

Nonlinear Model Predictive Controller Design for Identified Nonlinear Parameter Varying Model

Jiangan Lu*, Jie You, Qinmin Yang

State Key Laboratory of Industrial Control Technology, Department of Control Science and Engineering,
Zhejiang University, Hangzhou 310027, China
E-mail: jglu@iipc.zju.edu.cn*

Abstrak

Paper ini mengajukan suatu pengendali prediktif model tak-linier (MPC) yang baru berdasarkan model perubahan parameter taklinier (NPV) yang teridentifikasi. Pertama-tama, suatu skema model NPV dipresentasikan untuk identifikasi proses yang ditunjukkan dengan struktur model hibrid tak-linier Hammerstein parameter model yang bervariasi. Model hibrid Hammerstein menggabungkan jaringan saraf tiruan statik yang dinormalisasi dengan suatu fungsi transfer linier untuk mengidentifikasi sistem tak-linier secara umum pada setiap titik kerja tetap. Filosofi interpolasi model digunakan untuk memperoleh model global di seluruh domain operasi. Model NPV mempertimbangkan ketaklinieran dinamis transisi yang disebabkan oleh variasi titik kerja dan pemetaan tak-linier dari masukan ke keluaran pada titik kerja yang tetap. Di bawah kerangka kerja NPV yang baru, pengendalian dihitung melalui metoda linierisasi langkah jamak yang ditujukan untuk problem optimasi taklinier. Dalam skema yang diajukan, hanya pengujian berbiaya rendah yang diperlukan untuk pengenalan sistem dan pengendali dapat memperoleh kinerja keluaran yang lebih baik daripada metoda MPC yang berdasar model perubahan parameter linier (LPV). Contoh perhitungan numerik mengesahkan keefektifan pendekatan yang diajukan.

Kata kunci: perubahan parameter tak-linier (NPV), model Hammerstein, MPC tak-linier

Abstract

In this paper, a novel nonlinear model predictive controller (MPC) is proposed based on an identified nonlinear parameter varying (NPV) model. First, an NPV model scheme is present for process identification, which is featured by its nonlinear hybrid Hammerstein model structure and varying model parameters. The hybrid Hammerstein model combines a normalized static artificial neural network with a linear transfer function to identify general nonlinear systems at each fixed working point. Meanwhile, a model interpolating philosophy is utilized to obtain the global model across the whole operation domain. The NPV model considers both the nonlinearity of transition dynamics due to the variation of the working-point and the nonlinear mapping from the input to the output at fixed working points. Moreover, under the new NPV framework, the control action is computed via a multistep linearization method aimed for nonlinear optimization problems. In the proposed scheme, only low cost tests are needed for system identification and the controller can achieve better output performance than MPC methods based on linear parameter varying (LPV) models. Numerical examples validate the effectiveness of the proposed approach.

Keywords: Nonlinear parameter varying (NPV), Hammerstein model, nonlinear MPC.

1. Introduction

During last decades, model predictive control (MPC) has gained great success in a wide range of industrial applications. In most of those practices, MPC is designed based on linear models. However, linear MPC often results in poor performance when dealing with highly nonlinear processes. Meanwhile, although the nonlinear MPC can offer the potentials for improved performance, the main challenge is the high cost of modeling and identification of nonlinear process. Therefore, a more effective and efficient nonlinear identification technology for process prediction and optimization is crucial to the development of nonlinear MPC methodologies.

In terms of model identification of nonlinear systems, a promising method is to divide the system into a static nonlinear part and a linear dynamic component, so that the Hammerstein- model [2] or the Wiener-model [3] can be used to depict it. They have been

widely used in process control [2], [3] due to their simple structure and effective prediction capabilities. Especially, the two tools can provide a solution for time-invariant nonlinear systems.

However, when coping with a system with parameter varying property, the above methods will encounter difficulties while exciting the plant to perform thorough system identification tests across the whole scope of operation due to large disturbances and production loss. Hence, parameter varying process identification has attracted many attentions from both academia and industry [11]. Some results have been proposed in literature to discuss the input-output linear parameter varying (IO-LPV) methods, most of which are based on parameter interpolation techniques [11], [13], [14]. Another feasible approach is provided by interpolating the linear models [5]. That is to say, the global LPV model is retrieved by interpolating all local linear models with proper interpolating functions [5]-[8], [12]. According to literature [12], the model interpolating based input-output LPV (MI-IO-LPV) can achieve better results by approximately representing the process behavior in a thin envelop covering its operating trajectory. However, when dealing with highly nonlinear processes, LPV methods have difficulties to maintain the performance demands due to that the internal nonlinearities are ignored by using simple linear structures.

Therefore, in this paper, an identification method using nonlinear parameter varying (NPV) model frame is introduced to represent nonlinear processes. The NPV model is featured by its nonlinear hybrid Hammerstein model structure along with varying parameters. The hybrid Hammerstein model combines a normalized static artificial neural network with a linear transfer function to identify general nonlinear systems at each fixed working point. Meanwhile, a model interpolating philosophy is utilized to obtain the global model across the whole operation domain. Moreover, a nonlinear MPC law is developed based on the NPV model, which is computed via a multistep linearization method. The contribution of this paper can be summarized as: (1) providing an NPV model identification method which is low cost and reliable; (2) developing a nonlinear MPC algorithm for NPV plants which addresses both input and output constraints.

2. The Frame of the NPV Model

The NPV model is characterized by its nonlinear hybrid Hammerstein model structure along with varying parameters. This is not only because the nonlinear hybrid Hammerstein model [9] can represent complicated nonlinear processes accurately, but also it is very convenient for the nonlinear predictive controller designs.

Given a single-input single-output (SISO) nonlinear system, denote the input as $U(t)$ at time t and output as $Y(t)$. The nonlinear hybrid Hammerstein model consists of two parts: a normalized static nonlinear function and a linear transfer function. The nonlinear static nonlinear function $f(\cdot)$ is represented by the normalized direct linear feedback (DLF) neural network, which is showed in Figure 1.

Assume that input and output data are generated by a sampled NPV system:

$$\begin{aligned} Y(t) &= G(q, \delta(t)) \times X(t) + v(t) \\ &= G(q, \delta(t)) \times f[\delta(t), U(t)] + v(t) \end{aligned} \quad (1)$$

where,

$$G(q, \delta(t)) = \frac{B(q, \delta)}{A(q, \delta)} = \frac{[b_1(\delta)q^{-1} + \dots + b_n(\delta)q^{-n}]q^{-d}}{1 + a_1(\delta)q^{-1} + \dots + a_n(\delta)q^{-n}} \quad (2)$$

is the transfer function from $X(t)$ to $Y(t)$. $X(t)$ is a middle variable. q^{-1} denotes the unit delay operator, n is the model order. d is the delay from the input to the output. $f[\delta(t), U(t)]$ is a normalized static nonlinear function of the variable $\delta(t)$, which is called the working point variable (scheduling variable). It is a measured variable from the process or can be calculated from measurable process variables. Examples of working points are load of a power plant, and product grade of a polymer unit [6].

2.1. Static direct linear feedback (DLF) neural network

The static nonlinear DLF network model is given as:

$$X(t) = W_F U(t) + N_{BP}(U(t)) \quad (3)$$

where W_F is a constant matrix, $N_{BP}(\cdot)$ is a multilayer feedforward Back Propagation (BP) artificial neural network, and $X(t)$ is a middle variable. In fact, DLF network is composed of a linear mapping W_F and a nonlinear one $N_{BP}(\cdot)$.

DLF network is well known for its universal approximation property. Furthermore, it is very convenient to be integrated into the design of nonlinear predictive control [9]. Hence, a linear prediction controller can be easily extended to a nonlinear counterpart. On the other hand, the computing burden of a nonlinear predictive controller can be greatly reduced.

Remark 1: The static normalized nonlinear function $f[\delta(t), U(t)]$ can also be represented by other universal nonlinear function approximations, such as CMAC, RBF NN, splines, fuzzy logic, ANN and etc. [15], [16], although DLF neural network is utilized in this paper for demonstration purpose.

2.2. Linear transfer function

The dynamics of the industrial process is represented by a linear transfer function, which is widely adopted in the literature [6]. Several linear identification methods can be utilized to identify the unknown parameters within the linear block. In this paper, the asymptotic method [10] (ASYM) is used to obtain an unbiased model with low order.

Remark 2: At each fixed working point, an individual local nonlinear hybrid Hammerstein model is identified. On the contrary, LPV methods [6], [12] only build a simple linear model at a working point, and hence ignore the nonlinear mapping from the input to the output at fixed working points. However, our NPV model considers both the nonlinearity of transition dynamics due to the variation of the working-point and the nonlinear mapping from the input to the output at fixed working points.

3. NPV Model Identification

3.1. Model identification at each working point

A hybrid model is identified using the data set achieved at each working point. Denote the parameter vector of the model $G^\delta(q)$ as,

$$[a_1(\delta), \dots, a_n(\delta), b_1(\delta), \dots, b_n(\delta)] \quad (4)$$

Without loss of generality, assume that the process has three working points at

$$\delta_1 < \delta_2 < \delta_3$$

Denote the three identified nonlinear hybrid working-point models as,

$$\text{When } \delta = \delta_1 \quad \begin{aligned} X_1(t) &= W_{F1} U_1(t) + N_{BP1}(U_1(t)) \\ Y_1(t) &= G^{\delta_1}(q) X_1(t) \end{aligned} \quad (5a)$$

$$\text{When } \delta = \delta_2 \quad \begin{aligned} X_2(t) &= W_{F2} U_2(t) + N_{BP2}(U_2(t)) \\ Y_2(t) &= G^{\delta_2}(q) X_2(t) \end{aligned} \quad (5b)$$

$$\text{When } \delta = \delta_3 \quad \begin{aligned} X_3(t) &= W_{F3} U_3(t) + N_{BP3}(U_3(t)) \\ Y_3(t) &= G^{\delta_3}(q) X_3(t) \end{aligned} \quad (5c)$$

The identification training methods [9] to a nonlinear hybrid Hammerstein model can be used here.

3.2. Nonlinear static compensation by triangular interpolation

When system identification is undertaken for most industrial processes, it is usually not feasible or even possible to measure all input and output static data set for training the DLF network along the whole operating trajectory. That is to say, due to economic considerations, no steady-state tests in transition periods are available for model identification purpose. Therefore, before implementing any model based controllers, it is required to expand the local Hammerstein models at certain number of working points to the whole operation domain. In this paper, the triangular interpolation method is utilized to estimate the variation of model due to varying working point.

Firstly, the middle variable at an arbitrary working point $\delta(t)$ is denoted as follows:

$$\begin{aligned}\hat{X}(t) = & \phi_1(\delta)[W_{F1}U(t) + N_{BP1}(U(t))] \\ & + \phi_2(\delta)[W_{F2}U(t) + N_{BP2}(U(t))] \\ & + \phi_3(\delta)[W_{F3}U(t) + N_{BP3}(U(t))]\end{aligned}\quad (6)$$

where $\phi_1(\delta)$, $\phi_2(\delta)$, $\phi_3(\delta)$ are the weight functions. Essentially, they determine how much the system model at current working condition is close to that at individual working point. By recalling the triangular interpolation method, the weights are set to be the distances between the current working-point and the working point of the DLF network, i.e.

$$\phi_1(\delta) = \begin{cases} 1 & \delta < \delta_1 \\ \frac{\delta_2 - \delta}{\delta_2 - \delta_1} & \delta_1 \leq \delta \leq \delta_2 \\ 0 & \delta > \delta_2 \end{cases}\quad (7a)$$

$$\phi_2(\delta) = \begin{cases} 0 & \delta < \delta_2 \\ \frac{\delta - \delta_1}{\delta_2 - \delta_1} & \delta_1 \leq \delta \leq \delta_2 \\ \frac{\delta_3 - \delta}{\delta_3 - \delta_2} & \delta_2 < \delta \leq \delta_3 \\ 0 & \delta > \delta_3 \end{cases}\quad (7b)$$

$$\phi_3(\delta) = \begin{cases} 0 & \delta < \delta_2 \\ \frac{\delta - \delta_2}{\delta_3 - \delta_2} & \delta_2 \leq \delta \leq \delta_3 \\ 1 & \delta > \delta_3 \end{cases}\quad (7c)$$

Therefore, the middle variable across the entire working domain can be calculated.

3.3. Obtain nonlinear parameter varying model by interpolation

Similar to LPV model identification processes [6], instead of identifying a full nonlinear model in Eq. (1) and (2), an approximation model is built to represent the process along the operating-trajectory as follows:

$$\begin{aligned}Y(t) = & \eta_1(\delta)[G^{\delta_1}(q)\hat{X}(t)] + \eta_2(\delta)[G^{\delta_2}(q)\hat{X}(t)] \\ & + \eta_3(\delta)[G^{\delta_3}(q)\hat{X}(t)] + v(t)\end{aligned}\quad (8)$$

Several estimation methods [5]-[8] can be used here to parameterize the weight functions, $\eta_1(\delta)$, $\eta_2(\delta)$ and $\eta_3(\delta)$: cubic splines, polynomials, or piece-wise linear function. In our work, the cubic splines are taken.

Firstly, denote a set of knots $\{p_1, p_2, \dots, p_s\}$ for a working-point variable $\delta(t)$. The knots should span the whole process operation range. It is reported convenient [6] to distribute the

knots uniformly over the range $[\delta_{\min}, \delta_{\max}]$. A better option is to use the working points at which comparatively accurate models are available. Besides, the knots must be real numbers and satisfy the following inequality:

$$\delta_{\min} = p_1 < p_2 < \dots < p_s = \delta_{\max} \quad (9)$$

Thereafter, each cubic spline weight function $\eta(\delta)$ can be given as

$$\eta(\delta) = \lambda_1 + \lambda_2 \delta + \sum_{j=2}^{s-1} \lambda_{j+1} |\delta - p_j|^3 \quad (10)$$

where $[\lambda_1, \lambda_2, \dots, \lambda_s]$ are the parameters to be estimated. s is the order of cubic splines, which depends on the number of working points and the amount of total data.

Now, assume that all weight functions $\eta_1(\delta)$, $\eta_2(\delta)$ and $\eta_3(\delta)$ are written as Eq. (10). Then the parameter vector of the weighting functions can be denoted as:

$$\hat{\theta} = [\lambda_1^{\delta_1}, \lambda_2^{\delta_1}, \dots, \lambda_s^{\delta_1}, \lambda_1^{\delta_2}, \lambda_2^{\delta_2}, \dots, \lambda_s^{\delta_2}, \lambda_1^{\delta_3}, \lambda_2^{\delta_3}, \dots, \lambda_s^{\delta_3}]^T \quad (11)$$

Define the total data sets as follows:

$$Z^N = [U(t), Y(t), \delta(t), t = 1, 2, \dots, N] \quad (12)$$

Moreover, denote the output error of model in Eq. (8) as:

$$e_{OE}(t) = Y(t) - [\eta_1(\delta)\hat{Y}_1(t) + \eta_2(\delta)\hat{Y}_2(t) + \eta_3(\delta)\hat{Y}_3(t)] \quad (13)$$

where $\hat{Y}_i(t) = G^{\delta_i}(q)\hat{X}(t)$, $i = 1, 2, 3$, and $\hat{\theta}$ is the parameter vector to be determined. Thereafter, by using total testing data which include working-points data and transition test data, the parameter vector can be estimated by minimizing the output error loss function:

$$\hat{\theta} = \min_{\theta} \sum_{t=1}^N [e_{OE}(t)]^2 \quad (14)$$

4. Nonlinear MPC using the NPV model

4.1. Process Model

Assume that inputs and output data are generated by a sampled NPV system in Eq. (1) and (2). Then a state-space description of the NPV model can be derived as follows:

$$\begin{aligned} x(t+1) &= Ax(t) + B(w)\hat{X}(t) \\ Y(t) &= Cx(t) \end{aligned} \quad (15)$$

In order to imitate practical applications, step-like disturbance at the output, slow drifts or step-like disturbance at the input or states, and plant-model mismatch have to be introduced. The resulting augmented model for the NPV model [6] can be written as follows:

$$\begin{aligned} \begin{bmatrix} x(t+1) \\ d(t+1) \\ p(t+1) \end{bmatrix} &= \begin{bmatrix} A & B_d & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} x(t) \\ d(t) \\ p(t) \end{bmatrix} + \begin{bmatrix} B(\delta) \\ 0 \\ 0 \end{bmatrix} \hat{X}(t) + \varepsilon(t) \\ Y(t) &= [C \quad 0 \quad C_p] \begin{bmatrix} x(t) \\ d(t) \\ p(t) \end{bmatrix} + v(t) \end{aligned} \quad (16)$$

where $\varepsilon(t)$ and $v(t)$ are zero-mean white noise with specified covariance R_ε and R_v , $p(t)$ are augmented output disturbance variables, $d(t)$ are augmented input disturbance variables, B_d and C_p is the disturbance model.

$$A = \begin{bmatrix} A^{\delta_1} & & & & \\ & A^{\delta_2} & & & \\ & & A^{\delta_3} & & \\ & & & \ddots & \\ & & & & A^{\delta_n} \end{bmatrix}, \quad B(\delta) = \begin{bmatrix} B^{\delta_1} \\ B^{\delta_2} \\ B^{\delta_3} \\ \vdots \\ B^{\delta_n} \end{bmatrix}$$

$$C = [C^{\delta_1} \quad C^{\delta_2} \quad C^{\delta_3} \quad \dots \quad C^{\delta_n}], \quad C^{\delta_i} = [0 \quad 0 \quad 0 \quad \dots \quad 1]_{1 \times n_i}$$

$$A^{\delta_i} = \begin{bmatrix} 0 & 0 & \dots & 0 & -a_n(\delta_i) \\ 1 & 0 & \dots & 0 & -a_{n-1}(\delta_i) \\ 0 & 1 & \dots & 0 & -a_{n-2}(\delta_i) \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & -a_1(\delta_i) \end{bmatrix}_{n_i \times n_i}, \quad B^{\delta_i} = \begin{bmatrix} \eta_i(\delta)b_n(\delta_i) \\ \eta_i(\delta)b_{n-1}(\delta_i) \\ \dots \\ \eta_i(\delta)b_2(\delta_i) \\ \eta_i(\delta)b_1(\delta_i) \end{bmatrix}_{n_i \times 1}$$

4.2. Control Objective

The nonlinear MPC controller is based on the minimization of the following open-loop quadratic objective performance index:

$$\min_{\Delta \hat{X}(t|t), \dots, \Delta \hat{X}(t+M-1|t)} J(t) = \sum_{i=1}^P \|Y_r - Y(t+i|t)\|_Q^2 + \sum_{i=0}^M \|\Delta \hat{X}(t)(t+i|t)\|_R^2$$

$$s.t. \quad x(t+j+1|t) = Ax(t+j|t) + B(\delta)\hat{X}(t+j|t) + B_d d(t|t)$$

$$y(t+j|t) = Cx(t+j|t) + C_p p(t|t)$$

$$\Delta \hat{X}_{\min} \leq \Delta u(t+j|t) \leq \Delta \hat{X}_{\max} \quad (j=0, \dots, M-1)$$

$$\hat{X}_{\min} \leq u(t+j|t) \leq \hat{X}_{\max} \quad (j=0, \dots, M-1)$$

$$Y_{\min} \leq y(t+j|t) \leq Y_{\max} \quad (j=0, \dots, P)$$
(17)

where y_r defines a reference trajectory for the outputs; \hat{X}_{\min} and \hat{X}_{\max} are lower and upper operating limits for \hat{X} ; Y_{\min} and Y_{\max} are lower and upper operating for Y . P is a prediction horizon and M is a control horizon. $\Delta \hat{X}(t+j|t) = \hat{X}(t+j|t) - \hat{X}(t+j-1|t)$ is often referred as control move suppression to prevent aggressive control actions. Current intermediate control can be calculated by the following formula:

$$\hat{X}(t) = \hat{X}(t-1) + \Delta \hat{X}(t) \quad (18)$$

4.3. Derivation of actual nonlinear control input

The above Eq. (18) obtains optimization intermediate control $\hat{X}(t)$ which ensures the output of plant tracks the reference trajectory. But it is not the actual input. As can be seen in Figure 1, the relationship between intermediate control $\hat{X}(t)$ and actual input $U(t)$ is described by the static DLF network in Eq. (6).

This paper uses the following approximating formula (in this case, the sampling period should not be too large). Eq. (6) can be approximated:

$$\hat{X}(t) = \phi_1(\delta)[W_{F1}U(t) + N_{BP1}(U(t-1))] \\ + \phi_2(\delta)[W_{F2}U(t) + N_{BP2}(U(t-1))] \\ + \phi_3(\delta)[W_{F3}U(t) + N_{BP3}(U(t-1))] \quad (19)$$

Then the actual input can be obtained:

$$U(t) = (\phi_1(\delta)W_{F1} + \phi_2(\delta)W_{F2} + \phi_3(\delta)W_{F3}) \times [\hat{X}(t) - N_{BP1}(U(t-1)) - N_{BP2}(U(t-1)) - N_{BP3}(U(t-1))] \quad (20)$$

When it is necessary, the iteration correction of $U(t)$ can be denoted as:

$$U^{(k)}(t) = (\phi_1(\delta)W_{F1} + \phi_2(\delta)W_{F2} + \phi_3(\delta)W_{F3}) \times [\hat{X}(t) - N_{BP1}(U^{(k-1)}(t)) - N_{BP2}(U^{(k-1)}(t)) - N_{BP3}(U^{(k-1)}(t))] \quad (21)$$

where k indicates the number of iteration. For the first time, $U^{(k-1)}(t)$ can be obtained by Eq. (20). The number of iteration depends on the actual situation.

4.4. Multistep Linearization Method

For a multi-step-ahead control, however, the linear model may significantly deteriorate from the nonlinear process and therefore negatively influence the controller performance. This can be overcome by using multiple linear models derived along the nominal trajectory within the prediction horizon. This process is called multistep linearization method.

Control Algorithm.

- **Step1:** Initialization $\hat{X}(t) = \hat{X}_0$.
- **Step2:** Starting with the state estimation value $x(t|t)$ and the nominal input trajectory vector $\hat{X}(t)$, compute the nominal trajectory vector Y and W . Get the linearized models $\{G(t), G(t+1), \dots, G(t+P-1)\}$ around the nominal trajectory.
- **Step3:** Solve the corresponding QP problem. Update $\hat{X}(t) \leftarrow \hat{X}(t-1) + L\Delta\hat{X}$, L is a triangular matrix with elements of one on and below the diagonal.
- **Step4:** If the trajectory $\hat{X}(t)$ is converged or the QP iteration counter exceeds n_{\max} , then go to Step5. Else, go back to Step2 with iteration count $i = i + 1$.
- **Step5:** Implement the first computed move $\Delta\hat{X}(t|t)$ and update $\hat{X}(t)$ for the next time interval.
- **Step6:** Derivation of actual nonlinear control input

$$U^{(k)}(t) = (\phi_1(\delta)W_{F1} + \phi_2(\delta)W_{F2} + \phi_3(\delta)W_{F3}) \times [\hat{X}(t) - N_{BP1}(U^{(k-1)}(t)) - N_{BP2}(U^{(k-1)}(t)) - N_{BP3}(U^{(k-1)}(t))] , \text{ then go back to Step1 at time } t \leftarrow t + 1 .$$

5. Simulations

Given an S type nonlinear parameter varying function, the steady-state shape can be seen in the Figure 2.

$$x = f(u, \delta) = \frac{1}{1 + \delta^3 * e^{-(4u-10)}}, \quad \delta \in [1, 4] \quad (22)$$

We take a first order process as the linear transfer function. The transfer function in the continuous-time is

$$G(s, \delta) = \frac{Y(s, \delta)}{X(s, \delta)} = \frac{K(\delta)}{\chi(\delta)s + 1} \quad (23)$$

where $K(\delta) = 1 + \delta^2$, $\mu(\delta) = 3 + \delta^2$, $\delta \in [1, 4]$

Then, the simulated output is corrupted by a filtered white noise as:

$$v(t) = \frac{c}{1 - 0.9z^{-1}} e(t) \tag{24}$$

where $e(t)$ is a white noise sequence and the constant c is adjusted at the three working points so that the noise is 3% of the noise-free output in power. In order to obtain nonlinear hybrid working-point models, the process will be tested at three working points:

$$\delta_1 = 1, \delta_2 = 2.5, \delta_3 = 4$$

In the operation range $\delta \in [1, 4]$, the normalized static nonlinear mapping from the input to the output have an obvious nonlinear alter and process gain changes nonlinearly more than 10 times and time constant changes 10 times in the linear transfer function. Thus a linear model cannot obtain good approximation of the process behavior in the whole operation trajectory.

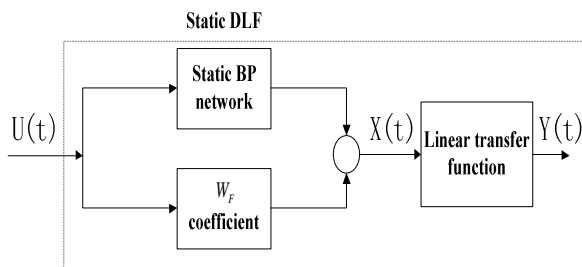


Figure 1. The architecture of a nonlinear hybrid Hammerstein model

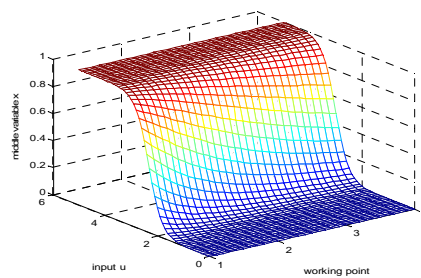


Figure 2. The normalized static nonlinear mapping from the input to the output using the DLF network

Here, we will show how well the proposed nonlinear parameter varying model can approximate the process over the whole operation trajectory. For generating input-output data, the process is simulated at a sampling time of 1 second. The input is a GBN (generalized binary noise) signal with average switch time of 20 seconds. The knots for the three weighting functions are the same and the order of cubic spline is 7.

The estimated normalized nonlinear mapping from the input to the output and step responds of NPV models at the working points 2 and 3.3 respectively are shown in Figure 3 and Figure 4. The weights functions of DLF networks and NPV models are shown in Figure 5. One can see that good control performance of nonlinear MPC using the NPV model can be obtained by comparisons with linear MPC and LPV-MPC [8] in Figure 6 and Figure 7.

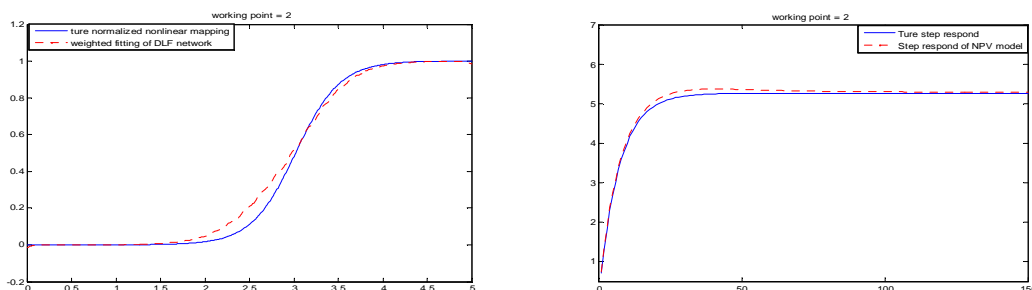


Figure 3. The normalized nonlinear mapping from the input to the output and step responses of NPV models at the working points 2. Left blue solid lines: true normalized nonlinear mapping; Left red dotted lines: weighted fitting of DLF network. Right blue solid lines: true step responds; Right red dotted lines: output of NPV model.

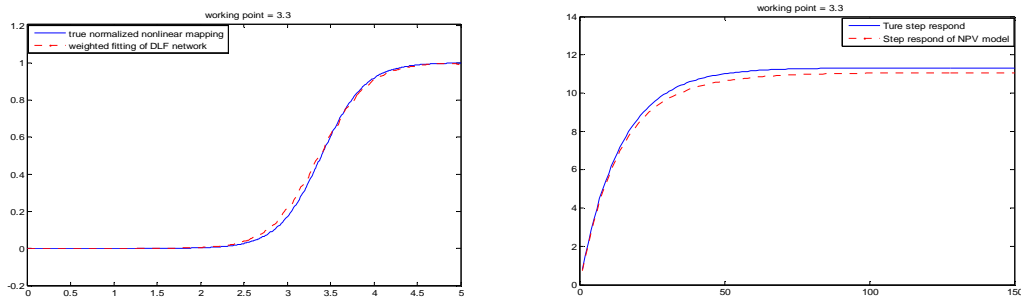


Figure 4. The normalized nonlinear mapping from the input to the output and step responses of NPV models at the working points 3.3. Left blue solid lines: true normalized nonlinear mapping; Left red dotted lines: weighted fitting of DLF network. Right blue solid lines: true step responds; Right red dotted lines: output of NPV model.

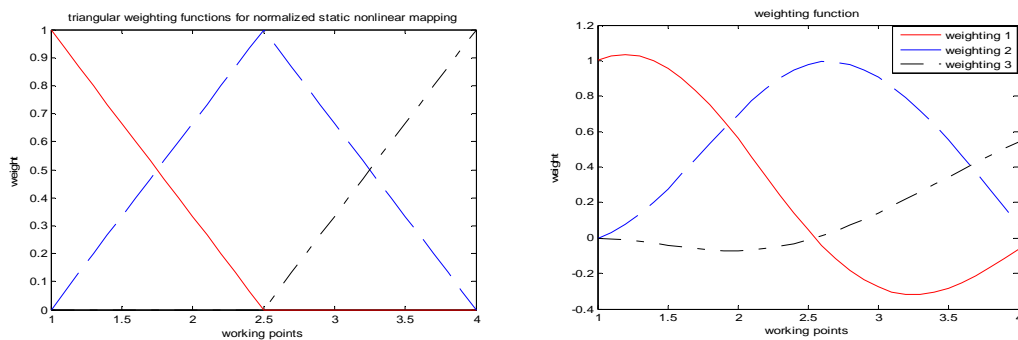


Figure 5. Left picture is triangular weighting functions for normalized static nonlinear mapping. Right picture is weighting functions of the nonlinear parameter varying model using noise data. Red solid line: weighting 1; blue dashed line: weighting 2; black dash dotted line: weighting 3

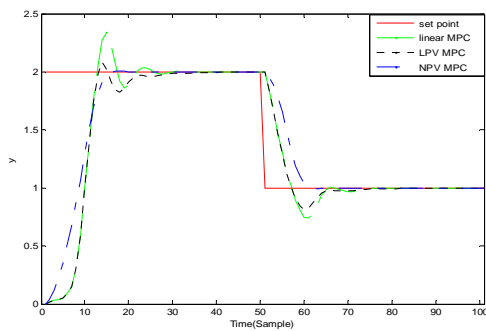


Figure 6. Output comparisons of linear MPC, LPV-MPC and NPV-MPC, Red solid line: set point. Cyan dashed line: linear MPC. Black dotted line: LPV-MPC. Blue dash dotted line: NPV-MPC.

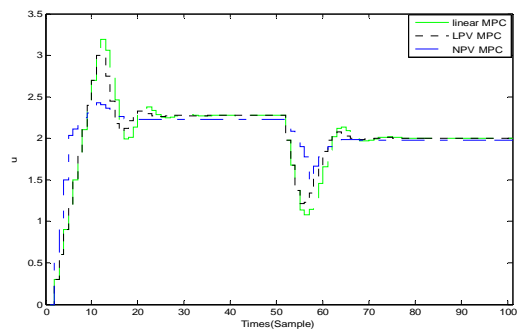


Figure 7. Input comparisons of linear MPC, LPV-MPC and NPV-MPC

6. Conclusions

In this paper, an NPV model identification method is proposed to represent the nonlinear process. It consists of local nonlinear hybrid Hammerstein model structure and varying model parameters. Further, a nonlinear MPC design is developed based on the identified NPV model. The control action is computed via a multistep linearization method of nonlinear optimization problem. Simulation examples demonstrate the results of model identification and the control performance of nonlinear MPC.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (NSFC)(NO. 21076179 and 61104008), and the National Basic Research Program of China (973 Program: 2012CB720500)

References

- [1] Palma F D, Magni L A. Multi-model Structure for Model Predictive Control. *Annual Reviews in Control*. 2004; 28: 47–52.
- [2] Perez E, Blasco X, Garcia-Nieto S and Sanchis J. Diesel Engine Identification and Predictive Control using Wiener and Hammerstein Models. In: *IEEE International Conference on Control Applications, CCA*. 2006: 2417–2423.
- [3] Gomez JC, Jutan A, Baeyens E. Wiener Model Identification and Predictive Control of a pHNeutralisation Process. *Proceedings of the IEE Control Theory and Applications*. 2004: 151: 329–338.
- [4] Akesson B M, Toivonen H T, Waller J B, and Nystrom R H. Neural Network Approximation of a Nonlinear Model Predictive Controller Applied to a pHNeutralization Process. *Computers and Chemical Engineering*. 2005; 29: 323–335.
- [5] Zhu YC, et. al. Multivariable process identification for MPC: the asymptotic method and its applications. *Journal of Process Control*. 1998; 8(2): 101-115.
- [6] Zuhua Xu, Zhu Y C, et. al. Nonlinear MPC using an Identified LPV Model. *Ind. Eng. Chem. Res*. 2009; 48: 3043-3051.
- [7] Zhu YC, et. al. A Method of LPV Model identification for control. *The international Federation of Automatic Control*. Seoul, Korea. July 2008: 6-11.
- [8] Zhu Y C, et. al. Identification and MPC control of a circulation Fluidized Bed Boiler using an LPV Model. *Dynamics and Control of Process Systems*. Leuven, Belgium. July 2010: 5-7.
- [9] Huang Daoping et. al. Multivariable Nonlinear Predictive Control Based on an Integrating Model: *Control Theory and Applications*. 1999; 16(1): 38-42.
- [10] Zhu YC. Multi variable System Identification for Process Control. Endhoven, Pergamon, Chapter 5, 6
- [11] V Laurain, M Gilson, R Tóth, H Garnier. Refined instrumental variable methods for identification of LPV Box–Jenkins models. *Automatica*. 2010; 46(6): 959–967.
- [12] Yu Zhao et. al. Prediction error method for identification of LPV models. *Journal of process control*. 2011; 22(1): 180-193.
- [13] M. Butcher, A. Karimi. Data-driven tuning of linear parameter-varying precompensators. *Int. J. Adapt. Control Signal Process*. 2010; 24(7): 592–609.
- [14] M. Butcher, A. Karimi, R. Longchamp. On the consistency of certain identification methods for linear parameter varying systems. in *17th IFAC World Congress*. Seoul, Korea. July 6–11 2008: 4018–4023.
- [15] Mardiyono, Reni Suryanita, Azlan Adnan. Intelligent Monitoring System on Prediction of Building Damage Index using Neural- Network. *TELKOMNIKA*. 2012; 10(1):155-164.
- [16] D Harikrishna, N V Srikanth. Dynamic Stability Enhancement of Power Systems Using Neural-Network Controlled Static-Compensator. *TELKOMNIKA*. 2012; 10(1): 9-16.