ELSEVIER

Chemical Engineering Journal 138 (2008) 274–282

Chemical Engineering Journal

www.elsevier.com/locate/cej

Wiener-neural identification and predictive control of a more realistic plug-flow tubular reactor

Mohammad M. Arefi *, A. Montazeri, J. Poshtan, M.R. Jahed-Motlagh

Iran University of Science and Technology, Tehran 16846, Iran

Received 22 January 2007; received in revised form 2 May 2007; accepted 17 May 2007

Abstract

Some chemical plants such as plug-flow tubular reactors have highly nonlinear behavior. Such processes demand a powerful identification method such as a neural-networks-based Wiener model. In this paper, a plug-flow reactor is simulated in a rather realistic environment by HYSYS, and the obtained data is in connection with MATLAB for identification and control purpose. The process is identified with NN-based Wiener identification method, and two linear and nonlinear model predictive controllers are applied with the ability of rejecting slowly varying unmeasured disturbances. The results are also compared with a common PI controller for temperature control of tubular reactor. Simulation results show that the obtained Wiener model has a good capability to predict the step response of the process. Parameters of both linear and nonlinear model predictive controllers are tuned and the best-obtained results are compared. For this purpose, different operating points are selected to have a wide range of operation for the nonlinear process. It is shown that the nonlinear controller has the fastest damped response in comparison with the other two controllers.

Keywords: Wiener-neural identification; Nonlinear model predictive control (NMPC); Tubular reactor; HYSYS simulator

1. Introduction

There are very few design techniques that can be proved to stabilize processes in the presence of nonlinearities and constraints. Model predictive control (MPC) – a model-based optimal control method – has been one of the successful controllers in manufacturing industries for the past two decades [1]. MPC refers to a class of computer control algorithms that control the future behavior of a plant through the use of an explicit process model. At each control interval, the MPC algorithm computes an open-loop sequence of manipulated variable adjustments in order to optimize future plant behavior. The first input in the optimal sequence is injected into the plant, and the entire optimization is repeated at subsequent control intervals [1]. By now, the application of MPC controllers based on linear dynamic models cover a wide range of applications, and linear MPC theory can be considered quite mature. Nevertheless, many manufacturing processes are inherently nonlinear and there are cases where nonlinear effects are significant and cannot be ignored. These include at least two broad categories of applications [1]:

1. Regulator control problems where the process is highly nonlinear and subject to large frequent disturbances (pH control, etc.).
2. Servo control problems where the operating points change frequently and span a wide range of nonlinear process dynamics (polymer manufacturing, ammonia synthesis, etc.).

In fact higher product quality specifications and increasing productivity demands, tighter environmental regulations and demanding economical considerations require to operate systems over a wide range of operating conditions and often near the boundary of the admissible region [2]. Besides, the operating point in some batch processes is not in steady-state and all of the operations are performed in transient conditions [3]. Under these conditions linear models are often not sufficient to describe the process dynamics adequately and nonlinear models must be used.

In recent years, several nonlinear model predictive control (NMPC) techniques from identification as well as control points of view are addressed for different processes in literatures.

* Corresponding author. Tel.: +98 21 77240492; fax: +98 21 77240490.
E-mail address: mmarefi@ee.iust.ac.ir (M.M. Arefi).

1385-8947/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.cej.2007.05.044

Among these techniques, neural networks play important role especially in the identification phase of NMPC algorithms due to their ability to learn by example [4–10]. Moreover, they also have the ability to learn the frequently complex dynamic behavior of a physical system. For an overall review on the application of neural networks in the control of chemical processes, the interested readers may refer to [11]. For example in Ref. [5], neural networks are used to develop a model for highly nonlinear CSTR and pH neutralization processes. A nonlinear internal model controller is designed based on these models and results are compared with a PID controller. In Ref. [6], an RBF neural network is used for modeling and control of an unstable CSTR process. The use of neural network for modeling of a CSTR process has also been reported in Ref. [7], where the controller is designed using classical optimization methods. The main problem with neural network as a model is that it performs well in the range of the data used for training the network, but has poor extrapolation property in other regions. To cope with this problem, dynamic nonlinear models are proposed [4,8]. The use of Wiener models where a linear dynamic model is followed by a static nonlinearity is one of the solutions. Several NMPC methods for pH neutralization and CSTR processes using Wiener model is addressed in [12–14]. For example in Refs. [12,13], a static nonlinear term is used to model the inverse of the nonlinearity of the plant and is selected as a polynomial with proper degree. Besides, in Ref. [14], the nonlinear term and its inverse are modeled using piecewise linear method. In Ref. [15], a nonlinear combination of Laguerre models followed by a single-layer neural network is introduced as an efficient nonlinear identification method used in MPC applications. The capability of this technique is showed by the identification of some highly nonlinear plants including pH and CSTR. In Ref. [16], a MIMO Wiener model of a polymerization reactor is identified and the model is used in an MPC scheme. The quality of proposed controller is also compared with that of linear MPC. This algorithm is based on the PI-MOESP method for the estimation of system matrices of the linear part [17]. In Ref. [18], a distillation column simulation model is used as a benchmark to demonstrate the benefits of a Wiener model based identification and control methodology. The results show the capability of this technique in identifying nonlinear ill-conditioned plant compared to existing linear techniques. Despite the fact that many manufacturing systems are nonlinear, however they have been controlled by PIDs. However, because of the tuning and robustness difficulties of PID controllers as reported in Refs. [19,20], a more reliable controller based on nonlinear model of the process may be needed.

In this paper, a nonlinear model predictive control is proposed based on classic optimization methods with nonlinear identification using Wiener model for a highly nonlinear plug-flow tabular reactor. Because of the capability of Wiener model structure in comparison with Hammerstein models in capturing complex nonlinear dynamics, a Wiener structure with a state-space model in linear part and neural networks in nonlinear part is selected. Both these parts are parameterized and an overall optimization is performed on the parameters based on collected data. This Wiener structure and identification technique is different from those reported in the literature [e.g. 12–18]. The results of identification and control are also compared with linear MPC and classical PI controller to show the superiority of the proposed method. In addition, to have more realistic simulations, the model of process is simulated in HYSYS, and the data is transferred (real-time) to MATLAB for identification and control purposes. After this introduction, the theory of Wiener identification using neural network as the static nonlinear term is presented. Also selection of the test signal to achieve the best result for identification is studied. In Section 3, the design of nonlinear model predictive control based on the identified Wiener model is presented. Simulation results for identification and control of plug-flow tubular reactor are given in Section 4.

2. NN-based Wiener identification

2.1. Wiener identification

Among the nonlinear black box models, the block-oriented models are efficient structures in nonlinear modeling. These models consist of a series connection of a linear dynamic element and a static nonlinear element.

A Wiener model consists of a dynamic linear block (H1) in cascade with a static nonlinearity at the output (H2), as shown in Fig. 1. Here $z(k) \in R^l$ is an intermediate signal that does not necessarily have a physical meaning. On the other hand, in the Hammerstein model the static input nonlinearity precedes the linear block.

In certain respects, Hammerstein models are very similar to the linear models on which they are based. For example, if $u(k)$ is a piecewise constant input sequence [e.g. pulses, steps, pseudo-random binary sequences (PRBS), etc.], for any static nonlinearity the intermediate variable sequence will also be a piecewise constant sequence with the same general character (specifically, with transitions at the same instants as $u(k)$, but assuming different values). Hammerstein models have been considered as alternatives to linear models in a number of chemical process applications [21].

In particular, while Hammerstein and Wiener models exhibit exactly the same steady state behavior, the differences in their transient responses can be quite significant. As a specific example, the general character of the step response can change with the sign and/or magnitude of the input step, unlike the case of the Hammerstein model, where this general character is determined entirely by the linear part [21]. Because of this behavior and the capability of modeling complex nonlinear dynamics by Wiener models led us to the selection of this model structure.

State-space representation of a Wiener model can be stated as follows:

$$x(k+1) = \mathbf{A}x(k) + \mathbf{B}u(k), \qquad z(k) = \mathbf{C}x(k) + \mathbf{D}u(k),$$
$$y(k) = f(z(k)) + v(k) \tag{1}$$

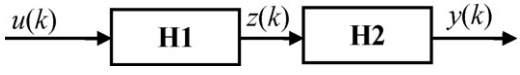

Fig. 1. The Wiener model.

where $x(k)$ is the $n \times 1$ state vector at time k, $u(k)$ the $m \times 1$ vector of control input, $y(k)$ the $l \times 1$ vector of measured output, and $v(k)$ is a measurement noise assumed to be zero-mean and independent of $u(k)$for all k's. The system matrices **A**, **B**, **C** and **D** are real with proper dimensions and $f(\cdot)$ is a nonlinear vector function defined on $R^l \to R^l$. The sequences of input and output data used for identification of (1) are available. Besides, it is assumed that the input sequences $\{u(k)\}$ are persistently exciting [22] and statistically independent of noise sequences $\{v(k)\}$. The systematic approach for identification of the above problem is stated completely in Ref. [23]. The first step is identification of linear part using state-space methods. So assuming the nonlinear mapping as an identity, the linear dynamics characterized by quadruple (**A**, **B**, **C** and **D**) will be identified. Then using the identified matrices (**A**, **B**, **C** and **D**), the output sequences of this LTI system $\{\hat{z}(k)\}_{k=1}^{N}$ will be computed. With this sequence, a primary identification of the nonlinear part of the Wiener model can be estimated. Here, this static nonlinear term is identified using a single layer neural network with the following structure:

$$f_s(z(k)) = \sum_{i=1}^{\upsilon} \left(\alpha(s,i)\phi \left(\sum_{j=1}^{l} \beta(s,i,j) z_j(k) + b(s,i) \right) \right) + b(s,\upsilon+1) + \varepsilon_s(k) \tag{2}$$

where $f_s(\cdot)$ and $z_s(k)$ are used to characterize the sth input and output of the nonlinear term. Besides, $\alpha(s,i)$, $\beta(s,i,j)$, $b(s,i)$ and $b(s,\upsilon+1)$ are unknown real coefficients stacked in the parameter vector $\boldsymbol{\theta} \in R^{l((l+2)\upsilon+1)}$ and must be estimated using nonlinear least square methods. υ is the number of neurons in the hidden layer and the last term $\varepsilon(k)$ shows the estimation error. The cost function minimized for estimation of $\boldsymbol{\theta}$ is:

$$\min_{\boldsymbol{\theta}} \sum_{k=1}^{N} \left\| \begin{array}{l} y_1(k) - \sum_{i=1}^{\upsilon} \left(\alpha(1,i)\phi \left(\sum_{j=1}^{l} \beta(1,i,j)\hat{z}_j(k) + b(1,i) \right) \right) + b(1,\upsilon+1) \\ \vdots \\ y_l(k) - \sum_{i=1}^{\upsilon} \left(\alpha(l,i)\phi \left(\sum_{j=1}^{l} \beta(l,i,j)\hat{z}_j(k) + b(l,i) \right) \right) + b(l,\upsilon+1) \end{array} \right\|^2 \tag{3}$$

Finally, the best parameters for the linear and nonlinear parts are identified with an optimization algorithm. For this purpose, the system matrices identified for the linear part and the parameter vector estimated for the nonlinear part are used as initial conditions for the calculation of the final parameters. Despite the parameter vector defined for the nonlinear part, a full parameterization of the Wiener model in (1) requires that the system matrices (**A**, **B**, **C** and **D**) and also the vector of initial conditions $x(1)$ be included in the parameter vector. To have minimum parameters for the matrices, the pair (**A** and **C**) must be transformed with similar transformations to what is called an output normal form.

Definition. The pair (**A** and **C**) of the system matrices (**A**, **B**, **C** and **D**) is in output normal form if $\mathbf{A}^{\mathrm{T}}\mathbf{A} + \mathbf{C}^{\mathrm{T}}\mathbf{C} = \mathbf{I}_n$, where $\mathbf{I}_n \in R^{n \times n}$ is an identity matrix. The above definition explicitly shows that matrix **A** must be asymptotically stable. In order for the state space description of the system to be unique, the matrices **A** and **C** are transformed such that $\begin{bmatrix} \mathbf{C} \\ \mathbf{A} \end{bmatrix}$ be lower triangular with positive elements in the diagonal. After these transformations, the parameterization can be performed using nl parameters. More details about this method of parameterization can be found in Refs. [24,25]. All parameters of the system matrices after this parameterization are stacked in the vector $\boldsymbol{\theta}_{\mathrm{on}}$. The estimation of all parameters of the parameterized Wiener system can be obtained by minimizing this performance index:

$$\min_{x(1),\boldsymbol{\theta}_{\mathrm{on}},\boldsymbol{\theta}} = \sum_{k=1}^{N} \| y(k) - \hat{y}(k, x(1), \boldsymbol{\theta}_{\mathrm{on}}, \boldsymbol{\theta}) \|^2 \tag{4}$$

where N is the number of samples used for identification. To obtain all parameters of the system, the above least square minimization must be solved. The method used here is Levenberg–Marquardt which tries to find the local minima of the performance index iteratively. If $\boldsymbol{\Theta}$ is the vector of all parameters, by defining $e(\boldsymbol{\Theta}) = y - \hat{y}(\boldsymbol{\Theta})$, where $e(\boldsymbol{\Theta})$ is the error between the target and output vector, the parameters of $(\boldsymbol{\Theta})$ can be updated in each iteration. Suppose the value of these parameters at iteration t of Levenberg–Marquardt algorithm is shown by $\boldsymbol{\Theta}(t)$, then this algorithm can be stated as follows:

$$\boldsymbol{\Theta}(t+1) = \boldsymbol{\Theta}(t) + \Delta\boldsymbol{\Theta}(t) \tag{5}$$

where $\Delta\boldsymbol{\Theta}$ is obtained by solving this set of nonlinear equations:

$$(\mathbf{J}^{\mathrm{T}}(t)\mathbf{J}(t) + \mu\mathbf{I})\Delta\boldsymbol{\Theta} = -\mathbf{J}^{\mathrm{T}} e(\boldsymbol{\Theta}(t)) \tag{6}$$

and $\mathbf{J}(t)$ is the Jacobian matrix with these derivatives:

$$\mathbf{J}_{ij} := \frac{\partial \hat{y}_i(\boldsymbol{\Theta}(t))}{\partial \boldsymbol{\Theta}_j}, \quad i = 1:\ N, \quad j = 1:\ \mathrm{length}\,(\boldsymbol{\Theta}). \tag{7}$$

The tuning parameter $\mu \in (0,\infty)$ is called the Levenberg factor and is necessary for convergence of the algorithm.

Although normalized gradient descent method has the fastest convergence response among gradient descent techniques, but its convergence speed is lower than that of Newton method [26]. Because of the computational complexity of Newton methods, Levenberg–Marquardt takes the advantages of Newton method with lower computation [26]. By setting μ to zero, the fast convergence of Gauss–Newton iteration for small residual problems is achieved. If μ is too small, the algorithm may diverge. The choice of μ in an adaptive manner will have some benefits. For big values of μ, the convergence is very slow. The algorithm tries to keep μ as small as possible. If the cost function decreases, the current step is accepted, and the ratio

of the actual decrease compared to the predicted decrease is checked. Then μ is decreased if the ratio was acceptable and increased otherwise. If the cost function increases, the step is rejected and calculations are repeated with an increased μ [26].

The derivatives in (7) are calculated analytically for the nonlinear part, and are approximated using forward difference method for the linear part [25]. All identification procedures are performed using SLICOT toolbox [25].

2.2. Test design

Some important factors which must be considered in designing the identification test for nonlinear systems are: the duration of the test signal, its amplitude and shape, its spectrum (the average switching time), the correlation of the test signal in each channel, and the number of manipulated variables in each test.

Traditionally, pseudo-random binary sequences (PRBS) are used as the inputs to a system in order to produce representative sets of data to be analyzed. In theory, a PRBS excites the range of dynamics present in a system so that a dynamic model can be produced which contains these dynamics. This is not sufficient, however, for fitting a Wiener model. Since these models have nonlinear gains, an input signal must be used which also demonstrates the response of the system to a range of amplitude changes. A signal that satisfies these criteria is a GMN [27] or a modified PRBS signal [13] which, in addition to random frequency, also exhibits random amplitude changes.

Since in nonlinear systems the test time depends mainly on the number of parameters in the model and the level of noise and unmeasured disturbances, longer test time is recommended in comparison with linear systems. This is typically considered about 16–25 times the settling time of the process. Other factors may be included by choosing one of the following excitation signals [27]:

I. *Staircase test*. In this type of test, the width of the pulses and their numbers must be selected properly.
II. *Generalized multiple-level noise* (*GMN*). This type of test is a multi-level extension of generalized binary noise. In this test the amplitude and the number of pulses must be selected suitably. The number of levels on this test is equal or greater than the degree of the nonlinear polynomial to be identified. Moreover, the average switching time of the test can be obtained from $T_{sw} = T/3$, where T is 98% of the process settling time.
III. *Filtered white uniform noise*. The flexibility in shaping the spectrum of this type of signal is its main advantage. Each spectrum may be realized with a proper filter. A first order low-pass filter is often suitable for this purpose.

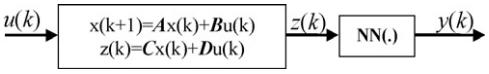

Fig. 2. The Wiener model for NMPC.

3. Nonlinear model predictive controller

If at time k, the future state and behavior of the plant is assumed to be known, they can be written in vector form in MIMO case as follows:

$$\mathbf{z}(k) = [\, z^{\mathrm{T}}(k+1) \quad z^{\mathrm{T}}(k+2) \quad \cdots \quad z^{\mathrm{T}}(k+P)\,]^{\mathrm{T}} \tag{8}$$

$$\mathbf{u}(k) = [\, u^{\mathrm{T}}(k+1) \quad u^{\mathrm{T}}(k+2) \quad \cdots \quad u^{\mathrm{T}}(k+M)\,]^{\mathrm{T}} \tag{9}$$

$$\mathbf{y}(k) = [\, y^{\mathrm{T}}(k+1) \quad y^{\mathrm{T}}(k+2) \quad \cdots \quad y^{\mathrm{T}}(k+P)\,]^{\mathrm{T}} \tag{10}$$

$$\mathbf{r}(k) = [\, r^{\mathrm{T}}(k+1) \quad r^{\mathrm{T}}(k+2) \quad \cdots \quad r^{\mathrm{T}}(k+P)\,]^{\mathrm{T}} \tag{11}$$

where $\mathbf{z}(k)$ is the vector of the linear model outputs, $\mathbf{u}(k)$ the vector of manipulating variables, $\mathbf{y}(k)$ the vector of the Wiener model outputs shown in Fig. 2, and $\mathbf{r}(k)$ is the vector consisting set points.

Also M and P are the control and prediction horizons, respectively. The predicted output of the linear model can be written as

$$\hat{\mathbf{z}}(k) = \beta \bar{\mathbf{u}}(k) + \xi x(k) \tag{12}$$

where β, ξ and $\bar{\mathbf{u}}(k)$ are defined by

$$\beta = \begin{bmatrix} \mathbf{CB} & \mathbf{D} & 0 & 0 & \cdots & 0 \\ \mathbf{CAB} & \mathbf{CB} & \mathbf{D} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{M-1}\mathbf{B} & \mathbf{CA}^{M-2}\mathbf{B} & \mathbf{CA}^{M-3}\mathbf{B} & \mathbf{CA}^{M-4}\mathbf{B} & \cdots & \mathbf{D} \\ \mathbf{CA}^{M}\mathbf{B} & \mathbf{CA}^{M-1}\mathbf{B} & \mathbf{CA}^{M-2}\mathbf{B} & \mathbf{CA}^{M-3}\mathbf{B} & \cdots & \mathbf{D}+\mathbf{CB} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{P-1}\mathbf{B} & \mathbf{CA}^{P-2}\mathbf{B} & \mathbf{CA}^{P-3}\mathbf{B} & \mathbf{CA}^{P-4}\mathbf{B} & \cdots & \mathbf{D}+\sum_{i=M+1}^{P}\mathbf{CA}^{P-i}\mathbf{B} \end{bmatrix} \tag{13}$$

$$\xi = \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^M \\ \mathbf{CA}^{M+1} \\ \vdots \\ \mathbf{CA}^P \end{bmatrix} \tag{14}$$

$$\bar{\mathbf{u}}(k) = [\, u^{\mathrm{T}}(k) \quad \mathbf{u}^{\mathrm{T}}(k)\,]^{\mathrm{T}} \tag{15}$$

Besides, the predicted output of the Wiener model is given by

$$\hat{\mathbf{y}}(k) = \begin{bmatrix} f(\hat{z}(k+1)) \\ f(\hat{z}(k+2)) \\ f(\hat{z}(k+3)) \\ \vdots \\ f(\hat{z}(k+P)) \end{bmatrix} = f(\hat{\mathbf{z}}(k)) \tag{16}$$

Finally, by solving and minimizing the following optimization problem the control signal applied to the process can be obtained:

$$\min_{u(k+1),u(k+2),\ldots,u(k+M)} J = \sum_{j=1}^{P} \|\hat{y}(k+j) - r(k)\|_{\mathbf{Q}}^2 + \sum_{j=1}^{M} \|\Delta u(k+j)\|_{\mathbf{S}}^2 + \sum_{j=1}^{M} \|u(k+j)\|_{\mathbf{R}}^2 \tag{17}$$

where

$$\Delta u(k+j) = u(k+j) - u(k+j-1) \tag{18}$$

In (17), **Q** and **S** are the weighting matrices for the output and rate of change of the control input and **R** is the weighting matrix for manipulated variable. In addition, it is assumed that:

$$u(k+j) = u(k+M) \quad j = M+1 \ldots P \tag{19}$$

In this optimization problem the eligible limits for the control input and its rate of change, and also those for the output signal may be considered with these inequalities:

$$\begin{aligned} &\mathbf{u}_{\min} \le \mathbf{u}(k) \le \mathbf{u}_{\max} && \forall k \\ &\mathrm{d}\mathbf{u}_{\min} \le \mathbf{u}(k) - \mathbf{u}(k-1) \le \mathrm{d}\mathbf{u}_{\max} && \forall k \\ &\mathbf{y}_{\min} \le \mathbf{y}(k) \le \mathbf{y}_{\max} && \forall k \end{aligned} \tag{20}$$

The optimization problem stated above can be solved with Successive Quadratic Programming (SQP) method. It is important to notice that the optimization time must be less than the sampling time of the process so that the obtained control signal can be applied to the process.

To compensate the eventual mismatch between the process and its model, and in order to consider unmeasured disturbances in the process, a term must be added to the predicted output of the plant [7], like:

$$\mathbf{d}(k) = \mathbf{y}(k) - \mathbf{y}_{\mathrm{m}}(k) \tag{21}$$

where $\mathbf{y}(k)$ is the output of the real process, and $\mathbf{y}_{\mathrm{m}}(k)$ is the model output. The modified predicted output will be:

$$\mathbf{y}_{\mathrm{pred}}(k+i) = \mathbf{y}_{\mathrm{m}}(k+i) + \mathbf{d}(k) \quad \text{for } i = 1, \ldots, P \tag{22}$$

4. Simulation results

4.1. Identification results

Reactors are the heart of many chemical processes, and dynamic simulation of these critical units is absolutely essential for the safe and profitable operation of the entire plant [28]. There are a tremendous number of different types of reactors that are used commercially. Reactors in which exothermic, irreversible reactions take place are the most challenging because of the potential for temperature runaways. The chemical and petroleum industries have experienced a number of devastating fires, explosions and emissions of toxic material during the last century caused by chemical reactors. Many of these could have been prevented by better design and operation. Therefore, reactor dynamics and control are probably the most vital parts of dynamic simulations [28].

Many industrial processes use tubular reactors, particularly those in which a solid catalyst is required. The typical tubular reactor has a vessel that is packed with solid catalyst. The important difference between CSTR reactors and tubular reactors is the distributed nature of latter: temperature and composition vary down the length of the tubular reactors, and they also vary with time. This makes the models and the dynamics more complex.

The case study considered in this paper is the chlorination of propylene. The model of the process is simulated numerically with HYSYS 3.1 software, and its nominal parameters are selected based on the data given in Ref. [28]. For control purpose, HYSYS is connected to MATLAB 7.1 using *HYSYSLIB* toolbox with some modifications [29] to have more access to the desired variables.

The process contains two parallel gas phase reactions. The first forms allyl chloride and HCL:

$$\mathrm{C_3H_6 + Cl_2 \rightarrow CH_2 = CH\text{-}CH_2Cl + HCl} \tag{23}$$

and the second forms 1,2-dichoro propane:

$$\mathrm{C_3H_6 + Cl_2 \rightarrow CH_2Cl\text{-}CHCl\text{-}CH_3} \tag{24}$$

Reaction rates have a first-order dependence on the partial pressure of the reactants. Using English units in HYSYS, the reaction rates are given in lb mol/h ft^3, with temperature in Rankin, activation energy in Btu/lb mol, and pressure in atmospheres:

$$R_1 = k_1 P_{\mathrm{C3}} P_{\mathrm{Cl_2}} = (2.06 \times 10^5\, \mathrm{e}^{-27{,}200/RT}) P_{\mathrm{C3}} P_{\mathrm{Cl_2}} \tag{25}$$

$$R_2 = k_2 P_{\mathrm{C3}} P_{\mathrm{Cl_2}} = (11.7\, \mathrm{e}^{-6860/RT}) P_{\mathrm{C3}} P_{\mathrm{Cl_2}} \tag{26}$$

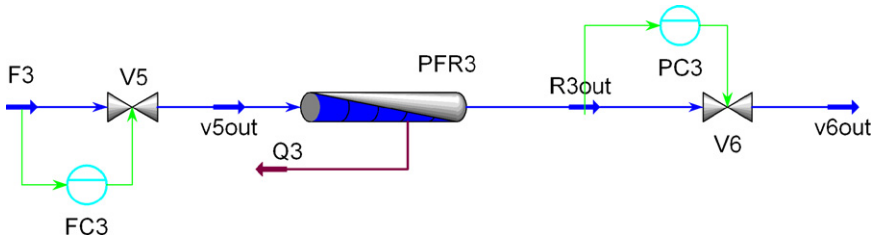

Fig. 3. Schematic representation of the plug-flow tubular reactor process.

If reactor is operated adiabatically, the temperature of the gas leaving the reactor is predicted by HYSYS to be 716 °F, and the chlorine concentration is 9.89 mol% (50% conversion). Several cases are considered in this section with varying types and amounts of heat transfer. In these cases the outlet temperatures are different, as are the conversions of chlorine. A control valve on the gas feeding the reactor is designed for a 20 psi drop when 50% open at design flow rate. A flow controller manipulates this valve to control feed flow. A valve on the exit line from the reactor is used to hold pressure in the reactor. This valve is designed for a 10 psi pressure drop when 50% open at design flow rate. In this paper, the use of tubular reactor without catalyst is considered.

In Ref. [28], two methodologies for temperature control of reactor are proposed. The first case uses the direct Q model, and the second case, available in HYSYS and used here, is one in which a coolant is used.

The aim is to control the temperature of the output liquid of the reactor (T_{R3out}) by manipulating the coolant flow (Q3). The nominal parameters of output temperature and cooling fluid flow rate are 272.4 °C and 0.5882 kg mol/h, respectively. The process is schematically depicted in Fig. 3.

Moreover, the feed flow of v5out is considered as the unmeasured disturbance of the process. In order to have a more realistic simulation of plug-flow reactor and to be able to tune the control parameters reliably for real applications, the important parameters to specify the heat-transfer information are given in Table 1; these must be known in advance.

The remaining parameters are calculated from those specified. Note that the temperature level of the cooling fluid is too high to use cooling water. A high temperature fluid, such as DowTherm, would be used to cool this reactor. Fig. 4 shows the open-loop response of the process for ±20% step change in flow Q3. It can be seen that the process is highly nonlinear and the steady-state gain for −20% changes is about 25% greater than that for +20%.

Table 1
Heat-transfer and coolant flow information of plug-flow reactor

Simulation parameters	Nominal value
Heat capacity of the coolant	75.00 kJ/kg mol-C
Inlet temperature of cooling fluid	204.4 °C
Available UA	149.08 kJ/C-h
Utility holdup	0.5126 kg mol
Mole flow (cooling flow rate)	0.4429 kg mol/h
Min flow (cooling flow rate)	0 kg mol/h
Max flow (cooling flow rate)	1.361 kg mol/h

To identify this process, a GMN signal at eight levels: 0, 0.3, 0.5882, 0.7, 0.9, 1.1, 1.2 and 1.361 is generated in MATLAB as the excitation signal. The average switching time between these levels is selected as 20 samples. This signal is applied as the input signal to the process modeled in HYSYS. Input and output data are gathered with sampling time of 1 min, and 2000 samples are used for identification purpose. Fig. 5 shows the input (coolant flow) and output (outlet reactor temperature) data collected for identification of the process.

The identification has been performed using the above-mentioned Wiener model with four neurons in the hidden layer.

To get a better view of how to select the number of hidden neurons, the steady state nonlinear gain of the process simulated in HYSYS is plotted Fig. 6. The nonlinear gain of the Wiener-neural model is also calculated and plotted in the same figure for comparison. The result for three, four and five hidden neurons is shown in this figure. It can be seen that the four-neuron model will result in the best match especially for low input amplitude. The obtained models are used in an NMPC scheme and the mean absolute error (MAE) of the controlled variable is calculated. These results also confirm that a four-neuron Wiener model is the best choice.

One thousand and five hundred samples of data are selected for identification, and the rest are used for validation of the obtained model. The validation of the identified model is shown in Fig. 7. To have a better validation, the step responses of the actual process and the identified model are shown in Fig. 8. It can be seen that the fitting of the Wiener model is very good for both cases.

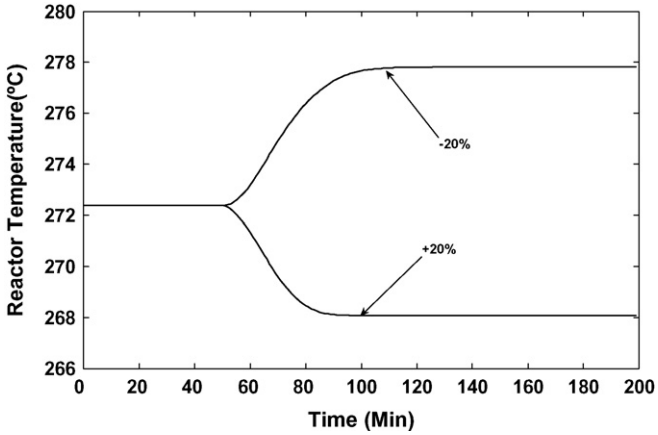

Fig. 4. Open-loop step-response of the plug-flow tubular reactor for changes in the coolant flow.

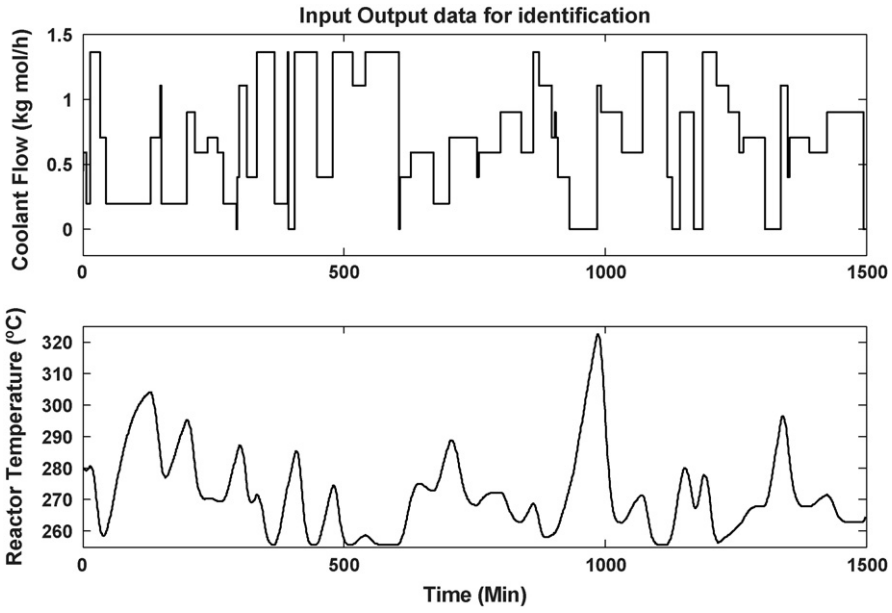

Fig. 5. GMN input and output signals for identification of tubular reactor.

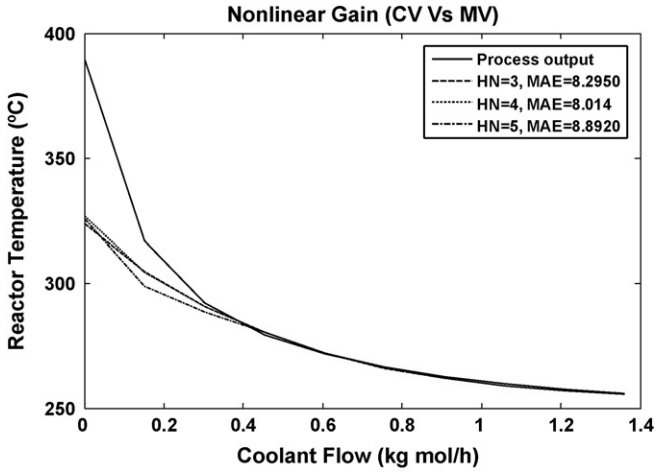

Fig. 6. Nonlinear steady state gain of Wiener model for different hidden layer neurons.

4.2. Control results

Setpoint tracking behavior of the regulator (closed-loop) system with NMPC, along with the coolant flow signal is shown in Fig. 9. It can be seen that the response shows a good tracking speed and low overshoot for all operating points. Also the control signal has a reasonable amplitude and rate of change with respect to constraints applied for optimization. Besides, the comparison of this result with linear MPC and PI controllers is shown in Fig. 9. It can be seen that the results approve the higher performance of the NMPC for different operating conditions, especially when it is far from the point where the linear model is identified. The prediction and control horizons are tuned by trial and error at 10 and 5, respectively. The weighting matrices are selected as $\mathbf{Q} = 1300$, $\mathbf{S} = 150$ and $\mathbf{R} = 1000$. Also a lower limit of 0 kg mol/h and an upper limit of 1.361 kg mol/h are chosen for imposing saturation constraints for the manipulated variable, and the corresponding values for rate of change of manipulated

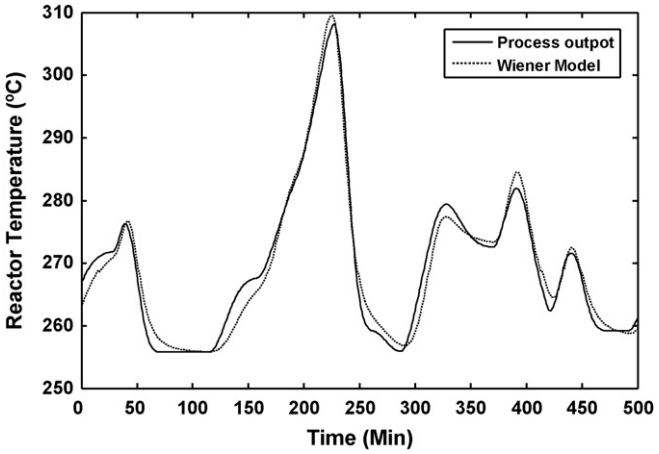

Fig. 7. Validation results for the identified NN-based Wiener model.

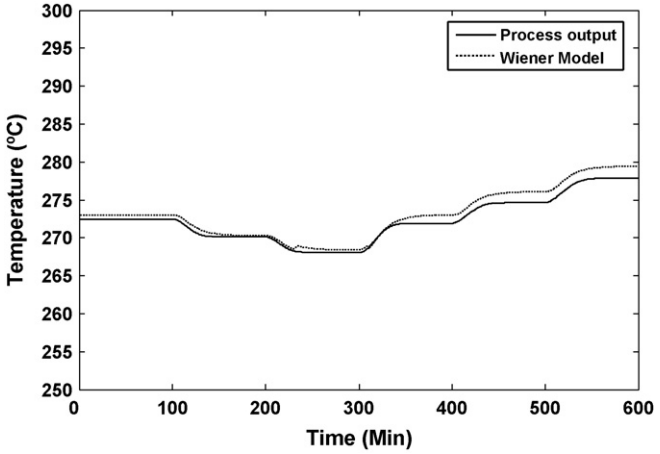

Fig. 8. Step response of the Wiener model for changes in the coolant flow.

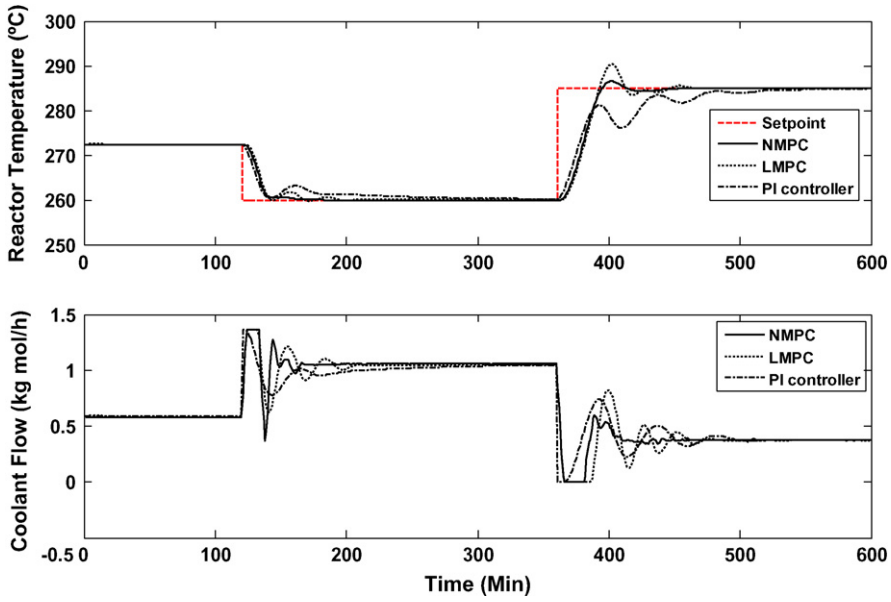

Fig. 9. Up: the performance of NMPC when tracking set-point in tubular reactor; down: the corresponding inlet coolant flow.

Table 2
Different criteria for applied controllers in set-point tracking

Criteria	Controller		
	NMPC	MPC	PI
MAE	1.2031	1.3544	1.9712
SAE	721.8633	812.6548	1182.7
MSE	18.3394	20.0719	19.5969
SSE	11,004	12,043	11,758

variable are ±0.2 kg mol/h. In Table 2, different criteria for different controllers are compared. As can be seen, the NMPC shows better performance compared to the other two controllers.

The computation time required for generating the control signal in the NMPC simulation is shown in Fig. 10. As can be seen in this figure, the maximum computation time for optimization is 0.5108 s which is sufficiently below the chosen sampling time

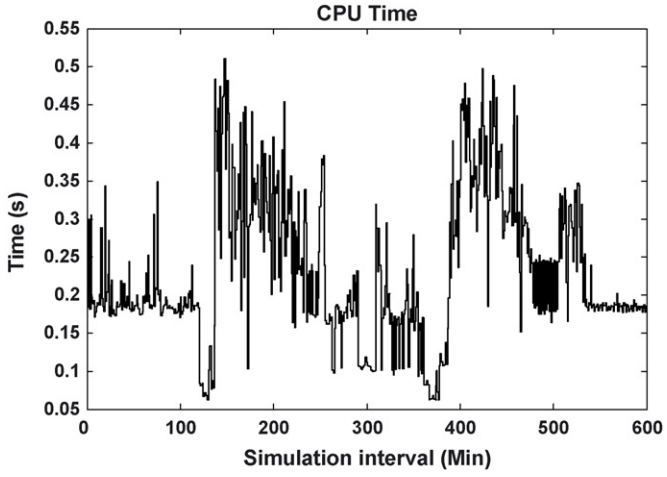

Fig. 10. Computation time of the CPU for SQP optimization.

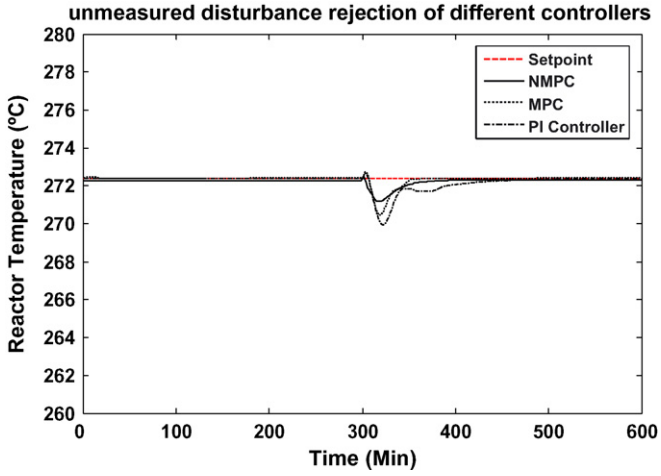

Fig. 11. The performance of the NMPC controller in rejecting unmeasured disturbance.

of 1 min for the process. The SQP optimization is performed using the *fmincon* function of MATLAB.

The performance of different controllers in rejecting +20% changes in the feed flow as the unmeasured disturbance is shown in Fig. 11. These results show the capability of the proposed NMPC controller in rejecting unmeasured disturbances.

5. Conclusions

In this paper, a nonlinear model predictive control for a tubular reactor process is simulated. This process has strong nonlinearity and wide range of operating points. These properties make the linear MPC techniques unsuitable and hence demand a more complex identification and controller design procedure. The process is simulated in a realistic environment with HYSYS for gathering required data, and is connected with MATALB for identification and control purposes.

A Wiener model is chosen with neural network as the static nonlinear term. Simulation results from the identification phase approve the validation of the identified model. Besides, the step responses of the plant and the identified model are in good agreement. This shows the ability of this type of model structure for modeling such a highly nonlinear process. Simulation of the NMPC in HYSYS for a wide range of operating points shows superior performance of the NMPC compared to the linear MPC and PI controllers. This is especially true when the operating condition of process is far from the point where the model for linear MPC is identified. Results show that in such conditions the linear MPC and PI controllers fail to follow the set point adequately, while the nonlinear MPC exhibits a desirable fast response with smoother changes in the control effort. Simulations also confirm that the designed controllers have the capability to reject slowly varying unmeasured disturbances which are common happens in chemical processes.

Acknowledgement

The authors would like to thank Dr. Vasile Sima from National Institute for Research & Development in Informatics for his useful help and comments on working with SLICOT software.

References

[1] S.J. Qin, T.A. Badgwell, An overview of nonlinear model predictive applications, in: IFAC Workshop on Nonlinear Model Predictive Control, Assessment and Future Direction, Ascona, Switzerland, 1998.
[2] R. Findeisen, L. Imsland, F. Allgower, B.A. Foss, State and output feedback nonlinear model predictive control, Eur. J. Control 9 (2/3) (2003) 190–207.
[3] E.F. Camacho, C. Bordons, Model Predictive Control, 2nd ed., Springer-Verlag, London, 2004.
[4] S. Piche, B.S. Rodsari, D. Johnson, M. Gerules, Nonlinear model predictive control using neural network, Proc. IEEE Contr. Syst. Mag. 20 (3) (2000) 53–62.
[5] E.P. Nahas, M.A. Henson, D.E. Seborg, Nonlinear internal model control strategy for neural networks models, Comput. Chem. Eng. 16 (12) (1992) 1039–1057.
[6] Ch. Venkateswarlu, K. Venkat Rao, Dynamic recurrent radial basis function network model predictive control of unstable nonlinear processes, Chem. Eng. Sci. 60 (2005) 6718–6732.
[7] J.J. Song, S. Park, Neural model predictive control for nonlinear chemical processes, J. Chem. Eng. Jpn. 26 (4) (1993) 347–354.
[8] H. Zhao, J. Guiver, R. Neelakantan, L.T. Biegler, A nonlinear industrial model predictive controller using integrated PLS and neural net state-space model, Control Eng. Pract. 9 (2001) 125–133.
[9] Z.K. Nagy, Model based control of a yeast fermentation bioreactor using optimally designed artificial neural networks, Chem. Eng. J. 127 (1–3) (2007) 95–109.
[10] M. Nikravesh, A.E. Farell, T.G. Stanford, Control of nonisothermal CSTR with time varying parameters via dynamic neural network control (DNNC), Chem. Eng. J. 76 (2000) 1–16.
[11] M. Azlan Hussain, Review of the applications of neural networks in chemical process control—simulation and online implementation, Artif. Intell. Eng. 13 (1999) 55–68.
[12] J.C. Gomez, A. Jutan, E. Baeyens, Wiener model identification and predictive control of a pH neutralization process, IEE Proc. Control Theory Appl. 151 (3) (2004).
[13] S.J. Norquay, A. Palazoglu, J.A. Romagnoli, Model predictive control based on Wiener models, Chem. Eng. Sci. 53 (1) (1998) 75–84.
[14] A.L. Cervantes, O.E. Agamennoni, J.L. Figueroa, A nonlinear model predictive control system based on Wiener piecewise linear models, J. Process Control 13 (2003) 655–666.
[15] G.B. Sentoni, L.T. Biegler, J.B. Guiver, H. Zhao, State-space nonlinear process modeling: identification and universality, AIChE J. 44 (10) (1998) 2229–2239.
[16] B.G. Jeong, K.Y. Yoo, H.K. Rhee, Nonlinear model predictive control using a Wiener model of a continuous methyl methacrylate polymerization reactor, Ind. Eng. Chem. Res. 40 (25) (2001) 5968–5977.
[17] D. Westwick, M. Verhaegen, Identifying MIMO Wiener system using subspace model identification methods, Signal Process. 52 (2) (1996) 235–258.
[18] H.H.J. Bloemen, C.T. Chou, T.J.J. van den Boom, V. Verdult, M. Verhaegen, T.C. Backx, Wiener model identification and predictive control for dual composition control of a distillation column, J. Process Control 11 (6) (2001) 601–620.
[19] R. Aguilar, A. Poznyak, R. Martinez-Guerra, R. Maya-Yescas, Temperature control in catalytic cracking reactors via a robust PID controller, J. Process Control 12 (2002) 695–705.
[20] C.G. Economou, M. Morari, Internal model control. 5. Extension to nonlinear systems, Ind. Eng. Chem. PDD 25 (2) (1986) 403–411.
[21] R.K. Pearson, M. Pottmann, Gray-box identification of block-oriented nonlinear models, J. Process Control 10 (2000) 301–315.
[22] M. Verhaegen, Identification of the deterministic term of MIMO state space models given in innovation form from input–output data, Automatica 30 (1) (1994) 61–74.
[23] M. Verhaegen, Identification of the temperature-product quality relationship in a multi-component distillation column, Chem. Eng. Commun. 163 (1998) 111–132.
[24] B. Haverkamp, State Space Identification: Theory and Practice, PhD thesis, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands, 2001.
[25] V. Sima, Fast numerical algorithm for Wiener systems identification, in: International Working Conference on Analysis and Optimization of Differential systems, Romania, September 10–14, 2003, pp. 375–386.
[26] C.T. Kelley, Iterative Methods for Optimization, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1999.
[27] Y. Zhu, Multivariable System Identification for Process Control, An Imprint of Elsevier Science, Pergamon, 2001.
[28] W.L. Layben, Plant Wide Dynamic Simulators in Chemical Processing and Control, Marcell Dekker Inc., 2002.
[29] O.T. Berglihn, A toolbox for using MATLAB as an activeX/COM controller for Hysys, Matlab Central, http://www.pvv.org/∼olafb/hysyslib/, 1999.