7

Nonlinear Acoustic Echo Cancellation

Fabian Küch and Walter Kellermann

Telecommunications Laboratory, University Erlangen-Nuremberg, Germany

While standard approaches for acoustic echo cancellation in telecommunication systems assume that the echo path to be identified can be modeled by a linear system, in practice, many loudspeaker systems involve non-negligible nonlinearities, e.g., caused by overloaded amplifiers due to low battery voltage of mobile communication receivers, or nonlinearities in the electroacoustic transduction as common with low-cost loudspeakers driven at high volume. Above a certain degree of nonlinear distortion, purely linear approaches are not able to provide a sufficient echo attenuation and nonlinear echo cancellers become desirable. Based on a nonlinear discrete-time model for the acoustic echo path we discuss different nonlinear adaptive structures for nonlinear acoustic echo cancellation and verify their effectiveness by measurements in real-world environments. While the frequency-dependent nonlinear behaviour of common electrodynamic loudspeakers can be modeled by Volterra filters, power filters are well suited to compensate memoryless saturation-type nonlinearities as they occur with overloaded amplifiers and miniaturized loudspeakers, e.g., in mobile phones.

7.1 Introduction

Linear adaptive filtering plays an important role in statistical signal processing and respective theoretical and practical results are well established [14]. In practice, however, nonlinear adaptive filtering often becomes desirable if the considered systems exhibit nonlinear behaviour. Acoustic echo cancellation represents an important example for such situations.

Standard approaches for the cancellation of acoustic echoes rely on the assumption that the echo path can be modeled by a linear system [5]. Accordingly, the acoustic echo canceller (AEC) is implemented as a linear filter. Since the echo path is unknown and, moreover, can change during operation of the echo canceller, the linear filter has to be realized adaptively. Unfortunately, the simple assumption of a linear echo path does not always hold in practice, as it does not include the behaviour of nonlinear audio hardware. The nonlinearly distorted components of the echo signal can not be captured by a linear AEC and, thus, are transmitted back to the far-end speaker who perceives an annoying copy of his own voice. Consequently, any non-negligible nonlinear distortion of the echo signal leads to a reduction of the echo attenuation achievable by purely linear approaches and, thus, impairs the quality of speech communication systems.

Possible sources for nonlinear distortion in the echo path are, e.g., small loudspeakers driven at high volume or overloaded amplifiers [32, 42, 43]. The problem of nonlinearly distorted echoes is especially common in mobile communication devices where high sound levels are desired with only low battery voltage available. For instance, in case of mobile phones operated in their hands-free mode, consumers usually prefer a nonlinear distortion of the loudspeaker signal over reduced output levels. Nonlinear echoes also occur in hands-free teleconferencing systems that include small-sized loudspeakers. If the consumer sets the loudspeaker system to its maximum volume, linear behaviour of small and/or cheap loudspeakers can not be expected anymore. The listening tests presented in [41] show that the accepted level of nonlinear distortion of speech is sufficiently high to cause annoying nonlinear echoes which can not be compensated by linear AECs.

To surpass echo cancellation performance of purely linear approaches, nonlinear methods have to be taken into consideration, where basically two approaches can be applied:

- nonlinear preprocessing of the loudspeaker signal,
- nonlinear adaptive filtering in the AEC.

The first approach aims at a linearization of the audio hardware components via nonlinear preprocessing of the received far-end signal. Then, the overall echo path to be modeled by the AEC consists of the acoustic echo path which is extended by the nonlinear preprocessing stage. In case of an ideal preprocessing of the loudspeaker signal, this overall echo path is linear and, thus, the AEC can also be realized as a linear filter. This approach can include methods known from the linearization of loudspeakers [8] and/or techniques that are used to compensate for the nonlinear distortion introduced by overloaded power amplifiers in digital communication systems [24]. Another method is to intentionally limit the excitation signal of the loudspeaker in order to avoid nonlinear behaviour of the loudspeaker and its amplifier. Note that in this case, the linear AEC has to be adapted with respect to the preprocessed signal. A major drawback imposed by these approaches is the required exact *a priori* knowledge of the nonlinearities of the loudspeaker system. This, however, implies that the nonlinear preprocessor of the echo cancellation unit can be designed only if the audio components of the loudspeaker system are accessible.

Here, we consider the more general approach of nonlinear adaptive filtering in the AEC in order to be more independent of the actual hardware in the loudspeaker system. It turns out that then only the type of nonlinearity in the acoustic echo path has to be known but not its exact properties, i.e., its parameter values. Here, we distinguish between two different types of nonlinear behaviour:

- nonlinearities with memory, as in case of a small loudspeaker driven at high volume,
- memoryless nonlinearities such as the saturation characteristic of overloaded amplifiers.

In the following, we apply certain polynomial filters [28] such as Volterra filters, truncated Taylor series expansions, and linear filters in order to model the behaviour of each of these nonlinearities. Based on these models, corresponding nonlinear filters can be developed which sufficiently model the overall nonlinear acoustic echo path. The goal is then to derive suitable adaptive algorithms for these nonlinear filters in order to provide a satisfying echo cancellation performance for the case that nonlinear audio hardware is included in telecommunication systems.

This chapter is organized as follows: In Sec. 7.2, we consider the properties of acoustic echo paths. After a discussion of nonlinear audio components, we introduce a discrete-time model of the acoustic echo path based on a nonlinear cascaded structure. The discussion of suitable adaptive approaches is divided into the following two sections. On the one hand, adaptive Volterra filters are considered in Sec. 7.3 and address nonlinearities with memory [28]. On the other hand, Sec. 7.4 focusses on the situations where the nonlinearity in the echo path can be considered as memoryless. It turns out that so-called power filters are more suitable than Volterra filters in this case [21]. The effectiveness of the discussed approaches in nonlinear acoustic echo cancellation is confirmed by experiments using real hardware.

7.2 Nonlinear Acoustic Echo Paths

For the design of nonlinear acoustic echo cancellers, it is essential to have sufficient knowledge about the properties of the underlying physical echo path. Therefore, we initially investigate the main components of typical acoustic echo paths. These results can then be used to obtain suitable nonlinear models for the identification of real echo paths.

The general structure of an acoustic echo path is illustrated in Fig. 7.1 and is common for hands-free telephone sets or mobile phones. The respective signal path is a cascade of digital-to-analog (D/A) converter, amplifier, loudspeaker, microphone, microphone preamplifier, and analog-to-digital (A/D) converter. Additionally, it comprises the acoustic propagation path of the speech signal between loudspeaker and microphone.

In general, the propagation path between loudspeaker and microphone can be considered as a linear system. It is commonly modeled by a linear FIR filter representing the room impulse response [5].

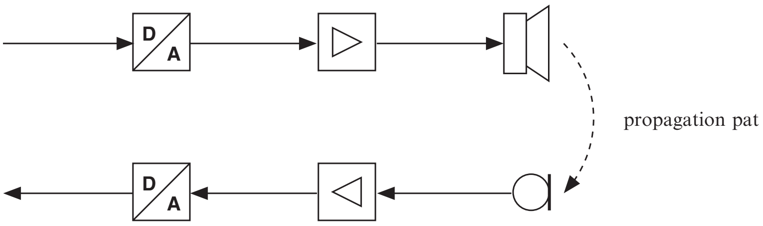


Fig. 7.1. Block diagram of the general structure of an acoustic echo path.

The microphone signals that are common with hands-free and mobile telephony have only moderate excitation levels. Thus, it is reasonable to assume a linear behaviour for the microphone and its preamplifier which is in accordance with the observations reported in [42].

From a system theoretical point of view, an ideal D/A converter can be described by an impulse response of a linear filter [34]. In practice, non-ideal hardware components can lead to a nonlinear mapping of the digital input signal to the analog output of the D/A converter [1, 34]. The same applies to A/D converters which, in addition, imply quantization of analog signals due to finite word lengths used for representation of digital signals. Early publications [1,6] address the problem of nonlinear network echo cancellation resulting from nonlinear D/A and A/D converters, respectively. With the modern, high-resolution converters used in todays telecommunication systems, it is mostly acceptable to neglect both, quantization errors, and any other nonlinear mapping characteristic caused by non-ideal signal conversion.

In this chapter we consider two sources for nonlinear distortion: the loudspeaker and its amplifier. The properties of these nonlinear system components are discussed next.

Amplifier nonlinearities are especially present in mobile communication devices. There, the dilemma arises to provide high signal levels while having only low battery voltage available. The consumers usually prefer an overloading of the amplifier over a reduction of the sound volume. The nonlinear behaviour of amplifiers can therefore be described as saturation characteristic with a soft clipping of large amplitude values [42]. Due to the limited bandwidth of telephone signals, amplifiers applied to audio applications can in general be considered as memoryless.

Many research efforts aimed at the characterization of the nonlinearities of electrodynamic loudspeakers [16, 37]. Summarizingly, one can distinguish between three different parts of the loudspeaker which may introduce nonlinear distortion: the acoustical part, the electromagnetic part, and the mechanical part. Nonlinearities in the acoustical part, such as nonlinear wave propagation play an important role in the modeling and linearization of horn loudspeakers [17]. However, as this type of loudspeaker is generally used in public announcement systems only, any nonlinearities caused by sound radiation are not considered here.

The nonlinearities in the electromagnetic part (also referred to as motor part) are mainly caused by the asymmetries of the magnetic flux, and its decay outside the air gap of the motor. Thus, the driving force on the voice coil is a nonlinear function of its position. Additionally, the self-inductance of the voice coil depends on its displacement, too.

In the mechanical part, the nonlinear dependency of the stiffness of the spider and the outer rim on the position of the voice coil has to be taken into account. It is worth mentioning that the characteristic of this nonlinearity is slowly time-varying, as the mechanical properties of the spider and the rim are changing in time due to changes in temperature and aging effects of the used materials.

A common approach to incorporate the above-mentioned nonlinearities into a loudspeaker model is to approximate the various nonlinear characteristics by a truncated Taylor series expansion for the decisive parameters. Then, the approximated parameters are introduced into the differential equations that describe the behaviour of the loudspeaker [16, 37]. However, such a representation of the loudspeaker is out of scope of this chapter, as this is not suited for adaptive realizations as required for the echo cancellation application. Consequently, we exploit the main result of [16, 37], i.e., the nonlinear behaviour of loudspeakers will be modeled by an appropriate Volterra filter. More precisely, we consider the loudspeaker as a black box, the input/output relation of which can sufficiently well be approximated by a second-order Volterra filter. On the other hand, the results presented in [23] indicate that for acoustic echo cancellation in mobile phones, a saturation-type behaviour of the miniaturized loudspeakers can be expected. In this case, specialized third-order polynomial filters with less memory support represent a more suitable choice.

Other sources for nonlinear distortion in loudspeaker systems are given by rattling and vibration effects caused by a strong physical coupling between loudspeaker, microphone, and their enclosure, as, e.g., common in mobile phones. However, this distortion can hardly be modeled or predicted, as it is of chaotic nature [3]. It should rather be considered as uncorrelated noise (analogously to any background noise) and, thus, be processed accordingly. The problem of vibrating system components is, however, not further considered here.

Furthermore, mechanical clipping can be observed in the loudspeaker for very high excitation levels if the available displacement range for the voice coil is not sufficiently large [37, 40]. This problem, however, is also not further considered here.

Discrete-Time Model for the Echo Path

As a result of the above discussion, we are now able to derive a discrete-time model for the acoustic echo path that is common in acoustic echo cancellation. Such a model can consist of the cascade of different linear and nonlinear components as illustrated by the block diagram Fig. 7.2. Obviously, the cascaded structure is a direct consequence of the cascade of the different system components shown in Fig. 7.1.

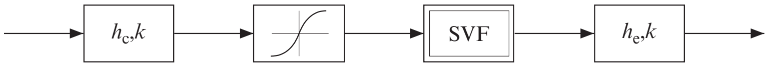


Fig. 7.2. Block diagram of the nonlinear model of the acoustic echo path.

The first block in Fig. 7.2 is the linear FIR filter $h_{c,k}$ representing the combination of all linear filtering steps involved in the D/A-conversion. Following [42], we assume that the loudspeaker amplifier can be regarded as memoryless soft clipping and, thus, be approximated by a truncated Taylor series expansion. Although a seventh-order Taylor series has been applied in [42] to model the behaviour of the amplifier, simulation results have shown that a third-order polynomial can sufficiently reproduce the influence of the amplifier nonlinearity. The corresponding block in Fig. 7.2 illustrates the soft clipping by showing a corresponding mapping characteristic. Note that for mobile phones, the level of nonlinear distortion introduced by the amplifier may depend on the charge level of the battery that provides the power supply. Thus, the parameters of the polynomial representing the amplifier have to be at least slowly time-variant.

The third block in Fig. 7.2 represents a second-order Volterra filter (SVF) that is used to simulate the loudspeaker nonlinearities. As already mentioned, the mechanical contribution to the nonlinear distortion is not constant over time due to fatigue of material. Consequently, the coefficients of the Volterra filter should also be at least slowly time-variant. In case of mobile phones, the nonlinear behaviour of miniaturized loudspeakers exhibit a memoryless saturation characteristic [23]. Then, the Volterra filter can be replaced by a Taylor series expansion. The cascade of the loudspeaker and its amplifier can thus be modeled by using only a single truncated Taylor series expansion, i.e., by discarding the third block in Fig. 7.2.

The last block comprises three cascaded linear models, representing the sound propagation path between loudspeaker and microphone, the microphone characteristic (including its preamplifier), and the A/D converter, respectively. As the linear FIR filter $h_{e,k}$ includes the room impulse response, it may be rapidly time-variant.

It can be shown that every parallel/cascaded combination of linear filters, truncated Taylor series expansions, and Volterra filters can be replaced by a corresponding Volterra filter exhibiting the same input/output relation. It is straightforward to verify that for the above model of the nonlinear echo path, this results in a fifth-order Volterra filter if the amplifier is represented by a third-order polynomial and a second-order Volterra filter is used as loudspeaker model. This approach, however, is not practicable due to the enormous number of required coefficients for higher-order Volterra filters [28].

Simplifications of the general model of the echo path according to Fig. 7.2 can be achieved if any *a priori* knowledge about the properties of the system to be identified can be exploited. In the following we assume that at least one of the nonlinear components in Fig. 7.2 can be neglected: In Sec. 7.3 we look at the case where solely the second-order Volterra filter is included in the actual model of the acoustic echo path, whereas in Sec. 7.4 the Taylor series expansion is considered as the only nonlinear component. Furthermore, the cascaded nature of the echo path can be taken into account for deriving more efficient overall models of the nonlinear echo path.

7.3 Volterra Filters

For the case that the small-sized loudspeaker of a hands-free telecommunication device represents the main source for nonlinear distortion in the echo path, it has to be modeled by a second-order Volterra filter. In this section we therefore discuss Volterra filters and corresponding adaptive realizations in both, time domain and frequency domain. The basic concepts of adaptive Volterra filtering are presented for the general case of P-th order Volterra filters. In more specific parts such as the control of the adaptation or the evaluation of the presented algorithms, we explicitly refer to the acoustic echo cancellation application and limit ourselves to second-order Volterra filters.

The output $d(n)$ of a P-th order Volterra filter is composed of the sum of the outputs of all kernels up to order P:

$$d(n) = \sum_{p=1}^{P} d_p(n). \tag{7.1}$$

Most commonly, the input/output relation of the p-th order kernel is expressed by [28]

$$d_p(n) = \sum_{k_{p,1}=0}^{N_p-1} \sum_{k_{p,2}=k_{p,1}}^{N_p-1} \cdots \sum_{k_{p,p}=k_{p,p-1}}^{N_p-1} h_{\boldsymbol{k}_p} \prod_{i=1}^{p} x(n-k_{p,i}), \tag{7.2}$$

where the memory lengths N_p of the Volterra kernels can in general be different for each order p. In Eq. 7.9, the index vector

$$\boldsymbol{k}_p = [k_{p,1},\, k_{p,2},\, \ldots,\, k_{p,p}] \tag{7.3}$$

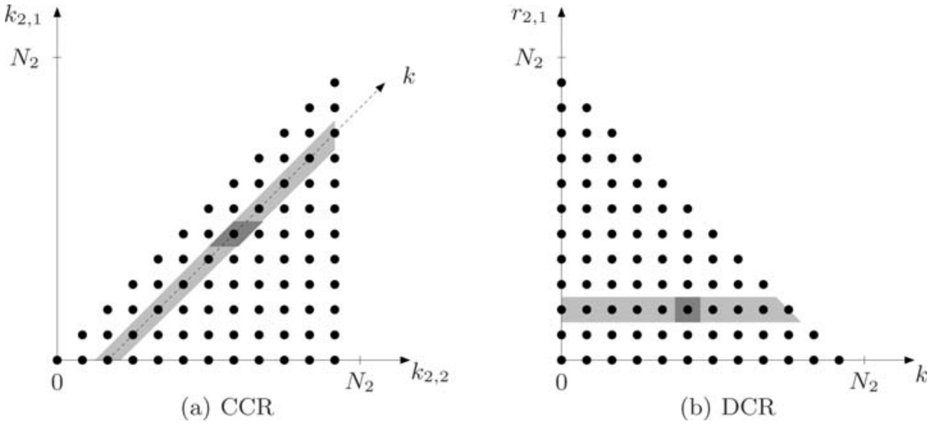
can be interpreted as reference to a certain coefficient $h_{\boldsymbol{k}_p}$ of the p-th order Volterra kernel in a p-dimensional Cartesian coordinate system. Thus, Eq. 7.2 is referred to as Cartesian coordinate representation (CCR) of Volterra filters [35]. As can be noticed from Eq. 7.2, there is a strong relation between *multidimensional linear filtering* and Volterra filters in CCR [28].

In the following we consider an alternative representation of Volterra filters which turns out to be more useful in the nonlinear echo cancellation context featuring nonlinear cascaded structures. Regarding [35], we apply the following change of coordinates:

$$k_{p,1} = k, \quad 0 \leq k \leq N_p - 1, \tag{7.4}$$

$$k_{p,i} = r_{p,i-1} + k, \quad 2 \leq i \leq p. \tag{7.5}$$

For interpreting the above coordinate transform, we recall that the set of indices $k_{p,1}$, $k_{p,2}$, $\ldots$, $k_{p,p}$ can be considered as Cartesian coordinates which corresponds to the p-dimensional sampled hypercube representing the p-th order Volterra kernel. The combination of the indices $r_{p,1}$, $r_{p,2}$, $\ldots$, $r_{p,p-1}$, and k can then be understood as reference to the kernel coefficients lying on a straight line which is parallel to the main diagonal of the Cartesian coordinate system. Following [35], we refer to these straight lines as diagonals, where the main diagonal is defined by setting $k_{p,1} = k_{p,2} = \ldots = k_{p,p}$ which implies $r_{p,1} = r_{p,2} = \ldots = r_{p,p-1} = 0$. Based on these interpretations, we consider the new set of indices $r_{p,1}$, $r_{p,2}$, $\ldots$, $r_{p,p-1}$, and k as coordinates of the so-called diagonal coordinate system. The relation between the CCR and the diagonal coordinate representation (DCR) is illustrated in Fig. 7.3 for the quadratic Volterra kernel. As an example, the diagonal corresponding to $r_{2,1} = 2$ is highlighted in both figures. Additionally, the dark quadrangles mark the indices $(k_{2,1}, k_{2,2}) = (5, 7)$ and $(r_{2,1}, k) = (2, 5)$, respectively, which reference the same kernel coefficient.



(a) CCR

(b) DCR

Fig. 7.3. Illustration of the relation between the CCR and DCR for a quadratic Volterra kernel ($p = 2$). Each ● corresponds to a kernel coefficient.

Analogously to the coefficient index vector $\boldsymbol{k}_p$ in Eq. 7.3, we introduce the two coefficient vectors

$$\boldsymbol{r}_p = [r_{p,1}, r_{p,2}, \ldots, r_{p,p-1}], \tag{7.6}$$

$$\boldsymbol{r}_{p,k} = [k, r_{p,1} + k, \ldots, r_{p,p-1} + k], \tag{7.7}$$

where for the linear kernel $\boldsymbol{r}_1 = [\,]$ and $\boldsymbol{r}_{1,k} = k$. The diagonal index vector $\boldsymbol{r}_p$ references a certain diagonal, whereas the corresponding coefficient index vector $\boldsymbol{r}_{p,k}$ references a certain kernel coefficient on that diagonal. Note that $\boldsymbol{r}_p$ has the length $p-1$, while $\boldsymbol{r}_{p,k}$ consists of p elements. The diagonal index vector associated with the main diagonal is obviously given by $\boldsymbol{r}_p = \mathbf{0}$. Thus, the index vector elements $r_{p,i}$ of $\boldsymbol{r}_p$ determine the distance of that diagonal from the main diagonal of the p-th order Cartesian coordinate system. It should be pointed out that with the definition of the coefficient index vector $\boldsymbol{r}_{p,k}$ in Eq. 7.7, the notation of the kernel coefficients have been kept unchanged, i.e.,

$$h_{\boldsymbol{r}_{p,k}} = h_{\boldsymbol{k}_p}, \quad \text{if } \boldsymbol{r}_{p,k} = \boldsymbol{k}_p. \tag{7.8}$$

The desired form of the input/output relation of the p-th order kernel is obtained by introducing the new index vectors $\boldsymbol{r}_p$ and $\boldsymbol{r}_{p,k}$ into Eq. 7.2 and by additionally changing the order of summation:

$$d_p(n) = \sum_{r_{p,1}=0}^{N_p-1} \cdots \sum_{r_{p,p-1}=r_{p,p-2}}^{N_p-1} \sum_{k=0}^{L_{\boldsymbol{r}_p}-1} h_{\boldsymbol{r}_{p,k}}\, x(n-k) \prod_{i=1}^{p-1} x(n - r_{p,i} - k), \tag{7.9}$$

where

$$L_{\boldsymbol{r}_p} = N_p - r_{p,p-1}, \tag{7.10}$$

obviously depends on both, the kernel order p, and the actual value of $r_{p,p-1}$. For the linear kernel ($p = 1$) we have $L_{\boldsymbol{r}_1} = N_1$. As proposed in [35], we refer to Volterra filters featuring the above form for computing the output of the p-th order Volterra kernel as Volterra filters in diagonal coordinate representation (DCR).

In the following we examine the relation between Volterra filters in DCR and *multichannel linear filtering*. For a deeper insight into the internal multichannel structure of the DCR, we introduce the input signal of the diagonal $\boldsymbol{r}_p$ according to

$$x_{\boldsymbol{r}_p}(n) = x(n) \prod_{i=1}^{p-1} x(n - r_{p,i}), \tag{7.11}$$

where $x_{\boldsymbol{r}_1}(n) = x(n)$. The corresponding output $d_{\boldsymbol{r}_p}(n)$ of the diagonal with index $\boldsymbol{r}_p$ is then given by

$$d_{\boldsymbol{r}_p}(n) = \sum_{k=0}^{L_{\boldsymbol{r}_p}-1} h_{\boldsymbol{r}_{p,k}}\, x_{\boldsymbol{r}_p}(n-k). \tag{7.12}$$

Obviously, $d_{\boldsymbol{r}_p}(n)$ can be considered as the output of the linear FIR filter $h_{\boldsymbol{r}_{p,k}}$ of length $L_{\boldsymbol{r}_p}$ with input $x_{\boldsymbol{r}_p}(n)$. In other words, $d_{\boldsymbol{r}_p}(n)$ results from the convolution of $x_{\boldsymbol{r}_p}(n)$ with the linear filter $h_{\boldsymbol{r}_{p,k}}$ and can therefore be expressed by

$$d_{\boldsymbol{r}_p}(n) = h_{\boldsymbol{r}_{p,n}} * x_{\boldsymbol{r}_p}(n), \tag{7.13}$$

where $*$ denotes convolution. The above definitions are used to rewrite the output $d_p(n)$ of the p-th order kernel according to Eq. 7.9:

$$d_p(n) = \sum_{r_{p,1}=0}^{N_p-1} \cdots \sum_{r_{p,p-1}=r_{p,p-2}}^{N_p-1} d_{\boldsymbol{r}_p}(n). \tag{7.14}$$

From the specific form of Eq. 7.14 we notice that $d_p(n)$ can be interpreted as the output of a linear multiple input/single output (MISO) system, where each diagonal with index vector $\boldsymbol{r}_p$ corresponds to one linear channel with input $x_{\boldsymbol{r}_p}(n)$. Extending this interpretation to the computation of the output of the Volterra filter according to Eq. 7.1, $d(n)$ can be considered as the output of a special MISO system featuring a combination of P multichannel structures, where each channel corresponds to one particular diagonal of the DCR.

Aiming at a compact vector notation for the computation of $d_p(n)$, we introduce the input signal vectors $\boldsymbol{x}_p(n)$ associated to the p-th order kernel vector $\boldsymbol{h}_p$ according to

$$\boldsymbol{x}_p(n) = \left[\ldots, x(n) \prod_{i=1}^{p-1} x(n - r_{p,i}), \ldots\right]^{\mathrm{T}}, \tag{7.15}$$

$$\boldsymbol{h}_p = \left[\ldots, h_{\boldsymbol{r}_{p,k}}, \ldots\right]^{\mathrm{T}}. \tag{7.16}$$

The $\binom{N_p+p-1}{p}$ elements of $\boldsymbol{x}_p(n)$ and $\boldsymbol{h}_p$ can in principle be arranged arbitrarily according to any given preferences. Of course, the elements in $\boldsymbol{x}_p(n)$ and $\boldsymbol{h}_p$ have to be arranged consistently such that

$$d_p(n) = \boldsymbol{h}_p^{\mathrm{T}} \boldsymbol{x}_p(n). \tag{7.17}$$

With the definitions of the vectors

$$\boldsymbol{x}_{\mathrm{VF}}(n) = \left[\boldsymbol{x}_1^{\mathrm{T}}(n), \boldsymbol{x}_2^{\mathrm{T}}(n), \ldots, \boldsymbol{x}_P^{\mathrm{T}}(n)\right]^{\mathrm{T}}, \tag{7.18}$$

$$\boldsymbol{h}_{\mathrm{VF}} = \left[\boldsymbol{h}_1^{\mathrm{T}}, \boldsymbol{h}_2^{\mathrm{T}}, \ldots, \boldsymbol{h}_P^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{7.19}$$

we can finally extend the vector notation also to the computation of the overall Volterra filter output $d(n)$:

$$d(n) = \boldsymbol{h}_{\mathrm{VF}}^{\mathrm{T}} \boldsymbol{x}_{\mathrm{VF}}(n). \tag{7.20}$$

We notice that Eq. 7.20 reflects the linearity of the output $d(n)$ with respect to the Volterra filter coefficients which are summarized in $\boldsymbol{h}_{\mathrm{VF}}$. This formal analogy to linear filtering can be exploited in order to straightforwardly extend adaptive algorithms known from linear adaptive filtering to Volterra filters.

7.3.1 Application to Cascaded Structures

In the following, we look at the configuration according to Fig. 7.4 in more detail which consists of the cascade of a Volterra filter $h_{\boldsymbol{r}_{p,k}}$ followed by a linear filter c_k.

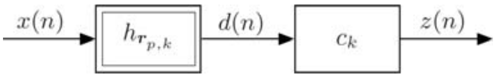


Fig. 7.4. Cascaded structure consisting of a second-order Volterra filter $h_{\boldsymbol{r}_{p,k}}$ followed by a linear FIR filter c_k.

First, we recall the assumption that the model of the echo path Fig. 7.2 can be simplified to the cascade of a linear filter, a second-order Volterra filter, and another linear filter. The cascade of a linear filter followed by a Volterra filter can be represented by a corresponding Volterra filter of the same order but with increased memory length. Thus, the assumed simplified model of the acoustic echo path represents a special case of the configuration shown in Fig. 7.4.

From an efficiency point of view, one might tend to directly use an adaptive implementation of the two cascaded units for realizing the nonlinear acoustic echo canceller. This approach has already been proposed in [11] for acoustic echo cancellation including nonlinearly distorting loudspeakers. However, it is challenging to assure convergence to the optimum solution or even assure a stable adaptation behaviour for cascaded structures. This problem has also been observed by the authors of [11]. As a remedy, they suggest to adapt the Volterra filter only after the linear postfilter has ’sufficiently’ converged. In order to circumvent any sophisticated adaptation control as required for the adaptation of cascades, we consider an equivalent Volterra model as a parallelized realization of the cascaded structure instead. It turns out that the DCR provides an elegant representation of such equivalent Volterra models.

As the convolution is a linear operation, the computation of the output of thecascaded structure directly follows from Eqs. 7.1, 7.13, 7.14:

$$z(n) = \sum_{p=1}^{P} z_p(n), \tag{7.21}$$

$$z_p(n) = \sum_{r_{p,1}=0}^{N_p-1} \sum_{r_{p,2}=r_{p,1}}^{N_p-1} \cdots \sum_{r_{p,p-1}=r_{p,p-2}}^{N_p-1} z_{\boldsymbol{r}_p}(n), \tag{7.22}$$

where the outputs $z_{\boldsymbol{r}_p}(n)$ of the respective DCR-channel read

$$\begin{aligned} z_{\boldsymbol{r}_p}(n) &= c_n * h_{\boldsymbol{r}_{p,n}} * x_{\boldsymbol{r}_p}(n) \\ &= g_{\boldsymbol{r}_{p,n}} * x_{\boldsymbol{r}_p}(n). \end{aligned} \tag{7.23}$$

We note from Eqs. 7.22, 7.23 that $z_p(n)$ can be considered as the output of a special p-th order Volterra kernel $g_{\boldsymbol{r}_{p,k}}$ with input $x(n)$, where the number and the position of the diagonals are not changed compared to the Volterra kernel $h_{\boldsymbol{r}_{p,k}}$. However, the length of the filter in each DCR-channel with index vector $\boldsymbol{r}_p$ is increased according to

$$\widetilde{L}_{\boldsymbol{r}_p} = L_{\boldsymbol{r}_p} + N_c - 1, \tag{7.24}$$

where N_c denotes the length of the linear filter c_k. Obviously, the corresponding CCR of the kernel $g_{\boldsymbol{r}_{p,k}}$ has the overall memory length $N_p + N_c - 1$, where only a corridor with respect to the main diagonal of width N_p has non-zero coefficients. The resulting region of support of the Volterra kernels, i.e., the non-zero coefficients, is illustrated in Fig. 7.5 for the quadratic kernel and the special case $N_2 = 4$ and $N_c = 16$. Comparing Fig. 7.5 with Fig. 7.3(b), the specific shape of the region of support becomes clear.

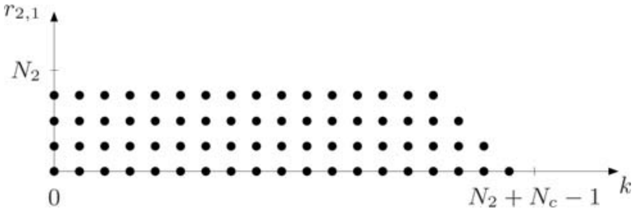


Fig. 7.5. Illustration of the quadratic Volterra kernel corresponding to the cascaded structure according to Fig. 7.4 for the special case $N_2 = 4$ and $N_c = 16$. Each • corresponds to a non-zero kernel coefficient.

The reduced region of support as required for Volterra models of cascaded structures according to Fig. 7.4, can easily be taken into account by appropriately modifying Eq. 7.14:

$$d_p(n) = \sum_{r_{p,1}=0}^{R_p-1} \sum_{r_{p,2}=r_{p,1}}^{R_p-1} \cdots \sum_{r_{p,p-1}=r_{p,p-2}}^{R_p-1} d_{\boldsymbol{r}_p}(n). \tag{7.25}$$

The parameter R_p is used here to specify the maximum distance of a diagonal with respect to the main diagonal. Introducing Eq. 7.12 into Eq. 7.25 yields

$$d_p(n) = \sum_{r_{p,1}=0}^{R_p-1} \sum_{r_{p,2}=r_{p,1}}^{R_p-1} \cdots \sum_{r_{p,p-1}=r_{p,p-2}}^{R_p-1} \sum_{k=0}^{L_{\boldsymbol{r}_p}-1} h_{\boldsymbol{r}_{p,k}}\, x_{\boldsymbol{r}_p}(n-k). \tag{7.26}$$

The output $d_{\boldsymbol{r}_p}(n)$ of the diagonal with index vector $\boldsymbol{r}_p$ is still computed according to Eq. 7.12, implying that the linear filter of the corresponding channel has the memory length $L_{\boldsymbol{r}_p} = N_p - r_{p,p-1}$. Choosing $R_p < N_p$ yields the desired reduced region of support compared to the case $R_p = N_p$ (as imposed by Eq. 7.14). The possibility to reduce the width R_2 of the quadratic kernel without impairing the echo cancellation performance of an adaptive second-order Volterra filter is exemplified in Sec. 7.3.4 for a real acoustic echo path.

Throughout the rest of this chapter, we always refer to the extended definition Eq. 7.26 when considering Volterra filters in DCR. Moreover, we restrict ourselves to the case $R_p \leq N_p$. Then, N_p still represents the *memory length*[1] of the p-th order kernel, whereas R_p is referred to as its *width*. It should be emphasized that the distinction between R_p and N_p does not imply an additional degree of freedom for the design of Volterra filters: We only explicitly exclude certain coefficients of the corresponding CCR which are *a priori* assumed to be zero.

The number of diagonals $N_{\text{diag},p}$ included in the p-th order kernel with width R_p is given by

$$N_{\text{diag},p} = \binom{R_p + p - 2}{p - 1}. \tag{7.27}$$

Note that only for $p \leq 2$, i.e., for the linear and the quadratic kernel, the width R_p equals to the number of diagonals. For the linear kernel R_1 obviously always equals one. The number of coefficients $N_{\text{coeff},p}$ of the p-th order kernel with memory length N_p and width R_p is obtained as

$$N_{\text{coeff},p} = \binom{R_p + p - 1}{p} + (N_p - R_p)N_{\text{diag},p} \tag{7.28}$$

with $N_{\text{diag},p}$ according to Eq. 7.27.

7.3.2 Time-domain Adaptive Volterra Filters

The fundamental problem of adaptive Volterra filtering in acoustic echo cancellation is illustrated in Fig. 7.6. From Fig. 7.6 we notice that the require-

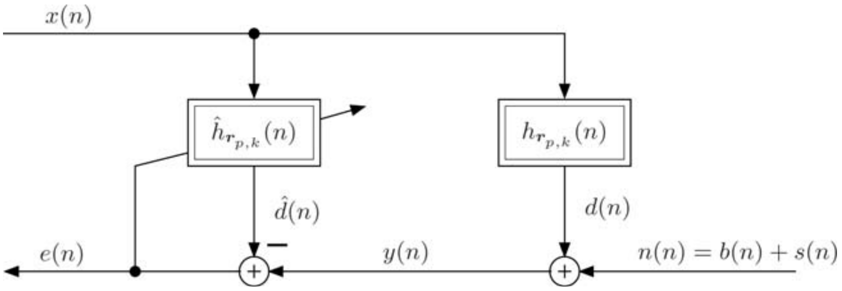


Fig. 7.6. General configuration for adaptive Volterra filtering in acoustic echo cancellation.

[1] Strictly speaking, the memory length is $N_p - 1$.

ments for adaptive Volterra filtering are basically equivalent to the task of a linear adaptive filtering in the echo cancellation context: The coefficients $\hat{h}_{\boldsymbol{r}_{p,k}}(n)$ of the adaptive Volterra filter have to be determined such that $\hat{d}(n)$ matches the output of the unknown system $d(n)$. As already indicated in Fig. 7.6, we assume in the following that the unknown system (i.e., the acoustic echo path) can can be characterized by the Volterra filter coefficients $h_{\boldsymbol{r}_{p,k}}(n)$.

Using the notation of Fig. 7.6, the error $e(n)$ is defined as

$$e(n) = y(n) - \hat{d}(n), \tag{7.29}$$

where the output of the P-th order adaptive Volterra filter reads

$$\hat{d}(n) = \hat{\boldsymbol{h}}_{\mathrm{VF}}^{\mathrm{T}}(n)\, \boldsymbol{x}_{\mathrm{VF}}(n). \tag{7.30}$$

The coefficient vector $\hat{\boldsymbol{h}}_{\mathrm{VF}}(n)$ is defined analogously to Eqs. 7.16, 7.19, but contains the kernel coefficients $\hat{h}_{\boldsymbol{r}_{p,k}}(n)$ of the adaptive Volterra filter instead of $h_{\boldsymbol{r}_{p,k}}(n)$. The output $d(n)$ of the unknown system is given by Eqs. 7.1, 7.26, i.e., we assume that it can be completely modeled by a P-th order Volterra filter. In the following we additionally assume that the order P and the memory lengths N_p are equal for both, the adaptive Volterra filter, and the unknown system.

The observed microphone signal $y(n)$ is given by

$$y(n) = d(n) + b(n) + s(n), \tag{7.31}$$

where $d(n)$ represents the actual echo signal. The external distortions $b(n)$ and $s(n)$ represent background noise and local speech, respectively, and are summarized to

$$n(n) = b(n) + s(n). \tag{7.32}$$

The residual echo $\varepsilon(n)$ is given by

$$\varepsilon(n) = d(n) - \hat{d}(n). \tag{7.33}$$

Similarly to linear adaptive filtering, the LMS algorithm represents the most commonly used adaptation algorithm for Volterra filters [27, 28] mainly because of its simplicity and robustness. This is especially important since Volterra filters imply a huge number of coefficients to be adapted, as can be noticed from Eq. 7.28. The update equation for the coefficient vector $\hat{\boldsymbol{h}}_{\mathrm{VF}}(n)$, applying the LMS algorithm, is given by [27]

$$\hat{\boldsymbol{h}}_{\mathrm{VF}}(n+1) = \hat{\boldsymbol{h}}_{\mathrm{VF}}(n) + \mu_{\mathrm{LMS}}(n)\, e(n)\boldsymbol{x}_{\mathrm{VF}}(n). \tag{7.34}$$

The step size control parameter $\mu_{\mathrm{LMS}}(n)$ can be chosen to vary for different coefficients, as discussed later in this section.

The standard NLMS algorithm for Volterra filters is obtained by normalizing the step size parameter $\mu_{\mathrm{LMS}}(n)$ according to

$$\mu_{\mathrm{LMS}}(n) = \frac{\alpha_{\mathrm{NLMS}}(n)}{\boldsymbol{x}_{\mathrm{VF}}^{\mathrm{T}}(n)\boldsymbol{x}_{\mathrm{VF}}(n)}, \quad 0 < \alpha_{\mathrm{NLMS}}(n) < 2. \tag{7.35}$$

The given range for the step size $\alpha_{\mathrm{NLMS}}(n)$ indicates the range where stable convergence can be expected [27].

Although the above normalization formally yields the NLMS algorithm for Volterra filters, it might not always be useful in practice. From the definition of $\boldsymbol{x}_{\mathrm{VF}}(n)$ in Eq. 7.18 it follows that the denominator of Eq. 7.35 is composed of the sum over different moments of $x(n)$, up to order $2P$. In general, the orders of magnitude of these moments significantly differ and, thus, the joint normalization of all Volterra kernels according to Eq. 7.35 is not suitable for higher-order Volterra filters. Considering

$$\boldsymbol{x}_p^{\mathrm{T}}(n)\boldsymbol{x}_p(n) \ll \max_i \left\{ \boldsymbol{x}_i^{\mathrm{T}}(n)\boldsymbol{x}_i(n) \right\}, \tag{7.36}$$

we realize that the adaptation of the coefficients of the p-th order kernel almost freezes for a joint normalization of the step size $\alpha_{\mathrm{NLMS}}(n)$.

There exist also more sophisticated algorithms for the adaptation of Volterra filters which can be used to circumvent the inherent slow convergence of the LMS algorithm. Prominent examples are the Affine Projection Algorithm (APA) [7] or the RLS algorithm [27] which are also well known in linear adaptive filtering [9]. Note that due to the huge number of coefficients that are associated with higher-order Volterra filters, the APA and the RLS algorithm are usually not realizable in practice.

There is another major difference between the matrix representation of linear filters and the matrix representation of Volterra filters: The input vector $\boldsymbol{x}_{\mathrm{VF}}(n)$ does not exhibit the tapped delay line structure of the input vector $\boldsymbol{x}(n) = \boldsymbol{x}_1(n)$ as used in linear filtering. Assuming stationary input $x(n)$, the autocorrelation matrix

$$\boldsymbol{R}_{\boldsymbol{xx}} = \mathrm{E}\left\{ \boldsymbol{x}(n)\boldsymbol{x}^{\mathrm{T}}(n) \right\}, \tag{7.37}$$

associated with the input vector $\boldsymbol{x}(n)$ for linear filters, features a Toeplitz form, whereas this is not true for the autocorrelation matrix

$$\boldsymbol{R}_{\boldsymbol{x}_{\mathrm{VF}}\boldsymbol{x}_{\mathrm{VF}}} = \mathrm{E}\left\{ \boldsymbol{x}_{\mathrm{VF}}(n)\boldsymbol{x}_{\mathrm{VF}}^{\mathrm{T}}(n) \right\}, \tag{7.38}$$

corresponding to the input vector $\boldsymbol{x}_{\mathrm{VF}}(n)$ of Volterra filters. Unfortunately, computationally efficient versions of the RLS and the APA for linear adaptive filters explicitly exploit the Toeplitz structure of $\boldsymbol{R}_{\boldsymbol{xx}}$ [9]. Therefore, these methods cannot be applied to adaptive Volterra filters in a straightforward manner. Nevertheless, there are still structural features in the input vector $\boldsymbol{x}_{\mathrm{VF}}(n)$ on which fast versions of the RLS algorithm for Volterra filters can be based on, as has been shown in [25] for the second-order case.

Adaptation Control for Second-order Volterra Filters

The performance of an adaptive algorithm strongly depends on the control of the adaptation. This is especially true for the LMS algorithm, as it implies rather slow convergence for correlated input signals. In the following, we therefore present a coefficient-dependent adaptation control for the LMS algorithm for second-order Volterra filters that corresponds to the approach proposed in [20].

Since we are aiming at a coefficient-dependent step size, we consider the LMS update equation for a single coefficient $\hat{h}_{\boldsymbol{r}_{p,k}}(n)$. From Eq. 7.34 we obtain

$$\hat{h}_{\boldsymbol{r}_{p,k}}(n+1) = \hat{h}_{\boldsymbol{r}_{p,k}}(n) + \mu_{\boldsymbol{r}_{p,k}}(n)\, e(n)\, x_{\boldsymbol{r}_p}(n-k). \tag{7.39}$$

For the following discussion, we assume that the input $x(n)$ is an independent, identically distributed (IID) random process, where the probability density function (PDF) of the amplitudes of $x(n)$ is an even function. Additionally, we assume that the coefficients of the adaptive Volterra filter are independent of the input signal. The assumed properties of $x(n)$ imply that the input of the linear kernel and the output of the quadratic kernel of both, the adaptive Volterra filter, and the unknown Volterra filter are orthogonal, i.e., $\mathrm{E}\{\, y_2(k)x(n-k)\,\} = \mathrm{E}\{\, \hat{y}_2(k)x(n-k)\,\} = 0$. It can be shown that then, the optimum filter coefficients of the linear adaptive kernel $\hat{h}_{\boldsymbol{r}_{1,k}}(n)$ are equal to the corresponding filter coefficients $h_{\boldsymbol{r}_{1,k}}(n)$. Correspondingly, it can be shown that the optimum coefficients for the quadratic kernel $\hat{h}_{\boldsymbol{r}_{2,k}}(n)$ are given by $h_{\boldsymbol{r}_{2,k}}(n)$. Thus, we introduce the coefficient errors of the linear and the quadratic kernels according to

$$m_{\boldsymbol{r}_{p,k}}(n) = h_{\boldsymbol{r}_{p,k}}(n) - \hat{h}_{\boldsymbol{r}_{p,k}}(n), \quad p \in \{1,2\}. \tag{7.40}$$

Following [20], the optimality criterion for determining the optimum coefficient-dependent step size $\mu_{\mathrm{opt},\boldsymbol{r}_{p,k}}(n)$ is chosen as the mean squared error between the actual coefficient error and the corresponding LMS update term:

$$J_{\mu_{\boldsymbol{r}_{p,k}}}(n) = \mathrm{E}\left\{ \left[m_{\boldsymbol{r}_{p,k}}(n) - \mu_{\boldsymbol{r}_{p,k}}(n)\, e(n)\, x_{\boldsymbol{r}_p}(n-k)\right]^2 \right\}, \quad p \in \{1,2\}. \tag{7.41}$$

As shown in [20], the optimum step size, which minimizes the cost function Eq 7.41, is obtained as

$$\mu_{\mathrm{opt},\boldsymbol{r}_{p,k}}(n) = \frac{\mathrm{E}\left\{ m_{\boldsymbol{r}_{p,k}}^2(n) \right\}}{\mathrm{E}\left\{ \varepsilon^2(n) + b^2(n) + s^2(n) \right\}}, \quad p \in \{1,2\}, \tag{7.42}$$

where it has been assumed that the input $x(n)$, the background noise $b(n)$, and the speech signal of the near-end talker $s(n)$ are mutually statistically independent processes. Interestingly, the form of the optimum step size is identical for both kernels, which results from the linearity of the output with respect to the coefficients of any kernel. Considering that the error signal can be expressed as

$$e(n) = \varepsilon(n) + b(n) + s(n), \tag{7.43}$$

we notice that the denominator in Eq. 7.42 can be identified as the second-order moment of the error signal $e(n)$. Since $e(n)$ is observable, its second-order moment can in general be estimated reliably. However, the mean squared coefficient errors $\mathrm{E}\left\{ m_{\boldsymbol{r}_{p,k}}^{2}(n) \right\}$ are not observable in practice, making a straightforward realization of the optimum step size impossible. Therefore, we apply the approach proposed in [20] in order to obtain models for the estimation of the unknown statistical terms.

For the following discussion it will be useful to distinguish between the residual echoes associated to each single Volterra kernel, and define

$$\varepsilon_p(n) = d_p(n) - \hat{d}_p(n). \tag{7.44}$$

The overall residual echo for second-order Volterra filters can then be expressed by

$$\varepsilon(n) = \varepsilon_1(n) + \varepsilon_2(n). \tag{7.45}$$

Note that for the assumed input signal $x(n)$, the residual echoes of the different Volterra kernels are orthogonal, and, thus,

$$\mathrm{E}\left\{ \varepsilon^2(n) \right\} = \mathrm{E}\left\{ \varepsilon_1^2(n) \right\} + \mathrm{E}\left\{ \varepsilon_2^2(n) \right\}. \tag{7.46}$$

As the residual echoes $\varepsilon_p(n)$ result from the misadjustment of the corresponding kernel coefficients, we rewrite Eq. 7.44 for the linear and the quadratic kernel in terms of the coefficient errors:

$$\varepsilon_1(n) = \sum_{k=0}^{N_1-1} m_{\boldsymbol{r}_{1,k}}(n) x(n-k), \tag{7.47}$$

$$\varepsilon_2(n) = \sum_{r_{2,1}=0}^{R_2-1} \sum_{k=0}^{L_{\boldsymbol{r}_2}-1} m_{\boldsymbol{r}_{2,k}}(n)\, x_{\boldsymbol{r}_2}(n-k). \tag{7.48}$$

For the considered zero-mean IID input $x(n)$, the mean squared residual echo of the linear kernel can then be expressed by

$$\mathrm{E}\left\{ \varepsilon_1^2(n) \right\} = \sum_{k=0}^{N_1-1} \mathrm{E}\left\{ m_{\boldsymbol{r}_{1,k}}^2(n) \right\} \mathrm{E}\left\{ x^2(n-k) \right\}. \tag{7.49}$$

As discussed in [20], in the echo cancellation context it is reasonable to apply a corresponding approximation of the mean squared residual echo of the quadratic kernel:

$$\mathrm{E}\left\{ \varepsilon_2^2(n) \right\} \approx \sum_{r_{2,1}=0}^{R_2-1} \sum_{k=0}^{L_{\boldsymbol{r}_2}-1} \mathrm{E}\left\{ m_{\boldsymbol{r}_{2,k}}^2(n) \right\} \mathrm{E}\left\{ x_{\boldsymbol{r}_2}^2(n-k) \right\}. \tag{7.50}$$

For a better understanding of the optimum step size, we follow [20] and introduce the kernel-independent auxiliary step-size factors

$$\alpha_{\mathrm{dt}}(n) = \frac{\mathrm{E}\left\{ \varepsilon^2(n) + b^2(n) \right\}}{\mathrm{E}\left\{ \varepsilon^2(n) + n^2(n) + s^2(n) \right\}}, \tag{7.51}$$

$$\alpha_{\mathrm{bn}}(n) = \frac{\mathrm{E}\left\{ \varepsilon^2(n) \right\}}{\mathrm{E}\left\{ \varepsilon^2(n) + b^2(n) \right\}}. \tag{7.52}$$

Note that for the definition of $\alpha_{\mathrm{dt}}(n)$ and $\alpha_{\mathrm{bn}}(n)$, the mutual statistical independence of the input signal, the background noise, and the near-end speech has been used. Additionally, we introduce the kernel-dependent auxiliary step-size factors

$$\alpha_{\varepsilon_p}(n) = \frac{\mathrm{E}\left\{ \varepsilon_p^2(n) \right\}}{\mathrm{E}\left\{ \varepsilon_1^2(n) \right\} + \mathrm{E}\left\{ \varepsilon_2^2(n) \right\}}, \quad p \in \{1, 2\}, \tag{7.53}$$

where the orthogonality property according to Eq. 7.46 has been exploited. Furthermore, we define coefficient-dependent step-size factors

$$\alpha_{\boldsymbol{r}_{1,k}}(n) = \frac{\mathrm{E}\left\{ m^2_{\boldsymbol{r}_{1,k}}(n) \right\}}{\sum\limits_{l=0}^{N_1-1} \mathrm{E}\left\{ m^2_{\boldsymbol{r}_{1,l}}(n) \right\} \mathrm{E}\left\{ x^2(n-l) \right\}} \tag{7.54}$$

for the adaptation of the linear kernel. The corresponding step sizes for the coefficients of the quadratic kernel are given by

$$\alpha_{\boldsymbol{r}_{2,k}}(n) = \frac{\mathrm{E}\left\{ m^2_{\boldsymbol{r}_{2,k}}(n) \right\}}{\sum\limits_{r_{2,1}=0}^{R_2-1} \sum\limits_{l=0}^{L_{\boldsymbol{r}_2}-1} \mathrm{E}\left\{ m^2_{\boldsymbol{r}_{2,l}}(n) \right\} \mathrm{E}\left\{ x^2_{\boldsymbol{r}_2}(n-l) \right\}}. \tag{7.55}$$

Note that the auxiliary step-size factors $\alpha_{\boldsymbol{r}_{1,k}}(n)$ and $\alpha_{\boldsymbol{r}_{2,k}}(n)$ are based on Eq. 7.49 and the approximation in Eq. 7.50, respectively.

With the above auxiliary step sizes, the optimum step size according to Eq. 7.42 can be approximated by a factorized version [20]:

$$\mu_{\mathrm{opt},\boldsymbol{r}_{p,k}}(n) \approx \alpha_{\mathrm{dt}}(n)\, \alpha_{\mathrm{bn}}(n)\, \alpha_{\varepsilon_p}(n)\, \alpha_{\boldsymbol{r}_{p,k}}(n), \quad p \in \{1, 2\}. \tag{7.56}$$

The influence of the different step-size parameters on the control of the adaptation is discussed next.

According to its definition $\alpha_{\mathrm{dt}}(n)$ accounts for double-talk (dt) situations, where $s(n) \neq 0$. In the echo cancellation context it is reasonable to implement $\alpha_{\mathrm{dt}}(n)$ as an on/off switch in combination with a double-talk detector [2], i.e., $\alpha_{\mathrm{dt}}(n) = 0$ if a near-end talker is active, in order to avoid divergence of the adaptive filter coefficients, and $\alpha_{\mathrm{dt}}(n) = 1$, otherwise.

The step-size factor $\alpha_{\mathrm{bn}}(n)$ controls the adaptation of the acoustic echo canceller with respect to the distortion introduced by background noise (bn) $b(n)$. Methods for the estimation of the product $\alpha_{\mathrm{dt}}(n)\alpha_{\mathrm{bn}}(n)$ have been thoroughly discussed in [26] and are not further considered here.

For an interpretation of $\alpha_{\varepsilon_p}(n)$ we note that the error introduced by a misadjusted linear kernel acts as an interference for the adaptation of the quadratic kernel, and vice versa. Hence, the step-size factors $\alpha_{\varepsilon_p}(n)$ can be regarded as an adaptation control with respect to interferences caused by the misadjusted Volterra kernels. As follows from Eq. 7.53, the computation of $\alpha_{\varepsilon_p}(n)$ requires knowledge of at least the ratio of the second-order moments of $\varepsilon_p(n)$ and $\varepsilon(n)$ which is in general not accessible. Therefore, a model for estimating the respective second-order moments is required. More precisely, we assume that the second-order moment of the residual echo of the linear kernel, i.e., $\varepsilon_1(n)$, is proportionate to the output of the adaptive linear kernel, i.e., $\hat{d}_1(n)$. Analogously, the second-order moment of $\varepsilon_2(n)$ is assumed to be proportionate to $\hat{d}_2(n)$:

$$\mathrm{E}\left\{\, \varepsilon_p^2(n) \,\right\} \approx \gamma_\varepsilon(n) \left[\delta_{\varepsilon_p} + \overline{\left| \hat{d}_p(n) \right|} \right], \quad p \in \{1, 2\}, \tag{7.57}$$

where $\overline{\left| \hat{d}_p(n) \right|}$ denotes a smoothed version of the magnitude of $\hat{d}_p(n)$. This estimation model can be regarded as the first term of a Taylor series expansion of the mean squared residual echoes with respect to the magnitude of the output of the corresponding kernel. The smoothing of the output is used to avoid significant variations of the estimates due to strongly varying amplitudes of the output signal. The offset term δ_{ε_p} can be used in Eq. 7.57 to manipulate the dependency of the kernel-dependent step size $\alpha_{\varepsilon_p}(n)$ on the corresponding kernel output $\hat{d}_p(n)$. Note that δ_{ε_p} is required especially in the beginning of the adaptation, where $\hat{d}_p(n) = 0$ if the Volterra coefficients were initialized with zero. The proportionality factor $\gamma_\varepsilon(n)$ represents the general convergence properties of the Volterra kernels, i.e., $\gamma_\varepsilon(n)$ decreases for a stable adaptation. However, the actual values $\gamma_\varepsilon(n)$ are not required explicitly, as the fraction appearing in the definition of $\alpha_{\varepsilon_p}(n)$ can be reduced correspondingly.

The coefficient-dependent step size $\alpha_{\boldsymbol{r}_{p,k}}(n)$ can be used to speed-up the adaptation of coefficients that cause large coefficient errors. However, the coefficient errors are not known and, therefore, we have to use models for estimating the respective second-order moments. A common assumption is that large coefficient magnitudes also cause large error magnitudes [36]. Consequently, we assume that the second-order moment of a certain coefficient error is proportionate to the magnitude of the corresponding adaptive coefficient:

$$\mathrm{E}\left\{ m_{\boldsymbol{r}_{p,k}}^2(n) \right\} \approx \gamma_{m,p}(n) \left[\beta_{m,p}(n) + \left| \hat{h}_{\boldsymbol{r}_{p,k}}(n) \right| \right], \quad p \in \{1, 2\}. \tag{7.58}$$

This estimation model can be considered as the first term of a Taylor series expansion of the mean squared coefficient error with respect to the magnitude of the corresponding coefficient of the adaptive Volterra filter. The time-variant proportionality factor $\gamma_{m,p}(n)$ reflects the reduction of the magnitude of the coefficient errors during the convergence of the adaptive filter. The actual value of $\gamma_{m,p}(n)$ does not have to be specified explicitly, as the fractions in Eq. 7.54 and Eq. 7.55 can be reduced respectively. The parameter $\beta_{m,p}(n)$ can be used to adjust the influence which the coefficients of the adaptive Volterra filter have on their associated LMS update term. Note that $\beta_{m,p}(n)$ should not equal zero in the beginning of the adaptation if all coefficients were initialized with zeroes. Otherwise, the coefficients remain at their initial values. For the computation of Eq. 7.54 and Eq. 7.55, we additionally replace the expectations with respect to the input by the corresponding instantaneous values.

It should be mentioned that there is a strong link between the coefficient-dependent step size presented above and the proportionate NLMS (PNLMS) for second-order Volterra filters proposed in [18]: If the parameters $\beta_{m,p}(n)$ are chosen according to

$$\beta_{m,1}(n) = \frac{\beta_{\mathrm{c},1}}{N_1} \sum_{k=0}^{N_1-1} \left| \hat{h}_{\boldsymbol{r}_{1,k}}(n) \right| , \tag{7.59}$$

$$\beta_{m,2}(n) = \frac{\beta_{\mathrm{c},2}}{N_{\mathrm{coeff},p}} \sum_{r_{2,1}=0}^{R_2-1} \sum_{k=0}^{L_{\boldsymbol{r}_2}-1} \left| \hat{h}_{\boldsymbol{r}_{2,k}}(n) \right| , \tag{7.60}$$

the PNLMS according to [18] results.

Simulations

To evaluate the performance of the step size control algorithm presented above, we present simulation results obtained for a second-order adaptive Volterra filter. In the experiment, the input has been wide-sense stationary coloured noise with a power spectral density (PSD) corresponding to the long-term PSD of speech. The nonlinear echo path has been modeled by a second-order Volterra filter in DCR, where the memory length of the linear kernel has been $N_1 = 320$. To account for the cascaded nature of nonlinear acoustic echo paths, the memory length of the quadratic kernel has been $N_2 = 64$, while its width is only $R_2 = 20$. The same region of support has also been chosen for the second-order adaptive Volterra filter. As double-talk detection algorithms are outside the scope of this chapter, we set $s(n) = 0$ in the following which implies $\alpha_{\mathrm{dt}} = 1$. An SNR of 30 dB has been preset with respect to $b(n)$ and $d(n)$. Since we are mainly interested in the improvements resulting from the kernel-dependent and the coefficient-dependent step size parameters, a fixed value $\alpha_{\mathrm{bn}} = 0.3$ has been used. The performance is measured using the *Echo Return Loss Enhancement* (ERLE) which is defined by

$$\mathrm{ERLE} = 10 \log_{10} \frac{\mathrm{E}\left\{d^2(n)\right\}}{\mathrm{E}\left\{\varepsilon^2(n)\right\}} \text{ [dB]}. \tag{7.61}$$

The level of nonlinear distortion has been preset such that the maximum achievable ERLE of a purely linear approach is limited to approximately 20 dB.

In Fig. 7.7, the ERLE graphs obtained for a second-order adaptive Volterra Filter (VF) applying two different realizations of its step size are compared to a purely linear approach. Here, we look at the case where the step size parameters $\alpha_{\varepsilon_p}(n)$ and $\alpha_{\boldsymbol{r}_{p,k}}(n)$ are estimated by means of the models Eq. 7.57 and Eq. 7.58. The constant parameter δ_{ε_p} of the model for the residual errors have been chosen to $\delta_{\varepsilon_p} = 0.001$ for both kernels. For the model of the coefficient errors, Eqs. 7.59 and 7.60 has been used, where $\beta_{\mathrm{c},1} = \beta_{\mathrm{c},2} = 1$. Note that this choice implies the practically realizable PNLMS algorithm for second-order Volterra filters [18]. Furthermore, Fig. 7.7 shows the ERLE-graph obtained for the kernel-independent NLMS with a fixed step size $\alpha_{\mathrm{NLMS}}(n) = 0.3$. Fig. 7.7 additionally depicts the ERLE of a linear adaptive filter which has been implemented analogously to the linear kernel of the adaptive Volterra filter with coefficient-dependent step-size control.

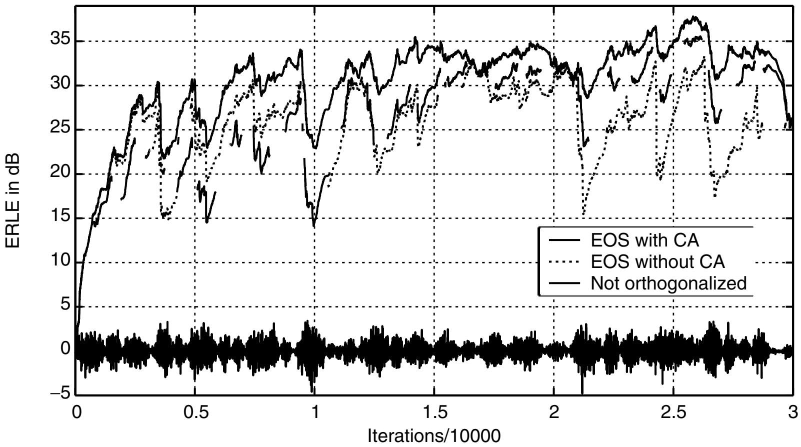


Fig. 7.7. ERLE obtained for a second-order adaptive Volterra filter (VF) applying the LMS algorithm with the proposed coefficient-dependent step size, and the kernel-independent NLMS, together with a linear approach for wide-sense stationary coloured noise input.

As can be seen, the second-order Volterra filter with coefficient-dependent step size clearly outperforms the corresponding Volterra filter applying a kernel-independent NLMS algorithm. We also notice from Fig. 7.7 that the achievable echo attenuation of the purely linear approach is limited due to the nonlinear distortion in the echo path.

7.3.3 Multidelay Adaptive Volterra Filters

DFT-domain approaches are very popular in linear adaptive filtering, as they allow for an increased convergence speed, while at the same time reducing the computational complexity [30, 38]. Corresponding DFT-domain methods which exploit fast convolution techniques via block processing are therefore desirable in adaptive Volterra filtering, too. For the derivation of such algorithms, we can basically distinguish between two different approaches for exploiting fast convolution methods. They are based on

- linear multidimensional filtering,
- linear multichannel filtering.

The first approach exploits the relation between linear multidimensional systems and Volterra filters in CCR [15, 22]. These methods are most efficient if the entire region of support of the Volterra kernels (i.e., $R_p = N_p$) has to be included. The second approach bases on the interpretation of Volterra filters in DCR as special linear MISO systems [19, 29], as already discussed above. The linear multidelay filter [30] applies partitioned block techniques which can be exploited to allow for different memory lengths for each kernel in the corresponding generalization to Volterra filters. This is especially attractive with acoustic echo cancellation for nonlinear loudspeaker systems, where it has been observed that the required memory length for the linear kernel is larger than that of the quadratic kernel [43]. Therefore, the restriction to a uniform DFT length for all Volterra kernels, as imposed in the DFT-domain approaches according to [15, 29], leads to inefficient system configurations, making the approaches [19, 22] more attractive for the acoustic echo cancellation application.

The DFT-domain algorithm presented in this section corresponds to [19] and represents an extension of the linear adaptive multidelay filter according to [30] to Volterra filters in DCR. An advantage of the resulting multidelay Volterra filter is that it preserves the flexibility with respect to choosing a desired region of support, as featured by the DCR.

Following the linear approach [30], a block-partitioned version of Eq. 7.13 is obtained by partitioning the linear filter $h_{\boldsymbol{r}_p,k}$ into $B_{\boldsymbol{r}_p}$ blocks of length N. In the following we assume that the memory lengths N_p of all Volterra kernels are integer multiples of the partition length N. From the definition of $L_{\boldsymbol{r}_p}$ in Eq. 7.10 we recall that the memory lengths of the filters in different channels are in general not uniform but can take on any value in the range $N_p - R_p < L_{\boldsymbol{r}_p} < N_p$. Consequently, the number of partitions $B_{\boldsymbol{r}_p}$ has to be chosen depending on the memory length $L_{\boldsymbol{r}_p}$ of the corresponding diagonal such that

$$\left(B_{\boldsymbol{r}_p} - 1\right) N < L_{\boldsymbol{r}_p} \leq B_{\boldsymbol{r}_p} N, \quad B_{\boldsymbol{r}_p}, N \in \mathbb{N}. \tag{7.62}$$

Aiming at a partitioned block version of the overlap/save method, we introduce zero-padded partitions of memory length $M = 2N$ according to

$$h_{\boldsymbol{r}_{p,b,l}} = \begin{cases} h_{\boldsymbol{r}_{p,bN+l}}, & 0 \leq l < N \;\wedge\; 0 \leq b < B_{\boldsymbol{r}_p}, \\ 0, & N \leq l < M \;\wedge\; 0 \leq b < B_{\boldsymbol{r}_p}. \end{cases} \tag{7.63}$$

It is important to note that in addition to the *explicit* zero-padding for $N \leq l < M$, the definition Eq. 7.63 also includes an *implicit* zero-padding in case of $L_{\boldsymbol{r}_p} < B_{\boldsymbol{r}_p} N$. For the last partition with index $b = B_{\boldsymbol{r}_p} - 1$, we additionally have to regard that

$$h_{\boldsymbol{r}_{p,bN+l}}\big|_{b=B_{\boldsymbol{r}_p}-1} = 0, \quad \text{for } l \geq L_{\boldsymbol{r}_p} - (B_{\boldsymbol{r}_p} - 1)N. \tag{7.64}$$

The partitioning and the zero-padding of the channel filters $h_{\boldsymbol{r}_{p,k}}$ according to Eq. 7.63 is illustrated in Fig. 7.8 for $B_{\boldsymbol{r}_p} = 6$, where $h_{\boldsymbol{r}_{p,b,l}}$ is shown for $b = 1$ and $b = 5$. Note that only the shaded areas contain nonzero coefficients. To exemplify the implicit zero-padding for $b = B_{\boldsymbol{r}_p} - 1$ according to Eq. 7.64, the memory length $L_{\boldsymbol{r}_p}$ has been chosen smaller than $B_{\boldsymbol{r}_p} N$.

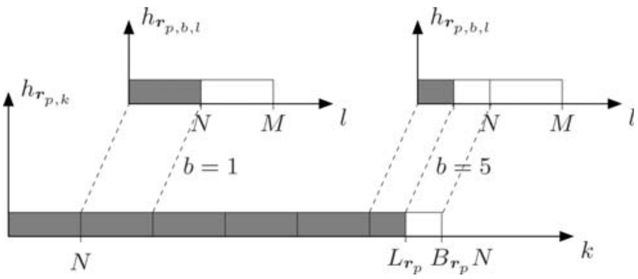


Fig. 7.8. Illustration of the partitioning according to Eq. 7.63 for $B_{\boldsymbol{r}_p} = 6$ and $L_{\boldsymbol{r}_p} \leq B_{\boldsymbol{r}_p} N$. The implicit zero-padding according to Eq. 7.64 occurs for $b = 5$.

The input signal of each partition $h_{\boldsymbol{r}_{p,b,l}}$ is defined as

$$x_{\boldsymbol{r}_p,b}(n) = x_{\boldsymbol{r}_p}(n - bN). \tag{7.65}$$

Introducing the definitions Eq. 7.63, Eq. 7.65 into Eq. 7.13 yields a block-partitioned version for computing the output of the channel $\boldsymbol{r}_p$:

$$d_{\boldsymbol{r}_p}(n) = \sum_{b=0}^{B_{\boldsymbol{r}_p}-1} h_{\boldsymbol{r}_{p,b,n}} * x_{\boldsymbol{r}_p,b}(n). \tag{7.66}$$

Furthermore, we introduce a block-time index ν and signal blocks of length $M = 2N$ for the input signals of each partition according to

$$x_{\boldsymbol{r}_p,b}(\nu, \kappa) = x_{\boldsymbol{r}_p,b}(\nu L + \kappa - N), \quad \text{for } 0 \leq \kappa < M, \tag{7.67}$$

in order to account for the block processing of the overlap/save method. Following [30], the block time shift L, representing the number of new samples of successive signal blocks, is defined using an overlap factor ρ, so that

$$L = \frac{N}{\rho}, \quad L \in \mathbb{N}. \tag{7.68}$$

Introducing the block time index ν into Eq. 7.66 yields

$$d_{\boldsymbol{r}_p}(\nu,\kappa) = \sum_{b=0}^{B_{\boldsymbol{r}_p}-1} h_{\boldsymbol{r}_{p,b,\kappa}} * x_{\boldsymbol{r}_p,b}(\nu,\kappa), \tag{7.69}$$

where the convolution is performed with respect to κ. The DFT-domain correspondence of Eq. 7.69 is given by

$$D_{\boldsymbol{r}_p}(\nu,m) = \sum_{b=0}^{B_{\boldsymbol{r}_p}-1} H_{\boldsymbol{r}_{p,b,m}}\, X_{\boldsymbol{r}_p,b}(\nu,m), \tag{7.70}$$

where $H_{\boldsymbol{r}_{p,b,m}}$ and $X_{\boldsymbol{r}_p,b}(\nu,m)$ denote the M-point DFT of $h_{\boldsymbol{r}_{p,b,l}}$ and $x_{\boldsymbol{r}_p,b}(\nu,\kappa)$, respectively. Analogously to the time domain, the computation of the DFT-domain output $D(\nu,m)$ of the Volterra filter is performed according to

$$D_p(\nu,m) = \sum_{r_{p,1}=0}^{R_p-1} \sum_{r_{p,2}=r_{p,1}}^{R_p-1} \cdots \sum_{r_{p,p-1}=r_{p,p-2}}^{R_p-1} D_{\boldsymbol{r}_p}(\nu,m), \tag{7.71}$$

$$D(\nu,m) = \sum_{p=1}^{P} D_p(\nu,m). \tag{7.72}$$

Let $\breve{d}(\nu,\kappa)$ represent the M-point inverse DFT of $D_p(\nu,m)$, i.e.,

$$\breve{d}(\nu,\kappa) = \mathcal{F}_M^{-1}\Big\{D(\nu,m)\Big\}, \tag{7.73}$$

where $\mathcal{F}_M^{-1}\{\cdot\}$ denotes the M-point inverse DFT. Taking the relation between circular and linear convolution into account [34], we notice that the first N elements of $\breve{d}(\nu,\kappa)$ are corrupted by time-domain aliasing, while the last N elements of $\breve{d}(\nu,\kappa)$ are equal to the desired output signal block of the P-th order Volterra filter at block time index ν. The output sequence $d(n)$ is finally obtained by applying the overlap/save method [34], i.e., by discarding the first N elements of $\breve{d}(\nu,\kappa)$ and setting

$$d(n) = \breve{d}(\nu, n - \nu L + N), \quad \nu L \leq n < \nu L + N\,, \tag{7.74}$$

for the last N elements of $\breve{d}(\nu,\kappa)$. Note that for an overlapping factor $\rho > 1$, only the last L elements represent new values of $d(n)$, whereas the remaining $N - L$ elements have already been computed in previous block time steps. However, choosing $\rho > 1$ is beneficial for the adaptive implementation of the Volterra filter, as then, the adaptation of the kernel coefficients is performed ρ times more frequently, resulting in an increased convergence speed of the adaptive algorithm.

It should be mentioned that for the special case that all kernels have the same memory length (i.e., $N_i = N_j$) and no partitioning is applied (i.e., $B_{\boldsymbol{r}_p} = 1$), the above algorithm reduces to the approach presented in [29].

The following discussion of the adaptation of the multidelay Volterra filter is based on the notation according to Fig. 7.6. The DFT-domain output of the adaptive Volterra filter $\hat{h}_{\boldsymbol{r}_{p,k}}(n)$ is computed analogously to Eqs. 7.70–7.72:

$$\widehat{D}_{\boldsymbol{r}_p}(\nu, m) = \sum_{b=0}^{B_{\boldsymbol{r}_p}-1} \widehat{H}_{\boldsymbol{r}_{p,b,m}}(\nu)\, X_{\boldsymbol{r}_p,b}(\nu, m), \tag{7.75}$$

$$\widehat{D}_p(\nu, m) = \sum_{r_{p,1}=0}^{R_p-1} \sum_{r_{p,2}=r_{p,1}}^{R_p-1} \cdots \sum_{r_{p,p-1}=r_{p,p-2}}^{R_p-1} \widehat{D}_{\boldsymbol{r}_p}(\nu, m), \tag{7.76}$$

$$\widehat{D}(\nu, m) = \sum_{p=1}^{P} \widehat{D}_p(\nu, m), \tag{7.77}$$

where $\widehat{H}_{\boldsymbol{r}_{p,b,m}}(\nu)$ denotes the DFT-domain correspondence of the adaptive Volterra filter coefficients $\hat{h}_{\boldsymbol{r}_{p,k}}(n)$. Furthermore, we define the DFT-domain error signal

$$\breve{E}(\nu, m) = D(\nu, m) - \widehat{D}(\nu, m). \tag{7.78}$$

As indicated by the symbol $\breve{}$, $\breve{E}(\nu, m)$ results from using the output of the Volterra filter based on the circular convolution according to Eq. 7.73, instead of the output based on linear convolution. Therefore, we introduce the windowed time-domain error signal according to [30]

$$e(\nu, \kappa) = \begin{cases} 0, & 0 \le \kappa < N, \\ \breve{e}(\nu, \kappa), & N \le \kappa < M, \end{cases} \tag{7.79}$$

where $\breve{e}(\nu, \kappa)$ denotes the M-point inverse DFT of $\breve{E}(\nu, m)$. The adaptation of the DFT-domain coefficients $\widehat{H}_{\boldsymbol{r}_{p,b,m}}(\nu)$ is then based on the DFT-domain correspondence of $e(\nu, \kappa)$, i.e.,

$$E(\nu, m) = \mathcal{F}_M\Big\{e(\nu, \kappa)\Big\}. \tag{7.80}$$

Regarding [19], the LMS-type update equation for the DFT-domain adaptive coefficients $\widehat{H}_{\boldsymbol{r}_{p,b,m}}(\nu)$ is given by

$$\widehat{H}_{\boldsymbol{r}_{p,b,m}}(\nu{+}1) = \widehat{H}_{\boldsymbol{r}_{p,b,m}}(\nu){+}\mu_p(\nu, m)\, \mathcal{F}_M\Big\{w_{\boldsymbol{r}_{p,b,l}}\ \mathcal{F}_M^{-1}\Big\{E(\nu, m) X^*_{\boldsymbol{r}_p,b}(\nu, m)\Big\}\Big\}. \tag{7.81}$$

The time-domain window function $w_{\boldsymbol{r}_{p,b,l}}$ is used to explicitly enforce the zero-padding of the time-domain partitions according to Eq. 7.63. For the definition of $w_{\boldsymbol{r}_{p,b,l}}$ we have to distinguish between different values of the partition index b. In case of $b < B_{\boldsymbol{r}_p} - 1$, the window function $w_{\boldsymbol{r}_{p,b,l}}$ is defined by

$$w_{\boldsymbol{r}_{p},b,l} = \begin{cases} 1, & 0 \leq l < N \ \wedge \ b < B_{\boldsymbol{r}_p} - 1, \\ 0, & N \leq l < M \ \wedge \ b < B_{\boldsymbol{r}_p} - 1, \end{cases} \tag{7.82}$$

whereas in case of $b = B_{\boldsymbol{r}_p} - 1$ we use the definition according to

$$w_{\boldsymbol{r}_{p},b,l} = \begin{cases} 1, & 0 \leq l < L_{\boldsymbol{r}_p} \ \wedge \ b = B_{\boldsymbol{r}_p} - 1, \\ 0, & L_{\boldsymbol{r}_p} \leq l < M \ \wedge \ b = B_{\boldsymbol{r}_p} - 1. \end{cases} \tag{7.83}$$

The differences in the definition of $w_{\boldsymbol{r}_{p},b,l}$ in dependence on the partition index b is due to the implicit zero-padding of $\hat{h}_{\boldsymbol{r}_{p},b,l}(\nu)$ resulting from Eq. 7.64. Due to the these time-domain constraints included in the update equation, Eq. 7.81 is commonly referred to as *constrained* adaptation algorithm.

For an implementation of the adaptive multidelay Volterra filter, the required distinction with respect to the definition of the time-domain window function $w_{\boldsymbol{r}_{p},b,l}$ is rather inconvenient. Therefore, it is beneficial to quantize the filter lengths $L_{\boldsymbol{r}_p}$ to integer multiples of the partition length N. In this case, $w_{\boldsymbol{r}_{p},b,l}$ can be replaced by the single window function

$$w_l = \begin{cases} 1, & 0 \leq l < N, \\ 0, & N \leq l < M, \end{cases} \tag{7.84}$$

which is then used for the adaptation of all partitions of each channel.

Step-size Normalization and Control for Second-Order Volterra Filters

A major advantage of linear DFT-domain adaptive filtering is the possibility to apply a frequency-dependent normalization of the step size [30, 38]. This approach is motivated by the approximate orthogonality property of the DFT, implying

$$\mathrm{E}\left\{ X^{*}_{\boldsymbol{r}_1,b}(\nu,i) X_{\boldsymbol{r}_1,b}(\nu,j) \right\} \approx 0, \quad \text{for } i \neq j, \tag{7.85}$$

if the DFT length M is sufficiently large [10]. In the following we assume that the corresponding orthogonality property also holds for the input of both, linear and nonlinear channels:

$$\mathrm{E}\left\{ X^{*}_{\boldsymbol{r}_p,a}(\nu,i) X_{\boldsymbol{s}_q,b}(\nu,j) \right\} \approx 0, \quad \text{for } i \neq j, \quad \forall\, \boldsymbol{r}_p,\, \boldsymbol{s}_q,\, a,\, b, \tag{7.86}$$

where $\boldsymbol{r}_p$ and $\boldsymbol{s}_q$ denote index vectors of certain diagonals of the Volterra filter and a and b denote a certain partition of these diagonals. It then directly follows from Eq. 7.86 that the normalization of the step size can be performed DFT bin-wise in case of adaptive multidelay Volterra filters, too. In the echo cancellation context it is reasonable to assume that the input signal of the linear kernel and the input signals of the diagonals of the quadratic kernel are orthogonal, implying

$$\mathrm{E}\left\{ X^{*}_{\boldsymbol{r}_1,a}(\nu,m) X_{\boldsymbol{r}_2,b}(\nu,m) \right\} = 0, \quad \forall\, a, b. \tag{7.87}$$

It can be shown that this orthogonality property is always fulfilled if the input signal $x(n)$ is a so-called spherically invariant random process (SIRP). Since SIRPs represent a realistic model for bandlimited speech [4], Eq. 7.87 in general holds in the echo cancellation application. Thus, the normalization of the step size $\mu_p(\nu, m)$ in Eq. 7.81 can be performed frequency- *and* kernel-dependently:

$$\mu_p(\nu, m) = \frac{\alpha_p(\nu, m)}{S_{X,p}(\nu, m)}, \quad p \in \{1, 2\} \tag{7.88}$$

Following the linear approach [39], the normalization factor $S_{X,1}(\nu, m)$ of the linear kernel partitions is computed recursively according

$$S_{X,1}(\nu, m) = \lambda S_{X,1}(\nu, m) + (1 - \lambda) \sum_{b=0}^{B_{\boldsymbol{r}_1}-1} \left| X_{\boldsymbol{r}_1,b}(\nu, m) \right|^2, \tag{7.89}$$

with the forgetting factor λ in the range $0 \leq \lambda < 1$. The normalization factor $S_{X,2}(\nu, m)$ used for the adaptation of the quadratic kernel is given by

$$S_{X,2}(\nu, m) = \lambda S_{X,2}(\nu, m) + (1 - \lambda) \sum_{r_{2,1}=0}^{R_2-1} \sum_{b=0}^{B_{\boldsymbol{r}_2}-1} \left| X_{\boldsymbol{r}_2,b}(\nu, m) \right|^2. \tag{7.90}$$

The step-size normalization according to Eq. 7.88 implies that $\alpha_p(\nu, m)$ has to control the adaptation with respect to the local distortions such as double-talk and background noise. Additionally, it has to take the interaction between different Volterra kernels into account, as a misadjusted linear kernel affects the adaptation of the quadratic kernel and vice versa. As in the time domain, we apply the factorization method for implementing $\alpha_p(\nu, m)$ and introduce

$$\alpha_p(\nu, m) = \alpha_{\mathrm{dt}}(\nu L)\ \alpha_{\mathrm{bn}}(\nu, m)\ \alpha_{\mathcal{E}_p}(\nu, m). \tag{7.91}$$

As in the time domain, step-size factor $\alpha_{\mathrm{dt}}(n)$ accounts for double-talk situations and has already been defined in Eq. 7.51. It should be mentioned that the double-talk detection used to implement $\alpha_{\mathrm{dt}}(\nu L)$ as an on/off switch, can also be performed in the DFT domain [2].

If $b(n)$ represents coloured noise, the level of distortion introduced by the background noise is in general different for each DFT bin. In this case it is advantageous to implement $\alpha_{\mathrm{bn}}(\nu, m)$ individually for each DFT bin. Regarding the derivation for linear DFT-domain adaptive filters in [31], the DFT-domain correspondence of Eq. 7.51 is given by

$$\alpha_{\mathrm{bn}}(\nu, m) = \frac{\mathrm{E}\left\{ \left| \mathcal{E}(\nu, m) \right|^2 \right\}}{\mathrm{E}\left\{ \left| \mathcal{E}(\nu, m) \right|^2 \right\} + \mathrm{E}\left\{ \left| B(\nu, m) \right|^2 \right\}}, \tag{7.92}$$

where $B(\nu, m)$ denotes the DFT-domain representation of the background noise $b(n)$ at block time index ν, and $\mathcal{E}(\nu, m)$ is the DFT-domain correspondence of the residual echo $\varepsilon(n)$. Methods for estimating the statistical terms required for computing $\alpha_{\mathrm{bn}}(\nu, m)$ are also presented in [31] and, thus, are not further considered here.

Analogously to its time-domain counterpart Eq. 7.53, the kernel-dependent auxiliary step size $\alpha_{\mathcal{E}_p}(\nu, m)$ is given by

$$\alpha_{\mathcal{E}_p}(\nu, m) = \frac{\mathrm{E}\left\{ \left|\mathcal{E}_p(\nu, m)\right|^2 \right\}}{\mathrm{E}\left\{ \left|\mathcal{E}_1(\nu, m)\right|^2 \right\} + \mathrm{E}\left\{ \left|\mathcal{E}_2(\nu, m)\right|^2 \right\}}, \quad p \in \{1, 2\}. \tag{7.93}$$

For estimating the mean squared magnitude of the DFT-domain residual echos $\mathcal{E}_p(\nu, m)$, it is possible to use the proportionality model according to Eq. 7.57 in its DFT-domain version:

$$\mathrm{E}\left\{ \left|\mathcal{E}_p(\nu, m)\right|^2 \right\} \approx \gamma_{\mathcal{E}}(\nu, m) \left[\delta_{\mathcal{E}_p} + \overline{\left|\widehat{D}_p(\nu, m)\right|} \right], \quad p \in \{1, 2\}, \tag{7.94}$$

where $\overline{|\widehat{D}_p(\nu, m)|}$ represents a smoothed version of the magnitude of $\widehat{D}_p(\nu, m)$. The meaning of the remaining parameters in Eq. 7.94 is equivalent to the corresponding parameters of the time-domain model Eq. 7.57. Simulation results indicate that in general the even simpler assumption of uniform mean squared residual echos, i.e,

$$\mathrm{E}\left\{ \left|\mathcal{E}_1(\nu, m)\right|^2 \right\} \approx \mathrm{E}\left\{ \left|\mathcal{E}_2(\nu, m)\right|^2 \right\} \tag{7.95}$$

can be used without loss in performance. Then, Eq. 7.94 reduces to a kernel-independent factor

$$\alpha_{\mathcal{E}_p}(\nu, m) \approx \frac{1}{2}, \quad p \in \{1, 2\}, \tag{7.96}$$

implying a kernel-independent step size $\alpha_p(\nu, m)$.

It should finally be mentioned that applying a coefficient-dependent DFT-domain step-size control does not further improve the performance of the adaptive multidelay Volterra filter. This results from the fact that the DFT already yields a sufficient decoupling of the adaptation of the kernel coefficients for different DFT bins.

Simulations

In the following we present simulation results in order to evaluate the performance of adaptive multidelay Volterra filters in the acoustic echo cancellation context. The nonlinear echo path has been modeled by a second-order Volterra filter in DCR with a memory length of $N_1 = 320$ taps for the linear kernel and a memory length of $N_2 = 64$ taps for the quadratic kernel. The width of the quadratic kernel has been set to $R_2 = 20$. The input signal has been wide-sense stationary coloured Gaussian noise. There has been no active near-end talker, i.e., $s(n) = 0$, and the variance of the additive white noise signal $b(n)$ has been chosen such that an SNR of 30 dB is obtained with respect to the variance of the echo signal $d(n)$.

The adaptive multidelay Volterra filter (MDVF) has been implemented with a partition length of $N = 64$ and an overlap factor $\rho = 4$. The number of partitions of the linear kernel has been $B_{\boldsymbol{r}_1} = 5$, implying $N_1 = 320$. In accordance with the echo path model, the memory length and the width of the quadratic kernel have been chosen to $N_2 = 64$ and $R_2 = 20$, respectively. Since the block length N matches the memory length of the quadratic kernel, no partitioning is applied to any diagonal of the quadratic kernel (implying $B_{\boldsymbol{r}_2} = 1$). The adaptation of the multidelay Volterra filter has been performed using the kernel-dependent normalization of the step size according to Eq. 7.88, where the fixed, kernel-independent value $\alpha_p(\nu, m) = 0.3$ has been used for both kernels.

Additionally, we consider a time-domain adaptive second-order Volterra filter that has the same region of support as the multidelay Volterra filter described above. The adaptation has been controlled applying the coefficient-dependent step size as already used for the simulations shown in Fig. 7.7.

The echo cancellation performance of the different approaches is illustrated in Fig. 7.9, where the ERLE obtained for a linear adaptive multidelay filter (MDF) that corresponds to the linear kernel of the adaptive multidelay Volterra filter is shown, too[2]. As can be clearly noticed, the convergence speed of the DFT-domain Volterra filter is significantly faster compared to the corresponding time-domain approach. This result shows the capability of DFT-domain methods to improve the convergence behaviour of adaptive Volterra filters. It can also be seen from Fig. 7.9 that the performance of the linear approach is clearly limited due to the nonlinear distortion in the echo path.

7.3.4 Application to Real Systems

Second-order Volterra filters have been introduced as a model for the nonlinear behaviour of loudspeakers. In the following we examine the suitability of this Volterra filter model when applied to real acoustic echo paths.

The experimental approach is divided into two parts, i.e., signal acquisition followed by simulations with recorded data: First, the echo signal is recorded in a room with low reverberation that also exhibits a low level of background noise. The actual experiments with respect to acoustic echo cancellation are then performed using the stored audio files of the input signal and the corresponding recording of the microphone signal. To provide an increased level

[2] Note that due to the definition of the ERLE in Eq. 7.61, the ERLE-values can be larger than the SNR with respect to the echo signal and the noise signal.

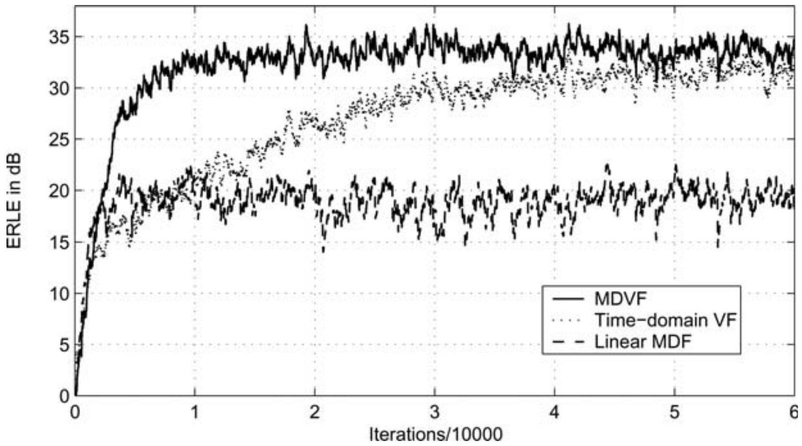


Fig. 7.9. ERLE obtained for a second-order adaptive multidelay Volterra filter (MDVF), a time-domain second-order adaptive Volterra filter, and a linear adaptive multidelay filter (MDF) for wide-sense stationary coloured noise input.

of background noise, an artificial noise signal is added to the recording of the microphone signal.

The loudspeaker used in the following experiments is a small electrodynamic loudspeaker with a diameter of 3.5 centimeters that is mounted in a closed box with a volume of about one liter. During the measurements it has been assured that the amplifier of the loudspeaker does not introduce significant nonlinear distortion. This allows for the desired isolated analysis of the nonlinear behaviour of the loudspeaker.

The coefficients of the linear and the quadratic kernels of the adaptive Volterra filters obtained from measurements of the considered loudspeaker with white Gaussian noise input are shown in Fig. 7.10. For illustrative reasons, we have used the Cartesian coordinate representation of the quadratic kernel, here. The zero-coefficients of the quadratic kernel corresponding to the indices $k_{2,2} < k_{2,1}$ result from the triangular representation of the Volterra filter according to Eq. 7.2. Note that the zero-valued coefficients for small values of $k_{2,1}$ correspond to the initial delay of the linear kernel. This initial delay results from the propagation of the echo signal on the direct path from the loudspeaker to the microphone. We further notice that the magnitudes of the quadratic kernel coefficients decay rapidly for increasing values of the coefficient indices $k_{2,1}$ and $k_{2,2}$. Nevertheless, the coefficients of the quadratic kernel have nonnegligible amplitudes within a large region in the $(k_{2,1}, k_{2,2})$-plane which confirms that the loudspeaker nonlinearities can not be considered as memoryless.

For the experimental results shown in Fig. 7.11, a speech signal sampled at 8 kHz has been used as input. The nonlinear echo canceller has been implemented as a second-order adaptive MDVF with memory lengths $N_1 = 320$ for

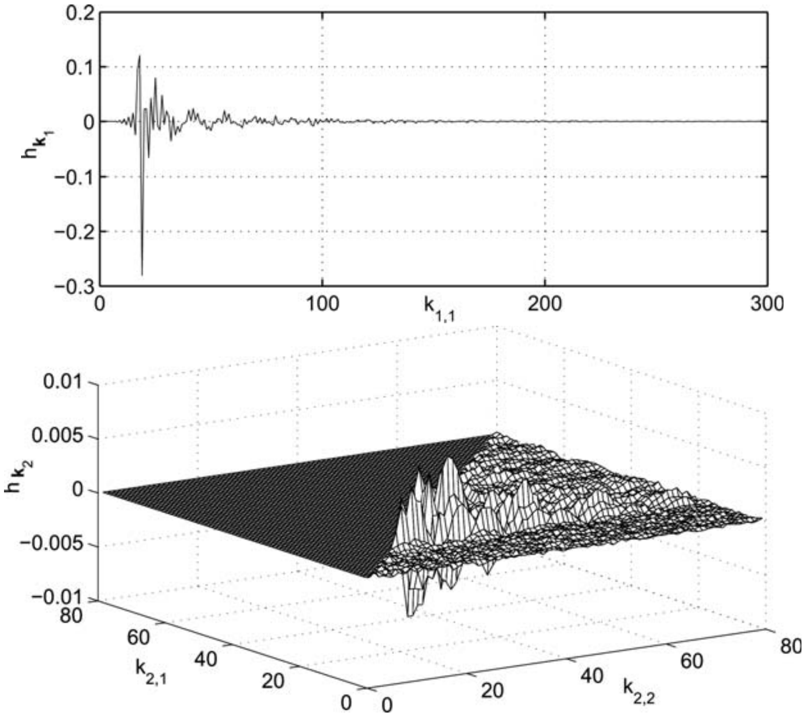


Fig. 7.10. Coefficients of the linear and quadratic Volterra kernels in Cartesian coordinate representation obtained from measurements in a room with low reverberation.

the linear kernel and $N_2 = 64$ for the quadratic kernel. The DFT length has been chosen to $M = 128$, implying that no partitioning has been applied to the quadratic kernel, whereas the linear kernel has been divided into $B_1 = 5$ partitions. To achieve faster convergence, an overlap factor of $\rho = 4$ has been used. The adaptation is performed according to Eq. 7.81, where the kernel-dependent normalization is applied. For both kernels, the normalized step size has been fixed to $\alpha_p(\nu, m) = 0.3$. For the simulation, a white Gaussian noise signal has been added to the recording of the microphone signal. The variance of the noise has been adjusted to provide an SNR of 30 dB relative to the microphone signal. The ERLE values obtained for the different kernel widths $R_2 = N_2 = 64$ and $R_2 = 15$ are shown in Fig. 7.11. For comparison, the result of a linear DFT-domain adaptive filter that corresponds to the linear kernel of the Volterra filters is given, too. As can be seen from Fig. 7.11, the performance of the purely linear approach is severely affected by the nonlinear distortion caused by the small-sized loudspeaker. We further notice that a remarkable increase of the echo attenuation can be achieved by both im-

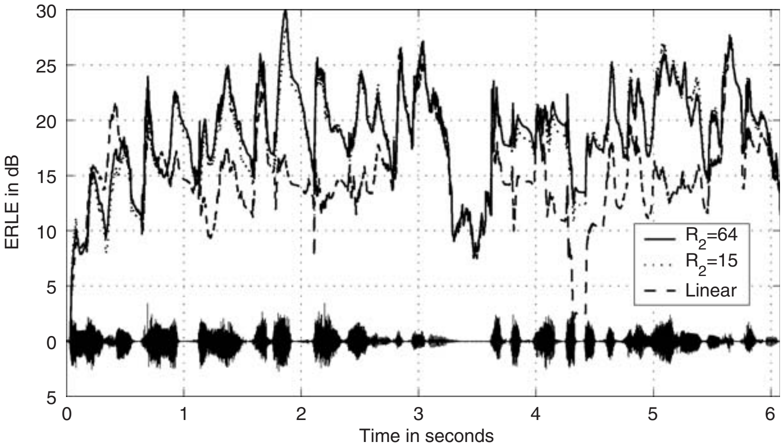


Fig. 7.11. ERLE obtained for second-order adaptive MDVFs with different width of the quadratic kernel together with the speech input. For comparison, the ERLE of a corresponding linear approach is shown, too.

plementations of the second-order Volterra filter. Especially during periods of high excitation levels, the adaptive MDVFs are able to improve the ERLE by 5 to 10 dB.

The DCR of Volterra filters has been motivated by its suitability to efficiently represent the cascaded structure that has been used as a simplified model for the acoustic echo path in case of nonlinearly distorting loudspeakers. The possibility to reduce the region of support of the adaptive Volterra filter without impairing the performance of the echo canceller is also shown in Fig. 7.11: The width of the quadratic kernel can be reduced to $R_2 = 15$ without any significant loss in achievable echo attenuation. The reduction of R_2 implies that the number of coefficients of the quadratic kernel is considerably decreased from $N_{\text{coeff},2} = 2080$ to $N_{\text{coeff},2} = 855$ in case of using only $R_2 = 15$ diagonals instead of $R_2 = 64$. Accordingly, the MDVF with reduced region of support increases the computational complexity compared to the linear approach only by a factor of four, whereas in case of $R_2 = 64$ a factor of thirteen results. Although not shown here, a further decrease of the width of the quadratic kernel yields a significant reduction of the achievable echo attenuation.

The experimental results according to Fig. 7.11 confirm the capability of second-order adaptive Volterra filters to cope with nonlinearly distorting loudspeakers. Furthermore, they illustrate that the advantageous structural features of the DCR allow for an efficient representation of the corresponding nonlinear acoustic echo path and according computational savings.

7.4 Power Filters

The nonlinear filters considered in this section are called power filters. They differ from general Volterra filters as they do not include nonlinear combinations of input samples taken at different time instances, while still representing a nonlinear system with memory. As shown later in this section, power filters represent a parallelized approximation of the echo path model according to Fig. 7.2 if the Volterra filter (SVF) is discarded. In other words, power filters represent a suitable approximation of the acoustic echo path if the nonlinear audio components can be considered as memoryless.

The block diagram shown in Fig. 7.12 illustrates the multichannel structure of a P-th order power filter. The input signal $x(n)$ is passed into P different channels. In the p-th channel, the input sample $x(n)$ is taken to the p-th power, and then passed through a linear filter $h_k^{(p)}$. The overall output $d(n)$ of the power filter is obtained by the summation over all channel outputs $d^{(p)}(n)$:

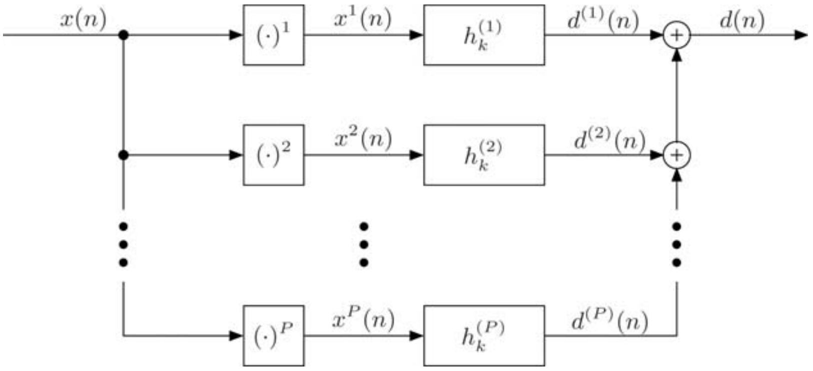


Fig. 7.12. Block diagram of a P-th order power filter.

$$d(n) = \sum_{p=1}^{P} d^{(p)}(n). \tag{7.97}$$

The output of the p-th channel results from the linear convolution of $x^p(n)$ with the filter coefficients $h_k^{(p)}$, i.e.,

$$d^{(p)}(n) = \sum_{k=0}^{N_p-1} h_k^{(p)}\, x^p(n-k). \tag{7.98}$$

Obviously, power filters can be interpreted as linear MISO systems, where the input of the p-th channel is given by the p-th power of $x(n)$. Comparing

Eq. 7.98 with Eq. 7.26, we notice that there is a strong relation between power filters and Volterra filters in diagonal coordinate representation: The p-th channel of a power filter corresponds to the main diagonal of a p-th order Volterra kernel in DCR. Thus, power filters can be considered as a special type of Volterra filters in DCR, where all kernels have width $R_p = 1$. Setting

$$h_k^{(p)} = h_{\boldsymbol{r}_{p,k}} \Big|_{\boldsymbol{r}_{p,k}=[k,k,\ldots,k]^{\mathrm{T}}} \tag{7.99}$$

shows the equivalency of Eq. 7.98 and Eq. 7.26 in case of $R_p = 1$.

For compactness, we rewrite Eq. 7.98 in vector notation:

$$d^{(p)}(n) = \boldsymbol{h}^{(p)^{\mathrm{T}}} \boldsymbol{x}^{(p)}(n), \tag{7.100}$$

where the input vector $\mathbf{x}^{(p)}(n)$ and the coefficient vector $\mathbf{h}^{(p)}$ are defined by

$$\boldsymbol{x}^{(p)}(n) = \left[x^p(n),\, x^p(n-1),\, \ldots,\, x^p(n-N_p+1)\right]^{\mathrm{T}}, \tag{7.101}$$

$$\boldsymbol{h}^{(p)} = \left[h_0^{(p)},\, h_1^{(p)},\, \ldots,\, h_{N_p-1}^{(p)}\right]^{\mathrm{T}}. \tag{7.102}$$

The DFT-domain implementation of power filters can in principle be obtained from Section 7.3.3 for the special case $R_p = 1$. For presentational convenience in upcoming sections, we assume in the following that no partitioning is applied to the channel filters $h_k^{(p)}$. The block length N is then chosen according to the maximum memory length of all channels, i.e.,

$$N = \max_p N_p. \tag{7.103}$$

The length of the DFT is $M = 2N$. The DFT-domain input vector

$$\boldsymbol{X}^{(p)}(\nu) = \left[X^{(p)}(\nu, 0),\, X^{(p)}(\nu, 1),\, \ldots,\, X^{(p)}(\nu, M-1)\right]^{\mathrm{T}} \tag{7.104}$$

corresponding to the time-domain input vector $\boldsymbol{x}_p(n)$ of the p-th channel is obtained from

$$\boldsymbol{X}^{(p)}(\nu) = \boldsymbol{F}_{M\times M} \left[x^p(\nu L - N),\, x^p(\nu L - N + 1),\, \ldots,\, x^p(\nu L + N - 1)\right]^{\mathrm{T}}, \tag{7.105}$$

where ν represents the block time index $n = \nu L$. In Eq. 7.105, $\boldsymbol{F}_{M\times M}$ is defined as the $M \times M$ DFT matrix which has elements of the form $e^{-j2\pi\kappa m/M}$. The block time shift $L = N/\rho$ has been introduced in Eq. 7.68. The DFT-domain coefficient vector corresponding to the p-th channel is given by

$$\boldsymbol{H}^{(p)} = \boldsymbol{F}_{M\times M} \left[\boldsymbol{h}^{(p)\mathrm{T}}\, \mathbf{0}^{\mathrm{T}}_{(M-N_p)\times 1}\right]^{\mathrm{T}}. \tag{7.106}$$

The DFT-domain representation for the output $\boldsymbol{D}^{(p)}(\nu)$ of the p-th channel is given by

$$\boldsymbol{D}^{(p)}(\nu) = \operatorname{diag}\left\{\boldsymbol{H}^{(p)}\right\} \boldsymbol{X}^{(p)}(\nu). \tag{7.107}$$

As in the time domain, the overall DFT-domain output vector $\boldsymbol{D}(\nu)$ is finally obtained by the summation over all channel outputs:

$$\boldsymbol{D}(\nu) = \sum_{p=1}^{P} \boldsymbol{D}^{(p)}(\nu). \tag{7.108}$$

The relation between the DFT-domain output vector $\boldsymbol{D}(\nu)$ (of length M) and the corresponding time-domain output block

$$\boldsymbol{d}(\nu) = [d(\nu L),\, d(\nu L+1),\, \ldots,\, d(\nu L+N-1)]^{\mathrm{T}} \tag{7.109}$$

of length N results from the overlap/save method and reads

$$\boldsymbol{d}(\nu) = \begin{bmatrix}\boldsymbol{0}_{N\times N} & \boldsymbol{I}_{N\times N}\end{bmatrix} \boldsymbol{F}_{M\times M}^{-1}\, \boldsymbol{D}(\nu). \tag{7.110}$$

Here, $\boldsymbol{0}_{N\times N}$ represents the $N \times N$ zero matrix and $\boldsymbol{I}_{N\times N}$ denotes the $N \times N$ identity matrix. Note that the matrix notation Eq. 7.110 corresponds to the element-wise notation of the overlap/save method according to Eqs. 7.73 and 7.74.

The discussion of power filters in the sequel is organized as follows: The application of adaptive power filters to nonlinear acoustic echo cancellation is motivated in Section 7.4.1 by showing that for certain applications the nonlinear echo path can be approximated by power filters. Orthogonalized versions of power filters in both, time domain and frequency domain are introduced in Section 7.4.2 in order to provide better performance of corresponding adaptive implementations. In Section 7.4.3, we apply adaptive orthogonalized power filters to real audio systems, including a nonlinear amplifier and the nonlinear loudspeaker of a mobile phone.

7.4.1 Application to Cascaded Structures

In the following we consider the cascaded structure shown in Fig. 7.13. It consists of the cascade of a linear filter w_k, a memoryless nonlinearity, and a second linear filter c_k. Comparing Fig. 7.13 with the model of the acoustic echo path according to Fig. 7.2, we notice that these two cascaded structures are equivalent if the Volterra filter (SVF) representing the nonlinear behaviour of the loudspeaker is discarded in Fig. 7.2. In practice, there are two cases, where Fig. 7.13 in fact models the nonlinear acoustic echo path well:

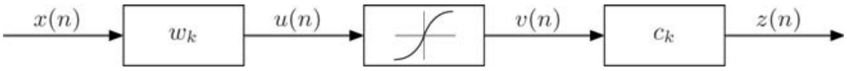


Fig. 7.13. Block diagram of the considered nonlinear cascaded structure.

- Loudspeakers can be regarded as almost linear if the required output sound level is well below the maximum output level. Then, the only source of nonlinear distortion is given by the amplifier, and the model of the acoustic echo path reduces to Fig. 7.13.
- The nonlinear behaviour of miniaturized loudspeakers operating close to the maximum level can be modeled sufficiently well by a memoryless saturation characteristic [23]. If both, the nonlinearity of the amplifier and the nonlinear characteristic of the loudspeaker are approximated by a truncated Taylor series expansion, their cascade can be modeled by a single Taylor series expansion, too. Again, the simplified model of the echo path according Fig. 7.13 results.

Note that the model according to Fig. 7.13 basically coincides with the model for the nonlinear echo path proposed in [32]. However, in [32] the authors use a continuously differentiable saturation characteristic based on a parametric function as an alternative model for the Taylor series expansion used here.

Using the notation given in Fig. 7.13, the output $v(n)$ of the memoryless nonlinearity yields

$$v(n) = \sum_{p=1}^{P} a_p \, u^p(n), \tag{7.111}$$

where a_p denote the coefficients of the truncated Taylor series expansion of the nonlinearity. The overall output $z(n)$ of the nonlinear cascade is then given by

$$z(n) = \sum_{p=1}^{P} \sum_{k=0}^{N_c-1} a_p c_k \, u^p(n-k), \tag{7.112}$$

where N_c denotes the filter length of c_k. Comparing Eq. 7.112 with Eq. 7.97 and Eq. 7.98 shows that $z(n)$ can be considered as the output of a P-th order power filter, having $u(n)$ as input. The coefficients of the linear filter associated to the p-th channel are obviously given by $c_k^{(p)} = a_p c_k$. Thus, we can rewrite Eq. 7.112 using the power filter model:

$$z(n) = \sum_{p=1}^{P} z^{(p)}(n), \tag{7.113}$$

$$z^{(p)}(n) = \sum_{k=0}^{N_c-1} c_k^{(p)} \, u^p(n-k). \tag{7.114}$$

Note that this interpretation of the computation of $z(n)$ corresponds to [23], where power filters are considered as parallelized implementation of the cascade of a memoryless nonlinearity and a linear filter.

Let us now consider the computation of the terms $u^p(n)$ which are required for computing $z^{(p)}(n)$. As $u(n)$ is the output of the linear filter w_k with memory length N_w, we obtain

$$u^p(n) = \sum_{k_1=0}^{N_w-1} \sum_{k_2=0}^{N_w-1} \cdots \sum_{k_p=0}^{N_w-1} \prod_{i=1}^{p} w_{k_i}\, x(n-k_i). \tag{7.115}$$

Due to the commutativity of the product terms in $\prod_{i=1}^{p} w_{k_i}\, x(n-k_i)$, we can rewrite Eq. 7.115 by changing the lower limits on its right hand side:

$$u^p(n) = \sum_{k_1=0}^{N_w-1} \sum_{k_2=k_1}^{N_w-1} \cdots \sum_{k_p=k_{p-1}}^{N_w-1} \Gamma(k_1, k_2, \ldots, k_p) \prod_{i=1}^{p} w_{k_i}\, x(n-k_i), \tag{7.116}$$

where $\Gamma(k_1, k_2, \ldots, k_p)$ denotes the number of possible distinct permutations of the indices $k_1, k_2, \ldots, k_p$. Comparing Eq. 7.116 with Eq. 7.2 we notice that $u^p(n)$ can be considered as the output of a specific p-th order Volterra kernel. The coefficients $w_{\mathbf{k}_p}$ of the corresponding Volterra kernel are obtained by equating $w_{\mathbf{k}_p} = \Gamma(k_1, k_2, \ldots, k_p) \prod_{i=1}^{p} w_{n_i}$. From the results in Section 7.3.1 it follows that the configuration according to Fig. 7.13 can exactly be represented by an appropriately chosen P-th order Volterra filter in DCR having memory length of $N_w + N_c - 1$ and width N_w. Note that the illustration in Fig. 7.5 can also serve as an example for the region of support of the corresponding quadratic kernel if we set $N_w = 4$ and $N_c = 16$. In general, such a Volterra model for acoustic echo paths is not practicable due to the enormously large required region of support of higher order kernels as can be noticed from Eq. 7.28. Thus, we look for an approximation of the equivalent Volterra filter by a corresponding power filter. For illustrative reasons, we decompose $u^p(n)$ into two parts:

$$u^p(n) = u^{(p)}(n) + u_{\mathrm{res},p}(n), \tag{7.117}$$

where, the first term on the right hand side of Eq. 7.117 is defined by

$$u^{(p)}(n) = \sum_{k=0}^{N_w-1} w_k^p\, x^p(n-k), \tag{7.118}$$

i.e., it results from linear filtering of $x^p(n)$ with the coefficients w_k^p. Note that $u^{(p)}(n)$ represents the output of the main diagonal of the p-th order Volterra kernel corresponding to Eq. 7.116. Discarding the residual term $u_{\mathrm{res},p}(n)$ in Eq. 7.117 yields an approximation for the computation of $z^{(p)}(n)$ according to

$$z^{(p)}(n) \approx \sum_{k=0}^{N_c-1} c_k^{(p)}\, u^{(p)}(n-k). \tag{7.119}$$

The approximation underlying Eq. 7.119 is illustrated in Fig. 7.14. As can be seen, the cascade of the linear filter w_k and a p-th order potentiator is replaced by the cascade of the potentiator followed by a linear filter with coefficients w_k^p. Using Fig. 7.5, this approximation can be illustrated for $p = 2$: All coefficients of the quadratic Volterra kernel are discarded expect for those lying on the main diagonal $r_{2,1} = 0$. Note that the approximation according to Fig. 7.14

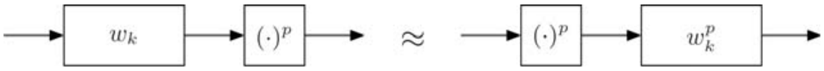


Fig. 7.14. Illustration of the approximation applied in Eq. 7.119.

represents an equality if the prefilter w_k is only a single delay. In this case, the model of the echo path according to Fig. 7.13 can be simplified to a corresponding cascade of a memoryless nonlinearity followed by a linear filter as proposed in [42]. Obviously, Eq. 7.119 *exactly* holds for the linear channel.

In case of the approximation Eq. 7.119, $z^{(p)}(n)$ can be interpreted as the output of the cascade of the linear filters $c_k^{(p)}$ and w_k^p, having $x^p(n)$ as input. If we finally introduce Eq. 7.118 in Eq. 7.119, we obtain the desired approximation of the nonlinear cascaded structure by a corresponding power filter:

$$z^{(p)}(n) \approx \sum_{k=0}^{N_g-1} g_k^{(p)}\, x^p(n-k), \tag{7.120}$$

where the coefficients $g_k^{(p)}$ of the power filter are given by

$$g_k^{(p)} = \sum_{l=0}^{N_c-1} c_l^{(p)}\, w_{k-l}^p. \tag{7.121}$$

The memory length of $g_k^{(p)}$ is $N_g = N_c + N_w - 1$.

The approximation of the nonlinear echo path model according to Fig. 7.13 using power filters can be regarded as a compromise between model accuracy and convergence behaviour of a corresponding adaptive implementation: The authors of [32] propose to realize the echo canceller by applying the same cascaded structure as used for the echo path model. However, it is challenging to assure convergence to the optimum solution or even assure stable adaptation behaviour for cascaded structures. This is especially true for the case that multiple linear filters are involved. The improvements with respect to convergence properties which should result from the inherent parallel nature of power filters are only achieved if a mutual orthogonalization of the channel inputs $x^p(n)$ is applied.

The approximation of Fig. 7.13 by power filters also represents a compromise between an exact model of the echo path and the approximation proposed in [42]. While [42] completely discards the prefilter, the power filter model includes part of the influence of w_k on the echo signal, as implied by Eq. 7.121. Experimental results indicate, however, that this increase in model accuracy does not improve the performance of corresponding adaptive implementations with respect to achievable echo attenuation.

Note that power filters realize the linear component of the echo path with only one single linear filter, whereas the approach [32] implicitly uses the cascade of two. This is an important property, since acoustic echo paths are usually only weakly nonlinear, i.e., the major contribution to the echo signal results from linear filtering of the input. Adaptive power filters therefore circumvent convergence problems that can not be excluded with adaptive structures that consist of the cascade of multiple linear filters.

7.4.2 Adaptive Orthogonalized Power Filters

The actual goal of the considerations presented in this section is the derivation of efficient adaptive implementations of power filters for their application to acoustic echo cancellation. The main obstacle to this is here that the input signals of the different channels of power filters, i.e., $x(n)$, $x^2(n)$, $\ldots$, $x^P(n)$ are not mutually orthogonal. Therefore, a direct adaptive implementation of the power filter structure according to Fig. 7.12 suffers from slow convergence as the adaptation of different channels interacts. To improve the performance of adaptive power filters, we discuss corresponding orthogonalized versions in the following.

Orthogonalization of the Input Signals

Following [23], we introduce a new set of mutually orthogonal input signals $x_{\mathrm{o}}^{(p)}(n)$ according to

$$x_{\mathrm{o}}^{(1)}(n) = x(n), \tag{7.122}$$

$$x_{\mathrm{o}}^{(p)}(n) = x^p(n) + \sum_{i=1}^{p-1} q_{p,i}\, x^i(n), \quad 1 < p \leq P. \tag{7.123}$$

The orthogonalization coefficients $q_{p,i}$ are chosen such that

$$\mathrm{E}\left\{ x_{\mathrm{o}}^{(i)}(n)\, x_{\mathrm{o}}^{(j)}(n) \right\} = 0, \quad \text{for } i \neq j. \tag{7.124}$$

A well-known approach for determining the orthogonalization coefficients $q_{p,i}$ is given by the Gram-Schmidt orthogonalization method [33]. The $p-1$ coefficients $q_{p,i}$ which are required for orthogonalizing the input of the p-th order channel can be obtained by solving

$$\begin{bmatrix} m_x^{(2)} & m_x^{(3)} & \ldots & m_x^{(p)} \\ m_x^{(3)} & m_x^{(4)} & \ldots & m_x^{(p+1)} \\ \vdots & \vdots & \ddots & \vdots \\ m_x^{(p)} & m_x^{(p+1)} & \ldots & m_x^{(2p-2)} \end{bmatrix} \begin{bmatrix} q_{p,1} \\ q_{p,2} \\ \vdots \\ q_{p,p-1} \end{bmatrix} = - \begin{bmatrix} m_x^{(p+1)} \\ m_x^{(p+2)} \\ \vdots \\ m_x^{(2p-1)} \end{bmatrix}, \tag{7.125}$$

where $m_x^{(k)}$ denotes the k-th order moment of $x(n)$, i.e.,

$$m_x^{(k)} = \mathrm{E}\left\{ x^k(n) \right\}. \tag{7.126}$$

If $x(n)$ is a stationary process and its statistics are *a priori* known, the orthogonalization coefficients $q_{p,i}$ are constant in time and can be calculated in advance. In practice, however, $m_x^{(k)}$ has to be replaced by corresponding time-variant estimates $\widehat{m}_x^{(k)}(n)$, especially if $x(n)$ is non-stationary. The estimation of $m_x^{(k)}$ can be performed, e.g., by applying the first order recursion

$$\widehat{m}_x^{(k)}(n) = \lambda \widehat{m}_x^{(k)}(n-1) + (1-\lambda) x^k(n). \tag{7.127}$$

The forgetting factor λ is in the range $0 \leq \lambda < 1$ and can be adjusted in order to adapt the estimation to the statistics of the input signal $x(n)$. Obviously, the orthogonalization coefficients $q_{p,i}(n)$ always depend on time if they are determined from Eq. 7.125 but based on time-variant estimates $\widehat{m}_x^{(k)}(n)$ of $m_x^{(k)}$.

For presentational convenience we assume in the following that all channels of the power filter have uniform memory length, i.e., $N_p = N$. The matrix representation for a block of N input samples corresponding to Eq. 7.122 and Eq. 7.123 reads

$$\boldsymbol{x}_{\mathrm{o}}^{(1)}(n) = \boldsymbol{x}^{(1)}(n), \tag{7.128}$$

$$\boldsymbol{x}_{\mathrm{o}}^{(p)}(n) = \boldsymbol{x}^{(p)}(n) + \sum_{i=1}^{p-1} \boldsymbol{Q}_{p,i}(n)\, \boldsymbol{x}^{(i)}(n), \quad 1 < p \leq P. \tag{7.129}$$

The orthogonalized signal vectors $\boldsymbol{x}_{\mathrm{o}}^{(p)}(n)$ are defined analogously to Eq. 7.101, i.e.,

$$\boldsymbol{x}_{\mathrm{o}}^{(p)}(n) = \left[x_{\mathrm{o}}^{(p)}(n),\, x_{\mathrm{o}}^{(p)}(n-1),\, \ldots,\, x_{\mathrm{o}}^{(p)}(n-N+1) \right]^{\mathrm{T}}. \tag{7.130}$$

In Eq. 7.129, $\boldsymbol{Q}_{p,i}(n)$ represents the diagonal orthogonalization matrix

$$\boldsymbol{Q}_{p,i}(n) = \mathrm{diag}\left\{ \left[q_{p,i}(n),\, q_{p,i}(n-1),\, \ldots,\, q_{p,i}(n-N+1) \right] \right\}. \tag{7.131}$$

The definition of $\boldsymbol{Q}_{p,i}(n)$ already includes the possible time-variance of its elements $q_{p,i}(n)$. Note that the orthogonalization Eq. 7.123 is performed in a sample-based manner: The coefficients $q_{p,i}(n)$ are determined such that the instantaneous orthogonality property Eq. 7.124 holds. The vectors $\boldsymbol{x}_{\mathrm{o}}^{(p)}(n)$, however, are in general mutually orthogonal, i.e.,

$$\mathrm{E}\left\{ \boldsymbol{x}_{\mathrm{o}}^{(i)}(n) \boldsymbol{x}_{\mathrm{o}}^{(j)\mathrm{T}}(n) \right\} = \boldsymbol{0}, \quad \text{for } i \neq j, \tag{7.132}$$

only if $x(n)$ is an IID random process. In case of correlated input, Eq. 7.132 is generally not satisfied. Nevertheless, we assume for the following that in practice Eq. 7.132 is met sufficiently well. For illustration we consider a zero-mean, first-order stationary Laplacian Markov process $x(n)$ with an autocorrelation function $\mathrm{E}\{\, x(n)x(n-k) \,\} = 0.9^{|k|}$. The corresponding normalized crosscorrelation function

$$c_{1,3}(k) = \frac{\mathrm{E}\{\, x(n)x^3(n-k) \,\}}{\sqrt{\mathrm{E}\{\, x^2(n) \,\}\,\mathrm{E}\{\, x^6(n) \,\}}} \tag{7.133}$$

between $x(n)$ and $x^3(n)$ is shown in Fig. 7.15 together with the normalized crosscorrelation function

$$c_{\mathrm{o},1,3}(k) = \frac{\mathrm{E}\left\{\, x_{\mathrm{o}}^{(1)}(n)x_{\mathrm{o}}^{(3)}(n-k) \,\right\}}{\sqrt{\mathrm{E}\left\{\, \left(x_{\mathrm{o}}^{(1)}(n)\right)^2 \,\right\}\mathrm{E}\left\{\, \left(x_{\mathrm{o}}^{(3)}(n)\right)^2 \,\right\}}} \tag{7.134}$$

between the orthogonalized signals $x_{\mathrm{o}}^{(1)}(n)$ and $x_{\mathrm{o}}^{(3)}(n)$. As indicated by Fig. 7.15, the orthogonality property Eq. 7.132 is valid for the considered example. Thus, it is also reasonable to assume that Eq. 7.132 is sufficiently satisfied for speech input, since long-term properties of speech are commonly modeled by a Laplacian process [4].

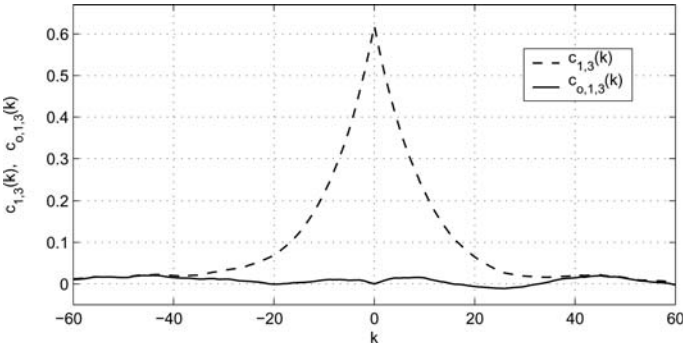


Fig. 7.15. Normalized crosscorrelation functions $c_{1,3}(k)$ and $c_{\mathrm{o},1,3}(k)$ between $x(n)$, $x^3(n)$ and $x_{\mathrm{o}}^{(1)}(n)$, $x_{\mathrm{o}}^{(3)}(n)$, respectively.

Note that, for correlated input, the orthogonalization according to Eq. 7.129 does not orthogonalize ('whiten') the samples within each input vector $\mathbf{x}_{\mathrm{o}}^{(p)}(n)$: Although the input vector of different channels are mutually orthogonal, in general

$$\mathrm{E}\left\{\, x_{\mathrm{o}}^{(p)}(n)x_{\mathrm{o}}^{(p)}(n-k) \,\right\} \neq 0, \quad 0 \leq k < N_p, \tag{7.135}$$

holds as an immediate consequence of the (auto-)correlation of $x(n)$. A *quasi-complete* orthogonalization can be achieved by considering the asymptotic orthogonalization property of the DFT for large transform lengths [10].

The DFT-domain correspondence of Eq. 7.128 and Eq. 7.129 yields

$$\boldsymbol{X}_{\mathrm{o}}^{(1)}(\nu) = \boldsymbol{X}^{(1)}(\nu), \tag{7.136}$$

$$\boldsymbol{X}_{\mathrm{o}}^{(p)}(\nu) = \boldsymbol{X}^{(p)}(\nu) + \sum_{i=1}^{p-1} \boldsymbol{\Phi}_{p,i}(\nu)\, \boldsymbol{X}^{(i)}(\nu), \quad 1 < p \leq P. \tag{7.137}$$

Regarding the definition of the DFT-domain input vectors $\boldsymbol{X}^{(p)}(\nu)$ in Eq. 7.105, the DFT-domain orthogonalization matrices $\boldsymbol{\Phi}_{p,i}(\nu)$ are given by

$$\boldsymbol{\Phi}_{p,i}(\nu) = \boldsymbol{F}_{M\times M}\, \mathrm{diag}\left\{ \Big[q_{p,i}(\nu L - N),\, \ldots,\, q_{p,i}(\nu L + N - 1) \Big] \right\} \boldsymbol{F}_{M\times M}^{-1}. \tag{7.138}$$

It is important to note that in contrast to the time-domain orthogonalization matrices $\boldsymbol{Q}_{p,i}(n)$, their DFT-domain counterparts $\boldsymbol{\Phi}_{p,i}(\nu)$ are in general not diagonal. With the DFT-domain vectors $\boldsymbol{X}_{\mathrm{o}}^{(p)}(\nu)$ we achieve a quasi-complete orthogonalization of power filters: On the one hand, the above discussion of Eq. 7.132 with respect to correlated input implies

$$\mathrm{E}\left\{ \boldsymbol{X}_{\mathrm{o}}^{(i)}(\nu)\boldsymbol{X}_{\mathrm{o}}^{(j)\mathrm{H}}(\nu) \right\} = \boldsymbol{0}, \quad \text{for } i \neq j. \tag{7.139}$$

On the other hand, the asymptotic orthogonalization property of the DFT additionally implies orthogonality of the DFT-domain input vector elements within each channel:

$$\mathrm{E}\left\{ X_{\mathrm{o}}^{(p)}(\nu,k) X_{\mathrm{o}}^{(p)^*}(\nu,m) \right\} \approx 0, \quad \text{for } k \neq m, \tag{7.140}$$

if the DFT length M is sufficiently large.

Equivalent Orthogonalized Structure

When using the orthogonalized channel inputs $x_{\mathrm{o}}^{(p)}(n)$ for computing the output of power filters, the coefficients of the corresponding orthogonalized versions have to be adjusted accordingly. In the following we show the relation between the coefficients of the original power filter and their orthogonalized counterparts. Furthermore, we discuss how a time-variant orthogonalization of the input affects the coefficients of the equivalent orthogonalized structure of power filters.

The output $d(n)$ of a P-th order power filter can be computed by using the orthogonalized input vectors $\boldsymbol{x}_{\mathrm{o}}^{(p)}(n)$, i.e.,

$$d(n) = \sum_{p=1}^{P} d_{\mathrm{o}}^{(p)}(n), \tag{7.141}$$

$$d_{\mathrm{o}}^{(p)}(n) = \boldsymbol{h}_{\mathrm{o}}^{(p)\mathrm{T}} \boldsymbol{x}_{\mathrm{o}}^{(p)}(n). \tag{7.142}$$

Obviously, $d(n)$ can equivalently be expressed by either using combinations of $\boldsymbol{h}^{(p)}$ and $\boldsymbol{x}^{(p)}(n)$, or using the corresponding pairs of vectors $\boldsymbol{h}_{\mathrm{o}}^{(p)}(n)$ and

$\boldsymbol{x}_{\mathrm{o}}^{(p)}(n)$. Following [23], we refer to the combination of the orthogonalized input vectors $\boldsymbol{x}_{\mathrm{o}}^{(p)}(n)$ and the corresponding filter coefficient vectors $\boldsymbol{h}_{\mathrm{o}}^{(p)}(n)$ as *equivalent orthogonalized structure (EOS)* of power filters. For determining the coefficients of the EOS, we notice that the right hand sides of Eq. 7.97 and Eq. 7.141 have to be equal, implying

$$\sum_{p=1}^{P} \boldsymbol{h}^{(p)\mathrm{T}} \boldsymbol{x}^{(p)}(n) = \sum_{p=1}^{P} \boldsymbol{h}_{\mathrm{o}}^{(p)\mathrm{T}} \boldsymbol{x}_{\mathrm{o}}^{(p)}(n). \tag{7.143}$$

Introducing the definition of the orthogonalized input vectors Eq. 7.128 and Eq. 7.129 into Eq. 7.143 and solving for $\boldsymbol{h}_{\mathrm{o}}^{(p)}(n)$ for each p (starting with $p = P$) leads to the relation between the original filter coefficients $\boldsymbol{h}^{(p)}$ and the coefficients of the corresponding EOS:

$$\boldsymbol{h}_{\mathrm{o}}^{(P)} = \boldsymbol{h}^{(P)}, \tag{7.144}$$

$$\boldsymbol{h}_{\mathrm{o}}^{(p)}(n) = \boldsymbol{h}^{(p)} - \sum_{i=p+1}^{P} \boldsymbol{Q}_{i,p}(n) \boldsymbol{h}_{\mathrm{o}}^{(i)}(n), \quad 1 \leq p < P. \tag{7.145}$$

We notice that due to the orthogonalization of the input vectors, all channels of order $i > p$ contribute to the p-th channel of the corresponding EOS. Note that Eq. 7.145 implies that for time-varying orthogonalization matrices $\boldsymbol{Q}_{i,p}(n)$ the coefficients of the EOS $\boldsymbol{h}_{\mathrm{o}}^{(p)}(n)$ will generally be time-variant, although the coefficients $\boldsymbol{h}^{(p)}$ may be constant in time.

For the discussion of the DFT-domain EOS of power filters we introduce the diagonal matrix

$$\boldsymbol{H}_{\mathrm{diag}}^{(p)} = \mathrm{diag}\left\{\boldsymbol{H}^{(p)}\right\}. \tag{7.146}$$

Then, Eq. 7.107 can be rewritten according to

$$\boldsymbol{D}^{(p)}(\nu) = \boldsymbol{H}_{\mathrm{diag}}^{(p)} \boldsymbol{X}^{(p)}(\nu). \tag{7.147}$$

Analogously to Eq. 7.108 and Eq. 7.147, the computation of the DFT-domain output vector $\boldsymbol{D}(\nu)$ can alternatively be expressed by using the orthogonalized input vectors $\boldsymbol{X}_{\mathrm{o}}^{(p)}(\nu)$:

$$\boldsymbol{D}(\nu) = \sum_{p=1}^{P} \boldsymbol{D}_{\mathrm{o}}^{(p)}(\nu), \tag{7.148}$$

$$\boldsymbol{D}_{\mathrm{o}}^{(p)}(\nu) = \boldsymbol{H}_{\mathrm{diag,o}}^{(p)} \boldsymbol{X}_{\mathrm{o}}^{(p)}(\nu). \tag{7.149}$$

The matrices $\boldsymbol{H}_{\mathrm{diag,o}}^{(p)}$ represent the DFT-domain EOS of the corresponding power filter. The relation between the coefficient matrices $\boldsymbol{H}_{\mathrm{diag,o}}^{(p)}$ of the EOS and the original coefficient matrices $\boldsymbol{H}_{\mathrm{diag}}^{(p)}$ is given by

$$\boldsymbol{H}^{(P)}_{\mathrm{diag,o}} = \boldsymbol{H}^{(P)}_{\mathrm{diag}}, \tag{7.150}$$

$$\boldsymbol{H}^{(p)}_{\mathrm{diag,o}}(\nu) = \boldsymbol{H}^{(p)}_{\mathrm{diag}} - \sum_{i=p+1}^{P} \boldsymbol{\Phi}_{i,p}(\nu)\boldsymbol{H}^{(i)}_{\mathrm{diag,o}}(\nu), \quad 1 \leq p < P. \tag{7.151}$$

In accordance to the time-domain EOS, the DFT-domain EOS has to be time-variant due to time-variant orthogonalization matrices $\boldsymbol{\Phi}_{i,p}(\nu)$.

Since the orthogonalization matrices $\boldsymbol{\Phi}_{i,p}(\nu)$ are in general not diagonal, it follows from Eq. 7.151 that this is also true for the DFT-domain coefficient matrices $\boldsymbol{H}^{(p)}_{\mathrm{diag,o}}(\nu)$. Thus, the DFT-domain EOS of a power filter requires a set of $M \times M$ coefficient matrices, although it can be completely described by the original $M \times 1$ DFT-domain coefficient vectors $\boldsymbol{H}^{(p)}$. Obviously, this DFT-domain EOS of power filters constitutes a very inefficient way to represent power filters. This problem can be circumvented by performing the orthogonalization of the DFT-domain input vectors in a 'block time'-based manner. Thereby, the orthogonalization coefficients $q_{i,p}(n)$ are updated only once per block time index ν. Note that this implies the assumption of a short time stationary input $x(n)$. The desired diagonal orthogonalization matrices are then obtained by modifying their definition Eq. 7.138 according to

$$\begin{aligned}\boldsymbol{\Phi}_{p,i}(\nu) &= \boldsymbol{F}_{M\times M}\,\mathrm{diag}\Big\{\Big[q_{p,i}(\nu L),\,\ldots,\,q_{p,i}(\nu L)\Big]\Big\}\,\boldsymbol{F}^{-1}_{M\times M} \\ &= q_{p,i}(\nu L)\,\boldsymbol{I}_{M\times M}.\end{aligned} \tag{7.152}$$

With these diagonal orthogonalization matrices, we can simplify Eq. 7.150 and Eq. 7.151 to a vector-based representation. The coefficient vectors of the DFT-domain EOS corresponding to the original vectors $\boldsymbol{H}^{(p)}$ are finally obtained as

$$\boldsymbol{H}^{(P)}_{\mathrm{o}} = \boldsymbol{H}^{(P)}, \tag{7.153}$$

$$\boldsymbol{H}^{(p)}_{\mathrm{o}}(\nu) = \boldsymbol{H}^{(p)} - \sum_{i=p+1}^{P} q_{i,p}(\nu L)\,\boldsymbol{H}^{(i)}_{\mathrm{o}}(\nu), \quad 1 \leq p < P. \tag{7.154}$$

Obviously, the introduction of the block time index for determining the orthogonalization matrices is not only suggested by the inherent block processing of DFT-domain approaches, but it also leads to a more efficient implementation. Therefore, we restrict ourselves to diagonal orthogonalization matrices $\boldsymbol{\Phi}_{p,i}(\nu)$ according to Eq. 7.152 throughout the rest of this chapter.

Accounting for the block processing of DFT-domain power filters, the estimation of the k-th order moments $m_x^{(k)}$ can be performed via block averaging, i.e.,

$$\widehat{m}_x^{(k)}(\nu L + l) = \frac{1}{M}\sum_{i=0}^{M-1} x^k(\nu L - N + i), \quad \text{for } 0 \leq l < L. \tag{7.155}$$

The estimates $\widehat{m}_x^{(k)}(\nu L + l)$ are then introduced into Eq. 7.125 for computing the orthogonalization coefficients $q_{i,p}(\nu L)$.

Let us now look at the inevitable adjustment of the filter coefficients of the EOS arising from the time-varying input orthogonalization. For the derivation of the required adjustment, we solve Eq. 7.145 for the time instant $n-1$ with respect to the original coefficient vector $\boldsymbol{h}^{(p)}$:

$$\boldsymbol{h}^{(p)} = \boldsymbol{h}_{\mathrm{o}}^{(p)}(n-1) + \sum_{i=p+1}^{P} \boldsymbol{Q}_{i,p}(n-1)\boldsymbol{h}_{\mathrm{o}}^{(i)}(n-1). \tag{7.156}$$

Let us now consider the changes in Eq. 7.156 that occur for the next time instant. Due to the time-variance of the EOS for time-variant orthogonalization matrices, Eq. 7.156 becomes

$$\boldsymbol{h}^{(p)} = \boldsymbol{h}_{\mathrm{o}}^{(p)}(n) + \sum_{i=p+1}^{P} \boldsymbol{Q}_{i,p}(n)\boldsymbol{h}_{\mathrm{o}}^{(i)}(n). \tag{7.157}$$

Assuming that the original coefficients of the power filter are constant in time, we can replace $\boldsymbol{h}^{(p)}$ in Eq. 7.156 by the right hand side of Eq. 7.157. Solving for $\boldsymbol{h}_{\mathrm{o}}^{(p)}(n)$ finally leads to the required coefficient adjustment: After each change of the orthogonalization matrices $\boldsymbol{Q}_{i,p}(n)$, the coefficients vectors $\boldsymbol{h}_{\mathrm{o}}^{(p)}(n)$ are recursively recomputed according to

$$\boldsymbol{h}_{\mathrm{o}}^{(p)}(n) = \boldsymbol{h}_{\mathrm{o}}^{(p)}(n-1) + \sum_{i=p+1}^{P} \left[\boldsymbol{Q}_{i,p}(n-1)\boldsymbol{h}_{\mathrm{o}}^{(i)}(n-1) - \boldsymbol{Q}_{i,p}(n)\boldsymbol{h}_{\mathrm{o}}^{(i)}(n)\right], \tag{7.158}$$

starting with $p = P-1$. From Eq. 7.144 we notice that no adjustment is required for the P-th order channel, i.e., $\boldsymbol{h}_{\mathrm{o}}^{(P)}(n) = \boldsymbol{h}_{\mathrm{o}}^{(P)}(n-1)$.

The necessity of this coefficient adjustment becomes obvious when regarding that each set of orthogonalization matrices $\boldsymbol{Q}_{i,p}(n)$ yields a corresponding EOS. This implies that after each change of the orthogonalization matrices both, a new set of input vectors and a new set of associated coefficient vectors have to be determined.

The above time-domain result can directly be used to obtain a corresponding adjustment for the DFT-domain EOS. With the definition of the simplified DFT-domain EOS according to Eq. 7.153 and Eq. 7.154, the DFT-domain counterpart to Eq. 7.158 is given by

$$\boldsymbol{H}_{\mathrm{o}}^{(p)}(\nu) = \boldsymbol{H}_{\mathrm{o}}^{(p)}(\nu-1) + \sum_{i=p+1}^{P} \left[q_{i,p}(\nu L - L)\boldsymbol{H}_{\mathrm{o}}^{(i)}(\nu-1) - q_{i,p}(\nu L)\boldsymbol{H}_{\mathrm{o}}^{(i)}(\nu)\right], \tag{7.159}$$

where we start with $p = P-1$. As in the time domain, no adjustment is required for the channel with the highest order, i.e., $\boldsymbol{H}_{\mathrm{o}}^{(P)}(\nu) = \boldsymbol{H}_{\mathrm{o}}^{(P)}(\nu-1)$.

Adaptation of Orthogonalized Power Filters

Since we consider the adaptation of *orthogonalized* power filters we express the output $\hat{d}(n)$ of the adaptive power filter according to Eq. 7.141 and Eq. 7.142:

$$\hat{d}(n) = \sum_{p=1}^{P} \hat{d}_{\mathrm{o}}^{(p)}(n), \tag{7.160}$$

$$\hat{d}_{\mathrm{o}}^{(p)}(n) = \hat{\boldsymbol{h}}_{\mathrm{o}}^{(p)\mathrm{T}}(n)\boldsymbol{x}_{\mathrm{o}}^{(p)}(n). \tag{7.161}$$

In the following we use the same notation for the signals as introduced in Fig. 7.6, i.e., the observed signal $y(n)$ is composed of the echo signal $d(n)$, background noise $b(n)$, and local speech $s(n)$. The error signal $e(n) = y(n) - \hat{d}(n)$ is the difference between the observed signal and the output of the adaptive power filter. The LMS update equation for the coefficients of the adaptive EOS is then given by

$$\hat{h}_{\mathrm{o},k}^{(p)}(n+1) = \hat{h}_{\mathrm{o},k}^{(p)}(n) + \mu_{\mathrm{o},k}^{(p)}(n)\, e(n) x_{\mathrm{o}}^{(p)}(n-k). \tag{7.162}$$

The control of the adaptation by appropriately choosing the step size $\mu_{\mathrm{o},k}^{(p)}(n)$ is discussed later in this section.

Note that the coefficient adjustment according to Eq. 7.158 is carried out first, and then the coefficients of the EOS are adapted subsequently by applying Eq. 7.162.

For deriving a DFT-domain adaptation of power filters we recall that they can be considered as linear multichannel system. This interpretation obviously also applies for the EOS of power filters. Thus, we can directly use the results of Section 7.3.3 that have been obtained for DFT-domain Volterra filters in DCR. Assuming diagonal orthogonalization matrices, the DFT-domain output $\widehat{D}(\nu, m)$ of the adaptive power filter is given by

$$\widehat{D}(\nu, m) = \sum_{p=1}^{P} \widehat{D}_{\mathrm{o}}^{(p)}(\nu, m), \tag{7.163}$$

$$\widehat{D}_{\mathrm{o}}^{(p)}(\nu, m) = \widehat{H}_{\mathrm{o}}^{(p)}(\nu, m) X_{\mathrm{o}}^{(p)}(\nu, m). \tag{7.164}$$

The adaptation algorithm for power filters immediately follows from the corresponding update equation Eq. 7.81 for multidelay Volterra filters:

$$\widehat{H}_{\mathrm{o}}^{(p)}(\nu{+}1, m) = \widehat{H}_{\mathrm{o}}^{(p)}(\nu, m) + \mu_{\mathrm{o}}^{(p)}(\nu, m)\, \mathcal{F}_M \Big\{ w_l\, \mathcal{F}_M^{-1} \Big\{ E(\nu, m) X_{\mathrm{o}}^{(p)^*}(\nu, m) \Big\} \Big\}. \tag{7.165}$$

The time-domain window function w_l is introduced to assure the zero-padding of the time-domain coefficient vectors according to Eq. 7.106 and has been defined in Eq. 7.84. Since we have assumed uniform memory length $N_p = N$ for all channels, the same window function can be applied for each order p.

Adaptation Control

First, we look at the control of the step size parameter $\mu_{\mathrm{o},n}^{(p)}(n)$ of the time-domain LMS algorithm according to Eq. 7.162 which corresponds to the coefficient-dependent step size for P-th order Volterra filters presented in Section 7.3.2. Due to the mutual orthogonality of all channel inputs of the EOS, the reasoning applied in Section 7.3.2 for second-order Volterra filters can correspondingly be applied for the derivation of an optimum coefficient-dependent step size for the adaptive EOS of P-th order power filters.

The coefficient error $m_{\mathrm{o},n}^{(p)}(k)$ with respect to the time-domain EOS of power filters is defined by

$$m_{\mathrm{o},n}^{(p)}(n) = h_{\mathrm{o},k}^{(p)}(n) - \hat{h}_{\mathrm{o},k}^{(p)}(n). \tag{7.166}$$

Analogously to Section 7.3.2, we use the mean squared error between the actual coefficient error and the corresponding LMS update term as optimality criterion for determining the optimum value of the step size $\mu_{\mathrm{o},k}^{(p)}(n)$, i.e.,

$$J_{\mu_{\mathrm{o},k}^{(p)}}(n) = \mathrm{E}\left\{ \left[m_{\mathrm{o},k}^{(p)}(n) - \mu_{\mathrm{o},k}^{(p)}(n) e(n) x_{\mathrm{o}}^{(p)}(n-k) \right]^2 \right\}. \tag{7.167}$$

As in Section 7.3.2, we assume that the input $x(n)$ is an IID random process with an even PDF, i.e., the orthogonality property Eq. 7.132 holds. We further assume that the adaptive coefficients $\hat{h}_{\mathrm{o},k}^{(p)}(n)$ are statistically independent of the input. Applying the same reasoning as in [20] for Volterra filters, it is straightforward to show that Eq. 7.42 correspondingly holds for orthogonalized power filters:

$$\mu_{\mathrm{opt,o},k}^{(p)}(n) = \frac{\mathrm{E}\left\{ \left[m_{\mathrm{o},k}^{(p)}(n) \right]^2 \right\}}{\mathrm{E}\left\{ \varepsilon^2(n) + b^2(n) + s^2(n) \right\}}. \tag{7.168}$$

Aiming at a factorized version of Eq. 7.168, we introduce the residual echo $\varepsilon_{\mathrm{o}}^{(p)}(n)$ of the p-th channel of the EOS according to

$$\varepsilon_{\mathrm{o}}^{(p)}(n) = d_{\mathrm{o}}^{(p)}(n) - \hat{d}_{\mathrm{o}}^{(p)}(n). \tag{7.169}$$

The overall residual echo $\varepsilon(n) = d(n) - \hat{d}(n)$ can then be written as

$$\varepsilon(n) = \sum_{p=1}^{P} \varepsilon_{\mathrm{o}}^{(p)}(n). \tag{7.170}$$

Since the orthogonality property Eq. 7.132 holds for the assumed input, we can express the mean square of the residual echo by

$$\mathrm{E}\left\{ \varepsilon^2(n) \right\} = \sum_{p=1}^{P} \mathrm{E}\left\{ \left[\varepsilon_{\mathrm{o}}^{(p)}(n) \right]^2 \right\}. \tag{7.171}$$

The expression Eq. 7.171 for computing the mean squared residual echo can be exploited to derive a factorized version of Eq. 7.168. The desired factorized version of Eq. 7.168 is obtained as

$$\mu_{\mathrm{opt,o},k}^{(p)}(n) = \alpha_{\mathrm{dt}}(n)\, \alpha_{\mathrm{bn}}(n)\, \alpha_{\varepsilon_{\mathrm{o}}}^{(p)}(n)\, \alpha_{\mathrm{o},k}^{(p)}(n). \tag{7.172}$$

The auxiliary step sizes $\alpha_{\mathrm{dt}}(n)$ and $\alpha_{\mathrm{bn}}(n)$ have been defined in Eq. 7.51 and Eq. 7.52, respectively, and account for double-talk and background noise. The channel-dependent step size parameter $\alpha_{\varepsilon_{\mathrm{o}}}^{(p)}(n)$ is used to control the adaptation with respect to mutual interferences caused by misadjusted channel filters. Analogously to Eq. 7.53, it is defined by

$$\alpha_{\varepsilon_{\mathrm{o}}}^{(p)}(n) = \frac{\mathrm{E}\left\{ \left[\varepsilon_{\mathrm{o}}^{(p)}(n)\right]^2 \right\}}{\sum\limits_{i=1}^{P} \mathrm{E}\left\{ \left[\varepsilon_{\mathrm{o}}^{(i)}(n)\right]^2 \right\}}. \tag{7.173}$$

The coefficient-dependent step size parameter $\alpha_{\mathrm{o},k}^{(p)}(n)$ is finally given by

$$\alpha_{\mathrm{o},k}^{(p)}(n) = \frac{\mathrm{E}\left\{ \left[m_{\mathrm{o},k}^{(p)}(n)\right]^2 \right\}}{\sum\limits_{l=0}^{N_p-1} \mathrm{E}\left\{ \left[m_{\mathrm{o},l}^{(p)}(n)\right]^2 \right\} \mathrm{E}\left\{ \left[x_{\mathrm{o}}^{(p)}(n-l)\right]^2 \right\}}, \tag{7.174}$$

which obviously corresponds to the coefficient-dependent step size for second-order Volterra filters according to Eqs. 7.54 and 7.55, respectively.

Analogously to Eq. 7.57, the second-order moments of the residual echoes $\varepsilon_{\mathrm{o}}^{(p)}(n)$ can be estimated using the model

$$\mathrm{E}\left\{ \left[\varepsilon_{\mathrm{o}}^{(p)}(n)\right]^2 \right\} \approx \gamma_{\varepsilon}(n) \left[\delta_{\varepsilon_p} + \overline{\left|\hat{d}_{\mathrm{o}}^{(p)}(n)\right|}\right]. \tag{7.175}$$

For realizing the coefficient-dependent step size $\alpha_{\mathrm{o},k}^{(p)}(n)$, we apply the proportionality model Eq. 7.58 for estimating the mean square of the coefficient errors, i.e.,

$$\mathrm{E}\left\{ \left[m_{\mathrm{o},k}^{(p)}(n)\right]^2 \right\} \approx \gamma_{m,p}(n) \left[\beta_{m,p}(n) + \left|\hat{h}_{\mathrm{o},k}^{(p)}(n)\right|\right]. \tag{7.176}$$

The meaning of the parameters appearing in the estimation models Eq. 7.175 and Eq. 7.176 have already been discussed in Section 7.3.2.

The derivation of a step-size control for the DFT-domain EOS of power filters follows the kernel-dependent approach for DFT-domain Volterra filters according to Section 7.3.3. Thus, we first introduce a channel-dependent normalization of $\mu_{\mathrm{o}}^{(p)}(\nu, m)$ according to

$$\mu_{\mathrm{o}}^{(p)}(\nu,m) = \frac{\alpha_{\mathrm{o}}^{(p)}(\nu,m)}{\widehat{S}_{\mathrm{o},X}^{(p)}(\nu,m)}. \tag{7.177}$$

The normalization factor $\widehat{S}_{\mathrm{o},X}^{(p)}(\nu,m)$ represents an estimate of

$$S_{\mathrm{o},X}^{(p)}(\nu,m) = \mathrm{E}\left\{ \left| X_{\mathrm{o}}^{(p)}(\nu,m) \right|^2 \right\}, \tag{7.178}$$

i.e., of the PSD of the input signal of the p-th channel of the EOS. $\hat{S}_{\mathrm{o},X}^{(p)}(\nu,m)$ can be obtained, e.g., analogously to Eq. 7.89.

The normalized step size $\alpha_{\mathrm{o}}^{(p)}(\nu,m)$ is implemented by using a corresponding factorized version according to

$$\alpha_{\mathrm{o}}^{(p)}(\nu,m) = \alpha_{\mathrm{dt}}(\nu L)\ \alpha_{\mathrm{bn}}(\nu,m)\ \alpha_{\mathcal{E}_{\mathrm{o}}}^{(p)}(\nu,m). \tag{7.179}$$

The auxiliary step-size parameters $\alpha_{\mathrm{dt}}(\nu L)$ and $\alpha_{\mathrm{bn}}(\nu,m)$ have already been discussed in Section 7.3.3 and are not further considered here.

The channel-dependent step size $\alpha_{\mathcal{E}_{\mathrm{o}}}^{(p)}(\nu,m)$ represents the DFT-domain correspondence of $\alpha_{\varepsilon_{\mathrm{o}}}^{(p)}(n)$. Regarding Eq. 7.139, it is obvious that the orthogonality property Eq. 7.171 also holds in the DFT-domain. The mean squared magnitude of the DFT-domain residual echo can then be written as

$$\mathrm{E}\left\{ |\mathcal{E}(\nu,m)|^2 \right\} = \sum_{i=1}^{P} \mathrm{E}\left\{ \left| \mathcal{E}_{\mathrm{o}}^{(i)}(\nu,m) \right|^2 \right\}. \tag{7.180}$$

Here, $\mathcal{E}_{\mathrm{o}}^{(p)}(\nu,m)$ represents the DFT-domain correspondence of the time-domain residual echo $\varepsilon_{\mathrm{o}}^{(p)}(n)$. Consequently, the channel-dependent step size $\alpha_{\mathcal{E}_{\mathrm{o}}}^{(p)}(\nu,m)$ can be defined correspondingly to Eq. 7.173:

$$\alpha_{\mathcal{E}_{\mathrm{o}}}^{(p)}(\nu,m) = \frac{\mathrm{E}\left\{ \left| \mathcal{E}_{\mathrm{o}}^{(p)}(\nu,m) \right|^2 \right\}}{\sum\limits_{i=1}^{P} \mathrm{E}\left\{ \left| \mathcal{E}_{\mathrm{o}}^{(i)}(\nu,m) \right|^2 \right\}}. \tag{7.181}$$

As in case of DFT-domain Volterra filters, the assumption of uniform mean squared magnitudes of the residual echoes for all p yields a good performance for DFT-domain adaptive power filter, too. Then, Eq. 7.181 simplifies to

$$\alpha_{\mathcal{E}_{\mathrm{o}}}^{(p)}(\nu,m) \approx \frac{1}{P}, \tag{7.182}$$

i.e., to its channel-independent form.

Simulations

The following simulations illustrate the effect of time-variant orthogonalization of adaptive power filters on their performance and show the necessity of an appropriate adjustment of the coefficients of the EOS.

For the simulations, the echo path has been modeled by the cascade of a third-order memoryless polynomial and a linear filter of length $N_c = 200$, i.e., it can exactly be represented by a third-order power filter. The input signal has been a zero-mean, uncorrelated, non-stationary Laplacian process [4]. A white noise signal $b(n)$ has been added to the echo signal $d(n)$, where the noise variance yields an SNR of 30 dB with respect to the variance of $d(n)$.

The echo canceller has been realized as a third-order time-domain power filter, where the memory length of each channel has also been chosen to $N_p = 200$. Fig. 7.16 shows the ERLE graphs obtained for the EOS of the power filter with coefficient adjustment (CA) according to Eq. 7.158, the EOS without CA, and the corresponding non-orthogonalized power filter. The orthogonalization of the input has been performed signal-adaptively, using the recursive estimation of the moments according to Eq. 7.127 with a forgetting factor $\lambda = 0.97$. The step-size control has been realized according to Eq. 7.172 with a fixed value of $\alpha_{\mathrm{dt}}(n)\alpha_{\mathrm{bn}}(n) = 0.3$. The models Eqs. 7.175, 7.176 have been applied to approximate the channel-dependent step size $\alpha_{\varepsilon_{\mathrm{o}}}^{(p)}(n)$ and the coefficient-dependent step size $\alpha_{\mathrm{o},k}^{(p)}(n)$, respectively. The model parameter δ_{ε_p} for estimating the mean squared residual echoes has been chosen to $\delta_{\varepsilon_p} = 0.001$ for all channels. The model for the coefficient-dependent auxiliary step sizes has been realized analogously to Eq. 7.59 and Eq. 7.60, where

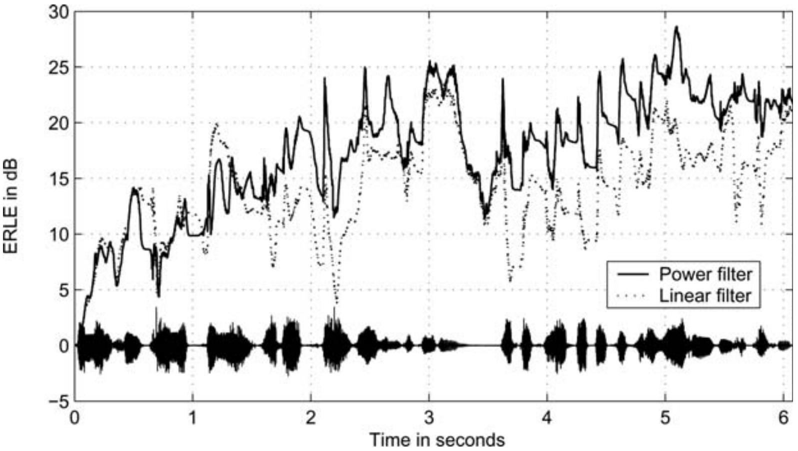


Fig. 7.16. ERLE obtained for different implementations of third-order adaptive power filters together with the uncorrelated, non-stationary input signal.

$\beta_{\mathrm{c},1} = \beta_{\mathrm{c},2} = 1$. To allow for a fair comparison of all approaches, the same adaptation control has been applied to all algorithms. It should, however be kept in mind that in case of non-orthogonalized power filters, this choice does not assure stable convergence, although it is advantageous with respect to convergence speed.

The limitation of achievable echo attenuation of the adaptive EOS without coefficient adjustment can clearly be seen in Fig. 7.16. This result confirms the importance of the coefficient adjustment required for time-variant orthogonalization matrices. We further notice from Fig. 7.16 that the EOS with coefficient adjustment outperforms the non-orthogonalized version in both, convergence speed and achievable echo attenuation.

7.4.3 Application to Real Systems

Power filters have been introduced as an approximation of the nonlinear cascaded model of the acoustic echo path according to Fig. 7.13. In this model, a Taylor series expansion has been used to approximate the nonlinear behaviour of the amplifier and the loudspeaker of a mobile phone, respectively. In this section we examine the suitability of these approximations when modeling real acoustic echo paths. Thereby, we look at the case of nonlinear distortion introduced by the amplifier and also consider the influence of the nonlinear behaviour of the loudspeaker of a mobile phone.

Nonlinear Amplifier

The experimental setup used for the following experiments consists of a commercial one-chip amplifier connected to an electro-dynamic loudspeaker with a diameter of six centimeters which has been placed in a room with low reverberation and a low background noise level. The power supply of the amplifier has been adjusted such that for high input levels the amplifier causes nonlinear distortion. Throughout the experiments, the nonlinear distortion introduced by the loudspeaker is negligible at the considered excitation levels.

The nonlinear echo canceller has been implemented as the adaptive EOS of a third-order power filter. The memory length of the linear kernel has been $N_1 = 256$ (implying a DFT length $M = 512$), whereas $N_2 = N_3 = 100$ has been chosen for the quadratic and the cubic channel. The orthogonalization of the channel inputs is performed block time-based, where the required moments of the input are estimated via block averaging according to Eq. 7.155. The DFT-domain EOS has been adapted applying using the fixed normalized step size $\alpha_{\mathrm{o},}^{(p)}(\nu, m) = 0.1$ for all channels. The input has been a speech signal sampled at 8 kHz. To simulate a higher level of background noise, a white Gaussian noise signal has been added to the recording of the real echo signal. The noise variance has been adjusted to give an SNR of 30 dB with respect to the measured microphone signal.

In Fig. 7.17, the ERLE obtained for the third-order adaptive EOS is compared to a linear approach which corresponds to the linear channel of the power filter. Except for the initial convergence phase, the adaptive power filter continuously provides an improvement of the echo cancellation performance, especially during periods of high excitation levels. Thereby, a gain of about 6 dB compared to the linear approach is well possible. Note that the applied third-order power filter increases the required number of multiplications only by a factor of 2.5 compared to the linear approach.

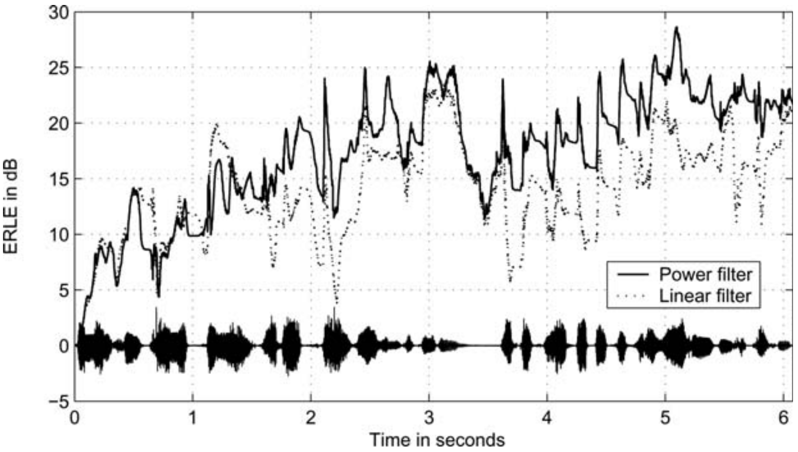


Fig. 7.17. ERLE obtained for the adaptive EOS of a third-order power filter and a linear approach together with the speech input.

Nonlinear Loudspeaker of a Mobile Phone

The nonlinear behaviour of a moderately-sized electro-dynamic loudspeaker has successfully been modeled by second-order Volterra filters. In this section we examine the very small electro-dynamic loudspeaker of a mobile phone. Due to its limited dimensions, the nonlinear behaviour of this type of loudspeakers is different from that used in Section 7.3.4 [23].

For the recordings, the loudspeaker has been mounted in the handset, while the microphone has been separated from it to avoid undesired vibration effects due to physical coupling of the loudspeaker and the microphone. During the measurements it has been assured that there is no nonlinear distortion introduced by overloading of the amplifier, i.e., the nonlinearity in the acoustic echo path is solely caused by the loudspeaker. The echo signal used for the simulations has been recorded in a room with low reverberation. The input has been a speech segment sampled at 8 kHz. A white Gaussian noise signal has been added to the recording of the microphone signal in order to simulate a background noise level corresponding to an SNR of 30 dB with respect to the acoustic echo. Since an algorithmic delay is not desirable in mobile phones, we consider the time-domain implementation of the EOS, where the memory length $N_1 = 250$ for the linear channel. Here, the memory lengths $N_2 = N_3 = 100$ for both, the quadratic and cubic channel are already sufficient. The orthogonalization of the channel inputs has been performed signal-adaptively, where the moments are estimated recursively according to Eq. 7.127 with a forgetting factor $\lambda = 0.97$.

In Fig. 7.18, the echo cancellation performance of the adaptive EOS of the third-order power filter is compared to a linear approach which corresponds to the linear channel of the power filter. As can be noticed, the performance of the linear adaptive filter is clearly inferior due to the nonlinear distortion introduced by the loudspeaker. The third-order power filter succeeds in improving the level of echo attenuation during almost the whole simulation period. Especially for speech segments that exhibit high excitation levels the increase of the ERLE is significant. Note that due to the short filters in the nonlinear channels, the computational complexity of the considered orthogonalized power filter is only two times higher than that of the linear filter.

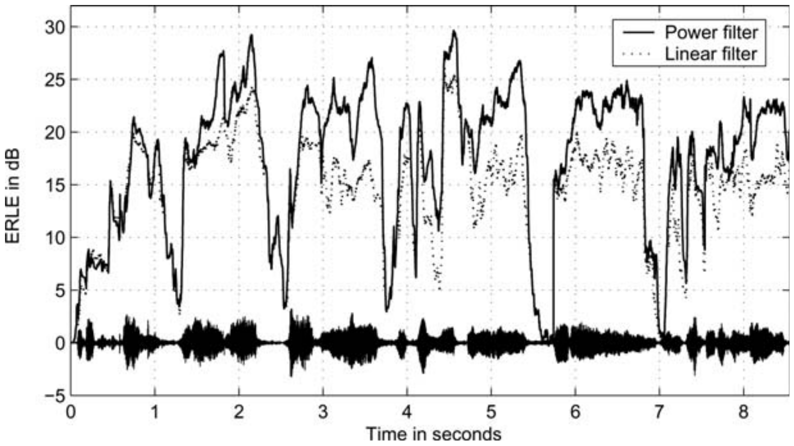


Fig. 7.18. ERLE obtained for the adaptive EOS of a third-order power filter and a corresponding linear approach for speech input.

7.5 Conclusions

In todays telecommunication devices often cheap audio hardware is included which introduces non-negligible nonlinear distortion into the loudspeaker signal. In case of hands-free telephone systems or mobile communication devices, these nonlinear audio components cause nonlinearly distorted acoustic echoes that can not be sufficiently attenuated by purely linear AECs. In this chapter, we have focused on special types of adaptive nonlinear filters which require only little *a priori* knowledge about the audio hardware actually included in the telecommunication device, i.e., Volterra filters and power filters.

If moderately-sized loudspeakers represent the only source of nonlinear distortion, second-order Volterra filters have been used to model the nonlinear acoustic echo path. Due to the assumption that only the loudspeaker introduces nonlinear distortion, the model of the acoustic echo path simplifies to a cascade of a second-order Volterra filter followed by a linear filter. It has been shown in Section 7.3 that the overall model of this cascade can be represented by a corresponding second-order Volterra filter that has a reduced region of support for the quadratic kernel. The DCR of Volterra filters allows for an elegant way to exploit this *a priori* knowledge about the acoustic echo path: The width of the quadratic kernel is simply chosen smaller than its memory length. By doing so, coefficients that are known (or assumed) to be zero can be explicitly excluded and inefficient system configurations can be avoided.

The DCR has also led to the interpretation of Volterra filters as a special type of linear multichannel systems. Based on that, efficient DFT-domain methods known from linear adaptive filtering could straightforwardly be extended to adaptive Volterra filters, too. The proposed MDVF does not affect the multichannel structure of the DCR and, therefore, preserves its advantageous features.

Experimental results obtained for a real loudspeaker system have been presented in order to verify the suitability of adaptive Volterra filters. In a realistic acoustic echo cancellation scenario, the echo attenuation has been improved by about 5 up to 10 dB compared to a linear approach. Due to the reduced width of the quadratic kernel, the computational complexity has only been increased by a factor of approximately four compared to the linear approach. Thus, second-order Volterra filters can be considered as a well suited approach to cope with nonlinear loudspeakers in hands-free telecommunication systems.

In Section 7.4 we have considered the case that only the amplifier or the miniaturized loudspeaker of a mobile phone cause the nonlinear distortion in the echo path. For this scenario, the model of the acoustic echo path simplifies to the cascade of a linear filter, a memoryless nonlinearity (modeled by a Taylor series expansion), and a second linear filter. It has been shown that power filters represent an efficient parallelized approximation of this overall model of the echo path. Since the saturation characteristics imply power filters of orders higher than two, the input signals of different channels are not mutually orthogonal anymore. In order to improve the performance of corresponding adaptive implementations, a method to signal-adaptively orthogonalize the inputs of the different channels of the power filter has been discussed.

Experimental results based on measurements with an overloaded amplifier have shown that the considered adaptive third-order power filter has been able to improve the echo attenuation of a purely linear AEC by about 6 dB. For the case that the loudspeaker of a mobile phone causes the nonlinear distortion in the echo path, third-order power filters are able to increase the achievable echo attenuation of a linear approach by approximately 5 dB.

Although the proposed nonlinear approaches provide significant improvements over purely linear adaptive filters, the achieved level of echo attenuation might not be sufficient in some applications. A common method in linear echo cancellation is to further suppress the residual echo that remains after the echo cancellation step. Usually, such methods apply postfiltering of the residual echo based on Wiener filtering techniques [12,13] which, however, have to be appropriately extended to account for nonlinear acoustic echo paths.

References

[1] O. Agazzi, D. G. Messerschmitt, D. A. Hodges: Nonlinear echo cancellation of data signals, *IEEE Trans. on Communications,* **30**(11), 2421–2433, Nov. 1982.

[2] J. Benesty, D. R. Morgan, J. H. Cho: A new class of doubletalk detectors based on cross-correlation, *IEEE Trans. on Acoustics, Speech, and Signal Processing,* **8**(2), 168–172, March 2000.

[3] A. N. Birkett, R. A. Goubran: Limitations of handsfree acoustic echo cancellers due to nonlinear loudspeaker distortion and enclosure vibration effects, *Proc. WASPAA '95,* 13–16, New Paltz, NY, USA, Oct. 1995.

[4] H. Brehm, W. Stammler: Description and generation of spherically invariant speech-model signals, *Signal Processing,* **12**(2), 119–141, March 1987.

[5] C. Breining, P. Dreiseitel, E. Hänsler, A. Mader, B. Nitsch, H. Puder, T. Schertler, G. Schmidt, J. Tilp: Acoustic echo control, *IEEE Signal Processing Magazine,* **16**(4), 42–69, July 1999.

[6] J. Chen, J. Vandevalle: Study of adaptive nonlinear echo canceller with Volterra expansion, *Proc. Processing ICASSP '89,* 1376–1379, Glasgow, UK, May 1989.

[7] A. Fermo, A. Carini, G. L. Sicuranza: Simplified Volterra filters for acoustic echo cancellation in GSM receivers, *Proc. EUSIPCO '00,* 2413–2416, Tampere, Finland, Sept. 2000.

[8] W. A. Frank: An efficient approximation to the quadratic Volterra filter and its application in realtime loudspeaker linearization, *Signal Processing,* **45**(1), 97–113, July 1995.

[9] G. O. Glentis, K. Berberidis, S. Theodoridis: Efficient least squares adaptive algorithms for FIR transversal filtering: a unified view, *IEEE Signal Processing Magazine,* **16**(4), 13–41, July 1999.

[10] R. M. Gray: On the asymptotic eigenvalue distribution of Toeplitz matrices, *IEEE Trans. on Information Theory,* **18**(6), 725–730, Nov. 1972.

[11] A. Guérin, G. Faucon, R. Le Bouquin-Jeannès: Nonlinear acoustic echo cancellation based on Volterra filters, *IEEE Trans. on Acoustics, Speech, and Signal Processing,* **11**(6), 672–683, Nov. 2003.

[12] S. Gustafsson, R. Martin, P. Vary: Combined acoustic echo control and noise reduction for hands-free telephony, *Signal Processing,* **64**(1), 21–32, 1998.

[13] E. Hänsler, G. Schmidt: *Acoustic Echo and Noise Control: A Practical Approach,* Hoboken, NJ, USA: Wiley, 2004.

[14] S. Haykin: *Adaptive Filter Theory,* Englewood Cliffs, NJ, USA: Prentice Hall, 1996.

[15] S. Im, E. J. Powers: A fast method of discrete third-order Volterra filtering, *IEEE Trans. on Signal Processing,* **44**(9), 2195–2208, Sept. 1996.

[16] W. Klippel: Dynamic measurement and interpretation of the nonlinear parameters of electrodynamic loudspeakers, *J. Audio Eng. Soc.,* **38**(12), 944–955, Dec. 1990.

[17] W. Klippel: Filter structures to compensate for nonlinear distortion of horn loudspeakers, *J. Audio Eng. Soc.,* Preprint 4102, 1995.

[18] F. Kuech, W. Kellermann: Proportionate NLMS algorithm for second-order Volterra filters and its application to nonlinear echo cancellation, *Proc. IWAENC '03,* 75–78, Kyoto, Japan, Sept. 2003.

[19] F. Kuech, W. Kellermann: A novel multidelay adaptive algorithm for Volterra filters in diagonal coordinate representation, *Proc. ICASSP '04,* **2**, 869-872, Montreal, Canada, May 2004.

[20] F. Kuech, W. Kellermann: Coefficient-dependent step-size for adaptive second-order Volterra filters, *Proc. EUSIPCO '04,* 1805–1808, Vienna, Austria, Sept. 2004.

[21] F. Kuech, W. Kellermann: Orthogonalized power filters for nonlinear acoustic echo cancellation, *Signal Processing,* submitted Jan. 2005.

[22] F. Kuech, W. Kellermann: Partitioned block frequency-domain adaptive second-order Volterra filter, *IEEE Trans. on Signal Processing,* **53**(2), 564–575, Feb. 2005.

[23] F. Kuech, A. Mitnacht, W. Kellermann: Nonlinear acoustic echo cancellation using adaptive orthogonalized power filters, *Proc. ICASSP '05,* **3**, 105–108, Philadelphia, PA, USA, March 2005.

[24] G. Lazzarin, S. Pupolin, A. Sarti: Nonlinearity compensation in digital radio systems, *IEEE Trans. on Communications,* **42**(2), 988–999, Feb. 1994.

[25] J. Lee, V. J. Mathews: A fast recursive least squares adaptive second order Volterra filter and its performance analysis, *IEEE Trans. on Signal Processing,* **41**(3), 1087–1102, March 1993.

[26] A. Mader, H. Puder, G. U. Schmidt: Step-size control for acoustic echo cancellation filters - an overview, *Signal Processing,* **80**(9), 1697–1719, Sept. 2000.

[27] V. J. Mathews: Adaptive polynomial filters, *IEEE Signal Processing Magazine,* **8**(3), 10–26, July 1991.

[28] V. J. Mathews, G. L. Sicuranza: *Polynomial Signal Processing,* New York, USA: John Wiley and Sons, 2000.

[29] M. Morháč: A fast algorithm of nonlinear Volterra filtering, *IEEE Trans. on Signal Processing,* **39**(10), 2353–2356, Oct. 1991.

[30] E. Moulines, O. A. Amrane, Y. Grenier: The generalized multidelay adaptive filter: Structure and convergence analysis,*IEEE Trans. on Signal Processing,* **43**(1), 14–28, Jan. 1995.

[31] B. H. Nitsch: A frequency-selective stepfactor control for an adaptive filter algorithm working in the frequency domain, *Signal Processing,* **80**(9), 1733–1745, Sept. 2000.

[32] B. S. Nollett, D. L. Jones: Nonlinear echo cancellation for hands-free speakerphones, *Proc. NSIP '97,* Mackinac Island, MI, USA, Sept. 1997.

[33] A. Papoulis, S. U. Pillai: *Probability, Random Variables and Stochastic Processes,* New York, USA: McGraw-Hill, 2002.

[34] J. G. Proakis, D. G. Manolakis: *Digital Signal Processing: Principles, Algorithms and Applications,* Englewood Cliffs, NJ, USA: Prentice Hall, 1996.

[35] G. M. Raz, B. D. Van Veen: Baseband Volterra filters for implementing carrier based nonlinearities, *IEEE Trans. on Signal Processing,* **46**(1), 103–114, Jan. 1998.

[36] M. Rupp, J. Cezanne: Robustness conditions of the LMS algorithm with time-variant matrix step-size, *Signal Processing,* **80**(9), 1787–1794, Sept. 2000.

[37] H. Schurer: *Linearization of Electroacoustic Transducers,* Enschede, Netherlands: Print Partners Ipskamp, 1997.

[38] J. J. Shynk: Frequency domain and multirate adaptive filtering, *IEEE Signal Processing Magazine,* **9**(1), 14–37, Jan. 1992.

[39] J.-S. Soo, K. K. Pang: Multidelay block frequency domain adaptive filter, *IEEE Trans. on Acoustics, Speech, and Signal Processing,* **38**(2), 373–376, Feb. 1990.

[40] M. Soria-Rodriguez, M. Gabbouj, N. Zacharov, M. S. Hämäläinen, K. Koivuniemi: Modeling and real-time auralization of electrodynamic loudspeaker non-linearities, *Proc. ICASSP '04,* **4**, 17–21, Montreal, Canada, May 2004.

[41] A. Stenger:*Kompensation akustischer Echos unter Einfluss von nichtlinearen Audiokomponenten,* Aachen, Germany: Shaker, 2001 (in German).

[42] A. Stenger, W. Kellermann: Adaptation of a memoryless preprocessor for nonlinear acoustic echo cancelling, *Signal Processing,* **80**(9), 1741–1760, Sept. 2000.

[43] A. Stenger, R. Rabenstein: Adaptive Volterra filters for acoustic echo cancellation, *Proc. NSIP '99,* **2**, 679–683, Antalya, Turkey, June 1999.