Chapter 16
Outlook

The goal of this chapter is to illustrate some further subjects in control engineering. In the previous sections single variable (*SISO*), linear systems with constant parameters (*LTI*) were considered. The systems in practice, however, are usually nonlinear, multivariable and have varying parameters. It is not surprising, that the solution of these kinds of problems needs higher level control engineering theory. Neither does this chapter deal with all these subjects, instead it gives a short summary of four areas, which belong to the modern theory of *SISO* systems. These are:

- Norms of control engineering signals and systems
- Methods of numerical optimization
- Introduction to system identification
- Iterative and adaptive control schemes.

16.1 Norms of Control Engineering Signals and Operators

A norm in a complex linear space is interpreted as a real number, called the norm of $\boldsymbol{x}$ and denoted by $\|\boldsymbol{x}\|$, which can be applied to any vector $\boldsymbol{x}$ of the space, and which satisfies the relationships below

$\|\boldsymbol{x}\| > 0$ if $\boldsymbol{x} \neq 0$, and $\|0\| = 0$.
$\|a\boldsymbol{x}\| = |a|\|\boldsymbol{x}\|$ for an arbitrary complex number a
$\|\boldsymbol{x}+\boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$, which is the so-called triangle inequality.

The same concept exists regarding the linear vector-spaces of dimension n, and formally the same is valid for functions, too.

The quality of the control—as was seen in the previous sections—is connected with the error signal, or to the sensitivity function. The error signal is a function of time, but the sensitivity function is a complex frequency function, thus they are all functions. Their magnitude somehow has to be defined, because their value at a given frequency does not characterize the whole function, not speaking about their

L. Keviczky et al., *Control Engineering*, Advanced Textbooks in Control and Signal Processing, https://doi.org/10.1007/978-981-10-8297-9_16

magnitude. A mathematical notion, the above norm, is used for characterizing the magnitude of a function. Next, some basic norms will be presented, whose definitions will explain their meaning.

16.1.1 Norms of Signals

$$\mathcal{L}_1 \text{ norm:} \quad \|u(t)\|_1 = \int_{-\infty}^{\infty} |u(t)|\mathrm{d}t \tag{16.1}$$

$$\mathcal{L}_2 \text{ norm:} \quad \|u(t)\|_2 = \sqrt{\int_{-\infty}^{\infty} |u(t)|^2\mathrm{d}t} \tag{16.2}$$

$$\mathcal{L}_\infty \text{ norm:} \quad \|u(t)\|_\infty = \max_t |u(t)| \tag{16.3}$$

In the practice usually input functions ($u(t) \equiv 0$, if $t<0$) are investigated, where the lower limit of their integral is zero.

From the integrals of errors (integral criteria) discussed in Chap. 4, $I_3 = IAE = \|e(t)\|_1$ is the $\mathcal{L}_1$ norm of $e(t)$, $I_2 = \|e(t)\|_2^2$ is the square of the $\mathcal{L}_2$ norm. The relationships are quite obvious, nevertheless the integral criteria are considered rather engineering quality measures, but the norms are strict mathematical definitions.

For non-final-time signals there is the definition of power as

$$\mathrm{pow}[u(t)] = \sqrt{\lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} |u(t)|^2\mathrm{d}t} \tag{16.4}$$

Note that final-time, constrained signals have only energy, their power is zero. Thus if $\|u(t)\|_2 < \infty$ then $\mathrm{pow}[u(t)] = 0$.

The simplest inequalities regarding these norms are

$$\mathrm{pow}[u(t)] \le \|u(t)\|_\infty; \quad \text{if} \quad \|u(t)\|_\infty < \infty \tag{16.5}$$

$$\|u(t)\|_2 \le \sqrt{\|u(t)\|_\infty \|u(t)\|_1}; \quad \text{if} \quad \|u(t)\|_\infty < \infty \text{ and } \|u(t)\|_1 < \infty \tag{16.6}$$

16.1.2 Operator Norms

Using the frequency function $H(j\omega)$ of an *LTI* system having a stable transfer function $H(s)$ the following norms can be defined.

$$\mathcal{H}_2 \text{ norm:} \quad \|H(j\omega)\|_2 = \sqrt{\frac{1}{2\pi}\int_{-\infty}^{\infty} |H(j\omega)|^2 \mathrm{d}\omega} \tag{16.7}$$

$$\mathcal{H}_\infty \text{ norm:} \quad \|H(j\omega)\|_\infty = \max_{\omega} |H(j\omega)| \tag{16.8}$$

These operator norms are usually called system norms.

The computation of the $\mathcal{H}_2$ norm can be performed on the basis of the PARSEVAL-theorem.

$$\|H(j\omega)\|_2^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty} |H(j\omega)|^2 \mathrm{d}\omega = \frac{1}{2\pi j}\oint H(-s)H(s)\mathrm{d}s = \sum \mathrm{Res}[H(-s)H(s)], \tag{16.9}$$

where the residues of $[H(-s)H(s)]$ have to be taken into consideration on the left half plane. The expression (16.9) can be used only (i.e., $\mathcal{H}_2$ is finite), if $H(s)$ is strictly proper and has no pole on the imaginary axis. It is worth noting that

$$\|H(j\omega)\|_2 = \sqrt{\int_{-\infty}^{\infty} |w(t)|^2 \mathrm{d}t} = \sqrt{\int_{0}^{\infty} |w(t)|^2 \mathrm{d}t}, \tag{16.10}$$

where $w(t)$ is the weighting function of the system having transfer function $H(s)$.

If the system $H(s)$ is given in state-space form $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}^{\mathrm{T}})$, then the $\mathcal{H}_2$ norm can be computed by the following expression

$$\|H(j\omega)\|_2 = \sqrt{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{c}}, \tag{16.11}$$

where

$$\boldsymbol{L} = \int_{0}^{\infty} e^{\boldsymbol{A}t}\boldsymbol{b}\boldsymbol{b}^{\mathrm{T}}e^{\boldsymbol{A}^{\mathrm{T}}t}\mathrm{d}t. \tag{16.12}$$

Instead of the computation of the integral (16.12), $\boldsymbol{L}$ can be simply determined by solving the system of linear equations for $\boldsymbol{L}$:

$$\boldsymbol{A}\boldsymbol{L} + \boldsymbol{L}\boldsymbol{A}^{\mathrm{T}} = -\boldsymbol{b}\boldsymbol{b}^{\mathrm{T}} \tag{16.13}$$

(see A.16.1 in Appendix A.5). Equation (16.13) can also be solved by the conventional solution technique for systems of linear equations if the unknown column vectors of $\boldsymbol{L}$ are collected into one column vector.

The computation of the $\mathcal{H}_\infty$ norm is not easy, though its geometrical interpretation is very simple: it is the farthest distance of the NYQUIST diagram of $H(j\omega)$ from the origin. Since $H(s)$ and $H(j\omega)$ are usually rational functions, the possible places of the extrema of the absolute value (the necessary condition) are derived from the zeros of the first order derivative. This equation, however, yields a high order system of polynomial equations even for a low order process, whose solution requires numerical techniques. That is why, instead of an analytical solution, numerical methods are used directly to determine the maximum of $|H(j\omega)|$. The $\mathcal{H}_\infty$ norm is finite if $H(s)$ is proper and has no pole on the imaginary axis or in the right half plane.

(The computation of the $\mathcal{H}_\infty$ norm for error-function operators can be performed by the NEVANLINNA-PICK approximation procedure, but its discussion goes beyond the content of this textbook.)

The most important inequality regarding the $\mathcal{H}_\infty$ norm is

$$\|H_1(j\omega)H_2(j\omega)\|_\infty \leq \|H_1(j\omega)\|_\infty \|H_2(j\omega)\|_\infty. \tag{16.14}$$

Keeping the former notations, let $u(t)$ be the input and $y(t)$ the output of the system with transfer function $H(s)$. The most important relationships of the signals and norms of the system are for stable processes:

$$\|y(t)\|_2 \leq \|H(j\omega)\|_\infty \|u(t)\|_2, \tag{16.15}$$

therefore it can be stated that the $\mathcal{H}_\infty$ norm is the upper limit of the gain of the $\mathcal{L}_2$ norm. Based on the inequality

$$\|y(t)\|_\infty \leq \|w(t)\|_1 \|u(t)\|_\infty \tag{16.16}$$

it can be simply seen that the $\mathcal{L}_1$ norm of the weighting function is the upper limit of the gain of the $\mathcal{L}_\infty$ norm. Thus the upper limit of the maximum of the unit step response $y(t) = v(t)$ (if $u(t) = 1(t)$) is equal to the integral of the absolute value of the weighting function. Similar relations are valid for the following inequality

$$\|y(t)\|_\infty \leq \|H(j\omega)\|_2 \|u(t)\|_2. \tag{16.17}$$

It comes from the comparison of (16.16) and (16.17) that

$$\|y(t)\|_\infty \leq \min\left\{\|w(t)\|_1 \|u(t)\|_\infty; \|H(j\omega)\|_2 \|u(t)\|_2\right\}, \tag{16.18}$$

where a more strict condition is applied. Thus the $\mathcal{H}_\infty$, $\mathcal{H}_2$ and $\mathcal{L}_1$ norms, for certain signals, can correspond to the upper limit of the gain.

Similar relationships can be formulated for the power of the input and output signals:

$$\text{pow}[y(t)] \leq \|H(j\omega)\|_{\infty} \text{pow}[u(t)], \tag{16.19}$$

by means of which we obtain

$$\text{pow}[y(t)] \leq \|H(j\omega)\|_{\infty} \|u(t)\|_{\infty}. \tag{16.20}$$

From the comparison of these two latter inequalities, it follows that

$$\begin{aligned} \text{pow}[y(t)] &\leq \min\left\{ \|H(j\omega)\|_{\infty} \text{pow}[u(t)]; \|H(j\omega)\|_{\infty} \|u(t)\|_{\infty} \right\} \\ &= \|H(j\omega)\|_{\infty} \min\left\{ \text{pow}[u(t)]; \|u(t)\|_{\infty} \right\} \end{aligned} \tag{16.21}$$

where the more strict condition is applied.

It was shown in Chap. 7 that the optimality of *YP* controllers applied for stable processes can be reached via the optimal choice of the embedded filters $G_{\mathbf{x}}|_{\mathbf{x}=\mathrm{r,n}}$ (transfer functions). Their optimality for the error transfer functions $R_{\mathbf{x}}(1 - G_{\mathbf{x}} P_{-} e^{-sT_{\mathrm{d}}})|_{\mathbf{x}=\mathrm{r,n}}$ can be ensured by the minimization of the operator norms $\mathcal{H}_2$ and $\mathcal{H}_{\infty}$.

16.2 Basic Methods of the Numerical Optimization

The optimization problems of control engineering can usually be formulated by seeking the minimum of a scalar-vector function $f(\boldsymbol{x})$. The function to be minimized may be e.g., an integral criterion, a signal or operator norm, the vector components of the vector space of the searching are the parameters of the controller. Basically two main groups of extremum seeking methods can be distinguished depending on whether only the value of the function can be computed, or also its first and second order derivatives can be determined at a point $\boldsymbol{x}$.

16.2.1 Direct Seeking Methods

In the case of direct seeking (*DS*) methods only the value of the function $f(\boldsymbol{x})$ can be computed at a given point of seeking the minimum. The most effective *DS* method is the so-called adaptive simplex method of Nelder and Mead. In an n-dimensional space, a simplex is a shape given by $(n+1)$ points. Thus in two-dimensional space it is a triangle, in the three-dimensional case it is a tetrahedron. Find the minimum of $f(\boldsymbol{x})$ in a two dimensional space. First consider the simplex *ABC* shown in Fig. 16.1. Compute the values of $f(\boldsymbol{x})$ at the three points of the simplex. Based on the values $f(\boldsymbol{x}_A), f(\boldsymbol{x}_B)$ and $f(\boldsymbol{x}_C)$, let us arrange in order of their magnitudes the corresponding coordinate vectors of the three points. Assume that the biggest value is obtained at the point $f(\boldsymbol{x}_A = \boldsymbol{x}_{\mathrm{o}})$. Mirror this point $\boldsymbol{x}_{\mathrm{o}}$ to the center point of the opposite—less by one order—shape, i.e., now via the middle point $\boldsymbol{x}_1$ of the line *BC* to the point $\boldsymbol{x}_2$. Then continue this procedure (stepping) in the obtained direction until the values of $f(\boldsymbol{x}_i)$ increase. Assume that at point $\boldsymbol{x}_4$ the value is $f(\boldsymbol{x}_4) > f(\boldsymbol{x}_3)$, i.e., the minimum seeking algorithm does not give a better

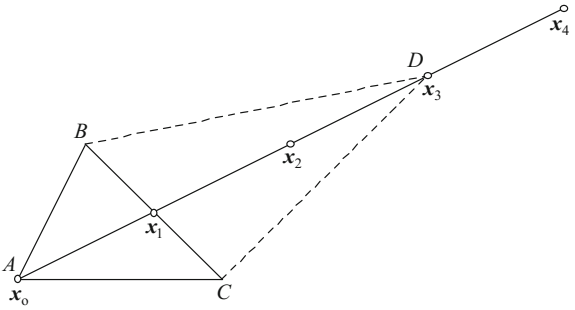

Fig. 16.1 Scheme of the adaptive simplex method

point. This means that the first point of the new simplex will be $\boldsymbol{x}_3$ and the simplex will be given by the triangle *BCD*. Then the point $\boldsymbol{x}_B$ belonging to the second biggest function value $f(\boldsymbol{x}_B)$ has to be mirrored on the middle point of line *CD*, then the seeking steps have to be continued in this direction. If all the points of the original simplex have already been mirrored, then we get into a completely new simplex whose form follows the form of the function $f(\boldsymbol{x})$ to a certain extent. The procedure is continued until mirroring all points of the simplex only a worse point is found, i.e., a bigger value of $f(\boldsymbol{x})$ is obtained. This case is called the limiting (or boundary) simplex. Then the sought minimum is inside this simplex.

The method is continued by formulating a new simplex with half size edges based on the worst point, i.e., by shrinking the simplex. The algorithm is started again from this shrunk simplex. The search method is stopped when the size of the limit simplex in each coordinate direction is within a certain accuracy threshold (the convergence limit).

The advantage of the simplex method is that it can easily handle both explicit constraints

$$\boldsymbol{x}_{\min} \leq \boldsymbol{x} \leq \boldsymbol{x}_{\max} \tag{16.22}$$

and so-called k implicit constraints, such as

$$g_j(\boldsymbol{x}) \leq 0 \quad j = 1, \ldots, k. \tag{16.23}$$

To achieve this, the starting point $\boldsymbol{x}_o$ has to fulfill the above conditions, then during the stepping the above restrictions are handled as if a bigger $f(\boldsymbol{x})$ had been obtained, thus the seeking in that direction has to be stopped.

The adaptive simplex method is able to find the minimum of a function of even a very special form with acceptable efficiency. Of course, it can determine only the minimum of a unimodal function, i.e., when $f(\boldsymbol{x})$ has only one extremum, or it can seek for a local minimum in a given region.

If the task is such that several extrema can be expected, i.e., $f(\boldsymbol{x})$ is multimodal, then the adaptive complex method can be applied. The "complex" is defined by a shape (set) given by $N > n + 1$ points in an n-dimensional space. Usually N is much

bigger than n, and the algorithm has to be started with an equally distributed point set in the search space. The algorithm operates similarly to the adaptive simplex method, but now the given point has to be mirrored via the geometrical center of all the other $N-1$ points, then the stepping has to be continued in this direction.

16.2.2 Gradient Based Methods

In the cases when the first and second derivates of the function $f(\boldsymbol{x})$ can be computed, then algorithms faster than the *DS* methods can also be constructed. The general canonical form of the methods using the gradient is the following iterative algorithm:

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \boldsymbol{G}(\boldsymbol{x}_i)\frac{\mathrm{d}f(\boldsymbol{x}_i)}{\mathrm{d}\boldsymbol{x}}. \tag{16.24}$$

The gradient methods can be basically distinguished by how to choose the weighting matrix $\boldsymbol{G}(\boldsymbol{x}_i)$ (Note that each version of the algorithms approximates the gradient $\mathrm{d}f(\boldsymbol{x}_i)/\mathrm{d}\boldsymbol{x}$ in a different way).

From the different ways of choosing the weighting matrix $\boldsymbol{G}(\boldsymbol{x}_i)$, the first is the so-called gradient method:

$$\boldsymbol{G}(\boldsymbol{x}_i) = G(\boldsymbol{x}_i) = \left[\frac{\mathrm{d}f^{\mathrm{T}}(\boldsymbol{x}_i)}{\mathrm{d}\boldsymbol{x}}\boldsymbol{H}(\boldsymbol{x}_i)\frac{\mathrm{d}f(\boldsymbol{x}_i)}{\mathrm{d}\boldsymbol{x}}\right]^{-1}, \tag{16.25}$$

where $\boldsymbol{H}(\boldsymbol{x}_i)$ is the HESSIAN matrix (see (A.1.31)) of the function $f(\boldsymbol{x})$ at the point $\boldsymbol{x}_i$. It is interesting that now $G(\boldsymbol{x}_i)$ is a scalar. This method uses a second order approximation in the direction opposite to the gradient, and puts the next iteration point at the minimum of the parabola taken in this direction. The significant disadvantage of this method is that in the case of "curving" valleys, it slows down because it cannot follow precisely the deepness shape of the valley.

The next method is the NEWTON-RAPHSON method (sometimes it is also called the GAUSS-NEWTON method), where

$$\boldsymbol{G}(\boldsymbol{x}_i) = [\boldsymbol{H}(\boldsymbol{x}_i)]^{-1}. \tag{16.26}$$

This method fits a general quadratic surface (multidimensional ellipsoid) at an iteration point and puts the next iteration point at the calculable extremum of this shape.

The above two methods using gradients have also very clear geometrical interpretations, the other methods can be considered as different combinations of these.

The gradient methods are much more effective than the *DS* ones for so-called "well behaved" functions, but for exceptional functions, e.g., having the form of a banana, they slow down. Their further disadvantage is that they are not very effective in the case of constraints, because they usually shrink to the trajectory

point crossing the constraining surface. In this case, certain techniques use the solution to push the iteration off this surface and the search starts again.

There are several procedures and software programs available for all the above methods in the different program packages and in an object oriented *CAD* environment.

It has been noted in Sect. 4.8 in connection with the square error area that its minimization generally provides an optimal step response function having a relatively high overshoot. Therefore it seems reasonable to construct the optimization task which performs this minimization of the integral criterion $I_2 = f(\boldsymbol{x})$ under the restriction for the overshoot $\sigma = g(\boldsymbol{x}) \leq 1.05$. This task guarantees a "nice" step response function with a small overshoot.

Example 16.1 The expression for the so-called "function of banana" frequently used in optimization tasks is

$$f(\boldsymbol{x}) = 100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2. \tag{16.27}$$

The function in 3D is shown in Fig. 16.2, whose minimum is at the point $\boldsymbol{x} = [1, 1]$.

The operation of the adaptive simplex method is illustrated in Fig. 16.3. The procedure starts from the point $\boldsymbol{x} = [-1.9, 2]$ and after 210 iterations it finds the minimum (i.e., it computes the function's value at 210 points).

Figure 16.4 shows the operation of the gradient method, more exactly its inability to find the minimum after computing the function's value at 210 points, and the gradients at 200 points, but it stopped at the beginning of the valley.

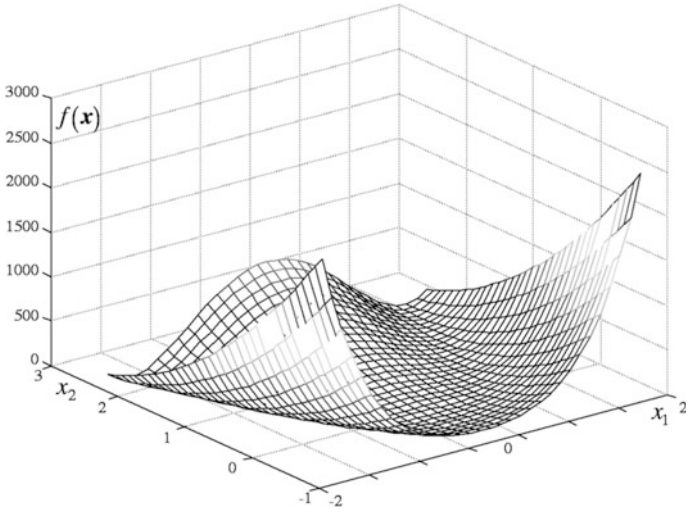

Fig. 16.2 The so-called "function of banana" applied quite often in optimization tasks

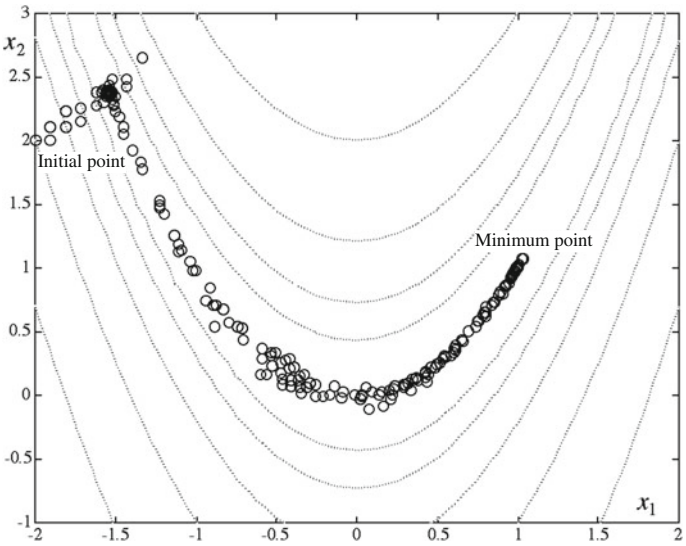

Fig. 16.3 Optimum seeking by simplex method

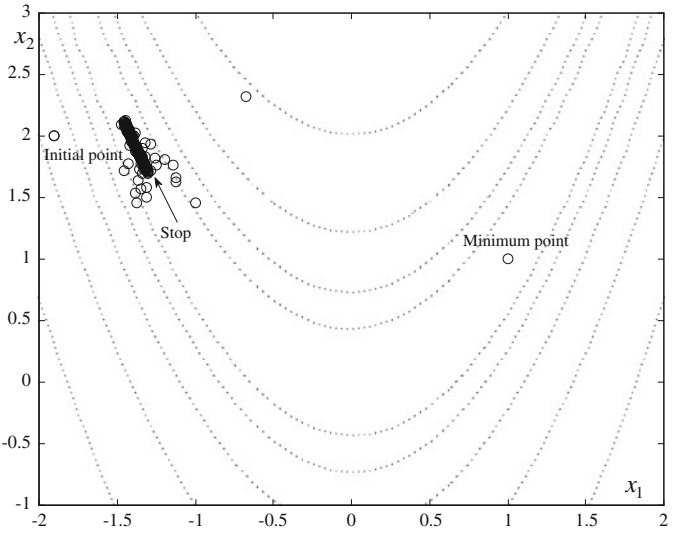

Fig. 16.4 The inability of the gradient method to find the minimum of (16.27)

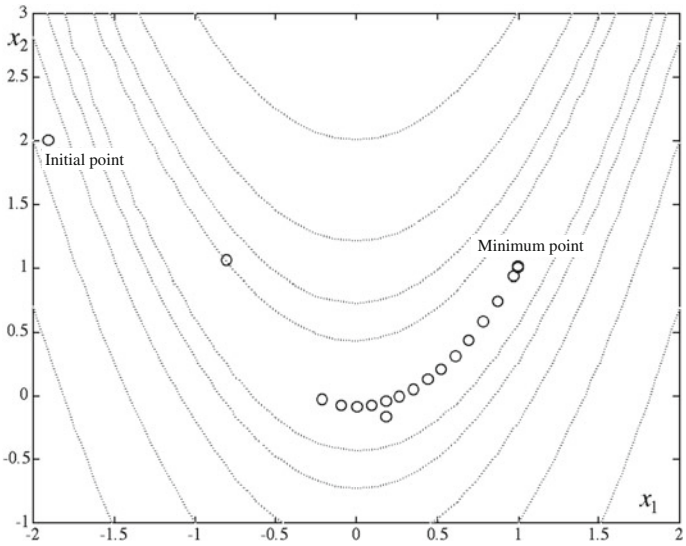

Fig. 16.5 Illustration of the effectiveness of the NEWTON-RAPHSON method

The effectiveness of the NEWTON-RAPHSON method is demonstrated in Fig. 16.5: the method found the minimum after 21 iterations. ■

16.3 Introduction to Process Identification

It has been seen in Sect. 2.4 that one of the basic tasks in control engineering is process identification, when the model $\hat{P}$ of the process P to be controlled is determined from the measurements of the input and output signals. Process identification, starting with simple grapho-analytical methods, has today become an independent (autonomous) discipline; its methods and results can be found in several books. With the spread of modern computational techniques, almost standard tools are available to solve the most important tasks. Here only some of the topmost methods are discussed, just to illustrate the applied algorithms and techniques.

Process identification methods substantially differ from each other, depending on the task to be solved, i.e., whether the static characteristics or the dynamic model of the process has to be determined.

16.3.1 Identification of Static Processes

Assume that the static characteristic of the process is a line $p_o + p_1 u$, which can be measured with measurement error e

$$y = p_o + p_1 u + e. \tag{16.28}$$

The input signal u is measured without error, or it is a known signal put into the system (active experiment). The input and output signals are measured jointly at N points. These values are approximated by the linear model

$$\hat{y} = \hat{p}_o + \hat{p}_1 u = \boldsymbol{f}^T(u)\hat{\boldsymbol{p}}; \quad \boldsymbol{f}^T(u) = [\,1 \quad u\,]; \quad \hat{\boldsymbol{p}} = [\,\hat{p}_o \quad \hat{p}_1\,]^T \tag{16.29}$$

as seen in Fig. 16.6. Here $\boldsymbol{f}(u)$ is called the vector of function components.

If the additive measurement error has zero average, then the so-called *Least Squares (LS)* method provides the unbiased estimation of the process parameters. The *LS* method takes the sum of the squares of the differences between the measured value and the model output at each point and optimizes it according to the criterion

$$V(\hat{\boldsymbol{p}}, N) = \frac{1}{2}\sum_{j=1}^{N}\left[y_j - \boldsymbol{f}^T(u_j)\hat{\boldsymbol{p}}\right]^2 = \frac{1}{2}[\boldsymbol{y} - \boldsymbol{F}_u\hat{\boldsymbol{p}}]^T[\boldsymbol{y} - \boldsymbol{F}_u\hat{\boldsymbol{p}}], \tag{16.30}$$

where

$$\boldsymbol{F}_u = \begin{bmatrix} 1 & u_1 \\ 1 & u_2 \\ \vdots & \vdots \\ 1 & u_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}^T(u_1) \\ \boldsymbol{f}^T(u_2) \\ \vdots \\ \boldsymbol{f}^T(u_N) \end{bmatrix} \quad \text{and} \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}. \tag{16.31}$$

The system of vector equations for the N samples is

$$\boldsymbol{y} = \boldsymbol{F}_u\boldsymbol{p} + \boldsymbol{e}, \tag{16.32}$$

where $\boldsymbol{e} = [\,e_1 e_1 \ldots e_N\,]^T$ is the vector of the measurement errors. The parameter estimation minimizing the sum of squares according to A.16.2 in Appendix A.5 is

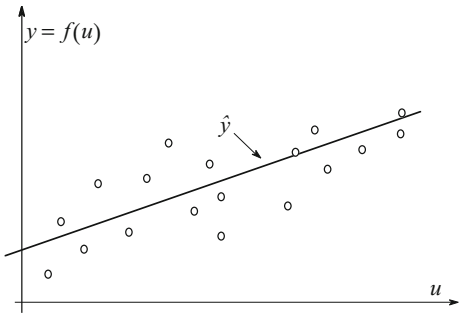

Fig. 16.6 Identification of linear static model

$$\hat{\boldsymbol{p}} = \left[\boldsymbol{F}_{\mathrm{u}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{u}}\right]^{-1}\boldsymbol{F}_{\mathrm{u}}^{\mathrm{T}}\boldsymbol{y}. \tag{16.33}$$

This estimation is unbiased, i.e., $E\{\hat{\boldsymbol{p}}\} = \boldsymbol{p}$, thus the expected value of $\hat{\boldsymbol{p}}$ is the unknown original parameter vector $\boldsymbol{p}$. If e has a normal distribution, then the $\hat{\boldsymbol{p}}$ obtained by the *LS* estimation has minimum variance and is the *best* estimator of $\boldsymbol{p}$.

In many cases the input signal u is not known in advance, just measured (passive experiment). If u is a random signal, then in order to get an unbiased estimation by the *LS* method, the independence of the signals e and u has to be assumed.

The computation of the solution (16.33) can be made easier by taking the following relationships into account

$$\boldsymbol{F}_{\mathrm{u}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{u}} = \sum_{j=1}^{N}\boldsymbol{f}(u_j)\boldsymbol{f}^{\mathrm{T}}(u_j) \quad \text{and} \quad \boldsymbol{F}_{\mathrm{u}}^{\mathrm{T}}\boldsymbol{y} = \sum_{j=1}^{N}\boldsymbol{f}(u_j)y_j. \tag{16.34}$$

Assume that the static characteristic of the process is a parabola $p_{\mathrm{o}} + p_1 u + p_2 u^2$, and the additive measurement error is e.

$$y = p_{\mathrm{o}} + p_1 u + p_2 u^2 + e. \tag{16.35}$$

The input signal u is assumed to be measured without error. The input and output signals are measured jointly at N points, and the following nonlinear (quadratic) model is fitted to the measured values, as seen in Fig. 16.7. Now introduce

$$\hat{y} = \hat{p}_{\mathrm{o}} + \hat{p}_1 u + \hat{p}_2 u^2 = \boldsymbol{f}^{\mathrm{T}}(u)\hat{\boldsymbol{p}}; \quad \boldsymbol{f}^{\mathrm{T}}(u) = \begin{bmatrix} 1 & u & u^2 \end{bmatrix}; \quad \hat{\boldsymbol{p}} = \begin{bmatrix} \hat{p}_{\mathrm{o}} & \hat{p}_1 & \hat{p}_2 \end{bmatrix}^{\mathrm{T}}. \tag{16.36}$$

Observe that the model $\hat{y} = \boldsymbol{f}^{\mathrm{T}}(u)\hat{\boldsymbol{p}}$ is still linear in the parameters. Thus the *LS* method can be applied unchanged if the matrix $\boldsymbol{F}_{\mathrm{u}}$ is formulated from the function component vector $\boldsymbol{f}(u)$ according to the quadratic model (16.36):

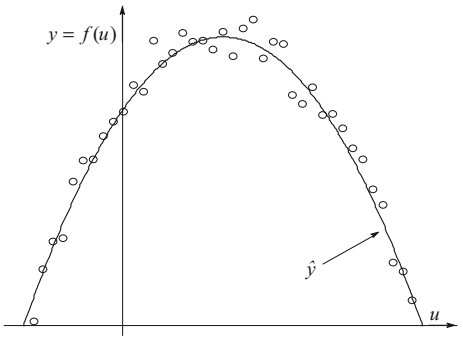

Fig. 16.7 Identification of a nonlinear (quadratic) static model

$$\boldsymbol{F}_{\mathrm{u}} = \begin{bmatrix} 1 & u_1 & u_1^2 \\ 1 & u_2 & u_2^2 \\ \vdots & \vdots & \vdots \\ 1 & u_N & u_N^2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}^{\mathrm{T}}(u_1) \\ \boldsymbol{f}^{\mathrm{T}}(u_2) \\ \vdots \\ \boldsymbol{f}^{\mathrm{T}}(u_N) \end{bmatrix} \tag{16.37}$$

and $\hat{\boldsymbol{p}}$ is computed again by Eq. (16.33).

Observe that a relatively wide class of functions can be written in a form that is linear in its parameters.

If the static characteristic is nonlinear, then an extremum seeking method is used to minimize $V(\hat{\boldsymbol{p}}, N)$, which, e.g., can be chosen from those discussed in Sect. 16.2.

16.3.2 Identification of Dynamic Processes

Nowadays the identification of dynamic processes exclusively means the determination of a discrete time (*DT*) model. It has been shown in Sect. 11.4 that a *DT* system given by the so-called filter form

$$y[k] = G\big(z^{-1}\big)u[k] = \frac{\mathcal{B}(z^{-1})z^{-d}}{\mathcal{A}(z^{-1})}u[k] = \frac{\mathcal{B}(z^{-1})z^{-d}}{1+\tilde{\mathcal{A}}(z^{-1})}u[k] \tag{16.38}$$

can be written in a form linear in parameters as

$$\begin{aligned} y[k] &= \mathcal{B}\big(z^{-1}\big)z^{-d}u[k] - \tilde{\mathcal{A}}\big(z^{-1}\big)y[k] = \boldsymbol{f}^{\mathrm{T}}(u,y,k)\boldsymbol{p}_{\mathrm{ba}} \\ &= b_1u[k{-}d{-}1] + b_1u[k{-}d{-}2] + \cdots + b_nu[k{-}d{-}n] - a_1y[k{-}1] - \cdots - a_ny[k{-}n] \end{aligned} \tag{16.39}$$

where

$$\begin{aligned} \boldsymbol{f}^T(u,y,k) &= [u[k-d-1]\, u[k-d-2] \ldots u[k-d-n] \; -y[k-1] \ldots \; -y[k-n]] \\ \boldsymbol{p}_{\mathrm{ba}} &= [b_1\; b_1 \ldots b_n\, a_1 \ldots a_n] \end{aligned} \tag{16.40}$$

This technique, by means of which the difference equation of the dynamic *DT* systems is made "quasi-linear", opens the possibility of formulating further process identification algorithms similar to the *LS* method. The measurement noise problems of *DT* systems, however, should be discussed in a basically different way than for static characteristics. The measuring situation is illustrated in Fig. 16.8.

Here the measurement of $u[k]$ is assumed to be without error, but the noiseless output signal $v[k]$ of the system is assumed to be measured with an additive measurement error $y_{\mathrm{n}}[k]$. This output noise $y_{\mathrm{n}}[k]$ derives from an independent, zero mean, so-called white noise via the noise model $\mathcal{C}(z^{-1})/\mathcal{D}(z^{-1})$.

This task essentially requires the identification of two models: the process model and the noise model. The task can be drastically simplified by certain assumptions made regarding the noise model. If the noise model has the form $1/\mathcal{A}(z^{-1})$, i.e.,

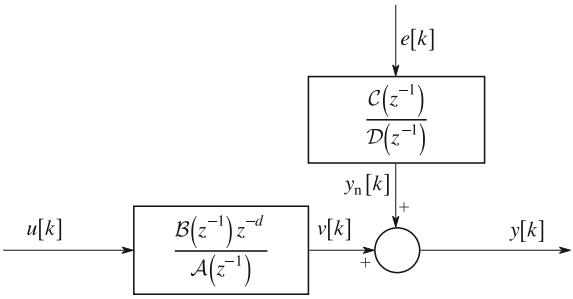

Fig. 16.8 Measured signals of a linear dynamic discrete-time system

$$y[k] = \frac{\mathcal{B}(z^{-1})z^{-d}}{\mathcal{A}(z^{-1})}u[k] + \frac{1}{\mathcal{A}(z^{-1})}e[k] \tag{16.41}$$

then the original process can be rewritten as

$$y[k] = \boldsymbol{f}^{\mathrm{T}}(u, y, k)\boldsymbol{p}_{\mathrm{ba}} + e[k]. \tag{16.42}$$

Observe that this form essentially corresponds to Eqs. (16.28) and (16.35) seen in the identification of the static characteristics, thus the *LS* method can be directly applied if the matrix $\boldsymbol{F}(u)$ is constructed from $\boldsymbol{f}^{\mathrm{T}}(u, y, k)$ instead of $\boldsymbol{f}(u)$, and $\hat{\boldsymbol{p}}_{\mathrm{ba}}$ is the parameter vector. Let us create first the vector

$$\boldsymbol{y} = [\,y[1] \quad y[2] \quad \ldots \quad y[N]\,]^{\mathrm{T}} \tag{16.43}$$

and the matrix

$$\boldsymbol{F}_{\mathrm{uy}} = \begin{bmatrix} \boldsymbol{f}^{\mathrm{T}}(u, y, 1) \\ \boldsymbol{f}^{\mathrm{T}}(u, y, 2) \\ \vdots \\ \boldsymbol{f}^{\mathrm{T}}(u, y, N) \end{bmatrix} \tag{16.44}$$

The parameter estimation by the *LS* method has also the form of (16.33)

$$\hat{\boldsymbol{p}}_{\mathrm{ba}} = \left[\boldsymbol{F}_{\mathrm{uy}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{uy}}\right]^{-1}\boldsymbol{F}_{\mathrm{uy}}^{\mathrm{T}}\boldsymbol{y}. \tag{16.45}$$

This estimation is asymptotically unbiased, i.e., $\mathrm{plim}_{N\to\infty}\{\hat{\boldsymbol{p}}\} = \boldsymbol{p}$, thus the probabilistic limit value of $\hat{\boldsymbol{p}}$ is the unknown original parameter vector $\boldsymbol{p}$. The independence of $e[k]$ has to be assumed, because $\boldsymbol{f}^{\mathrm{T}}(u, y, k)$ has measured values depending on $e[k]$. If $e[k]$ has a normal distribution, then the $\hat{\boldsymbol{p}}$ resulting from the *LS* estimation is the minimum variance (best) estimator of $\boldsymbol{p}$.

Unfortunately the form of the noise model $1/\mathcal{A}(z^{-1})$ is very special, so it cannot be used generally. The noise model $\mathcal{C}(z^{-1})/\mathcal{A}(z^{-1})$ can be considered as a more general form, when

$$y[k] = \frac{\mathcal{B}(z^{-1})z^{-d}}{\mathcal{A}(z^{-1})}u[k] + \frac{\mathcal{C}(z^{-1})}{\mathcal{A}(z^{-1})}e[k] = \frac{\mathcal{B}(z^{-1})z^{-d}}{1+\tilde{\mathcal{A}}(z^{-1})}u[k] + \frac{1+\tilde{\mathcal{C}}(z^{-1})}{\mathcal{A}(z^{-1})}e[k]. \tag{16.46}$$

This form preserves the generality of the noise model $\mathcal{C}(z^{-1})/\mathcal{D}(z^{-1})$, but it contains a great number of redundant parameters because of bringing the fractions to a common denominator. The quasi-linearization by (16.36) can be easily performed here, too.

$$\begin{aligned} y[k] &= \mathcal{B}(z^{-1})z^{-d}u[k] - \tilde{\mathcal{A}}(z^{-1})y[k] + \tilde{\mathcal{C}}(z^{-1})e[k] + e[k] \\ &= b_1u[k-d-1] + b_2u[k-d-2] + \cdots + b_nu[k-d-n] - a_1y[k-1] - \cdots - a_ny[k-n] \\ &\quad + c_1e[k-1] + \cdots + c_ne[k-n] + e[k] = \boldsymbol{f}^{\mathrm{T}}(u,y,e,k)\boldsymbol{p}_{\mathrm{bac}} + e[k] \end{aligned} \tag{16.47}$$

The most important disadvantage of this form of the model is that the past values of $e[k]$ in the vector $\boldsymbol{f}(u,y,e,k)$ are not known. But if the estimation $\hat{\boldsymbol{p}}_{\mathrm{bac}}$ of the $\boldsymbol{p}_{\mathrm{bac}}$ is known then an estimation $\hat{e}[k]$ of the source noise can always be computed in the form

$$\hat{e}[k] = y[k] - \boldsymbol{f}^{\mathrm{T}}(u,y,\hat{e},k)\hat{\boldsymbol{p}}_{\mathrm{bac}}, \tag{16.48}$$

where now the computed (estimated) value $\hat{e}[k]$ is in $\boldsymbol{f}(u,y,\hat{e},k)$. Creating the matrix

$$\boldsymbol{F}_{\mathrm{uy\hat{e}}} = \begin{bmatrix} \boldsymbol{f}^{\mathrm{T}}(u,y,\hat{e},1) \\ \boldsymbol{f}^{\mathrm{T}}(u,y,\hat{e},2) \\ \vdots \\ \boldsymbol{f}^{\mathrm{T}}(u,y,\hat{e},N) \end{bmatrix} \tag{16.49}$$

formally again an *LS* estimation is obtained based on (16.35) and (16.43) in the form

$$\hat{\boldsymbol{p}}_{\mathrm{bac}} = \left[\boldsymbol{F}_{\mathrm{uy\hat{e}}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{uy\hat{e}}}\right]^{-1}\boldsymbol{F}_{\mathrm{uy\hat{e}}}^{\mathrm{T}}\boldsymbol{y}. \tag{16.50}$$

Since the series $\hat{e}[k]$, $(k = 1,\ldots,N)$, is always computed for a given $\hat{\boldsymbol{p}}_{\mathrm{bac}}$, here only an iteration method can be realized, i.e., the series $\hat{e}[k]$ has to be computed after each estimation step. The iteration is continued until the difference between the consecutively estimated parameter vectors becomes less than a given error. (This iteration is called a relaxation-type one.) The solution (16.50) belonging to Eq. (16.47) is called the *Extended Least Squares* (*ELS*) method. Several other versions of this method are known, using different noise models, which has resulted in a huge number of methods in the literature.

Theoretically the most accurate method can be obtained by minimizing the loss function

$$V(\hat{\boldsymbol{p}}_{\text{bac}}, N) = \frac{1}{2}\sum_{j=1}^{N}\left\{y[j] - \boldsymbol{f}^{\text{T}}(u, y, \hat{e}, j)\hat{\boldsymbol{p}}_{\text{bac}}\right\}^2 = \frac{1}{2}\left[\boldsymbol{y} - \boldsymbol{F}_{\text{uy}\hat{\text{e}}}\hat{\boldsymbol{p}}_{\text{bac}}\right]^{\text{T}}\left[\boldsymbol{y} - \boldsymbol{F}_{\text{uy}\hat{\text{e}}}\hat{\boldsymbol{p}}_{\text{bac}}\right] \tag{16.51}$$

in the space of the parameter vector $\hat{\boldsymbol{p}}_{\text{bac}}$, which means a general minimum seeking problem. Even for different noise models, the first and second order derivates of $V(\hat{\boldsymbol{p}}_{\text{bac}}, N)$ with respect to the parameters can be relatively easily computed, so effective minimum seeking algorithms can be constructed this way. The methods directly minimizing (16.50) are called *Maximum Likelihood* (*ML*) methods. This method requires zero average, normal, white noise for $e[k]$.

Those methods which use simultaneously available N data-pairs of the input and output signals are called "off-line" or "batch" methods. All the above methods belong to this category.

There are measurement situations when the model obtained formerly by an estimation method is modified (renewed) after getting a new measured data-pair. These methods are called "on-line" or "recursive" identification methods.

16.3.3 Discrete-Time to Continuous-Time Transformation

It has been seen during the discussion of the basic discrete-time process identification methods that these methods—deriving from their character—provide the operators of models $\hat{G}(z^{-1})$ or $\hat{G}(q^{-1})$ constructed by the estimated parameters $\hat{\boldsymbol{p}}_{\text{ba}}$ of the pulse transfer function $G(z^{-1})$ or pulse transfer operator $G(q^{-1})$ of the process. (From the process identification point of view there is no importance attached to these notations and meanings.) Here $\hat{G}(z^{-1})$ is used. In many cases, however, the model $\hat{P}(s)$ of the original CT system is required as a result of the identification. This conversion, i.e., the equivalence at the sampling points, can be solved only by assuming a holding term of a given type. In connection with Eqs. (11.30) and (11.31) it has been already shown that in the case of a zero order hold, thus applying an *SRE* transformation, the parameter matrices of the DT state equations are

$$\boldsymbol{F} = e^{\boldsymbol{A}T_{\text{s}}} \quad \text{and} \quad \boldsymbol{g} = \boldsymbol{A}^{-1}\left(e^{\boldsymbol{A}T_{\text{s}}} - \boldsymbol{I}\right)\boldsymbol{b}. \tag{16.52}$$

Formally, the parameter matrices of the *SRE* equivalent CT systems can be obtained by the reverse of the equations

$$\boldsymbol{A} = \frac{1}{T_{\text{s}}}\ln(\boldsymbol{F}) \quad \text{and} \quad \boldsymbol{b} = \frac{1}{T_{\text{s}}}\ln(\boldsymbol{F})(\boldsymbol{F} - \boldsymbol{I})^{-1}\boldsymbol{g}. \tag{16.53}$$

Here $\ln(\boldsymbol{F})$ is the logarithm of the matrix $\boldsymbol{F}$, which is defined and computed by the definitions valid for matrix functions (see $e^{\boldsymbol{A}}$ in Chap. 3). Based on the above the algorithm of the discrete-continuous transformation is:

1. Based on the estimated $\hat{\boldsymbol{p}}_{\mathrm{ba}}$, construct the state-space description of a DT model by the controllable canonical form $\hat{\boldsymbol{F}}_{\mathrm{c}}; \hat{\boldsymbol{g}}_{\mathrm{c}}; \hat{\boldsymbol{c}}_{\mathrm{c}}$ or an observable $\hat{\boldsymbol{F}}_{\mathrm{o}}; \hat{\boldsymbol{g}}_{\mathrm{o}}; \hat{\boldsymbol{c}}_{\mathrm{o}}$ canonical form.
2. Using the transformation Eq. (16.53), compute the state-space form $\hat{\boldsymbol{A}}; \hat{\boldsymbol{b}}; \hat{\boldsymbol{c}}$ of the CT model. This step results in parameter matrices of general form having $n^2 + 2n$ parameters.
3. Transform the CT state-space model $\hat{\boldsymbol{A}}; \hat{\boldsymbol{b}}; \hat{\boldsymbol{c}}$ to a controllable or observable canonical form by either the transformation matrix $\boldsymbol{T}_{\mathrm{c}} = \boldsymbol{M}_{\mathrm{c}}^{\mathrm{c}}(\boldsymbol{M}_{\mathrm{c}})^{-1}$ or $\boldsymbol{T}_{\mathrm{o}} = \left(\boldsymbol{M}_{\mathrm{o}}^{\mathrm{o}}\right)^{-1}\boldsymbol{M}_{\mathrm{o}}$ from which the parameters of the transfer function $\hat{P}(s)$ of the CT model sought can be easily determined from the non-redundant structure corresponding to the canonical form. Note that in the case of the canonical form, it is not necessary to compute the whole matrix $\hat{\boldsymbol{A}}_{\mathrm{c}}$ or $\hat{\boldsymbol{A}}_{\mathrm{o}}$, it is sufficient to compute the first row or column.

The above transformation techniques are the most compact ones, but of course, there are different ways to solve the problem. The same accurate result can be obtained by decomposing $\hat{G}(z^{-1})$ into partial fractions and then the discrete-continuous transformation can be made term-by-term.

16.3.4 Recursive Parameter Estimation

First consider the recursive version of the *LS* method. Assume that N data-pairs are processed and the *LS* estimate is available in the form

$$\hat{\boldsymbol{p}}[N] = \left\{\boldsymbol{F}^{\mathrm{T}}[N]\boldsymbol{F}[N]\right\}^{-1}\boldsymbol{F}^{\mathrm{T}}[N]\boldsymbol{y}_N. \tag{16.54}$$

If we want to modify our estimate obtained by (16.54) using the new data-pairs $u[N+1]$ and $y[N+1]$ measured in the $[N+1]$-th time instant, then it can be computed by the following recursive relationships

$$\hat{\boldsymbol{p}}[N+1] = \hat{\boldsymbol{p}}[N] + \boldsymbol{R}[N+1]\boldsymbol{f}(N+1)\left\{y[N+1] - \boldsymbol{f}^{\mathrm{T}}(N+1)\hat{\boldsymbol{p}}[N]\right\} \tag{16.55}$$

and

$$\boldsymbol{R}[N+1] = \boldsymbol{R}[N] - \frac{\boldsymbol{R}[N]\boldsymbol{f}(N+1)\boldsymbol{f}^{\mathrm{T}}(N+1)\boldsymbol{R}[N]}{1+\boldsymbol{f}^{\mathrm{T}}(N+1)\boldsymbol{R}[N]\boldsymbol{f}(N+1)} \tag{16.56}$$

(see A.16.3 in Appendix A.5). Here $\boldsymbol{f}(N+1)$ means a general function independently of whether the method is applied to a static or dynamic process model. The so-called convergence matrix $\boldsymbol{R}[N]$ is

$$\boldsymbol{R}[N] = \left\{ \sum_{j=1}^{N} \boldsymbol{f}(j)\boldsymbol{f}^{\mathrm{T}}(j) \right\}^{-1} = \left\{ \boldsymbol{F}^{\mathrm{T}}[N]\boldsymbol{F}[N] \right\}^{-1}. \tag{16.57}$$

The equation-pair (16.55) and (16.56) belong to the family of the so-called learning, adaptive estimation algorithms, which are included in the canonical equation of the general *Stochastic Approximation* (*SA*):

$$\hat{\boldsymbol{p}}[k+1] = \hat{\boldsymbol{p}}[k] + \boldsymbol{R}[k+1] \frac{\mathrm{d}V(\hat{\boldsymbol{p}}, k)}{\mathrm{d}\hat{\boldsymbol{p}}}. \tag{16.58}$$

These *SA* algorithms differ from each other in the choice of the convergence matrix $\boldsymbol{R}[k]$ and the way to compute the gradient. Here only the best-known method has been discussed.

If the parameters of the process are varying, then it might be necessary to forget in a certain sense the validity of the former model and take into account the new measurements with higher importance. To solve this so-called "forgetting" problem, assume that the past is forgotten by using the following matrix

$$\boldsymbol{F}[N+1] = \begin{bmatrix} \lambda \boldsymbol{F}[N] \\ \boldsymbol{f}^{\mathrm{T}}(N+1) \end{bmatrix} \text{instead of } \boldsymbol{F}[N+1] = \begin{bmatrix} \boldsymbol{F}[N] \\ \boldsymbol{f}^{\mathrm{T}}(N+1) \end{bmatrix}$$

where the forgetting factor is $0 \leq \lambda \leq 1$. If $\lambda =$ constant, then it is enough to use the following convergence matrix

$$\boldsymbol{R}[N+1] = \frac{1}{\lambda^2} \left\{ \boldsymbol{R}[N] - \frac{\boldsymbol{R}[N]\boldsymbol{f}(N+1)\boldsymbol{f}^{\mathrm{T}}(N+1)\boldsymbol{R}[N]}{\lambda^2 + \boldsymbol{f}^{\mathrm{T}}(N+1)\boldsymbol{R}[N]\boldsymbol{f}(N+1)} \right\} \tag{16.59}$$

instead of (16.56).

A constant forgetting factor, however, may cause problems, if the new measurements do not have significantly new information, since this algorithm forgets exponentially the old information, and so $\boldsymbol{R}[N]$ may become singular. Therefore the choice of the corresponding forgetting strategy is the most critical part of the adaptive estimation method.

Note that Eqs. (16.55), (16.56) and (16.60) are usually called naive programming formulas. By means of them the method can be simply presented but numerically they behave badly. They are mostly used for purposes of demonstration or, simulation. In practice the canonical, diagonal form of $\boldsymbol{R}[N]$ and its recursive forms are used: this solution works best from the numerical point of view. This method uses the so-called GIVENS transformation.

16.3.5 Model Validation

During process identification the determination of a model of acceptable correctness (accuracy) can be made only by an iterative process. Its main steps are presented in Fig. 16.9.

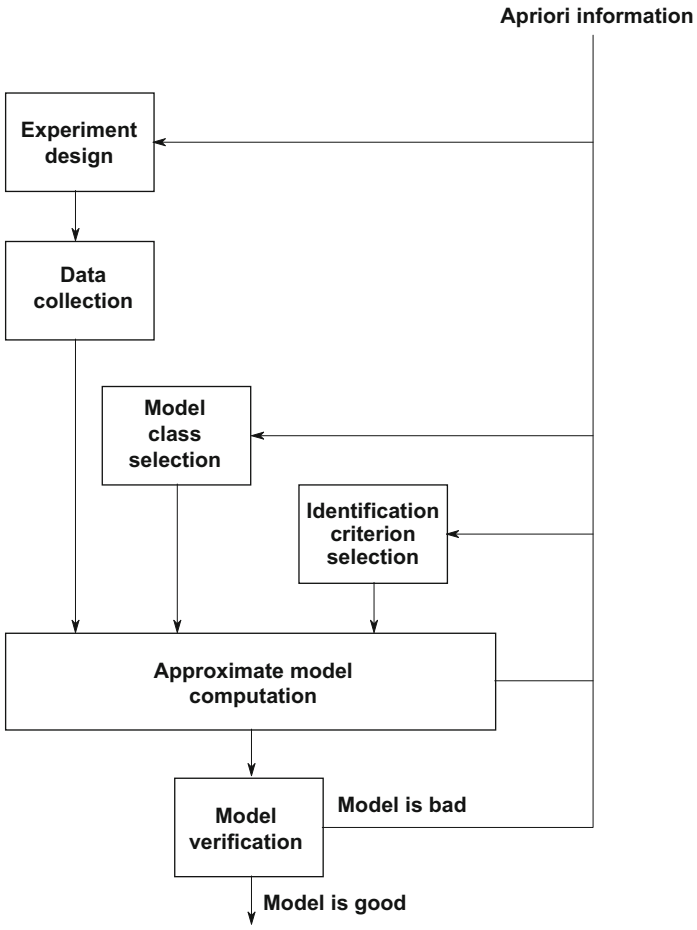

Fig. 16.9 The scheme of the whole process identification

The identification starts using certain preliminary information. First the so-called "design of experiments" is performed. In this step the optimal allocation of the measurement points is determined for static modeling. For dynamic modeling, however, one assumes that optimal input signals representing the significant frequency region are generated. This latter is called input design. The accuracy of the final model depends significantly on this step, therefore several theories deal with the optimal solution of this task [see, for example, Sect. 7.5].

The effect of the optimal measurement points or input signal for the process is realized by active experiment designs, and the data are collected during the experiments.

Based on the preliminary information, the class of the model and the identification criterion are chosen, then the determination of an approximate model is performed (parameter estimation).

Then the model output and the measured output of the process obtained for the same input are compared. From the deviations, qualitative, goodness of fit measures can be constructed for checking the acceptability of the model (model validation).

If the accuracy of the model is not satisfactory then the iteration is continued with a newer experiment design. The procedure is stopped when the accuracy of the model is acceptable.

16.4 Iterative and Adaptive Control Schemes

In the previous sections some off-line and on-line methods of process identification have been discussed. Not only can the process model be improved by repeated experiments, but also the controller, if a controller is designed and realized based on the model, applied in a closed-loop with the computed optimal parameters. This joint task is required, on the one hand, at the initial tuning of the regulator, and on the other hand, in the case of a process with slowly varying parameters, in the continuous adaptation of the regulator (adaptive control).

In the case of modern, microprocessor based compact controllers, nowadays there is embedded possibility for a certain kind of automatic tuning. The commercial controllers usually apply the ÅSTRÖM-type relay-tuning (see Sect. 8.3).

The more demanding optimal controller is based on the iterative strategy of a certain learning-adaptive version of joint identification-control (*simultaneous identification and control).* This strategy assumes that the identification is performed without opening the closed control loop, i.e., under normal operation conditions. The identification is usually off-line, i.e., it is based on the simultaneous processing of N data pairs. Based on the obtained model an optimal controller is determined and this controller is used in the next off-line experiment. By this technique the optimality of the controller can be gradually improved as the model becomes more and more accurate, while the normal operation of the process is hardly disturbed.

Certainly it is also possible to improve the parameter estimation of the process applying a recursive parameter estimating technique in every sampling instant, and the optimal controller output is applied to the process only delayed by the computation time of the optimal controller. In the case of today's fast operating computing equipment, this solution, to a very good approximation, can be considered as simultaneous processing in the case of significantly slower processes. This strategy is called adaptive control. The determination of a reliable controller is not an easy task. A recursive parameter estimation algorithm is required which does not forget the learned model if the new measurements do not have significant new information. If the quantity of the new information is considerable, then it is able to follow the slowly varying parameters by due forgetting strategies.

In the case of certain, so-called predictive controllers, the process model is not identified directly, but finally in the algorithms, the determination of the process model is always present.