in \mathbb{R}$ and correspondingly N' linear models, the general PWA formulation introducing the special sets variables h'_i and λ'_i, is as follows [53]:

$$\begin{aligned}
f(z) \approx f_{PWA}(z) &= \sum_{i=1}^{N'+1} \lambda'_i f(z_i), \\
z &= \sum_{i=1}^{N'+1} \lambda'_i z_i, \\
\sum_{i=1}^{N'+1} \lambda'_i &= 1, \\
\lambda'_1 &\leq h'_1, \\
\lambda'_i &\leq h'_i + h'_{i-1}, \forall i \in \{2, 3, ..., N'\} \\
\lambda'_{N'+1} &\leq h'_{N'}, \\
\lambda'_i &\geq 0, \forall i \in \{1, 2, ..., N' + 1\} \\
\sum_{i=1}^{N'} h'_i &= 1, \\
h'_i &\in \{0, 1\}^{N'}
\end{aligned} \tag{15}$$

It should be noted here that the above formulation allows only two adjacent λ_i's to be non-zero.

Substituting the highly non-convex $f(z)$ in the PCA-ANN optimisation formulation (Eq.10)with the above PWA reformulation (Eq.15), the general PCA-ANN-PWA based MILP optimisation problem can be obtained:

$$
\begin{aligned}
\min_{\boldsymbol{d}} \quad & G(\boldsymbol{d}, \boldsymbol{u'}) \\
s.t. \boldsymbol{z}^j &= \sum_{i=1}^{N_d} w_{j,i}^1 d_i + b_j^1, && \forall j \in \{1,2,...,n\} \\
h_j &= \sum_{i=1}^{N'+1} \lambda_i^j f(z_i), && \forall j \in \{1,2,...,n\} \\
z^j &= \sum_{i=1}^{N'+1} \lambda_i^j z_i, && \forall j \in \{1,2,...,n\} \\
\sum_{i=1}^{N'+1} \lambda_i^j &= 1, && \forall j \in \{1,2,...,n\} \\
\lambda_1^j &\le h_1^j, && \forall j \in \{1,2,...,n\} \\
\lambda_i^j &\le h_i^j + h_{i-1}^j, && \forall i \in \{2,3,...,N'\}, \forall j \in \{1,2,...,n\} \\
\lambda_{N'+1}^j &\le h_{N'}^j, && \forall j \in \{1,2,...,n\} \\
\lambda_i^j &\ge 0, && \forall i \in \{1,2,...,N'+1\}, \forall j \in \{1,2,...,n\} \\
\sum_{i=1}^{N'} h_i^j &= 1 && \forall j \in \{1,2,...,n\} \\
h_i^j &\in \{0,1\}^{N'} && \forall j \in \{1,2,...,n\} \\
u_l' &= \sum_{j=1}^{n} w_{l,j}^o h_j + b_l^o, && \forall l \in \{1,2,...,k\} \\
\boldsymbol{u'} &= (u_1', u_2', ..., u_k'), \\
g_{cons}'(\boldsymbol{d}, \boldsymbol{P^T u'}) &\le 0
\end{aligned}
\tag{16}
$$

where g_{cons}' is a PWA formulation of the possibly nonlinear inequality constraints. The following case studies consider optimisation problems with linear inequality constraints, which is enough to validate the efficiency of PWA formulation of nonlinear ANN models on computational cost and accuracy.

3.2 | Illustrative example

To verify the efficiency of the above PWA formulation, global optimisation is performed for the surrogate ANN model and the ANN-PWA model of the peaks function (Eq.(11)). Here, the ANN model is the reduced model (No PCA reduction was necessary). Two different PWA models (with 30 and 58 linear segments, respectively) were constructed following the above adaptive procedure. Tab.1 shows the comparison of the optimal solutions computed using the three reduced models with adaptive and uniform linear segments. Almost the same optimal solutions are computed, which are close to the global optimum value of the FOM (-6.551). The proposed ANN-PWA model with 30 linear segments required significantly less computational time compared to the other two. In fact we can observe a 4-fold reduction compared to the ANN-PWA model with 58 linear segments and a major 30-fold reduction compared to the ANN formulation. Furthermore, the uniform PWA formulation produced worse solutions than the proposed adaptive scheme. Meanwhile, solving the same-size MILP problem resulting from the uniform partitioning scheme is more computationally-consuming, implying stronger relaxation tightness than that of the proposed adaptive PWA scheme.

TABLE 1 Comparative results of ANN model and ANN-PWA models

Model	Solver	Optimal value	CPU time (s)	Rel.tolerance
ANN(1 layer, 52 neurons, tanh)	BARON	-6.555	30294.26	0.002
ANN-PWA(30 adaptive linear segments)	CPLEX	-6.542	1004.71	0.002
ANN-PWA(30 uniform linear segments)	CPLEX	-6.406	418.66	0.002
ANN-PWA(58 adaptive linear segments)	CPLEX	-6.540	4190.16	0.002
ANN-PWA(58 uniform linear segments)	CPLEX	-6.489	2840.36	0.002

3.3 | Case study

Here a typical tubular reactor where an exothermic reaction takes place [54], was used as an illustrative example to show how quantity and quality of samples affect PCA projection errors. The reactor model consists of 2 differential equations in dimensionless form:

$$
\begin{aligned}
0 &= \frac{1}{Pe_1}\frac{\partial^2 C}{\partial y^2} - \frac{\partial C}{\partial y} + Da(1-C)exp(T/(1+T/\gamma)) \\
0 &= \frac{1}{LePe_2}\frac{\partial^2 T}{\partial y^2} - \frac{1}{Le}\frac{\partial T}{\partial y} - \frac{\beta}{Le}T + BDa(1-C)exp(T/(1+T/\gamma)) + \frac{\beta}{Le}T_w \\
b.c. & \\
&\frac{\partial C}{\partial y} - Pe_1 C = 0, \frac{\partial T}{\partial y} - Pe_2 T = 0, \quad at \quad y = 0 \\
&\frac{\partial C}{\partial y} = 0, \frac{\partial T}{\partial y} = 0, \quad at \quad y = 1
\end{aligned}
\tag{17}
$$

C and T are the dimensionless concentration and temperature respectively, while C_{exit} is dimensionless output concentration. Da is the Damköhler number, $Le = 1$ the Lewis number, $Pe_1 = 5$ the Peclet number for mass transport and $Pe_2 = 5$ for heat transport, $\beta = 1.5$ a dimensionless heat transfer coefficient, C the dimensionless adiabatic temperature rise, $\gamma = 10$ the dimensionless activation energy, $T_w = 0$ the adiabatic wall temperature and y the dimensionless longitudinal coordinate. The model equations were discretized using central finite differences over 250 computational nodes, resulting in 500 algebraic equations, which comprise our in-house simulator, which was subsequently treated as a black-box (input/output) code. The initial aim was then to represent the nonlinear behaviour of 500 outputs (distributed dimensionless temperature and concentration) with respect to the single design variable Da, varying in the range [0.121, 0.400]. For comparison purposes, we generated 6 different sample groups of 10, 20, 30, 40, 50, 60 LHC samples, respectively. Then PCA projection was employed to compute the dominant PCs, with the maximum energy/variance ratio set to 99.8 %. Finally, we utilised 500 uniform design points in the above range (0.121, 0.400), resulting in a total of 250000 outputs, to test the PCA model prediction accuracy outside the sample set utilised to generate the corresponding PCs. The results are shown in Tb.2.

TABLE 2 Comparative results of PCA for different numbers of samples

Number of Samples	Number of PCs	Total error (250000 points)	Maximum error
10	2	5220	3.16
20	2	5350	3.23
30	2	3680	0.86
40	2	4060	3.03
50	3	1170	0.16
60	3	1210	0.50

Tb.2 shows that the smaller sample groups (10, 20, 30, 40, LHC samples respectively) require only 2 PCs to capture the variance of the corresponding data sets while the larger sample groups (50, 60 LHC samples, respectively) need 3 PCs, indicating that fewer samples would miss some global information (PCs) of the whole design space. This is also the possible reason that the resulting PCA projections from the smaller sample groups generate higher errors in the validation process. One exception to this is the 40-sample case, which is less accurate than 30-sample one despite the higher number of samples. The possible explanation is that the generated 40 samples happen to be less representative than the generated 30 LHC samples. A similar situation occurs with the 50 and 60 sample groups. Both the total and the maximum errors of 50-sample group are small enough, implying these 50 samples include sufficient information to accurately represent the whole design space through the three computed PCs.

Importantly, the PCA step reduces the original 500 distributed state variables into only 3 PC variables, which significantly decreases the computations for training ANN models and for subsequently performing deterministic global optimisation of the trained models. We employed the 50 LHC samples group for training our ANNs. To avoid over- and under-fitting, the defined domain was randomly divided into a training, a validation and a test set with respective size ratios of 0.7 : 0.15 : 0.15. The machine-learning algorithm was implemented through the MATLAB Neural Network Toolbox to fit the corresponding weights and biases by minimizing the mean squared error (MSE) between the ANN model and the training set using Levenberg-Marquardt algorithm and the early stopping procedure. To obtain a suitable number of neurons in the hidden layer, the training process was repeated using an increasing number of neurons until the MSE for all three sets became less than a pre-defined tolerance, here $1x10^{-4}$. We tested the training process for a 5-neuron shallow ANN generated for the PCA-reduced model with a single input and three outputs, which took less than 1s for the 849 iterations required for convergence. Without the PCA reduction step, each iteration requires 12.41s for the single input and 500 output ANN with the same hidden layer structure, to slowly reduce the training errors. Nevertheless, convergence could not be achieved in 20 consecutive training runs for this ANN due to large validation and/or test errors, implying that a larger ANN with more layers is needed to accommodate the relevant input/output information. This would in turn lead to a much higher number of computations for deterministic global optimisation of the trained ANN. All runs were performed on a Desktop (Intel® Core(TM) CPU 3.3 GHz, 8 GB memory, 564-bit operating system) running Windows 7.

We subsequntly investigated the computational efficiency of the PCA-ANN-PWA optimisation framework through the maximisation of the exit concentration of the tubular reactor system by manipulating the temperature of 3 cooling

zones along its length. The mathematical formulation of the optimisation problem is as follows :

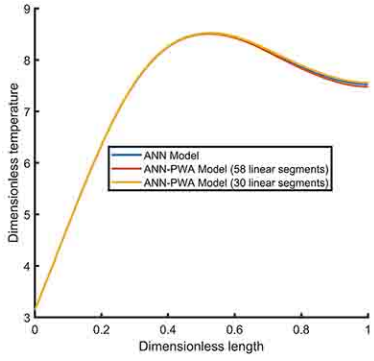
$$\begin{aligned} &\max_{T_{wi}} C_{exit} \\ s.t. \quad &PCA-ANN \quad model \\ &T_w(y) = \sum_{i=1}^{3} \left(H(y-y_{i-1}) - H(y) - y_i\right) T_{wi} \end{aligned} \tag{18}$$

Here C_{exit} is the dimensionless output concentration. The system parameters are Pe_1 = 5, Pe_2 = 5, Le = 1, β = 1.5, γ = 10, B = 12, D_a=0.1. T_w is the adiabatic wall temperature, T_{wi} are the corresponding temperatures at the three cooling zones, i and H is the Heaviside step function.

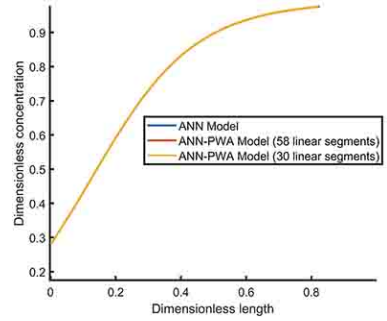
As before, the rector model, Eq.(17), discretized in 250 central finite differences comprises our in-house FOM simulator. PCA reduction was performed first to reduce the 500 state variables down to 12. Subsequently 20- to 40-neuron ANNs were employed to obtain a reduced PCA-ANN model comprising 3 inputs, and 12 state variables (outputs). The optimisation results, depicted in Tab.3 and Fig.5, are computed to compare the optimisation performance using the PCA-ANN model and two PCA-ANN-PWA models with 30 and 58 linear segments, respectively. All three computational cases converge to almost the same solutions, with objective function value close to 0.99998, which is the value computed by performing optimisation with the FOM. The maximum error is 1.52%, and the optimal solution profiles for concentration and temperature distributions are very close to each other for all models (Fig.5). Fig.6 compares the computational time required to perform optimisation using the PCA- ANN and the PCA-ANN-PWA models with different number of ANN neurons. The limit time (max time for computations to stop) was set to 36000 seconds. The computational time increases rapidly with more neurons for all three surrogate models. The computational cost reaches the limit time for the PCA-ANN model with 40 neurons while the CPU time required for the two PCA-ANN-PWA models is less than 1000 seconds. It can be also seen that the computational time required is significantly less for both PCA-ANN-PWA models, irrespective of the number of ANN neurons, implying the high computational efficiency of the proposed PWA methodology.

TABLE 3 Optimal result comparisons for surrogate tubular reactor models

Model	Solver	Optimal value	True value (full model)	Error	CPU (s)
PCA-ANN(1 layer, 20 neurons, tanh)	BARON	0.98682	0.99998	1.32%	455.09
PCA-ANN-PWA(30 linear segments)	CPLEX	0.98483	0.99998	1.52%	6.12
PCA-ANN-PWA(58 linear segments)	CPLEX	0.98623	0.99998	1.38%	14.32



(a) Solution profiles for temperature



(b) Solution profiles for concentration

FIGURE 5 Solution profiles for dimensionless temperature and concentration

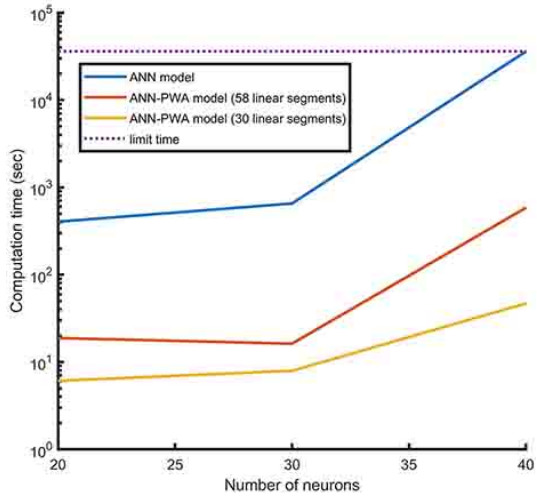


FIGURE 6 Computational time (seconds) for different numbers of neurons

4 | DEEP RECTIFIER NEURAL NETWORK BASED FORMULATION

The PCA-ANN-PWA global optimisation framework worked efficiently for the peaks function and the tubular reactor cases. However, the PWA step will in general lead to additional approximation error especially for large-scale problems. Past efforts in computer science have developed efficient activation functions, such as the sigmoid and the $tanh(\cdot)$ function. The S-shaped sigmoid function can transfer any input signal into the range [0,1] while the zero centered $tanh(\cdot)$ function can map the output values in the interval [-1,1]. Both of them can learn features of high nonlinear functions efficiently. Nevertheless, the high non-convexity of these functions makes ANN training hard. The continuous piecewise linear functions, including the $ReLU$ function and its variants, have been adopted to deal with this problem. In this work, the widely applied $ReLU$ function is utilised in order to reserve computational accuracy and

maintain the ability to use the advanced MILP solver. Nevertheless, shallow neural networks require a significantly larger number of nodes in one hidden layer to successfully represent a complex function, while deep neural networks result in more complex and non-convex training errors [55] due to their multi-layer structure. Low-complexity two- or three-hidden layer NNs are, however, enough to capture the low-dimensional nonlinear behaviour of PCA-reduced systems. Although deep rectifier NN-based MILP problems have been formulated in previous studies [56], the combination of PCA and deep rectifier NN has not been reported before to the best of our knowledge. The mathematical equations for the deep rectifier NNs are similar to those in fig. 1 with more hidden layers and activation function $f(z) = max(0, z)$, which can be reformulated into a piecewise linear function through the big-M method [57]:

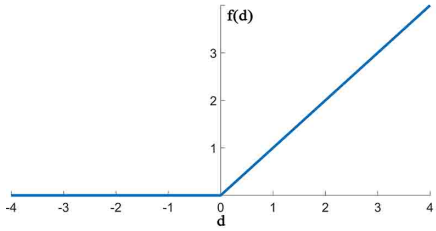
$$
\begin{aligned}
z_1^{j_1} &= \sum_{i=1}^{N_d} w_{j_1,i}^1 d_i + b_{j_1}^1, && \forall j_1 \in \{1,2,...,n_1\} \\
z_1^{j_1} &= z'^{j_1}_1 - z''^{j_1}_1, && \forall j_1 \in \{1,2,...,n_1\} \\
z'^{j_1}_1 &\le M_1(1 - bz_1^{j_1}), && \forall j_1 \in \{1,2,...,n_1\} \\
z''^{j_1}_1 &\le M_1 bz_1^{j_1}, && \forall j_1 \in \{1,2,...,n_1\} \\
z_2^{j_2} &= \sum_{j_1=1}^{n_1} w_{j_2,j_1}^2 z'^{j_1}_1 + b_{j_2}^2, && \forall j_2 \in \{1,2,...,n_2\} \\
z_2^{j_2} &= z'^{j_2}_2 - z''^{j_2}_2, && \forall j_2 \in \{1,2,...,n_2\} \\
z'^{j_2}_2 &\le M_2(1 - bz_2^{j_2}), && \forall j_2 \in \{1,2,...,n_2\} \\
z''^{j_2}_2 &\le M_2 bz_2^{j_2}, && \forall j_2 \in \{1,2,...,n_2\} \\
&\vdots && \vdots \\
z_\theta^{j_\theta} &= \sum_{j_{\theta-1}=1}^{n_{\theta-1}} w_{j_\theta,j_{\theta-1}}^\theta h_{\theta-1}^{j_{\theta-1}} + b_{j_\theta}^\theta, && \forall j_\theta \in \{1,2,...,n_\theta\} \\
z_\theta^{j_\theta} &= z'^{j_\theta}_\theta - z''^{j_\theta}_\theta, && \forall j_\theta \in \{1,2,...,n_\theta\} \\
z'^{j_\theta}_\theta &\le M_\theta(1 - bz_\theta^{j_\theta}), && \forall j_\theta \in \{1,2,...,n_\theta\} \\
z''^{j_\theta}_\theta &\le M_\theta bz_\theta^{j_\theta}, && \forall j_\theta \in \{1,2,...,n_\theta\} \\
z'^{j_i}_i &\ge 0, && \forall i \in \{1,2,...,\theta\}, \forall j_i \in \{1,2,...,n_i\} \\
z''^{j_i}_i &\ge 0 && \forall i \in \{1,2,...,\theta\}, \forall j_i \in \{1,2,...,n_i\} \\
bz_i^{j_i} &\in \{0,1\} && \forall i \in \{1,2,...,\theta\}, \forall j_i \in \{1,2,...,n_i\} \\
u'_l &= \sum_{j_\theta=1}^{n_\theta} w_{l,j_\theta}^o z'^{j_\theta}_\theta + b_l^o, && \forall l \in \{1,2,...,k\}
\end{aligned}
\tag{19}
$$

where M_i is the big-M constant, here equal to 10000, which was enough to capture the global optimum of the small peaks function example without numerical issues. $z'^{j_i}_i$ and $z''^{j_i}_i$ are the auxiliary non-negative variables, $bz_i^{j_i}$ is the auxiliary binary variable and $h_i^{j_i}$ is the output value from the j_ith ReLU based neuron of the ith hidden layer. θ is the number of hidden layers and n_i is number of neurons at the ith hidden layer. Substituting the ANN model equations in the PCA-ANN optimisation formulation (Eq.10) with the above reformulation (Eq.20), the following PCA-DNN(ReLU)

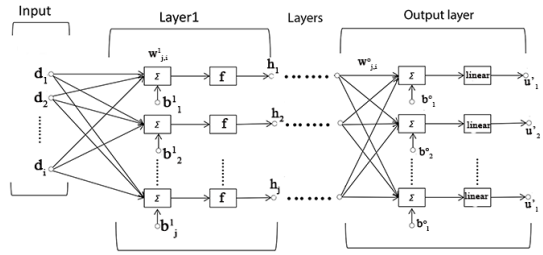
based MILP optimisation formulation can be obtained:

$$\begin{aligned} \min_{\boldsymbol{d}=[d_1,d_2,\ldots,d_{N_d}]} \quad & G(\boldsymbol{d},\boldsymbol{u'}) \\ s.t. \quad & Eq.(19); \\ & \boldsymbol{u'} = (u'_1, u'_2, \ldots, u'_k), \\ & \boldsymbol{g}'_{cons}(\boldsymbol{d}, \boldsymbol{P}^{\boldsymbol{T}}\boldsymbol{u'}) \leq 0 \end{aligned} \tag{20}$$

where $Eq.(19)$ denotes the equality constraints constructed using deep neural networks through the big-M method. This way, an improved framework is formulated using a deep neural network (DNN) with rectified linear units (ReLU) as illustrated in Fig.7, which is first tested with the peaks function, and then with a more challenging large-scale combustion process.



(a) ReLU activation function



(b) Deep artificial neural network

FIGURE 7 Deep neural network with rectified linear units

4.1 | Illustrative example

To verify the increased computational efficiency of the reLu-DNN models, global optimisation is first performed using four surrogate models (ANN, ANN-PWA, tanh-DNN and relu-DNN model, respectively) for the peaks function presented above. Tab.4 shows the optimisation results. The small-scale tanh-DNN could replace the larger shallow ANN model, resulting in significant computational savings, of more than one order of magnitude. The relu-DNN model requires more neurons than the tanh-DNN model, due to the simpler structure of the $relu$ activation function. Despite the fact that the relu-DNN model is larger, its optimisation cost is much lower, two orders of magnitude less than that of the tanh-NN models. The rapid global optimisation computations when using the relu-NN model are attributed to the advanced MILP solver algorithm utilised. Furthermore, the computation cost using the relu-DNN model is much less than that that of using the ANN-PWA model with 30 linear segments because of the large(r) number of linear models involved in the PWA formulation. More linear models lead to more binary variables, requiring more branching steps hence reducing the computational efficiency.

TABLE 4 Comparative optimisation results of different ANN models

Model	Solver	Optimal value	CPU time (s)	Rel.tolerance
ANN(1 layer, 52 neurons, $tanh$)	BARON	-6.555	30294.26	0.002
ANN-PWA (30 linear segments)	CPLEX	-6.542	1004.71	0.002
DNN(2 layers, 8-8 neurons, $tanh$)	BARON	-6.558	2579.68	0.002
DNN(2 layers, 40-40 neurons, $relu$)	CPLEX	-6.543	25.93	0.002

4.2 | Case study

To further test the significant advantages of the deep rectifier neural network in our global optimisation formulation, a more challenging combustion process [58, 23] is considered here. The combustion process takes place in a horizontal cylindrical combustor, 1.8m in length and 0.45m in diameter with a fuel nozzle that has a diameter 0.0045m. .The chemical reactions in the combustor are the following:

- $CH_4 + 2\,O_2 \rightarrow CO_2 + 2\,H_2O$
- $C_2H_4 + 3\,O_2 \rightarrow 2\,CO_2 + 2\,H_2O$
- $C_3H_8 + 5\,O_2 \rightarrow 3\,CO_2 + 4\,H_2O$
- $C_4H_{10} + 8.5\,O_2 \rightarrow 4\,CO_2 + 5\,H_2O$

In addition a complex NO mechanism, comprising thermal NO, prompt NO and N_2O intermediate mechanisms is also taken into account. The fuel NO mechanism was ignored due to the small amount of nitrogen in the feed. Thermal efficiency can be improved by increasing combustion temperature, which however, inevitably leads to more pollutant emissions, such as NOx. The NOx production is dominated by the thermal NO mechanism, given below, which is very sensitive to temperature.

- $O + N_2 \rightleftharpoons NO + N$
- $O_2 + N \rightleftharpoons NO + O$
- $N + OH \rightleftharpoons NO + H$

This work focuses on the optimisation of inlet operational conditions (shown in Tb. 5) in order to minimize NOx emissions. In addition to chemical reactions, multiple physical phenomena are involved, including complex turbulent flows, heat and mass transfer. Commercial CFD software was used, namely ANSYS/FLUENT, to construct high-fidelity CFD models to compute velocity, temperature and component fraction fields.

4.2.1 | CFD Model Description

The computational domain for the CFD model consisted of a 2-dimensional axisymmetric depiction of the combustor To ensure that computations are grid independent, numerical experiments using 5481, 6381, 9081 and 14832 computational cells were performed Finally, 9081 computational cells (9332 nodes) were chosen as solutions did not change with more computational cells/nodes. The renormalisation group (RNG) $k-\epsilon$ turbulence model for fluid flow was employed. The eddy-dissipation model was employed for the species transport equations because the overall reaction

rate was controlled by turbulent mixing. To take into account the effects of thermal radiation, including absorption and scatting coefficients, a discrete ordinates (DO) radiation model was used.

The second-order upwind scheme was applied for the space derivatives of the advection terms in all transport equations. The SIMPLE algorithm was employed to handle the velocity-pressure coupling in the flow field equations. Convergence criteria required the residual for the energy equation to be below 1×10^{-6} and residuals for the other model equations to be below 1×10^{-3}. The mass-weighted-averages of temperature at the exit and the maximum temperature of the entire fluid were also monitored as other convergence criteria. The base case inlet conditions used are given below. For the fuel gas, the base value for the inlet velocity was 100 m/s and that for the inlet temperature was 298K. The inlet composition was as follows: CH_4: 87.8%, C_2H_4: 4.6%, C_3H_8: 1.6%, C_4H_{10}: 0.5%, N_2: 5.5%. For preheated air, the inlet velocity was 85 m/s and the inlet temperature 1473 K while the inlet composition was: O_2: 19.5%, N_2: 59.1%, H_2O: 15%, CO_2: 6.4%, NO: 110 ppm. Five independent variables were used to optimise the whole process. The independent variables along with their allowable ranges are listed in Tab.5.

TABLE 5 Range of independent variables

Variables	Range	Units
inlet air velocity	[85,125]	m/s
inlet fuel velocity	[80,120]	m/s
oxygen mass fraction (inlet air)	[18.5, 19.5]	%
inlet air temperature	[1450, 1600]	K
inlet fuel temperature	[298, 398]	K

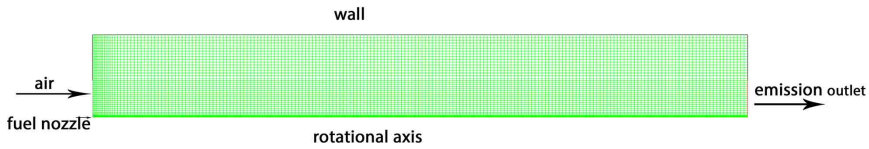


FIGURE 8 Two-dimensional geometry of a single axisymmetric combustor can and its mesh

4.2.2 | Model reduction

Although the high-fidelity CFD model can provide accurate simulation results, its black-box characteristics and overall complexity make further optimisation and control tasks computationally tedious. Reduced surrogate models need to be developed to deal with the challenges arising. The LHC sampling method was utilised to collect 1024 CFD samples, which took around 6 days using 4-CPU parallel computing. The input variables are the ones listed in Tab.5, while the output results are the physical field data along with the average NOx emission at the outlet surface. Due to the high dimensionality of the FOM, direct mapping of the input-output relationship would result in extremely large-scale ANN surrogate models, which could easily exceed the capability of current optimisation algorithms. Therefore, the PCA step was first employed and then surrogate ANN models were constructed based on the PCA-reduced models. ANN models were built for the field data, to construct the redcuced PCA-ANN constraints and for the average output NOx emission to formulate the ANN-reduced objective function. The field data include axial and radial velocity, temperature, N_2, H_2O, O_2, CO_2, C_4H_{10}, CH_4, C_2H_4, C_3H_8, and NO fraction concentrations (12 state variables). It should be noted that the average output NOx emission is only one variable so does not require PCA reduction. In this work, PCA was performed separately for each state variable. While some PCA methods compute principal components for all state variables together, we found that working on each state variable we could generate more accurate principal components. The standard criterion, of capturing 99.99% of the total energy, was set. This way,

the reduced surrogate models, were built, as displayed in Tab.6.

TABLE 6 Number of PCs and corresponding ANN models

Variables	Number of PCs	DNN (2 layers, tanh)	DNN (2 layers, relu)
		No of neurons	No of neurons
Axial velocity	4	14, 14	14, 14
Radial velocity	9	15, 15	22, 22
Temperature	6	16, 16	16, 16
N_2 concentration fraction	7	19, 19	24, 24
H_2O concentration fraction	8	15, 15	18, 18
O_2 concentration fraction	6	17, 17	20, 20
CO_2 concentration fraction	7	12, 12	14, 14
C_4H_{10} concentration fraction	6	15, 15	17, 17
CH_4 concentration fraction	7	26, 26	28, 28
C_2H_4concentration fraction	6	10, 10	18, 18
C_3H_8concentration fraction	6	18, 18	24, 24
NO concentration fraction	4	12, 12	14, 14
Objective: output NOx emission	-	ANN (1 layer, tanh)	ANN (1 layer, relu)
	-	14	30

4.2.3 | Model validation

Model validation was performed for the reduced models before the subsequent optimisation step, taking into account two aspects, representation ability and prediction ability. The representation ability of the reduced models was tested through the comparison between the FOM and the ROMs on the base case inlet conditions. Computational results show only very small differences, especially for N_2, C_4H_{10}, CH_4, C_2H_4, C_3H_8, and NO fraction fields. The above species fraction fields are close to uniform distribution across the combustor, except for the small area near the fuel nozzle. Fig.9(a), 9(b), 9(c), 9(d), 9(e) depict the velocity field, temperature field, O_2 , CO_2 and H_2O concentration fraction field of FOM, tanh-ROM and relu-ROM, respectively for the inlet base values. The five contour diagrams illustrate that flow, temperature and mass fraction fields of FOM and ROMs are very close, indicating the strong representation ability of the ROMs. Moreover, the tanh-DNN reduced models show smaller difference from the FOM than the relu-DNN reduced models, especially for the temperature field, implying the better accuracy of the tanh-DNN models due to the non-linearity of the tanh function. Tab.7 shows the comparison of maximum field values between FOM and ROMs and the corresponding errors. The largest error is only 0.56%. To test the ROMs prediction ability, 40 random inlet condition points different than the base case ones were chosen and compered with FOM results. The largest error was less than 5% indicating that the ROMs can be reliably used for further optimisation studies. Fuerhtermore, the ROMs exhibit significant computational savings compared to the full-order CFD models as expected. The average CPU time for the CFD model (run in ANSYS/FLUENT) is approximately 1560 CPU seconds, while each ROM requires less than 0.1 CPU seconds and can be efficiently used to perform global optimisation studies.

TABLE 7 Average value comparison of FOM and ROMs

Variables	FOM	relu-ROMs	errors	tanh-ROMs	errors
Velocity(m/s)	29.82089	29.6652	0.40%	29.70166	0.56%
Temperature(K)	1625.259	1621.948	0.03%	1625.702	0.20%
H_2O mass fraction	0.151743	0.1518191	0.02%	0.1517722	0.05%
O_2 mass fraction	0.1906906	0.1908453	0.01%	0.1907176	0.08%
CO_2 mass fraction	0.0663304	0.06627689	0.00%	0.06633245	0.08%
Computational time for each sample	1560	<0.1	-	<0.1	-

4.2.4 | Global optimisation

In this section, global optimisation is implemented using the validated reduced models. The general mathematical formulation is given in Eq.(20). In the combustion optimisation problem, $\boldsymbol{d}$ are the 5 inlet operation parameters, and $\boldsymbol{u'}$ are the 76 reduced state variables . The objective function $G(\boldsymbol{d}, \boldsymbol{u'})$ represents the average outlet NOx emission. The allowable ranges for the input variables are given in Tab.5, while the bounds for the state variables are given in Tab.8. It should be noted that the state variable bounds are implemented through the inverse projection

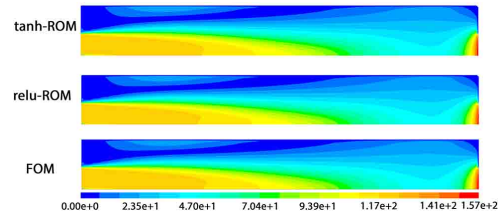
$$lb \leq \boldsymbol{P^T u'} \leq ub \tag{21}$$

where lb and ub denote lower and upper bounds, respectively.

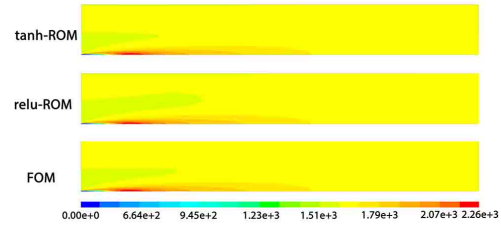
Finally, a MILP problem with 29,903 linear constraints, corresponding to the equality constraints and 488 binary variables corresponding to the total number of ANN neurons is formulated for the relu-based ROM, while a NLP problem with 28247 linear constraints, and 392 nonlinear terms is constructed for the tanh-based ROM. The limit value for the computational time was set to be 100 hours. Both of the relative and absolute tolerances were set to be 0.002.

The NLP problem did not converge to a feasible solution in BARON within the allowable time, probably due to the high non-convex activation function $tanh$ and large number of variables than inhibited the branch-and-bound algorithm.

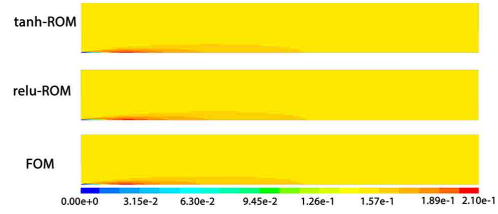
The $relu$-based MILP problem converged in 1001.94s in CPLEX. The computed optimal solution was: NOx emission: 110.17 ppm, air velocity: 95.07 m/s, fuel velocity:119.08m/s, oxygen fraction concentration (air): 18.50 %, air temperature: 1450 K and fuel temperature: 369.83K. To validate the computed optimal solutions, we performed a full CFD simulation in ANSYS/FLUENt using the calculated optimal inlet conditions. The outlet NOx emission was computed to be 113.26 ppm, which was very close to the calculated optimum with an error of approximately 2.73 %, which is small enough for most industrial cases. Fig.10(a), 10(b), 10(c), 10(d) ,10(e), depict a comparison of the main field state varibales at the optimal; conditions computed by the reduced and the full models, respectively. As it can be observed, the optimal solution computed through the ROM is very close to FOM simulation using the optimal inlet conditions. Tab.9 gives a comparison of the corresponding max values across the whole domain. The performance of the reduced model is very close to the full model with the biggest error being less than 3%. The computational cost for the ReLu-based MILP problem is significantly reduced compared to the NLP problem, which in this case could not converge, signifying the efficiency of the our model reduction-based global optimisation methodology.



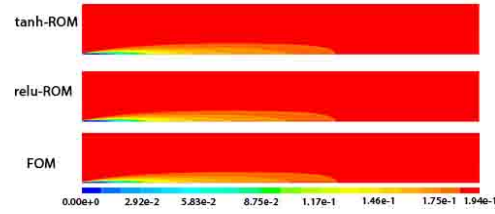
(a) Velocity (m/s) comparison among full model and reduced models



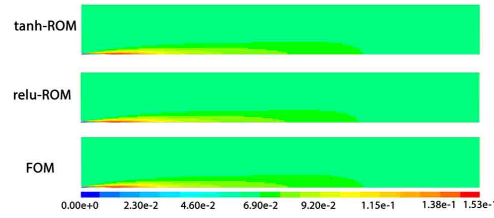
(b) Temperature (K) comparison among full model and reduced models



(c) H_2O mass comparison among full model and reduced models



(d) O_2 mass comparison among full model and reduced models

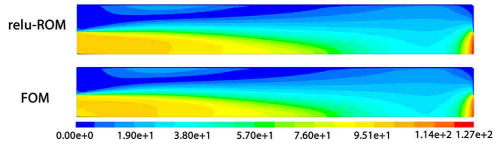


(e) CO_2 mass comparison among full model and reduced models

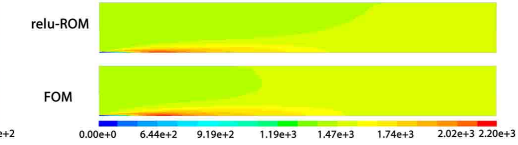
FIGURE 9 Comparison of velocity, temperature and concentration fraction field between FOM and ROMs

TABLE 8 Range of state variables

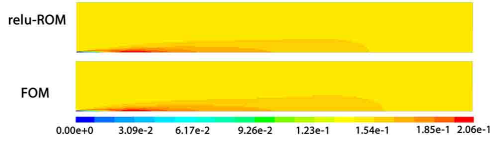
State variables	Range	Units
Axial velocity	[-150,150]	m/s
Radial velocity	[-150,150]	m/s
Temperature	[0, 2200]	K
N_2 concentration fraction	[0,1]	-
H_2O concentration fraction	[0,1]	-
O_2 concentration fraction	[0,1]	-
CO_2 concentration fraction	[0,1]	-
C_4H_{10} concentration fraction	[0,1]	-
CH_4 concentration fraction	[0,1]	-
C_2H_4 concentration fraction	[0,1]	-
C_3H_8 concentration fraction	[0,1]	-
NO concentration fraction	[0,1]	-



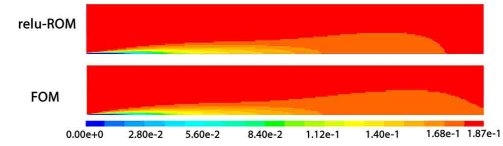
(a) Velocity (m/s) comparison among full model and reduced models under optimal condition



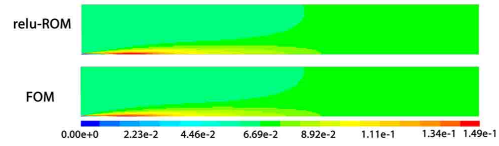
(b) Temperature (K) comparison among full model and reduced models under optimal condition



(c) H_2O mass fraction comparison among full model and reduced models under optimal condition



(d) O_2 mass fraction comparison among full model and reduced models under optimal condition



(e) CO_2 mass fraction comparison among full model and reduced models under optimal condition

FIGURE 10 Comparison of optimal velocity, temperature and fraction concentration field between FOM and ROM

TABLE 9 Average value comparison of FOM and ROMs

Variables	FOM	relu-ROMs	errors
Velocity(m/s)	23.11901	23.73662	2.67%
Temperature(K)	1487.854	1456.745	2.09%
H_2O mass fraction	0.1522371	0.1520701	0.11%
O_2 mass fraction	0.1803872	0.1806236	0.13%
CO_2 mass fraction	0.06699268	0.06689615	0.14%
Output NOx emission (ppm)	113.26	110.17	2.73%

5 | CONCLUSIONS

This paper presents model-reduction, machine-learning based global optimisation framework for large-scale nonlinear steady-state systems. A double model reduction, comprising principal component analysis and artificial neural networks, were first employed to construct accurate surrogate reduced models through a machine-learning step, which was then utilised by deterministic global optimisation methods. The high non-convexity of the activation function in reduced ANN models affects the computational speed branch-and-bound algorithms. To overcome this barrier, two improvements were proposed. Firstly, when nonlinear activation functions are preferable, a piece-wise affine reformulation to transform the nonlinear branching into binary variables resulting in a MILP problem with higher computational efficiency. Secondly, the implementation of a continuous piece-wise linear activation function-based deep ANN structure to improve computational accuracy. A number of case studies of different size including the peaks function, a tubular reactor and a complex large-scale combustion process were employed to illustrate the favorable performance of the presented framework. Despite the efficient implementation of the developed methodology, there are still remaining challenges to efficiently compute the global optimum for large-scale optimisation problems. Firstly, this work assumes enough representative samples as a basis to construct the reduced order models. Smart sampling methods to achieve optimal trade-off between quality and quantity are important for improving both efficiency and accuracy, as well as verification methods to guarantee the accuracy of the computed solutions[59]. Secondly, global optimisation even using reduced surrogate models is still computationally expensive. Advanced data techniques and MILP algorithms [60] may further improve computational efficiency of this optimisation framework.

acknowledgements

The financial support of the University of Manchester and China Scholarship Council joint scholarship (file no. 201706250031) for MT's PhD studies is gratefully acknowledged

conflict of interest

The authors declare there is no conflict of interest.

references

[1] Boukouvala F, Hasan MF, Floudas CA. Global optimization of general constrained grey-box models: new method and its application to constrained PDEs for pressure swing adsorption. Journal of Global Optimization 2017;67(1-2):3–42.

[2] Tao M, Guo K, Huang Z, Liu H, Liu C. A hybrid optimization method to design shapes of three-dimensional flow channels. Chemical Engineering Research and Design 2016;114:190–201.

[3] Park S, Li Y. Integration of biological kinetics and computational fluid dynamics to model the growth of Nannochloropsis salina in an open channel raceway. Biotechnology and bioengineering 2015;112(5):923–933.

[4] Yang GL, Zhou WH, Liu F. Simulation of flow field of high-pressure water-jet from nozzle with FLUENT [J]. Journal of Lanzhou University of Technology 2008;34(2):49–52.

[5] Kleber A. Simulation of air flow around an Opel Astra vehicle with FLUENT. Journal Article, International Technical Development Center Adam Opel AG 2001;.

[6] Multiphysics C. Introduction to COMSOL Multiphysics®. COMSOL Multiphysics, Burlington, MA, accessed Feb 1998;9:2018.

[7] Fluent A. Ansys fluent. Academic Research Release 2015;14.

[8] Jasak H, Jemcov A, Tukovic Z, et al. OpenFOAM: A C++ library for complex physics simulations. In: International workshop on coupled methods in numerical dynamics, vol. 1000 IUC Dubrovnik Croatia; 2007. p. 1–20.

[9] Schilders WH, Van der Vorst HA, Rommes J. Model order reduction: theory, research aspects and applications, vol. 13. Springer; 2008.

[10] Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. science 2006;313(5786):504–507.

[11] Theodoropoulou A, Adomaitis RA, Zafiriou E. Model reduction for optimization of rapid thermal chemical vapor deposition systems. IEEE Transactions on Semiconductor Manufacturing 1998;11(1):85–98.

[12] Xie W, Bonis I, Theodoropoulos C. Data-driven model reduction-based nonlinear MPC for large-scale distributed parameter systems. Journal of Process Control 2015;35:50–58.

[13] Malik MR, Isaac BJ, Coussement A, Smith PJ, Parente A. Principal component analysis coupled with nonlinear regression for chemistry reduction. Combustion and flame 2018;187:30–41.

[14] Bonis I, Theodoropoulos C. Model reduction-based optimization using large-scale steady-state simulators. Chemical engineering science 2012;69(1):69–80.

[15] Petsagkourakis P, Bonis I, Theodoropoulos C. Reduced Order Optimization of Large-Scale Nonlinear Systems with Nonlinear Inequality Constraints Using Steady State Simulators. Industrial & Engineering Chemistry Research 2018;57(30):9952–9963.

[16] Bonis I, Xie W, Theodoropoulos C. Multiple model predictive control of dissipative PDE systems. IEEE Transactions on Control Systems Technology 2013;22(3):1206–1214.

[17] Luna-Ortiz E, Theodoropoulos C. An input/output model reduction-based optimization scheme for large-scale systems. Multiscale Modeling & Simulation 2005;4(2):691–708.

[18] Theodoropoulos C. Optimisation and linear control of large scale nonlinear systems: a review and a suite of model reduction-based techniques. In: Coping with Complexity: Model Reduction and Data Analysis Springer; 2011.p. 37–61.

[19] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. science 1983;220(4598):671–680.

[20] Chambers LD. Practical handbook of genetic algorithms: complex coding systems, vol. 3. CRC press; 2019.

[21] Floudas CA, Akrotirianakis IG, Caratzoulas S, Meyer CA, Kallrath J. Global optimization in the 21st century: Advances and challenges. Computers & Chemical Engineering 2005;29(6):1185–1202.

[22] Pires JCM, Martins FG, Sousa S, Alvim-Ferraz M, Pereira M. Selection and validation of parameters in multiple linear and principal component regressions. Environmental Modelling & Software 2008;23(1):50–55.

[23] Lang Yd, Malacina A, Biegler LT, Munteanu S, Madsen JI, Zitney SE. Reduced order model based on principal component analysis for process simulation and optimization. Energy & Fuels 2009;23(3):1695–1706.

[24] Henao CA, Maravelias CT. Surrogate-based superstructure optimization framework. AIChE Journal 2011;57(5):1216–1232.

[25] Fahmi I, Cremaschi S. Process synthesis of biodiesel production plant using artificial neural networks as the surrogate models. Computers & Chemical Engineering 2012;46:105–123.

[26] Tawarmalani M, Sahinidis NV. A polyhedral branch-and-cut approach to global optimization. Mathematical Programming 2005;103(2):225–249.

[27] Schweidtmann AM, Mitsos A. Deterministic global optimization with artificial neural networks embedded. Journal of Optimization Theory and Applications 2019;180(3):925–948.

[28] Tao M, Li J, Theodoropoulos C. Reduced model-based global optimisation of large-scale steady state nonlinear systems. In: Computer Aided Chemical Engineering, vol. 46 Elsevier; 2019.p. 1039–1044.

[29] Houska B, Chachuat B. Global optimization in Hilbert space. Mathematical programming 2019;p. 1–29.

[30] Faure H. On the star-discrepancy of generalized Hammersley sequences in two dimensions. Monatshefte für Mathematik 1986;101(4):291–300.

[31] de Aguiar PF, Bourguignon B, Khots M, Massart D, Phan-Than-Luu R. D-optimal designs. Chemometrics and intelligent laboratory systems 1995;30(2):199–210.

[32] Loh WL, et al. On Latin hypercube sampling. Annals of statistics 1996;24(5):2058–2080.

[33] Boukouvala F, Floudas CA. ARGONAUT: AlgoRithms for Global Optimization of coNstrAined grey-box compUTational problems. Optimization Letters 2017;11(5):895–913.

[34] Henao CA. A superstructure modeling framework for process synthesis using surrogate models. PhD thesis, The University of Wisconsin-Madison; 2012.

[35] Eason J, Cremaschi S. Adaptive sequential sampling for surrogate model generation with artificial neural networks. Computers & Chemical Engineering 2014;68:220–232.

[36] Liu H, Ong YS, Cai J. A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design. Structural and Multidisciplinary Optimization 2018;57(1):393–416.

[37] Richardson M. Principal component analysis. URL: http://people maths ox ac uk/richardsonm/SignalProcPCA pdf (last access: 35 2013) Aleš Hladnik Dr, Ass Prof, Chair of Information and Graphic Arts Technology, Faculty of Natural Sciences and Engineering, University of Ljubljana, Slovenia ales hladnik@ ntf uni-lj si 2009;6:16.

[38] Hotelling H. Analysis of a complex of statistical variables into principal components. Journal of educational psychology 1933;24(6):417.

[39] Berkooz G, Holmes P, Lumley JL. The proper orthogonal decomposition in the analysis of turbulent flows. Annual review of fluid mechanics 1993;25(1):539–575.

[40] Jirsa VK, Friedrich R, Haken H, Kelso JS. A theoretical model of phase transitions in the human brain. Biological cybernetics 1994;71(1):27–35.

[41] Park S, Lee JJ, Yun CB, Inman DJ. Electro-mechanical impedance-based wireless structural health monitoring using PCA-data compression and k-means clustering algorithms. Journal of intelligent material systems and structures 2008;19(4):509–520.

[42] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural networks 1989;2(5):359–366.

[43] Yu H, Wilamowski BM. Levenberg-marquardt training. Industrial electronics handbook 2011;5(12):1.

[44] MacKay DJ. A practical Bayesian framework for backpropagation networks. Neural computation 1992;4(3):448–472.

[45] Hastie T, Tibshirani R, Friedman JH, Friedman JH. The elements of statistical learning: data mining, inference, and prediction, vol. 2. Springer; 2009.

[46] Misener R, Floudas CA. ANTIGONE: algorithms for continuous/integer global optimization of nonlinear equations. Journal of Global Optimization 2014;59(2-3):503–526.

[47] Rehfeldt D, Koch T. SCIP-Jack—a solver for STP and variants with parallelization extensions: An update. In: Operations Research Proceedings 2017 Springer; 2018.p. 191–196.

[48] Smith JD, Neto AA, Cremaschi S, Crunkleton DW. CFD-based optimization of a flooded bed algae bioreactor. Industrial & Engineering Chemistry Research 2012;52(22):7181–7188.

[49] Johnson SG, The NLopt nonlinear-optimization package, ab-initio. mit. edu/nlopt; 2015.

[50] Keßler T, Mertens N, Kunde C, Nentwich C, Michaels D, Engell S, et al. Efficient global optimization of a novel hydroformylation process. In: Computer Aided Chemical Engineering, vol. 40 Elsevier; 2017.p. 2113–2118.

[51] Misener R, Gounaris CE, Floudas CA. Global optimization of gas lifting operations: A comparative study of piecewise linear formulations. Industrial & Engineering Chemistry Research 2009;48(13):6098–6104.

[52] Misener R, Floudas C. Piecewise-linear approximations of multidimensional functions. Journal of optimization theory and applications 2010;145(1):120–147.

[53] Floudas CA. Nonlinear and mixed-integer optimization: fundamentals and applications. Oxford University Press; 1995.

[54] Jensen KF, Ray WH. The bifurcation behavior of tubular reactors. Chemical Engineering Science 1982;37(2):199–222.

[55] Lee JH, Shin J, Realff MJ. Machine learning: Overview of the recent progresses and implications for the process systems engineering field. Computers & Chemical Engineering 2018;114:111–121.

[56] Grimstad B, Andersson H. ReLU networks as surrogate models in mixed-integer linear programs. Computers & Chemical Engineering 2019;131:106580.

[57] Belotti P, Liberti L, Lodi A, Nannicini G, Tramontani A. Disjunctive inequalities: applications and extensions. Wiley Encyclopedia of Operations Research and Management Science 2010;.

[58] Wei Z, Li X, Xu L, Tan C. Optimization of operating parameters for low NO x emission in high-temperature air combustion. Energy & Fuels 2012;26(5):2821–2829.

[59] Botoeva E, Kouvaros P, Kronqvist J, Lomuscio A, Misener R. Efficient verification of relu-based neural networks via dependency analysis. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34; 2020. p. 3291–3299.

[60] Anderson R, Huchette J, Tjandraatmadja C, Vielma JP. Strong convex relaxations and mixed-integer programming formulations for trained neural networks. arXiv preprint arXiv:181101988 2018;.