# 1

# Introduction

The class of block-oriented nonlinear models includes complex models which are composed of linear dynamic systems and nonlinear static elements. Wiener and Hammerstein models are the most known and the most widely implemented members of this class. A model is called the Wiener model if the linear dynamic block (element) precedes the nonlinear static one. In the Hammerstein model, the connection order is reversed.
Models of nonlinear static elements can be realized in different forms such as polynomials, splines, basis functions, wavelets, neural networks, look-up tables, and fuzzy models. Impulse response models, pulse transfer models, and state space models are common representations of linear dynamic systems. Depending on realization forms of both of these elements, various structures of Wiener and Hammerstein models can be obtained. To evaluate and compare them, the following properties are commonly taken into account: approximation accuracy, extrapolation behavior, interpolation behavior, smoothness, sensitivity to noise, available parameter optimization methods, and available structure optimization methods.
From the approximation theorem of Weierstrass, it follows that any continuous function defined on the interval $[a, b]$ can be approximated arbitrarily closely by a polynomial. Polynomial models are widely used as models of nonlinear elements. A great advantage of polynomial models is effective parameter optimization, which can be performed off-line with the least squares method or on-line with its recursive version. Moreover, structure selection can also be performed effectively with the orthogonal least squares algorithm, in which the set of regressors is transformed into a set of orthogonal basis vectors. With this algorithm, it is possible to calculate individual contribution from each basis vector to output variance.
Polynomials have some fundamental disadvantages as well. First of all, although any continuous function can be approximated arbitrary closely by a polynomial, some nonlinear functions require a very high polynomial order. In multi-input multi-output models, the number of parameters grows strongly when the number of inputs increases. This, increases the model uncer-

A. Janczak: Identification of Nonlinear Systems, LNCIS 310, pp. 1–30, 2005.

tainty and may cause the optimization problem numerically ill-conditioned. The other disadvantages of polynomials are their oscillatory interpolation and extrapolation properties. Therefore, in practice, the application of polynomial models is recommended only in some specific cases where the system structure can be assumed to be approximately of the polynomial type.

An alternative to polynomial models are neural network models of the multilayer perceptron architecture. Multilayer perceptrons are feedforward neural networks containing one or more hidden layers of nonlinear elements, but one hidden layer is the most common choice in practice. The application of multilayer perceptrons in approximation problems is justified by their universal approximation property which states that a single hidden layer is sufficient to uniformly approximate any continuous function with support in a unit hypercube [31]. Multilayer perceptrons, owing to their advantages such as high approximation accuracy, lower numbers of nodes and weights in comparison with other model architectures, capability to generate a wide variety of functions, are the most frequently applied neural networks. Unlike polynomials, multilayer perceptron models do not suffer oscillatory interpolation and extrapolation behavior. They reveal a tendency to monotonic interpolation. The extrapolation behavior is also smooth but in a long range the network response tends to a constant value owing to the saturation of commonly applied sigmoidal functions. Multilayer perceptron models are very useful for high dimensional problems as well. This comes from the fact that the numer of weights in a multilayer perceptron model is proportional to the number of inputs. In contrast to polynomial models, multilayer perceptron models can be successfully used not only to represent systems described by polynomials but also by infinite power series expansions. Summarizing, as they offer the user some different interesting features, multilayer perceptron models are complementary to polynomial ones.

Obviously, multilayer perceptron models are not free of drawbacks. The most significant of them are the use of local optimization methods to update the weights, and a risk of getting trapped in a shallow local minimum. This leads often to the necessity of repeated training with different weight initializations. Moreover, trial and error techniques have to be used for some parameters such as initial weights, learning rates, etc. Also, the available model structure optimization methods are rather computationally intensive.

Now, with the development of the identification of Wiener and Hammerstein systems it is possible to systemize different techniques and to present them in a unified framework. We attempt at such a presentation in this monograph by reviewing the existing approaches along with a presentation of original research papers and some results that have not been published yet.

Neural network models of Wiener and Hammerstein systems considered in Chapters 2 and 3 are composed of a multilayer perceptron model of the nonlinear element and one or more linear nodes with tapped delay lines constituting a model of the linear dynamic system. Series-parallel Wiener models contain also another multilayer perceptron model of the inverse nonlinear element.

Two basic configurations of models, i.e., series-parallel and parallel models are discussed. In series-parallel models, the gradient can be calculated with the well-known backpropagation method. In parallel models, only a crude approximation of the gradient can be obtained with the backpropagation method. Therefore, two other methods, referred to as the sensitivity method and the backpropagation through time method, which provide the exact value of the gradient or its more accurate approximation, should be taken into account. All these gradient calculation methods are derived in a unified manner, both for the SISO and MIMO cases. Computational complexity of the methods is analyzed and expressed in terms of polynomial orders and the number of unfolded time steps in the case of the truncated backpropagation through time method. The accuracy of gradient calculation with the truncated backpropagation through time method is analyzed as well. It is shown that the accuracy of gradient calculation depends on the numbers of discrete time steps necessary for the impulse responses of sensitivity models to decrease to negligible small values. Based on this result, adaptive procedures for adjusting the number of unfolded time steps are proposed to meet the specified degrees of accuracy. The original contribution of the book comprises new approaches to the identification of Wiener and Hammerstein systems and some concerned theoretical results. For both SISO and MIMO neural network models, the following gradient calculation methods are derived and analyzed:

- Backpropagation for series-parallel Wiener models.
- Backpropagation for parallel Wiener models.
- Sensitivity method for parallel Wiener models.
- Truncated backpropagation through time for parallel Wiener models.
- Backpropagation for series-parallel Hammerstein models.
- Backpropagation for parallel Hammerstein models.
- Sensitivity method for parallel Hammerstein models.
- Truncated backpropagation through time for parallel Hammerstein models.

Having the rules for gradient calculation derived, various gradient-based learning algorithms can be implemented easily. Sequential versions of the steepest descent, prediction error, and combined steepest descent and recursive least squares algorithm are discussed in detail in Chapters 2 and 3.

Another group of identification methods, derived and discussed in the book, uses polynomial representation of Wiener systems:

- Least squares method for polynomial Wiener systems with the linear term.
- Least squares method for polynomial Wiener systems without the linear term.
- Combined least squares and instrumental variables method for polynomial Wiener systems with the linear term.
- Combined least squares and instrumental variables method for polynomial Wiener systems without the linear term.

- Recursive prediction error method.
- Recursive pseudolinear regression method.

All these polynomial-based identification methods employ a pulse transfer function representation of the system dynamics and polynomial models of the nonlinear element or its inverse. In spite of the fact that polynomial models of both Wiener and Hammerstein systems can be expressed in linear-in-parameters forms, such transformations lead to parameter redundancy as transformed models have a higher number of parameters than the original ones. The number of parameters of transformed models grows strongly with increasing the model order. As a result, the variance error increases and some numerical problems may occur as well. Moreover, as shown in Chapter 4, to transform a Wiener system into the linear-in-parameters form, the nonlinearity have to be invertible, i.e., the nonlinear mapping must be strictly monotonic. It is also shown that the least squares parameter estimates are inconsistent. To obtain consistent parameter estimates, a combined least-squares and instrumental-variables method is proposed. The restrictive assumption of invertibility of the nonlinear element is no more necessary in both the prediction error method and the pseudolinear regression method. In this case, however, the Wiener model is nonlinear in the parameters and the parameter estimation becomes a nonlinear optimization task.

Wiener and Hammerstein models have found numerous industrial applications for system modelling, control, fault detection and isolation. Chapter 6 gives a brief review of applications which includes the following systems and processes:

- pH neutralization process,
- heat exchangers,
- distillation columns,
- chromatographic separation process,
- polymerization reactor,
- quartz microbalance polymer-coated sensors,
- hydraulic plants,
- electro-hydraulic servo-systems,
- pneumatic valves,
- pump-valve systems,
- electrooptical dynamic systems,
- high power amplifiers in satellite communication channels,
- loudspeakers,
- active noise cancellation systems,
- charging process in diesel engines,
- iron oxide pellet cooling process,
- sugar evaporator.

Wiener and Hammerstein models reveal the capability of describing a wide class of different systems and apart from industrial examples, there are many other applications in biology and medicine.

## 1.1 Models of dynamic systems

This section gives a brief review of discrete-time models of time-invariant dynamic systems. By way of introduction, the section starts with linear model structures and shows nonlinear models as generalizations of linear ones.

### 1.1.1 Linear models

According to Ljung [112], a time-invariant model can be specified by the impulse response $h(n)$, the spectrum $\Phi(\omega)$ of the additive disturbance $H(q^{-1})\varepsilon(n)$, and the probability density function $f_\varepsilon(\cdot)$ of the disturbance $\varepsilon(n)$. The output $y(n)$ of a discrete-time linear model excited by an input $u(n)$ and disturbed additively by $\varepsilon(n)$ is

$$y(n) = G(q^{-1})u(n) + H(q^{-1})\varepsilon(n), \tag{1.1}$$

where $G(q^{-1})$ is called the input pulse transfer function, $H(q^{-1})$ is the noise pulse transfer function, and $q^{-1}$ denotes the backward shift operator. Unless stated more precisely, we assume $\varepsilon(n)$ to be a zero-mean stationary independent stochastic process. Representing $G(q^{-1})$ and $H(q^{-1})$ as rational functions leads to the general model structure of the following form [112, 149]:

$$A(q^{-1})y(n) = \frac{B(q^{-1})}{F(q^{-1})}u(n) + \frac{C(q^{-1})}{D(q^{-1})}\varepsilon(n), \tag{1.2}$$

where

$$A(q^{-1}) = 1 + a_1q^{-1} + \cdots + a_{na}q^{-na}, \tag{1.3}$$

$$B(q^{-1}) = b_1q^{-1} + \cdots + b_{nb}q^{-nb}, \tag{1.4}$$

$$C(q^{-1}) = 1 + c_1q^{-1} + \cdots + c_{nc}q^{-nc}, \tag{1.5}$$

$$D(q^{-1}) = 1 + d_1q^{-1} + \cdots + d_{nd}q^{-nd}, \tag{1.6}$$

$$F(q^{-1}) = 1 + f_1q^{-1} + \cdots + f_{nf}q^{-nf}. \tag{1.7}$$

In practice, the structure (1.2) is usually too general. Depending on which of the polynomials (1.3) – (1.7) are used, 32 different model sets can be distinguished. A few commonly used structures, which belong to the general family of structures, are listed below.

**Finite impulse response (FIR) structure**. The choice of $A(q^{-1}) = C(q^{-1}) = D(q^{-1}) = F(q^{-1}) = 1$ results in the simplest model structure known as the finite impulse response model

$$y(n) = B(q^{-1})u(n) + \varepsilon(n). \tag{1.8}$$

The output of the model (1.8) is a weighted sum of $nb$ past inputs $u(n-1)$, …, $u(n-nb)$:

$$y(n) = b_1 u(n-1) + \cdots + b_{nb} u(n-nb) + \varepsilon(n). \tag{1.9}$$

The optimal one-step-ahead predictor, i.e., the predictor that minimizes the prediction error variance is

$$\hat{y}(n|n-1) = B(q^{-1})u(n) = b_1 u_(n-1) + \cdots + b_{nb} u(n-nb). \tag{1.10}$$

Introducing the parameter vector $\boldsymbol{\theta}$,

$$\boldsymbol{\theta} = \big[b_1 \ldots b_{nb}\big]^T, \tag{1.11}$$

and the regression vector $\mathbf{x}(n)$,

$$\mathbf{x}(n) = \big[u(n-1) \ldots u(n-nb)\big]^T, \tag{1.12}$$

(1.10) can equivalently be expressed in a regression form:

$$\hat{y}(n|n-1) = \mathbf{x}^T(n)\boldsymbol{\theta}. \tag{1.13}$$

The FIR model is able to approximate asymptotically stable dynamic systems quite well if their impulse responses decay reasonably fast.

**Autoregressive (AR) structure**. The AR model structure is defined with the choice of $B(q^{-1}) = 0$, and $C(q^{-1}) = D(q^{-1}) = F(q^{-1}) = 1$:

$$y(n) = \frac{1}{A(q^{-1})}\varepsilon(n). \tag{1.14}$$

In this case, the parameter vector $\boldsymbol{\theta}$ and the regression vector $\mathbf{x}(n)$ become

$$\boldsymbol{\theta} = \big[a_1 \ldots a_{na}\big]^T, \tag{1.15}$$

$$\mathbf{x}(n) = \big[-y(n-1) \ldots -y(n-na)\big]^T. \tag{1.16}$$

**Moving average (MA) structure**. The MA model structure corresponds to the choice of $B(q^{-1}) = 0$, and $A(q^{-1}) = D(q^{-1}) = F(q^{-1}) = 1$:

$$y(n) = C(q^{-1})\varepsilon(n). \tag{1.17}$$

The parameter vector $\boldsymbol{\theta}$ and the regression vector $\mathbf{x}(n)$ are

$$\boldsymbol{\theta} = \big[c_1 \ldots c_{nc}\big]^T, \tag{1.18}$$

$$\mathbf{x}(n) = \big[\varepsilon(n-1) \ldots \varepsilon(n-nc)\big]^T. \tag{1.19}$$

**Autoregressive with exogenous input (ARX) structure**. The ARX model structure can be obtained with the choice of $C(q^{-1}) = D(q^{-1}) = F(q^{-1}) = 1$:

$$y(n) = \frac{B(q^{-1})}{A(q^{-1})}u(n) + \frac{1}{A(q^{-1})}\varepsilon(n). \tag{1.20}$$

For the ARX model, the parameter vector $\boldsymbol{\theta}$ and the regression vector $\mathbf{x}(n)$ have the forms

$$\boldsymbol{\theta} = \big[a_1 \dots a_{na}\ b_1 \dots b_{nb}\big]^T, \tag{1.21}$$

$$\mathbf{x}(n) = \big[-y(n-1)\dots - y(n-na)\ u(n-1)\dots u(n-nb)\big]^T. \tag{1.22}$$

**Autoregressive moving average (ARMA) structure**. A combination of the autoregressive model and the moving average model results in the ARMA model. It can be obtained with the choice of $B(q^{-1}) = 0$ and $D(q^{-1}) = F(q^{-1}) = 1$:

$$y(n) = \frac{C(q^{-1})}{A(q^{-1})}\varepsilon(n). \tag{1.23}$$

In this case, the parameter vector $\boldsymbol{\theta}$ and the regression vector $\mathbf{x}(n)$ become

$$\boldsymbol{\theta} = \big[a_1 \dots a_{na}\ c_1 \dots c_{nc}\big]^T, \tag{1.24}$$

$$\mathbf{x}(n) = \big[-y(n-1)\dots - y(n-na)\ \varepsilon(n-1)\dots \varepsilon(n-nc)\big]^T. \tag{1.25}$$

**Autoregressive moving average with exogenous input (ARMAX) structure**. The ARMAX model is the most general structure of all those considered up to now as it contains all of them as special cases. To obtain the ARMAX model, we choose $D(q^{-1}) = F(q^{-1}) = 1$:

$$y(n) = \frac{B(q^{-1})}{A(q^{-1})}u(n) + \frac{C(q^{-1})}{A(q^{-1})}\varepsilon(n). \tag{1.26}$$

For the ARMAX model, the parameter vector $\boldsymbol{\theta}$ and the regression vector $\mathbf{x}(n)$ are defined as follows:

$$\boldsymbol{\theta} = \big[a_1 \dots a_{na}\ b_1 \dots b_{nb}\ c_1 \dots c_{nc}\big]^T, \tag{1.27}$$

$$\begin{aligned}\mathbf{x}(n) = \big[&-y(n-1)\dots - y(n-na)\ u(n-1)\dots u(n-nb)\\ &\varepsilon(n-1)\dots \varepsilon(n-nc)\big]^T.\end{aligned} \tag{1.28}$$

**Output error (OE) structure**. The OE model can be obtained if we choose $A(q^{-1}) = C(q^{-1}) = D(q^{-1}) = 1$:

$$y(n) = \frac{B(q^{-1})}{F(q^{-1})}u(n) + \varepsilon(n). \tag{1.29}$$

In this case, the parameter vector $\boldsymbol{\theta}$ and the regression vector $\mathbf{x}(n)$ are defined as

$$\boldsymbol{\theta} = \big[f_1 \dots f_{nf}\ b_1 \dots b_{nb}\big]^T, \tag{1.30}$$

$$\mathbf{x}(n) = \big[-\hat{y}(n-1)\dots - \hat{y}(n-nf)\ u(n-1)\dots u(n-nb)\big]^T, \tag{1.31}$$

where

$$\hat{y}(n) = \frac{B(q^{-1})}{F(q^{-1})} u(n). \tag{1.32}$$

**Box-Jenkins (BJ) structure**. A structure which is a more general development of the OE model, is called the Box-Jenkins model. To obtain the BJ model, the choice of $A(q^{-1}) = 1$ should be made:

$$y(n) = \frac{B(q^{-1})}{F(q^{-1})} u(n) + \frac{C(q^{-1})}{D(q^{-1})} \varepsilon(n). \tag{1.33}$$

The one-step-ahead predictor for the BJ model has the form [112]:

$$\hat{y}(n|n-1) = \frac{D(q^{-1})}{C(q^{-1})} \frac{B(q^{-1})}{F(q^{-1})} u(n) + \frac{C(q^{-1}) - D(q^{-1})}{C(q^{-1})} y(n). \tag{1.34}$$

### 1.1.2 Nonlinear models

Nonlinear counterparts of linear model structures can be defined assuming that there is a nonlinear relationship between the actual system output and past system inputs, the past system or model outputs, and the actual and past additive disturbances. For nonlinear input-output models we have

$$y(n) = g\big(\mathbf{x}(n), \boldsymbol{\theta}\big) + \varepsilon(n) \tag{1.35}$$

or, in the predictor form,

$$\hat{y}(n|n-1) = g\big(\mathbf{x}(n), \boldsymbol{\theta}\big). \tag{1.36}$$

Depending on the form of $\mathbf{x}(n)$, nonlinear model structures known as NFIR, NAR, NMA, NARMA, NARX, NARMAX, NOE, NBJ can be defined. The function $g(\cdot)$ is a nonlinear mapping, which for any given $\boldsymbol{\theta}$ maps $\mathbb{R}^d$ to $\mathbb{R}^p$, where $d$ is the number of regressors (elements of the vector $\mathbf{x}(n)$) and $p$ is the number of model outputs. In a parametric approach, $g(\cdot)$ is expressed by a function expansion:

$$g\big(\mathbf{x}(n), \boldsymbol{\theta}\big) = \sum_{m=1}^{ng} \beta_m g_m\big(\mathbf{x}(n)\big), \tag{1.37}$$

where $g_m(\cdot)$ is called the basis function, and $\boldsymbol{\theta} = [\beta_1 \ldots \beta_{ng}]^T$. There are different forms of function expansions, used for nonlinear systems representation, which are based on polynomials, Volterra kernels, Fourier series, piecewise constant functions, radial basis functions, wavelets, kernel estimators, neural networks, and fuzzy models [112].

**Discrete-time Volterra models**. If the function $g(\cdot)$ is analytic, then the system response can be represented by the Volterra series [109]:

$$\begin{aligned} y(n) = \sum_{i_1=1}^{n} h_1(i_1)u(n-i_1) + \sum_{i_1=1}^{n}\sum_{i_2=1}^{n} h_2(i_1,i_1)u(n-i_1)u(n-i_2)+ \\ \cdots + \varepsilon(n), \end{aligned} \tag{1.38}$$

where the kernel functions $h_j(i_1,\ldots,i_j)$, $j = 1, 2, \ldots$, describe system dynamics. Two basic problems associated with practical application of Volterra series are difficulty concerning the measurement of Volterra kernel functions and the convergence of Volterra series [145]. The Volterra series representation is really a Taylor series with memory. Therefore, the problem of convergence is the same as that of Taylor series representation of a function, i.e., the Volterra series representation of a system may converge for only a limited range of the system input amplitude. To circumvent this problem, Wiener formed a new set of functionals from the Volterra functionals – G-funtionals which have an orthogonal property when their input is a Gaussian process.

Extensive studies of Volterra models show their successful application to low order systems. The reasons for this system order limitation are practical difficulties in extending kernel estimation to orders higher than the third [117].

**Kolmogorov-Gabor models**. The application of generalized polynomial models to the representation of nonlinear dynamic systems

$$y(n) = f\big(u(n-1),\ldots,u(n-nb), y(n-1),\ldots,y(n-na)\big) + \varepsilon(n) \tag{1.39}$$

results in Kolmogorov-Gabor models:

$$\begin{aligned} y(n) = a_0 + \sum_{i_1=1}^{nab} a_{i_1}x_{i_1}(n) + \sum_{i_1=1}^{nab}\sum_{i_2=1}^{i_1} a_{i_1i_2}x_{i_1}(n)x_{i_2}(n) \\ + \cdots + \sum_{i_1=1}^{nab}\sum_{i_2=1}^{i_1}\cdots\sum_{i_l=1}^{i_{l-1}} a_{i_1i_2\ldots i_l}x_{i_1}(n)x_{i_2}(n)\ldots x_{i_l}(n) + \varepsilon(n), \end{aligned} \tag{1.40}$$

where $nab = na + nb$,

$$x_j(n) = \begin{cases} u(n-j) & \text{if } \ 1 \leqslant j \leqslant nb \\ y(n-j+nb) & \text{if } \ nb < j \leqslant nab. \end{cases} \tag{1.41}$$

The number of parameters $M$ in (1.40) increases strongly as $nab$ or $l$ grow [123]:

$$M = \frac{(l+nab)!}{l!nab!}. \tag{1.42}$$

The large model complexity of the Kolmogorov-Gabor model restricts its practical applicability and leads to reduced polynomial models containing only selected terms of (1.40). Note that (1.40) has the NARX structure. The other nonlinear structures such as NFIR, NAR, NMA, NARMA, NARX, NARMAX, NBJ can be obtained via a relevant redefinition of $x_j(n)$.

**Nonlinear orthonormal basis function models (NOBF)**. The main disadvantage of both the FIR and NFIR models is that many parameters may be needed to describe a system adequately if its impulse response decays slowly. This disadvantage can be reduced by introducing linear filters which incorporate prior knowledge about process dynamics. Orthonormal Laguerre and Kautz filters, which have orthonormal impulse responses, are commonly applied. The Laguerre filter can be described with only one parameter $\alpha$, a real pole

$$L_k(q) = \frac{1}{q-\alpha}\left(\frac{1-\alpha q}{q-\alpha}\right)^{k-1}. \tag{1.43}$$

Therefore, this kind of filters is suitable for modelling systems with well-damped behavior. For systems with resonant behavior, Kautz filters are suitable as they have a complex pole pair. In practice, estimates of a system dominant pole or dominant conjugate complex poles are used. The regression vector for the NOBF model has the form

$$\mathbf{x}(n) = \big[L_1(q)\ L_2(q)\ \ldots\ L_r(q)\big]^T. \tag{1.44}$$

### 1.1.3 Series-parallel and parallel models

Models of dynamic systems can be used in two basic configurations: a prediction configuration or a simulation configuration. The prediction configuration permits the prediction of future system outputs based on past system inputs and outputs. Examples of the prediction configuration are the FIR, AR, ARX models in the linear case, and the NFIR, NOBF, NAR, NARX models in the nonlinear case. In the simulation configuration, future system outputs are also predicted but only on the basis of past system inputs without employing past system outputs. The OE, BJ, and NOE, NBJ models are examples of the simulation configuration. In the system identification literature, the one-step prediction configuration is called a series-parallel model and the simulation configuration is called a parallel model [121, 123]. Parallel models, being dynamic systems themselves, are also called recursive models as their mathematical description has the form of difference equations. In contrast to parallel models, series-parallel models are described by algebraical equations. In the context of neural networks, these models are called feedforward ones. In the identification process, model parameters are calculated in such a way so as to minimize a chosen cost function dependent on the identification error $e(n)$. The two model configurations above entail two different definitions of the identification error. For the series-parallel model, the identification error is called the equation error, for the parallel model – the output error.

### 1.1.4 State space models

The extension of a linear state space model to the nonlinear case results in the following nonlinear state space model:

$$\mathbf{x}(n+1) = \mathbf{h}\big(\mathbf{x}(n), \mathbf{u}(n)\big) + \boldsymbol{\upsilon}(n), \tag{1.45}$$

$$\mathbf{y}(n) = \mathbf{g}\big(\mathbf{x}(n)\big) + \boldsymbol{\varepsilon}(n), \tag{1.46}$$

where $\mathbf{x}(n) \in \mathbb{R}^{nx}$, $\mathbf{u}(n) \in \mathbb{R}^{nu}$, $\boldsymbol{\upsilon}(n) \in \mathbb{R}^{nx}$, $\mathbf{y}(n) \in \mathbb{R}^{ny}$, $\boldsymbol{\varepsilon}(n) \in \mathbb{R}^{ny}$. If the system state $\mathbf{x}(n)$ is available for measurement, the identification is equivalent to the determination of the vector functions $\mathbf{h}(\cdot)$ and $\mathbf{g}(\cdot)$:

$$\hat{\mathbf{x}}(n+1) = \mathbf{h}\big(\mathbf{x}(n), \mathbf{u}(n)\big), \tag{1.47}$$

$$\hat{\mathbf{y}}(n) = \mathbf{g}\big(\mathbf{x}(n)\big). \tag{1.48}$$

The model (1.47), (1.48) is of the series-paralel type. In practice, at least some of state variables are unknown and they have to be estimated. If no state variables are measured, simultaneous estimation of system states and the determination of the functions $\mathbf{h}(\cdot)$ and $\mathbf{g}(\cdot)$ is required. The high complexity of such a task is the main reason for the dominance of the much simpler input-output approaches [123].

### 1.1.5 Nonlinear models composed of sub-models

Except for artificial "phenomena" created by mathematical equations, models are always different from modelled phenomena [131]. Mathematical models are simply equations which are derived on the basis of the first principles and/or the experiment data and they are different from the underlaying phenomena of a nonmathematical nature. In general, models can be characterized by their time resolution and granularity where granularity specifies the level of details included into the model.

Up to now, we have considered models which are classified as black box models [148]. These models are based on the measurement data only, i.e., their parameters and structure are determined through experiments. Typically, the parameters of black box models have no interpretation in terms of physical, chemical, biological, economical, and other laws.

Another class of models are white box models as they are completely derived from the underlying principal laws. Even if some of the parameters are estimated from data, a model is included into this class as well. In contrast to black box models, the parameters of white box models have a clear interpretation [94].

Gray box models combine features of the white and black box models. They are derived both on the basis of the underlying laws and the measurement data. For example, the system structure can be determined utilizing a priori knowledge about the system nature, and system parameters can be estimated from data.

Nonlinear models of a given internal structure composed of sub-models, referred to as also block-oriented models, are members of the class of gray box models. Wiener and Hammerstein models, shown in Figs 1.1 – 1.2, are two

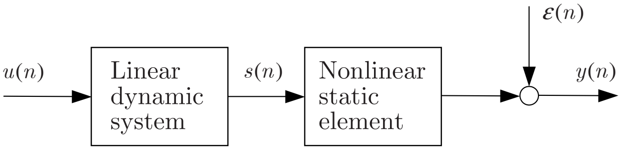

**Fig. 1.1.** SISO Wiener system

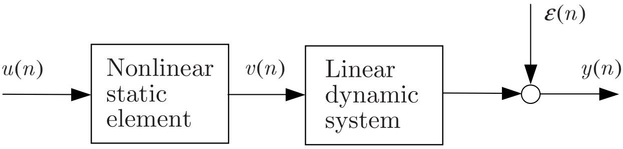

**Fig. 1.2.** SISO Hammerstein system

well-known examples of such models, composed of sub-models [61, 108, 109]. Both of them contain a linear dynamic system and a nonlinear static element in a cascade. While in the Wiener system the nonlinear element follows the linear dynamic system, in the Hammerstein model, both of these sub-models are connected in reverse order. The Wiener model is given by

$$y(n) = f\left(\frac{B(q^{-1})}{A(q^{-1})}u(n)\right) + \varepsilon(n), \tag{1.49}$$

where $f(\cdot)$ denotes the nonlinear function describing the nonlinear element, and $B(q^{-1})/A(q^{-1})$ is the pulse transfer function of the linear dynamic system. With the same notation, the Hammerstein model can be expressed as

$$y(n) = \frac{B(q^{-1})}{A(q^{-1})}f\big(u(n)\big) + \varepsilon(n). \tag{1.50}$$

The multi-input multi-output (MIMO) Wiener model can be described by the following equation:

$$\mathbf{y}(n) = \mathbf{f}\big(\mathbf{s}(n)\big) + \boldsymbol{\varepsilon}(n), \tag{1.51}$$

where $\mathbf{f}(\cdot) : \mathbb{R}^{ns} \to \mathbb{R}^{ny}$ is a nonzero vector function, $\mathbf{y}(n) \in \mathbb{R}^{ny}$, $\boldsymbol{\varepsilon}(n) \in \mathbb{R}^{ny}$. The output $\mathbf{s}(n)$, $\mathbf{s}(n) \in \mathbb{R}^{ns}$, of the MIMO linear dynamic system is

$$\mathbf{s}(n) = -\sum_{m=1}^{na} \mathbf{A}^{(m)}\mathbf{s}(n-m) + \sum_{m=1}^{nb} \mathbf{B}^{(m)}\mathbf{u}(n-m), \tag{1.52}$$

where $\mathbf{u}(n) \in \mathbb{R}^{nu}$, $\mathbf{A}^{(m)} \in \mathbb{R}^{ns\times ns}$, $\mathbf{B}^{(m)} \in \mathbb{R}^{ns\times nu}$. In a similar way, the MIMO Hammerstein model can be obtained by connecting a MIMO linear dynamic system with a MIMO static nonlinear element – Fig. 1.4. The output of the MIMO Hammerstein model is

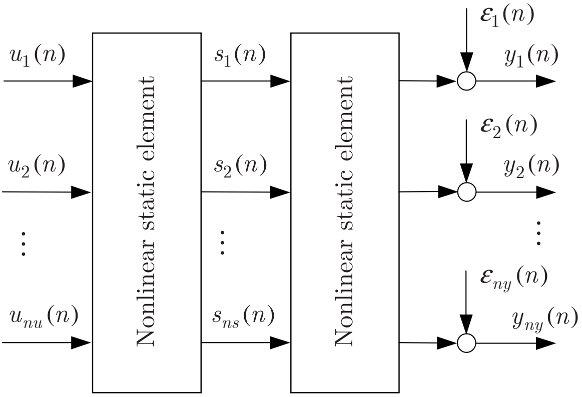

**Fig. 1.3.** MIMO Wiener system

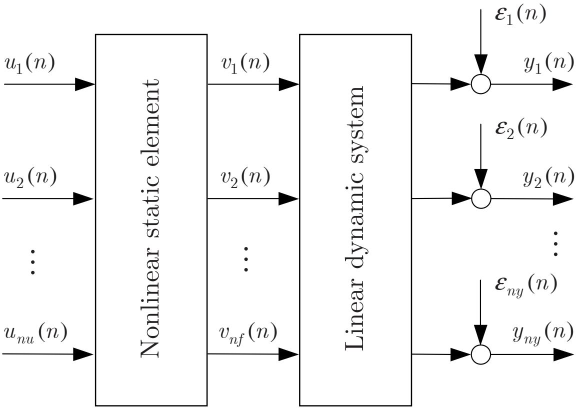

**Fig. 1.4.** MIMO Hammerstein system

$$\mathbf{y}(n) = -\sum_{m=1}^{na} \mathbf{A}^{(m)}\mathbf{y}(n-m) + \sum_{m=1}^{nb} \mathbf{B}^{(m)}\mathbf{f}\big(\mathbf{u}(n-m)\big) + \boldsymbol{\varepsilon}(n), \quad (1.53)$$

where $\mathbf{A}^{(m)} \in \mathbb{R}^{ny\times ny}$, $\mathbf{B}^{(m)} \in \mathbb{R}^{ny\times nf}$, $\mathbf{f}(\cdot) : \mathbb{R}^{nu} \to \mathbb{R}^{nf}$ is a nonzero vector function, $\mathbf{u}(n) \in \mathbb{R}^{nu}$.

Note that the definitions (1.51), (1.52), and (1.53) describe models with a coupled static part and coupled dynamics. We can also consider models with an uncoupled static part and coupled dynamics or a coupled static part and uncoupled dynamics as special cases of these general forms.

The general Wiener model considered by Sieben [147] is a SISO (single-input single-output) model in which a nonlinear static MISO (multi-input single-output) element follows a linear SIMO (single-input multiple-output) dynamic

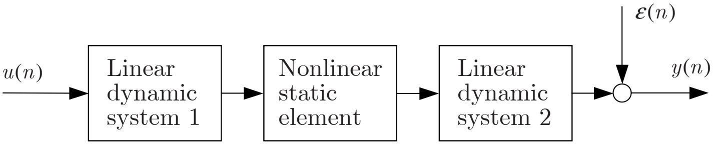

**Fig. 1.5.** SISO Wiener-Hammerstein system

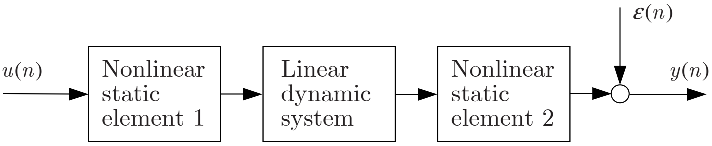

**Fig. 1.6.** SISO Hammerstein-Wiener system

system. Another structure, known as the Uryson model [40], consists of several Hammerstein models in parallel, each path having the same input and with several outputs summed.

More complicated structures arise through the interconnection of three submodels in a cascade. In this way, structures called Wiener-Hammerstein (Fig. 1.5) and Hammerstein-Wiener models (Fig. 1.6) can be obtained. The Wiener-Hammerstein structure is given by

$$y(n) = \frac{B_2(q^{-1})}{A_2(q^{-1})} f\left[\frac{B_1(q^{-1})}{A_1(q^{-1})} u(n)\right] + \varepsilon(n), \tag{1.54}$$

where $B_1(q^{-1})/A_1(q^{-1})$ and $B_2(q^{-1})/A_2(q^{-1})$ are pulse transfer functions of the first and the second linear dynamic system, respectively. The identification of Wiener-Hammerstein systems with correlation methods was studied by Billings and Fakhouri [17] and Hunter and Korenberg [71]. A recursive identification method for the MISO Wiener-Hammerstein model was proposed by Boutayeb and Darouach [21]. Bloemen *et al.* [19] considered the application of Hammerstein-Wiener models to the predictive control problem.

Bai [7] developed a two-stage identification algorithm for Hammerstein-Wiener systems in which optimal parameter estimates of both the nonlinear elements and the linear dynamic system are obtained using the RLS algorithm followed by singular value decomposition of two matrices. The algorithm is convergent in the absence of noise and convergent with the probability 1 in the presence of white noise.

Recently, the identification of Hammerstein-Wiener systems was also studied by Bai [9]. In the Hammerstein-Wiener model, two nonlinear blocks, described by the functions $f_1(\cdot)$ and $f_1(\cdot)$, are separated by a linear dynamic system:

$$y(n) = f_2\left[\frac{B(q^{-1})}{A(q^{-1})} f_1\big(u(n)\big)\right] + \varepsilon(n). \tag{1.55}$$

### 1.1.6 State-space Wiener models

A state-space description of MIMO Wiener systems with $nu$ inputs, $ny$ outputs, $nx$ states variables, and $ns$ internal variables has the form [113, 158, 164]

$$\mathbf{x}(n+1) = \mathbf{A}\mathbf{x}(n) + \mathbf{B}\mathbf{u}(n), \tag{1.56}$$

$$\mathbf{s}(n) = \mathbf{C}\mathbf{x}(n) + \mathbf{D}\mathbf{u}(n), \tag{1.57}$$

$$\mathbf{y}(n) = \mathbf{f}\big(\mathbf{s}(n)\big) + \boldsymbol{\varepsilon}(n), \tag{1.58}$$

where $\mathbf{x}(n) \in \mathbb{R}^{nx}$, $\mathbf{u}(n) \in \mathbb{R}^{nu}$, $\mathbf{s}(n) \in \mathbb{R}^{ns}$, $\mathbf{y}(n) \in \mathbb{R}^{ny}$, and $\mathbf{f}(\cdot) : \mathbb{R}^{nf} \rightarrow \mathbb{R}^{ny}$ is a nonzero vector function, $\boldsymbol{\varepsilon}(n) \in \mathbb{R}^{ny}$ is a zero-mean stochastic process. The parallel representation of the state-space Wiener model has the form

$$\hat{\mathbf{x}}(n+1) = \mathbf{A}\hat{\mathbf{x}}(n) + \mathbf{B}\mathbf{u}(n), \tag{1.59}$$

$$\hat{\mathbf{s}}(n) = \mathbf{C}\hat{\mathbf{x}}(n) + \mathbf{D}\mathbf{u}(n), \tag{1.60}$$

$$\hat{\mathbf{y}}(n) = \mathbf{f}\big(\hat{\mathbf{s}}(n)\big). \tag{1.61}$$

Replacing the model state $\hat{\mathbf{x}}(n)$ in (1.59) and (1.60) with the system state $\mathbf{x}(n)$ results in the series-parallel representation of the state-space Wiener model:

$$\hat{\mathbf{x}}(n+1) = \mathbf{A}\mathbf{x}(n) + \mathbf{B}\mathbf{u}(n), \tag{1.62}$$

$$\hat{\mathbf{s}}(n) = \mathbf{C}\mathbf{x}(n) + \mathbf{D}\mathbf{u}(n), \tag{1.63}$$

$$\hat{\mathbf{y}}(n) = \mathbf{f}\big(\hat{\mathbf{s}}(n)\big). \tag{1.64}$$

### 1.1.7 State-space Hammerstein models

A state space MIMO Hammerstein model can be described by the following equations [157]:

$$\mathbf{x}(n+1) = \mathbf{A}\mathbf{x}(n) + \mathbf{B}\mathbf{v}(n), \tag{1.65}$$

$$\mathbf{y}(n) = \mathbf{C}\mathbf{x}(n) + \mathbf{D}\mathbf{v}(n) + \boldsymbol{\varepsilon}(n), \tag{1.66}$$

$$\mathbf{v}(n) = \mathbf{f}\big(\mathbf{u}(n)\big), \tag{1.67}$$

where $\mathbf{x}(n) \in \mathbb{R}^{nx}$, $\mathbf{u}(n) \in \mathbb{R}^{nu}$, $\mathbf{y}(n) \in \mathbb{R}^{ny}$, $\boldsymbol{\varepsilon}(n) \in \mathbb{R}^{ny}$ is a zero-mean stochastic process, $\mathbf{v}(n) \in \mathbb{R}^{nf}$, and $\mathbf{f}(\cdot)$: $\mathbf{f}(\cdot) : \mathbb{R}^{nu} \rightarrow \mathbb{R}^{nf}$ is a nonzero vector function. The MIMO Hammerstein model in a parallel state-space representation has the form

$$\hat{\mathbf{x}}(n+1) = \mathbf{A}\hat{\mathbf{x}}(n) + \mathbf{B}\hat{\mathbf{v}}(n), \tag{1.68}$$

$$\hat{\mathbf{y}}(n) = \mathbf{C}\hat{\mathbf{x}}(n) + \mathbf{D}\hat{\mathbf{v}}(n), \tag{1.69}$$

$$\hat{\mathbf{v}}(n) = \mathbf{f}\big(\mathbf{u}(n)\big). \tag{1.70}$$

Replacing the model state $\hat{\mathbf{x}}(n)$ in (1.68) and (1.69) with the system state $\mathbf{x}(n)$, we obtain the series-parallel representation of the state-space Hammerstein model:

$$\hat{\mathbf{x}}(n+1) = \mathbf{A}\mathbf{x}(n) + \mathbf{B}\hat{\mathbf{v}}(n), \tag{1.71}$$

$$\hat{\mathbf{y}}(n) = \mathbf{C}\mathbf{x}(n) + \mathbf{D}\hat{\mathbf{v}}(n), \tag{1.72}$$

$$\hat{\mathbf{v}}(n) = \mathbf{f}\big(\mathbf{u}(n)\big). \tag{1.73}$$

## 1.2 Multilayer perceptron

Multilayer feedforward neural networks, also referred to as multilayer perceptrons, are the most widely known and used neural networks [60, 68, 69, 127, 144, 169]. In the multilayer perceptron (MLP), the neurons are ordered into one or more hidden layers and connected to an output layer. This type of neural network is used in Chapters 2 and 3 intensively for modelling both the nonlinear element and its inverse.

### 1.2.1 MLP architecture

The $i$th output $x_i(n)$ of the first hidden layer (Fig. 1.7) is

$$x_i(n) = \varphi\left(\sum_{j=1}^{nu} w_{ij}^{(1)} u_j(n) + w_{i0}^{(1)}\right), \tag{1.74}$$

where $w_{ij}^{(1)}$ is the $j$th weight of the $i$th neuron, $w_{i0}^{(1)}$ is the bias of the $i$th neuron, $\varphi(\cdot)$ is the activation function of hidden layer neurons, $nu$ is the number of inputs, and $u_j(n)$ is the $j$th input. Common choices of the activation function are sigmoidal functions such as the logistic function

$$\varphi(x) = \frac{1}{1+\exp(-x)} \tag{1.75}$$

and the hyperbolic tangent function

$$\varphi(x) = \tanh(x) = \frac{1-\exp(-2x)}{1+\exp(-2x)}. \tag{1.76}$$

The outputs of the first hidden layer can be connected to the successive hidden layers and finally to the output layer. In the commonly used MLP neural network with one hidden layer, the outputs $x_i(n)$, $i = 1, \ldots, M$, where $M$ is the number of hidden layer neurons, are transformed by the output layer into the outputs $y_k(n)$:

$$y_k(n) = \phi\left(\sum_{i=1}^{M} w_{ki}^{(2)} x_i(n) + w_{k0}^{(2)}\right), \tag{1.77}$$

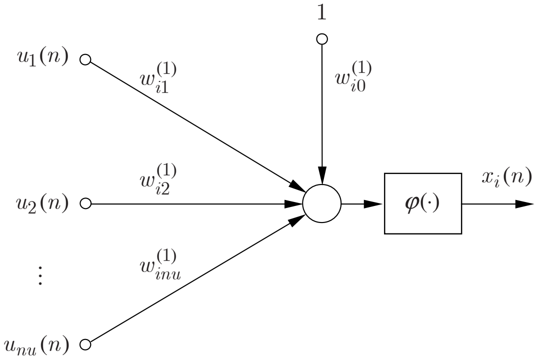

**Fig. 1.7.** The $i$th hidden layer neuron

where $w_{ki}^{(2)}$ is the $i$th weight of the $k$th output neuron, $w_{k0}^{(2)}$ is the bias of the $k$th output neuron, and $\phi(\cdot)$ is the activation function of output layer neurons. Although $\phi(\cdot)$ can be a nonlinear function, the linear activation function $\phi(x) = x$ is a typical choice.

### 1.2.2 Learning algorithms

Nonlinear optimization of neural network weights is the most common technique used for training the MLP. Using gradient-based learning methods, the cost function $J$, typically the sum of squared errors between the system output and the neural network model output, is minimized. As a result, neural network weights are adjusted along the negative gradient of the cost function. The backpropagation (BP) learning algorithm is an implementation of the gradient descent optimization method for weight updating. The backpropagation learning algorithm uses the backpropagation algorithm as a technique for the computation of the gradient of the MLP w.r.t. its weights [68, 123, 132]. In spite of its computational simplicity, training the MLP with the BP learning algorithm may cause several problems such as very slow convergence, oscillations, divergence, and the "zigzagging" effect. A large number of improvements, extensions, and modifications of the basic BP learning algorithm have been developed to circumvent this problem [60, 68, 123].

The reason for the slow convergence of the BP learning algorithm is that it operates on the basis of a linear approximation of the cost function. To achieve a significantly higher convergence rate, higher order approximations of the cost function should be used. Examples of such learning techniques are the Levenberg-Marquard method, quasi-Newton methods, conjugate gradient methods [68, 123], and the RLS learning algorithms [144].

To extract information from the training data and increase the the effectiveness of learning, some data preprocessing such as filtering, removing redundancy, and removing outliers is usually necessary.

Also, scaling the data is essential to make learning algorithms robust and decrease the learning time. The recommended scaling technique is based on removing the mean and scaling signals to the same variance [127]. Alternatively, the mean can be removed from signals and zero-mean signals scaled with respect to their maximum absolute values, to obtain values in a specified interval, e.g. $[-1, 1]$.

In general, to minimize the learning time, applying nonzero-mean input signals should be avoided. This comes from the fact that the learning time for the steepest descent algorithm is sensitive to variations in the condition number $\lambda_{max}/\lambda_{min}$, where $\lambda_{max}$ is the largest eigenvalue of the Hessjan of the cost function and $\lambda_{min}$ is its smallest nonzero eigenvalue. Experimental results, show that for nonzero-mean input signals the condition number $\lambda_{max}/\lambda_{min}$ is larger than for zero-mean input signals [68]. Note that the choice of asymmetric activation function, e.g. the logistic function, introduces systematic bias for hidden layer neurons. This has a similar effect on the condition number $\lambda_{max}/\lambda_{min}$ as nonzero-mean inputs. Therefore, to increase the convergence speed of gradient-based learning algorithms, the choice of antisymmetric activation functions, such as the hyperbolic tangent, is recommended.

### 1.2.3 Optimizing the model architecture

In practice, MLP models of real processes are of a rather large size. It is well known that models should not be too complex because they would learn noise and thus generalize badly to new data. On the other hand, they should not be too simple because they would not be capable to capture the process behavior. In the MLP with one hidden layer, the problem of architecture optimizing boils to choosing the number of hidden layer nodes and eliminating insignificant weights.

The overall model error is composed of two components – a bias error which express the systematic error caused by the restricted model flexibility and a variance error, being the stochastic error due to the restricted accuracy of parameter estimates. Both these components of the model error are in conflict, known as a bias/variance dilemma, because the bias error decreases and the variance error increases for growing model complexity. Therefore it is necessary to find a compromise – the so called bias/error tradeoff. To accomplish this, the optimization of the model architecture is necessary.

There are two groups of methods used to optimize the neural network architecture, known as network growing and network pruning. In network growing methods, new nodes or layers are added starting from a small size network until the enlarged structure meets the assumed requirements. A well-known example of a growing method is the cascade-correlation algorithm [36]. In pruning methods, the initial structure is large, then we prune it by weakening or eliminating some selected weights.

The idea of pruning is based on the assumption that there is a large amount of redundant information stored in a fully connected MLP. Network pruning is commonly accomplished by two approaches – one based on complexity regularization and the other based on removing some weights using information on second-order derivatives of the cost function.

In complexity regularization methods, the complexity penalty term is added to the cost function. The standard cost function in back-propagation learning is the mean-square error. Depending on the form of the complexity penalty term different regularization techniques can be defined such as the weight decay, the weight elimination, the approximate smoother, the Chauvin's penalty approach [60, 123]. In fact, regularization methods do not change the model structure but reduce the model flexibility by keeping some weights at their initial values or constraining their values. In this way the reduction of the variance error can be achieved at the price of the bias error.

The optimal brain damage (OBD) [107] and the optimal brain surgeon (OBS) [67] are the most widely known an used methods based on the use of information on second-order derivatives of the cost function. Both of them are used for reducing the size of the network by selectively deleting the weights. Both of them employ the second-order Taylor expansion of the cost function about the operating point – the $nw$-dimensional weight vector $\mathbf{w}^* = [w_1^* \dots w_{nw}^*]^T$ for which the cost function has a local minimum. Their objective is to find a set of weights whose removing cause the least change of the cost function. To achieve reasonable low computational complexity, only diagonal terms of the second-order Taylor expansion are included into the definition of the saliency of parameters in the OBD method. This corresponds to the assumption that the Hessjan matrix is diagonal matrix. The OBD method is an iterative procedure of the following form:

1. Train the MLP to minimum mean-square error of the cost function.
2. Compute the diagonal second-order derivatives $h_{ii}$, $i = 1, \dots, nw$, of the cost function.
3. Compute the saliencies for weights: $S_i = h_{ii}(w_i^*)^2/2$.
4. Delete some weights that have small saliencies.
5. Return to Step 1.

No such assumption about the Hessjan matrix is made in the OBS method and the OBD can be considered as a special case of the OBS.

## 1.3 Identification of Wiener systems

Many different approaches to Wiener system identification have been proposed based on correlation analysis, linear optimization, nonparametric regression, nonlinear optimization with different nonlinear models such as polynomials, neural networks, wavelets, orthogonal functions, and fuzzy sets models.

**Correlation methods.** Billings and Fakhouri used a correlation analysis approach to the identification of block-oriented systems based on the theory

of separable processes [14]. When the input is a white Gaussian signal, it is possible to separate the identification of the linear dynamic system from the identification of the nonlinear element [15, 17]. For Wiener systems, the first order correlation function $R_{uy}(k)$ of the system input $u(n)$ and the system output $y(n)$ is directly proportional to the impulse response $h(k)$ of the linear system, and the second order correlation function $R_{u^2y}(k)$ is directly proportional to the square of $h(k)$. Therefore, if $R_{uy}^2(k)$ and $R_{u^2y}(k)$ are equal except for a constant of proportionality, the system has a Wiener-type structure.
A correlation approach to the identification of direction-dependent dynamic systems using Wiener models was used by Barker *et al.* [12]. They considered Wiener systems containing the linear dynamic part with different transfer functions for increasing and decreasing system output. The Wiener system was excited with maximum-length pseudo-random or inverse maximum-length pseudo-random binary signals. The determination of Wiener model parameters was performed by matching the system and model correlation functions, outputs, and discrete Fourier transforms of the outputs.

**Linear optimization methods.** In linear optimization methods, it is assumed that a model can be parameterized by a finite set of parameters. The nonlinear element is commonly modelled by a polynomial along with a pulse transfer function model of the linear dynamic system. A Wiener model parameterized in this way is nonlinear in the parameters and parameter estimation becomes a nonlinear optimization problem. Note that the inversion of the Wiener model is a Hammerstein model whose linear part is described by the inverse transfer function and nonlinear element is described by the inverse nonlinear function. A necessary condition for such a transformation is the invertibility of the function $f(\cdot)$ describing the nonlinear part of the Wiener model. To obtain an asymptotically stable inverse model, the Wiener model should be minimum phase. The inverse Wiener model is still nonlinear in the parameters but it is much more convenient for parameter estimation as it can be transformed into the linear-in-parameters MISO form with the method proposed by Chang and Luus [27] for Hammerstein systems. The parameters of the transformed model can be calculated with the least squares method. Assuming the knowledge of $f(\cdot)$, the identification of the inverse Wiener system was considered by Pearson and Pottman [138]. To overcome the practical difficulty related to the fact that the inverse system identification approach penalizes prediction errors of the input signal $u(n)$ instead of the output signal $y(n)$, they used the weighted least squares method with weighting parameters that weight the relative importance of the error.
A polynomial inverse model of the nonlinear element and a frequency sampling filter model of the linear dynamic system

$$\hat{s}(n) = \sum_{m=0}^{n_f-1} G(jw_m)F_m(n), \tag{1.78}$$

where $\hat{s}(n)$ is the output of the linear dynamic model, $G(jw_m)$, $m = 0, \ldots, n_f - 1$, is the discrete frequency response of the linear system at $w_m = 2\pi m/n_f$, $F_m(n)$ is the output of the $m$th frequency sampling filter defined as

$$F_m(n) = \frac{1}{n_f} \frac{1 - q^{-n_f}}{1 - e^{j(2\pi m/n_f)} q^{-1}} u(n), \tag{1.79}$$

where $u(n)$ is the system input and $q^{-1}$ is the backward shift operator, was used by Kalafatis *et al.* [95]. The model output $\hat{y}(n)$ can then be expressed in the linear-in-parameters form:

$$\hat{y}(n) = \sum_{m=0}^{n_f-1} G(jw_m) F_m(n) - \sum_{k=2}^{r} \gamma_k y^k(n), \tag{1.80}$$

where $y(n)$ is the system output and $\gamma_k$, $k = 2, \ldots, r$, are the parameters of the polynomial inverse model of the nonlinear element. The parameters $G(jw_m)$ and $\gamma_k$ can be calculated with the least squares method. Unfortunately, such an approach leads to inconsistent parameter estimates for Wiener systems with additive output disturbances as the regression vector is correlated with the disturbance. To overcome this problem, an iterative algorithm was proposed [96], which consists of the following three steps: First, parameter estimates are calculated using the least squares method. Then using the obtained estimates, predicted system outputs are calculated. Finally, to calculate corrected parameter estimates, the predicted system outputs are employed in another estimation step using the least squares method. The above estimation procedure is repeated until the parameter estimates converge to constant values.

The recursive least squares scheme is used in the orthonormal basis function-based identification method proposed by Marciak *et al.* [116]. They use a noninverted model of the linear system composed of Laguerre filters and an inverse polynomial model of the nonlinear element.

In the frequency approach to the identification of Wiener systems of Bai [10], the phase estimate of the linear dynamic system output is determined based on the discrete Fourier transform (DFT) of the filtered system output. Having the phase estimate, the structure of nonlinearity can be determined from the graph of the system output versus an estimated linear dynamic system output and approximated by a polynomial.

The identification of Wiener systems based on a modified series-parallel Wiener model, defined by a non-inverted pulse transfer model of the linear element and an inverse polynomial model of the nonlinear element, can be performed with the method proposed by Janczak [80, 83]. The modified series-parallel model is linear in the parameters and its parameters are calculated with the least squares method. The method requires the nonlinear function $f(\cdot)$ to be invertible and the linear term of the polynomial model to be nonzero. However, in the case of additive output noise, direct application of this approach results in inconsistent parameter estimates. As a remedy against such a situation, a combined least squares and instrumental variables estimation procedure is proposed. A detailed description of this method, along with its extension to the identification of Wiener systems without the linear term of the nonlinear characteristic, is given in Chapter 4.

Parameter estimation of Wiener systems composed of a finite impulse response (FIR) model of the linear system and an inverse polynomial model of the nonlinear element was considered by Mzyk [119]. To obtain consistent parameter estimates, the instrumental variables method is employed with instrumental variables defined as a sum of powered input signals and some tuning constants selected by the user.

The identification of MIMO Wiener systems with the use of basis functions for the representation of both the linear dynamic system and the nonlinear element was proposed by Gómez and Baeyens [44]. It is assumed that the nonlinear element is invertible and can be described by nonlinear basis functions. The MIMO linear dynamic system is represented by rational orthonormal bases with fixed poles (OBFP). Special cases of OBFP are the FIR, Laguerre, and Kautz bases. Under the above assumptions, the MIMO Wiener model can be transformed into the linear-in-parameters form and parameter estimates of the transformed model that minimize the quadratic cost function on prediction errors are calculated using the least squares method. To calculate matrix parameters of the nonlinear element from the parameter estimates of the transformed model, the singular value decomposition technique is applied.

**Nonparametric regression methods.** Parametric regression methods are based on a restrictive assumption concerning the class of nonlinear functions. The kernel nonparametric regression approach, which considerably enlarges the class of nonlinearities identified in Wiener systems, was introduced by Greblicki [46] and studied further in [48]. Next, the idea of nonparametric regression was advanced by employing orthogonal series for recovering the inverse of the nonlinear characteristic [47]. For nonparametric regression algorithms that use trigonometric, Legendre and Hermite orthogonal functions, pointwise consistency was shown and the rates of convergence were given. Greblicki also proposed and analyzed recursive identification algorithms based on the kernel regression for both discrete-time [51] and continuous-time Wiener systems [50, 52].

**Nonlinear optimization methods.** The identification of Wiener systems using prediction error methods was discussed in [85, 88, 128, 165, 166]. Wigren [165] analyzed recursive Gauss-Newton and stochastic gradient identification algorithms assuming that system nonlinearity is known a priori. He established conditions for local and global convergence of parameter estimates to the true system parameters at correlated measurement disturbances. Another recursive prediction error method, proposed by Wigren [166], estimates the parameters of a pulse transfer function model of the linear system and a piecewise linear model of the nonlinear element. With the technique of a linearized differential equation, the local convergence of estimates to the system parameters is proved. It is also shown that the input signal should be such that there is energy in the whole range of piecewise linear approximation.

Vörös [162] used a parametric model to describe Wiener systems with a special kind of discontinuous nonlinear element – a piecewise-linear function with a preload and a dead zone. The pure preload, dead zone, and a two-segment piecewise-linear asymmetric nonlinearities are special cases of the general discontinuous nonlinearity. He proposed an identification method that uses a description of the nonlinear element based on the key separation principle. Such a formulation of the problem makes it possible to transform the nonlinear element model into the pseudolinear-in-parameters form. Parameter estimation is conducted iteratively as the model contains three internal variables that are unmeasurable. The iterative procedure is based on the use of parameter estimates from the preceding step to estimate these unmeasurable internal variables.

Another approach to the identification of Wiener systems with the assumed forms of hard-type nonlinearity parameterized by a single unknown parameter $a$ was proposed by Bai [8]. Examples of nonlinearities of such type are the saturation, preload, relay, dead zone, hysteresis-relay, and the hysteresis. To find the unknown parameter $a$, the separable least squares method is used, in which the identification problem is transformed into a one-dimensional minimization problem. As the cost function is one-dimensional, global search methods can be applied to find its minimum. Alternatively, it is also possible to find the minimum directly from the plot of the cost function versus $a$. Having found the optimal estimate of $a$, the parameters of the linear dynamic system can be estimated using the least squares method. For this approach, the conditions under which the strong consistency of parameter estimates can be achieved are given. Although the separable least squares method can be extended to the two-dimensional case easily, its extension to the case of nonlinearities parameterized by a larger number of parameters is more complicated.

An iterative scheme for the identification of Wiener systems with the prediction error method was proposed by Norquay *et al.* [128]. In this approach, the nonlinear element is modelled by a polynomial. To calculate the approximated Hessjan, the Levenberg-Marquardt method is used along with the calculation of the gradient via the simulation of sensitivity models. A recursive version of this method was used in [85, 88].

The pseudolinear regression algorithm, described by Janczak [86, 88], can be obtained from the prediction error scheme if the dependence of both the delayed and powered output signals of the linear model on model parameters is ignored. In this way, the model is treated as a linear-in-parameters one. Such a simplification reduces computational complexity of the algorithm at the price of gradient approximation accuracy.

A neural network-based method for the identification of Wiener systems was developed by Al-Duwaish [3]. In this method, the linear dynamic system and the nonlinear element are identified separately. First, the parameters of the linear system, described by a linear difference equation, are estimated with the recursive least squares (RLS) algorithm based on a response to a small input signal which ensures a linear perturbation of the nonlinear element. Then having the linear system identified, the backpropagation learning algorithm is applied to train a multilayer neural network model of the nonlinear element. This step is performed with another response to an increased input signal which perturbs the nonlinear element nonlinearly.

The identification of a MISO Wiener system was studied by Ikonen and Najim [72]. Based on the input-output data from a pump-valve pilot system, a neural network Wiener model was trained with the Levenberg-Marquardt method.

Visala *et al.* [159] used a MIMO Wiener model for the modelling of the chromatographic separation process. The linear dynamic part of their model is composed of Laguerre filters while the MIMO feedforward neural network is employed as a static nonlinear mapping. It is assumed that the MIMO linear dynamic model is decoupled. Parameter estimation is reduced to training the nonlinear part of the model with the Levenberg-Marquardt method as the linear dynamics is assumed to be known. This means that suitable values of Laguerre parameters are selected on the basis of a priori information.

The identification of a nonlinear dynamic system using a model composed of a SIMO dynamic system followed by a MISO nonlinear element was studied by Alataris *et al.* [1]. Their methodology employs a multilayer perceptron with a single hidden layer and polynomial activation functions as a model of static nonlinearity. The inputs to this models are outputs of a Laguerre filter bank used as a model of the linear dynamic system. The parameters of the neural network model are adjusted by means of the gradient descent method. The choice of a real pole, being the only degree of freedom of Laguerre filters, is based on a trial and error rule. Applying the polynomial activation function permits an easy transition between the neural network and Volterra series models.

A neural network model of both the linear dynamic system and the nonlinear element is used to describe SISO Wiener systems in the single step sequential procedure proposed by Janczak [89]. For more details about this approach and its extension to the MIMO case refer to Chapter 2.

The identification of Wiener systems using a genetic method, revealing a global optimization property, was studied by Al-Duwaish [5]. In this approach, the parameters of a Wiener model composed of a pulse transfer function model of the linear part and a polynomial model of the nonlinear element are calculated. The pulse transfer function model is defined in the pole-zero form. With the fitness function defined as a sum of squared errors between the system and model outputs and by performing genetic operations of cross-over, mutation, and selection, new generations of solutions are generated and evaluated until predetermined accuracy of approximation is achieved. A parallel neural network Wiener model was also trained using recursive evolutionary programming with a time-dependent learning rate by Janczak and Mrugalski [93].

## 1.4 Identification of Hammerstein systems

As in the case of Wiener systems, discussed in Section 1.3, we will review briefly various identification methods for Hammerstein systems.

**Correlation methods.** For a white Gaussian input, the computation of the cross-correlation function makes it possible to decouple the identification of Hammerstein systems and identify the linear dynamic system and the nonlinear element separately [14, 15, 16, 17]. First, the impulse response $h(k)$ of the linear dynamic system is estimated using the correlation technique. The first order correlation function $R_{uy}(k)$ is directly proportional to the impulse response, $R_{uy}(k) = \alpha h(k)$. Then if necessary, the parameters of the pulse transfer function $\mathcal{Z}[\alpha h(k)] = B(q^{-1})/A(q^{-1})$ can be calculated from the impulse response easily. Having an estimate of $\alpha h(k)$ available, the parameters of a polynomial model of the nonlinear element can be calculated with the least squares algorithm [16].
For Hammerstein systems, the second order correlation function $R_{u^2y}(k)$ is directly proportional to the impulse response of the linear element and provides a convenient test of the system structure. If the first and second order correlation functions are equal except for a constant of proportionality, the system must have the structure of a Hammerstein model.

**Linear optimization methods.** In contrast to correlation methods, which use a nonparametric model of the linear dynamic system and a parametric model of the nonlinear element, linear optimization methods use parametric representations for both parts of the Hammerstein system.
The parameters of a pulse transfer function representation of the linear element and a polynomial model of the nonlinear element can be estimated using the iterative least squares method proposed by Narendra and Gallman [120]. The method is based on an alternate adjustment of the parameters of the linear and nonlinear parts of the model.
Another iterative approach was proposed by Haist *et al.* [63] for Hammerstein systems with correlated additive output disturbances. This method, being an extension of the method of Narendra and Gallman, overcomes its obvious drawback of biased estimates via estimation of both parameters of the Hammerstein model and a linear noise model.
The transformation of the SISO Hammerstein model into the MISO form makes it possible to estimate the parameters of the transformed model utilizing the least squares method noniteratively [27]. In spite of its simplicity and the elegant form of a one step solution, this method has an inconvenience in the form of redundancy in the calculation of the parameters of the nonlinear element. More precisely, $nb$ different sets of these parameters can be calculated from the parameters of the transformed MISO model, where $nb$ is the nominator order of the pulse transfer function.
With instrumental variables defined as linear filter outputs and powers of

system inputs, instrumental variables methods were studied by Stoica and Söderström [152]. They proved that the instrumental variables approach gives consistent parameter estimates under mild conditions.

An algorithm that uses an OBFP model of the MIMO linear dynamic system and a nonlinear basis function model of the MIMO nonlinear element was proposed by Gómez ana Baeyens [44]. In this approach, the MIMO Hammerstein model is transformed into the linear-in-parameters form and its parameters are calculated using the least squares method. To calculate matrix parameters of the nonlinear element from parameter estimates of the transformed model, the singular value decomposition technique is applied. The algorithm provides consistent parameter estimates under weak assumptions about the persistency of the excitation of system inputs.

Bai [11] proposed a two-step algorithm that decouples the identification of the linear dynamic system from the identification of the nonlinear element. In the first step, the algorithm uses a pseudo-random binary sequence (PRBS) input to identify the linear dynamic system. With the PRBS signal, the effect of nonlinearity can be eliminated as any static nonlinear function can be completely characterized by a linear function under the PRBS input. Therefore, any identification method of linear systems can be applied to obtain a linear dynamic model. The identification of the nonlinear element is made in the other step. As the PRBS signal assumes only two values, a new input signal that is rich enough, e.g., a pseudo-random sequence of a uniform distribution is used to identify the nonlinear element. An important advantage of this decoupling technique is that the asymptotic variance of the estimate of the Hammerstein system transfer function is equal to the asymptotic variance of the estimate of the system transfer function in the linear case [126].

A piecewise linear model of the nonlinear element and a pulse transfer function model of the linear dynamic system are used in the identification scheme proposed by Giri et al. [43]. The Hammerstein model, defined in this way, is transformed into the linear-in-parameters form and parameter estimation is performed using a recursive gradient algorithm augmented by a parameter projection. To ensure the convergence of the model to the true system, a persistently exciting input sequence is generated.

**Nonparametric regression methods.** In parametric regression methods, it is assumed that the nonlinear characteristic belongs to a class that can be parameterized by a finite set of parameters. Clearly, such an assumption is a very restrictive one. For example, a commonly used polynomial representation of the nonlinear element excludes typical discontinuous characteristics such as dead-zone limiters, hard-limiters, and quantizers. A very large class of nonlinear functions, including all measurable $L_2$ functions, can be identified using nonparametric regression. A nonparametric regression approach to the identification of Hammerstein systems was originally proposed by Greblicki and Pawlak [55] and studied further in [56, 57, 103]. Kernel regression identification procedures for different block-oriented systems, including Hammerstein ones, were also studied by Krzyżak and Partyka [105]. Nonparametric regression methods comprise two separate steps. First, the impulse response function of the dynamic system is estimated with a standard correlation method. Then the characteristic of the nonlinear element $f(\cdot)$ is estimated as a kernel regression estimate

$$\hat{f}\big(u(n)\big) = \frac{\sum\limits_{k=0}^{N-1} y(k+1) K\left(\dfrac{u(n)-u(k)}{l(N)}\right)}{\sum\limits_{k=0}^{N-1} K\left(\dfrac{u(n)-u(k)}{l(N)}\right)}, \tag{1.81}$$

where $N$ is the number of measurements, $K(\cdot)$ is the kernel function, $l(\cdot)$ is a sequence of positive numbers. In this definition, 0/0 is understood as zero.
It can be shown that nonparametric regression estimates converge to the true characteristic of the nonlinear element as the number of measurements $N$ tends to infinity. Greblicki and Pawlak also showed that for sufficiently smooth characteristics, the rate of convergence is $O(N^{-2/5})$. All of the above identification algorithms recover the nonlinear characteristic in a nonrecursive manner. Two other algorithms based on kernel regression estimates, proposed by Greblicki and Pawlak [58], allow one to identify the nonlinear characteristic recursively.
Another class of nonparametric methods uses orthogonal expansions of the nonlinear function. Greblicki [45] considered a Hammerstein system driven by a random white input, with the system output disturbed by random a white noise, and proposed two nonparametric procedures based on the trigonometric and Hermite orthogonal expansions. Both of these algorithms converge to the nonlinear characteristic of the system in a pointwise manner, and the integrated error converges to zero. The pointwise convergence rate is $O(N^{-(2q-1)/4q})$ in probability, where $q$ is the number of derivatives of the nonlinear characteristic. The identification of Hammerstein systems by algorithms based on the Hermite series expansion with the number of terms depending nonlinearly on input-output measurements was considered by Krzyżak *et al.* [106]. In this approach, the system is driven by the stationary white noise, and the linear and nonlinear components are estimated simultaneously. The application of the Fourier series estimate to identify nonlinearities in block-oriented systems, including Hammerstein ones, was studied by Krzyżak [102, 104]. For nonlinear functions and input signal densities having finite Fourier series expansions, such an approach has two advantages over kernel regression ones – higher computational efficiency and higher rates of convergence. Recovering the nonlinear function using a Legendre polynomial-based method with an adaptively selected number of terms was studied by Pawlak [136]. It was shown that the estimate of $f(\cdot)$ is globally consistent and the rates of convergence were established. Greblicki and Pawlak [59] considered also the identification of Hammerstein systems with Laguerre polynomials.
An alternative to nonparametric methods based on orthogonal expansions

are algorithms that employ multiresolution approximation. In the context of Hammerstein systems identification, the Haar multiresolution analysis was first used by Pawlak and Hasiewicz [137]. This idea was studied further by Hasiewicz [64, 65, 66]. Haar multiresolution approximation algorithms converge pointwise. Their forms and convergence conditions are the same for white and correlated additive output noise. Their another advantage is the faster convergence rate in comparison with other nonparametric identification algorithms that use orthogonal series expansions.

The idea of the identification of nonlinearities in block-oriented systems with Daubechies wavelets was studied by Śliwiński and Hasiewicz [155]. As the lack of a closed analytical form makes Daubechies wavelets practically unapplicable, they used an estimation procedure that employs approximations that are easy to compute.

**Nonlinear optimization methods.** A prediction error approach to the identification of Hammerstein systems was discussed by Eskinat *et al.* [34]. Contrary to the least squares approach used by Chang and Luus [27], prediction error methods make it possible to estimate the parameters of a pulse transfer function of the linear system and a polynomial nonlinear characteristic element directly, without any transformation of the parameters, and there is no problem of parameter redundancy. The prediction error method uses the Levenberg-Marquardt method to approximate the Hessjan of the sum-squared cost function. Neglecting the fact that a model is nonlinear in the parameters and treating it as a linear one leads to pseudolinear regression methods. An example of such an approach is the method proposed by Boutayeb and Darouach [21], in which the parameters of a MISO Hammerstein system with the output disturbed additively by correlated noise are estimated recursively.

An identification method for Hammerstein systems which uses a two-segment model of the nonlinear element, composed of separate polynomial maps for positive and negative inputs, was proposed by Vörös [161]. This method also employs models in the pseudolinear form to calculate parameters iteratively. The idea of the method is based on splitting the nonlinear characteristic, which can be described accurately by a polynomial of a high order, into two segments that can be approximated with polynomials of a much lower order. A similar technique was also applied by Vörös [160] for the identification of Hammerstein systems with the nonlinear element described by a discontinuous function.

A neural network approach can be applied to the identification of Hammerstein systems with nonlinear elements described by a continuous function. The identification of Hammerstein systems with neural network models was considered by Su and McAvoy [153]. They used the steady-state and the transient data to train a neural network Hammerstein model. In this approach, a neural network model of the nonlinear element and a model of the linear dynamic system are trained separately. First, the neural network is trained with the backpropagation (BP) learning algorithm on the steady-state data. After the training, the neural network serves as a nonlinear operator. The linear dynamic model is then trained on the set of transient data employing system input variables transformed by the nonlinear operator as inputs. Considering two basic configurations of the linear dynamic model, i.e., the series-parallel and the parallel one it is shown that the gradients of the cost function can be obtained via the backpropagation method for the series-parallel model, and the backpropagation through time method for the parallel one. Having the gradient calculated, a steepest descent or a conjugate gradient algorithm is suggested to adjust the model parameters.

The identification of SISO Hammerstein systems by multilayered feedforward neural networks was also studied by Al-Duwaish *et al.* [3]. They considered a parallel neural network Hammerstein model trained recursively using the BP learning algorithm for training a model of the nonlinear element and the recursive least squares algorithm (RLS) for training the linear dynamic model. In this approach, partial derivatives of the squared cost function are calculated in an approximate way without taking into account the dependence of past model outputs on model parameters. This corresponds to Equations (3.36) – (3.38). Moreover, the fact that partial derivatives of the model output w.r.t. the weights of the nonlinear element model depend not only on the actual but also on past outputs of the nonlinear element model is not taken ino account. Both of these simplifications of gradient calculation reduce computational complexity of training at the price of reduced accuracy of gradient calculation and may result in a decreased convergence rate. The extension of the RLS/BP algorithm to a MIMO case is discussed in [4].

A genetic approach to the identification of Hammerstein systems is considered in [5]. In this approach, the pole-zero form of the linear dynamic system and the nonlinearity of a known structure but unknown parameters are identified.

In general, SISO neural network Hammerstein models can be represented by a multilayer perceptron model of the nonlinear element and a linear node with two tapped delay lines used as a model of the linear system [73]. Both the series-parallel and parallel models can be considered. They have a similar architecture, the only difference is the feedback connection in the parallel model [100]. For the series-parallel model, the gradient of its output with respect to model parameters can be obtained using the computationally effective backpropagation algorithm. The calculation of the gradient in parallel models can be made with the sensitivity method or the backpropagation through time method [75, 77, 90]. As the Hammerstein model contains a linear part, it is also possible to use non-homogeneous algorithms that combine the recursive least squares or recursive pseudolinear regression algorithms with all of the above-mentioned methods [75]. The neural network Hammerstein models have simple architectures, commonly with a few tens of processing nodes and adjustable weights, and the trained models are easy to be applied in practice. Due to the simple architecture of neural network Hammerstein models, their training algorithms have low computational complexity. More details on the neural network approach to the identification of Hammerstein system can be found in Chapter 3.

## 1.5 Summary

The material presented in this chapter starts with a concise introductory note explaining an underlaying incentive to write the book. Next, Section 1.1 contains a review of discrete time models of dynamic systems. It starts with the well-known linear model structures and shows nonlinear models as generalizations of linear ones. Section 1.2 introduces the multilayer perceptron, a neural network architecture that is used in both neural network Wiener and Hammerstein models. Next, in Sections 1.3 and 1.4, various available identification methods for Wiener and Hammerstein systems are briefly reviewed and classified into the following four groups: correlation methods, linear optimization methods, nonparametric regression methods, and nonlinear optimization methods.