PART IV

ROBUST ADAPTIVE FILTERING ARCHITECTURES

—Wherever we are, what we hear is mostly noise.
When we ignore it, it disturbs us.
When we listen to it, we find it fascinating.
John Cage

10

Filtering Architectures Based on Adaptive Combination of Filters

Contents

10.1 Adaptive combination of filters **146**

10.2 Convex combination of adaptive filters **149**

10.3 Adaptation of mixing parameters **152**

10.3.1 LMS adaptation of a convex combination of two filters 152

10.3.2 A normalized adaptation 153

10.4 Conclusions **155**

In the previous two parts of this work we have seen interesting adaptive algorithms for linear and nonlinear modelling of the acoustic impulse response. Even if such algorithms have shown remarkable results not always they provide optimal performance. In fact, they might suffer the initial choice of parameter settings when conditions of the environment, or in general of a system to identify, change during the adaptation, such that the initial setting becomes unsatisfying. In the linear case, such a situation may occur due to a nonstationary or a change in the environment which leads to a different choice of the step size parameter rather than the filter length or the regularization factor. Similarly, in the nonlinear case, a kind of nonlinearity highly varying, in amplitude or in time, may require to change the filter design during the adaptation. Moreover, another important troubling situation occurs when the desired signal is not known *a priori*, thus it is difficult to choose whether adopting a linear filter or a nonlinear model.

In order to tackle these problems we introduce robust adaptive filtering architectures based on the *adaptive combination of filters*. The idea of filters combination is very interesting because it is possible to model a wide range of applications [81, 67]. Using such technique it is possible to develop *combined filtering architectures* able to change their parameter setting automatically during the adaptation. An experimental example of combined filtering architectures for acoustic application can be found in Chapter 11.

Moreover, the adaptive combination of filters may be used also to develop *collaborative filtering architectures* able to model an impulse response apart from its nature, whether it is linear or nonlinear. This results very useful in acoustic applications, such as AEC, when it is not possible to know *a priori* if the AIR conveys any nonlinearity, thus biasing the design choices about an acoustic echo canceller. An experimental example of collaborative filtering architecture for AEC can be found in Chapter 12.

However, first of all in this chapter it is necessary to introduce the adaptive combination of filters.

10.1 ADAPTIVE COMBINATION OF FILTERS

Real-world processes comprise both linear and nonlinear components, together with deterministic (that can be precisely described by a set of equations) and stochastic ones. In this way, models used to describe these real-world processes can be classified with a certain degree of nonlinearity and uncertainty, and described in a diagram (see Fig. 10.1). In literature only few cases

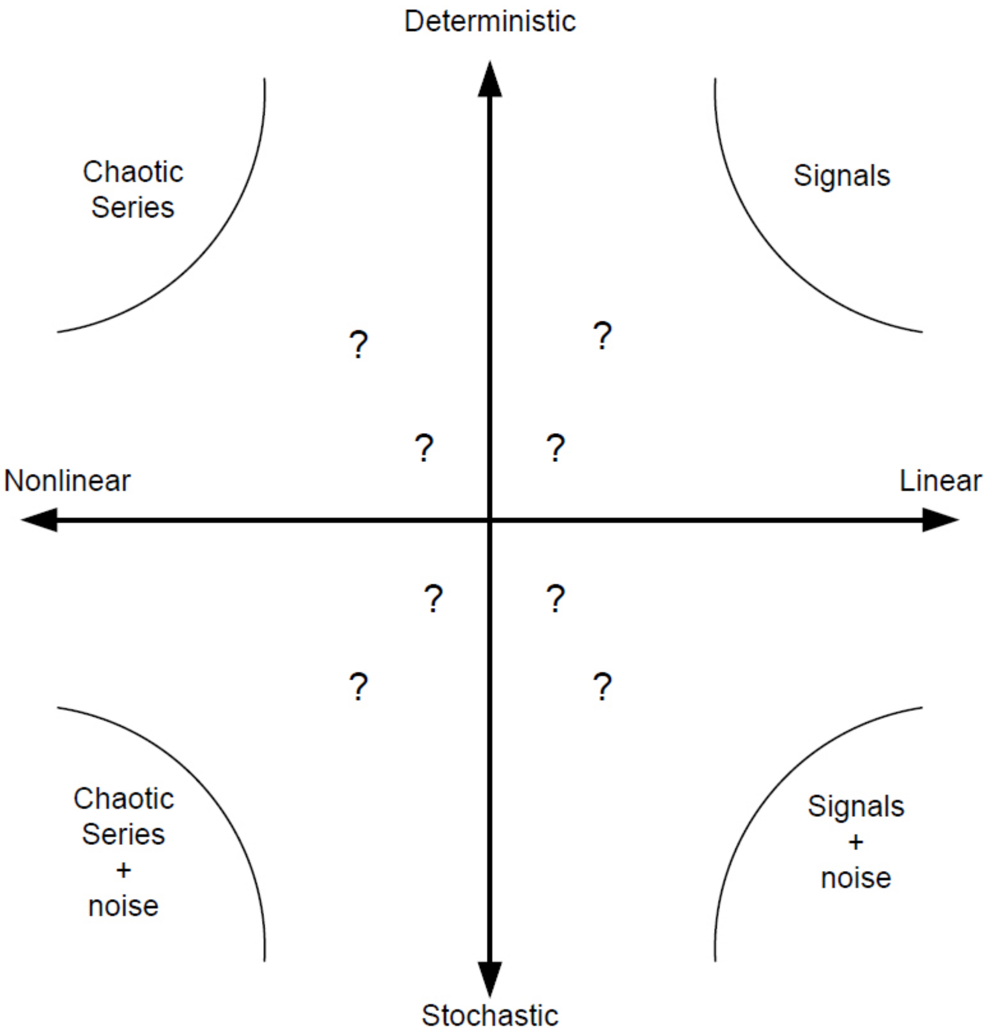

Fig. 10.1: *Possible variety of signals spanned by a certain degree of nonlinearity and uncertainty.*

as the linear stochastic ARMA and chaotic models are well understood, while real-world processes are often a combination of the previous four possibilities. In order to automatically take into account all the previous possibilities, a possible solution is to think to a system that automatically selects the right subsystem working on the relative quadrant.

It is possible to generalize Fig. 10.1 to the adaptive filtering, such that each subsystem corresponds to an adaptive filter. Using the fusion of the outputs of adaptive filters it is possible to produce a single hybrid filtering architecture

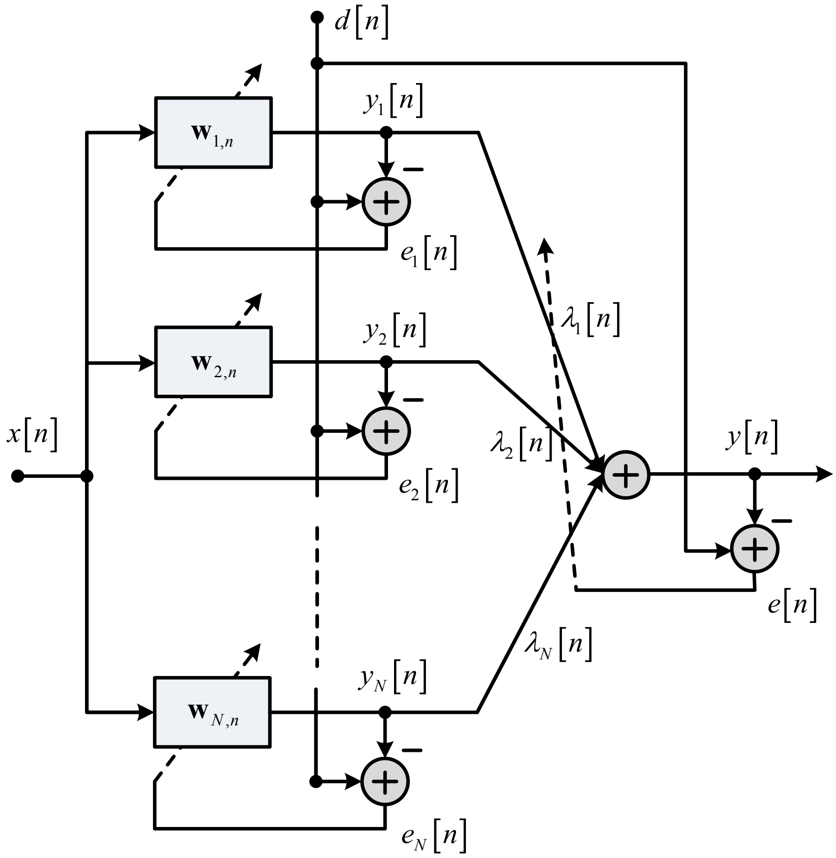

Fig. 10.2: *Adaptive combination of transversal adaptive filters.*

which provides at each time-instant the best performance among those of individual adaptive filters [67].

Adaptive combination of filters, as depicted in Fig. 10.2, consists of multiple individual adaptive subfilters operating in parallel and all feeding into a mixing algorithm which produces the single output of the filter [5, 73]:

$$\begin{aligned} y\left[n\right] &= \sum_{i=1}^{N} \lambda_i \left[n\right] y_i \left[n\right] \\ &= \sum_{i=1}^{N} \lambda_i \mathbf{x}_{i,n}^T \mathbf{w}_{i,n-1} \end{aligned} \tag{10.1}$$

where N is the number of filters in parallel, $y_i[n]$, are the outputs of the individual filters, with $i = 1, \dots, N$, and $\lambda_i[n]$ are the mixing parameters, which are nothing but the coefficients of the filter on the output stage. Such mixing parameters can be updated using an adaptive algorithm. Therefore, the mixing coefficients are also adaptive and combine the outputs of each subfilter based on the estimate of their current performance on the input signal from their instantaneous output error. The mixing parameters are updated in such a way to minimize the global MSE in output. This minimization may be subjected to a constraint. The most used optimization constraints in the adaptive combination of filters are the affine and the convex constraints.

The *affine combination of adaptive filter* is characterized by an affine constraint, according to which:

$$\sum_{i=1}^{N} \lambda_i[n] = 1. \tag{10.2}$$

On the other side, the *convex combination of filters*, in addition to satisfy the affine constraint, is characterized by the fact that all the mixing parameters are not negative, i.e.:

$$\sum_{i=1}^{N} \lambda_i[n] = 1 \quad \text{with} \quad 0 \leq \lambda_i[n] \leq 1, \;\; i = 1, \dots, N \tag{10.3}$$

In the next section we deepen the convex combination which is quite used in acoustic applications.

10.2 CONVEX COMBINATION OF ADAPTIVE FILTERS

A simple form of mixing algorithm for two adaptive filters is a convex combination. Convexity can be described as [26]:

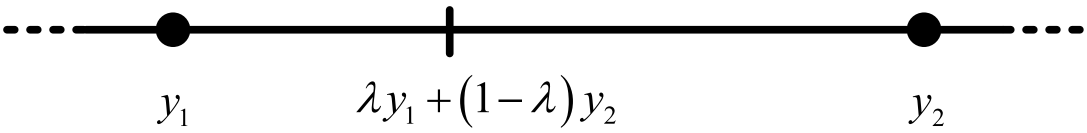

Fig. 10.3: *Convexity.*

$$\lambda y_1 + (1-\lambda)\, y_2 \tag{10.4}$$

where $\lambda \in [0,1]$. For y_1 and y_2 being two points on a line, as shown in Fig. 10.3, their convex mixture (10.4) will lie on the same line between y_1 and y_2. The convex combination between two adaptive filters is represented in Fig. 10.4, in which, due to the convex constraint, the mixing parameters can be written as $\lambda_1 = \lambda$ and $\lambda_2 = 1 - \lambda$.

Therefore, in this case the output of the combined structure can be written as:

$$y\,[n] = \lambda y_1\,[n] + (1-\lambda)\, y_2\,[n] \tag{10.5}$$

It has been showed, in [5, 6], that the convex combination method is universal with respect to the component filters, i.e., in steady-state, it performs at least as well as the best component filter. Furthermore, when the correlation between the *a priori* errors of the components is low enough, their combination is able to outperform both of them [6]. This is the reason why the convex combination of filters is very attractive in adaptive filtering. In fact, it is known that on-line adaptation of certain filter parameters or even cost functions has been attempted to influence filter performance, such as adjusting the forgetting factor of recursive least squares (RLS) algorithms [164] or minimizing adjustable cost functions [25, 105]. However, a widespread use of adaptive combination of filters is to optimally set the step size parameter. Variable step size adaptive filters (see also Section 5.5) allow the filters to dynamically adjust

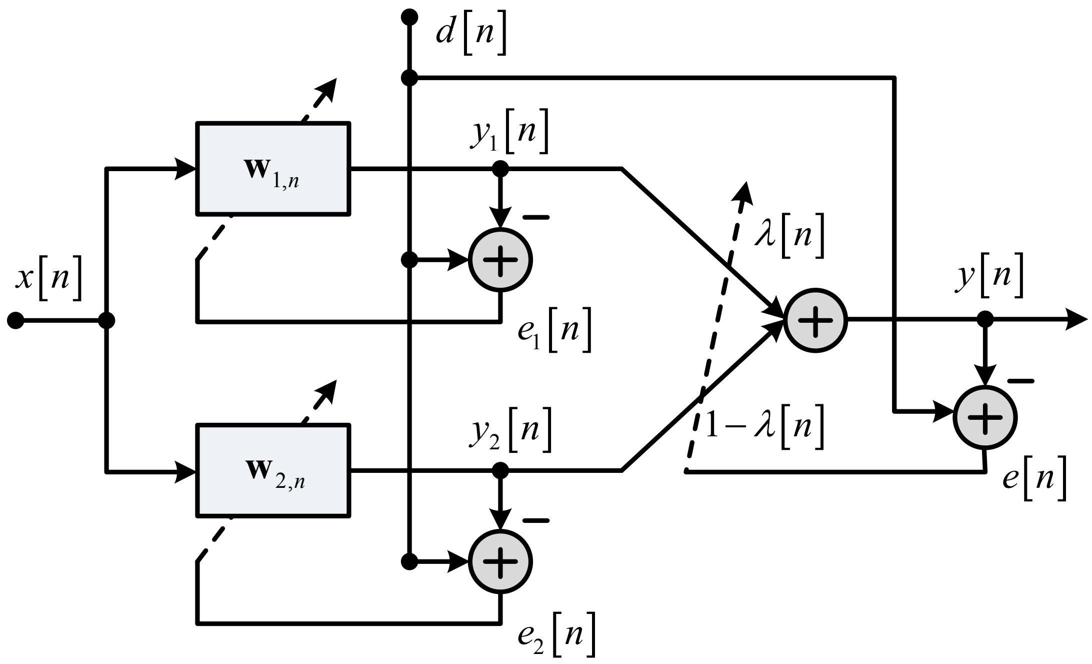

Fig. 10.4: *Convex combination of two adaptive filters.*

their performance in response to conditions in the input data and error signals [58, 75, 124]. For example, it is possible to choose a convex combination of two adaptive filters [84, 8], one fast, i.e. with a large step size value, and one slow, i.e. with a small step size value. These filters are combined in such a manner that the advantages of both component filters are kept: the rapid convergence from the fast filter, and the reduced steady-state error from the slow filter. This scheme, that has also proven to outperform previous variable step approaches, is an analogy of a well-known neurological fact: human brains combine fast and coarse reactions against abrupt changes in the environment, with an early processing at the amygdala, and more elaborated but slower responses taken in the neocortex at a conscious level [7].

10.3 ADAPTATION OF MIXING PARAMETERS

The adaptation of the mixing parameters follows the updating rule of stochastic gradient adaptive algorithms (see Section 4.4). As it is possible to see also from Fig. 10.2 and Fig. 10.4, the individual filters are independently adapted using their own error signals, while the combination, both affine and convex, is adapted by means of a stochastic gradient algorithm in order to minimize the error of the overall structure. In this section we introduce the LMS and the NLMS adaptation for the mixing parameters, however other stochastic gradient algorithms might be adopted.

10.3.1 LMS adaptation of a convex combination of two filters

Let us consider the convex combination of two adaptive filters, as depicted in Fig. 10.4, described by equation (10.5). Let M the length of both the adaptive filter and let the input signal buffer $\mathbf{x}_n \in \mathbb{R}^M$. The *least mean square* updating equations for the two filters result:

$$\mathbf{w}_{i,n} = \mathbf{w}_{i,n-1} + \mu_i \mathbf{x}_n^T e_i[n], \quad \text{with } i = 1, 2 \tag{10.6}$$

where:

$$e_i[n] = d[n] - y_i[n] \tag{10.7}$$

is the instantaneous error relative to individual filters.

Concerning the mixing parameter $\lambda[n]$, the adaptation may be carried out in convex mode imposing that $0 \leq \lambda[n] \leq 1$ by means of a sigmoidal activation function defined as:

$$\begin{aligned} \lambda[n] &= \operatorname{sgm}(a[n]) \\ &= \frac{1}{1 + e^{-a[n]}} \end{aligned} \tag{10.8}$$

i.e., such that $\lambda[n]$ derive from the adaptation of an auxiliary parameter, $a[n]$, which is updated by means of a gradient descent rule, such as $a[n+1] = a[n] + \Delta a[n]$. Therefore, $\Delta a[n]$ may be computed applying a *least mean square* adaptation rule:

$$\begin{aligned}
\Delta a[n] &= -\frac{1}{2}\mu_a \frac{\partial e^2[n]}{\partial a[n]} \\
&= -\mu_a e[n] \frac{\partial \left(d[n] - \lambda[n] y_1[n] - (1-\lambda[n]) y_2[n]\right)}{\partial \lambda[n]} \frac{\partial \lambda[n]}{\partial a[n]} \\
&= \mu_a e[n] \left(y_1[n] - y_2[n]\right) \lambda[n] \left(1 - \lambda[n]\right).
\end{aligned} \tag{10.9}$$

where μ_a is a step size parameter.

The benefits of employing the sigmoidal activation function are twofold. First, it serves to keep $\lambda[n]$ within the desired range $[0, 1]$. Second, as seen from (10.9), the adaptation rule of $a[n]$ reduces both the stochastic gradient noise and the adaptation speed near $\lambda[n] = 1$ and $\lambda[n] = 0$ when the combination is expected to perform close to one of its component filters without degradation. Still, note that the update of $a[n]$ in (10.9) stops whenever $\lambda[n]$ is too close to the limit values of 0 or 1. To circumvent this problem, we shall restrict the values of $a[n]$ to lie inside a symmetric interval $[-a^+, a^+]$, which limits the permissible range of $\lambda[n]$ to $[1-\lambda^+, \lambda^+]$, where $\lambda^+ = \mathrm{sgm}(a^+)$ is a constant close to 1. In this way, a minimum level of adaption is always guaranteed.

10.3.2 A normalized adaptation

In [9] a normalized adaptation scheme has been introduced in order to be more robust to changes in the filtering scenario. Considering equation (10.7), it is possible to rewrite (10.5) as:

$$y[n] = y_2[n] + \lambda[n]\left(e_2[n] - e_1[n]\right) \tag{10.10}$$

so that we can think of the overall combination scheme as a two-stage adaptive filter. In the first stage, the two component filters operate independently of each other and according to their own rules, while the second layer consists of a filter with input signal $e_2[n] - e_1[n]$ that minimizes the overall error.

This interpretation of the combination scheme suggests that further advantages could be obtained if we used a normalized LMS rule for adapting the mixing parameter rather than standard LMS. Since $e_2[n] - e_1[n]$ plays the role of the input signal at this level, it makes sense to use the following adaptation scheme:

$$a[n+1] = a[n] + \mu_a \frac{\lambda[n](1-\lambda[n])}{(e_2[n]-e_1[n])^2} e[n](e_2[n]-e_1[n]). \tag{10.11}$$

In practice, however, the performance of this scheme is quite unsatisfactory given that the instantaneous value $(e_2[n] - e_1[n])^2$ is a very poor estimate of the power of the "second stage" input signal. Similar to the normalized LMS (NLMS) algorithm with power normalization [120], better behaviour is obtained from:

$$a[n+1] = a[n] + \frac{\mu_a}{r[n]} \lambda[n](1-\lambda[n]) e[n](e_2[n]-e_1[n]) \tag{10.12}$$

where:

$$r[n] = \beta r[n-1] + (1-\beta)(e_2[n]-e_1[n])^2 \tag{10.13}$$

is a rough (low-pass filtered) estimate of the power of the signal of interest. Selection of the forgetting factor β is rather easy. For instance, using $\beta = 0.9$ gives a good enough approximation, and typically ensures that $r[n]$ is adapted faster than any component filter.

10.4 CONCLUSIONS

In this chapter adaptive combination of filters has been introduced. In the following two chapters we use this technique to develop robust combined filtering architectures, for the linear modelling, and collaborative filtering architectures, for the nonlinear modelling. Adaptive combination of filters still remains a fertile argument for future researches since, as we have seen, the adaptation of mixing parameters is conducted by means of stochastic adaptive algorithms; it can be thinkable to adopt more appropriate adaptation rules, especially for the modelling of an acoustic path.