Data-driven Model Reduction-based Nonlinear MPC for Large-Scale Distributed Parameter Systems

Weiguo Xie[1], Ioannis Bonis, and Constantinos Theodoropoulos[*]

School of Chemical Engineering and Analytical Science, University of Manchester, Sackville Street, Manchester M13 9PL, UK.

Abstract

Model Predictive Control (MPC) has been effectively applied in process industries since the 1990s. Models in the form of closed equation sets are normally needed for MPC, but it is often difficult to obtain such formulations for large nonlinear systems. To extend nonlinear MPC (NMPC) application to nonlinear distributed parameter systems (DPS) with unknown dynamics, a data-driven model reduction-based approach is followed. The proper orthogonal decomposition (POD) method is first applied off-line to compute a set of basis functions. Then a series of artificial neural networks (ANNs) are trained to effectively compute POD time coefficients. NMPC, using sequential quadratic programming is then applied. This paradigm combines elements of gain scheduling, NMPC, model reduction and ANN for effective control of nonlinear DPS. The novelty of this POD model reduction-based MPC is to apply POD's highly efficient linear decomposition and convert detailed space-state model to reduced model with function of only 1 dimensional in time. The stabilization/destabilisation of a tubular reactor with recycle is used as an illustrative

[1] Currently at the University of Queensland, Australia

[*] Corresponding author: k.theodoropoulos@manchester.ac.uk

example to demonstrate the efficiency of our methodology. Case studies with inequality constraints are also presented.

Keywords: Proper Orthogonal Decomposition, Nonlinear Model Predictive Control, sequence of Artificial Neural Networks, Distributed Parameter Systems, control of highly nonlinear systems.

1. Introduction

Model Predictive Control (MPC) has been widely used in the process industries. With this technique, online optimization is performed successively at each control timestep, using the underlying model to predict future dynamics for a certain prediction horizon and resulting in a sequence of (tentative) control moves for a given control horizon. Of these moves only the first one is implemented and the rest are discarded. In general, nonlinear MPC is mostly used in batch operations, while linear MPC is more often applied in continuous operations [1]. At the same time, a high-dimensional non-linear model is in general expensive to evaluate and utilizing it in a nonlinear MPC control strategy incurs high computational cost, which may prove problematic in real-time applications.

Model reduction can be used to significantly reduce the dimensionality of complex nonlinear dynamic systems, which, in turn, can lead to the successful design and implementation of nonlinear MPC [2 – 8]. Effective model reduction strategies for control applications have the following features:

- are implemented in automated procedures,
- lead to a good approximation of the original system,

- preserve the dynamic properties of the complex system and
- are highly efficient in reducing the computational cost.

Among the most effective model reduction methods is Proper Orthogonal Decomposition (POD) that has been applied in a non-linear MPC framework [8 - 9] and also for MPC in conjunction with mesoscopic simulators [10]. POD is also termed Karhunen–Loève expansion and Principal Component Analysis. Although this class of methods may rely exclusively on equations-based reduction, usually what is implemented is a variant of the reduction process employing the snapshot method [11].

We recently proposed an off-line model reduction technique based on POD and combined with Trajectory Piecewise-Linearization (TPWL) method for producing low-order linear MPC controllers for nonlinear large-scale distributed parameter systems of partial differential equations [2 - 5]. This POD-Finite Element (FEM) based reduced model is nonlinear only in the time dimension. However, this method requires the detailed governing equations and the application of Galerkin projection.

Artificial Neural Networks (ANN), inspired by the structure and functional properties of biological neural networks, are composed of interconnected elements (neurons) with certain functions. ANN models provide a response on the given information from input layers. The first attempt to produce highly complex behaviors was made in 1943 by using many basic “all-or-none” artificial neurons [12]. Hebbian learning has been introduced from the perspective of the psychologist [13], which opened the door for associative learning for ANNs. Since then, with the development of the modern computers, applications of ANNs have been made in many areas including system identification and control, process modeling, data processing, visualization etc. It has

been found that multilayer feed-forward networks with as few as one hidden layer can be universal approximators for many nonlinear functions [14-15]. More recently, ANN associated with POD has been developed for nonlinear dynamic models of distributed reacting systems [4, 16]. A dynamic radial basis function (RBF) neural network has been used to model distributed parameter systems (DPSs) and then implemented in MPC configurations [17].

This paper describes the development of a data-driven model reduction-based artificial neural network technique for use within an NMPC framework. As mentioned above, the POD-FEM-based reduced model converts nonlinear large systems into one dimension, time being the single model variable, and then applies ANN for time coefficients modeling. After these, a nonlinear MPC control strategy can be applied for nonlinear large-scale distributed systems. This data-driven approach can be used in any "black-box" system. An important feature of the proposed methodology is the efficient handling of inequality constraints, by incorporating them in the formulation of the reduced optimization problem. Generally speaking, one of the strengths of MPC is its ability to handle constraints. Although not all control problems are constrained, many do require the satisfaction of nonlinear inequalities. The need to tackle inequalities is accommodated in purpose-designed algorithms for predictive and optimal control [18-19]. Such constraints could express design constraints for process operation [20], economic/environmental [21] or stability conditions [22].

The proposed methodology has two main features: the reduction in computational time in comparison to NMPC based on first-principle models and the capability to handle black-box systems. Computational savings are attributed to employing model order reduction for the mathematical simplification of the system at hand. POD not only reduces the dimensionality of the problem, but also results in a model consisting

of a product of a (constant) term expressing the spatial variation and a time-varying one. Usually engineering systems are dissipative and can be described with a relatively low-dimensional reduced model, resulting in significant computational savings over using the full order model. In the proposed strategy, the selection of ANNs and switching rules also affect the online computational time. Of equal importance to reducing cost is the ability to handle black-box systems, for which either a numerical integrator exists that allows no explicit access to the equations, or only experimental data are available. Hence, we effectively tackle with the problem of controlling a black-box system non-neglecting its spatial variation and utilizing the dominant dynamics identified offline.

The rest of the paper is organized as follows: In section 2, a brief introduction of POD model reduction, artificial neural network, and procedure of POD-ANN is given. In section 3, three nonlinear MPC control case studies based on a tubular reactor are used to illustrate the features of this technique. Finally, the conclusions of this work are discussed in section 4.

2. POD-ANN MODEL REDUCTION

2.1 Proper Orthogonal Decomposition (POD) Model Reduction

Proper orthogonal decomposition (POD) is based on the spectral theory of compact, self-adjoint operators expressed in the Karhunen-Loeve decomposition theorem [23]. POD is a powerful method to capture the most “energy” in an average sense for efficient linear decomposition in terms of data compression [24]. The “energy” of a given mode is calculated from the magnitude of the eigenvalue corresponding to that mode. The reduced set of global eigenfunctions (basis functions) can either be

obtained analytically or statistically, by employing the method of snapshots [11]. The latter approach is used more often, as an analytical solution does not exist for all classes of systems, and is the approach applied here.

The main feature of POD is that it distinguishes the temporal and the spatial variations and approximates the original nonlinear system by a low-order nonlinear model that comprises terms exhibiting only one kind of variation (time or space). Spatial variation is captured by the basis functions, which roughly speaking constitute a basis for the feasible point subspace of the original system, i.e. ideally all feasible solutions can be expressed as a linear combination of these vectors. Starting from an initial condition, the subsequent state variable vectors are computed as linear combinations of the basis functions. Time coefficients express the time variation of the system at hand and are the weights of the linear combination mentioned above. Generally speaking these coefficients are given by nonlinear expressions that can be obtained either analytically [3] or statistically. In this work, we employ appropriately trained ANNs for the calculation of these functions.

In figure 1, the solid arrow pathway shows the common procedure for the POD model reduction method. The steps of constructing the POD model are:

1. Empirically collect time evolving *n* data points from the dynamic model or from experiments for the chosen appropriate range of parameters;
2. Express the snapshots as perturbations of their mean: calculate the mean snapshot $\bar{x}(z) = \frac{1}{n}\sum_{j=1}^{n} x(z)$ and subtract $\bar{x}(z)$ from each sample so that the modified set has a mean of zero;

3. Construct a two-point correlation matrix, *C*, of size *n*×*n* from the sample set of the previous step;
4. Calculate the *m* orthonormal empirical global basis functions ϖ_j, *j*=1,…,*m* of the POD model, where *m*<<*N* (*N* being the dimension of the full model and *m* the dimension of the POD model). ϖ_j are calculated by solving an eigenvalue problem of the correlation matrix: $CW=\lambda W$ and subsequently $\varphi_n(z) = \sum_{j=1}^{M} W_j\, x(z,t_j), \quad n=1,2,...,m$. The value of *m* is determined by setting a threshold for capturing the system's "energy": $E_{tot} = \sum_{j=1}^{M} \lambda_j$;
5. Express the state variables *x(z,t)* of the system (where *z* are spatial coordinates) as linear combinations of $\varpi(z)$ and some time coefficients *a(t)*:

$$x(z,t) = \sum_{j=1}^{m} a_j(t)\varpi_j(z) + \bar{x}(z) \tag{1}$$

6. Compute expressions for the time coefficients. In this work, these expressions are the appropriately trained ANNs.

As mentioned in the introduction, the reduced non-linear model from the POD method is 1-dimensional, time being the only variable, irrespective of the dimensionality of the original problem. POD has been applied on many systems e.g. to produce low-order models for the nonlinear MPC of parabolic PDEs systems [25], and for the optimization [26] and control [16] of reduced order models of transport-reaction processes. However, the commonly used Galerkin-POD model reduction methods is not suitable for "black-box" systems, since the equations of the system are explicitly required for the derivation of expressions for the time-evolution of the time coefficients via the Galerkin projection of the original set of equation onto the basis

functions. Employing ANNs hurdles this obstacle by rendering the procedure completely data-driven. No access to the original set of equations is required, nor is residual information used, as in other morel reduction techniques. Here only state variable information is needed for a training set of data, but additional information may be utilized if available as we will discuss in Section 2.3.

2.2 Artificial Neural Network

The backbone of this work is employing a sequence of ANNs for the calculation of the time coefficients. Because appropriately trained artificial neural networks can be good approximators for any nonlinear function, ANN can be used to compute POD time coefficients. The Galerkin projection step can then be potentially avoided.

A standard feed-forward artificial neural network without recurrence is shown in figure 2. There are three main layers including the input layer, hidden layers, and the output layer. Although, it has been found that this kind of feed-forward network with one hidden layer can approximate any nonlinear function [14-15], it is quite difficult to capture the dynamic behaviour of nonlinear systems using limited neurons.

An Elman neural network [27] from the Neural Network Tool-box in MATLAB, shown in figure 3, is a type of recurrent neural network. Where, p is the control input and y is the output; all LW, IW, $a_1(k-1)$, and b are obtained by network training; $\tan sig(x) = \frac{2}{1+\exp(-2x)} - 1$ and $purelin(x) = x$.

This neural network can recognize and generate a temporal pattern, which is suitable for simulation of dynamic systems. The advantage of Elman Neural Network comparing to feed-forward networks is the recurrent connections of the context layer

provide the system with a short-term memory. The hidden units do not only observe the actual input but, via the context layer, also obtain information on their own state at the last time step [28]. This feature can be useful in model predict control. Therefore, the Elman neural network has been chosen for the POD-ANN model reduction method for the non-linear dynamic large-scale distributed system in this research.

In this work, rather than having one ANN that yields values for the time coefficients, given the time and the values of the control variables, we consider having a sequence of ANNs in a gain-scheduling approach. Namely, although the structure of the nonlinear predictive controller does not alter at run-time, the underlying model does. It is outside the scope of this work to optimize the regions of validity for each ANN. Ideally, one would analyze the dynamics of the original system and identify regions of similar dynamics that can be modelled using the same ANN. The boundaries between regions are model switching points. The trivial solution to this consideration is to partition the temporal domain evenly and disregard the dynamics of the system. This is the approach followed in this paper and in practice it may prove the only realistic approach for a real system, the analysis of which would be difficult and time consuming, or even impossible if only a black-box simulator is available. As far as the system of the case study is concerned, in recent works we explored optimal (linear) model switching, both based on position in state space [29], and time [4].

The structure of the ANNs does not alter and therefore the number of neuron layers and the number of neurons is the same throughout the sequence of ANNs. Switching between members of the sequence is time-based and is automatic. Therefore, the proposed technique may be seen as multiple NMPC. Using a sequence of ANNs rather than a single one gives us more flexibility and enables us to handle systems of higher nonlinearity and richer behavior in parametric space.

2.3. POD-ANN Procedure

The procedure for the POD-ANN method includes similar steps to the typical implementation the POD method (Figure 1) apart from the Galerkin projection step. In the proposed strategy, the values for the POD time coefficients are given by a sequence of ANNs. Training of the neural network is very important, as it determines the quality of the approximation. In real time, whenever a function evaluation is required in the context of the NMPC optimization problem, the suitable ANN from the model pool is selected and is fed the values of the control variables. The output is the values of the time coefficients, which are then combined with the basis function vectors and the mean state vector as in Equation (1), to give an estimation for the state variables vector. It is this (approximate) state vector that is used for the evaluation of the objective function.

In order to train the neural network efficiently, a least square optimization has been applied to sampling cases to catch the dynamics of the time coefficients in the POD-ANN reduced model. Training is performed offline using process data (values for the state variables) from purposefully-designed experiments. The optimal design of these experiments is outside the scope of this work and indeed there are many ways that this can be achieved. Here we use Taguchi's orthogonal experimental design to obtain values for the control variables in each of the experiments. Offline simulations using the full-scale, high-dimensional, model are run and the resulting process data are used in a least-squares problem which minimizes the distance between the actual state vector and the POD-ANN estimation of it. This problem here is solved using the Levenberg-Marquardt backpropagation method.

2.4. Nonlinear MPC

The basis for the work presented here is Nonlinear Model Predictive Control. Although linear MPC is widely used in process industry, its nonlinear variant is usually avoided and applied only in cases where the nonlinearity cannot be disregarded, either because the underlying system is indeed very nonlinear, or the operating conditions vary significantly during normal operation. This is the reason why NMPC is usually only used in batch operations, fine chemicals production and other specialty applications.

The main concept of NMPC is the same as in its linear version: the controller is predicting a sequence of future moves (values for the control variables), that minimize the difference between the predicted state variables and the desired ones (reference trajectory) and simultaneously minimize the required control energy, i.e. the size of the subsequent control moves difference. This can be formulated in an objective function, as follows:

$$J = \min_{\Delta U}(\Big(x(t) - x_{ref}(t)\Big)^T \cdot Q \cdot (x(t) - x_{ref}(t)) + (\Delta U)^T \cdot R \cdot \Delta U) \quad (2)$$

Where $x(t)$ is the state variables vector, $x_{ref}(t)$ is the desired trajectory that need not be constant, ΔU is the difference of value of a control variables from the previous one. Matrices Q and R are weights that can be used as handles to signify the relative importance of each state and control variable and the relative importance of path following versus the control energy minimization. The optimization problem may have inequality constraints, such as control variables rate of change limits, and state variable constraints and equality constraints expressing physical relations between variables.

The vector $x(t)$ is given by Equation (1), in which the constant (at run-time) quantities $\varpi(z)$ and $\bar{x}(z)$ participate, as well as the time coefficients $a(t,u)$ given by the appropriate ANN model. The sequence of ANNs is wrapped in a procedure that selects the appropriate ANN, evaluates it and combines its output with the spatial part of the POD model to yield a state vector estimate.

Solving the NMPC optimization problem is performed online using a deterministic quasi-Newton method. The optimizer employs the wrapper mentioned above for function evaluations. Typically, the size of the optimization problem is small and full reconstruction of the state vector can be avoided as we will show in the next Section. In such cases, the use of special optimizers that are meant for large-scale systems can be avoided.

The proposed strategy is also compatible with stochastic and non-gradient-based deterministic optimizers, which typically incur more function evaluations. This is possible because model order reduction effectively reduces the cost of function evaluations. Using a stochastic optimizer might also tackle the issue of getting trapped at a local minimum and miss the global one, which is always a possibility when using a local nonlinear optimization method.

3. Case study

3.1 The tubular reactor with recycle

The tubular reactor with recycle depicted in figure 4 can be modeled by two sets of partial differential equations [30] in a spatial domain $z \in [0,1]$:

$$C_t = -\frac{\partial C}{\partial z} + \frac{1}{Pe_C}\frac{\partial^2 C}{\partial z^2} - f(C,T)$$
$$T_t = -\frac{\partial T}{\partial z} + \frac{1}{Pe_T}\frac{\partial^2 T}{\partial z^2} + B_T f(C,T) + \beta_T (T_c - T) \quad (3)$$

Where, C and T are dimensionless concentration and dimensionless temperature, respectively. T_C corresponds to the dimensionless temperature of the cooling medium and $f(C,T) = B_C C \exp(\frac{\gamma T}{1+T})$ is the reaction term. The values of the parameters used are: Pe_C=7.0, Pe_T=7.0, B_C=0.1, B_T=2.5, γ=10.0 and β_T=2.0, with Pe_C, Pe_T being the Peclet numbers for mass and heat transport respectively, B_C being the (dimensionless) heat transfer coefficient, B_T being the (dimensionless) temperature rise and γ being the activation energy. For a given recycling ratio r, the boundary conditions for concentration and temperature at $z = 0$ are [31]:

$$\frac{\partial C}{\partial z} = -Pe_C[(1-r)(1+C_0) + rC(t,1) - C(t,0)]$$
$$\frac{\partial T}{\partial z} = -Pe_T[(1-r)(1+T_0) + rT(t,1) - T(t,0)] \quad (4)$$

The boundary conditions at $z = 1$ are $dC/dz = 0$ and $dT/dz = 0$. The reactor exhibits oscillations at C_0=T_0=T_c=0 for r=0.5 [32]. The model was discretized into 16 nodes for spatial domain, and the FEM has been applied to solve the resulting ordinary differential equations. The data from the FEM solution of the full-scale model above are used to train and/or test the ANNs used in our methodology.

3.2 Control Objective

It can be seen that the tubular reactor shows stable behavior for r=0 (Fig. 5a) while it undergoes sustained oscillations for r=0.5 (Fig. 5b).

Here we present three case studies with different control objectives:

For the first case study the control objective was to stabilize the reactor with *r*=0.5 to behave like the system with *r*=0 by introducing a number of jacket temperature zones (actuators). The objective function is as follows:

$$J = \min_{\Delta u} \left(T(t) - T_{ref}(t)\right)^T Q\left(T(t) - T_{ref}(t)\right) + \Delta U^T R \Delta U \tag{5}$$

Where, $T_{ref}(t)$ is the reference temperature (r=0) and ΔU is the control on the actuators.

Then, the following objective function can be obtained by applying equation 1 to replace temperature term.

$$J = \min_{\Delta u} \left((\sum_{k=1}^{m} \alpha_{k\_T}(t) \varpi_{k\_T}(x) + \overline{T_{16}}) - T_{ref}(t) \right)^T Q \left((\sum_{k=1}^{m} \alpha_{k\_T}(t) \varpi_{k\_T}(x) + \overline{T_{16}}) - T_{ref}(t) \right) + \Delta U^T R \Delta U \tag{6}$$

Subject to:

$$a_1(t) = \tanh(IW_{1,1} \cdot U + LW_{1,1} \cdot a_1(t-1) + b_1) \tag{7}$$

$$a_2(t) = \tanh(LW_{2,1} \cdot a_1(t) + LW_{2,2} \cdot a_2(t-1) + b_2) \tag{8}$$

$$\alpha_{k\_T}(t) = LW_{3,2} \cdot a_2(t) + b_3 \tag{9}$$

The above equations (7), (8) and (9) are from Figure 3 with 3 layers Elman neural network (tansig/tansig/purlin), in which tansig was replaced by tanh of the mathematical form. Where, *LW*, *IW*, and *b* are obtained by ANN training, and $a_1(t)$, $a_2(t)$ are internal output parameters of the first two layers of the neural network.

For the second case study the control objective was to destabilize the reactor with r=0 to behave like the system with r=0.5 by introducing a number of jacket temperature zones. The objective function and constraints have a similar form to equations 4 to 8. Here $T_{ref}(t)$ is the reference state (r=0.5). This case study illustrates the capability of the proposed method to enforce the desired dynamic behavior on the closed-loop system. Whereas the first case was about stifling the dynamics, here we consider exciting them. Although this case was designed on academic merits rather than engineering relevance, in some cases it may be desired that the advanced controller excites the system dynamics to a point, so that parameter identification may be performed simultaneously to control [33].

For the third case study the control objective was based on the second case study and nonlinear inequality constraints were added to limit the variance of the state variables within the predictive horizons.

$$\text{Var(CA)}=\text{E[CA]}^2-(\text{E[CA]})^2 \text{ and } \text{Var(TA)}=\text{E[TA]}^2-(\text{E[TA]})^2 \qquad (10)$$

Where, E[CA] (or E[TA]) is the expected (mean) value of CA (or TA),

$$CA=[\sum_{j=1}^{m} a\_c_j(t_1)\varpi\_c_j(z) \;,\; \cdots \;,\; \sum_{j=1}^{m} a\_c_j(t_p)\varpi\_c_j(z)\;] \quad \text{and}$$

$$TA=[\sum_{j=1}^{m} a\_t_j(t_1)\varpi\_t_j(z) \;,\; \cdots \;,\; \sum_{j=1}^{m} a\_t_j(t_p)\varpi\_t_j(z)\;],$$

$\varpi\_c_j(z)$ (or $\varpi\_t_j(z)$) is the jth basis function for concentration (or temperature) at space point z, and $\alpha\_c_j(t_i)$ (or $\alpha\_t_j(t_i)$) is the jth time coefficient corresponding to the jth basis function for concentration (or temperature) at time point t_i (i=1, t_p) where t_p is the predictive time horizon.

3.3 Data Sampling

A method based on orthogonal experimental design methodology has been applied for data sampling. There were 8 jacket temperature zones (actuators) for the implementation of the computed control actions. In Taguchi's orthogonal experimental design, the use of the L12 orthogonal array has been highly recommended and many successful cases have been reported [34]. The first 8 columns of the L12 (2^{11}) orthogonal array, as listed in Table 1, have been used in this work.

If Heaviside step functions (whose value is 0 or 1) are applied to facilitate sampling, then for 8 actuators we need $2^8 = 256$ runs. Therefore, only 12 runs for sampling based on orthogonal experimental design have dramatically reduced the number for experiments. Taking 11 samples over the range of (dimensionless) cooling temperatures [-1, 1], we have $12*11=132$ runs. The full-scale FEM model was used for sampling and the sampling time was 15 time units. As far as ANN training is concerned, the maximum number of epochs to train was 1,000 and the minimum performance gradient was 10^{-5}. In general, it will improve the accuracy of the method by increasing the maximum number of epochs; however, the trade-off will be longer training time. The number chosen was 1,000 due to manageability of the work with considerable accuracy.

3.4 POD-ANN Model Reduction

In the first case study (reactor stabilization), it is shown in figure 6a and 6b that five global basis functions for concentration and temperature were computed based on the

132 samples collected, since m=5, eigenfunctions for concentration (or temperature) capture 99.7% (or 98.3%) of the system's energy.

The comparison between the full and reduced model for the dynamics of the reactor middle and output points is shown in figure 7a (concentration) and 7b (temperature). The reduced model gave good predictions of the complex reactor dynamics.

In the second and third case study to destabilize the reactor, it is shown in figure 8a and 8b that five global basis functions for concentration and temperature were computed based on the 132 samples collected with r=0. The number of eigenfunctions considered was m=5, for concentration (or temperature) to capture 99.5% (or 98.4%) of the system's energy.

The comparison between the full and reduced model for the dynamics of the reactor middle and output point is shown in figure 9a (concentration) and 9b (temperature). The sequence of any one ANN was chosen by equally 5 time steps (with each time step of 0.01 secs). The selection of ANN-based surrogate model follows is time-based and results in an approach similar to gain scheduling. Since both the reference trajectory and the open-loop system dynamics are known a priori, time-scheduling of ANN can be performed offline. This strategy is based on the implicit assumption that the closed-loop system dynamics would be reasonably similar to the reference. Indeed, since a high-resolution surrogate model is employed, this assumption is intuitively legitimate and can be practically validated with offline numerical experiments, such as the ones in Figures 7 and 9. The reduced model again successfully computed the complex reactor dynamics.

3.5 Nonlinear MPC

Nonlinear MPC has been applied using the MATLAB NAG Toolbox to obtain the control laws for the three case studies. Control time horizon is one time step (0.01 sec) in the case studies, and prediction time horizon is three time steps (0.03 sec). An implementation of a quasi-Newton algorithm included in theMATLAB NAG Toolbox, has been applied in the first case study. In Figure 10, results of the nonlinear MPC using 8 actuators (cooling zones) to stabilize the tubular reactor system are shown. Figure 10a shows the control law for the 1st and 8th actuators. The control output and the reference profile are shown in Figure 10b, demonstrating that the reactor can be efficiently stabilized using our technique.

The same quasi-Newton algorithm has also been applied in the second case study. In Figure 11, results of the nonlinear MPC using 8 actuators to destabilize the tubular reactor system are shown. Figure 11a shows the control law for the 1st and 8th actuators, respectively. The control output and the reference profile are shown in Figure 11b. As it can be seen the closed-loop dynamics follow the reference trajectory successfully. This illustrates that our nonlinear MPC control based on POD-ANN can be used to track any kind of (arbitrary) reference profile.

An implementation of the sequential quadratic programming (SQP) method included in the NAG Toolbox for MATLAB has been applied in the third case study. This optimizer has been favored over the quasi-Newton Algorithm used for case studies 1 and 2 due to the existence of inequality constraints. This method allows constraints including simple bounds on the variables, linear constraints, and smooth nonlinear constraints. The inequality constraints, $Var(CA) \leq 0.1$ and $Var(TA) \leq 0.1$, have been used in the third case study. In Figure 12, results of the nonlinear MPC using 8 actuators to

destabilize the tubular reactor system are shown. Figure 12a shows the control law for the 1^{st} and 8^{th} actuators. The control output and the reference profile are shown in Figure 12b, again demonstrating that the controlled system can successfully follow reference trajectory. It can also be seen from figure 13 that only the first 5 time units are affected by inequality constraints posed here.

4. Conclusions

A data-driven, model reduction-based, artificial neural network (ANN) approach has been developed for nonlinear MPC applications to highly nonlinear, distributed parameter systems of high dimensionality. This efficient model reduction-based technique combines the proper orthogonal decomposition (POD) with ANNs. The POD-ANN methodology enables the use of nonlinear MPC for large scale non-linear "black-box" systems. The key features of the proposed work are the reduction of dimensionality using POD, the suitability for handling black-box systems exploiting the data-driven nature of ANNs and the use of a sequence of low-order nonlinear models within the NMPC framework. The existence of a sequence of ANN-based models rather than a single one is a novelty of the work described here and enables tackling systems of high nonlinearity, where training a single ANN would give an insufficiently good approximation of the original system. This method can effectively facilitate the use of nonlinear MPC for large scale distributed systems, as it was demonstrated in three case studies, where stabilization of a tubular reactor undergoing sustained oscillations was performed, destabilization of a tubular reactor, and destabilization of a tubular reactor with inequality constraints. In a subsequent publication, we plan to implement a piecewise linear MPC associated with POD-ANN

for control of large-scale non-linear "black-box" systems, by which even higher computational efficiency can be achieved.

Acknowledgements

The authors would like to acknowledge the financial support of the EC FP6 Project: CONNECT [COOP-2006-31638] and the EC FP7 project CAFE [KBBE-212754].

References

[1] W.P. Heath, G. Li, A.G. Wills, B. Lennox. The robustness of input constrained model predictive control to infinity-norm bound model uncertainty. *ROCOND06, 5th IFAC Symposium on Robust Control Design, Toulouse, France, 5th-7th July; 2006.*

[2] W. Xie, C. Theodoropoulos. An off-line model reduction-based technique for on-line linear MPC applications for nonlinear large- scale distributed systems. *Comput. Aided Chem. Eng.* 28 (2010) 409-414.

[3] W. Xie, I. Bonis, C. Theodoropoulos. Off-line model reduction for on-line linear MPC of nonlinear large-scale distributed systems. *Comput. Chem. Eng.* 35 (2011) 750-757.

[4] W. Xie, I. Bonis, C. Theodoropoulos, Linear MPC based on data-driven Artificial Neural Networks for large-scale nonlinear distributed parameter systems. *Comput. Aided Chem. Eng.* 30 (2012) 1212-1216.

[5] I. Bonis, W. Xie, C. Theodoropoulos, A linear model predictive control algorithm for nonlinear large-scale distributed parameter systems. *AIChE J.* 58 (2012) 801-811.

[6] I. Bonis, C. Theodoropoulos, Model reduction-based optimization using large-scale steady-state simulators. *Chem. Eng. Sci.* 69 (2102) 69-80.

[7] C. Theodoropoulos, Optimisation and Linear Control of Large Scale Nonlinear Systems: A Review and a Suite of Model Reduction-Based Techniques. In: A.N. Gorban, D. Roose, ed. Coping with Complexity: Model Reduction and Data Analysis. Berlin-Heidelberg, Springer. 2010, pp 37-61.

[8] M.H. Li, P.D. Christofides, Optimal control of diffusion-convection-reaction processes using reduced-order models. *Comput. Chem. Eng.* 32 (2008) 2123-2135.

[9] M.R. Garcia, C. Vilas, J.R. Banga, A.A. Alonso, Exponential Observers for Distributed Tubular (Bio) Reactors. *AIChE J.* 54 (2008) 2943–2956.

[10] C. Oguz, M.A. Gallivan, Optimization of a thin film deposition process using a dynamic model extracted from molecular simulations. *Automatica* 44(2008) 1958–1969.

[11] L. Sirovich L. Turbulence and the dynamics of coherent structures, parts i-iii. *Quart. Appl. Math*. 45 (1987) 561-590.

[12] W. McCulloch W, W. Pitts, A logical calculus of ideas immanent in nervous activity. *Bulletin Math. Biophys*. 5 (1943) 115-33.

[13] D.O. Hebb, The organization of behavior. New York: Wiley & Sons; 1949.

[14] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators. *Neural Networks.* 2 (1989) 359–366.

[15] K. Hornik, Some new results on neural network approximation. *Neural Networks.* 6 (1993) 1069–1072

[16] S.Y. Shvartsman, C. Theodoropoulos, R. Rico-Martinez, I.G. Kevrekidis, E.S. Titi, T.J. Mountziaris, Order reduction for nonlinear dynamic models of distributed reacting systems. *J. Proc. Control*. 10 (2000) 177- 184.

[17] E. Aggelogiannaki, H. Sarimveis H. Nonlinear model predictive control for distributed parameter systems using data driven artificial neural network models. *Comput. Chem. Engin.* 32 (2008) 1225-1237.

[18] M. Diehl, A. Walther, H.G. Bock, E. Kostina, An adjoint-based SQP algorithm with quasi-Newton Jacobian updates for inequality constrained optimization. *Optim. Methods Soft.* 25 (2010) 531–552.

[19] M. Von Schwerin, O. Deutschmann, V. Schulz, Process optimization of reactives systems by partially reduced SQP methods. *Comput. Chem. Eng. 24 (*2000) 89–97.

[20] D.D. Brengel, W.D. Seider, Multistep nonlinear predictive controller. *Industr. Engin. Chem. Res.* 28 (1989) 1812–22.

[21] F. Martinsen, L.T. Biegler, B.A. Foss, A new optimization algorithm with application to nonlinear MPC. *J. Proc. Control.* 14 (2004) 853–865.

[22] P.D. Couchman, M. Cannon, B. Kouvaritakis, Stochastic MPC with inequality stability constraints. *Automatica* 42 (2006) 2169 –2174.

[23] E. Wong, Stochastic Process in Information and Dynamical Systems. McGraw-Hill; 1971.

[24] P. Holmes, J.L. Lumley, G. Berkooz, Turbulence, coherent structures, dynamical systems and symmetry. Cambridge University Press; 1996.

[25] J. Baker J, P.D. Christofides, Finite-dimensional approximation and control of non-linear parabolic PDE systems. *Int J Control* 73 (2000) 439–456.

[26] E. Bendersky, P.D. Christofides, Optimization of transport-reaction processes using nonlinear model reduction. *Chem. Eng. Sci.* 55 (2000) 4349–4366.

[27] J.L. Elman, Finding structure in time. *Cognitive Sci*. 14 (1990) 179-211.

[28] H. Cruse, Neural Networks as Cybernetic Systems (2nd and revised edition). bmm615. Brains, Minds and Media, Bielefeld, Germany. (urn:nbn:de:0009-3-6153), published October 2006.

[29] I. Bonis, W. Xie, C. Theodoropoulos, Multiple model predictive control of dissipative PDE systems. *IEEE Trans. Control Syst. Tech.* 22 (2013) 1206-1214.

[30] K.F. Jensen, W.H. Ray, The bifurcation behaviour of tubular reactors. *Chem. Eng. Sci.* 37 (1982) 199–222.

[31] C. Antoniades, P.D. Christofides, Integrated optimal actuator/sensor placement and robust control of uncertain transport-reaction processes. *Chem Eng Sci*. 56 (2001) 4517–4535.

[32] A.A. Alonso, C.E. Frouzakis, I.G. Kevrekidis, Optimal sensor placement for state re-construction of distributed process systems. *AIChE J*. 50 (2004) 1438-1452.

[33] H. Genceli, M. Nikolaou. "New approach to constrained predictive control with simultaneous model identification." AIChE journal 42.10 (1996): 2857-2868.

[34] H.T. Tsai, The partial aliasing patterns for Taguchi's L12 orthogonal array. *Int. J. Qual. Reliability Manag*. 12 (1995) 54-60.

List of Tables:

1. L_{12} (2^{11}) Orthogonal array from Taguchi's orthogonal experimental design (Tsai, 1995), with symbol 1 being replaced by 0 and -1 being replaced by 1.

Table 1. L_{12} (2^{11}) Orthogonal array from Taguchi's orthogonal experimental design (Tsai, 1995), with symbol 1 being replaced by 0 and -1 being replaced by 1.

Run	A	B	C	D	E	F	G	H	I	J	K
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	1	1	1	1	1
3	0	0	1	1	1	0	0	0	1	1	1
4	0	1	0	1	1	0	1	1	0	0	1
5	0	1	1	0	1	1	0	1	0	1	0
6	0	1	1	1	0	1	1	0	1	0	0
7	1	0	1	1	0	0	1	1	0	1	0
8	1	0	1	0	1	1	1	0	0	0	1
9	1	0	0	1	1	1	0	1	1	0	0
10	1	1	1	0	0	0	0	1	1	0	1
11	1	1	0	1	0	1	0	0	0	1	1
12	1	1	0	0	1	0	1	0	1	1	0

List of Figures

11. Nonlinear MPC results for the second case study (a) control law for 1^{st} and 8^{th} actuators (b) control and reference profile.

12. Nonlinear MPC results for the third case study (a) control law for 1^{st} and 8^{th} actuators (b) control and reference profile.

13. Results after implementing inequality constraint $Var(CA) \leq 0.1$ and $Var(TA) \leq 0.1$ for the third case study.

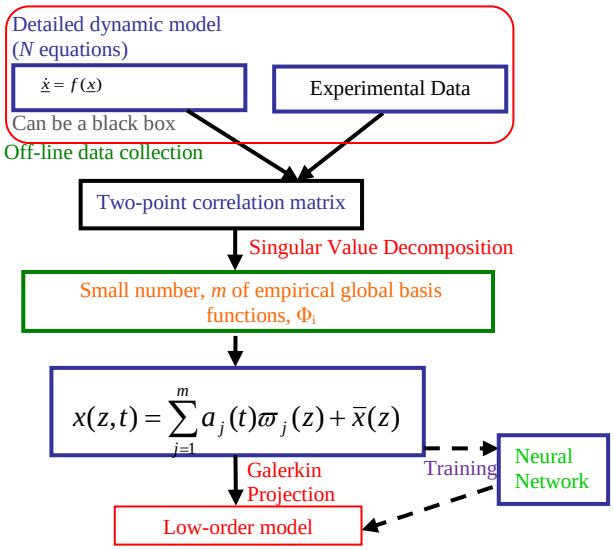


Figure 1. Schematic diagram for POD (with solid arrow) and POD-ANN (with dash arrow) model reduction

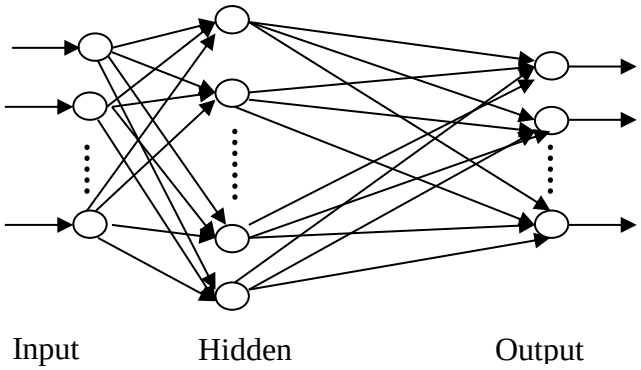


Figure 2. Neural network topology of a standard feed-forward neural network with no recursive connections.

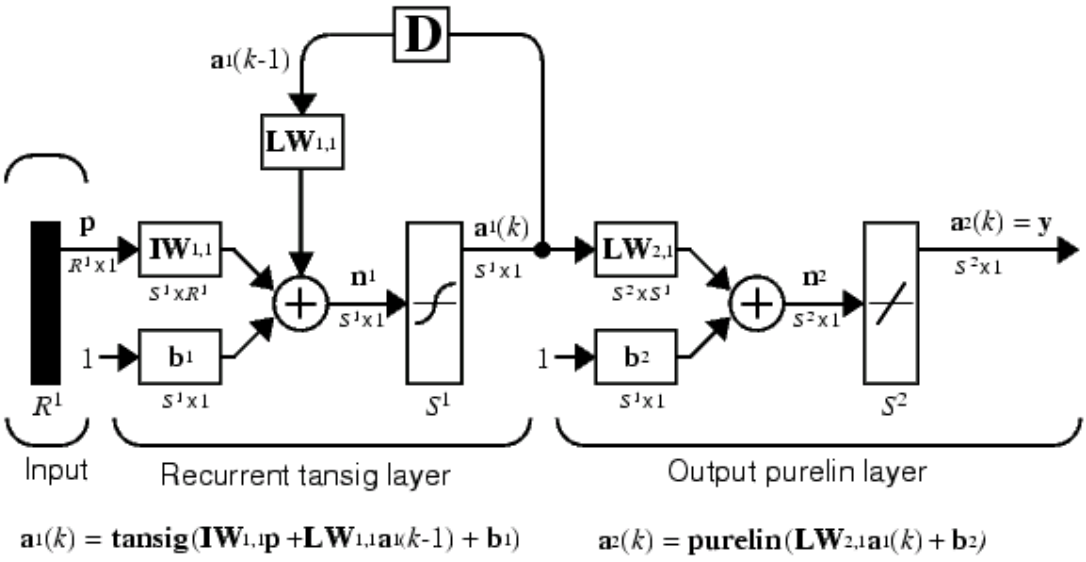


Figure 3. Elman neural network (adapted from Neural Network Toolbox, MATLAB User's Guide).

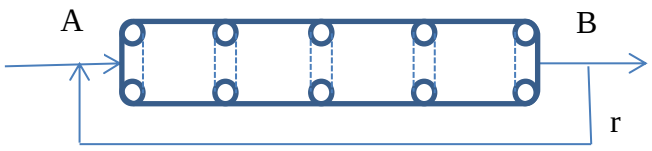
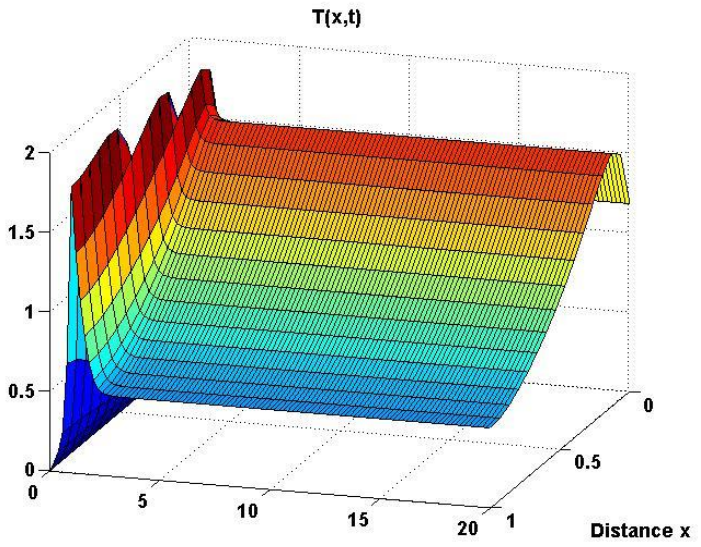
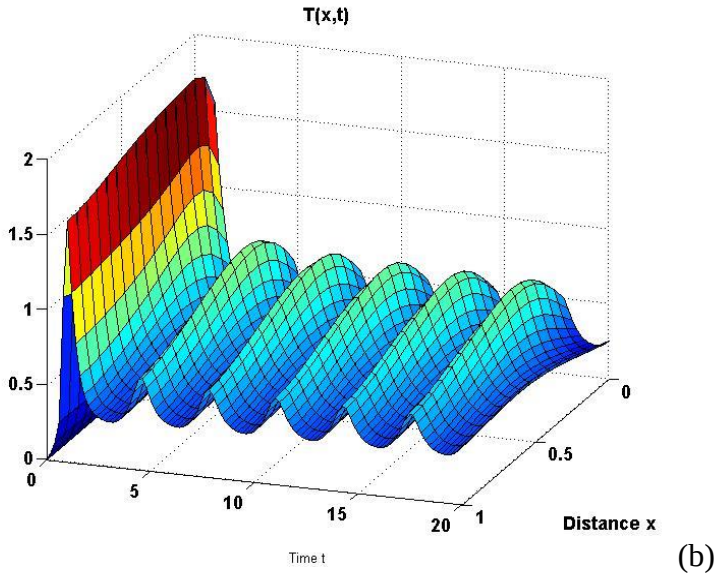


Figure 4. Tubular reactor with recycle.

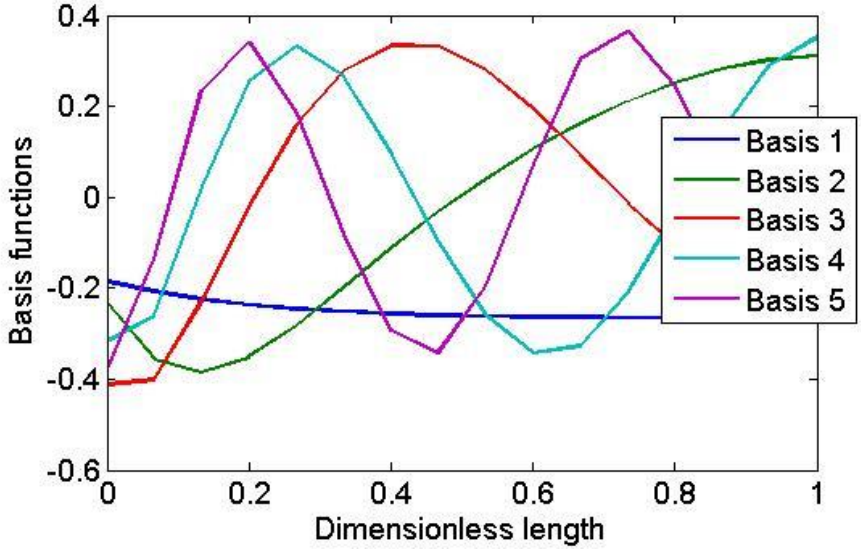


(a)

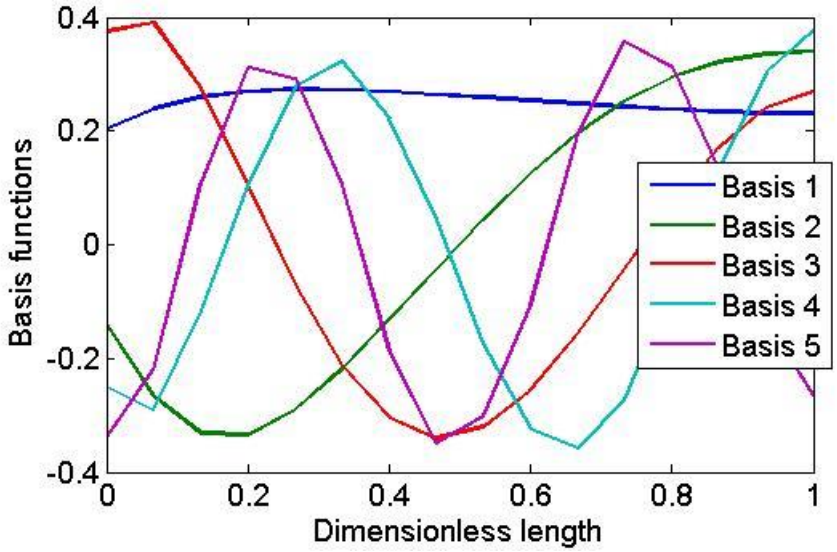


(b)

Figure 5. Temperature profiles of tubular reactor (a) $r = 0$ (b) $r = 0.5$.

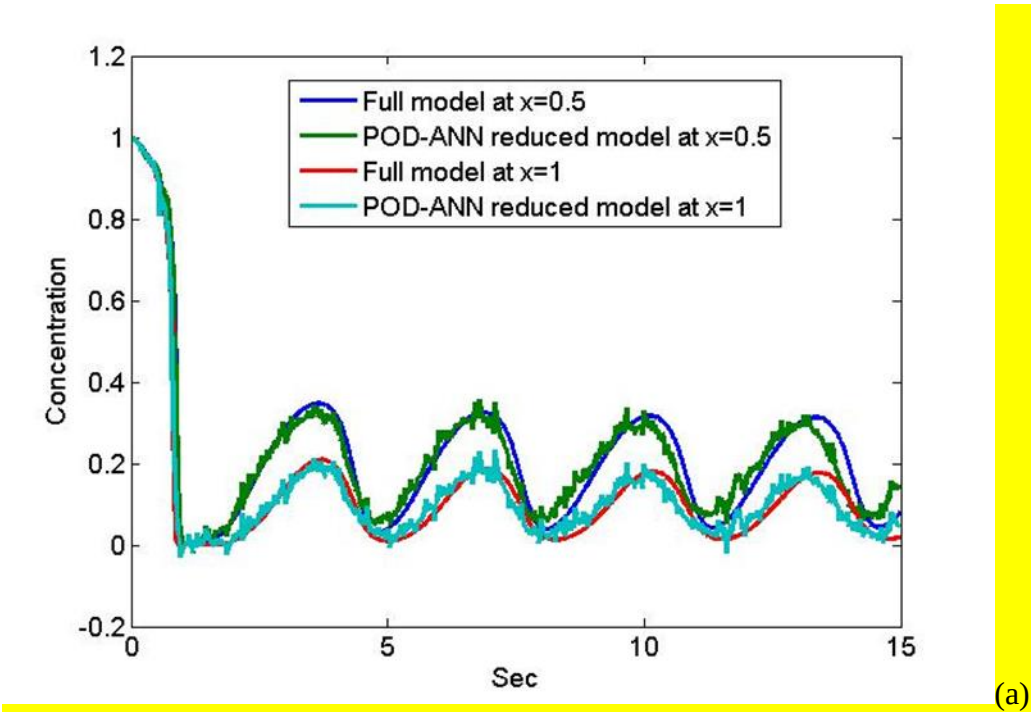


(a)

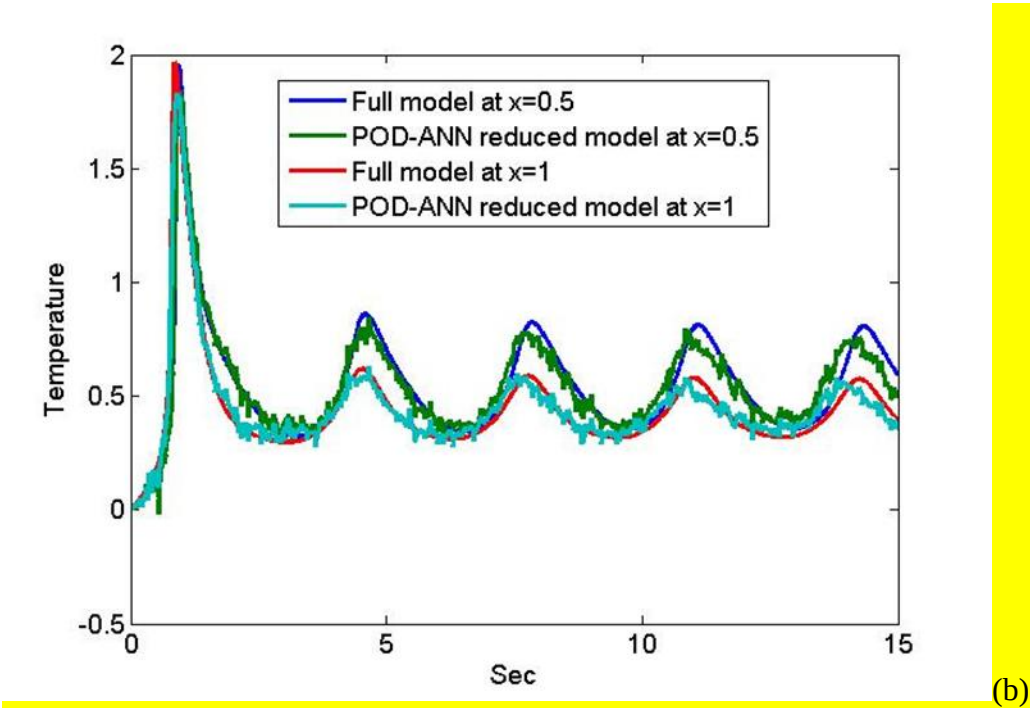


(b)

Figure 6. Global basis functions for (a) concentration (b) temperature from the sampling data of tubular reactor with $r = 0.5$.

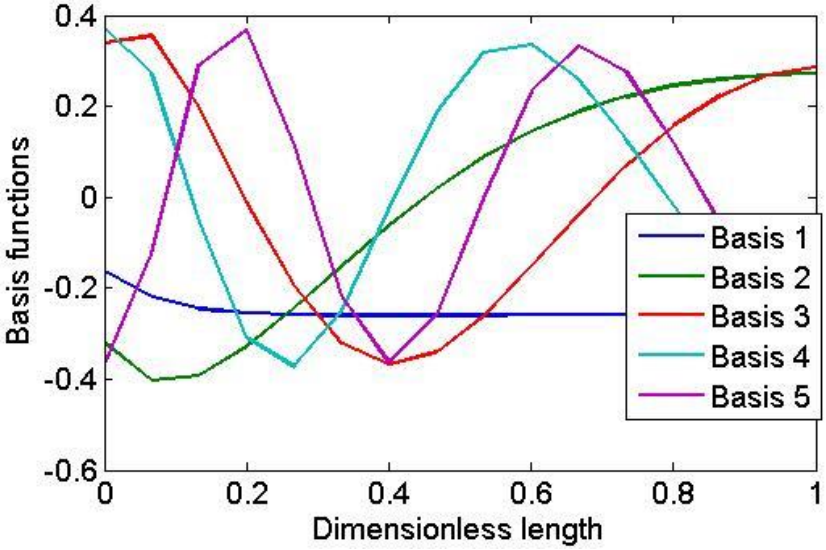


(a)

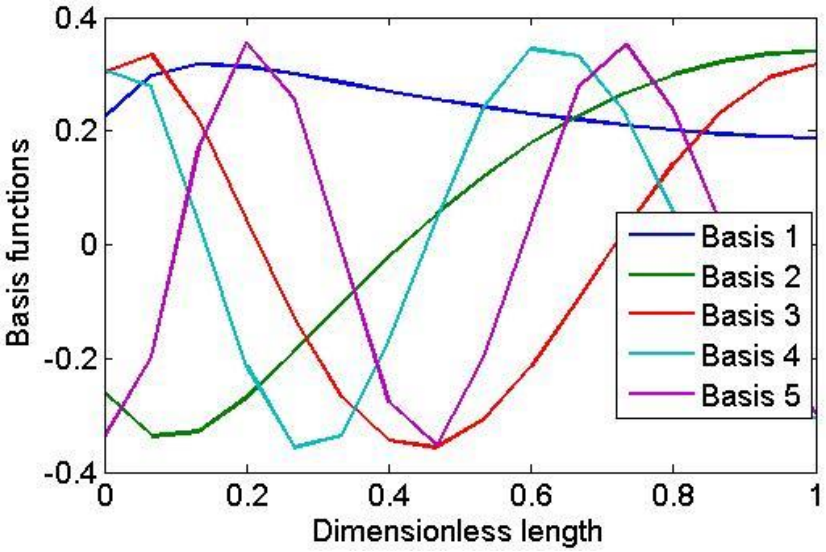


(b)

Figure 7. Comparison between (a) concentration (b) temperature predictions of full model and POD-ANN reduced model at the middle and output points for tubular reactor with $r = 0.5$.

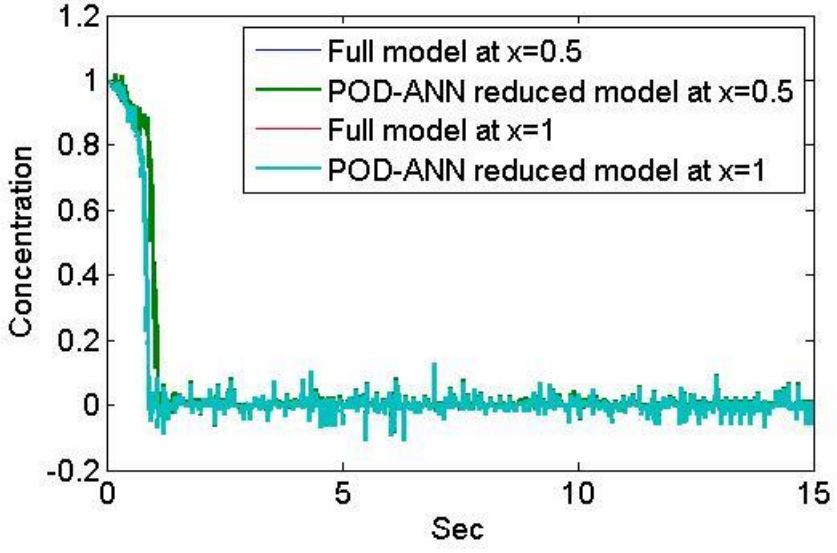


(a)

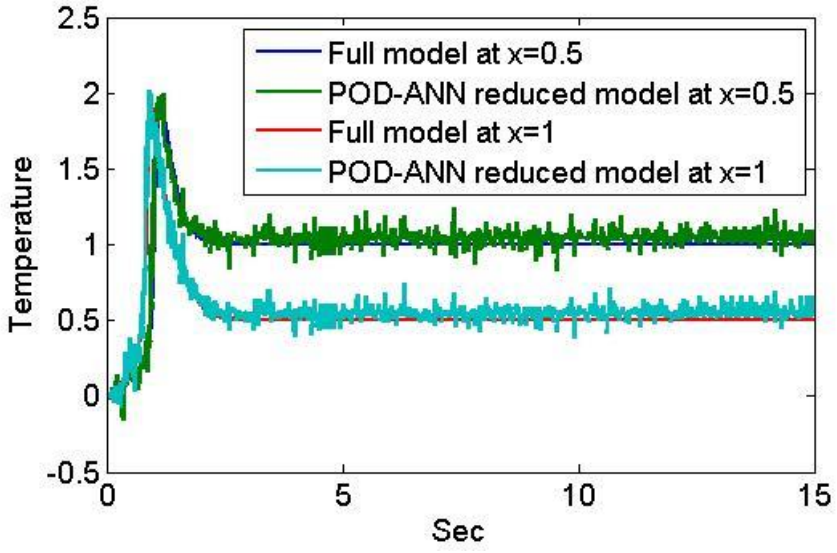


(b)

Figure 8. Global basis functions for (a) concentration (b) temperature from the sampling data of tubular reactor with $r = 0$.

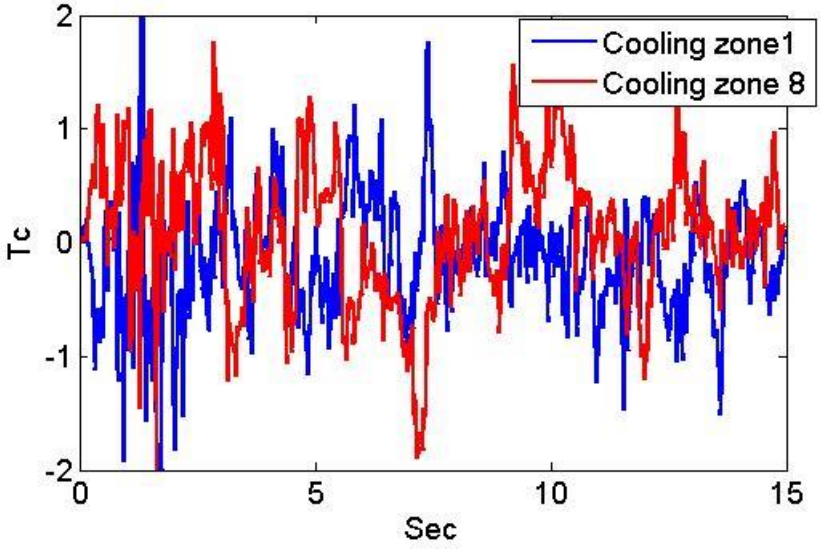


(a)

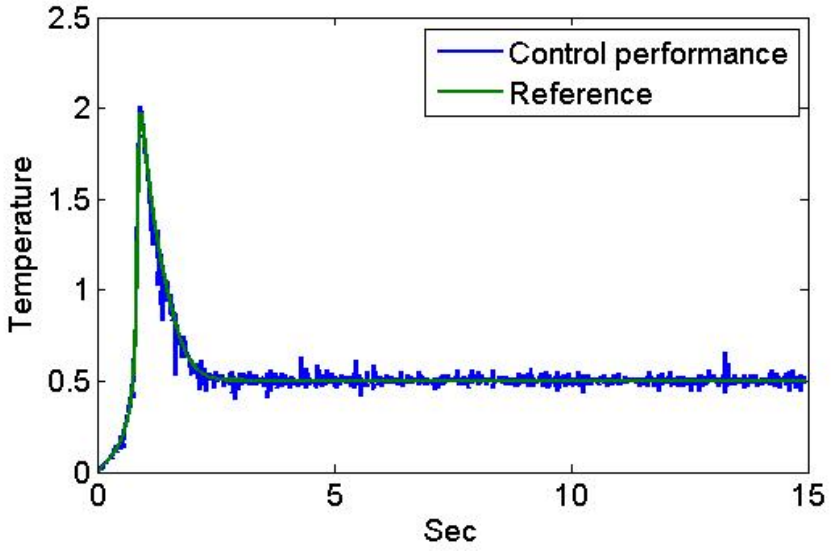


(b)

Figure 9. Comparison between (a) concentration (b) temperature predictions of full model and POD-ANN reduced model at the middle and output points for tubular reactor with $r = 0$.

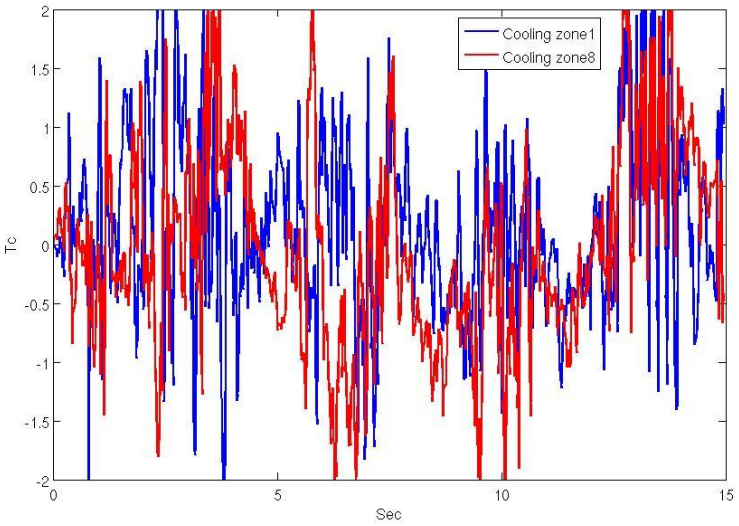


(a)

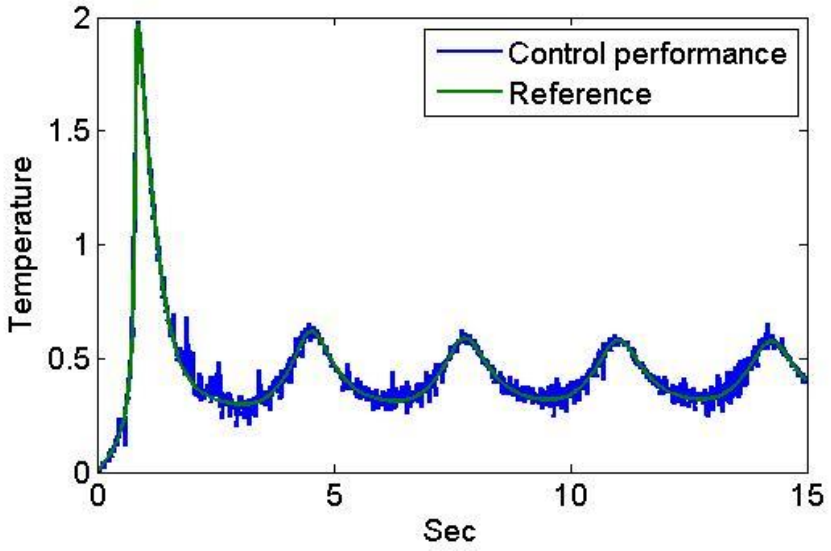


(b)

Figure 10. Nonlinear MPC results for first case study (a) control law for 1st and 8th actuators (b) control and reference profile.

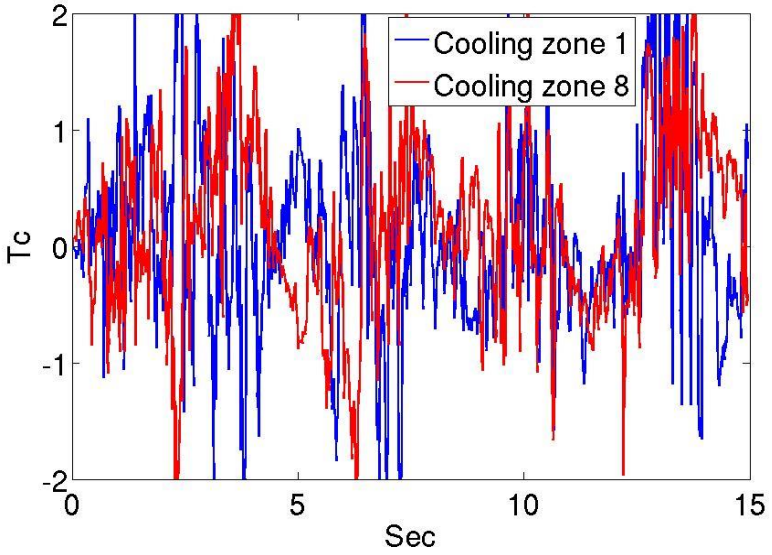


(a)

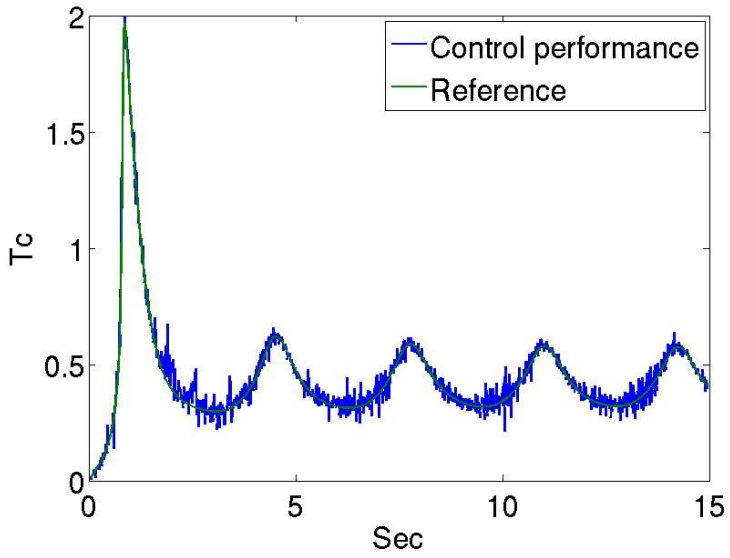


(b)

Figure 11. Nonlinear MPC results for second case study (a) control law for 1st and 8th actuators (b) control and reference profile.



(a)



(b)

Figure 12. Nonlinear MPC results for third case study (a) control law for 1st and 8th actuators (b) control and reference profile.

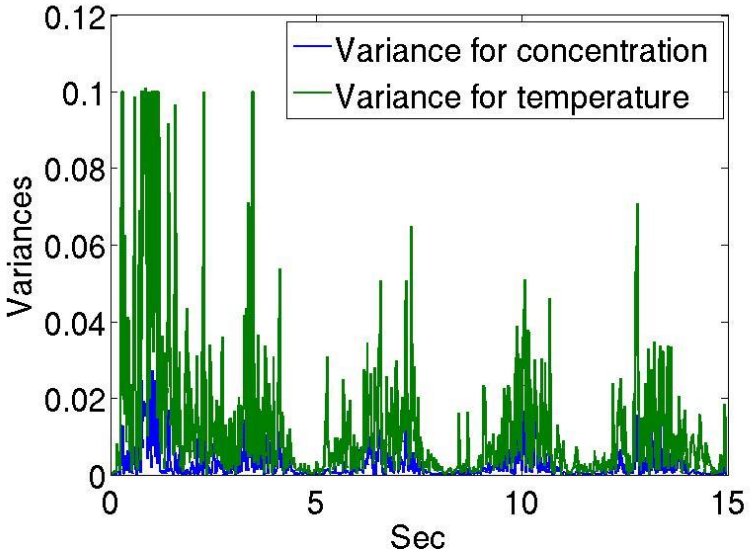


Figure 13. Results after implementing inequality constraint $Var(CA) \leq 0.1$ and $Var(TA) \leq 0.1$ for third case study.