5

Proportionate Adaptive Algorithms

Contents

In the recent past, a family of proportionate adaptive filters has been proposed for use in network telephony and acoustic applications. Proportionate algorithms offer better convergence and tracking performances than standard stochastic algorithms when the echo path is sparse. In this chapter, we describe proportionate algorithms introducing an alternative perspective on proportionate adaptive filters.

5.1 INTRODUCTION

Nowadays, *acoustic echo cancellation* (AEC) is a key application in modern speech communication systems. Echo phenomena are generated in speech devices by a microphone-loudspeaker coupling, such as a far-end signal is sent out by a loudspeaker and crosses an echo path before being acquired by the microphone. Therefore, the acquired signal contains an echo contribution which may be cancelled by means of an acoustic echo canceller. The main component of an echo canceller is the adaptive filter which aims at estimating the *acoustic impulse response* (AIR). Such applications require adaptive filters with hundreds or even thousands of taps and their success depends on the nature of the AIR [57]. Often enough the impulse response is time-varying and it is affected by echo path changes, different degrees of sparseness, double-talk events and under-modelling noise [57, 12].

Classic algorithms based on stochastic gradient, such as *least mean square* (LMS) and *normalized LMS* (NLMS), distribute the adaptation energy among all filter coefficients causing a very slow convergence for long filters [120, 59]. As a result, the application of these filtering algorithms to acoustic applications becomes unpractical. In order to address this problem, in the last years it has been conceived to act on the nature of AIRs. In fact, for both network and acoustic scenarios, echo path have a specific property, which can be used in order to help the adaptation process. Indeed, these systems are sparse in nature, i.e., only a small percentage of the impulse response components have a significant magnitude while the rest are zero or small [40].

The "sparseness" character of the echo paths inspired the idea to "proportionate" the algorithm behaviour, i.e., to update each coefficient of the filter independently of the others, by adjusting the adaptation step size in proportion to the magnitude of the estimated filter coefficient. In this manner, the adaptation gain is "proportionately" redistributed among all the coefficients, emphasizing the large ones in order to speed up their convergence, and consequently to increase the overall convergence rate. This means that the

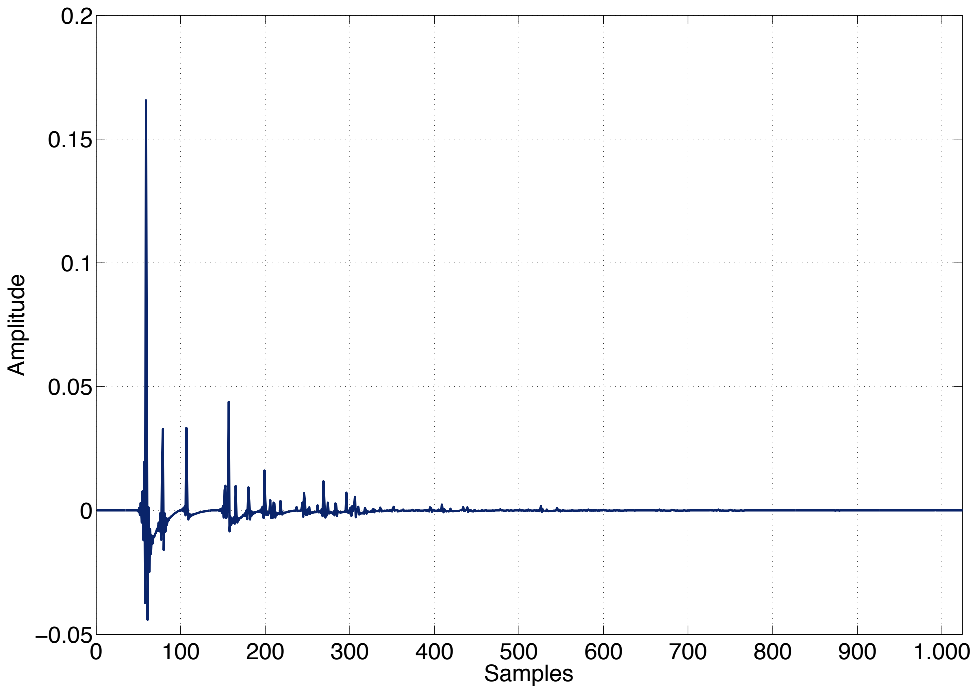

Fig. 5.1: *A sparse acoustic impulse response.*

region with higher energy of the sparse impulse response is adapted faster than the tail of the AIR. An example of sparse AIR can be found in Fig. 5.1, in which the difference between the early reflections and the tail of the AIR is quite clear.

The first proportionate algorithm was proposed by Duttweiler [40]; he defined the *Proportionate NLMS Algorithm* (PNLMS) algorithm, whose idea was to make the step size of each tap proportional to current absolute value of the estimated weight. PNLMS converges and tracks much faster than the NLMS algorithm when the impulse response that we need to identify is sparse. However, its behaviour degrades significantly when the impulse response is dispersive. PNLMS++ algorithm [50] partially solves the above mentioned problem by alternating the update process between NLMS and PNLMS. PNLMS++ seems a little bit less sensitive to the assumption of a sparse impulse response than PNLMS, so it is far from the optimal solution. In [13], the *improved PNLMS* (IPNLMS) was proposed where each step size shows a better balance between the fixed step size of NLMS and the large amount of proportionality in PNLMS. As a result, IPNLMS always converges and tracks better than NLMS and PNLMS, no matter how sparse the impulse response is.

Another filter that unevenly weights the adaptation of the different taps of the filter is the *EG± algorithm* [71], based on the exponentiated gradient adaptation. Nevertheless, it has been proved [14, 93] that IPNLMS is a very good approximation of the EG± algorithm, while being more convenient from a practical point of view. Unfortunately, as any other gradient-based adaptive filter, IPNLMS is subject to some compromises due to the selection of its parameters. As a matter of fact, a large step size results in faster convergence, while the residual misalignment is reduced for small step sizes. Moreover, the choice of the proportionality factor imposes a behaviour trade-off for channels with different degrees of sparseness [13].

In order to achieve faster convergence for a wide range of echo paths, it is possible to combine the ideas of proportionate algorithms with the general *affine projection algorithm* (APA). In [48], it is shown that a robust *proportionate affine projection algorithm* (PAPA) converge faster than NLMS and performs significantly better even during a double-talk situation. Moreover, in [119], it is proved that an *improved PAPA* (IPAPA) easily outperforms all the above mentioned proportionate algorithms and its performance does not depend on the type of the impulse response. Furthermore, the choice of a proper value for the proportionate factor has no any significant impaction on the IPAPA tracking properties comparing to the IPNLMS. Moreover, in the last years proportionate APAs have been improved [64, 152, 149, 78] until coming to an *efficient proportionate APA* [102], which takes into account the "history" of the proportionate factors.

Proportionate algorithms improve adaptive filtering performance when the AIR is sparse; however, even in proportionate algorithms some problems may occur in the choice of the parameters. A key parameter in adaptive echo cancellation is the *step size* which governs the stability and the adaptation speed of the filtering algorithm. The choice of the step size sets the trade-off between convergence, tracking ability and steady state misalignment. In order to achieve the best trade-off, several *variable step size* (VSS) algorithms have been proposed [58, 80, 124, 99]. In general, classic algorithms assume an exact modelling situation, i.e. the length of the adaptive filter is equal to the length of the system that has to be modeled. Since echo paths are extremely long, under-modelling situations, in which the length of the adaptive filter is shorter than the length of the echo path, often occur in echo cancelling applications. The residual echo due to the unmodelled part of the impulse response can be viewed as additional noise, also named under-modelling noise, that affects the performance of the algorithm. In [101], the under-modelling case has been considered.

In this chapter, we derive a novel perspective on proportionate algorithms and then we define a new block-based proportionate APA and a variation of it based on the recursive update of the covariance matrix. Furthermore, we investigate the introduction of a variable step size. The chapter is organized as follows: in Section 5.2 a new framework for the derivation of proportionate algorithms is derived. Section 5.3 introduces the derivation of algorithms using the new framework while the analytical description of the proposed *proportionate block APA* is introduced in Section 5.4. In Section 5.5, variable step size based proportionate algorithms are investigated.

5.2 AN ALTERNATIVE PERSPECTIVE ON PROPORTIONATE ADAPTIVE FILTERS

In order to give an overall description of the proportionate algorithms, we derive a general framework based on a novel perspective on the proportionate algorithms using a *natural gradient* adaptive rule [2] and employing the least perturbation property [120] by means of which we suggest the family of proportionate APA filters.

5.2.1 General properties of adaptive algorithms

Adaptive algorithms are usually introduced as an approximate iterative solution of a *global optimization problem* as they are derived, in the steepest descent implementation, by replacing the actual gradient vector with an instantaneous approximation of it (see Section 4.3). It turns out that, starting from an energy point of view and some general properties of the adaptive algorithms, it is possible to define a class of algorithms that can be seen as an *exact*, i.e. non-approximate, solution of a *local optimization problem* [120].

For this purpose, let us consider the regression vector $\mathbf{d}_n \in \mathbb{R}^K$ containing the K more recent samples of the observed desired signal:

$$\mathbf{d}_n = \begin{bmatrix} d\,[n] & d\,[n-1] & \dots & d\,[n-K+1] \end{bmatrix}^T \tag{5.1}$$

where K is known as *projection order* (see Section 4.4). Similarly, the data matrix of the input signal $\mathbf{X}_n \in \mathbb{R}^{K \times M}$ can be expressed as:

$$\mathbf{X}_n = \begin{bmatrix} \mathbf{x}_n^T \\ \mathbf{x}_{n-1}^T \\ \dots \\ \mathbf{x}_{n-K+1}^T \end{bmatrix}^T \tag{5.2}$$

$$= \begin{bmatrix} x\,[n] & x\,[n-1] & \cdots & x\,[n-M+1] \\ x\,[n-1] & x\,[n-2] & \cdots & x\,[n-M] \\ \vdots & \vdots & \ddots & \vdots \\ x\,[n-K+1] & x\,[n-K] & \cdots & x\,[n-K-M+2] \end{bmatrix}$$

Moreover, let us assume to dispose, at n-th time instant, of some weight estimate of the previous iteration, $\mathbf{w}_{n-1}$, so that it is possible to define the *a priori* error signal:

$$\mathbf{e}_n = \mathbf{d}_n - \mathbf{X}_n \mathbf{w}_{n-1}, \tag{5.3}$$

and the *a posteriori* error signal:

$$\boldsymbol{\varepsilon}_n = \mathbf{d}_n - \mathbf{X}_n \mathbf{w}_n. \tag{5.4}$$

Introducing the step size parameter $\mu[n]$ in its general time-varying form and denoting with $\boldsymbol{\alpha}_n \in \mathbb{R}^K = \operatorname{diag}(\mu_0[n], \ldots, \mu_{K-1}[n])$ the corresponding diagonal matrix, it is possible to write the relation between the *a posteriori* and the *a priori* error signals:

$$\boldsymbol{\varepsilon}_n = (\mathbf{I} - \boldsymbol{\alpha}_n)\, \mathbf{e}_n \tag{5.5}$$

It can be notice that in case of constant step size value the diagonal matrix can be written omitting the time index as $\boldsymbol{\alpha} = \mu \mathbf{I}$, where μ is the fixed step size value.

The relation (5.5), in which $\mathbf{0} < \boldsymbol{\alpha}_n < \mathbf{I}$, expresses an energy constraint between *a priori* and *a posteriori* errors, thus entailing the *passivity* of the corresponding adaptive circuit scheme.

Taking into account equation (5.5) and denoting with:

$$\widetilde{\mathbf{w}}_n = \mathbf{w}_n - \mathbf{w}_{n-1} \tag{5.6}$$

the vector that adjust the coefficients of the estimated filter, we can define a cost function as:

$$J(\mathbf{w}_n) = \|\widetilde{\mathbf{w}}_n\|_2^2 \tag{5.7}$$

Due to the fact that the filter weights at steady state no longer change during adaptation, it follows that any adaptive algorithm that minimized $J(\mathbf{w}_n)$ can be expressed as an *exact* method of local minimization, which is a constrained optimization problem:

$$\begin{aligned} \mathbf{w}^{\mathrm{opt}} &= \arg\min_{\mathbf{w}_n} \|\widetilde{\mathbf{w}}_n\|_2^2 \\ \text{subject to} \quad \boldsymbol{\varepsilon}_n &= (\mathbf{I} - \boldsymbol{\alpha}_n)\, \mathbf{e}_n \end{aligned} \tag{5.8}$$

Such optimization problem describes the steepest descent adaptation process. This process continues iteratively until the value of $J(\mathbf{w}_n)$ reaches a suitably-small value; at that point $\mathbf{w}_n$ is close to $\mathbf{w}^{\mathrm{opt}}$. With a proper selection of $\mu[n]$, the steepest descent method adjusts $\mathbf{w}_n$ in a way that $\lim_{n\to\infty} \mathbf{w}_n = \mathbf{w}^{\mathrm{opt}}$. Such an algorithm allows $\mathbf{w}_n$ to converge to $\mathbf{w}^{\mathrm{opt}}$.

Equation (5.8) represents the so-called *least perturbation property* and it is equivalent to seek a solution $\mathbf{w}_n$ that is closest to $\mathbf{w}_{n-1}$ in the Euclidean norm sense, under an equality constraint between $\mathbf{e}_n$ and $\boldsymbol{\varepsilon}_n$. The constraint is most relevant when $\mu[n]$ is a small value, such that $(\mathbf{I} - \boldsymbol{\alpha}_n) < \mathbf{I}$, because, when the step size $\mu[n]$ is small enough, the magnitude of the *a posteriori* error $\boldsymbol{\varepsilon}_n$ will always be less than that of the *a priori* error $\mathbf{e}_n$, i.e.:

$$|\boldsymbol{\varepsilon}_n| < |\mathbf{e}_n| \tag{5.9}$$

An important consequence of the least perturbation property is that *a priori* and *a posteriori* errors tend to zero at steady state. In other words, as explained in [71], an adaptive algorithm should be characterized by a reasonable balance between the conservative (keep information gained in previous iterations) and corrective requirements (ensure that any new information gained increases the result accuracy).

Therefore, in conclusion, any adaptive algorithm can be derived and characterized taking into account the following *general properties*:

(a) the magnitude of the *a posteriori* error is always less than the *a priori* error, i.e. $|\boldsymbol{\varepsilon}_n| < |\mathbf{e}_n|$;

(b) at steady state, for $n \to \infty$, the weights no longer change during adaptation (*least perturbation property*);

(c) at steady state, for $n \to \infty$, *a priori* and *a posteriori* errors tend to zero.

5.2.2 Natural gradient adaptation

In order to take advantage from these properties, instead of the steepest descent method we may adopt a different procedure to construct the coefficient updates that takes into account the "non-isotropic nature" of the parameter space. *Natural gradient adaptation* [2], [37] is a modified gradient search that changes the standard gradient update procedure according to the non-Euclidean nature of the parameter space [51]. The resulting updates are based on a "non-straight-line" distance metric that is defined by the Riemannian geometry of the parameter space [3], [3]. According to the natural gradient procedure, the cost function in (5.7) can be rewritten as:

$$\begin{aligned} J\left(\mathbf{w}_n\right) &= \|\widetilde{\mathbf{w}}_n\|^2_{\mathbf{G}_n} \\ &= \widetilde{\mathbf{w}}_n^T \mathbf{G}_n \widetilde{\mathbf{w}}_n \end{aligned} \tag{5.10}$$

where $\mathbf{G}_n \in \mathbb{R}^{M \times M}$ is a *Riemannian metric tensor*, which is a positive-definite matrix, whose entries at n-th time instant depend on the coefficients of the filter at time instant $n-1$. The Riemannian metric tensor characterizes the intrinsic curvature of a particular manifold in M-dimensional space. In the case of the Euclidean space the Riemannian tensor is the identity matrix $\mathbf{G}_n = \mathbf{I}$, such that (5.10) reduces to (5.7).

Before recasting the least perturbation property with the use of the Riemannian metric tensor, let us consider the following aspect. The formalization in (5.8) of the least perturbation property has merely theoretical significance as it is based on the knowledge of *a priori* and *a posteriori* errors. For a more constructive use of the general properties (a)-(c), it is necessary to define the energy constraint as function of the only *a priori* error. Left multiplying both sides of (5.6) with $\mathbf{G}_n \mathbf{X}_n$ and then adding and subtracting the desired signal vector $\mathbf{d}_n$ defined in (5.1), it is possible to express the energy constraint in (5.5) just as a function of the *a priori* error. That is:

$$\begin{aligned}\mathbf{G}_n\mathbf{X}_n\widetilde{\mathbf{w}}_n &= \mathbf{G}_n\mathbf{X}_n\mathbf{w}_n - \mathbf{G}_n\mathbf{X}_n\mathbf{w}_{n-1}\\ &= \mathbf{G}_n\left[-\left(\mathbf{d}_n - \mathbf{X}_n\mathbf{w}_n\right) + \left(\mathbf{d}_n - \mathbf{X}_n\mathbf{w}_{n-1}\right)\right]\\ &= \mathbf{G}_n\left(-\boldsymbol{\varepsilon}_n + \mathbf{e}_n\right)\\ &= \mathbf{G}_n\boldsymbol{\alpha}_n\mathbf{e}_n.\end{aligned} \tag{5.11}$$

Hence, we can formally rewrite the least perturbation property (5.8) as:

$$\begin{aligned}\mathbf{w}^{\mathrm{opt}} &= \arg\min_{\mathbf{w}_n}\|\widetilde{\mathbf{w}}_n\|^2_{\mathbf{G}_n}\\ \text{subject to} \quad \mathbf{G}_n\mathbf{X}_n\widetilde{\mathbf{w}}_n &= \mathbf{G}_n\boldsymbol{\alpha}_n\mathbf{e}_n.\end{aligned} \tag{5.12}$$

The update equation can be straightly derived solving the system relative to the constraint (5.11). Thus, it results:

$$\widetilde{\mathbf{w}}_n = \left(\mathbf{G}_n\mathbf{X}_n\right)^{\#}\boldsymbol{\alpha}_n\mathbf{e}_n \tag{5.13}$$

where $\left(\mathbf{G}_n\mathbf{X}_n\right)^{\#}$ is a pseudo-inverse matrix. Expliciting $\widetilde{\mathbf{w}}_n$ we can write:

$$\mathbf{w}_n - \mathbf{w}_{n-1} = \left(\mathbf{G}_n\mathbf{X}_n\right)^T\left(\mathbf{X}_n\mathbf{G}_n\mathbf{X}_n^T\right)^{-1}\boldsymbol{\alpha}_n\mathbf{e}_n. \tag{5.14}$$

Inserting the regularization parameter δ, we achieve the general update equation of the family of *normalized natural gradient* (NNG) algorithms:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \left(\mathbf{G}_n\mathbf{X}_n\right)^T\left(\delta\mathbf{I} + \mathbf{X}_n\mathbf{G}_n\mathbf{X}_n^T\right)^{-1}\boldsymbol{\alpha}_n\mathbf{e}_n. \tag{5.15}$$

In case of Euclidean space, when $\mathbf{G}_n = \mathbf{I}$, for a unitary projection order, i.e. $K = 1$, and a fixed step size, i.e. each diagonal element of $\boldsymbol{\alpha}$ is equal to a fixed scalar value μ, the update equation (5.15) describes the *normalized least mean square* (NLMS) algorithm:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu \frac{\mathbf{x}_n e[n]}{\mathbf{x}_n^T \mathbf{x}_n + \delta_{\mathrm{NLMS}}} \tag{5.16}$$

On the other hand, when the projection order is $K > 1$, equation (5.15) yields the *affine projection algorithm* (APA) in its standard form [98] in case of Euclidean space, or the *natural APA* (NAPA) [65], in case of Riemannian space.

5.3 DERIVATION OF PROPORTIONATE ALGORITHMS

Starting from equation (5.15), it is possible to derive a complete formulation of the class of proportionate algorithms. Different proportionate algorithms can be obtained simply changing the projection order K and the Riemannian tensor $\mathbf{G}_n$. In particular, in proportionate algorithms, the Riemannian tensor is consider as a full-blown sparseness constraint which weight the input signal; this is why $\mathbf{G}_n$ is called *proportionate matrix*.

The simplest proportionate algorithm is the *proportionate normalized least mean squares* in its improved version (IPNLMS) [13], whose derivation can be achieved choosing a unitary projection order $K = 1$ and a diagonal proportionate matrix $\mathbf{G}_n \in \mathbb{R}^{M \times M}$ built up in order to adjust the step sizes of the individual taps of the filter in a way that each step size turns out to be proportional to the corresponding filter coefficient:

$$\mathbf{G}_n = \mathit{diag}\left\{g_0[n], \ldots, g_{M-1}[n]\right\} \tag{5.17}$$

The diagonal elements at n-th time instant are computed from the estimate of the filter coefficients at time instant $n-1$ in such a way that a larger coefficient receives a larger increment, thus increasing the convergence rate of the coefficient. The result is that active coefficients are adjusted faster than non-active coefficients. Hence, proportionate algorithms converge much faster than classic algorithms for sparse impulse responses.

The choice of diagonal elements differentiates proportionate NLMS algorithms proposed in literature [40, 50, 13]. However, the most efficient choice, which exploits the "proportionate" idea better than other PNLMS algorithms, is the one proposed in the IPNLMS [13]. According to that, diagonal elements are:

$$g_l[n] = \frac{1-\alpha_\mathrm{p}}{2M} + (1+\alpha_\mathrm{p})\frac{|w_l[n-1]|}{2\|\mathbf{w}_{n-1}\|_1 + \xi} \tag{5.18}$$

where:

$$\|\mathbf{w}_{n-1}\|_1 = \sum_{l=0}^{M-1} |w_l[n-1]| \tag{5.19}$$

In (5.18), the coefficient index $l = 0, \ldots, M-1$ and ξ is a small positive number which avoids divisions by zero; the *proportionality factor* α_p balances the proportionality and its recommended value is 0 or -0.5 [13]. For $\alpha_\mathrm{p} = -1$, the IPNLMS is equal to NLMS. For α_p close to 1, the IPNLMS behaves like the PNLMS. The regularization parameter δ_p in IPNLMS is chosen as:

$$\delta_\mathrm{p} = \frac{1-\alpha_\mathrm{p}}{2M}\delta_\mathrm{NLMS}. \tag{5.20}$$

Similarly to the development of PNLMS and IPNLMS, if we consider a projection order $K > 1$, we can derive the *proportionate affine projection algorithm* (PAPA) [48] and the *improved PAPA* (IPAPA) [64, 119]. However, we describe an efficient version of proportionate APA which considers the "history" of the proportionate factors [102]. Besides the projection order, the relevant difference of the proportionate APA compared to IPNLMS is the construction of $\mathbf{G}_n$. In fact, the proportionate matrix for $K > 1$ can be built up as a rectangular matrix, that we denote as $\mathbf{G'}_n \in \mathbb{R}^{K\times M}$ to distinguish from (5.17), in which the first row contains the proportionate weight computed at n-th time instant, $\mathbf{g}_n \in \mathbb{R}^M = \begin{bmatrix} g_0[n] & \ldots & g_{M-1}[n] \end{bmatrix}$, while the other $K-1$ rows contain the previous $K-1$ realizations of $\mathbf{g}_n$:

$$\mathbf{G'}_n = \begin{bmatrix} \mathbf{g}_n^T \\ \mathbf{g}_{n-1}^T \\ \cdots \\ \mathbf{g}_{n-K+1}^T \end{bmatrix}. \tag{5.21}$$

The matrix product in (5.15) can be written in this case as a Hadamard product:

$$\begin{aligned} \mathbf{\Gamma}_n &= \mathbf{G'}_n \odot \mathbf{X}_n \\ &= \begin{bmatrix} \mathbf{g}_n^T \odot \mathbf{x}_n^T \\ \mathbf{g}_{n-1}^T \odot \mathbf{x}_{n-1}^T \\ \cdots \\ \mathbf{g}_{n-K+1}^T \odot \mathbf{x}_{n-K+1}^T \end{bmatrix} \end{aligned} \tag{5.22}$$

where the operator $\odot$ denotes the Hadamard product, i.e. $\mathbf{a} \odot \mathbf{b} = \begin{bmatrix} a_0 b_0 & a_1 b_1 & \dots & a_{M-1} b_{M-1} \end{bmatrix}^T$, being $\mathbf{a}$ and $\mathbf{b}$ two vectors of length M. Therefore, using (5.22), the update equation of (5.15) can be rewritten in case of PAPA algorithms as:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \boldsymbol{\alpha} \mathbf{\Gamma}_n^T \left(\delta_{\mathrm{p}} \mathbf{I} + \mathbf{\Gamma}_n \mathbf{X}_n^T \right)^{-1} \mathbf{e}_n. \tag{5.23}$$

Due to the fact that equation (5.21) takes into account the past $K-1$ realization of the proportionate elements, the PAPA described in (5.23) can be considered as an efficient algorithm since this "proportionate memory" increases its performance [102].

Another advantage of the PAPA in (5.23) is the lower computational complexity compared with the classical proportionate-type APA, such as [48, 64, 119]. This is because the matrix $\mathbf{\Gamma}_n$ in (5.22) can be realized recursively, since it contains $K-1$ rows, whose products are computed in previous iterations. Thus, the rows from 1 to $K-1$ of the matrix $\mathbf{\Gamma}_{n-1}$ can be used directly for computing the matrix $\mathbf{\Gamma}_n$, i.e. they become the rows from 2 to K of $\mathbf{\Gamma}_n$.

This is not the case of the classical proportionate-type APA, where all the rows of $\mathbf{\Gamma}_n$ have to be evaluated at each iteration, because all of them are multiplied with the same vector $\mathbf{g}_n$. Concluding, the evaluation of $\mathbf{\Gamma}_n$ in the classical proportionate APAs needs KM multiplications, while the evaluation of $\mathbf{\Gamma}_n$ from (5.22), i.e. considering the "proportionate memory", requires only M multiplications. This advantage becomes more apparent when the projection order increases. Moreover, the fact that $\mathbf{\Gamma}_n$ has the time-shift property, like the data matrix $\mathbf{X}_n$, could be a possible opportunity to establish a link with the *fast APA* [52, 140]. It is also likely possible to derive efficient ways to compute the linear system involved in (5.23). This point in particular will address in the next section.

5.4 PROPORTIONATE BLOCK APA

In this section we propose a variation of the PAPA described in (5.23) based on the block processing of the input signal [141, 11, 1, 115]. Block processing is an effective approach to reduce the computational complexity, however in proportionate case it may assume a further sense due to the time-shift properties of the proportionate input matrix. In fact, in sample-by-sample PAPA the time-shift property of the input matrix is the same of the proportionate matrix allowing a computational saving; however, in *proportionate block APA* (PBAPA), the proportionate matrix is still subjected to the same shifting of PAPA, while the input matrix is subjected to a shift equal to the length of block. This may initially appear as a drawback since at each iteration the whole data matrix has to be weighted by the whole proportionate matrix, thus loosing the computational advantage of PAPA. Moreover, PBAPA may show a slower convergence rate due to less frequent updating of the adaptive filter. However, choosing a block length equal to the projection order, the computation cost remains the same of PAPA due to the fact that the block processing requires $1/K$ of the iterations compared to PAPA, thus the computational cost results $KM/K = M$. Moreover, the different time-shifting properties of $\mathbf{X}_n$ and

$\mathbf{G}_n$ can be seen as an interpolation of the proportionate weighting over input blocks, and this may improve the steady-state behaviour of the filter compared to PAPA.

The update equation of PBAPA is similar to (5.23); however, at each iteration the input data matrix (5.2) does not receive just a sample but a block of K samples. Moreover, a further correction term can be introduced to narrow the convergence gap with the sample-by-sample PAPA and to develop a *fast* version of PBAPA. In addition, it can be also possible to derive efficient ways to compute the inversion of the covariance matrix by means of recursive techniques, following what done in [141, 134].

5.5 VARIABLE STEP SIZE PROPORTIONATE ALGORITHMS

The overall performance of proportionate algorithms is governed by the step size parameter, which controls the filter trade-off between convergence, tracking ability and steady state misalignment. A constant value of the step size can set *a priori* performance compromise, however, it is not an optimal solution and in many cases it can produce not satisfying performance. In particular, this may occur in acoustic applications, in which nonstationary signals, such as speech, may alter initial conditions. In order to address this compromise, a *variable step size* (VSS) may be adopted. Therefore, even for proportionate algorithms, a performance improvement may be expected using a variable step size. Considering a variable step size, each element of the diagonal matrix $\boldsymbol{\alpha}_n$ in (5.15) may be different from the others, being time-varying.

In this section we derive the overall formulation of VSS-based proportionate algorithms, starting from equation (5.15), from which it is possible to derive the VSS-IPNLMS or the VSS-PAPA in its several versions. We generalize the proportionate algorithms in order to achieve a better robustness also in nonstationary conditions, double-talk events, path changes and under-modelling situations of the impulse response. For this purpose we introduce the *generalized variable step size proportionate algorithm for under-modelling scenarios*.

Under-modelling situations occur when the length of the adaptive filter is shorter than the length of the echo path, and this is often the rule in acoustic applications where AIRs are extremely long for a real-time adaptation. Under-modelling an AIR may introduce an additional noise to the near-end signal, generated by the part of the system that cannot be modelled. The power of the under-modelling noise cannot be estimated in a direct way due to the fact that it is not available in a real scenario. Therefore, its contribution cannot be evaluated.

Denoting with M_{A} the length of the acoustic impulse response $\mathbf{w}_0$, let us consider an under-modelling situation in which $M < M_{\text{A}}$; it is possible to break up the data input matrix in the following way:

$$\mathbf{X}_{\text{UM},n} = \begin{bmatrix} \mathbf{X}_n & \mathbf{X}_{\text{A},n} \end{bmatrix} \tag{5.24}$$

where $\mathbf{X}_n$ is defined as (5.2), and $\mathbf{X}_{\text{A},n} \in \mathbb{R}^{K\times(M_A-M)}$ is the data matrix referred to the under-modelled part of the AIR:

$$\mathbf{X}_{\text{A},n} = \begin{bmatrix} \mathbf{x}_{\text{A},n}^T \\ \mathbf{x}_{\text{A},n-1}^T \\ \cdots \\ \mathbf{x}_{\text{A},n-K+1}^T \end{bmatrix}^T \tag{5.25}$$

$$= \begin{bmatrix} x[n-M] & x[n-M-1] & \cdots & x[n-M_{\text{A}}+1] \\ x[n-M-1] & x[n-M-2] & \cdots & x[n-M_{\text{A}}] \\ \vdots & \vdots & \ddots & \vdots \\ x[n-M-K+1] & x[n-M-K] & \cdots & x[n-K-M_{\text{A}}+2] \end{bmatrix}$$

Similarly, in an under-modelling scenario, we can split the AIR in two parts, a modelled part and an unmodelled one:

$$\mathbf{w}_{0,\mathrm{UM}} = \begin{bmatrix} \mathbf{w}_0 & \mathbf{w}_{0,\mathrm{A}} \end{bmatrix} \tag{5.26}$$

where:

$$\mathbf{w}_0 = \begin{bmatrix} w_{0,0} & w_{0,1} & \dots & w_{0,M-1} \end{bmatrix} \tag{5.27}$$

and:

$$\mathbf{w}_{0\mathrm{A}} = \begin{bmatrix} w_{0,M} & w_{0,M+1} & \dots & w_{0,M_A-1} \end{bmatrix}. \tag{5.28}$$

Let us note that the AIR vectors do not have any time index since they are assumed to be time invariant.

As a consequence, taking into account K subsequent realizations, the resulting echo path in under-modelling case, that we denote as $\overline{\mathbf{x}}_{\mathrm{UM},n} \in \mathbb{R}^K$, can be decomposed in a modelled term $\overline{\mathbf{x}}_n$ and an unmodelled term $\overline{\mathbf{x}}_{\mathrm{A},n}$, which represents the under-modelling noise:

$$\begin{aligned} \overline{\mathbf{x}}_{\mathrm{UM},n} &= \overline{\mathbf{x}}_n + \overline{\mathbf{x}}_{\mathrm{A},n} \\ &= \mathbf{X}_n \mathbf{w}_0 + \mathbf{X}_{\mathrm{A},n} \mathbf{w}_{0\mathrm{A}} \end{aligned} \tag{5.29}$$

The term $\overline{\mathbf{x}}_{\mathrm{A},n}$ acts like an additional noise for the adaptive process, so that the desired signal in under-modelling case can be rewritten as:

$$\mathbf{d}_{\mathrm{UM},n} = \overline{\mathbf{x}}_n + \overline{\mathbf{x}}_{\mathrm{A},n} + \mathbf{q}_n \tag{5.30}$$

where $\mathbf{q}_n$ is the near-end contribution which can be composed of a near-end speech signal $\mathbf{s}_n$ and a near-end background noise $\mathbf{v}_n$. In (5.30) we assume that $\overline{\mathbf{x}}_n$ and $\overline{\mathbf{x}}_{\mathrm{A},n}$ are uncorrelated. Now, squaring and then taking the expectations of both sides of (5.30) results in:

$$\mathrm{E}\left\{\mathbf{d}_{\mathrm{UM},n}^2\right\} = \mathrm{E}\left\{\overline{\mathbf{x}}_n^2\right\} + \mathrm{E}\left\{\overline{\mathbf{x}}_{\mathrm{A},n}^2\right\} + \mathrm{E}\left\{\mathbf{q}_n^2\right\} \tag{5.31}$$

Moreover, according to the least perturbation property (5.8), we assume that filter coefficients converge at steady-state, thus:

$$\mathrm{E}\left\{\overline{\mathbf{x}}_n^2\right\} \approx \mathrm{E}\left\{\mathbf{y}_n^2\right\} \tag{5.32}$$

where $\mathbf{y}_n = \mathbf{X}_n \mathbf{w}_{n-1}$ is adaptive filter output signal. As a consequence,

$$\mathrm{E}\left\{\overline{\mathbf{x}}_{\mathrm{A},n}^2\right\} + \mathrm{E}\left\{\mathbf{q}_n^2\right\} = \mathrm{E}\left\{\mathbf{d}_{\mathrm{UM},n}^2\right\} - \mathrm{E}\left\{\mathbf{y}_n^2\right\}. \tag{5.33}$$

Moreover, it is possible to assume that at steady-state the noise contributions converge to the *a posteriori* error, defined in (5.4), so taking into account the energy relation (5.5) it is possible to write:

$$\begin{aligned}\mathrm{E}\left\{\overline{\mathbf{x}}_{\mathrm{A},n}^2\right\} + \mathrm{E}\left\{\mathbf{q}_n^2\right\} &\approx \mathrm{E}\left\{\boldsymbol{\varepsilon}_n^2\right\} \\ &= \left(\mathbf{I} - \boldsymbol{\alpha}_n\right)\mathrm{E}\left\{\mathbf{e}_n^2\right\}.\end{aligned} \tag{5.34}$$

Therefore, replacing (5.34) in (5.33), it is possible to derive an expression of the variable step size parameter vector:

$$\boldsymbol{\alpha}_n = \mathbf{I} - \sqrt{\frac{\mathrm{E}\left\{\mathbf{d}_{\mathrm{UM},n}^2\right\} - \mathrm{E}\left\{\mathbf{y}_n^2\right\}}{\mathrm{E}\left\{\mathbf{e}_n^2\right\}}}. \tag{5.35}$$

From a practical point of view, we evaluate the expectations in terms of power estimates, thus each diagonal element of $\boldsymbol{\alpha}_n$ can be written as:

$$\mu_l\left[n\right] = \left|1 - \frac{\sqrt{\left|\widehat{\sigma}_{\mathrm{d}}^2\left[n-l\right] - \widehat{\sigma}_{\mathrm{y}}^2\left[n-l\right]\right|}}{\widehat{\sigma}_{\mathrm{e}}^2\left[n-l\right] + \zeta}\right| \tag{5.36}$$

where $l = 0, \ldots, K-1$. Let us note that in order to make the reading clearer, in (5.36) and in the following we omit the subscript "UM" for the desired signal.

The general parameter $\widehat{\sigma}_{\theta}^{2}[n]$ represents the power estimate of the sequence $\theta[n]$, considering $\theta = \{d, y, e\}$ and can be computed as:

$$\widehat{\sigma}_{\theta}^{2}[n] = \beta\widehat{\sigma}_{\theta}^{2}[n-1] + (1-\beta)\,\theta^{2}[n] \tag{5.37}$$

where β is a forgetting factor chosen as $\beta = 1 - 1/(QM)$, with $Q > 1$. The initial value is $\widehat{\sigma}_{\theta}^{2}[0] = 0$. Furthermore, a small positive number ζ should be added in (5.37) to avoid division by zero. In order to satisfy the steady-state approximation (5.34), as suggested in [101], the process starts using a fixed step size value for the first M iterations when the estimate of the coefficients may be influenced only by the system noise $v[n]$. However, even if we do not consider this "trick", the experimental results will prove that performance degradation is not very significant, especially when the value of the projection order is increased [100]. Another practical consideration is that the computation of the power estimates in (5.36) could lead to minor deviations from the previous theoretical conditions; this is the reason why in (5.36) we consider the absolute value of the step size parameter. Nevertheless, when echo path changes occur, the power of the estimate of the echo signal $\widehat{\sigma}_{y}^{2}[n]$ may be larger than the power of the desired signal $\widehat{\sigma}_{d}^{2}[n]$. This is the reason why, in order to avoid complex values, in (5.36) we take also the absolute value of the difference under the square root.