

Chapter 2

Identification and Control of Nonlinear Systems Using a Piecewise-Linear Hammerstein Model

Gregor Dolanc and Stanko Strmčnik

2.1 Introduction

This chapter addresses the problem of identification and control of a special class of nonlinear processes whose dynamics can be approximated by a Hammerstein model. A Hammerstein model consists of a serial connection of a static nonlinear function and a linear dynamic transfer function. These models are very relevant to practice since many industrial processes exhibit this type of nonlinear dynamic behaviour. Both identification and control design for this class of processes has therefore attracted considerable attention from researchers in academia as well as from practitioners in industry.

Identification has been addressed by many authors and a rich set of methods for parameter estimation of the nonlinear and linear part of the model have been proposed. Examples include iterative methods [28, 30] and non-iterative over-parameterisation methods [7, 19, 25]. The least squares method is usually employed to estimate the model parameters [21], although other approaches are also used, e.g. instrumental variables [31] and the maximum likelihood method [8, 15].

The nonlinear static part of the Hammerstein model was originally proposed in the form of a polynomial function. However, models with various other representations were studied as well, e.g. models with two-segment piecewise-linear nonlinearity [24], cubic splines [36], preload nonlinearity [25], two-segment polynomial nonlinearities [33], discontinuous asymmetric nonlinearities [32], hysteresis [20] and Bezier functions [18]. These methods assume that input signals used for the identification ensure persistent excitation, which means that signals are distributed over the entire range of operation. A popular choice for excitation signals is white noise or related random inputs, although simpler waveforms like the random phase multisine signals proposed in [9] and [10] can also be used. The main problem with such signals is that they cannot always be applied to industrial processes, or their

G. Dolanc (✉) · S. Strmčnik

Department of Systems and Control, Jožef Stefan Institute, Ljubljana, Slovenia

use is even prohibited because of strict technological limitations as well as system performance requirements.

All the mentioned methods are parametric methods. Nonparametric methods represent an alternative approach that has also been used to represent and identify the Hammerstein model [6, 14].

The use of a Hammerstein model for control is also widely discussed in the literature. A common approach is to adopt the existing linear controller design, such as [2, 3, 22, 34, 35], where the linear pole placement method was adapted in various ways. A similar approach can be found in [27], where a generalised minimum variance controller is accommodated to the single polynomial-based Hammerstein model. In [17] the utilisation of the Hammerstein model is proposed in a way such that the nonlinearity is approximated by the Bezier function. Several other control laws based on the Hammerstein model are also discussed in the literature, e.g. dead-beat control [29], adaptive dead-beat feedforward compensation of measurable disturbances [5], indirect adaptive control based on linear quadratic control and an approximation of the nonlinear static function using neural networks [23], nonlinear dynamic compensation with passive nonlinear dynamics [16], etc. Another possibility is to use the Hammerstein model within predictive control laws, e.g. [1, 13], which has gained in popularity, not only in research, but also in industrial practice.

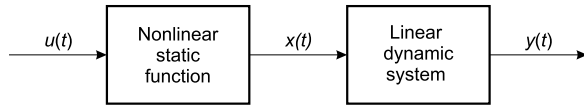
The problem with the majority of the methods mentioned above is that constraints and limitations encountered in the commissioning and operation stage are largely ignored in the design stage. A serious issue is the fact that a high level of expertise is needed to put the controllers mentioned above to work, particularly during the commissioning and tuning stage. Additional problems may arise due to the high computational load and limited freedom in selecting the excitation signals. In order to accommodate these issues, we present and demonstrate an approach to identification and control of nonlinear processes of the Hammerstein type, based on piecewise-linear approximation of the static nonlinear function.

This chapter is organised as follows. First, we will review the original form of the Hammerstein model and briefly highlight the shortcomings that limit its practical applicability. Based on this, we will introduce a new form of the Hammerstein model with modified parameterisation, which will eliminate the main practical limitations inherent in the original formulation of the Hammerstein model. Next, we will propose a parameter estimation algorithm, accommodated to the proposed model structure. Finally, we will present a novel pole placement controller, tuned according to the identified model parameters. The usability of the identification and control algorithms will be demonstrated by a simulation example and experimental application on a sintering process.

2.2 Original Form of the Hammerstein Model and Its Limitations

The Hammerstein model belongs to the class of block-oriented nonlinear models, which can be decomposed into nonlinear static blocks and linear dynamic blocks.

Fig. 2.1 Structure of the Hammerstein model



In the case of a single input, single output Hammerstein model, the nonlinear static function is followed by a linear dynamic system, as follows from Fig. 2.1. If the sequence of blocks is reversed, we get a Wiener model.

The nonlinear static function is originally proposed in the form of a polynomial function, while the linear dynamic system is assumed to be either a linear discrete time or a continuous time transfer function. The internal signal x , which links the nonlinear static function and the linear dynamic system, is assumed to be non-measurable. Consequently, the parameters of the nonlinear static function and the linear dynamic system cannot be estimated separately.

The original form of the Hammerstein model has several practical limitations:

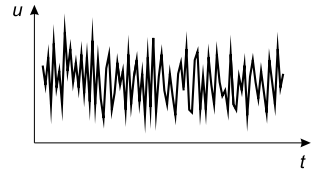
1. Parameter estimation is very sensitive to the type of excitation signal. If the excitation signal is limited within a narrow interval, the model will properly predict the process output only if inputs are from this interval. Elsewhere, model predictions might be quite poor. The original Hammerstein model thus requires excitation signals distributed over the entire range of operation. This is a serious drawback, since the application of such signals may often be prohibitive in real processes due to various technological limitations.
2. Polynomial representation of the input nonlinearity of the original Hammerstein model does not enable the approximation of discontinuous processes; however, such processes appear quite frequently in practical applications.
3. If the original form of the Hammerstein model is integrated into a control law, the calculation of the control signal usually requires inversion of the polynomial equation, which can in general only be done numerically. This inversion has to be repeated in every control interval, which leads to a high computational load.

To alleviate these drawbacks, we propose a modified model structure, the main idea of which is to use a piecewise-linear function to represent the model nonlinearity.

2.3 The Piecewise-Linear Hammerstein Model

The model and the essentials of the associated parameter estimation algorithm were presented in [11]. The idea was to use piecewise-linear representation of the nonlinear static function of the model. It should be noted that the idea of using piecewise representations is not new, different kinds of piecewise representations have been used in the Hammerstein model [9, 10, 36]. But this was usually motivated by achieving more accurate approximation of the nonlinear static function compared to that obtained by continuous functions (e.g. single polynomials). Our motivation for

Fig. 2.2 An example of rich excitation signal



using piecewise representation is different, we want to improve the practical applicability of the model. We will show that using the piecewise-linear approximation directly reduces practical limitations presented in the previous section.

First, owing to the piecewise-linear representation of the static nonlinearity, identification does not require rich excitation signals over the entire range of operation, as presented in Fig. 2.2. This kind of signal can cause the process to go out of control or can even cause damage to the process.

Identification can be performed in the presence of more realistic, temporarily bounded signals. Here we mean signals which can be expressed as a sum of two components: a slow varying component and a fast varying one. The first component can be slowly increasing or decreasing signal, e.g. a ramp function (Fig. 2.3, left). The second component is bounded on an interval which is significantly narrower than the entire region of the input signal. For example, it can be implemented in terms of sequence of pulses (Fig. 2.3, centre). The sum of both components is shown in Fig. 2.3, right.

Such kinds of signals are much more likely to be acceptable for applications in industrial processes than classical persistent excitation waveforms. Since the amplitude is temporarily bounded within a narrow range, it means that only a tiny section of the nonlinear static function will be excited at that time. Piecewise-linear representation of the static nonlinearity and the corresponding identification algorithm, which will be presented below, allow for identification of the excited section only, while keeping unexcited sections unchanged.

The second benefit of a piecewise-linear representation of the nonlinear static function is the possibility to account for the discontinuous static functions as well as static functions with a discontinuous first derivative.

And third, the problem of the computational burden of the inversion of the nonlinear static function is completely circumvented since the piecewise-linear function has a very simple analytical inverse, and thus requires only a minimum computational effort during each control interval of the controller.

These advantages become extremely important when practical applications of the control algorithm are considered.

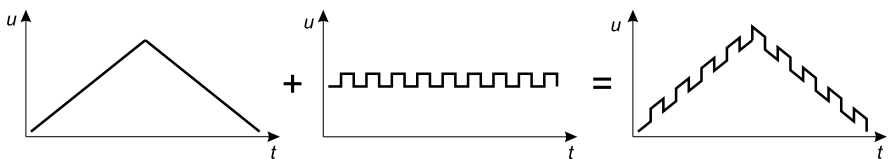


Fig. 2.3 Realistic signal with temporarily bounded amplitude

2.3.1 Piecewise-Linear Functions

A general nonlinear static function can be approximated by a piecewise-linear function [26], which is composed of a number of line segments connected to each other

$$x(u) = \mathbf{l}(u, \mathbf{u})^T \cdot \mathbf{x} \quad (2.1)$$

In (2.1) u is input to the nonlinear static function and x is an output. The function is defined by vectors \mathbf{u} and \mathbf{x} , which determine the positions of the joints of line segments:

$$\mathbf{x} = [x_0 \quad x_1 \quad \dots \quad x_j \quad \dots \quad x_m]_{((m+1) \times 1)}^T \quad (2.2)$$

$$\mathbf{u} = [u_0 \quad u_1 \quad \dots \quad u_j \quad \dots \quad u_m]_{((m+1) \times 1)}^T \quad (2.3)$$

The vector \mathbf{x} contains x -coordinates of joints while vector \mathbf{u} contains u -coordinates, which are called knots. Knots have to be arranged in a monotonically increasing order

$$u_0 < u_1 < \dots < u_j < u_{j+1} < \dots < u_m \quad (2.4)$$

Furthermore, in (2.1) $\mathbf{l}(u, \mathbf{u})$ is a vector of “tent functions”

$$\mathbf{l}(u, \mathbf{u}) = [l_0 \quad l_1 \quad \dots \quad l_j \quad \dots \quad l_m]_{((m+1) \times 1)}^T \quad (2.5)$$

Hereinafter, instead of $\mathbf{l}(u, \mathbf{u})$, the shorter denotation $\mathbf{l}(u)$ will be used. The elements of vector $\mathbf{l}(u)$ are defined as follows:

$$l_0(u) = \begin{cases} \frac{u_1 - u}{u_1 - u_0} & \text{if } u_0 \leq u < u_1 \\ 0 & \text{if } u_1 \leq u \leq u_m \end{cases} \quad (2.6)$$

$$l_j(u) = \begin{cases} 0 & \text{if } u_0 \leq u < u_{j-1} \\ \frac{u - u_{j-1}}{u_j - u_{j-1}} & \text{if } u_{j-1} \leq u < u_j \\ \frac{u_{j+1} - u}{u_{j+1} - u_j} & \text{if } u_j \leq u < u_{j+1} \\ 0 & \text{if } u_{j+1} \leq u \leq u_m \end{cases} \quad j = 1 \dots m - 1 \quad (2.7)$$

$$l_m(u) = \begin{cases} 0 & \text{if } u_0 \leq u < u_{m-1} \\ \frac{u - u_{m-1}}{u_m - u_{m-1}} & \text{if } u_{m-1} \leq u \leq u_m \end{cases} \quad (2.8)$$

It can be seen that the vector $\mathbf{l}(u)$ contains only two nonzero elements for any value u . Their position and values depend on the amplitude of the input signal u , as follows from Eqs. (2.6)–(2.8) The situation is illustrated in Fig. 2.4.

The inversion of a piecewise-linear function results in another piecewise-linear function, where vectors \mathbf{x} and \mathbf{u} exchange roles

$$u = \mathbf{l}(x, \mathbf{x})^T \cdot \mathbf{u} \quad (2.9)$$

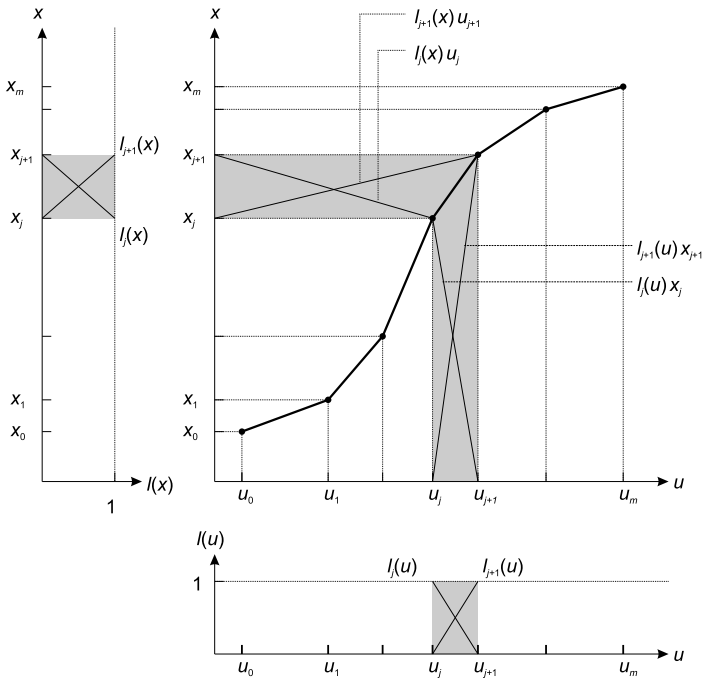


Fig. 2.4 Parameterisation of the piecewise-linear function ($u_j \leq u < u_{j+1}$)

The piecewise-linear function $x(u)$ is always continuous and contains discontinuities of the first derivative, which are located in knots. Therefore, it is possible to approximate nonlinear functions with the discontinuous first derivative as well as the discontinuous nonlinear functions. In the first case, the position of the discontinuity of the first derivative and the position of the arbitrary knot should match as closely as possible. In the second case, a position of discontinuity u_d of the nonlinear function has to be surrounded by two knots ($u_d - \Delta u_1$) and ($u_d + \Delta u_2$), where Δu_1 and Δu_2 represent small distances from the point of discontinuity, as follows from Fig. 2.5.

2.3.2 Parameterisation of the Hammerstein Model with Piecewise-Linear Functions

Now, let us merge the piecewise-linear function and the linear dynamic system of the model. First, let us assume the classical structure of the Hammerstein model (Fig. 2.1), where u is the input and x is output of the nonlinear static function. Simultaneously, x is the input and y is the output of the linear dynamic system. The linear dynamic system is described by the discrete time difference equation

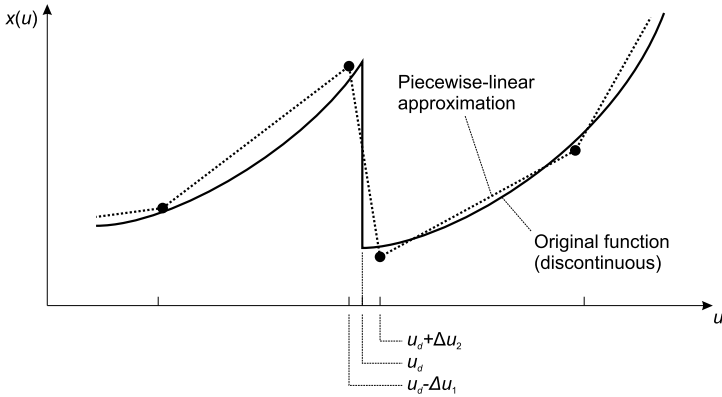


Fig. 2.5 Approximation of the discontinuous static function

$$\begin{aligned} y(k) + a_1 y(k-1) + a_2 y(k-2) + \dots + a_n y(k-n) \\ = b_0 x(k-d) + b_1 x(k-d-1) + b_n x(k-d-n) \end{aligned} \quad (2.10)$$

where n is the order and d is the delay of the linear dynamic system. Let the output x from the nonlinear static function be expressed using the piecewise-linear function (2.1). The terms $b_i x(k-d-i)$, appearing on the right-hand side of Eq. (2.10), can then be expressed as

$$b_i x(k-d-i) = b_i \mathbf{I}(u(k-d-i))^T \mathbf{x} = \mathbf{I}(u(k-d-i))^T b_i \mathbf{x}, \quad i = 0 \dots n \quad (2.11)$$

If the rightmost term of Eq. (2.11) is put into Eq. (2.10) and if $y(k)$ is expressed explicitly, the following discrete time difference equation is obtained, which represents the piecewise-linear Hammerstein model:

$$\begin{aligned} y(k) = -a_1 y(k-1) - a_2 y(k-2) - \dots - a_n y(k-n) + \mathbf{I}(u(k-d))^T b_0 \mathbf{x} \\ + \mathbf{I}(u(k-d-1))^T b_1 \mathbf{x} + \dots + \mathbf{I}(u(k-d-n))^T b_n \mathbf{x} \end{aligned} \quad (2.12)$$

Equation (2.12) is multilinear in parameters and can be arranged in the following vector form:

$$y(k) = \boldsymbol{\psi}^T(k) \boldsymbol{\theta} \quad (2.13)$$

In Eq. (2.13) $\boldsymbol{\psi}$ is the data vector and $\boldsymbol{\theta}$ is the parameter vector structured as follows:

$$\boldsymbol{\psi} = \begin{bmatrix} -y(k-1) & -y(k-2) & \dots & -y(k-n) & \mathbf{I}(u(k-d))^T \\ \mathbf{I}(u(k-d-1))^T & \mathbf{I}(u(k-d-n))^T & & & \end{bmatrix}_{((n+(n+1)(m+1)) \times 1)}^T \quad (2.14)$$

$$\boldsymbol{\theta} = \begin{bmatrix} a_1 & a_2 & \dots & a_n & b_0 \mathbf{x}^T & b_1 \mathbf{x}^T & \dots & b_n \mathbf{x}^T \end{bmatrix}_{((n+(n+1)(m+1)) \times 1)}^T \quad (2.15)$$

In the data vector, the terms $\mathbf{I}(u(k-d-i))$, $i = 0 \dots n$, are vectors of the “tent functions” in particular time instances, defined by Eqs. (2.5), (2.6), (2.7) and (2.8). The structure of elements $b_i \mathbf{x}$, $i = 0 \dots n$, of the parameter vector $\boldsymbol{\theta}$ is the following:

$$\begin{aligned} b_i \mathbf{x} &= b_i [x_0 \quad x_1 \quad \dots \quad x_m]^T = [b_i x_0 \quad b_i x_1 \quad \dots \quad b_i x_m]^T \\ &= [bx_{i,0} \quad bx_{i,1} \quad \dots \quad bx_{i,m}]_{(m+1) \times 1}^T = \mathbf{b}\mathbf{x}_i \end{aligned} \quad (2.16)$$

In Eq. (2.16) the products $b_i x_j$ represent “linear parameters”, denoted as $bx_{i,j}$, $i = 0 \dots n$, $j = 0 \dots m$.

$$bx_{i,j} = b_i x_j \quad (2.17)$$

They are called linear since in the model they appear in linear combination with the data. The linear parameters $bx_{i,j}$ can be arranged in subvectors $\mathbf{b}\mathbf{x}_i$, $i = 0 \dots n$ of the data vector $\boldsymbol{\theta}$. Considering this, $\boldsymbol{\theta}$ can be rewritten in terms of “linear parameters”

$$\begin{aligned} \boldsymbol{\theta} &= [a_1 \quad a_2 \quad \dots \quad a_n \mid bx_{0,0} \quad bx_{0,1} \quad \dots \quad bx_{0,m} \mid bx_{1,0} \quad bx_{1,1} \quad \dots \quad bx_{1,m} \mid \\ &\quad bx_{n,0} \quad bx_{n,1} \quad \dots \quad bx_{n,m}]^T \\ &= [a_1 \quad a_2 \quad \dots \quad a_n \mid \mathbf{b}\mathbf{x}_0^T \quad \mathbf{b}\mathbf{x}_1^T \quad \dots \quad \mathbf{b}\mathbf{x}_n^T]^T \end{aligned} \quad (2.18)$$

The identification algorithm, which will be presented below, will estimate the “linear parameters”. The set of “linear parameters” should be distinguished from the set of “basic parameters”, which are

$$a_1 \quad a_2 \quad \dots \quad a_n \quad b_0 \quad b_1 \quad b_2 \quad \dots \quad b_n \quad x_0 \quad x_1 \quad x_2 \quad \dots \quad x_m \quad (2.19)$$

If the “basic parameters” are known, then the “linear parameters” can be uniformly calculated. On the other hand, if the “linear parameters” are known, then the “basic parameters” cannot be uniformly calculated. If the nonlinear static function is multiplied by a nonzero real constant c , and if the linear dynamic part is divided by the same constant, the resulting model has the same input-output behaviour. This redundancy of “basic parameters” can be resolved, for example, by fixing the static gain of the linear dynamic part. This means that the following equality must hold for the parameters of the linear dynamic part, if the static gain is fixed to unity:

$$\sum_{i=0}^n b_i = 1 + \sum_{i=1}^n a_i \quad (2.20)$$

By summing Eq. (2.17) for $i = 0 \dots n$, we get

$$x_j = \frac{\sum_{i=0}^n bx_{i,j}}{\sum_{i=0}^n b_i}, \quad j = 0 \dots m \quad (2.21)$$

In Eq. (2.21) parameters b_i are unknown. The sum of parameters b_i , $i = 0 \dots n$, can be expressed as a sum of parameters a_i , $i = 1 \dots n$, using Eq. (2.20), since

parameters a_i appear explicitly in the vector of the “linear parameters” θ

$$x_j = \frac{\sum_{i=0}^n b x_{i,j}}{1 + \sum_{i=1}^n a_i}, \quad j = 0 \dots m \quad (2.22)$$

After parameters x_j are estimated, also parameters b_i can be estimated using the equalities derived from Eq. (2.17)

$$b_i = \frac{b x_{i,0}}{x_0} = \dots = \frac{b x_{i,j}}{x_j} = \dots = \frac{b x_{i,m}}{x_m}, \quad i = 0 \dots n \quad (2.23)$$

A solution for b_i can be obtained using least squares

$$b_i = \frac{\sum_{j=0}^m (x_j \cdot b x_{i,j})}{\sum_{j=0}^m x_j^2}, \quad i = 0 \dots n \quad (2.24)$$

Alternatively, the basic parameters b_i and x_j can be estimated using an algorithm based on the singular value decomposition [4], which gives more general results but also requires more computational effort.

2.3.3 Redundancy of Linear Parameters

Redundancy of “linear parameters” is a special property of the piecewise-linear Hammerstein model which becomes important during parameter estimation. It should be distinguished from the already mentioned redundancy of the “basic parameters”. Let us assume that a process is described by the model with parameters θ defined in Eq. (2.18). In that case, there exists a model with parameters θ^* which has the same transfer function between u and y as the model with parameters θ

$$\theta^* = [a_1 \ a_2 \ \dots \ a_n \ | \ b x_{0,0}^* \ b x_{0,1}^* \ \dots \ b x_{0,m}^* \ | \ b x_{1,0}^* \ b x_{1,1}^* \ \dots \ b x_{1,m}^* \ | \ b x_{n,0}^* \ b x_{n,1}^* \ \dots \ b x_{n,m}^*]^T \quad (2.25)$$

The subvectors $\mathbf{b}x_i^*$ of θ^* differ from the subvectors $\mathbf{b}x_i$ of θ . The relation between $\mathbf{b}x_i$ and $\mathbf{b}x_i^*$, $i = 0 \dots n$ reads

$$\begin{aligned} \mathbf{b}x_i^* &= [b x_{i,0}^* \ b x_{i,1}^* \ \dots \ b x_{i,m}^*]^T \\ &= [b x_{i,0} \ b x_{i,1} \ \dots \ b x_{i,m}]^T + c_i \mathbf{1}_{((m+1) \times 1)} \end{aligned} \quad (2.26)$$

Finally, the model with parameters θ^* and the model with parameters θ have identical transfer functions if

$$\sum_{i=0}^n c_i = 0 \quad (2.27)$$

To prove this, the difference $\Delta y(k)$ between the output $y^*(k)$ of the model with parameters θ^* and the output $y(k)$ of the nominal model with parameters θ has to be calculated

$$\Delta y(k) = y^*(k) - y(k) = \boldsymbol{\psi}^T(k)\boldsymbol{\theta}^* - \boldsymbol{\psi}^T(k)\boldsymbol{\theta} = \boldsymbol{\psi}^T(k)(\boldsymbol{\theta}^* - \boldsymbol{\theta}) \quad (2.28)$$

Furthermore, the difference $(\boldsymbol{\theta}^* - \boldsymbol{\theta})$ can be expressed as

$$\boldsymbol{\theta}^* - \boldsymbol{\theta} = [\Delta \mathbf{a}^T \quad \Delta \mathbf{b}\mathbf{x}_0^T \quad \Delta \mathbf{b}\mathbf{x}_1^T \quad \dots \quad \Delta \mathbf{b}\mathbf{x}_n^T]^T \quad (2.29)$$

According to the definition of $\boldsymbol{\theta}^*$, it follows that

$$\Delta \mathbf{a} = \mathbf{0}_{(n \times 1)} \quad (2.30)$$

and

$$\Delta \mathbf{b}\mathbf{x}_i = \mathbf{b}\mathbf{x}_i^* - \mathbf{b}\mathbf{x}_i = c_i \mathbf{1}_{((m+1) \times 1)}, \quad i = 0 \dots n \quad (2.31)$$

By accounting for Eqs. (2.30) and (2.31) in Eq. (2.28), $\Delta y(k)$ can be expressed

$$\Delta y(k) = \boldsymbol{\psi}^T(k) \cdot (\boldsymbol{\theta}^* - \boldsymbol{\theta}) = \sum_{i=1}^n 0 \cdot y(k-i) + \sum_{i=0}^n \mathbf{1}(u(k-d-i))^T \cdot c_i \mathbf{1}_{((m+1) \times 1)} \quad (2.32)$$

According to the definition of the ‘‘tent functions’’ $\mathbf{1}$, in Eqs. (2.5)–(2.8) the right-most term of Eq. (2.32) can be expressed as

$$\begin{aligned} & \mathbf{1}(u(k-d-i))^T \cdot c_i \mathbf{1}_{((m+1) \times 1)} \\ &= \left[\mathbf{0}_{1 \times j} \left(\frac{u_{i+1} - u(k-d-i)}{u_{i+1} - u_i} \right) \left(\frac{u(k-d-i) - u_i}{u_{i+1} - u_i} \right) \mathbf{0}_{1 \times (m-j-1)} \right] \\ & \cdot c_i \mathbf{1}_{((m+1) \times 1)} = c_i \end{aligned} \quad (2.33)$$

If this is put into Eq. (2.32), the difference $\Delta y(k)$ can be expressed as

$$\Delta y(k) = \sum_{i=1}^n 0 \cdot y(k-i) + \sum_{i=0}^n c_i \quad (2.34)$$

The first sum of the right side of Eq. (2.34) equals zero. Because of the required condition (2.27) also the second sum of the right side of Eq. (2.34) equals zero, which means that $\Delta y(k) = 0$. This proves that the model with parameters $\boldsymbol{\theta}^*$ has an identical transfer function to the model with parameters $\boldsymbol{\theta}$.

2.3.4 Active and Inactive Parameters

There is another important property of the proposed piecewise-linear Hammerstein model. At any time instance the output $y(k)$ only depends on some of the model

parameters. The output always depends on $a_i, i = 1 \dots n$. In addition, it also depends on two parameters from each subvector $\mathbf{b}\mathbf{x}_i, i = 0 \dots n$, which are multiplied by the two nonzero elements of $\mathbf{I}(u(k-d-i))$. Let the parameters on which the model output depends be referred to as “active parameters”. The rest are called “inactive parameters”. Let p be a position index defined as

$$u_{j-1} \leq u(k-d-i) < u_j \quad \Rightarrow \quad p(k-d-i) = j \quad (2.35)$$

The “active parameters” within subvector $\mathbf{b}\mathbf{x}_i$ can then be expressed as

$$bx_{i,p(k-d-i)-1} \quad \text{and} \quad bx_{i,p(k-d-i)}, \quad i = 0 \dots n \quad (2.36)$$

2.4 Identification of Model Parameters

The parameters of the proposed piecewise-linear Hammerstein model can be identified by many standard methods since the model in Eq. (2.13) is linear in parameters. Online methods, simple to implement, are of primary interest. The well known recursive least squares method (RLS) with forgetting factor λ is appropriate and summarised below:

$$y_e(k) = \boldsymbol{\psi}^T(k)\boldsymbol{\theta}(k-1) \quad (2.37)$$

$$e(k) = y(k) - y_e(k) \quad (2.38)$$

$$\boldsymbol{\theta}(k) = \boldsymbol{\theta}(k-1) + \Delta\boldsymbol{\theta}(k) \quad (2.39)$$

$$\Delta\boldsymbol{\theta}(k) = \mathbf{K}(k) \cdot e(k) \quad (2.40)$$

$$\mathbf{K}(k) = \frac{\mathbf{P}(k-1) \cdot \boldsymbol{\psi}(k)}{\lambda + \boldsymbol{\psi}^T(k) \cdot \mathbf{P}(k-1) \cdot \boldsymbol{\psi}(k)} \quad (2.41)$$

$$\mathbf{P}(k) = \left[\mathbf{P}(k-1) - \frac{\mathbf{P}(k-1) \cdot \boldsymbol{\psi}(k) \cdot \boldsymbol{\psi}^T(k) \cdot \mathbf{P}(k-1)}{\lambda + \boldsymbol{\psi}^T(k) \cdot \mathbf{P}(k-1) \cdot \boldsymbol{\psi}(k)} \right] \cdot \frac{1}{\lambda} \quad (2.42)$$

In principle, this method is convenient for identification of the piecewise-linear Hammerstein model after the following modifications have been implemented:

- management of active and inactive parameters;
- compensation of parameter offset;
- tracking the inactive parameters.

2.4.1 Management of Active and Inactive Parameters

In Sect. 2.3.4 it was shown that the model output $y(k)$ only depends on “active parameters”. Since the remaining “inactive parameters” have no influence on the

model output, there is no information available to update their values. The identification algorithm should therefore be modified in such a way as to stop the updating of “inactive parameters” and to restart updating parameters in the transition from “inactive” to “active”. If the identified process is assumed to be time variant, there may be a need to rapidly update the parameters immediately after they become “active”.

To stop identification of a particular parameter in vector θ (to “freeze” its value), a corresponding element (gain) in vector \mathbf{K} in Eq. (2.41) has to be set to zero. This is achieved by setting the corresponding row and column of matrix \mathbf{P} to zero. To restart identification of the particular parameter in θ , the corresponding element (gain) in vector \mathbf{K} should be set to some high value. This is achieved by setting the corresponding diagonal element of matrix \mathbf{P} to some high value. A way to achieve the required modifications of matrix \mathbf{P} is to perform the following transformation:

$$\mathbf{P}_m(k-1) = \mathbf{A}_m \cdot \mathbf{P}(k-1) \cdot \mathbf{A}_m + \mathbf{B}_m \quad (2.43)$$

and then use $\mathbf{P}_m(k-1)$ instead of $\mathbf{P}(k-1)$ in Eqs. (2.41) and (2.42). Both matrices \mathbf{A}_m and \mathbf{B}_m are diagonal:

$$\mathbf{A}_m = \text{diag}(\alpha) \quad (2.44)$$

$$\mathbf{B}_m = \text{diag}(\beta) \quad (2.45)$$

The structures of the vectors α and β are

$$\alpha = [\alpha_a^T \quad \alpha_{bx0}^T \quad \alpha_{bx1}^T \quad \dots \quad \alpha_{bxn}^T]^T \quad (2.46)$$

$$\beta = [\beta_a^T \quad \beta_{bx0}^T \quad \beta_{bx1}^T \quad \dots \quad \beta_{bxn}^T]^T \quad (2.47)$$

α_a and β_a correspond to the continuously identified parameters a_i , $i = 1 \dots n$. Thus, both vectors are constants:

$$\alpha_a = \mathbf{1}_{(n \times 1)} \quad (2.48)$$

$$\beta_a = \mathbf{0}_{(n \times 1)} \quad (2.49)$$

Other elements correspond to the parameters in vectors \mathbf{bx}_i , $i = 0 \dots n$. For each \mathbf{bx}_i , two cases are possible depending on the parameter state (active or inactive):

- (a) there is no change of state within \mathbf{bx}_i with respect to the previous step
 $p(k-d-i) = p(k-d-i-1)$

$$\alpha_{bxi} = [\mathbf{0}_{(1 \times (p(k-d-i)-1))} \quad [1 \quad 1] \quad \mathbf{0}_{(1 \times (m-p(k-d-i)))}]_{((m+1) \times 1)}^T \quad (2.50)$$

$$\beta_{bxi} = \mathbf{0}_{((m+1) \times 1)}$$

(b) there is a change of state within \mathbf{bx}_i with respect to the previous step $p(k-d-i) \neq p(k-d-i-1)$.

$$\begin{aligned} \alpha_{bxi} &= \mathbf{0}_{(m+1) \times 1} \\ \beta_{bxi} &= rval \cdot \left[\mathbf{0}_{(1 \times (p(k-d-i)-1))} \quad [1 \quad 1] \quad \mathbf{0}_{(1 \times (m-p(k-d-i)))} \right]_{(m+1) \times 1}^T \end{aligned} \quad (2.51)$$

In case a), the combination of α_{bxi} and β_{bxi} allows the identification of “active parameters” within \mathbf{bx}_i only. In case b), the combination of α_{bxi} and β_{bxi} restarts the identification of these parameters within bx_i that become “active”. This is achieved by setting the corresponding elements of β_{bxi} to a high value $rval$ (for example $rval = 10^5$).

2.4.2 Compensation of the Parameter Offset

In Sect. 2.3.3 it was shown that the model with parameters θ^* described by Eq. (2.25) has the same transfer function as the model with parameters θ described by Eq. (2.18) as long as condition (2.27) is fulfilled. This was called redundancy of the “linear parameters”. Consequently, the identification algorithm has no control over offsets c_i , $i = 0 \dots n$. There is no guarantee that the result of identification will be a model with a nominal parameter set θ with $c_i = 0$, $i = 0 \dots n$. On the contrary, a set of parameters θ^* with an arbitrary set of offsets fulfilling condition (2.27) can be the result of the identification.

In order to estimate the nominal parameters θ from the identified parameters θ^* a procedure is needed which estimates offsets c_i of \mathbf{bx}_i^* , $i = 0 \dots n$, and subtracts them from the parameters of \mathbf{bx}_i^* of the identified model θ^* . It has to be guaranteed that the procedure does not change the transfer function of the model. Before the introduction of the compensation procedure, some properties of the piecewise-linear Hammerstein model have to be discussed.

Assume that the “basic parameters” (2.19) of the piecewise-linear Hammerstein model are known. Then the particular subvectors of vector θ can be expressed according to Eq. (2.16). Assume also that offsets c_i , $i = 0 \dots n$, satisfying condition (2.27), are known. The subvectors of vector θ^* can then be expressed using Eq. (2.26). Now we can calculate the following quantities:

- average values of the parameters of particular subvectors \mathbf{bx}_i , $i = 0 \dots n$ of vector θ

$$\overline{bx_i} = \frac{1}{m+1} \sum_{j=0}^m bx_{i,j} = \frac{1}{m+1} \sum_{j=0}^m b_i x_j = b_i \frac{1}{m+1} \sum_{j=0}^m x_j = b_i \bar{x} \quad (2.52)$$

- average values of the parameters of particular subvectors \mathbf{bx}_i^* of vector θ

$$\begin{aligned}\overline{bx_i^*} &= \frac{1}{m+1} \sum_{j=0}^m bx_{i,j}^* = \frac{1}{m+1} \sum_{j=0}^m (b_i x_j + c_i) \\ &= b_i \frac{1}{m+1} \sum_{j=0}^m x_j + \frac{1}{m+1} (m+1)c_i = b_i \bar{x} + c_i\end{aligned}\quad (2.53)$$

- standard deviations of the parameters of particular subvectors \mathbf{bx}_i of vector θ

$$\begin{aligned}\sigma_{bx_i} &= \left[\frac{1}{m+1} \sum_{j=0}^m (\overline{bx_i} - bx_{i,j})^2 \right]^{\frac{1}{2}} \\ &= \left[\frac{1}{m+1} \sum_{j=0}^m (b_i \bar{x} - b_i x_j)^2 \right]^{\frac{1}{2}} \\ &= \left[b_i^2 \frac{1}{m+1} \sum_{j=0}^m (\bar{x} - x_j)^2 \right]^{\frac{1}{2}} = |b_i| \sigma_x\end{aligned}\quad (2.54)$$

- standard deviations of the parameters of particular subvectors \mathbf{bx}_i^* of vector θ^*

$$\begin{aligned}\sigma_{bx_i^*} &= \left[\frac{1}{m+1} \sum_{j=0}^m (\overline{bx_i^*} - bx_{i,j}^*)^2 \right]^{\frac{1}{2}} \\ &= \left[\frac{1}{m+1} \sum_{j=0}^m (b_i \bar{x} + c_i - b_i x_j - c_i)^2 \right]^{\frac{1}{2}} \\ &= \left[b_i^2 \frac{1}{m+1} \sum_{j=0}^m (\bar{x} - x_j)^2 \right]^{\frac{1}{2}} = |b_i| \sigma_x\end{aligned}\quad (2.55)$$

- sum of the average values of all \mathbf{bx}_i , $i = 0 \dots n$

$$\sum_{i=0}^n \overline{bx_i} = \sum_{i=0}^n b_i \bar{x} = \bar{x} \sum_{i=0}^n b_i\quad (2.56)$$

- sum of the average values of all \mathbf{bx}_i^* , $i = 0 \dots n$, by taking into account Eq. (2.27)

$$\sum_{i=0}^n \overline{bx_i^*} = \sum_{i=0}^n (b_i \bar{x} + c_i) = \sum_{i=0}^n b_i \bar{x} + \sum_{i=0}^n c_i = \sum_{i=0}^n b_i \bar{x} = \bar{x} \sum_{i=0}^n b_i\quad (2.57)$$

From the calculations above the following facts are evident:

$$\begin{aligned}\overline{bx_i^*} &= \overline{bx_i} + c_i \\ \sigma_{bx_i}^* &= \sigma_{bx_i} \\ \sum_{i=0}^n \overline{bx_i^*} &= \sum_{i=0}^n \overline{bx_i}\end{aligned}$$

Hence, a procedure for the estimation of the unknown nominal model with parameters θ from the identified model with parameters θ^* can be proposed. From Eqs. (2.52) and (2.57) an expression for the estimation of $\overline{bx_i}$ of the nominal model θ can be derived

$$\overline{bx_i} = \frac{b_i}{\sum_{i=0}^n b_i} \cdot \sum_{i=0}^n \overline{bx_i^*} \quad (2.58)$$

The unknown parameters b_i , $i = 0 \dots n$, appearing in Eq. (2.58) can be estimated from the calculated standard deviation of the parameters of the particular subvector \mathbf{bx}_i^* in vector θ^* . From (2.55) it follows that

$$|b_i| = \frac{\sigma_{bx_i}^*}{\sigma_x} \quad (2.59)$$

To estimate b_i , the sign(s) of b_i has to be determined. First define

$$s_x = \text{sign}(x_m - x_0) \quad (2.60)$$

$$s_{bx_i}^* = \text{sign}(bx_{i,m}^* - bx_{i,0}^*) \quad (2.61)$$

If Eqs. (2.26) and (2.16) are employed in Eq. (2.61), then it follows

$$\begin{aligned}s_{bx_i}^* &= \text{sign}((bx_{i,m} + c_i) - (bx_{i,0} + c_i)) = \text{sign}(bx_{i,m} - bx_{i,0}) \\ &= \text{sign}(b_i \cdot (x_m - x_0)) = \text{sign}(b_i) \cdot \text{sign}(x_m - x_0) = \text{sign}(b_i) \cdot s_x\end{aligned} \quad (2.62)$$

From Eq. (2.62) we get

$$\text{sign}(b_i) = \frac{s_{bx_i}^*}{s_x} \quad (2.63)$$

Now b_i can be completely estimated as follows:

$$b_i = \text{sign}(b_i) \cdot |b_i| = \frac{s_{bx_i}^*}{s_x} \cdot \frac{\sigma_{bx_i}^*}{\sigma_x} \quad (2.64)$$

Note that in Eq. (2.64) s_x and σ_x are unknown. If Eq. (2.64) is used in Eq. (2.58), then s_x and σ_x are cancelled and an expression for the estimation of the average

values of the parameters of the particular subvector \mathbf{bx}_i of the nominal model $\boldsymbol{\theta}$ is obtained

$$\overline{bx_i} = \frac{s_{bx_i}^* \sigma_{bx_i}^*}{\sum_{i=0}^n s_{bx_i}^* \sigma_{bx_i}^*} \cdot \sum_{i=0}^n \overline{bx_i^*}, \quad i = 0 \dots n \quad (2.65)$$

The average values $\overline{bx_i^*}$ and standard deviations $\sigma_{bx_i}^*$ in Eq. (2.65) are calculated using the leftmost terms of Eqs. (2.53) and (2.55), while signs s_{bx_i} are calculated using Eq. (2.61). By subtracting Eq. (2.52) from Eq. (2.53) the expression for the estimation of c_i is obtained

$$\tilde{c}_i = \overline{bx_i^*} - \overline{bx_i}, \quad i = 0 \dots n \quad (2.66)$$

The estimates of \mathbf{bx}_i can be calculated by subtracting estimates of c_i from the estimated \mathbf{bx}_i^*

$$bx_{i,j} = bx_{i,j}^* - \tilde{c}_i, \quad i = 0 \dots n, \quad j = 0 \dots m \quad (2.67)$$

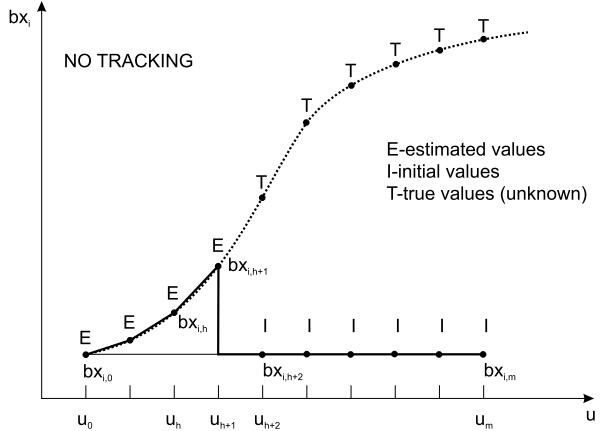
2.4.3 Tracking the Inactive Parameters

Assume that the excitation signal used for identification is a ramp function (or a similar signal with the amplitude gradually increasing with time) with an additive excitation component with bounded amplitude. Particular parameters of subvectors $\mathbf{bx}_i, i = 0 \dots n$ are then estimated consecutively as they interfere with the amplitude of the excitation signal. In the initial phase of identification it is therefore beneficial to introduce a mechanism which would use the already estimated parameters of \mathbf{bx}_i to improve the initial values of the parameters of \mathbf{bx}_i not yet estimated. The idea is as follows. If the nonlinear static function of the process is continuous, then the values of adjacent parameters of \mathbf{bx}_i are very likely close to each other. Setting the value of the parameter not yet estimated close to the value of the closest estimated parameter provides a better starting point for identification than the original initial value. A possible way to implement this idea is to introduce a mechanism of “tracking inactive parameters”. First, note that in each step of the proposed recursive identification algorithm subvectors $\mathbf{bx}_i, i = 0 \dots n$ are updated. Note also that only “active parameters” are changed in every \mathbf{bx}_i , while “inactive parameters” remain unchanged

$$\begin{aligned} \Delta \mathbf{bx}_i(k) = & \left[\mathbf{0}_{(1 \times (p(k-d-i)-1))} \quad \Delta bx_{i,p(k-d-i)-1}(k) \right. \\ & \left. \Delta bx_{i,p(k-d-i)}(k) \quad \mathbf{0}_{(1 \times (m-p(k-d-i)))} \right]^T \end{aligned} \quad (2.68)$$

In this sense, tracking can be interpreted as equalizing the change of the particular “inactive parameter” of $\Delta \mathbf{bx}_i(k)$ to the change of the closest “active parameter” of

Fig. 2.6 Tracking the inactive parameters not active



the same vector. This means that vector (2.68) has to be replaced with the following vector:

$$\Delta \mathbf{b}x_i(k)' = \left[s_1 \Delta bx_{i,p(k-d-i)-1}(k) \mathbf{1}_{(1 \times (p(k-d-i)-1))} \quad \Delta bx_{i,p(k-d-i)-1}(k) \right. \\ \left. \Delta bx_{i,p(k-d-i)}(k) \quad s_1 \Delta bx_{i,p(k-d-i)}(k) \quad \mathbf{1}_{(1 \times (m-p(k-d-i)))} \right]^T \quad (2.69)$$

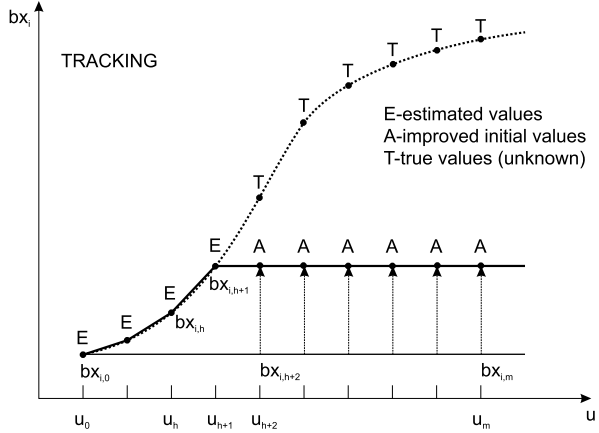
In Eq. (2.69) s_1 acts as a two-state switch to enable ($s_1 = 1$) or disable ($s_1 = 0$) the tracking mechanism. Note that if $s_1 = 0$, then Eq. (2.68) equals Eq. (2.69). The modified changes of θ considering the tracking mechanism are

$$\Delta \theta = \left[\Delta a_1 \quad \Delta a_2 \quad \dots \quad \Delta a_n \quad \Delta \mathbf{b}x_0'^T \quad \Delta \mathbf{b}x_1'^T \quad \dots \quad \Delta \mathbf{b}x_n'^T \right]^T \quad (2.70)$$

It is obvious that tracking takes effect only on “inactive parameters”. Tracking is not a mandatory procedure but it can speed up convergence of the parameters during the initial phase of identification, especially if the initial values of the parameters are only poor estimates of the true values. The effect of tracking is illustrated in Figs. 2.6 and 2.7, which present the parameters of vector $\mathbf{b}x_i$ in the initial stage of identification.

First, consider the situation without tracking (Fig. 2.6), where it is assumed that parameters $bx_{i,j}, j = 0 \dots h + 1$ have already been estimated, while parameters $bx_{i,j}, j = h + 2 \dots m$ remain at their initial values since they have not yet interfered with the excitation signal. Then, consider the same situation but with tracking enabled (Fig. 2.7). It can be seen that the values of parameters $bx_{i,j}, j = h + 2 \dots m$ are changed in parallel with the closest active parameter (in this particular case $bx_{i,h+1}$). Thus, improved initial estimates were obtained, which are obviously closer to the true values than the original initial values. Due to the much better starting point, the convergence of the parameters is accelerated. After estimates of all parameters have been obtained, and if the process is slowly time variant or invariant, the tracking has no considerable effect and can be switched off. If the process has a discontinuous

Fig. 2.7 Tracking the inactive parameters activated



static function at $u = u_d$ where ($u_j < u_d < u_{j+1}$), then the tracking procedure has to be rearranged

$$\Delta \mathbf{b}\mathbf{x}'_i(k) = \begin{cases} \text{diag}([\mathbf{1}_{1 \times (j+1)} \ \mathbf{0}_{1 \times (m-j)}])^T \cdot \Delta \mathbf{b}\mathbf{x}'_i(k) & \text{if: } u < u_j \\ \text{diag}([\mathbf{0}_{1 \times (j+1)} \ \mathbf{1}_{1 \times (m-j)}])^T \cdot \Delta \mathbf{b}\mathbf{x}'_i(k) & \text{if: } u > u_{j+1} \\ \text{diag}([\mathbf{0}_{1 \times (j+1)} \ \mathbf{0}_{1 \times (m-j)}])^T \cdot \Delta \mathbf{b}\mathbf{x}'_i(k) & \text{if: } u_j \leq u \leq u_{j+1} \end{cases} \quad (2.71)$$

and finally

$$\Delta \theta = [\Delta a_1 \quad \Delta a_2 \quad \dots \quad \Delta a_n \quad \Delta \mathbf{b}\mathbf{x}_0''^T \quad \Delta \mathbf{b}\mathbf{x}_1''^T \quad \dots \quad \Delta \mathbf{b}\mathbf{x}_n''^T]^T \quad (2.72)$$

2.5 Controller Design

In this section the proposed piecewise-linear Hammerstein model is utilised for control. We will begin with the design of a general linear controller and then modify this approach to be integrated with the piecewise-linear Hammerstein model as presented in [12].

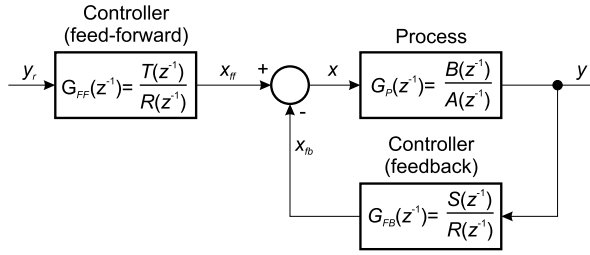
2.5.1 Linear Controller

First, let us recall the well-known general linear controller with two degrees of freedom, as shown in Fig. 2.8.

In Fig. 2.8, the transfer function G_P represents the process and is defined as

$$G_P(z^{-1}) = \frac{y(z^{-1})}{x(z^{-1})} = \frac{B(z^{-1})}{A(z^{-1})} \quad (2.73)$$

Fig. 2.8 General linear controller setup



The design goal is that the closed loop transfer function between y_r and y equals the desired closed loop transfer function G_D

$$G_D(z^{-1}) = \frac{y(z^{-1})}{y_r(z^{-1})} = \frac{B_D(z^{-1})}{A_D(z^{-1})} \quad (2.74)$$

The controller consists of two transfer functions, G_{FF} and G_{FB} , which are defined by the polynomials R , S and T :

$$G_{FF}(z^{-1}) = \frac{x_{ff}(z^{-1})}{y_r(z^{-1})} = \frac{T(z^{-1})}{R(z^{-1})}, \quad G_{FB}(z^{-1}) = \frac{x_{fb}(z^{-1})}{y(z^{-1})} = \frac{S(z^{-1})}{R(z^{-1})} \quad (2.75)$$

$$x(z^{-1}) = x_{ff}(z^{-1}) - x_{fb}(z^{-1})$$

The polynomials R , S and T should be designed in such a way that the closed-loop transfer function G_{CL}

$$G_{CL}(z^{-1}) = \frac{y(z^{-1})}{y_r(z^{-1})} = \frac{B(z^{-1})T(z^{-1})}{A(z^{-1})R(z^{-1}) + B(z^{-1})S(z^{-1})} \quad (2.76)$$

of the process and the controller equals the desired transfer function G_D (2.74). Polynomials R , S and T are designed by the pole placement method, details can be found in [3]. It is assumed that all zeros of the process G_P are minimum phase and well damped. Consequently, they can be cancelled within the closed-loop transfer function. To achieve cancellation, the controller polynomial R has to fulfil the following condition:

$$R = R_1 B \quad (2.77)$$

The two controller polynomials R_1 and S are used to match the closed-loop poles of the transfer function (2.76) to the desired closed-loop poles of the transfer function (2.74). To achieve this, the following equality, known as a Diophantine equation, must hold:

$$AR_1 + S = A_0 A_D \quad (2.78)$$

In Eq. (2.78) A_0 represents an observer polynomial that is a part of the controller but is cancelled within the closed-loop transfer function (2.76). To solve Eq. (2.78), one must first determine the orders of the polynomials R , S , T and A_0 , as well

as the orders of the polynomials of the desired closed-loop transfer function (2.74). Following the results in [3], the orders of the polynomials have to fulfil the following conditions:

$$\deg A_D - \deg B_D \geq \deg A - \deg B \quad (2.79)$$

$$\deg A_0 \geq 2 \deg A - \deg A_D - \deg B - 1 \quad (2.80)$$

$$\deg R_1 = \deg A_0 + \deg A_D - \deg A \quad (2.81)$$

$$\deg S < \deg A \quad (2.82)$$

Finally, by solving Eq. (2.78), polynomials R_1 and S are determined. Polynomial R is then determined using Eq. (2.77) and polynomial T using Eq. (2.83)

$$T = B_D A_0 \quad (2.83)$$

The controller difference equation can be expressed using polynomials R , S and T

$$r_0 x(k) = \sum_{i=0}^{\deg T} t_i y_r(k-i) - \sum_{i=0}^{\deg S} s_i y(k-i) - \sum_{i=1}^{\deg R} r_i x(k-i) \quad (2.84)$$

and finally the control signal $x(k)$ can be calculated

$$x(k) = \frac{1}{r_0} \left[\sum_{i=0}^{\deg T} t_i y_r(k-i) - \sum_{i=0}^{\deg S} s_i y(k-i) - \sum_{i=1}^{\deg R} r_i x(k-i) \right] \quad (2.85)$$

In Eqs. (2.84) and (2.85), r_i , s_i and t_i are parameters of polynomials R , S and T , respectively. Note that the polynomial R is the result of polynomial multiplication, as follows from Eq. (2.77). Consequently, the corresponding vector of parameters \mathbf{r} of polynomial R can be expressed using the convolution (*) of the vectors of parameters \mathbf{r}^1 and \mathbf{b} of polynomials R_1 and B , respectively

$$\mathbf{r} = \mathbf{r}^1 * \mathbf{b} \quad (2.86)$$

The particular element of vector \mathbf{r} can be expressed as

$$r_i = \sum_{h=h_{\min}}^{h_{\max}} r_i^1 \cdot b_{i-h}, \quad i = 0 \dots \deg R \quad (2.87)$$

where

$$h_{\min} = \max(0, i - \deg B) \quad (2.88)$$

$$h_{\max} = \min(i, \deg R_1) \quad (2.89)$$

2.5.2 Controller Design Based on a Piecewise-Linear Hammerstein Model

Let us now move on to the control design of nonlinear processes described by a piecewise-linear Hammerstein model. Let the linear part of the piecewise-linear Hammerstein model be equal to the linear process model described by Eq. (2.10) or Eq. (2.74) and let the nonlinear static function be expressed using the piecewise-linear function defined by Eq. (2.1). The idea is to design the nonlinear controller in such a way that the closed loop transfer function of the resulting controller and the piecewise-linear Hammerstein model given by Eq. (2.12) become equivalent to the desired closed loop transfer function (2.74). This means that the controller must compensate for the input nonlinearity (2.1) of the process. If the basic parameters (2.19) were known, the input nonlinearity (2.1) could be compensated for simply by inserting its inverse into the input of the piecewise-linear Hammerstein model. In such a case, the linear controller (2.84) could be used with parameters tuned according to the parameters of Eq. (2.10). But the result of the identification algorithm is a set of linear parameters (2.18), while the basic parameters are in general not known. Consequently, the controller in Eq. (2.84) has to be modified to take into account the input nonlinearity, which is integrated in the set of linear parameters. The idea is to express $x(k-i)$ in Eq. (2.84) by $u(k-i)$ using Eq. (2.1). In this way, the terms $r_i x(k-i)$ can be expressed as follows:

$$r_i x(k-i) = r_i \cdot \mathbf{I}(u(k-i), \mathbf{u})^T \cdot \mathbf{x} = \mathbf{I}(u(k-i), \mathbf{u})^T \cdot r_i \cdot \mathbf{x} \quad (2.90)$$

In Eq. (2.90), $r_i \mathbf{x}$ represents a vector with elements

$$r_i \cdot \mathbf{x} = [r_i x_0 \quad r_i x_1 \quad \dots \quad r_i x_m]^T \quad (2.91)$$

Using expression (2.87), a particular element $r_i x_j$ of vector (2.91) can be expressed as

$$r_i x_j = x_j r_i = x_j \sum_{h=h_{\min}}^{h_{\max}} r_i^1 \cdot b_{i-h} = \sum_{h=h_{\min}}^{h_{\max}} r_i^1 \cdot (b_{i-h} x_j), \quad i = 0 \dots \text{deg } R \quad (2.92)$$

The terms $(b_{i-h} x_j)$ appearing in Eq. (2.92) represent “linear parameters” of the piecewise-linear Hammerstein model. This means that it is not necessary to know the basic parameters (2.19) of the model. The set of linear parameters given by Eq. (2.18) is sufficient to express the controller parameters. This is an important fact, since the set of linear parameters is a direct result of the proposed identification procedure.

If the rightmost term of Eq. (2.90) is used in Eq. (2.84), then the following equation is obtained:

$$\mathbf{I}(u(k), \mathbf{u})^T \cdot r_0 \cdot \mathbf{x} = \sum_{i=0}^{\text{deg } T} t_i y_r(k-i) - \sum_{i=0}^{\text{deg } S} s_i y(k-i) - \sum_{i=1}^{\text{deg } R} \mathbf{I}(u(k-i), \mathbf{u})^T \cdot r_i \cdot \mathbf{x} \quad (2.93)$$

The equation can be arranged in a vector form

$$\mathbf{l}(u(k), \mathbf{u})^T \cdot r_0 \cdot \mathbf{x} = \boldsymbol{\psi}_C^T(k) \cdot \boldsymbol{\theta}_C \quad (2.94)$$

where $\boldsymbol{\psi}_C$ is the data vector and $\boldsymbol{\theta}_C$ is the parameter vector, as follows:

$$\boldsymbol{\psi}_C = \left[-\mathbf{l}(u(k-1), \mathbf{u}) \quad \dots \quad -\mathbf{l}(u(k - \deg R), \mathbf{u}) \mid y_r(k) \quad y_r(k-1) \quad \dots \right. \\ \left. y_r(k - \deg T) \mid \dots \quad -y(k) \quad -y(k-1) \quad \dots \quad -y(k - \deg S) \right]^T \quad (2.95)$$

$$\boldsymbol{\theta}_C = \left[r_1 \mathbf{x}^T \quad \dots \quad r_{\deg R} \mathbf{x}^T \mid t_0 \quad t_1 \quad \dots \quad t_{\deg T} \mid s_0 \quad s_1 \quad \dots \quad s_{\deg S} \right]^T \quad (2.96)$$

Note that Eq. (2.94) does not express the control signal u explicitly; instead it expresses the following product, denoted as $g(k)$:

$$\mathbf{l}(u(k), \mathbf{u})^T \cdot r_0 \cdot \mathbf{x} = g(k) \quad (2.97)$$

To express the control signal $u(k)$ explicitly, Eq. (2.97) has to be inverted. As explained above, this is very simple because inversion of a piecewise-linear function is also a piecewise-linear function, while the roles of u and x are reversed, as follows from Eq. (2.9),

$$u(k) = \mathbf{l}(g(k), (r_0 \mathbf{x}))^T \cdot \mathbf{u} \quad (2.98)$$

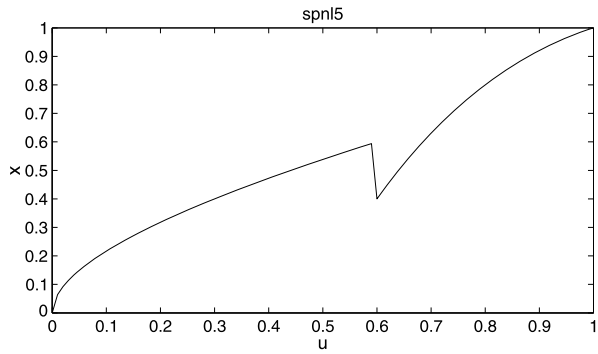
The equation represents the inversion of the nonlinear static function embedded in the model and not explicitly known. Note that due to the simplicity of expression (2.98), calculation of the control signal $u(k)$ is a computationally undemanding task. This is not so in the case of the classic, i.e. single-polynomial-based Hammerstein model. In this case it is necessary to invert the embedded polynomial, of a possibly high degree, in order to calculate the control signal $u(k)$, as shown in, e.g., [2]. The inversion of a polynomial can, in general, only be done numerically, which is a very demanding computational task that needs to be repeated in each sampling interval of the controller. The advantage of a controller based on the piecewise-linear Hammerstein model thus becomes obvious.

2.6 Simulation Study

The simulation study is divided in two parts: identification and control. First the identification algorithm was tested. The process was simulated by a continuous time system arranged in the form of the Hammerstein model. The static function of the process, shown in Fig. 2.9, was nonlinear and discontinuous

$$x(u) = \begin{cases} 0.1953u + 0.6233u^{\frac{1}{2}} & \text{if: } 0.0 \leq u < 0.6 \\ -6.6229 - 6.4046u + 14.0275u^{\frac{1}{2}} & \text{if: } 0.6 \leq u \leq 1.0 \end{cases} \quad (2.99)$$

Fig. 2.9 The nonlinear static function of the process



The linear dynamic system of the process implemented in terms of the following second order continuous time transfer function is

$$G_P(s) = \frac{y(s)}{u(s)} = \frac{(15s + 1)}{(35s + 1)(10s + 1)} \tag{2.100}$$

The input signal u was a sum of two components. The first component was a periodic slowly increasing/decreasing ramp ($T_{\text{period}} = 2000$ s, $u_{\text{min}} = 0.0$, $u_{\text{max}} = 1.0$). The second component was a periodic square pulse sequence ($T_{\text{period}} = 37.7$ s, $u_{\text{min}} = -0.006$, $u_{\text{max}} = 0.006$, duty cycle $\delta = 50\%$). The sum of both components was bounded in the range of operation ($0 \leq u \leq 1$). The measurement noise v was added to the process output to achieve a more realistic situation. The time profiles of the resulting signal u , the process output y and the measurement noise v are shown in Fig. 2.10.

It can be seen that the amplitude of the signal u is temporarily bounded in the range, which is relatively narrow when compared to the range of operation. As mentioned above, the identification of the classical (single polynomial based) Hammer-

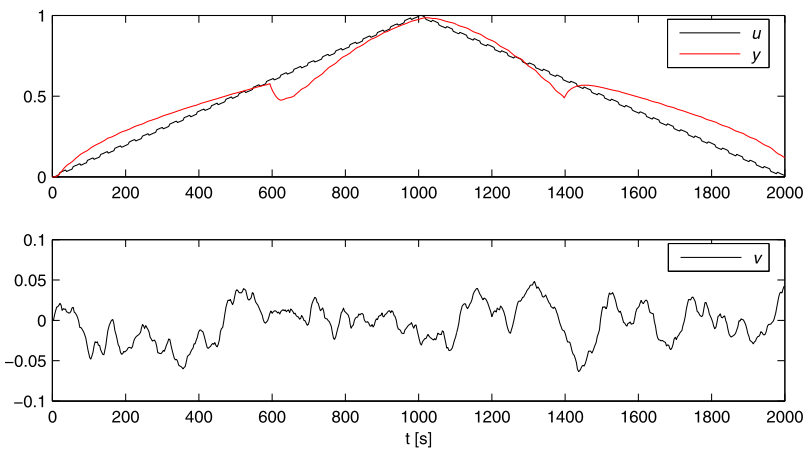


Fig. 2.10 Time profiles of the signals during identification

stein model would require an excitation signal with amplitude fairly distributed over the entire range of operation.

The linear dynamic part of the piecewise-linear Hammerstein model was chosen to be of the second order ($n = 2$) and the number of knots of the piecewise linear static function was 11 ($m = 10$). Note that the position of knots \mathbf{u} has to be chosen by the user considering possible *a priori* information on the degree of nonlinearity and positions of possible discontinuities:

- if the nonlinear static function of the process is highly nonlinear, then equidistant positioning of knots may not be optimal, instead the density of knots should be increased in the regions where higher nonlinearity is expected;
- if the nonlinear static function is discontinuous, then each point of discontinuity, u_d , has to be surrounded by two knots at positions $u_d \pm \Delta u$, where Δu is a small deviation from the point of discontinuity;
- if the input static nonlinearity has a discontinuous first derivative at u_d , then one of the knots should be placed at this point, since the first derivative of the piecewise-linear function is discontinuous at knots \mathbf{u} .

Since the nonlinear static function defined in Eq. (2.99) is discontinuous at $u = 0.6$, the position of the knots had to be arranged so as to closely surround the position of the discontinuity by positioning two knots at $u = 0.6 \pm 0.05$, i.e. $u = 0.595$ and $u = 0.605$

$$\mathbf{u} = [0.0 \quad 0.12 \quad 0.24 \quad 0.36 \quad 0.48 \quad \mathbf{0.595} \quad \mathbf{0.605} \quad 0.7 \quad 0.8 \quad 0.9 \quad 1.0] \quad (2.101)$$

During the experiment all modifications of the identification algorithm (described in Sects. 2.4.1, 2.4.2 and 2.4.3) were activated. Signals were sampled with a sampling interval $T_S = 3$ s. To compare the identified model and the process, the continuous transfer function of the process $G_P(s)$ was transformed into the discrete time transfer function $G_P(z^{-1})$ assuming signal sampling using the zero-order hold element

$$G_P(z^{-1}) = \frac{y(z^{-1})}{u(z^{-1})} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{0 + 0.1176 z^{-1} - 0.0963 z^{-2}}{1 - 1.6587 z^{-1} + 0.6800 z^{-2}} \quad (2.102)$$

The identified model parameters a_i , $i = 1 \dots n$ can be directly compared with the corresponding ideal parameters of the process $G_P(z^{-1})$, given by Eq. (2.102). The model parameters b_i , $i = 0 \dots n$, and x_j , $j = 0 \dots m$ are not expressed explicitly in the set of identified parameters $\boldsymbol{\theta}$, but only implicitly within the identified subvectors $\mathbf{b}\mathbf{x}_i$, $i = 0 \dots n$. Therefore, we compared elements of the identified subvectors $\mathbf{b}\mathbf{x}_i$ with the elements of the ideal subvectors, which were calculated using Eq. (2.16), by taking the ideal parameters b_i and x_j . The ideal parameters b_i were taken from the discrete time transfer function of the linear part of the process in Eq. (2.102). The ideal parameters x_j were calculated by Eq. (2.99) for $u = u_j$, $j = 0 \dots m$, as given by (2.101). Thus we obtain

$$\mathbf{x} = [0.0 \quad 0.2394 \quad 0.3522 \quad 0.4443 \quad 0.5256 \quad 0.5970 \quad 0.4132 \quad 0.6301 \quad 0.80 \\ 0.9206 \quad 1.0]^T \quad (2.103)$$

Based on this, we can write down a complete set of ideal process parameters:

$$\begin{aligned} [a_1 \quad a_2]^T &= [-1.6587 \quad 0.6800]^T \\ \mathbf{b}\mathbf{x}_0 &= [0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T \\ \mathbf{b}\mathbf{x}_1 &= [0.0000 \quad 0.0281 \quad 0.0414 \quad 0.0523 \quad 0.0618 \quad 0.0702 \quad 0.0486 \\ &\quad 0.0741 \quad 0.0941 \quad 0.1083 \quad 0.1176]^T \\ \mathbf{b}\mathbf{x}_2 &= [0.0000 \quad -0.0231 \quad -0.0339 \quad -0.0428 \quad -0.0506 \quad -0.0575 \\ &\quad -0.0398 \quad -0.0607 \quad -0.0770 \quad -0.0887 \quad -0.0963]^T \end{aligned} \quad (2.104)$$

The result of identification is the following set of parameters:

$$\begin{aligned} [a_1 \quad a_2]^T &= [-1.6271 \quad 0.6520]^T \\ \mathbf{b}\mathbf{x}_0 &= [0.0001 \quad 0.0004 \quad 0.0018 \quad 0.0013 \quad 0.0013 \quad 0.0014 \quad -0.0009 \\ &\quad -0.0005 \quad -0.0007 \quad -0.0009 \quad -0.0012]^T \\ \mathbf{b}\mathbf{x}_1 &= [0.0006 \quad 0.0225 \quad 0.0364 \quad 0.0479 \quad 0.0577 \quad 0.0660 \quad 0.0526 \\ &\quad 0.0783 \quad 0.0990 \quad 0.1131 \quad 0.1227]^T \\ \mathbf{b}\mathbf{x}_2 &= [0.0004 \quad -0.0194 \quad -0.0286 \quad -0.0374 \quad -0.0452 \quad -0.0521 \\ &\quad -0.0417 \quad -0.0622 \quad -0.0787 \quad -0.0897 \quad -0.0974]^T \end{aligned} \quad (2.105)$$

We can observe good agreement between the ideal and identified parameters a_1 and a_2 . Comparison of the ideal and identified subvectors $\mathbf{b}\mathbf{x}_i$ can most easily be performed graphically, as in Fig. 2.11. Also in this case we observe good agreement. The minor deviation is mainly a consequence of the measurement noise added during the identification.

Once the model parameters are known, the control system can be designed. The controller was designed according to the identified set of “linear parameters” given in Eq. (2.105) and the desired closed-loop transfer function, which was chosen to be

$$G_D(s) = \frac{y(s)}{y_r(s)} = \frac{(s+1)}{(5s+1)^2} \quad (2.106)$$

The discrete time equivalent of this function is

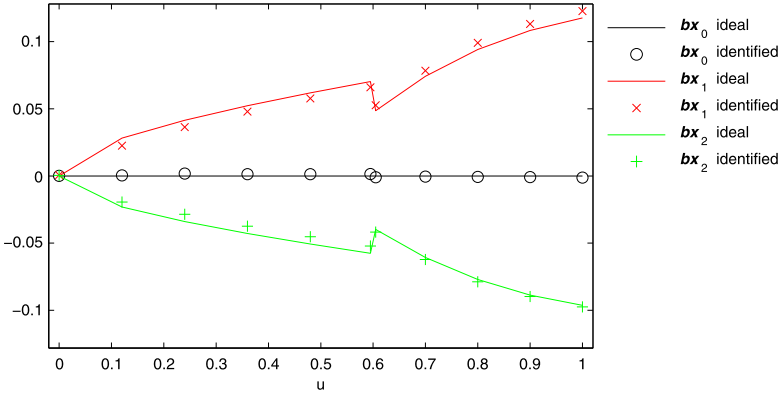


Fig. 2.11 Comparison between the ideal and the identified subvectors \mathbf{bx}_0 , \mathbf{bx}_1 , \mathbf{bx}_2

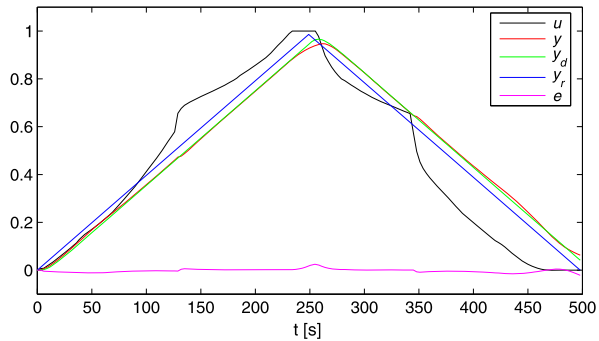
$$\begin{aligned}
 G_D(z^{-1}) &= \frac{y(z^{-1})}{y_r(z^{-1})} = \frac{b_{d1}z^{-1} + b_{d2}z^{-2}}{1 + a_{d1}z^{-1} + a_{d2}z^{-2}} \\
 &= \frac{0.1878z^{-1} + 0.0158z^{-2}}{1 - 1.0976z^{-1} + 0.3012z^{-2}} \quad (2.107)
 \end{aligned}$$

Note that the orders of the polynomials in this transfer function have to fulfil the condition in Eq. (2.79). The orders of the other polynomials are as follows: $\deg A_0 = 0$, $\deg R_1 = 0$ and $\deg S = 1$. This fulfils conditions (2.80)–(2.82). The controller parameters were expressed in terms of the linear parameters of the piecewise-linear Hammerstein model and the parameters of the desired closed-loop transfer function. First, by solving Eq. (2.78) the parameters of R_1 and S were expressed. Next, using Eq. (2.83) the parameters of T were obtained. Finally, using Eq. (2.92) the controller parameters \mathbf{rx}_i were expressed in terms of the linear parameters \mathbf{bx}_i . The result is the following set of controller parameters, which is automatically tuned based on identified model parameters:

$$\begin{aligned}
 [\mathbf{rx}_0^T \quad \mathbf{rx}_1^T]^T &= [\mathbf{bx}_1^T \quad \mathbf{bx}_2^T]^T \\
 [s_0 \quad s_1]^T &= [(a_{d1} - a_1) \quad (a_{d2} - a_2)]^T \\
 [t_0 \quad t_1]^T &= [b_{d1} \quad b_{d2}]^T
 \end{aligned} \quad (2.108)$$

To test the control performance of the control system, the output y of the controlled process was compared to the output y_d of the desired closed-loop transfer function. During the simulation, both the controlled process and the desired closed-loop transfer function were exposed to the same reference signal y_r as the input. In Fig. 2.12 it can be seen that the output of the controlled process y agrees very well with the output of the desired closed-loop transfer function y_d . The nonlinearity and discontinuity of the process are almost completely compensated for. In fact, the presence of the nonlinearity can only be seen from the control signal u . A minor deviation ($e = y_d - y$) is only noticeable in case the control signal u is saturated and

Fig. 2.12 Controller operation (u —control signal, y —process output, y_d —output of the desired closed-loop transfer function, y_r —reference signal, $e = y_d - y$)



around the point of discontinuity. This simulation example confirms the usability of the proposed controller. The controller successfully compensates for the process nonlinearity as well as for the discontinuity.

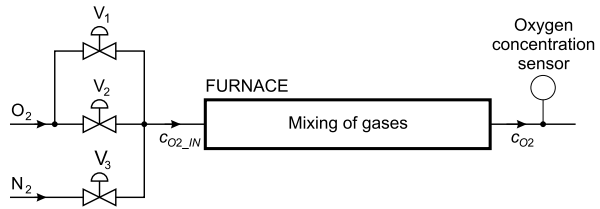
2.7 Experimental Implementation

Let us now demonstrate the usability of the piecewise-linear Hammerstein model on an industrial case study, i.e. oxygen concentration control in a sintering process of ferromagnetic material.

2.7.1 Description of the Process and the Experimental Environment

Sintering is a process that produces solid objects from powder by heating the material in sintering furnaces. The sintering process is also widely used in the production of ferromagnetic materials. The properties of a ferromagnetic material strongly depend on the process parameters during sintering (i.e. the time profiles of the temperature and atmosphere composition). During the development of the material production process it is therefore necessary to determine the optimal time profile of the process parameters which lead to the desired properties of the ferromagnetic material. In order to do this, a theoretical background is usually combined with experimental optimisation, which takes place in a laboratory environment using special sintering furnaces and corresponding control equipment providing accurate and repeatable control of the main process parameters, i.e. the temperature and atmosphere composition. In this example we will focus on a specific experimental sintering process where the atmosphere is composed of oxygen and nitrogen. During the process, the oxygen concentration and temperature in the furnace have to follow the prescribed time profiles. We will focus only on the problem of the oxygen concentration, which is controlled by adjusting the flow rates of oxygen and nitrogen. The two gasses continuously mix, enter the furnace, mix with the gas inside the furnace, and finally exit

Fig. 2.13 Setup for oxygen concentration control during the sintering process



to the atmosphere. The flow rates of the oxygen and nitrogen are controlled by mass flow control valves. Depending on the material produced, the oxygen concentration time profile may be required to vary over a very wide range, e.g. from 100 % vol. down to very low values, such as 0.01 % vol. To achieve such a wide control range of the oxygen concentration, the oxygen flow rate has to be adjusted over a very wide range, too. Since the useful control range of a typical mass flow control valve is limited to, e.g., 2–100 % of the full scale range, several mass flow control valves with different maximum flow rates must be used, and a wide enough range of the oxygen flow rate is achieved by valve switchover. In the installation considered two valves are used, V_1 with a small range and V_2 with a big range. V_3 is the mass flow control valve for nitrogen. The simplified situation is shown in Fig. 2.13.

The process equipment consists of the following three subsystems (see Fig. 2.14): *A*—electrically heated sintering furnace, *B*—oxygen concentration sensor and *C*—oxygen concentration and temperature control device, containing the mass flow control valves (V_1 , V_2 and V_3), the auxiliary on/off valves and the Mitsubishi programmable logic controller (PLC), series A1S. The PLC reads concentration from the oxygen concentration sensor and adjusts the control signals to the mass flow control valves using a PID control algorithm.

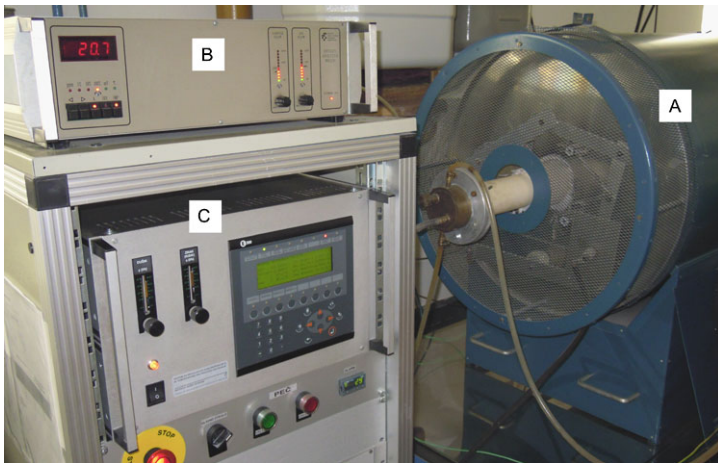


Fig. 2.14 Experimental setup: *A*—sintering furnace, *B*—oxygen concentration sensor and *C*—oxygen concentration and temperature control device

For the purpose of experimental assessment of the piecewise-linear Hammerstein control algorithm, a personal computer was connected to the PLC via an RS-232 serial communication. On the personal computer the identification and control environment for the piecewise-linear Hammerstein model was implemented.

Two types of experiments were performed, identification and control. During the identification experiments, the personal computer generated the control signal u , sending it to the PLC and simultaneously sampling the oxygen concentration response. As soon as the experiment was completed, the parameters of the piecewise-linear Hammerstein model were identified and the controller parameters were calculated. The controller parameters were then downloaded from the personal computer to the PLC, where controller Eqs. (2.94)–(2.98) were implemented in addition to the existing default PID algorithm. During the control experiment, the control signal u was generated by the PLC and sampled together with the oxygen concentration by the personal computer for the purpose of documenting and evaluating the results.

2.7.2 Process Analysis and Controller Design

Application of the piecewise-linear Hammerstein controller was motivated by problems caused by drifts in the control valves V_1 , V_2 and V_3 . These drifts result in discontinuities during valve switchover which seriously compromise the control performance of the existing PID controller.

In order to better understand the problem of oxygen concentration control, let us analyse the process of gas mixing by mathematical modelling. The resulting mathematical model will also help us to determine the structure of the piecewise-linear Hammerstein model.

The model should describe the dynamic relation between the control input u and the process output, i.e. the oxygen concentration c_{O_2} inside the furnace. Let us start with modelling of the input flow rate ϕ_s , which is the sum of the volumetric flow rates of oxygen ϕ_{O_2} and nitrogen ϕ_{N_2} entering the furnace. Due to technological reasons, ϕ_s must always be constant and in our case is 30 standard litres per hour (sl/h)

$$\phi_{O_2} + \phi_{N_2} = \phi_s = 30 \text{ sl/h} \quad (2.109)$$

The above requirement can be fulfilled by controlling both gas flow rates by means of the common control signal $u(0 \dots 1)$ using the following functions:

$$\phi_{O_2}(u) = 30u \quad (2.110)$$

$$\phi_{N_2}(u) = 30(1 - u) \quad (2.111)$$

As explained above, the system has two mass flow control valves for oxygen (V_1 , V_2) and one for nitrogen (V_3). The flow rates of valves are proportional to their

voltage command signals (v_1 , v_2 and v_3) as follows:

$$\begin{aligned}\phi_{O_2}(v_1) &= k_1 \frac{v_1}{5} + n_1, & \phi_{O_2}(v_2) &= k_2 \frac{v_2}{5} + n_2, \\ \phi_{N_2}(v_3) &= k_3 \frac{v_3}{5} + n_3\end{aligned}\quad (2.112)$$

whereby the valve gains k_1 , k_2 and k_3 represent maximum flow rates. Nominally, they have the following values: $k_1 = 20$ l/h, $k_2 = 100$ l/h and $k_3 = 30$ l/h. Constants n_1 , n_2 and n_3 represent offsets of the control valves and are ideally zero. All three command signals (v_1 , v_2 and v_3) are in the range $0 \dots 5$ V, where 0 V means zero flow and 5 V means the maximum flow rate. They are generated by analogue outputs of the programmable logic controller and are functions of the common control signal u :

$$v_1 = g_1 u, \quad v_2 = g_2 u, \quad v_3 = g_3(1 - u) \quad (2.113)$$

where g_1 , g_2 and g_3 are gains implemented in the programmable logic controller. Let us now express flow rates as functions of the common control signal $u(0 \dots 1)$. For the oxygen flow rate we take into account the switchover between the small (V_1) and big (V_2) mass flow control valves. The switching point is set at 15 l/h, which corresponds to $u = 0.5$, as follows from Eq. (2.110). Below 15 l/h, valve V_1 is used and V_2 is closed, above 15 l/h valve V_1 is closed and V_2 is in use:

$$\phi_{O_2}(u) = \begin{cases} k_1 \frac{g_1 u}{5} + n_1 & \text{if } 0 \leq u \leq 0.5 \\ k_2 \frac{g_2 u}{5} + n_2 & \text{if } 0.5 < u \leq 1 \end{cases} \quad (2.114)$$

$$\phi_{N_2}(u) = k_3 \frac{g_3(1 - u)}{5} + n_3 \quad (2.115)$$

Since relations (2.114) and (2.115) must equal relations (2.110) and (2.111), the gains g_1 , g_2 and g_3 must be appropriately determined:

$$g_1 = \frac{30 \cdot 5}{k_1} = 7.5, \quad g_2 = \frac{30 \cdot 5}{k_2} = 1.5, \quad g_3 = \frac{30 \cdot 5}{k_3} = 5 \quad (2.116)$$

If gains (g_1 , g_2 and g_3) equal the values calculated above in Eq. (2.116) and if the valve gains (k_1 , k_2 and k_3) and offsets (n_1 , n_2 and n_3) equal their nominal values, then Eq. (114) is continuous and linear. But if valve gains and offsets differ from the nominal values, Eq. (114) becomes discontinuous. Since the valve gains and offsets are defined by the analogue electronic circuits of control valves which are subject to drift, the nonlinearity and discontinuity of the relation between the control signal u and the flow rate of oxygen is a common situation during normal operation.

The outputs of mass flow control valves are connected together and gasses then enter the furnace via a common pipeline. Within the pipeline, gases are blended into a uniform gas mixture. Since the cross section of the pipeline is small (4 mm), the nitrogen and oxygen are assumed to blend completely already before entering the furnace. The oxygen concentration in the gas mixture entering the furnace is

denoted by c_{O2_IN} and it can be expressed as a ratio between the oxygen flow rate and the total flow rate

$$c_{O2_IN} = \frac{\phi_{O2}}{\phi_s} \quad (2.117)$$

Equations (2.114), (2.115) and (2.117) are all static relations and represent the nonlinear static function of the Hammerstein system.

Let us now concentrate on the dynamic part of the process. We are interested in the dynamic relationship between the oxygen concentration c_{O2_IN} in the gas mixture entering the furnace and the oxygen concentration c_{O2} in the mixture leaving the furnace. If gas diffusion inside the furnace was instantaneous, the relationship between the input and output concentration could be represented by a linear first order differential equation with static gain equal to one and a time constant proportional to the furnace volume. However, due to the specific shape of the furnace volume (a tube with an internal diameter of approximately 6 cm and length 1.3 m), gas diffusion is not instantaneous and introduces additional dynamics into the system. Theoretical modelling of the mixing and diffusion dynamics would be complex and possibly inaccurate. Instead, we estimated the actual gas mixing process dynamics by observing the response of the output concentration to the step change of the control signal u . We found that the response follows a second order linear system with a dominating time constant around 600 s. In our analysis we did not take into account the following two phenomena:

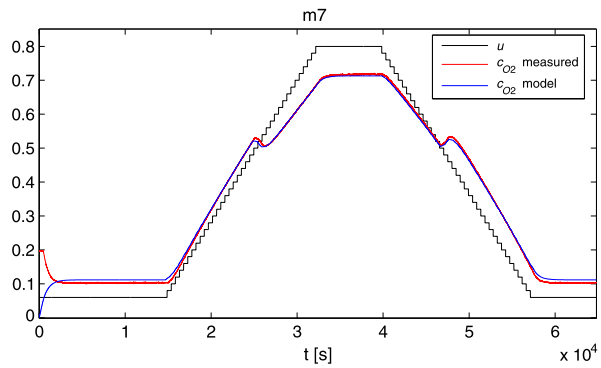
- the transport delay due to the transport of the gas via pipelines from the mass flow control valves to the furnace and from the furnace to the oxygen concentration sensor;
- the dynamic response of the oxygen concentration sensor mounted at the outlet of the furnace.

However, further evaluation shows that the transport delay in the pipes and the time constant of the oxygen sensor are both within a few seconds, which can obviously be neglected.

The analysis performed up to this point shows that the process under consideration can be described by a Hammerstein model. The relations (2.114), (2.115) and (2.117) represent the nonlinear static function of the model, while the dynamic relation between c_{O2_IN} and c_{O2} represents the linear dynamic part. As explained, the linear dynamic part can be described in terms of a second order linear differential equation with static gain equal to one.

In order to implement the piecewise-linear Hammerstein controller, it was first necessary to determine its structure. We have chosen the order of the dynamic linear part to be $n = 2$, which is in accordance with the measured process response. The number of knots was kept at the default value ($m = 10$), although a lower number would probably also be acceptable, since the characteristics of both valves are expected to remain more or less linear. The knots were arranged to surround the point

Fig. 2.15 Identification and model evaluation (u —control signal, c_{O_2} measured—oxygen concentration response, c_{O_2} model—simulated concentration)



of switchover ($u = 0.5$)

$$\mathbf{u} = [0.0 \quad 0.1 \quad 0.2 \quad 0.3 \quad 0.4 \quad \mathbf{0.495} \quad \mathbf{0.505} \quad 0.625 \quad 0.750 \quad 0.875 \quad 1.0]^T \quad (2.118)$$

Note that just before experimenting, all three mass flow control valves were calibrated, which means that all three valve gains and offsets (k, n) were close to their ideal values. To demonstrate the effect of non-ideal valve gains and offsets, we simulated a change in the valve gain k_1 of the oxygen valve V_1 from a nominal 20 l/h to 24 l/h by multiplying gain g_1 by a factor of 1.2. As explained above, such a change in the valve characteristic may happen in reality due to drift.

The next step was to determine the time profile of the excitation signal u . For model identification, the excitation signal u should skim across the whole operation region. This was achieved by changing the signal in steps from 0.06 to 0.8 and back to 0.06. The level of each step was 0.02 and the step duration was 500 seconds, which is comparable to the estimated predominant time constant of the process. Figure 2.15 shows the time profile of the excitation signal u and the resulting oxygen concentration response (c_{O_2} measured) on which the effect of the discontinuous static function is clearly visible. During identification, the sampling interval was chosen to be $T_S = 30$ sec, which is adequate for the estimated time constants of the process. The identification procedure provided the following set of “linear parameters”:

$$\begin{aligned} [a_1 \quad a_2]^T &= [-1.6647 \quad 0.6775]^T \\ \mathbf{bx}_0 &= [0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T \\ \mathbf{bx}_1 &= [0.0000 \quad 0.0135 \quad 0.0269 \quad 0.0404 \quad 0.0538 \quad 0.0666 \quad 0.0680 \\ &\quad 0.0841 \quad 0.1010 \quad 0.1178 \quad 0.1346]^T \\ \mathbf{bx}_2 &= [0.0000 \quad -0.0116 \quad -0.0232 \quad -0.0347 \quad -0.0463 \quad -0.0573 \\ &\quad -0.0585 \quad -0.0724 \quad -0.0868 \quad -0.1013 \quad -0.1158]^T \end{aligned} \quad (2.119)$$

Fig. 2.16 Piecewise-linear Hammerstein controller operation (u —control signal, c_{O_2} —concentration, c_{O_2d} —output of the desired closed-loop transfer function, c_{O_2r} —reference signal)

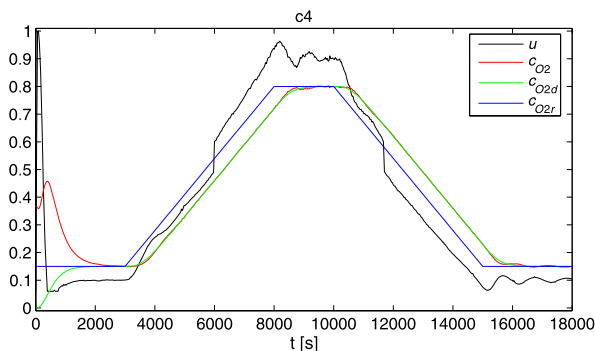


Figure 2.15 also shows the response of the identified piecewise-linear Hammerstein model (the c_{O_2} model). It can be seen that the actual concentration and the model response are very similar, which means that the model quality is adequate.

The next step was the determination of the controller parameters. To do this, we first defined the desired closed loop response in terms of the following continuous time transfer function:

$$G_D(s) = \frac{1}{(200s + 1)^3} \quad (2.120)$$

Note that in the preceding simulation study a second order transfer function was used to define the desired closed loop response. But in this case we designed the controller with additional integral action, which required a third order transfer function. The continuous time transfer function was then converted to a discrete time form using a 30 sec sampling interval

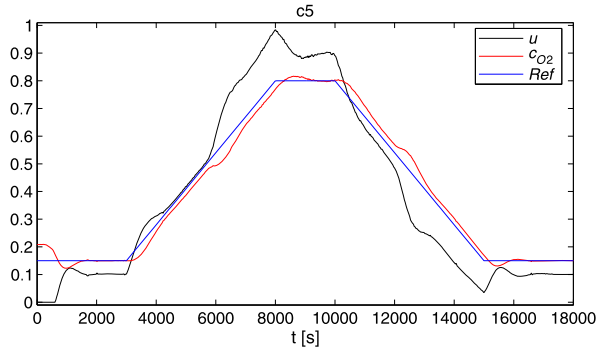
$$\begin{aligned} G_D(z^{-1}) &= \frac{b_{d1}z^{-1} + b_{d2}z^{-2} + b_{d3}z^{-3}}{1 + a_{d1}z^{-1} + a_{d2}z^{-2} + a_{d3}z^{-3}} \\ &= \frac{0.0005z^{-1} + 0.0018z^{-2} + 0.0004z^{-3}}{1 - 2.5821z^{-1} + 2.2225z^{-2} - 0.6376z^{-3}} \end{aligned} \quad (2.121)$$

Finally, the controller parameters were determined from the identified linear parameters (2.119) and the parameters of the desired closed loop transfer function (2.121). Note that the presence of the integrator in the controller required an extended set of parameters:

$$\begin{aligned} [\mathbf{rx}_0^T \quad \mathbf{rx}_1^T \quad \mathbf{rx}_2^T]^T &= [\mathbf{bx}_1^T \quad (\mathbf{bx}_2^T - \mathbf{bx}_1^T) \quad -\mathbf{bx}_2^T]^T \\ [s_0 \quad s_1 \quad s_2]^T &= [(a_{d1} - a_1 + 1) \quad (a_{d2} - a_2 + a_1) \quad (a_{d3} + a_2)]^T \\ [t_0 \quad t_1 \quad t_2]^T &= [b_{d1} \quad b_{d2} \quad b_{d3}]^T \end{aligned} \quad (2.122)$$

The operation of the controller was tested on the real process and the results are shown in Fig. 2.16. It can be seen that the measured oxygen concentration (c_{O_2}) follows the desired closed loop response (c_{O_2d}) very well, which means that the

Fig. 2.17 PID controller operation (u —control signal, c_{O_2} —concentration, Ref —reference signal)



controller is well tuned to the process dynamics. Note that the parameters of the piecewise-linear controller are derived directly from the identified model parameters and no manual tuning is necessary. In addition, the presence of the process discontinuity can only be observed from the control signal (u) and not from the oxygen concentration, which means that the controller effectively identifies and compensates for the discontinuity.

In Fig. 2.16 one can notice a time delay between the desired closed loop response (c_{O_2d}) and the concentration setpoint (c_{O_2r}). This delay is induced by the closed loop transfer function (2.120). By reducing the time constants of the poles of (2.120) we could reduce the time delay but then we would also increase the risk of system instability. Note that the existence of the time delay is tolerable in all cases where the setpoint (c_{O_2r}) is prescribed in advance in terms of a time profile. In such cases the time delay is easily compensated for by modifying the time profile of the setpoint (c_{O_2r}). The considered oxygen concentration control problem belongs to this class of problems, so the time delay does not entail any drawback.

For comparison, the results of control using the built-in PID controller are shown in Fig. 2.17. Here we can see the non-ideal time profile of the concentration at the point of valve switchover since the discontinuity is not compensated for by the controller. The effect is visible in time intervals 6000–7000 and 12,000–13,000 seconds. We can also notice overshoots when the setpoint signal changes from ramp to constant value. The overshoots are a consequence of the imperfect manual tuning of the PID controller parameters. Note that overshoots do not appear in Fig. 2.16 since the piecewise-linear Hammerstein controller is tuned according to the identification results, which means nearly perfect tuning.

2.8 Problems and Limitations in Applying the Theory

As explained above in Sect. 2.2, several problems and limitations may occur while applying the concept of the original form of the Hammerstein model in practice. We identified three major properties of the model which restrict the practical applicability and lead to potential problems. The main goal of this chapter was to overcome

the identified drawbacks and to improve the practical applicability by modifying the original form of the model. A theoretical analysis along with the simulation results and experimental implementation demonstrate that the principal goal was achieved and the main drawbacks were eliminated relatively effectively.

However, both the identification procedure and the control algorithm may still face some problems or limitations when applied to particular processes. Below we identify them briefly.

One of the problems is related to the number and arrangement of knots. Both the number and the position of the knots are not determined automatically, but rather are a matter of designer decision. In cases of mild nonlinearity, the identification will most likely provide good results with the default arrangement, i.e. 11 equidistantly distributed knots. But in cases of functions with a higher degree of nonlinearity, discontinuities, or a discontinuous first derivative, the default arrangement is no longer optimal and must be set manually. This can be done either by using prior knowledge about the process or from information gathered during initial identification with default parameters.

The identification algorithm of the piecewise-linear Hammerstein model is based on recursive least squares identification of linear systems. It is well known that this kind of algorithm is sensitive to the presence of measurement noise in the measured process output signal. If noise level is relatively low or moderate (as in the presented simulation study and experimental implementation), the estimated model parameters are expected to be close to true values. However, if the level of noise is high, then the estimated parameters will likely be inaccurate and the model will not describe the process well enough. If such model is employed for control, the control performance is not expected to be good. Therefore special attention has to be devoted to the quality of the signals, and the proper measures must be taken to either prevent the noise or at least minimise it. Sometimes the noise is not a consequence of the measurement method and/or signals, but it originates from the process itself. In such cases the noise cannot easily be reduced and identification will most likely face problems. A possible solution would be the application of an identification algorithm less sensitive to noise, e.g. instrumental variables.

For successful identification, special attention has to be devoted to the selection of the input signal and sampling of the process response. If the input signal is composed of serial step functions (as in the experimental implementation presented in Fig. 2.15), then the duration and amplitude of the step is important. Step duration should be sufficiently long to capture the response of the largest time constant of the process and the sampling interval should be short enough to not miss the response of the shortest time constant. The amplitude of the steps is also important. It should be smaller than the distance between knots, but also big enough so that the amplitude of the process response is well above the measurement noise. In order to determine the right excitation signal, some *a priori* information about the process dynamic structure is very useful. Alternatively, preliminary identification based on a single step response should be performed in order to obtain the initial information about the process dynamics. Once this is done, the identification signal can be determined and complete identification can be performed.

The control algorithm is based on a linear pole placement controller for linear processes. This method is relatively simple and theoretically sound, but the major problem is that its design parameters are not directly related to the classical performance requirements. More specifically, the design parameters of the pole placement controller are given in terms of poles and zeros of the desired closed loop transfer function (2.74), (2.106). But the typical performance requirements are less specific and they are usually given in terms of rise time, settling time, etc. The problem is related to the fact that a given set of performance requirements can be fulfilled by many different desired closed loop transfer functions, and some of them may lead to a less robust controller, i.e. one that is sensitive to the mismatch between the process and the identified model. In such a case, several different desired transfer functions may need to be tested before a satisfactory result is obtained.

The pole placement controller may also be sensitive to the measurement noise, which can be reflected in the control signal u , which can harm the performance of the closed-loop system and increase wear in the actuators. The problem of measurement noise can be reduced by filtering the process output. However, the presence of the filter generally changes the process dynamics, which may decrease control performance and stability due to mismatch between the process and the model. If filtering is used, it is necessary to treat it as part of the process and identification should be performed according to the filtered process output.

If the mentioned problems and limitations occur, they can be handled in most cases, but they require designer intervention and experience. Unfortunately, this interaction cannot easily be generalised since it is very case-dependent.

2.9 Conclusion

The research presented in this chapter was a response to the challenge presented by the need to modify the standard form of the Hammerstein model in order to alleviate the drawbacks which hinder its implementation in practice. We proposed a piecewise-linear Hammerstein model with piecewise-linear representation of the nonlinear static function, as opposed to the single polynomial that is used in the original version of the Hammerstein model. Thanks to this, three improvements were obtained which directly increase the practical applicability of the model.

Firstly, the proposed algorithm does not require persistent excitation over the entire range of operation. Instead, an excitation signal with temporarily bounded amplitude is sufficient. This is important when industrial processes are considered, since in this case only signals with a bounded amplitude region are allowed to be applied. Due to the linearity in the model parameters, a classic least squares-based identification algorithm could be used as a basis for the development of the new identification approach. This algorithm was then adapted and enhanced to take into account the specifics of the identification signal and properties of the applied piecewise-linear Hammerstein model.

Secondly, the proposed model is very convenient for describing processes with highly nonlinear and/or discontinuous memoryless static functions. In the case of

highly nonlinear static functions, the density of knots can be increased in the region of high nonlinearity, thus increasing the precision. In the case of discontinuous static functions, each point of discontinuity can be surrounded by two close-standing knots, thus enabling approximation of the discontinuity.

Finally, it was shown that the proposed model can very easily be integrated into a self-tuning control algorithm with a simple structure and low computational effort, which enables execution also in programmable logic controllers. This is due to simple analytical inversion of the embedded nonlinear static function of the model, implemented as a piecewise-linear function. In addition, it was shown that the controller parameters can be expressed in terms of “linear model parameters”, which are a direct result of the identification, while the basic parameters of the piecewise-linear Hammerstein model do not have to be expressed explicitly. This enables automatic tuning of the controller parameters.

Although the motivation of the work was to improve the practical applicability of the control method, there are still some remaining issues which may hinder implementation in some cases and reduce control performance. One such problem is the level of measurement noise, which can worsen both identification and control if it is too high. Experience also shows that the proper selection of the time profile of the identification signal and the sampling interval are very important for successful identification. Furthermore, the structure of the model (the order of the linear part and the distribution of knots) has to be determined manually, which may be difficult when the process is not well understood or no information exists. Finally, the control goal is expressed in terms of the desired closed loop transfer function. This is not directly related to traditional engineering design criteria and leads to redundancy, since many different transfer functions may satisfy the particular engineering criteria. The mentioned issues can be handled, but they require manual interaction, based on designer experience.

Acknowledgements The financial support of the Slovenian Research Agency through Programme P2-0001 is gratefully acknowledged.

This chapter is based on: Dolanc G, Strmčnik S (2005) Identification of non-linear systems using a piecewise-linear Hammerstein model. *Systems & Control Letters*, 54:145–158, ©Elsevier and Dolanc G, Strmčnik S (2008) Design of a nonlinear controller based on a piecewise-linear Hammerstein model. *Systems & Control Letters*, 57:332–339, ©Elsevier.

References

1. Abonyi J, Babuska R, Ayala Botto M, Szeifert F, Lajos N (2000) Identification and control of nonlinear systems using fuzzy Hammerstein models. *Industrial and Engineering Chemistry Research* 39:4302–4314
2. Anbumani K, Patnaik LM, Sarma IG (1985) Self-tuning pole-placement in nonlinear systems of the Hammerstein model. *IEEE Transactions on Industrial Electronics* 32:166–170
3. Åström KJ, Wittenmark B (1989) *Adaptive Control*. Addison-Wesley, Reading
4. Bai EW (1998) An optimal two-stage identification algorithm for Hammerstein-Wiener nonlinear systems. *Automatica* 34:333–338

5. Bhat J, Chidambaram M, Madhavan KP (1990) Adaptive feedforward control of Hammerstein nonlinear systems. *International Journal of Control* 51:237–242
6. Billings SA, Fakhouri SY (1979) Non-linear system identification using the Hammerstein model. *International Journal of Systems Science* 10:567–578
7. Chang FHI, Luus R (1971) A noniterative method for identification using Hammerstein model. *IEEE Transactions on Automatic Control* 16:464–468
8. Chen CH, Fassois SD (1992) Maximum likelihood identification of stochastic Wiener–Hammerstein-type non-linear systems. *Mechanical Systems and Signal Processing* 6:135–153
9. Crama P, Schoukens J (2001) Initial estimates of Wiener and Hammerstein systems using multisine excitation. *IEEE Transactions on Instrumentation and Measurement* 50:1791–1795
10. Crama P, Schoukens J, Pintelon R (2003) Generation of enhanced initial estimates for Wiener systems and Hammerstein systems. In: *Proceedings of the 13th IFAC Symposium on System Identification, Rotterdam, The Netherlands*, pp 857–862
11. Dolanc G, Strmčnik S (2005) Identification of non-linear systems using a piecewise-linear Hammerstein model. *Systems & Control Letters* 54:145–158
12. Dolanc G, Strmčnik S (2008) Design of a nonlinear controller based on a piecewise-linear Hammerstein model. *Systems & Control Letters* 57:332–339
13. Fruzzetti KP, Palazoglu A, McDonald KA (1997) Nonlinear model predictive control using Hammerstein models. *Journal of Process Control* 7:31–41
14. Greblicki W, Pawlak M (1987) Hammerstein system identification by non-parametric regression estimation. *International Journal of Control* 45:343–354
15. Haber R, Keviczky L (1974) The identification of the discrete-time Hammerstein model. *Periodica–Polytechnica* 18:71–84
16. Haddad WM, Chellaboina V (2001) Nonlinear control of Hammerstein systems with passive nonlinear dynamics. *IEEE Transactions on Automatic Control* 46:1630–1634
17. Hong X, Mitchell RJ (2006) A pole assignment controller for Bezier–Bernstein polynomial based Hammerstein model. In: *Proceedings of the International Control Conference, Glasgow, Scotland, United Kingdom*
18. Hong X, Mitchell RJ (2006) Bezier–Bernstein polynomial based Hammerstein model and identification algorithm. In: *Proceedings of the International Control Conference, Glasgow, Scotland, United Kingdom*
19. Hsia TC (1976) A multi-stage least squares method for identifying Hammerstein model nonlinear systems. In: *Proceedings of the IEEE Conference on Decision and Control including the 15th Symposium on Adaptive Processes, Clearwater, Florida*, pp 934–938
20. Hsu JT, Ngo KDT (1997) A Hammerstein-based dynamic model for hysteresis phenomenon. *IEEE Transactions on Power Electronics* 12:406–413
21. Isermann R, Münchhof M (2011) *Identification of Dynamic Systems*. Springer, Berlin
22. Keviczky L, Banyasz C (2000) Generic two-degree of freedom control systems for linear and nonlinear processes. *Systems Science* 26:5–24
23. Knohl T, Xu WM, Unbehauen H (2003) Indirect adaptive dual control for Hammerstein systems using ANN. *Control Engineering Practice* 11:377–385
24. Kung MC, Womack BF (1984) Discrete time adaptive control of linear dynamic systems with a two-segment piecewise-linear asymmetric nonlinearity. *IEEE Transactions on Automatic Control* 29:1170–1172
25. Kung MC, Womack BF (1984) Discrete time adaptive control of linear systems with preload nonlinearity. *Automatica* 20:477–479
26. Lancaster P, Šalkauskas K (1986) *Curve and Surface Fitting: An Introduction*. Academic Press, San Diego
27. Ma Z, Jutan A, Bajić VB (2000) Nonlinear self-tuning controller for Hammerstein plants with application to a pressure tank. *International Journal of Computers, Systems and Signals* 1:221–230
28. Narendra KS, Gallman PG (1966) An iterative method for the identification of nonlinear systems using a Hammerstein model. *IEEE Transactions on Automatic Control* 11:546–550

29. Nesić D, Mareels IMY (1998) Dead-beat control of simple Hammerstein models. *IEEE Transactions on Automatic Control* 43:1184–1188
30. Stoica P (1981) On the convergence of an iterative algorithm used for Hammerstein system identification. *IEEE Transactions on Automatic Control* 26:967–969
31. Stoica P, Soderstrom T (1982) Instrumental variable methods for identification of Hammerstein Systems. *International Journal of Control* 35:459–476
32. Vörös J (1997) Parameter identification of discontinuous Hammerstein systems. *Automatica* 33:1141–1146
33. Vörös J (1999) Iterative algorithm for parameter identification of Hammerstein systems with two-segment nonlinearities. *IEEE Transactions on Automatic Control* 44:2145–2149
34. Zhang J, Lang S (1989) Explicit self-tuning control for a class of non-linear systems. *Automatica* 25:593–596
35. Zhu QM, Guo LZ (2002) A pole placement controller for non-linear dynamic plants. Part I. *Journal of Systems and Control Engineering* 216(6):467–476
36. Zhu Y (2002) Estimation of an N-L-N Hammerstein-Wiener model. *Automatica* 38:1607–1614