Factorized f-Step Radial Basis Function Model for Model Predictive Control

D. K. Siong Tok*, Yiran Shi[&], Yantao Tian[&], Lei Gu[#$], Ding-Li Yu*

* Process Control Group, Liverpool John Moores University, Liverpool, U.K.

[&] Department of Control Theory and Engineering, College of Communication Engineering, Jilin University, Changchun, China.

School of Electronic Information, Changchun Architecture & Civil Engineering College, Changchun, China.

[$] Corresponding author: gulxx@sina.com

Abstract. This paper proposes a new factorized f-step radial basis function network (FS-RBF) model for model predictive control (MPC). The strategy is to develop a f-step predictor for nonlinear dynamic systems and implement it with a RBF network. In contrast to the popular NARX-RBF model, the developed FS-RBF model is capable of making a designated sequence of future output prediction without requiring the unknown future process measurements. Furthermore, the developed FS-RBF model is factorized into two parts, with one part including past plant input/output and the other part including the future input/output. When this model is used as the internal model in the MPC, the factorization enables an explicit objective function for the on-line optimization in the MPC. Thus, the computing load in solving the optimization problem is greatly reduced. The developed model is used in MPC and applied to a continuous-stirred tank reactor (CSTR). The simulation results are compared with that of MPCs with other two models. The comparison confirms that the developed model make more accurate prediction so that the MPC performance is better, it also uses much less computing time than the other two models based MPC.

Keywords: RBF network, f-step prediction model, model factorization, model predictive control, CSTR.

1. Introduction

Model predictive control (MPC) is a popular advanced control method which has many successful industry applications and its challenges is reported in [1]. Due to its popularity and promising long term success, the development and achievements of existing MPC methods have been consistently reviewed [1-4] since its introduction in 1978 [5]. One of the biggest benefits of MPC approach is it can be applied to multi-input, multi-output (MIMO) processes by taking the constraints into considerations. A MPC is a model-based controller which has an internal model that uses process measurements to predict the future behaviour of a process. A good prediction accuracy of the internal model is a fundamental requirement to achieve a good control performance. The use of linear and nonlinear internal models to characterise the MPC are called as linear MPC (LMPC) and nonlinear MPC (NMPC), respectively. The advantage of LMPC approaches is they are convex problem which simplify the optimization problem but many real application processes are nonlinear where a large mismatch between the linear model and real process is often encountered, eventually resulting in a poor control performance. Conversely, a nonlinear internal model provides a better match to the real processes but the challenge in NMPC is the optimization problems are non-convex and it has to be solved using nonlinear programming [6].

In modelling nonlinear dynamic processes, radial basis function (RBF) network models have been studied intensively due to its learning abilities and simple architecture [7]. The application of RBF networks to approximate nonlinear function was studied by Broom and Lowe [8]. The RBF networks have been investigated and developed in various form to model industrial systems [9, 10] and to predict outputs for nonlinear dynamic processes [11, 12]. A RBF network that based on a nonlinear autoregressive with exogenous input model (NARX) is renowned for its promising abilities in modelling nonlinear dynamic system [13]. Therefore, a NARX-RBF model is often used as an internal model in MPC (RBF-MPC) and the applications can be found in [14-17]. However, there are several drawbacks in RBF-MPC. The first drawback lies in the lack of efficiency in long range predictions due to its accumulated errors in each prediction step [18, 19]. Another disadvantage is that its future output predictions are made depending on future unknown process measurements. This means that the network cannot be factorized according to the past and future information, and the objective function is needed to be computed numerically. To overcome these issues, Bhartiya

and Whiteley [19] developed a factorable p-Step control model-based RBF network which produces efficient long predictions and applied it in MPC. Their results showed that the model performed better than the cascaded 1-step ahead prediction [19]. However, one of the major drawbacks in [19] is its unrealistic huge network structure which increases the model complexity and the computation time when solving an optimization problem in MPC.

To address these problems, this paper proposes a new factorized f-step model-based RBF network (FS-RBF). The objective of this work is twofold. The first objective is to introduce a f-step predictor and implement it with a RBF network. Secondly, the novelty in this work is the factorization of the FS-RBF network. As a result, an explicit objective function of the MPC is derived to reduce the computing load in solving the online optimization problem. The proposed model is evaluated in term of the modelling performance, model compactness, and computational requirement using a continuous-stirred tank reactor (CSTR) plant.

Both the prediction performance and the computing time for long range prediction of the developed model are evaluated by comparing the results with that of two existing models. By applying to the MPC of the CSTR process, the simulation results show that not only does the developed model suggested network outperform the other two models; more importantly, the developed model uses a more compact structure. Secondly, the control performance and computational requirement in solving the optimization problem of the proposed network is verified by comparing it with existing control approaches. The purposed network is proved to more computationally efficient while achieving a good control performance.

The paper is presented in the following structure. Section 2 describes the development of FS prediction model. Section 3 explains the training algorithm of FS-RBF model; it also presents the prediction performance of proposed FS-RBF model and the comparison with other two models. Section 4 describes the factorization of FS-RBF model. The evaluation of control performance and computational requirement of the proposed model-based MPC and comparison studies are presented in Section 5.

2. f-Step prediction model

A p-Step control (PSC) model was developed in [19] and implemented with RBF network. It was reported that the PSC model was capable of predicting future process outputs over a

prediction horizon without requiring future unknown process outputs. However, the flaw of the PSC model lies in an unrealistic massive network structure that is required to achieve a satisfactory prediction performance. This has incurred a big computational load and prevent the network model to be applied to the systems with fast dynamics. Moreover, there is a limitation in the PSC model in selecting the model parameter, i.e. the condition, $N_u \geq 2$ is required. To address these problems, an improved model, f-step (FS) prediction model, is developed in this paper and is implemented with RBF network. In this section, the derivation of the FS model is described.

In the applications of RBF networks [14, 20], a continuous-time nonlinear system is represented by a NARX model in (1).

$$y_k = f\left[y_{k-1}, \ldots, y_{k-Ny}, u_{k-1}, \ldots, u_{k-Nu}\right] + e_k \tag{1}$$

where $u \in \Re^m$ and $y \in \Re^n$ are the input and output vectors with N_u and N_y being the output and input orders respectively and e is the error. $f[*]$ is a vector-valued nonlinear function. There are two types of predictor structure as depicted in Fig.1, where A RBF network can be trained as a one-step-ahead (OSA) predictor or a multistep-ahead (MSA) predictor.

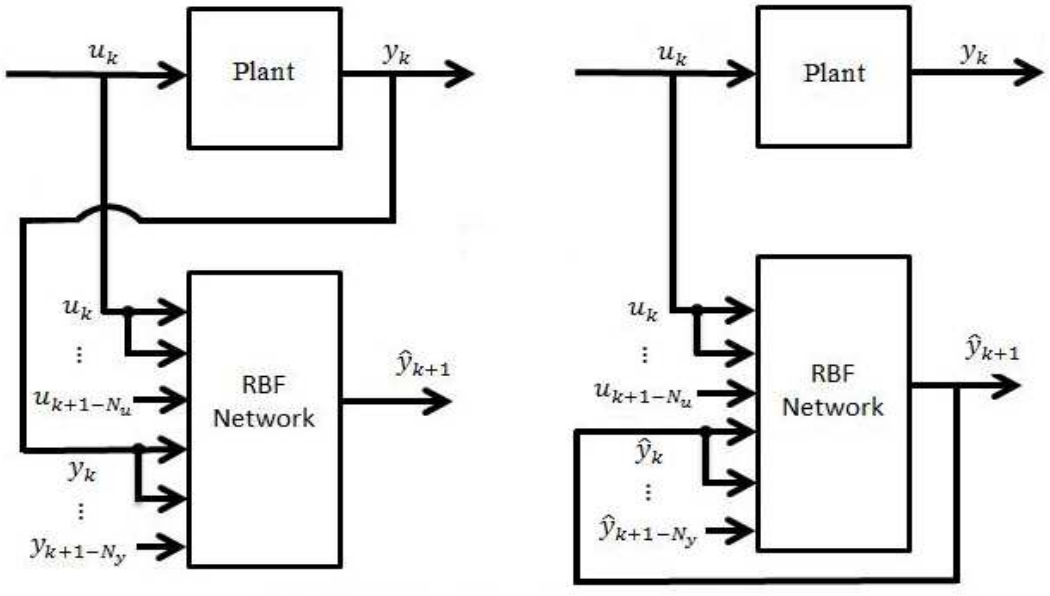

Fig.1 Block diagrams of OSA and MSA predictors.

From Fig.1, it is observed that an OSA predictor is trained using the plant inputs $u_k, \ldots, u_{k+1-N_u}$ and outputs $y_k, \ldots, y_{k+1-N_y}$ at sample time k to estimate the one-step-ahead

prediction $\hat{y}_{k+1}$. For the OSA predictor, the future predictions $\hat{y}_{k+1}, \ldots, \hat{y}_{k+f}$ over a prediction horizon H_p are described as

$$\begin{aligned}
\hat{y}_{k+1} &= f\left[y_k, \ldots, y_{k+1-N_y}, u_k, \ldots, u_{k+1-N_u}\right] \\
\hat{y}_{k+2} &= f\left[y_{k+1}, \ldots, y_{k+2-N_y}, u_{k+1}, \ldots, u_{k+2-N_u}\right] \\
&\vdots \\
\hat{y}_{k+f} &= f\left[y_{k+H_p-1}, \ldots, y_{k+H_p-N_y}, u_{k+H_p-1}, \ldots, u_{k+H_p-N_u}\right]
\end{aligned} \tag{2}$$

In contrast, in the MSA predictor the predicted outputs $\hat{y}_{k+1}, \ldots, \hat{y}_{k+H_p-1}$, instead of the plant outputs, are iteratively used in the future sample predictions $\hat{y}_{k+2}, \ldots, \hat{y}_{k+f}$ across a prediction horizon H_p as described as

$$\begin{aligned}
\hat{y}_{k+1} &= f\left[y_k, \ldots, y_{k+1-N_y}, u_k, \ldots, u_{k+1-N_u}\right] \\
\hat{y}_{k+2} &= f\left[\hat{y}_{k+1}, y_k, \ldots, y_{k+2-N_y}, u_{k+1}, \ldots, u_{k+2-N_u}\right] \\
&\vdots \\
\hat{y}_{k+f} &= f\left[\hat{y}_{k+H_p-1}, \ldots, \hat{y}_{k+H_p-N_y}, u_{k+H_p-1}, \ldots, u_{k+H_p-N_u}\right].
\end{aligned} \tag{3}$$

From (2), it is understood that the MSA predictor is less accurate compared with the OSA because the MSA uses the predicted outputs iteratively as its inputs, which introduces accumulated modelling error to the future predictions. However, though the MSA predictor is not as accurate as the OSA predictor, its ability to predict the multistep-ahead behaviour is essential for being used in the model predictive control [14, 21].

In order to improve the inaccurate prediction and at the same time to reserve the multistep-ahead prediction ability of the MSA predictor, a new FS prediction model is proposed here. The FS model is designed to make predictions over a prediction horizon H_p without using the future process outputs. It therefore makes use of the advantages of both the OSA and MSA predictors. The derivation of FS model commences with a NARX model in (1). To illustrate the concept of proposed FS model, consider an example with output orders $N_u = 2$ and input orders $N_y = 2$ to make predictions across a prediction horizon $H_p = 3$. Using this example, the NARX model in (1) can be expressed as

$$\hat{y}_k = F[y_{k-1}, y_{k-2}, u_{k-1}, u_{k-2}] \tag{4}$$

The outputs $y_{k-1}, \dots, y_{k-2}$ can be described in prediction forms of

$$\hat{y}_{k-1} = F[y_{k-2}, y_{k-3}, u_{k-2}, u_{k-3}]$$

$$\hat{y}_{k-2} = F[y_{k-3}, y_{k-4}, u_{k-3}, u_{k-4}] \tag{5}$$

Now, using (5) to approximate y_{k-1}, y_{k-2} in (4), it becomes

$$\begin{aligned}\hat{y}_k &= F[F[y_{k-2}, y_{k-3}, u_{k-2}, u_{k-3}], F[y_{k-3}, y_{k-4}, u_{k-3}, u_{k-4}], u_{k-1}, u_{k-2}] \\ &= F[F[F[y_{k-3}, y_{k-4}, u_{k-3}, u_{k-4}], y_{k-3}, u_{k-2}, u_{k-3}], \\ &\quad F[y_{k-3}, y_{k-4}, u_{k-3}, u_{k-4}], u_{k-1}, u_{k-2}]\end{aligned} \tag{6}$$

Using a function G to represent the composite function F in (6),

$$\hat{y}_{k|k-3} = G[y_{k-3}, y_{k-4}, u_{k-1}, u_{k-2}, u_{k-3}, u_{k-4}] \tag{7}$$

Using (7) for predictions over $H_p = 3$, they are as follows,

$$\hat{y}_{k+1|k-2} = G[y_{k-2}, y_{k-3}, u_k, u_{k-1}, u_{k-2}, u_{k-3}]$$

$$\hat{y}_{k+2|k-1} = G[y_{k-1}, y_{k-2}, u_{k+1}, u_k, u_{k-1}, u_{k-2}]$$

$$\hat{y}_{k+3|k} = G[y_k, y_{k-1}, u_{k+2}, u_{k+1}, u_k, u_{k-1}] \tag{8}$$

Equation (7) can be extended to a general form,

$$\hat{y}_{k|k-f} = G\left[y_{k-f}, \dots, y_{k-f+1-N_y}, u_{k-1}, .., u_{k-f+1-N_u}\right] \tag{9}$$

or alternatively,

$$\hat{y}_{k+f|k} = G\left[y_k, \dots, y_{k+1-N_y}, u_{k-1+f}, .., u_{k+1-N_u}\right]. \tag{10}$$

From (10), it can be observed that the f-step prediction $\hat{y}_{k+f|k}$ requires process outputs up to k^{th} sample time, which are all available at current sample period k. In other words, only process output measurements up to k^{th} sample time are required for the prediction of outputs up to $(k+f)^{\text{th}}$ sample time. This means that the dependency on the future predicted outputs $y_{k+1}, \dots, y_{k+f-1}$ over a prediction horizon H_p is eliminated. Therefore, it improves the prediction accuracy. Also, the future inputs from current sample instant k to the prediction

sample instants k+ f are required, and this can just be used as the variables to be optimized in the MPC.

3. Modelling A CSTR

To demonstrate the effectiveness of the developed FS-RBF model for multistep ahead prediction, the FS-RBF model is developed for a CSTR and compared with other two frequently used RBF models.

3.1 FS-RBF Training Algorithm

The structure of the developed FS-RBF model is the same as that of normal RBF network with only difference being different input vectors. Therefore, except for forming input vector by different input/output sample values, the training of the network including find centre vector and width for the Gaussian functions in each hidden layer node and training of the connected weights between the hidden layer and the output layer, are all the same. As in [17] and [23], the K-means clustering algorithm and p-nearest centres algorithm are used in this work to determine the centres and width based on the training data set. The recursive Least Squares algorithm (RLS) is used to optimize the connection weights. The network input vector x_k for the FS-RBF model in (9) is

$$x_k = \left[y_{k-f} \ \ldots \ y_{k-f+1-N_y} \ \ u_{k-1} \ \ldots \ u_{k-f+1-N_u} \right] \tag{11}$$

where u and y are system input and output, respectively. A Gaussian function is used as the activation function and the hidden layer output $\phi_i(k)$ is given as

$$\phi_i(k) = \exp\left(-\frac{\|x_k - c_i\|^2}{\sigma_i^2} \right), i = 1, \ldots, n_h \tag{12}$$

where n_h is the number of hidden neurons and $c_i \in \Re^n$ is the ith center. σ_i represents the ith width of the Gaussian function. The network output is

$$\hat{y}(k) = W^T(k)\phi(k) \tag{13}$$

where $W_k \in \Re^{n_h \times p}$ is the weighting matrix connecting hidden layer nodes and network outputs, and p is the number of outputs. The RLS training algorithm [22] is used to train the developed FS-RBF model.

$$L(t) = \frac{P(t-1)\varphi(t)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)}, \tag{14}$$

$$\widehat{w}(t) = \widehat{w}(t-1) + L(t)[y(t) - \varphi^T(t)\widehat{w}(t-1)], \tag{15}$$

$$P(t) = \frac{1}{\lambda(t)}\left[P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)}\right] \tag{16}$$

where $\widehat{w}(t)$ and $\varphi(t)$ represent the network weights and activation function outputs at time, t. $P(t)$ and $L(t)$ are middle terms. $\lambda(t)$ is a forgetting factor which is in the range of (0,1).

3.3 CSTR Dynamics and Data Acquisition

A CSTR in [23, 24] is selected as the example process for evaluation of modelling and control of the developed FS-RBF model because this process has nonlinear dynamics not only in static gain and also in dynamic parameters. It is therefore is often employed as a benchmark for nonlinear control evaluation. The plant is described by the following nonlinear differential equations,

$$\dot{C_a}(t) = \frac{q}{v}\big(C_{a0} - C_a(t)\big) - k_0 C_a(t) e^{-\frac{E}{RT(t)}} \tag{19}$$

$$\dot{T}(t) = \frac{q}{v}\big(T_0 - T(t)\big) + k_1 C_a(t) e^{-\frac{E}{RT(t)}} + k_2 q_c(t)\left(1 - e^{-\frac{k_3}{q_c(t)}}\right)\big(T_{c0} - T(t)\big) \tag{20}$$

The reactor is used to mix two chemicals to produce a product compound A. A type of exothermic reaction takes place in the reactor, which slows down the reaction resulting with nonlinear dynamics. The objective of the control system is to control the concentration of product compound A, $C_a(t)$ with temperature $T(t)$ by manipulating the flow rate of coolant, $q_c(t)$. Therefore, the input is the flow rate of coolant q_c and the output is the concentration of compound A, $C_a(t)$. For the reactor the nominal values of the physical parameters are listed

in Appendix A. The nonlinearity of the plant is illustrated in Fig. 2, where the step response at different operating points is displayed. It can be observed that the dynamics are more underdamped when the concentration is higher.

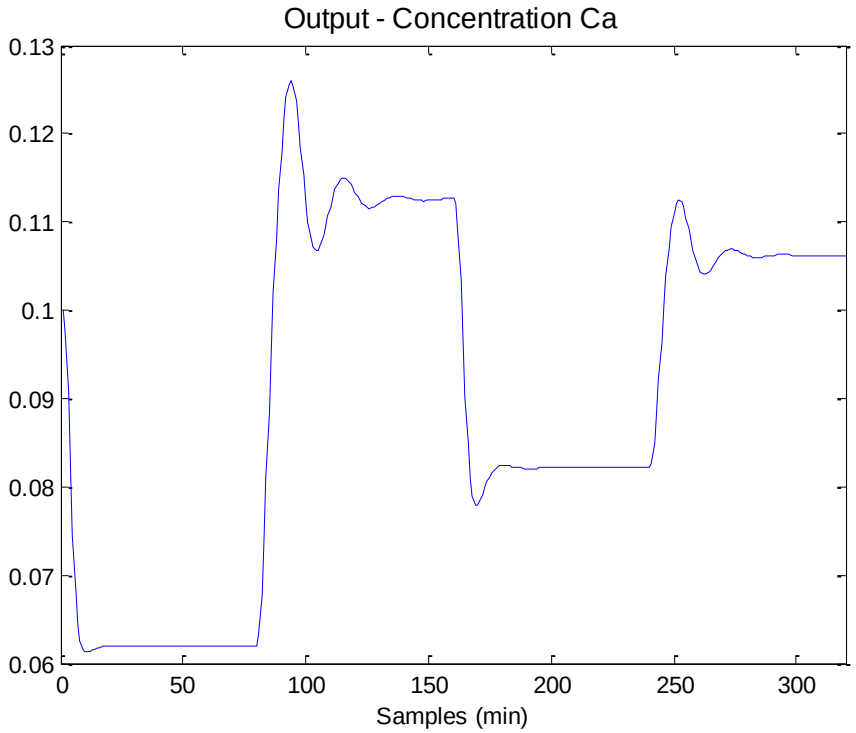

Fig. 2 Step response of the CSTR plant.

The sampling period is chosen to be 0.1min. A set of persistently exciting input signal is designed as shown in Fig.3 to generate a set of 1400 output data samples. The excitation input signal consists of a random amplitude sequence (RAS) of large amplitude superimposed on it with a RAS of small amplitude, to capture the dynamic behaviours of the plant at all frequencies and at all different levels of operating space. The collected input-output data points are halved into two sets - first 700 data samples are used as training data and the remainders are used as validation data. The orders of all variables in the network input vector are selected according to the orders of them in the reactor dynamic equations and are carefully tuned to give the best generalization result. In the meantime, the numbers of centres are decided considering a trade-off between the network size and the prediction error.

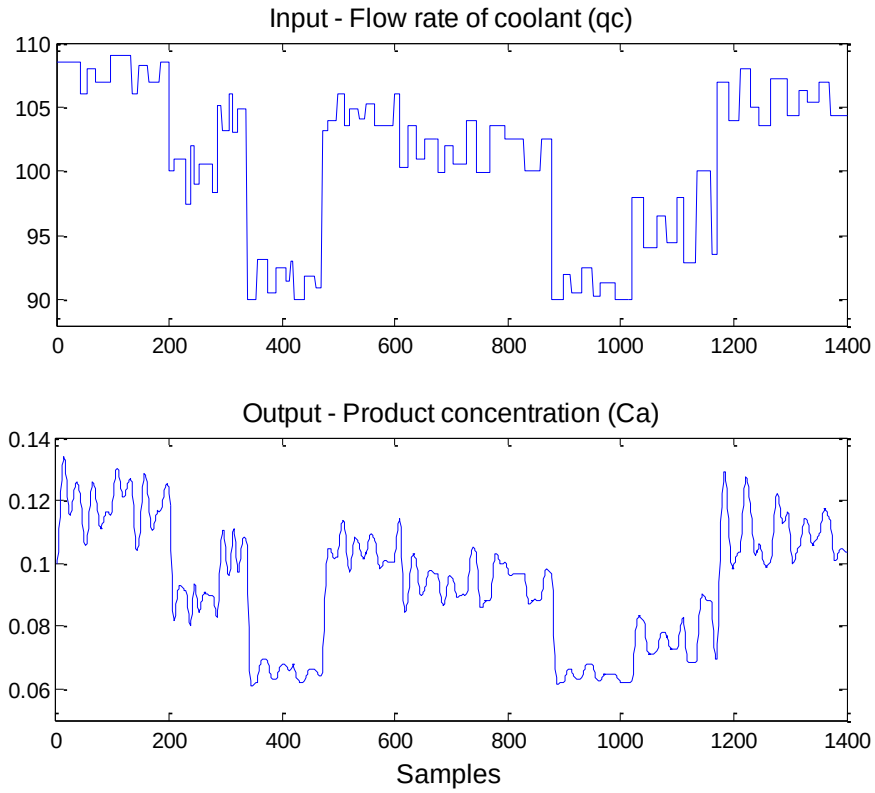

Fig.3 The collected data samples

3.2 f-Step RBF Modelling and Comparison

In this section, the prediction performance and model compactness of the FS-RBF model are evaluated with the training and validation date sets, and compared with the NARX-RBF model and the PSC-RBF model. The NARX-RBF model is of the structure shown in right figure in Fig.1 and is able to predict for multistep-ahead with the predicted output as model input. The PSC-RBF model can predict for multi-step ahead outputs [19]. To effectively evaluate the ability of accurate long range prediction, the proposed FS-RBF model and PSC-RBF model are tested with three different prediction horizons $H_p = 5, 10, 20$, respectively. Both training and validation data sets are scaled to [0 1] to minimize the error caused by the difference between ranges of different variables.

$$u_s = \frac{u - \min(u)}{\max(u) - \min(u)} \qquad y_s = \frac{y - \min(y)}{\max(y) - \min(y)} \tag{18}$$

where u and y are input and output in raw data and u_s and y_s are the scaled data; $\min(.)$ and $\min(.)$ are the minimum values of input and output, respectively. The scaled output predictions are then scaled back after the model is used.

All the three network models are trained with the RLS algorithm in Section 3.2. For a fair comparison, the parameters in all networks are carefully tuned. The K-means clustering algorithm is used to find the position of centres and the radius of the Gaussian functions is calculated using P-nearest neighbour algorithm. The mean absolute error (MAE) is used to measure the prediction errors,

$$MAE = \frac{1}{N}\sum_{j=1}^{N}|y_i - \hat{y}_i| \tag{17}$$

where N is the number of data samples. After training, the validation data set is applied to the three types of model and model prediction results are recorded in Table 1.

Table 1 Performance comparison of different RBF-based models

Models	NARX-RBF	PSC-RBF			FS-RBF		
H_p	5	5	10	20	5	10	20
N_u	3	3	2	2	3	2	2
N_y	3	3	2	2	3	2	2
nh	43	150	216	362	44	36	44
Training data MAE	0.0154	0.0086	0.0157	0.0156	0.0080	0.0104	0.0116
Validation data MAE	0.0121	0.0070	0.0125	0.0176	0.0056	0.0075	0.0081

From the prediction results in Table 1 it is evidence that the following four points. First, Both FS-RBF and PSC-RBF models have more accurate predictions than the NARX-RBF for 5-step-ahead prediction. This is because the former two models used measured input/output data rather than the latter model used predicted output as model input to bring in accumulated

error. Secondly, For long-range, Hp=10 and Hp=20, the FS-RBF predictions are more accurate than that by the PSC-RBF model, especially for the validation data. Thirdly, in addition to the more accurate prediction, the FS-RBF model has a much smaller size than the corresponding PSC-RBF model for all the three different prediction horizons. This point is important as the bigger size of the model will lead to a much increased computing load in MPC optimization. Fourthly, the FS-RBF model prediction has only a slight degradation with the increase of the prediction horizon compared with the PSC-RBF model. This point is very important when the model is used as the inter model in the nonlinear MPC.

As can be seen, the PSC-RBF model has the largest network structures as listed in Table 1. With the prediction horizon increasing, the PSC-RBF model needs to use unrealistically large network structures to achieve satisfactory predictions. This prevents the PSC-RBF model to be used in MPC for systems with slow dynamics only, for example a temperature control system. Unlike the PSC model, the proposed FS model is able to maintain its structure size while producing satisfactory predictions for long range.

4. Factorization of FS RBF Network for MPC

The purpose to factorize FS-RBF network is to derive an explicit objection function for MPC to reduce the computation burden in solving the online optimization problem. For a NARX model, it is impossible to factorize because its future predictions are dependent on unknown future process measurement. Although PSC model manages to offer the factorization form, an unrealistic large network structure is required to obtain a satisfied modelling performance as shown in Section 3.3. As the FS-RBF model is different from the PS model, the derivation of factorization is completely different. Due to the factorability of exponential function, the prediction output of a RBF network can be rewritten as

$$\hat{y}_{k+i} = \sum_{j=1}^{n_h} w_j \exp(Past + Future)$$

$$= \begin{bmatrix} w_1 \exp(Past) \\ \vdots \\ w_{n_h} \exp(Past) \end{bmatrix}^T \begin{bmatrix} \exp(Future) \\ \vdots \\ \exp(Future) \end{bmatrix}$$

$$= \hat{y}_p^T \hat{y}_f \tag{21}$$

with $\hat{y}_p^T$ and $\hat{y}_f$ denote the past and future matrices, respectively.

Based on FS model in (9), $u_{k-1+f}, .., u_{k+1-N_u-1}$ and u_{k+1-N_u} can be called as past inputs and future inputs, respectively. In MPC, the future inputs u_{k+1-N_u} are the variables to be optimized. The objective function of MPC is described by

$$V_k = \sum_{i=H_w}^{H_p} \left\| \hat{y}_{k+i|k} - r_{k+i|k} \right\|_{Q_i}^2 + \sum_{i=0}^{H_u-1} \left\| \Delta \hat{u}_{k+i|k} \right\|_{R_i}^2 \tag{22}$$

where R_i and Q_i are the penalty on the changes of inputs and the errors between output and desired set point $r_{k+i|k}$, respectively. $\Delta \hat{u}_{k+i|k}$ represents the changes in input. H_w and H_u represent the window parameter and control horizon, respectively. From (22), it can be noticed that the objective function penalizes the changes in input $\Delta \hat{u}_{k+i|k}$. Therefore, it is necessary to re-formulate FS model in (9). The changes in input are described by

$$\begin{aligned}
\Delta \hat{u}_k &= \hat{u}_k - u_{k-1} \\
\hat{u}_k &= \Delta \hat{u}_k + u_{k-1} \\
\hat{u}_{k+1} &= \Delta \hat{u}_{k+1} + \hat{u}_k \\
&= \Delta \hat{u}_{k+1} + (\Delta \hat{u}_k + u_{k-1}) \\
&\vdots \\
\hat{u}_{k+f-1} &= \Delta \hat{u}_{k+f-1} + \Delta \hat{u}_{k+f-2} + \cdots + \Delta \hat{u}_{k+1} + \Delta \hat{u}_k + u_{k-1}
\end{aligned} \tag{23}$$

with $\hat{u}$ denotes the future prediction. Then, substitute (23) into (10), the FS model becomes

$$\begin{aligned}
\hat{y}_{k+1} &= G\left(y_{k+1-f}, \dots, y_{k+2-f-N_y}, u_{k-2-f-N_u}, \dots, u_{k-1}, (u_{k-1} + \Delta u_k)\right) \\
&\vdots \\
\hat{y}_{k+f} &= G\Big(y_k, \dots, y_{k+1-N_y}, u_{k-1}, u_{k-1} \\
&\qquad + \Delta u_k, \dots, \left(\Delta \hat{u}_{k+f-1} + \Delta \hat{u}_{k+f-2} + \cdots + \Delta \hat{u}_{k+1} + \Delta \hat{u}_k \right. \\
&\qquad \left. + u_{k-1}\right)\Big)
\end{aligned} \tag{24}$$

Note that the order of inputs is rearranged for factorization purpose. To illustrate the factorized FS-RBF model, an example with $H_p = 3$, $N_y = 2$, $N_u = 2$, and $n_h = 3$ is considered. The f-step predictions within prediction horizon in (10) are

$$\begin{aligned}\hat{y}_{5|1} &= G(y_2, y_1, u_1, u_2, u_3, u_4)\\ \hat{y}_{6|2} &= G(y_3, y_2, u_2, u_3, u_4, u_5)\\ \hat{y}_{7|3} &= G(y_4, y_3, u_3, u_4, u_5, u_6)\end{aligned} \tag{25}$$

and the changes in inputs are rewritten according to (23)

$$\begin{aligned}\hat{u}_4 &= \Delta\hat{u}_4 + u_3\\ \hat{u}_5 &= \Delta\hat{u}_5 + \hat{u}_4 = \Delta\hat{u}_5 + (\Delta\hat{u}_4 + u_3)\\ \hat{u}_6 &= \Delta\hat{u}_6 + \hat{u}_5 = \Delta\hat{u}_6 + \Delta\hat{u}_5 + (\Delta\hat{u}_4 + u_3)\end{aligned} \tag{26}$$

By substituting (26) into (25), the predictions become

$$\begin{aligned}\hat{y}_{5|1} &= G\big(y_2, y_1, u_1, u_2, u_3, (\Delta\hat{u}_4 + u_3)\big)\\ \hat{y}_{6|2} &= G\big(y_3, y_2, u_2, u_3, (\Delta\hat{u}_4 + u_3), (\Delta\hat{u}_5 + \Delta\hat{u}_4 + u_3)\big)\\ \hat{y}_{7/3} &= G\big(y_4, y_3, u_3, (\Delta\hat{u}_4 + u_3), (\Delta\hat{u}_5 + \Delta\hat{u}_4 + u_3), (\Delta\hat{u}_6 + \Delta\hat{u}_5 + \Delta\hat{u}_4 \\ &\qquad + u_3)\big)\end{aligned} \tag{27}$$

Using (21), the output predictions can be factorized as

$$\hat{y}_{5/1} = \begin{bmatrix} w_1 \exp\left(\frac{\left(c_{1,1}-y_2\right)^2+\left(c_{1,2}-y_1\right)^2+\left(c_{1,3}-u_1\right)^2+\left(c_{1,4}-u_2\right)^2+\left(c_{1,5}-u_3\right)^2}{\sigma^2}\right) \\ w_2 \exp\left(\frac{\left(c_{2,1}-y_2\right)^2+\left(c_{2,2}-y_1\right)^2+\left(c_{2,3}-u_1\right)^2+\left(c_{2,4}-u_2\right)^2+\left(c_{2,5}-u_3\right)^2}{\sigma^2}\right) \\ w_3 \exp\left(\frac{\left(c_{3,1}-y_2\right)^2+\left(c_{3,2}-y_1\right)^2+\left(c_{3,3}-u_1\right)^2+\left(c_{3,4}-u_2\right)^2+\left(c_{3,5}-u_3\right)^2}{\sigma^2}\right) \end{bmatrix}^T \times \begin{bmatrix} \exp\left(\frac{\left(c_{1,6}-u_3-\Delta\hat{u}_4\right)^2}{\sigma^2}\right) \\ \exp\left(\frac{\left(c_{2,6}-u_3-\Delta\hat{u}_4\right)^2}{\sigma^2}\right) \\ \exp\left(\frac{\left(c_{3,6}-u_3-\Delta\hat{u}_4\right)^2}{\sigma^2}\right) \end{bmatrix} \quad (28)$$

$$\hat{y}_{6/2} = \begin{bmatrix} w_1 \exp\left(\frac{\left(c_{1,1}-y_3\right)^2+\left(c_{1,2}-y_2\right)^2+\left(c_{1,3}-u_2\right)^2+\left(c_{1,4}-u_3\right)^2}{\sigma^2}\right) \\ w_2 \exp\left(\frac{\left(c_{2,1}-y_3\right)^2+\left(c_{2,2}-y_2\right)^2+\left(c_{2,3}-u_2\right)^2+\left(c_{2,4}-u_3\right)^2}{\sigma^2}\right) \\ w_3 \exp\left(\frac{\left(c_{3,1}-y_3\right)^2+\left(c_{3,2}-y_2\right)^2+\left(c_{3,3}-u_2\right)^2+\left(c_{3,4}-u_3\right)^2}{\sigma^2}\right) \end{bmatrix}^T \times \begin{bmatrix} \exp\left(\frac{\left(c_{1,5}-u_3-\Delta\hat{u}_4\right)^2+\left(c_{1,6}-u_3-\Delta\hat{u}_4-\Delta\hat{u}_5\right)^2}{\sigma^2}\right) \\ \exp\left(\frac{\left(c_{2,5}-u_3-\Delta\hat{u}_4\right)^2+\left(c_{2,6}-u_3-\Delta\hat{u}_4-\Delta\hat{u}_5\right)^2}{\sigma^2}\right) \\ \exp\left(\frac{\left(c_{3,5}-u_3-\Delta\hat{u}_4\right)^2+\left(c_{3,6}-u_3-\Delta\hat{u}_4-\Delta\hat{u}_5\right)^2}{\sigma^2}\right) \end{bmatrix} \quad (29)$$

$$\hat{y}_{7/3}$$

$$= \begin{bmatrix} w_1 \exp\left(\dfrac{\left(c_{1,1}-y_4\right)^2+\left(c_{1,2}-y_3\right)^2+\left(c_{1,3}-u_3\right)^2}{\sigma^2}\right) \\ w_2 \exp\left(\dfrac{\left(c_{2,1}-y_4\right)^2+\left(c_{2,2}-y_3\right)^2+\left(c_{2,3}-u_3\right)^2}{\sigma^2}\right) \\ w_3 \exp\left(\dfrac{\left(c_{3,1}-y_4\right)^2+\left(c_{3,2}-y_3\right)^2+\left(c_{3,3}-u_3\right)^2}{\sigma^2}\right) \end{bmatrix}^T \tag{30}$$

$$\times \begin{bmatrix} \exp\left(\dfrac{\left(c_{1,4}-u_3-\Delta\hat{u}_4\right)^2+\left(c_{1,5}-u_3-\Delta\hat{u}_4-\Delta\hat{u}_5\right)^2+\left(c_{1,6}-u_3-\Delta\hat{u}_4-\Delta\hat{u}_5-\Delta\hat{u}_6\right)^2}{\sigma^2}\right) \\ \exp\left(\dfrac{\left(c_{2,4}-u_3-\Delta\hat{u}_4\right)^2+\left(c_{2,5}-u_3-\Delta\hat{u}_4-\Delta\hat{u}_5\right)^2+\left(c_{2,6}-u_3-\Delta\hat{u}_4-\Delta\hat{u}_5-\Delta\hat{u}_6\right)^2}{\sigma^2}\right) \\ \exp\left(\dfrac{\left(c_{3,4}-u_3-\Delta\hat{u}_4\right)^2+\left(c_{3,5}-u_3-\Delta\hat{u}_4-\Delta\hat{u}_5\right)^2+\left(c_{3,6}-u_3-\Delta\hat{u}_4-\Delta\hat{u}_5-\Delta\hat{u}_6\right)^2}{\sigma^2}\right) \end{bmatrix}$$

with $c_{i,j}$ representing the j^{th} element of the centre vector c_i. This example shows that the output predictions of FS-RBF model is factorized into the past and the future matrices form as in (21). On the right hand side in (28-30), the first factor represents $\hat{y}_p^T$ which consists of all known process measurement, whilst the second factor represents $\hat{y}_f$ which consists of the changes in inputs to be optimized in MPC. The general form of the factorized FS-RBF model is described by

$$\begin{bmatrix} \hat{y}_{k+1} \\ \vdots \\ \hat{y}_{k+f} \end{bmatrix} = \begin{bmatrix} \hat{y}_{past,k+1}^T \, \hat{y}_{future,k+1} \\ \vdots \\ \hat{y}_{past,k+f}^T \, \hat{y}_{future,k+f} \end{bmatrix} \tag{31}$$

$$\hat{y}_{past,k+1} = \begin{bmatrix} w_1 exp\left\{-\dfrac{1}{\sigma^2}\begin{bmatrix} \left(c_{1,1}-y_{k+1-f}\right)^2+\cdots+\left(c_{1,N_y}-y_{k+2-f-N_y}\right)^2+ \\ \left(c_{1,N_y+1}-u_{k+2-f-N_u}\right)^2+\cdots+\left(c_{1,N_y+N_u+f-2}-u_{k-1}\right)^2 \end{bmatrix}\right\} \\ \vdots \\ w_{n_h} exp\left\{-\dfrac{1}{\sigma^2}\begin{bmatrix} \left(c_{n_h,1}-y_{k+1-f}\right)^2+\cdots+\left(c_{n_h,N_y}-y_{k+2-f-N_y}\right)^2 \\ +\left(c_{n_h,N_y+1}-u_{k+2-f-N_u}\right)^2+\ldots+\left(c_{n_h,N_y+N_u+f-2}-u_{k-1}\right)^2 \end{bmatrix}\right\} \end{bmatrix}$$

$$\vdots$$

$$\hat{y}_{past,k+f} = \begin{bmatrix} w_1 exp\left\{-\frac{1}{\sigma^2}\left[\left(c_{n_h,1} - y_{k+1-p}\right)^2 + \cdots + \left(c_{n_h,N_y} - y_{k+2-p-N_y}\right)^2 + \left(c_{n_h,N_y+1} - u_{k-1}\right)^2\right]\right\} \\ \vdots \\ w_{n_h} exp\left\{-\frac{1}{\sigma^2}\left[\left(c_{n_h,1} - y_{k+1-p}\right)^2 + \cdots + \left(c_{n_h,N_y} - y_{k+2-p-N_y}\right)^2 + \left(c_{n_h,N_y+1} - u_{k-1}\right)^2\right]\right\} \end{bmatrix} \quad (32)$$

$$\hat{y}_{future,k+1} = \begin{bmatrix} exp\left\{-\frac{1}{\sigma^2}\left[\left(c_{1,N_y+N_u+f-1} - u_{k-1} - \Delta\hat{u}_k\right)^2\right]\right\} \\ \vdots \\ exp\left\{-\frac{1}{\sigma^2}\left[\left(c_{n_h,N_y+N_u+f-1} - u_{k-1} - \Delta\hat{u}_k\right)^2\right]\right\} \end{bmatrix}$$

$$\vdots$$

$$\hat{y}_{future,k+H_u} = \begin{bmatrix} exp\left\{-\frac{1}{\sigma^2}\left[\begin{matrix}\left(c_{1,N_y+2} - u_{k-1} - \Delta\hat{u}_k\right)^2 + \cdots + \\ \left(c_{1,k+N_y} - \Delta\hat{u}_{k+H_u-1} - \Delta\hat{u}_{k+H_u-2} - \cdots - \Delta\hat{u}_{k+1} - \Delta\hat{u}_k - u_{k-1}\right)\end{matrix}^2\right]\right\} \\ \vdots \\ exp\left\{-\frac{1}{\sigma^2}\left[\begin{matrix}\left(c_{n_h,N_y+2} - u_{k-1} - \Delta\hat{u}_k\right)^2 + \cdots + \\ \left(c_{n_h,k+N_y} - \Delta\hat{u}_{k+H_u-1} - \Delta\hat{u}_{k+H_u-2} - \cdots - \Delta\hat{u}_{k+1} - \Delta\hat{u}_k - u_{k-1}\right)\end{matrix}^2\right]\right\} \end{bmatrix}$$

$$\vdots$$

$$\hat{y}_{future,k+f} = \begin{bmatrix} exp\left\{-\frac{1}{\sigma^2}\left[\begin{matrix}\left(c_{1,N_y+2} - u_{k-1} - \Delta\hat{u}_k\right)^2 + \cdots + \\ \left(c_{1,k+N_y} - \Delta\hat{u}_{k+H_u-1} - \Delta\hat{u}_{k+H_u-2} - \cdots - \Delta\hat{u}_{k+1} - \Delta\hat{u}_k - u_{k-1}\right)\end{matrix}^2\right]\right\} \\ \vdots \\ exp\left\{-\frac{1}{\sigma^2}\left[\begin{matrix}\left(c_{n_h,N_y+2} - u_{k-1} - \Delta\hat{u}_k\right)^2 + \cdots + \\ \left(c_{n_h,k+N_y} - \Delta\hat{u}_{k+H_u-1} - \Delta\hat{u}_{k+H_u-2} - \cdots - \Delta\hat{u}_{k+1} - \Delta\hat{u}_k - u_{k-1}\right)\end{matrix}^2\right]\right\} \end{bmatrix} \quad (33)$$

Notice all control inputs remain constant from H_{u+1} to f in MPC. With the factorization of FS-RBF model, the objective function in MPC in (22) becomes explicit in the changes of inputs,

$$V_k = \sum_{i=H_w}^{H_p} \left\| \hat{y}_{past,\ k+i}^T\, \hat{y}_{future,k+i}(\Delta\hat{u}_{k+i}) - r_{k+i|k} \right\|_{Q_i}^2 + \sum_{i=0}^{H_u-1} \left\| \Delta\hat{u}_{k+i|k} \right\|_{R_i}^2 \quad (34)$$

From (34), it is understood that unlike the NARX-RBF model, the factorized FS-RBF model provides an analytical form of objective function for MPC. $\hat{y}_{past,k+1}^{T}, \ldots, \hat{y}_{past,k+f}^{T}$ in (32) are only computed once at each sample time, which reduces the computational requirement when solving the optimization problem.

5. MPC of The CSTR

To evaluate the effectiveness of the proposed FS-RBF model-based MPC (MPC-FS), the CSTR plant in Section 3.3 is considered. The CSTR plant possesses different nonlinear characteristics in different level of product concentration. Thus, the control performance is assessed in three levels of product concentration: high (0.11mol/l), middle (0.09mol/l), and low (0.065mol/l). Step changes of the set-point in two different levels are used to further test the dynamic response of the control approach. The upper and lower bound constraints for the control variable, coolant flow rate q_c , are set to 110 and 90, respectively. There is no constraint imposed on the outputs. The MPC approaches based on both the NARX-RBF model and the PSC-RBF model for the CSTR are developed. The control system performances and the computing loads with the two different models are then compared with the MPC based on the FS-RBF model in the optimization algorithm.

5.1 Control Performance

To provide a fair comparison, the prediction horizon H_p and control horizon H_u for all control approaches are set to 10 and 5, respectively. It is realized that in MPC, the control performance is strongly based on the prediction accuracy of the internal model that represents the real process. Thus, based on the modeling performance, the model orders and number of centers are selected from Table 1 in Section 3.3. In practice, the parameters of three control approaches are well tuned and recorded in Table 2.

Table 2 Control Parameters

Control Strategy	Parameters
MPC-NARX	$R_i = 1.1, Q_i = 0.9$
MPC-FS	$R_i = 1.2, Q_i = 0.75$

MPC-PSC	$R_i = 1.15, Q_i = 0.8$

In the first control experiment a small step change in the middle level of concentration $C_a = 0.09mol/l$ as shown in Fig.4 is used as the set-point. The set point is a rectangular waveform where the product concentration level decreases from 0.1mol/l to 0.09mol/l, and then increases from 0.09mol/l to 0.1mol/l again. The control performances of all compared control strategies are shown in Fig. 4 and the MAE are recorded in Table 3. The MPC-NARX has the largest overshoot among three control strategies, for both drop and rise of the set-point. Meanwhile, the MPC-PSC has the largest steady-state error in both drop and rise scenarios. Conversely, the MPC-FS clearly has the least steady-state error. From the MAE presented in Table 3, the overall control performance of MPC-FS is the best among the group of three models.

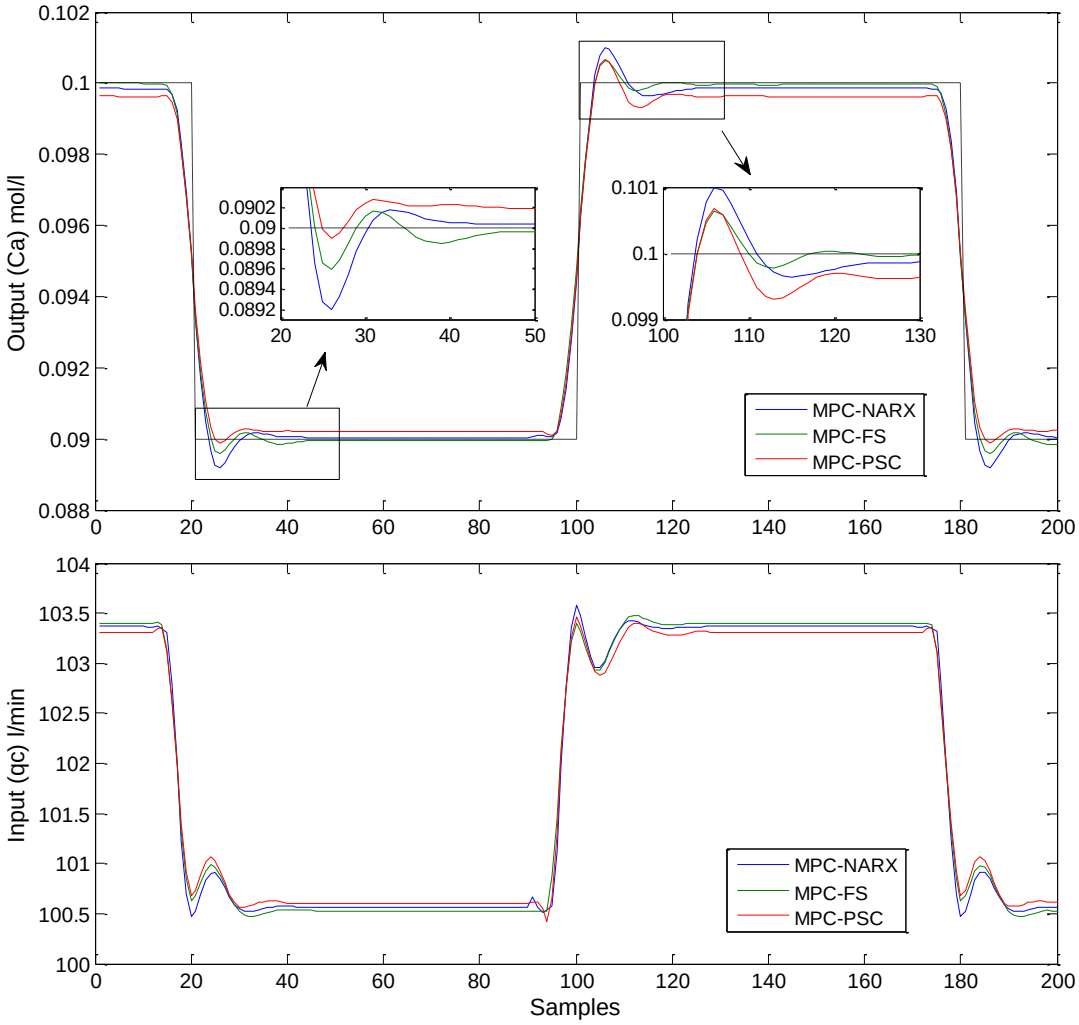

Fig. 4 Control performance in middle level of product concentration

The second experiment is executed with a large step change in the set-point, which involves the low and high levels of product concentration, $C_a = 0.11$ and $C_a = 0.065$, respectively. The product concentration increases from 0.09mol/l to the highest level 0.11mol/l, and then decreases to the lowest level 0.065mol/l. The process responses with the manipulated variables are displayed in Fig. 5. Besides, the MAE is given in Table 3.

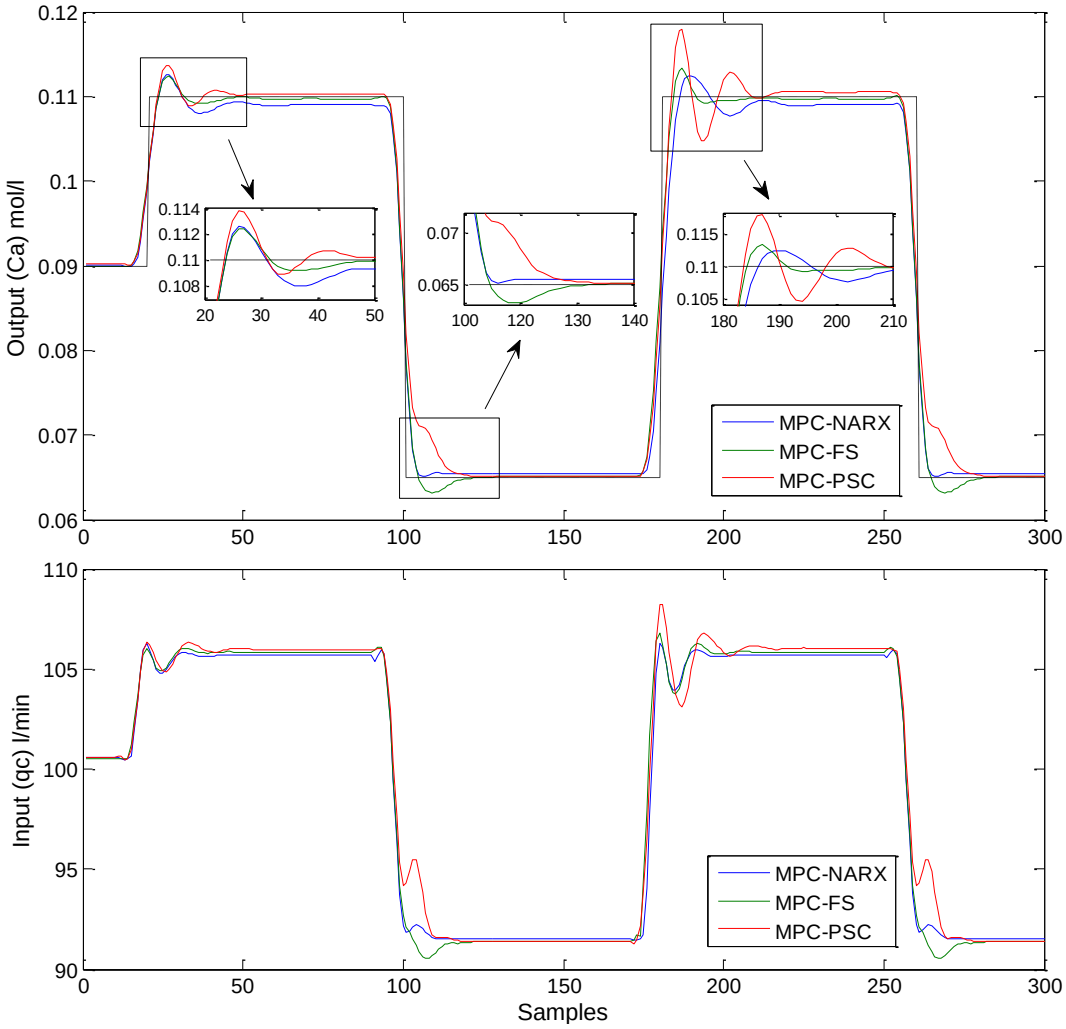

Fig. 5 Control performance in low and high level of product concentration

In the both rise of step changes, the MPC-PSC has the largest overshoot. The MPC-NARX and MPC-FS has similar overshoot when the set-point jumps, but the MPC-NARX has severer oscillation. When the set-point drops, the MPC-PSC has the slowest response. Meanwhile, in the same scenario, there is a small acceptable overshoot for MPC-FS. In both

rise and drop scenarios, the proposed MPC-FS has the smallest steady-state error, as illustrated in Fig. 5. Furthermore, from Table 3, the MAE clearly suggests that the proposed MPC-FS has the best overall control performance.

As a whole, in the evaluation of control performances, it is verified that the proposed MPC-FS has a better performance than that of MPC-NARX and MPC-PSC. In the control in the middle level of product concentration, it can be said that all the three control approaches have similar performance. For the control in low and high level of product concentration, the advantage of proposed MPC-FS over the other two approaches is much more obvious. This is because the nonlinearity of CSTR plant is much stronger on the high level of product concentration than the other regions of the operating space. In addition, as presented in Section 3.3, the MPC-FS has the best modelling performance among the three RBF models, which is the main reason for the better control performance.

Table 3 Control Performance

Control Strategy	MAE (Middle level / Low & High level)
MPC-NARX	0.00037880 / 0.0016
MPC-FS	0.00032564 / 0.0013
MPC-PSC	0.00051959 / 0.0017

5.2 Computing Load

As mentioned previously, although a nonlinear model appears to be a better internal model in MPC, one of disadvantages of nonlinear MPC (NPMC) is its large computational load in on-line optimization of control variables. Therefore, after the evaluation of modelling and control performance, it is imperative to evaluate the computational requirement of all the control strategies in previous section. In this work, MATLAB R2009a on an Intel Core i3 laptop with Windows 7 system is used to carry out this simulation. For a fair comparison, the optimization problem in all control strategies is solved using sequential quadratic programming [25]. The computation time in solving the online optimization problem is measured using tic-toc command in Matlab.

The online computing times of all control algorithms at each sample time for the control in middle level and low/high level of product concentration are shown in Fig. 6 and 7, respectively. The total computation times are recorded in Table 4. It is observed that the computing times of all the three approaches are high when step changes occur. Then, the computing times gradually reduce as the outputs track the constant set-point as illustrated in both Fig. 6 and 7. This is because the control value used this sample time is set as the initial value of the control for next sample time, so that the computing load is greatly reduced. For middle level control, the highest computation time of MPC-NARX, MPC-FS and MPC-PSC are 6s, 3.26s and 9.327s, respectively. For low/high control, the highest computation time of MPC-NARX, MPC-FS and MPC-PSC are 5s, 3.33s and 33s, respectively. Considering the computing time using Matlab is more than 10 times high compared that using the industrial C code, the computing time 6 seconds using Matlab is equivalent to 0.6 seconds using the C code, which is acceptable for the sampling time of 3 second. While the model MPC-PCS for large set-point change uses 33 seconds to solve the optimization problem, so that this algorithm would not be recommended for practical application.

Furthermore, it is obvious the MPC-PSC has the largest total computation time as presented in Table 4. Although the MPC-PSC is a factorized approach, its unrealistic huge network structure to acquire a satisfying modeling performance has increased the size of factorized matrices in (32-33). Therefore, the computation load has inevitably become greater during the execution of optimized control variable.

On the other hand, it is noticeable that the computation time of MPC-FS for low/high level is not as efficient as for middle level due to the high nonlinearity characteristic in this region. Despite this, the proposed MPC-FS is proved to have advantages over MPC-NARX with lower computation times every time step changes occur as shown in Fig. 6 and 7. Furthermore, Table 4 suggests that in overall, the MPC-FS is more efficient than the MPC-NARX. This is because unlike the MPC-NARX, the MPC-FS has the advantages using the factorized model in (32) where $\hat{y}^T_{past,k+1}, \dots, \hat{y}^T_{past,k+f}$ are only needed to be computed once at each sample time instant which reduces the computational requirement. In this simulation, it is proved that the MPC-FS is a more efficient control strategy in term of control results and computational requirement.

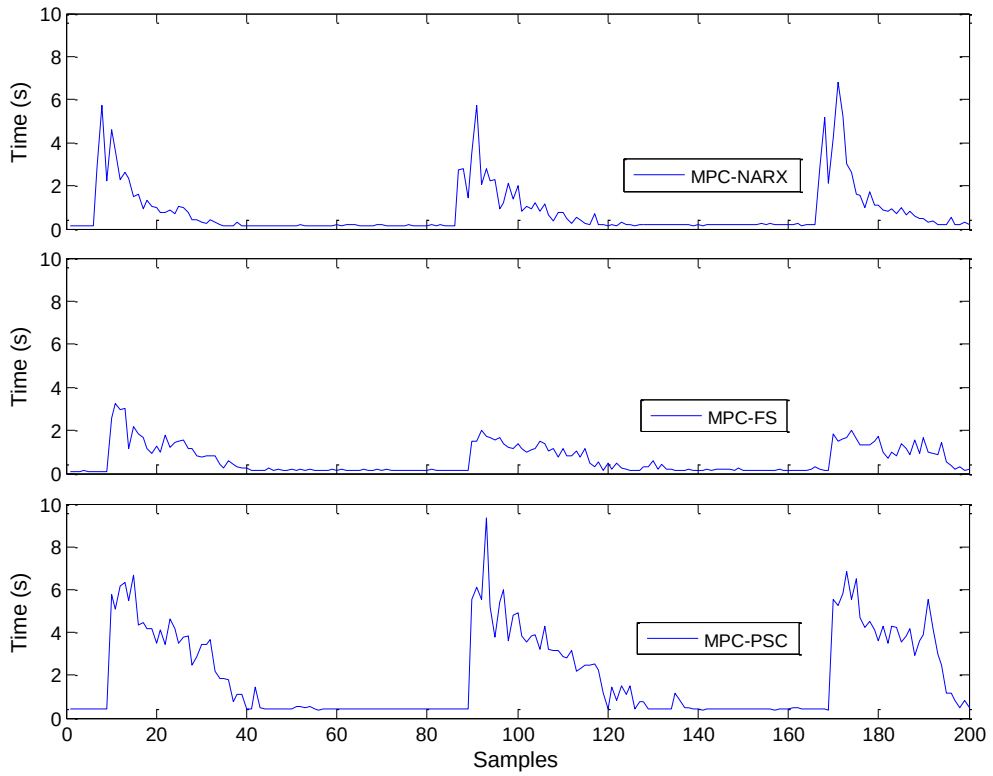

Fig. 6 Computing times of all control approaches at the control in middle level.

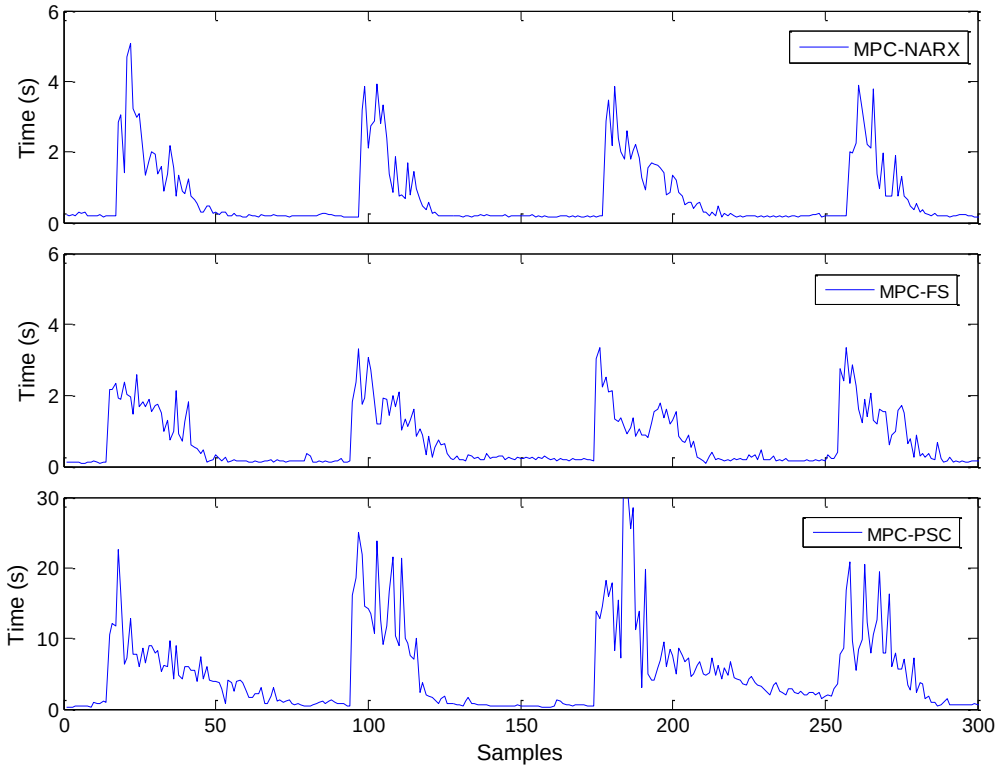

Fig. 7 Computing times of all control approaches at the control in low and high level.

Table 4 Total computation times

Control Strategy	Total Computation Times (s) Middle level / Low & High level
MPC-NARX	155.4303 / 221.0456
MPC-FS	125.3157 / 213.2456
MPC-PSC	394.3540 / 1565.8

6. Conclusion

In conclusion, a new FS-RBF model is developed for nonlinear dynamic process and its factorized form is developed for the application in MPC. A FS prediction model is firstly developed and implemented with RBF network. The effectiveness of the FS-RBF model is verified in modelling a CSTR plant. The comparison results demonstrate that the FS-RBF model outperforms the PSC-RBF model in term of prediction accuracy and model compactness. In addition, the proposed network model matches the NARX-RBF model in model compactness and it presents a better modelling performance. With these two advantages, the developed network model is more effective to be used in MPC for output prediction. After that, the factorization of proposed network is derived and an explicit MPC's objective function is obtained to reduce the computational load. The control performance and the computational load of the MPC based on the proposed model are evaluated by controlling the CSTR plant. Comparing with the existing control approaches, the results show that the proposed control approaches possesses a more efficient and better control performance. These advantages proved that the developed factorized FS-RBF model provided a better approach to be applied in MPC.

Appendix A

CSTR Parameters

product concentration	C_a	0.1 mol/l
reactor temperature	T	438.54 K
coolant flow rate	q_c	103.41 l/min

process flow rate	q	100 l/min
feed concentration	C_{a0}	1 mol/l
inlet coolant temperature	T_{c0}	350 K
CSTR volume	v	100 l
heat transfer coefficient	h_a	7 x 10^5 cal/min/K
reaction rate constant	k_o	7.2 x 10^{10} min^{-1}
activation energy term	E/R	1 x 10^4 K
heat of reaction	ΔH	-2 x 10^5 cal/mol
liquid densities	ρ, ρ_c	1 x 10^3 g/l
specific heats	C_p, C_{pc}	1 cal/g/k

$$k_1 = -\frac{\Delta H k_0}{\rho C_p} \qquad k_2 = \frac{\rho_c}{\rho C_p v} \qquad k_3 = \frac{h_a}{\rho_c C_{pc}}$$

References

1. Forbes, M.G., et al., Model Predictive Control in Industry: Challenges and Opportunities. IFAC-PapersOnLine, 2015. **48**(8): p. 531-538.
2. Carcia, C.E., D.M. Prett, and M. Morari, Model predictive control: theory and practice - a survey. Automatica, 1989. **25**(3): p. 335-348.
3. Qin, S.J. and T.A. Badgwell, A survey of industrial model predictive control technology. Control Engineering Practice, 2003. **11**(7): p. 733-764.
4. Mayne, D.Q., Model predictive control: Recent developments and future promise. Automatica, 2014. **50**(12): p. 2967-2986.
5. Richalet, J., et al., Model predictive heuristic control: applications to industrial processes. Automatica, 1978. **14**: p. 413-428.
6. Allgower, F., R. Findeisen, and Z.K. Nagy, Nonlinear model predictive control: from theory to application. J. Chin. Inst. Chem. Engrs, 2004. **35**(3): p. 299-315.
7. Hao, Y., et al., Advantages of radial basis function networks for dynamic system design. Industrial Electronics, IEEE Transactions on, 2011. **58**(12): p. 5438-5450.
8. Broomhead, D.S. and D. Lowe, Multivariable functional interpolation and adaptive networks. Complex Systems, 1988. **2**: p. 321-355.

9. Hong-Gui Han, Lu-Ming Ge, Jun-Fei Qiao, An adaptive second order fuzzy neural network for nonlinear system modelling, Neurocomputing, Vol. 214, 2016, Pages 837-847.

10. Chi Zhang, Haikun Wei, Liping Xie, Yu Shen, Kanjian Zhang, Direct interval forecasting of wind speed using radial basis function neural networks in a multi-objective optimization framework, Neurocomputing, Vol. 205, September 2016, Pages 53-63.

11. Yiqian Cui, Junyou Shi, Zili Wang, Lazy quantum clustering induced radial basis function networks (LQC-RBFN) with effective centers selection and radii determination, Neurocomputing, Vol. 175, January 2016, Pages 797-807.

12. Q.P. Ha, H. Wahid, H. Duc, M. Azzi, Enhanced radial basis function neural networks for ozone level estimation, Neurocomputing, Vol. 155, May 2015, Pages 62-70.

13. Diaconescu, E., The use of NARX neural networks to predict chaotic time series. WSEAS Transactions on Computer Research, 2008. **3**(3): p. 182-191.

14. Wang, S.W., et al., Adaptive neural network model based predictive control for air–fuel ratio of SI engines. Engineering Applications of Artificial Intelligence, 2006. **19**(2): p. 189-200.

15. Samek, D. and P. Dostal, Artificial neural network with radial basis function in model predictive control of chemical reactor. Mechanics, 2009. **28**(3): p. 91-95.

16. Wang, D., Y. Zhou, and X. He, Radial basis function neural network-based model predictive control for freeway traffic systems. International Journal of Intelligent Systems Technologies and Applications, 2007. **2**(4): p. 370-388.

17. Ding-Li Yu, Ding-Wen Yu, and J. Barry Gomm, Nueral model adaptation and predictive control of a chemical process rig. IEEE Transactions on Control Systems Technology, 2006. **14**(5): p. 828-840.

18. Su, T.H. and T.J. McAvoy, Artificial neural network for nonlinear process identification and control. Nonlinear Process Control, ed. M.A. Henson and D.E. Seborg. 1997, Englewood Cliffs, NJ: Prentice Hall. 371-428.

19. Bhartiya, S. and J.R. Whiteley, Factorized approach to nonlinear MPC using a radial basis function model. AIChE Journal, 2001. **47**(2): p. 358-368.

20. Yu, D.L. and D.W. Yu, A new structure adaptation algorithm for RBF networks and its application. Neural Computing and Applications, 2007. **16**(1): p. 91-100.

21. Yu, D.L., et al. Adaptive RBF model for model-based control. in Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on. 2004.

22. Ljung, L., System Identification: Theory for the User. Second ed. 1999: Prentice-Hall.

23. Lightbody, G. and G.W. Irwin, Nonlinear control structures based on embedded neural system models. Neural Networks, IEEE Transactions on, 1997. **8**(3): p. 553-567.

24. Morningred, J.D., et al. An adaptive nonlinear predictive controller. in American Control Conference, 1990. 1990.

25. Spellucci, P., An sqp method for general nonlinear programs using only equality constrained subproblems. Mathematical Programming 82, 1998: p. 413-448.