ELSEVIER

Signal Processing 46 (1995) 15–30

SIGNAL PROCESSING

Normalization and convergence of gradient-based algorithms for adaptive IIR filters

Markus Rupp*

Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106, USA

Received 2 October 1992; revised 2 November 1994

Abstract

In the last years several algorithms for adaptive IIR filters have been proposed. However, their practical usage involves considerations such as finding the global minimum, possible occurrence of instability and uncertainty about the speed of convergence. Following a deterministic approach this paper presents a generalization of these adaptive IIR algorithms. The algorithms can be classified into two groups: those which do not and those which do filter the adaptation error. For the first group a normalization rule is presented and convergence properties assuming slow time-variant filters are given. Stronger results for general time-variant filters could only be given for a small set of algorithms. For the second group ideas of normalizations are presented and their effects for convergence are shown. Validity is proven by applying these ideas to the simplified hyperstable adaptive recursive filter (SHARF) algorithm. Considerations about special constraints for normalizations close the paper.

Zusammenfassung

In den letzten Jahre wurden verschiedene Algorithmen für adaptive IIR Filter vorgeschlagen. Ihr parktischer Einsatz leidet jedoch an Schwierigkeiten, wie dem Finden eines globalen Minimums, möglichen Instabilitäten und der ungewissen Konvergenzgeschwindigkeit. Einer deterministischen Betrachtungsweise folgend präsentiert dieser Artikel eine Vereinheitlichung der adaptiven IIR Algorithmen. Die Algorithmen können in zwei Gruppen unterteilt werden: solche ohne und mit Filterung des Fehlersignals. Für die erste Gruppe wird eine Normierungsvorschrift angegeben und Konvergenzeigenschaften, basierend auf der Annahme langsam veränderlicher Filter, angegeben. Für eine kleine Gruppe von Algorithmen konnten sogar Bedingungen für allgemeine zeitlich veränderliche Systeme gegeben werden. Für die zweite Gruppe werden Ideen für die Normierung präsentiert und ihre Auswirkungen für die Konvergenz gezeigt. Die Gültigkeit dieser Ideen wird am Beispiel des Simplified Hyperstable Adaptive Recursive Filter (SHARF) Algorithmus validiert. Überlegungen für spezielle Normierungsbedingungen beschließen den Artikel.

Résumé

Plusieurs algorithmes pour des filtres RII adaptatifs ont été proposés ces dernières années. Leur usage nécessite toutefois des considérations sur des sujets comme le fait de trouver le minimum global, l'occurence possible d'instabilités

*E-mail: markus@agni.ece.ucsb.edu.

SSDI 0165-1684(95)00069-0

et l'incertitude concernant la vitesse de convergence. Suivant une approche déterministe, cet article présente une généralisation de ces algorithmes RII adaptatifs. Les algorithmes peuvent être séparés en deux groupes: ceux qui filtrent et ceux qui ne filtrent pas l'erreur d'adaptation. Pour le premier groupe, une règle de normalisation est présentée, et leurs propriétés de convergence sont données, supposant des filtres variants lentement dans le temps. Les resultats pour les filtres variant dans le temps de manière générale n'ont pu être donnés que pour un petit ensemble d'algorithmes. En ce qui concerne le deuxieme groupe, des idées de normalisation sont présentées, et leurs effets sur la convergence montrés. Leur validité est prouvée en appliquant ces idées à l'algorithme du filtre récursif adaptatif hyperstable simplifié (FRAHS). Des considèrations sur des contraintes de normalisation spéciales closent cet article.

Keywords: Normalization; Convergence; Stability; Gradient-type algorithm; Adaptive IIR filters

1. Introduction

Since the derivation of Feintuch's algorithm (RLMS) [9] there have been several suggestions for improving the behavior of adaptive IIR filters. Some ideas like Stearns' algorithm [24] and alternate filtering mode (AFM) [7] have been dropped. Others like series-Parallel filtering (SPLMS) [1], equation error formulation (EEFLMS) [22], bias-remedy LMS (BRLMS) [14, 15] and simplified hyperstable adaptive recursive filter (SHARF) [12] remain as candidates for further research. However, all of these algorithms are rather difficult to handle when dealing with applications involving speech signals. If the filter is in the signal path, e.g. in a hybrid application, the SPLMS algorithms decorrelates the transmitted speech, worsening the quality of the signal. The EEFLMS algorithm causes a bias in the estimation of the parameter set, thus resulting in instability. The BRLMS algorithm offers to reduce the bias but not all problems with the choice of the step-size have been solved yet. Even the SHARF algorithm which is based on the concept of hyperstability cannot ensure stability in every situation.

Since hybrids are typical applications for adaptive IIR filters [18], Fig. 1 depicts an example of echo cancelation on a hybrid. The hybrid G has to decouple the near- and far-end speech signals. The near-end speaker signal $u(k)$ has to be transmitted to the subscriber side whereas the far-end speech signal $n(k)$ has to be transmitted to the near-end loudspeaker. Since the hybrid is not an ideal device, an echo of the near-end speech appears at the loudspeaker. This echo has to be estimated by a system identification of the hybrid. As illustrated in Fig. 1,

$$e_0(k) = d(k) - \hat{y}(k), \tag{1.1a}$$

$$d(k) = n(k) + y(k), \tag{1.1b}$$

$$y(k) = \sum_{i=1}^{M_a} a_i y(k-i) + \sum_{j=0}^{M_b-1} b_j u(k-j). \tag{1.1c}$$

The problem consists of finding the parametrs a_i for $i = 1, \ldots, M_a$ and b_j for $j = 0, \ldots, M_b - 1$ while observing the input signal $u(k)$ and the output signal $d(k)$ of the hybrid. For a clearer description a vector notation is often used:

$$\boldsymbol{a}^{\mathrm{T}} = [a_1, a_2, \ldots, a_{M_a}], \tag{1.2a}$$

$$\boldsymbol{b}^{\mathrm{T}} = [b_0, b_1, \ldots, b_{M_b-1}], \tag{1.2b}$$

$$\boldsymbol{w}^{\mathrm{T}} = [\boldsymbol{a}^{\mathrm{T}}, \boldsymbol{b}^{\mathrm{T}}], \tag{1.2c}$$

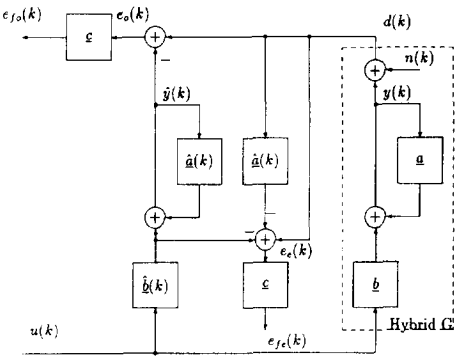

Fig. 1. Adaptive filter structure for echo cancelling.

$$\boldsymbol{y}^{\mathrm{T}}(k) = [y(k-1), y(k-2), \ldots, y(k-M_a)], \quad (1.2d)$$

$$\boldsymbol{u}^{\mathrm{T}}(k) = [u(k), u(k-1), \ldots, u(k-M_b+1)], \quad (1.2e)$$

$$\boldsymbol{z}^{\mathrm{T}}(k) = [\boldsymbol{y}^{\mathrm{T}}(k), \boldsymbol{u}^{\mathrm{T}}(k)]. \quad (1.2f)$$

The parameters for the transversal part $\boldsymbol{b}$ and the recursive part $\boldsymbol{a}$ have been combined to form a new vector $\boldsymbol{w}$ of order $M = M_a + M_b$. Now the hybrid can easily be described as

$$d(k) = n(k) + \boldsymbol{z}^{\mathrm{T}}(k)\boldsymbol{w}. \quad (1.3)$$

In a very similar way the echo canceler can be described by an estimated parameter set

$$\hat{\boldsymbol{a}}^{\mathrm{T}}(k) = [\hat{a}_1(k), \hat{a}_2(k), \ldots, \hat{a}_{M_a}(k)], \quad (1.4a)$$

$$\hat{\boldsymbol{b}}^{\mathrm{T}}(k) = [\hat{b}_0(k), \hat{b}_1(k), \ldots, \hat{a}_{M_b-1}(k)], \quad (1.4b)$$

$$\hat{\boldsymbol{w}}^{\mathrm{T}}(k) = [\hat{\boldsymbol{a}}_{\mathrm{T}}(k), \hat{\boldsymbol{b}}_{\mathrm{T}}(k)]. \quad (1.4c)$$

In order to simplify matters, M_a and M_b are the same for the plant G as well as the adaptive filter. Since the following calculations do not deal with error surfaces, the results are also true for different orders. The signal vector for calculating the estimated hybrid output $\hat{y}(k)$ can be constructed by either using former estimated values $\hat{y}(k-m)$ for $m = 1, \ldots, M_a$ or observed values $d(k-m)$ for $m = 1, \ldots, M_a$:

$$\hat{\boldsymbol{y}}^{\mathrm{T}}(k) = [\hat{y}(k-1), \hat{y}(k-2), \ldots, \hat{y}(k-M_a)], \quad (1.5a)$$

$$\boldsymbol{d}^{\mathrm{T}}(k) = [d(k-1), d(k-2), \ldots, d(k-M_a)], \quad (1.5b)$$

$$\boldsymbol{z}_{\mathrm{o}}^{\mathrm{T}}(k) = [\hat{\boldsymbol{y}}^{\mathrm{T}}(k), \boldsymbol{u}^{\mathrm{T}}(k)], \quad (1.5c)$$

$$\boldsymbol{z}_{\mathrm{e}}^{\mathrm{T}}(k) = [\boldsymbol{d}^{\mathrm{T}}(k), \boldsymbol{u}^{\mathrm{T}}(k)]. \quad (1.5d)$$

Depending on the choice of signal vector of the canceler, an output error (subscript 'o') or an equation error (subscript 'e') method is used. Now, the formula for a gradient-based algorithm can be given very generally:

$$\hat{\boldsymbol{w}}(k+1) = \hat{\boldsymbol{w}}(k) + \mu(k)e_{\mathrm{a}}(k)\boldsymbol{\psi}(k). \quad (1.6)$$

In (1.6) an adaptation error $e_{\mathrm{a}}(k) \in \{e_{\mathrm{o}}(k), e_{\mathrm{e}}(k), e_{\mathrm{c}}(k)\}$ and a gradient term $\boldsymbol{\psi}(k) \in \{\boldsymbol{z}_{\mathrm{o}}(k), \boldsymbol{z}_{\mathrm{e}}(k), \boldsymbol{z}_{\mathrm{c}}(k), \boldsymbol{z}_{\mathrm{fo}}(k), \boldsymbol{z}_{\mathrm{fe}}(k), \boldsymbol{z}_{\mathrm{fc}}(k)\}$ defining the direction of adaptation are used. The algorithms can be described according to the choice of adaptation error and gradient term. A summary of the possibilities is given in Table 1. The subscript 'c' is used for a correction term,

$$\boldsymbol{z}_{\mathrm{c}}(k) = \tau(k)\boldsymbol{z}_{\mathrm{o}}(k) + (1-\tau(k))\boldsymbol{z}_{\mathrm{e}}(k), \quad (1.7)$$

which is necessary for incorporating the BRLMS algorithm [15] within the same description scheme. The variable $\tau(k)$ lies between zero and one and its influence will be described later. Although the Steiglitz–McBride [25] algorithm is usually of Newton type, a gradient algorithm is possible as well. The same filter structure has also been used for the Filtered-u LMS algorithm, an extension to the Filtered-x LMS FIR version, which has been applied successfully in active noise cancelation [6]. Recently, two new algorithms [3, 4] being also a member of this table appeared in the literature.

Thus the adaptation error $e_{\mathrm{a}}(k)$ can be

1. the output error, $e_{\mathrm{o}}(k) = d(k) - \hat{\boldsymbol{w}}^{\mathrm{T}}(k)\boldsymbol{z}_{\mathrm{o}}(k)$,
2. the equation error, $e_{\mathrm{e}}(k) = d(k) - \hat{\boldsymbol{w}}^{\mathrm{T}}(k)\boldsymbol{z}_{\mathrm{e}}(k)$,
3. a corrected error, $e_{\mathrm{c}}(k) = d(k) - \hat{\boldsymbol{w}}^{\mathrm{T}}(k)\boldsymbol{z}_{\mathrm{c}}(k)$,

Table 1
Possibilities for gradient-based algorithms

	Gradient $\boldsymbol{\psi}(k)$					
e_{a}	$\boldsymbol{z}_{\mathrm{o}}$	$\boldsymbol{z}_{\mathrm{e}}$	$\boldsymbol{z}_{\mathrm{c}}$	$\boldsymbol{z}_{\mathrm{fo}}$	$\boldsymbol{z}_{\mathrm{fe}}$	$\boldsymbol{z}_{\mathrm{fc}}$
e_{o}	RLMS			Stearns'	AFM	
e_{e}		SPLMS, EEFLMS	BRLMS	GIVE		
e_{c}		(BRLMS1)	(BRLMS2)			
e_{fo}	SHARF			Steiglitz–McBride, FuLMS		[4]
e_{fe}						
e_{fc}						

4. a filtered output error, $e_{fo}(k) = e_e(k) + \sum_{i=1}^{P} c_i \times e_0(k-i)$,
5. a filtered equation error, $e_{fe}(k) = e_e(k) + \sum_{i=1}^{P} c_i e_e(k-i)$,
6. a filtered corrected error, $e_{fc}(k) = e_c(k) + \sum_{i=0}^{P} c_i e_c(k-i)$.

In a very similar way the gradient term $\psi(k)$ can be

1. the vector with the estimated output signal $z_0(k)$,
2. the vector with the measured output signal $z_e(k)$,
3. the corrected vector $z_c(k)$, a linear combination of $z_0(k)$ and $z_e(k)$,
4. a filtered version of $z_0(k)$: $z_{fo}(k) = z_0(k) + \sum_{i=1}^{M_a} \hat{a}_i(k) z_{fo}(k-i)$,
5. a filtered version of $z_e(k)$: $z_{fe}(k) = z_e(k) + \sum_{i=1}^{M_a} \hat{a}_i(k) z_{fe}(k-i)$,
6. a filtered version of $z_c(k)$: $z_{fc}(k) = z_c(k) + \sum_{i=1}^{M_a} \hat{a}_i(k) z_{fc}(k-i)$.

Thus, 36 different algorithms are possible and these can be divided into two groups. The first group, in the first three rows, does not filter the adaptation error. The remaining algorithms in the second group all use a filtered adaptation error. Of the 36 different algorithms, only nine (labeled in Table 1) have been thoroughly investigated. The two algorithms in parentheses (BRLMS1, BRLMS2) will be discussed later as examples.

The main topic this paper addresses is how the step-size $\mu(k)$ can be chosen to guarantee convergence for all these algorithms. The difference between convergence of the algorithm and stability of the resulting filter must here be emphasized. In contrast to transversal filters it is possible that a convergent algorithm produces an unstable IIR filter. In this case the parameter set $\hat{\boldsymbol{w}}(k)$ moves to a region in the parameter space corresponding to poles outside of the unit circle. Section 2 presents normalization rules to assure convergence of the first group. Section 3 proposes concepts for convergent algorithms for the second group and uses the example of the SHARF algorithm to illustrate how the concepts influence the convergence behavior. Section 4 validates the normalization rules for the main algorithms by results of real-time measurements. Section 5 presents additional ideas for normalizations of adaptive IIR filters with special constraints.

2. Normalization of algorithms without filtered adaptation error

Normalization has often been shown to be a useful tool in improving the behavior of adaptive algorithms. Normalization used in the LMS algorithm has demonstrated the following properties [19, 26]:

1. The algorithm is independent of the input signal level.
2. Convergence of the algorithm is guaranteed independent of the input sequence.
3. The convergence speed is increased.

These promising results provide adequate motivation to further investigate normalizations. However, since gradient-based algorithms are nonlinear a proof for convergence is required for some problems. In the past, analyses have fallen under two broad categories for the input sequences: one deterministic, the other stochastic. Although the deterministic description leads to more general results, often it is not possible to prove convergence, and, therefore, stochastic methods are required. In this paper, however, a pure deterministic description is presented leading to normalization rules that lead to convergence conditions for a class of algorithms.

2.1. A deterministic eigenvalue analysis

For the class of algorithms without a filtered update error, the adaptation rule can be given as follows:

$$\hat{\boldsymbol{w}}(k+1) = \hat{\boldsymbol{w}}(k) + \mu(k) e_a(k) \boldsymbol{z}_1(k), \tag{2.1a}$$

$$e_a(k) = e_2(k) = d(k) - \boldsymbol{z}_2^{\mathrm{T}}(k) \hat{\boldsymbol{w}}(k). \tag{2.1b}$$

Here, two different vectors $\boldsymbol{z}_1(k)$ and $\boldsymbol{z}_2(k)$ are used. The gradient term $\boldsymbol{z}_1(k) \in \{z_0(k), z_e(k), z_c(k), z_{fo}(k), z_{fe}(k), z_{fc}(k)\}$ can be one of the six vectors described in the last section, whereas $\boldsymbol{z}_2(k) \in \{z_0(k), z_e(k), z_c(k)\}$ is unfiltered. A minimal error $e_m(k)$ is introduced:

$$e_m(k) = d(k) - \boldsymbol{z}_2^{\mathrm{T}}(k) \boldsymbol{w}. \tag{2.2}$$

This is the adaptation error[1] when using the optimal solution $\boldsymbol{w}$ for a given vector $\boldsymbol{z}_2(k)$. The

[1] The concept of a conditional minimal error $e_m(k)$ should not be confused with minimizing the performance index $E[e_0^2(k)]$.

adaptation error can therefore be rewritten in terms of the weight-error vector $\varepsilon(k) = \hat{\boldsymbol{w}} - \boldsymbol{w}(k)$ as

$$e_a(k) = e_m(k) - \varepsilon^T(k)\boldsymbol{z}_2(k). \tag{2.3}$$

It is now possible to describe the weight-error vector $\varepsilon(k)$ as an inhomogeneous system of first order:

$$\begin{aligned}\varepsilon(k+1) &= \varepsilon(k) + \mu(k)d(k)\boldsymbol{z}_1(k) \\ &\quad - \mu(k)\boldsymbol{z}_1(k)\boldsymbol{z}_2^T(k)\hat{\boldsymbol{w}}(k) \\ &= (\boldsymbol{I}_{M\times M} - \mu(k)\boldsymbol{z}_1(k)\boldsymbol{z}_2^T(k))\varepsilon(k) \\ &\quad + \mu(k)e_m(k)\boldsymbol{z}_1(k),\end{aligned} \tag{2.4}$$

where $\boldsymbol{I}_{M\times M}$ denotes the identity matrix of dimension M. The objective of finding conditions for convergence is now related to the eigenvalues of the transition matrix $T(k) = \boldsymbol{I}_{M\times M} - \mu(k)\boldsymbol{z}_1(k)\boldsymbol{z}_2^T(k)$. For time-invariant transition matrices it is sufficient to show that no eigenvalues are outside the unit disc in order to ensure convergence. We shall follow this idea even though the transition matrix is time-varying now. If the transition matrix is slowly time-varying, additional to the condition on the eigenvalues, it is required that $\|T(k) - T(k-1)\| < c$ [23]. All the conditions that have to be satisfied are:

1. No eigenvalue of $T(k)$ lies outside the unit disc.
2. $T(k)$ is bounded, i.e. $\|T(k)\| < \infty$.
3. $T(k)$ is slowly time-varying, i.e. $\|T(k) - T(k-1)\| < c$.

Thus, we assume the transition matrix to alter very slowly, which is true either for small step-sizes or strongly correlated input sequences. The eigenvalue analysis of the transition matrix leads to one time-varying eigenvalue

$$\lambda(k) = 1 - \mu(k)\boldsymbol{z}_1^T(k)\boldsymbol{z}_2(k) \tag{2.5}$$

corresponding to the right (left) eigenvector $\boldsymbol{z}_1(k)$ ($\boldsymbol{z}_2(k)$). The remaining $M-1$ eigenvalues equal one (geometric multiplicity of order $M-1$) corresponding to the $M-1$ linear independent eigenvectors that are orthogonal to the right (left) eigenvector $\boldsymbol{z}_1(k)$ ($\boldsymbol{z}_2(k)$). In order to bound the time-varying eigenvalue $\lambda(k)$, a first hint for a normalization is revealed,

Normalization 1:

$$\mu(k) = \frac{\alpha}{\boldsymbol{z}_1^T(k)\boldsymbol{z}_2(k)}, \tag{2.6}$$

resulting in a constant eigenvalue $\lambda = 1 - \alpha$. Thus, no instantaneous eigenvalues of the system are outside the unit circle for the normalized step-size α between zero and two. Obviously, the smaller the angle between the two vectors $\boldsymbol{z}_1(k)$ and $\boldsymbol{z}_2(k)$, the smaller the step-size $\mu(k)$. The concept of having the eigenvalues inside the unit disc is equivalent to the requirement that the corrupted a posteriori error is smaller than the a priori error:

$$|e_a(k, |\varepsilon(k+1))| < |e_a(k)|.$$

Plugging in Normalization 1 into definition (2.3) for $e_a(k)$ we obtain

$$e_a(k, |\varepsilon(k+1)) = (1-\alpha)e_a(k),$$

and, therefore, the a posteriori error becomes smaller than the a priori error for α between zero and two.

Although the eigenvalues remain bounded, condition 3 for slowly time-varying systems may not be satisfied, since $\alpha\|(\boldsymbol{z}_1(k)\boldsymbol{z}_2^T(k)/\boldsymbol{z}_1^T(k)\boldsymbol{z}_2(k)) - (\boldsymbol{z}_1(k-1)\boldsymbol{z}_2^T(k-1)/\boldsymbol{z}_1^T(k-1)\boldsymbol{z}_2(k-1))\|$ is not necessarily bounded. Also even if the homogeneous system may cause a decreasing weight-error vector, the driving term can increase $\varepsilon(k)$. The squared L_2-norm of the perturbation in (2.4) with Normalization 1 is considered:

$$\alpha^2 e_m^2(k)\frac{\boldsymbol{z}_1^T(k)\boldsymbol{z}_1(k)}{(\boldsymbol{z}_1^T(k)\boldsymbol{z}_2(k))^2} \geqslant \alpha^2 e_m^2(k)\frac{1}{\boldsymbol{z}_2^T(k)\boldsymbol{z}_2(k)}. \tag{2.7}$$

Therefore, problems can occur if $\boldsymbol{z}_1^T(k)\boldsymbol{z}_2(k) = 0$ or if $\|\boldsymbol{z}_2(k)\|_2$ is not bounded from below. If we assume persistent excitation, i.e. $\|\boldsymbol{z}_2(k)\|_2 > 0$, the latter is not a problem; if not, adding a positive constant in the denominator does not help since the inner product of $\boldsymbol{z}_1^T(k)\boldsymbol{z}_2(k)$ can become negative. Therefore, this normalization can cause the inhomogeneous part exceeding every limit, even if $\|\boldsymbol{z}_1(k)\|_2$ and $\|\boldsymbol{z}_2(k)\|_2$ are very large. The more orthogonal the vectors are, the more emphasized the effect of the driving term; even in the case $\boldsymbol{z}_1(k) = \boldsymbol{z}_2(k)$, there can be an amplification. In a stochastic approach, the strength of this effect depends on the statistics of the input process [26, 19]. An improvement for the undesired amplification can be achieved by using the step-size $\mu(k) = \alpha/(c + \boldsymbol{z}_1^T(k)\boldsymbol{z}_1(k))$ since the inhomogeneous

part is now bounded by the positive constant c. However, if $z_1^T(k)z_2(k) < 0$ the eigenvalue

$$\lambda(k) = 1 - \alpha \frac{z_1^T(k)z_2(k)}{c + z_1^T(k)z_1(k)}$$

will become larger than one for every positive α. Therefore, this idea is not further considered.

2.2. Normalization Rule 2

Since Normalization Rule 1 leads to difficulties for the disturbance term, another rule has to be found. Applying Schwarz's inequality it can be shown that

$$\left| \frac{z_1^T(k)z_2(k)}{z_1^T(k)z_1(k)z_2^T(k)z_2(k)} \right| \leq \left| \frac{1}{z_1(k)z_2^T(k)} \right| . \tag{2.8}$$

Since both terms have the same sign, the left term can be used instead of the right one. This leads to

Normalization 2:

$$\mu(k) = \frac{\alpha z_1^T(k)z_2(k)}{z_1^T(k)z_1(k)z_2^T(k)z_2(k)} . \tag{2.9}$$

Because of (2.8) Normalization 2 is always lower than or equal to Normalization 1 and hence also satisfies the condition that no eigenvalue of the homogeneous part of (2.4) is outside the unit disc. Under the assumption of a time-invariant transition matrix convergence is again guaranteed for $0 < \alpha < 2$. However, for slowly time-varying systems the step-size has now to satisfy the condition $\|T(k) - T(k-1)\| < c$. Thus,

$$\alpha \left\| \frac{z_1^T(k)z_2(k)z_1(k)z_2^T(k)}{z_1^T(k)z_1(k)z_2^T(k)z_2(k)} - \frac{z_1^T(k-1)z_2(k-1)z_1(k-1)z_2^T(k-1)}{z_1^T(k-1)z_1(k-1)z_2^T(k-1)z_2(k-1)} \right\| < c.$$

This condition can be satisfied for every vector $z_1(k)$, $z_2(k)$ if α is small enough.

Furthermore, Normalization 2 also bounds the inhomogeneous system (2.4) for lower bounded vectors $z_2(k)$ which can be proved by the Schwarz's inequality as well. The squared L_2-norm of the inhomogeneous part is considered again:

$$\alpha^2 e_m^2(k) \frac{(z_1^T(k)z_2(k))^2}{z_1^T(k)z_1(k)(z_2^T(k)z_2(k))^2} \leq \alpha^2 e_m^2(k) \frac{1}{z_2^T(k)z_2(k)} . \tag{2.10}$$

Eq. (2.10) shows that the driving term is now bounded as long as $\|z_2(k)\|_2$ is lower bounded. In the case of nonpersistent excitation, adding a small positive constant to the denominator provides boundedness again. Although normalization can guarantee convergence of the homogeneous equation, the inhomogeneous part can be amplified and, thus, the steady-state error can increase. Based on the statistics of the input signal [26, 19] this effect can be quantified.

The various normalization terms for the known algorithms are listed in Table 2. Having no eigenvalue of the update system outside the unit circle is certainly a desirable property. However, as mentioned before, while dealing with general time-varying systems, this property does not guarantee convergence.

Table 2
Possible normalizations for gradient-based algorithms without a filtered adaptation error

Algorithm	Normalization 1	Normalization 2
Feintuch	$\frac{1}{z_o^T(k)z_o(k)}$	$\frac{1}{z_o^T(k)z_o(k)}$
Stearns	$\frac{1}{z_{fo}^T(k)z_o(k)}$	$\frac{z_{fo}^T(k)z_o(k)}{z_o^T(k)z_o(k)z_{fo}^T(k)z_{fo}(k)}$
AFM	$\frac{1}{z_{fe}^T(k)z_o(k)}$	$\frac{z_{fe}^T(k)z_o(k)}{z_o^T(k)z_o(k)z_{fe}^T(k)z_{fe}(k)}$
SPLMS, EEFLMS	$\frac{1}{z_e^T(k)z_e(k)}$	$\frac{1}{z_e^T(k)z_e(k)}$
BRLMS	$\frac{1}{z_c^T(k)z_e(k)}$	$\frac{z_c^T(k)z_e(k)}{z_e^T(k)z_e(k)z_c^T(k)z_c(k)}$

2.3. Contraction mapping

It is possible that time-varying systems are not convergent although no eigenvalue is outside of the unit circle at any time. The previous approach of calculating the decisive eigenvalue certainly gives good hints but not a strict proof for convergence. In order to lead to a convergent algorithm the homogeneous part of (2.4) must describe a contraction mapping. The mapping operator

$$T(k) = \boldsymbol{I}_{M\times M} - \mu(k)\boldsymbol{z}_1(k)\boldsymbol{z}_2^{\mathrm{T}}(k) \tag{2.11}$$

is called a contraction operator, if for every $\boldsymbol{\varepsilon}(k)$ and an arbitrary norm

$$\|T(k)[\boldsymbol{\varepsilon}(k)]\| = \gamma(k)\,\|\boldsymbol{\varepsilon}(k)\| \tag{2.12}$$

the Lipschitz constant $\gamma(k) < 1$ for every time instant k. The question arises whether it is possible to find a step-size $\mu(k)$ in order to set $\gamma(k) < 1$ without a knowledge of the error vector $\boldsymbol{\varepsilon}(k)$. Because of the Schwarz's inequality two constants $-1 < K_i(k) < 1$ exist, such that

$$K_i(k)\sqrt{\boldsymbol{\varepsilon}^{\mathrm{T}}(k)\boldsymbol{\varepsilon}(k)\boldsymbol{z}_i^{\mathrm{T}}(k)\boldsymbol{z}_i(k)} = \boldsymbol{\varepsilon}^{\mathrm{T}}(k)\boldsymbol{z}_i(k)$$
$$\text{for } i = 1, 2. \tag{2.13}$$

If the L_2-norm is applied, $\gamma(k)$ can be interpreted as the largest singular value of $T(k)$, i.e. $\max_{\boldsymbol{\varepsilon}(k)}\gamma(k) = \|T(k)\|_{2,\mathrm{ind}}$. Plugging (2.11) and (2.13) into (2.12), the following is obtained for $\gamma(k)$:

$$\gamma^2(k) = 1 - 2\mu(k)K_1(k)K_2(k)\sqrt{\boldsymbol{z}_1^{\mathrm{T}}(k)\boldsymbol{z}_1(k)\boldsymbol{z}_2^{\mathrm{T}}(k)\boldsymbol{z}_2(k)} + \mu^2(k)K_2^2(k)\boldsymbol{z}_1^{\mathrm{T}}(k)\boldsymbol{z}_1(k)\boldsymbol{z}_2^{\mathrm{T}}(k)\boldsymbol{z}_2(k). \tag{2.14}$$

Both $K_1(k)$ and $K_2(k)$ depend on the unobservable weight-error vector $\boldsymbol{\varepsilon}(k)$ and their product $K_1(k)K_2(k)$ can have the opposite polarity of the step-size $\mu(k)$ not leading to a contraction operator. Only if $\boldsymbol{z}_1(k) = \boldsymbol{z}_2(k)$ it follows $K_1(k) = K_2(k)$ for every time instant k and the result is

$$\gamma^2(k) = 1 - K_1^2(k)\boldsymbol{z}_1^{\mathrm{T}}(k)\boldsymbol{z}_1(k) \times(2\mu(k) - \mu^2(k)\boldsymbol{z}_1^{\mathrm{T}}(k)\boldsymbol{z}_1(k)). \tag{2.15}$$

Since both vectors $\boldsymbol{z}_1(k)$ and $\boldsymbol{z}_2(k)$ are now identical, both normalization rules coincide and assure contraction for $\alpha \in [0, 2]$. Therefore, only three (symmetric) algorithms are convergent for every arbitrary input sequence: RLMS, SPLMS and BRLMS2 (see Table 1). The convergence of all other algorithms depends strongly on the statistics of the driving signal $u(k)$. Given the joint statistics of $\boldsymbol{z}_1(k)$, $\boldsymbol{z}_2(k)$ the expectation $E[\gamma(k)]$ can be calculated, resulting in contraction in the mean. Thus, the contraction mapping concept leads to conditions on the largest singular value, whereas the concept of slowly time-varying systems and time-invariant systems require conditions on the largest eigenvalue. A more rigorous treatment of gradient-type algorithms including the noise can be found in [21] and extensions to adaptive IIR filters in [20].

2.4. Examples

Some examples might emphasize the advantages of Normalization 2. Fig. 2 depicts the first example. A system to be identified is specified by three poles ($M_a = 3$, $M_b = 5$) and is taken from a measured hybrid. The transfer function is

$$G(z) = \frac{0.315 - 1.798z^{-1} - 0.365z^{-2} + 3.06z^{-3} + 0.1z^{-4}}{1 - 1.186z^{-1} - 0.185z^{-2} + 0.446z^{-3}} \tag{2.16}$$

having one complex pole pair. First, the Normalized Stearns (NStearns) algorithm is tested. Its normalized step-size α varies from 0.2 to 1.8 in steps of 0.4. In the figure averaged sample functions of the relative system mismatch $S_{\mathrm{rel}}(k) = E[\boldsymbol{\varepsilon}^{\mathrm{T}}(k)\boldsymbol{\varepsilon}(k)/\boldsymbol{\varepsilon}^{\mathrm{T}}(0)\boldsymbol{\varepsilon}(0)]$ are shown. The depicted curves are sample averages for 500 trials. If instability of the filter occurred for a longer time, the sample function was dropped. The driving process was a white process with a modified Bessel function K_0 pdf which closely resembles speech signals [2]. This particular density function is given as (see [11, p. 958, no. 8.432.3])

$$f_u(x) = \frac{1}{\pi}\int_1^\infty \frac{\mathrm{e}^{-|x|t}}{\sqrt{t^2-1}}\,\mathrm{d}t.$$

In Fig. 2 a steady-state noise with a variance of -20 dB has disturbed the signal. Its variance has been increased to $+20$ dB for 2000 steps (= 1/4 s for an 8 kHz sampling rate), as can be seen in the

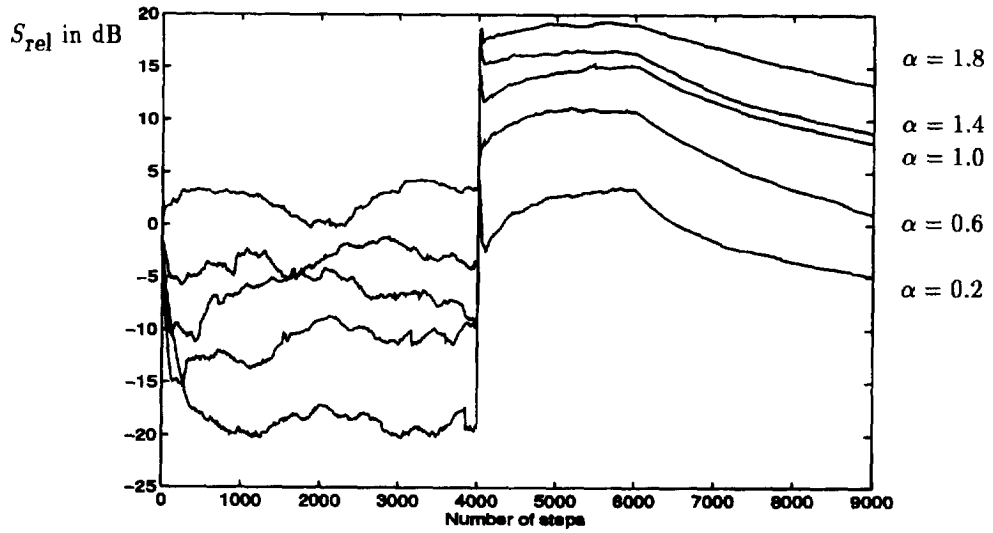

Fig. 2. Relative system mismatch of NStearns algorithm. The curves are associated with different step-sizes α.

figure. The fastest convergence is reached for an $\alpha = 0.2$. Even for $\alpha = 1.8$ convergence is obtained. For $\alpha > 2$ the algorithm shows instability as expected. After a strong perturbation the algorithm does not show the same convergence behavior as in the initial condition case. Here, the example shows that convergence can be obtained even for the Stearns' algorithm, which often shows instability, when not using normalization. However, it should be noted that the filter itself resulting from these adaptation processes was often unstable. Only the algorithm remained convergent.

A second example is the BRLMS algorithm. As already shown in Table 1 there exist three BRLMS-like algorithms. Only the one entitled BRLMS has already been published in the literature. In [14, 15] it has been shown that under certain conditions, the algorithm behaves in a stable manner, but the resulting filter was not proved to be stable as well. Normalization Rule 2 has been applied to the three algorithms obtaining Normalized BRLMS (NBRLMS, NBRLMS1, NBRLMS2) algorithms. The rule for the variable $\tau(k)$ has been chosen as

$$\tau(k) = \max\left(0, 1 - \frac{\sum_{i=1}^{M_a} d^2(k-i)}{\sum_{i=1}^{M_a} e_o^2(k-i)}\right). \tag{2.17}$$

When the disturbance is small then the quotient of the $d(k)$ and $e_o(k)$ expression will be small in the beginning and larger when the adaptation causes a decreasing weight-error vector. If suddenly a larger disturbance occurs, the quotient is again small and the variable $\tau(k)$ large. This behavior is depicted in Fig. 3 as well as the relative system mismatch for the NBRLMS2 algorithm. All three algorithms have been tested under the same conditions as those in the Stearns example. While NBRLMS and NBRLMS2 both showed good results, the NBRLMS1 version showed worse properties. The adaptation process for NBRLMS1 removed only small parts of the echos, whereas NBRLMS and NBRLMS2 showed a strong echo cancelation effect. The depicted curves are sample averages for 500 runs. The number of unstable filter results was slightly smaller for NBRLMS2 (11) than for the other two algorithms (NBRLMS:44 and NBRLMS1:55). Under these circumstances all NBRLMS-like algorithms behaved at least in a convergent manner.

3. Normalization for algorithms with filtered adaptation error

Although it was possible to show the efficiency of Normalization 2 in the case of algorithms with unfiltered adaptation error, the second group of

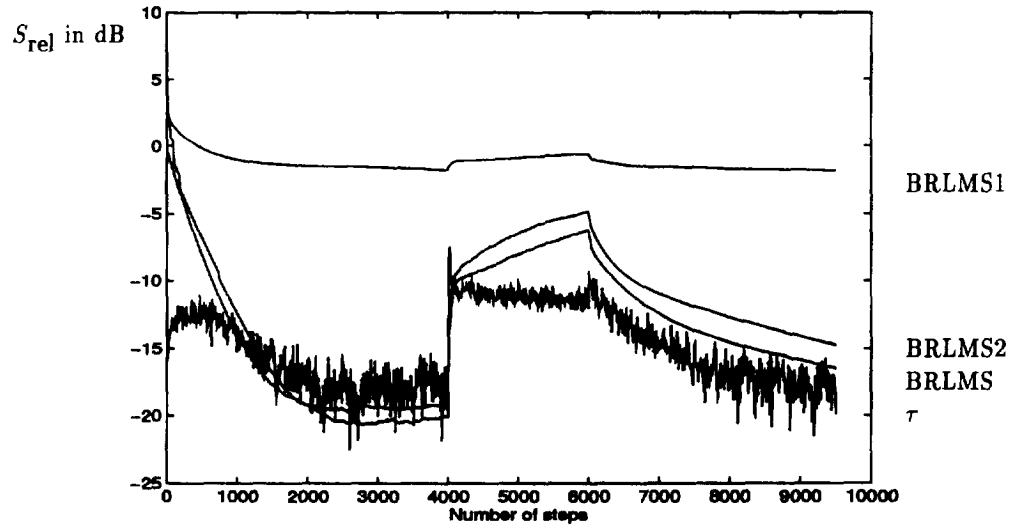

Fig. 3. Relative system mismatch of NBRLMS algorithms using $\alpha = 0.1$.

algorithms is more difficult to handle. As in the previous section the state-space approach will be used to describe the situation. With a filtered error the adaptation equation can be written as

$$\hat{\boldsymbol{w}}(k+1) = \hat{\boldsymbol{w}}(k) + \mu(k) e_a(k) \boldsymbol{z}_1(k), \tag{3.1a}$$

$$e_a(k) = C[e_2(k)] \tag{3.1b}$$

$$= e_2(k) + \sum_{i=1}^{P} c_i e_2(k-i). \tag{3.1c}$$

where the error $e_2(k)$ is the same as in (2.1b). Describing the situation with the weight-error vector $\boldsymbol{\varepsilon}(k)$ and using $e_2(k) = d(k) - \hat{\boldsymbol{w}}^T(k)\boldsymbol{z}_2(k)$ $= e_m(k) - \boldsymbol{\varepsilon}^T(k)\boldsymbol{z}_2(k)$ a vector differential system or order P is obtained:

$$\boldsymbol{\varepsilon}(k+1) = \boldsymbol{\varepsilon}(k) + \mu(k)\boldsymbol{z}_1(k)\left(e_2(k) + \sum_{i=1}^{P} c_i e_2(k-i)\right) \tag{3.2a}$$

$$= (\boldsymbol{I}_{M\times M} - \mu(k)\boldsymbol{z}_1(k)\boldsymbol{z}_2^T(k))\boldsymbol{\varepsilon}(k) - \mu(k)\boldsymbol{z}_1(k)\sum_{i=1}^{P} c_i \boldsymbol{z}_2^T(k-i)\boldsymbol{\varepsilon}(k-i) + \mu(k)\boldsymbol{z}_1(k)\left(e_m(k) + \sum_{i=1}^{P} c_i e_m(k-i)\right). \tag{3.2b}$$

The first line of (3.2b) shows the homogeneous equation and the second line describes the disturbance term. Treating such a system is very difficult. In the literature it is common to simplify the situation by assuming only slow changes in $\boldsymbol{\varepsilon}(k)$ for small step-sizes $\mu(k)$, resulting approximately in a first-order system again. But as (3.2b) shows there is indeed a system of higher order and thereby different dynamic behavior occurs. The possible difference will be demonstrated later in this section on the example of the SHARF algorithm. In order to clarify the effect of the higher order, the simplest case, being the homogeneous solution for a system of order two ($P = 1$), is investigated:

$$\begin{pmatrix} \boldsymbol{\varepsilon}(k+1) \\ \boldsymbol{\varepsilon}(k) \end{pmatrix} = \begin{pmatrix} \boldsymbol{I}_{M\times M} - \mu(k)\boldsymbol{z}_1(k)\boldsymbol{z}_2^T(k) & -c_1\mu(k)\boldsymbol{z}_1(k)\boldsymbol{z}_2^T(k-1) \\ \boldsymbol{I}_{M\times M} & \boldsymbol{0}_{M\times M} \end{pmatrix} \times \begin{pmatrix} \boldsymbol{\varepsilon}(k) \\ \boldsymbol{\varepsilon}(k-1) \end{pmatrix}. \tag{3.3}$$

For this equation only two eigenvalues $\lambda_{1,2}(k)$ unequal to one or zero are of interest. They can be investigated for the two eigenvectors $(\lambda_1(k)\boldsymbol{z}_1^T(k),$

$z_1^T(k)$) and $(\lambda_2(k)z_1^T(k), z_1^T(k))$. The corresponding characteristic equation is given by

$$\lambda^2(k) - \lambda(k)(1 - \mu(k)z_2^T(k)z_1(k)) + c_1\mu(k)z_2^T(k-1)z_1(k) = 0. \tag{3.4}$$

The solution of this reads

$$\lambda_{1,2}(k) = \frac{1 - \mu(k)z_2^T(k)z_1(k)}{2} \pm \sqrt{\left(\frac{1 - \mu(k)z_2^T(k)z_1(k)}{2}\right)^2 - c_1\mu(k)z_2^T(k-1)z_1(k)}. \tag{3.5}$$

If Normalization Rule 1 for the first-order part is applied, a simpler expression is obtained:

$$\lambda_{1,2}(k) = \frac{1 - \alpha}{2} \pm \sqrt{\left(\frac{1 - \alpha}{2}\right)^2 - c_1\alpha\delta(k)}, \tag{3.6a}$$

$$\delta(k) = \frac{z_2^T(k-1)z_1(k)}{z_2^T(k)z_1(k)}. \tag{3.6b}$$

Obviously, the term $\delta(k)$ changes with time depending on the data. Due to its value the coefficient c_1 amplifies or damps the effect of $\delta(k)$. If Normalization 2 is used instead of Rule 1, the fluctuations of $\delta(k)$ become smaller. Since the product $\tilde{c} = c_1\delta(k)$ influences the eigenvalues, we shall focus on its effect. The two decisive eigenvalues $\lambda_{1,2}$ have been computed for $\tilde{c}$ running from negative to positive values. The result of this can be seen in Table 3 and Fig. 4. When $\tilde{c}$ runs from -1 to zero both eigenvalues go inwards the unit circle. For $\tilde{c} = 0$ one eigenvalue equals zero and the second one $1 - \alpha$. When $\tilde{c} > 0$ both eigenvalues come closer until at $\tilde{c} = (1 - \alpha)^2/4\alpha$ both eigenvalues coincide. If $\tilde{c}$ is now further increased, the eigenvalues become complex. Their real part remains constant $(1 - \alpha)/2$ and only the imaginary part becomes bigger for increasing $\tilde{c}$. Eventually, for $\tilde{c} = 1/\alpha$ the unit circle is reached. Thus, the limits are

$$-1 < c_1\delta(k) < \frac{1}{\alpha}. \tag{3.7}$$

Depending on the data, the term $\delta(k)$ causes a change in $c_1\delta(k)$ with every time instant k. Therefore, there will not be a time-constant eigenvalue pair $\lambda_{1,2}(k)$ but there is a continuous movement around an average point. The solution in (3.6a) suggests to normalize the filter coefficient c_1 as well, in order to avoid this undesired time dependency. Unfortunately, the inhomogeneous part is affected as well and will cause increased disturbance.

If both vectors $z_1(k)$ and $z_2(k)$ are equal to $z_o(k)$ the SHARF algorithm is obtained. The time-varying term $\delta(k)$ now reads

$$\delta(k) = \frac{z_o^T(k-1)z_o(k)}{z_o^T(k)z_o(k)}. \tag{3.8}$$

Table 3
Decisive eigenvalues $\lambda_{1,2}$ as a function of $\tilde{c}$

$\tilde{c}$	$\lambda_{1,2}$
-1	$\{-\alpha, 1\}$
0	$\{0, 1-\alpha\}$
$\frac{(1-\alpha)^2}{4\alpha}$	$\left\{\frac{1-\alpha}{2}, \frac{1-\alpha}{2}\right\}$
$\frac{1}{\alpha}$	$\frac{1-\alpha}{2} \pm \frac{\sqrt{\alpha^2 - 2\alpha - 3}}{2}$

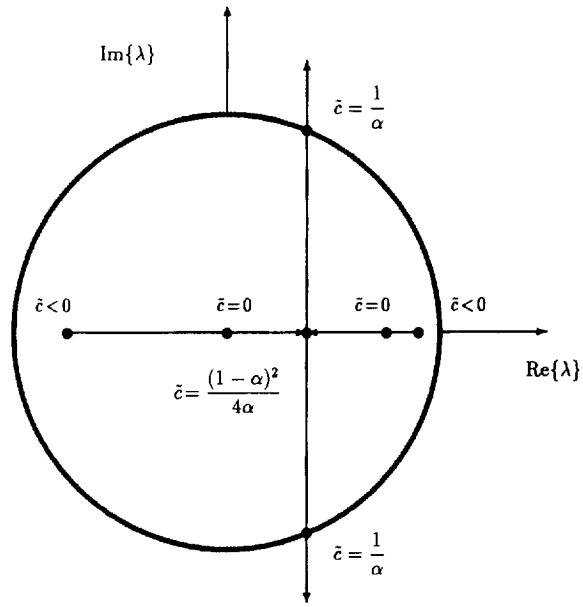

Fig. 4. Root loci for eigenvalues $\lambda_{1,2} = ((1 - \alpha/2) \pm \sqrt{((1 - \alpha^2/2))^2 - \tilde{c}\alpha}$ as a function of the filter coefficient $\tilde{c}$.

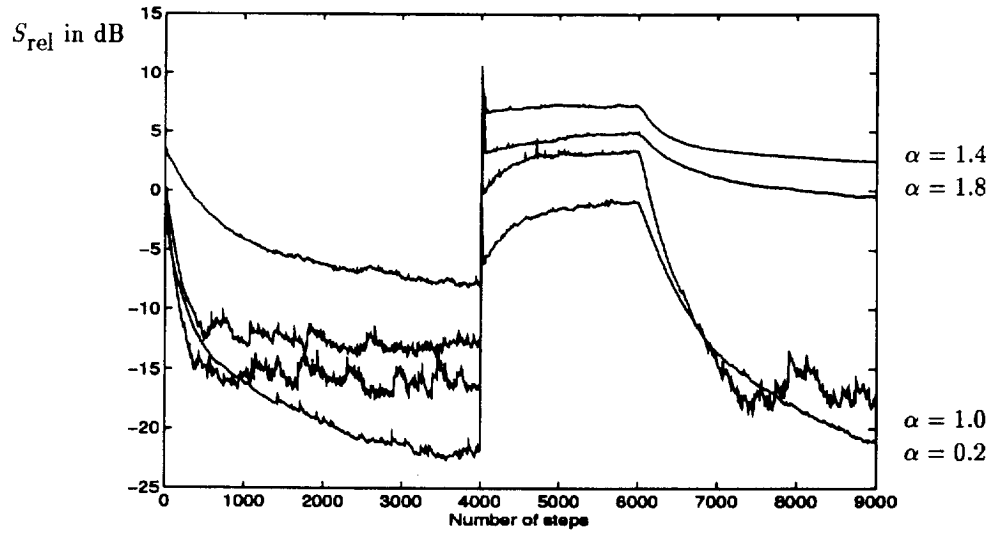

Fig. 5. Relative system mismatch for the NSHARF algorithm using Normalization 1 (= Normalization 2) and $C(z) = 1 - 0.9z^{-1}$.

This term shows some boundedness properties. Applying Schwarz's inequality again leads to

$$\left(\frac{z_o^T(k-1)z_o(k)}{z_o^T(k)z_o(k)}\right)^2 \leqslant \frac{z_o^T(k-1)z_o(k-1)}{z_o^T(k)z_o(k)} \approx 1. \quad (3.9)$$

Since the length of the vector $z_o(k)$ does not change much from one step to the next, the absolute value of the time-varying term is approximately bounded by one: $|\delta(k)| \leqslant 1$. Incorporating (3.7) and in order to be on the safe side for reasonable step-sizes $\alpha \leqslant 1$ the filter constant should be $|c_1| < 1$. Simulations for the SHARF algorithm have in fact shown that the time-varying term $\delta(k)$ moves around 1 for negative c_1, but moves around -1 for positive c_1, thus keeping the product $c_1\delta(k)$ always negative. The algorithm seems to push the poles back to position in the unit circle. Another extreme is obtained, if the constant $c_1 = 0$. Then, the SHARF algorithm simplifies to Feintuch's algorithm. The main effect of a filter constant $c_1 \neq 0$ is a second eigenvalue being unequal to zero which enables the algorithm to increase the convergence speed.

Fig. 5 depicts a simulation example. The Normalized SHARF[2] algorithm (NSHARF) with only one coefficient ($c_1 = -0.9$) is used. The situation is the same as in the previous simulation examples. The algorithm with a fixed step-size μ reached only 5 dB after 4000 iterations and was very sensitive to additive noise. The small step-size assumption resulting in a first-order system ($P = 0$) is not very helpful here. In this case there is only one eigenvalue decisive for the convergence: $\lambda(k) = 1 - \mu(k)z_o^T(k)(z_o(k) + c_1 z_o(k-1))$. If in this case the Normalization Rule 1 was used, the maximal convergence speed was low and the limit for convergence had already been obtained for small α. Also Normalization Rule 2 was checked for this case resulting in a relatively small convergence speed. The higher-order model ($P = 1$), however, and the Normalization Rule 1 chosen for it,

$$\mu(k) = \frac{\alpha}{z_o^T(k)z_o(k)},$$

showed much improvement. The normalized step-size α varies from 0.2 to 1.8 in steps of 0.4. The case $\alpha = 0.6$ is dropped, since the curve coincides with the $\alpha = 1.0$ case. The fastest convergence has been achieved for $\alpha = 1$. Even for $\alpha = 1.8$ convergence has been obtained, but for $\alpha > 2$ the algorithm showed instability as expected from (3.6a). This example shows that there is indeed a lot of room to

[2] Since $z_1(k) = z_2(k)$, both normalization rules coincide for SHARF.

improve the convergence speed of already existing algorithms by a careful choice of normalization.

The case of higher system orders $P > 1$ can be handled in a similar way. The corresponding eigenvectors to the eigenvalues that are decisive for convergence are $(\lambda_i^P(k)\mathbf{z}_1^T(k), \lambda_i^{P-1}(k)\mathbf{z}_1^T(k), \ldots, \mathbf{z}_1^T(k))$ for $i = 1, \ldots, P+1$. The characteristic polynomial equation has to be solved:

$$\lambda^{P+1}(k) + \lambda^P(k)(\mu(k)\mathbf{z}_2^T(k)\mathbf{z}_1(k) - 1) + \sum_{i=1}^{P} \lambda^{P-i}(k) c_i \mu(k) \mathbf{z}_2^T(k-i)\mathbf{z}_1(k) = 0. \quad (3.10)$$

It is to be expected that for larger delays i the expression $\mathbf{z}_2^T(k-i)\mathbf{z}_1(k)$ will become increasingly smaller, and, therefore, the parts with a larger index lose importance.

4. Measurement results

The most important algorithms in Table 1 have been implemented on a DSP56001 fixed-point signal processor. The plant to identify was a hybrid in a real 'PBX'. The situation has already been depicted in Fig. 1. In order to find the best normalized step-size α for Normalization Rule 1, the algorithms were driven by speech signals in a first step. The step-size was increased until the steady-state error began to sound disturbing. After that, for measurements the input signal $u(k)$ on the local speaker side was a white Gaussian noise sequence, while the disturbance $n(k)$ was the noise from the PBX. The signal-to-noise ratio was measured to be about 30 dB. For all IIR filters nine transversal part coefficients ($M_b = 9$) and three ($M_a = 3$) recursive part coefficients were used. A 32-tap transversal filter with NLMS algorithm is given as a comparison.

Table 4 depicts the results. The echo return loss enhancement (ERLE) was measured after 166 ms in order to given an impression of the convergence speed. A second value gives the ERLE when the steady-state error has been reached. The column labeled 'After-burst' describes the ERLE reaction after a short burst arising from the subscriber side. The AFM algorithm with the normalization from [8] as well as with the new normalization showed

Table 4
Measured results with Normalization Rule 2

Algorithm	α	$ERLE_{166}$ (dB)	$ERLF_\infty$ (dB)	After-burst
AFM	2^{-7}	10	17	17 dB after 500 ms
NAFM	0.25	4	13	13 dB after 500 ms
NStearns	1	11	15	18 dB direct
NRLMS	1	22	23	23 dB direct
NSHARF	1	22	23	23 dB direct
NLMS	0.5	31	31	31 dB direct

undesirable behavior. For both versions the convergence speed was low and the reaction after the burst was very slow. Generally speaking, both prefiltered algorithms, AFM and Stearns, did not show very good behavior. Even the steady-state error is bigger than those of the other algorithms although the same filter order was used. NSHARF as well as NRLMS showed about the same behavior. Both algorithms behaved fast and reached 23 dB for steady-state. The values given here are measures from one typical situation. There were also situations where the IIR filters yield the same steady-state as the 32-tap transversal filter. But even when the 32-tap transversal filter has lower steady-state error, it costs more in terms of computational complexity. As mentioned above all the algorithms were also checked with speech signals in order to assure that no unexpected effects occur.

5. Other possibilities for normalization

In this section alternative normalizations are considered. As already mentioned, normalizations usually serve to make the algorithms independent of changes in the input level. Since the convergence behavior depends on the unknown system itself, it is of interest to make the algorithm independent of the system as well. A first step in this direction is to make the algorithm independent of the system gain additionally to its independence on the signal level. Fig. 6 depicts the situation. On the one hand, an algorithm has to be independent of the input signal level. This can be described by an input signal $u_1(k)$ with constant level one and a multiplication factor

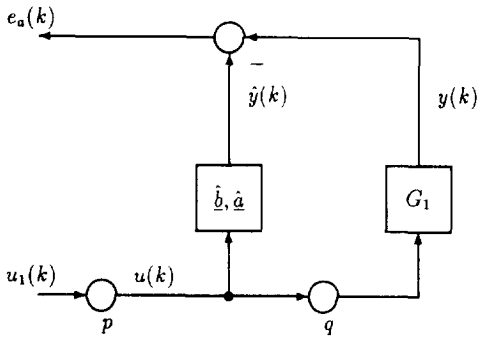

Fig. 6. Model for changes in the signal level p and system gain q

p that denotes the level, i.e. $u(k) = pu_1(k)$. On the other hand, an algorithm should be independent of different system gains. Thus, we assume the system G_1 to have unit gain and a multiplicative factor q describes the gain of the system. In terms of the plant parameters $\boldsymbol{a}$ and $\boldsymbol{b}$ the gain q influences only the transversal part coefficients $\boldsymbol{b}$ but not the recursive part coefficients $\boldsymbol{a}$. But how are the signals affected by p and q? Since

$$y(k) = \sum_{i=1}^{M_a} a_i y(k-i) + \sum_{j=0}^{M_b} b_j u(k-j),$$

the input $u(k)$ as well as the output $y(k)$ are proportional to p. Since the system gain q influences only the transversal part coefficients $\boldsymbol{b}$, they also affect the output signal $y(k)$. Thus $y(k)$ is proportional to pq. For reasons of simplicity let us further assume that the noise is zero. Then, the update error e_a is

$$e_a(k) = y(k) - \hat{\boldsymbol{a}}^{\mathrm{T}}(k)\boldsymbol{\psi}_a(k) - \hat{\boldsymbol{b}}^{\mathrm{T}}(k)\boldsymbol{\psi}_b(k),$$

where we split the gradient into two parts $\boldsymbol{\psi}^{\mathrm{T}}(k) = [\boldsymbol{\psi}_a^{\mathrm{T}}(k), \boldsymbol{\psi}_b^{\mathrm{T}}(k)]$. Since $\boldsymbol{\psi}_a(k)$ consists of output values, it is proportional to pq. Since so is the inner product $\hat{\boldsymbol{b}}^{\mathrm{T}}(k)\boldsymbol{\psi}_b(k) = \hat{\boldsymbol{b}}^{\mathrm{T}}(k)\boldsymbol{u}(k)$, the update error is proportional to pq as well. As long as fixed coefficients c_i are used, a filtering of the error does not change this relation. The normalization rule should take this dependency into account and provide an update of the coefficients $\hat{\boldsymbol{a}}(k)$ that is independent of p and q and of $\hat{\boldsymbol{b}}(k)$ that is independent of p but proportional to q. The following normalization addresses both points:

$$\hat{\boldsymbol{a}}(k+1) = \hat{\boldsymbol{a}}(k) + \frac{\alpha}{\boldsymbol{\psi}_a^{\mathrm{T}}(k)\boldsymbol{\psi}_a(k)} C[e_a(k)]\boldsymbol{\psi}_a(k), \quad (5.1a)$$

$$\hat{\boldsymbol{b}}(k+1) = \hat{\boldsymbol{b}}(k) + \frac{\alpha}{\boldsymbol{\psi}_b^{\mathrm{T}}(k)\boldsymbol{\psi}_b(k)} C[e_a(k)]\boldsymbol{\psi}_b(k). \quad (5.1b)$$

Formally, the equations are independent of changes in the system gain q and in the input level p. However, since the algorithms are nonlinear in the parameters, different q cause a slightly different behavior. Moreover, since the estimated values for the system output signals are rather small during the initial phase, the normalization might cause an extreme gain for parameter set $\hat{\boldsymbol{a}}(k)$ of the recursive part which results in instability. This can be remedied by adding a small positive constant to the normalizing term. However, when q is varied, the constant must also be varied in order to be exact. The normalization rule in (5.1a) and (5.1b) should only be seen as an example. In the following, we shall develop more sophisticated rules following the results of the former sections.

Since both vector parts have been split, two different step-sizes, one for each part, can allow more freedom. Therefore, in the next paragraph two different step-sizes $\mu_a(k)$ and $\mu_b(k)$ are considered. The application of two different step-sizes has been proposed in the literature [17, 12], but a condition on these parameters was left open. For the transversal filter case a condition was given in [16] assuming Gaussian statistics for the input sequence. The update equations for algorithms with unfiltered update error become

$$\boldsymbol{\varepsilon}(k+1) = \left(\boldsymbol{I}_{M\times M} - \begin{pmatrix}\mu_a(k)\boldsymbol{z}_{1a}(k)\\ \mu_b(k)\boldsymbol{z}_{1b}(k)\end{pmatrix}\boldsymbol{z}_2^{\mathrm{T}}(k)\right)\boldsymbol{\varepsilon}(k) + e_{\mathrm{m}}(k)\begin{pmatrix}\mu_a(k)\boldsymbol{z}_{1a}(k)\\ \mu_b(k)\boldsymbol{z}_{1b}(k)\end{pmatrix}. \quad (5.2)$$

Since a system of order one is obtained again, there is only one eigenvalue decisive for the behavior of the algorithm:

$$\lambda(k) = 1 - \mu_a(k)\boldsymbol{z}_{2a}^{\mathrm{T}}(k)\boldsymbol{z}_{1a}(k) - \mu_b(k)\boldsymbol{z}_{2b}^{\mathrm{T}}(k)\boldsymbol{z}_{1b}(k). \quad (5.3)$$

Here, the vectors $z_1(k)$ and $z_2(k)$ have been split into their original components, now specified with the additional subscript 'a' or 'b'. The corresponding eigenvector reads $(\mu_a(k)z_{1a}(k), \mu_b(k)z_{1b}(k))$. Due to Normalization 2, it is now possible to choose the following

Normalization 3:

$$\mu_a(k) = \alpha_a \frac{z_{1a}^{\mathrm{T}}(k)z_{2a}(k)}{z_{1a}^{\mathrm{T}}(k)z_{1a}(k)z_{2a}^{\mathrm{T}}(k)z_{2a}(k)}, \tag{5.4a}$$

$$\mu_b(k) = \alpha_b \frac{z_{1b}^{\mathrm{T}}(k)z_{2b}(k)}{z_{1b}^{\mathrm{T}}(k)z_{1b}(k)z_{2b}^{\mathrm{T}}(k)z_{2b}(k)}. \tag{5.4b}$$

Applying these normalizations to (5.3) the eigenvalue $\lambda(k)$ is located inside the unit circle for

$$0 < \alpha_a + \alpha_b < 2. \tag{5.5}$$

Since Normalization 3 also bounds the driving term, (5.5) gives the only necessary condition in order to have the eigenvalue $\lambda(k)$ inside the unit circle. Of course, the transition matrix in (5.2) is typically not a contraction mapping, and, therefore, the normalization does not necessarily lead to convergence for every input sequence. The optimal choice for α_a and α_b remains an open question as well.

The concept of using several different step-sizes can be generalized; for example, partitioning a longer vector into smaller parts with separate step-sizes can be of some advantage. In the extreme case every one of the M parameters can obtain its own step-size $\mu_i(k)$, $i = 1, \ldots, M$. This has been proposed, for example, for the SHARF algorithm [12, 13]. Normalization 3 then reduces to

$$\mu_i(k) = \alpha_i \frac{z_{1i}^{\mathrm{T}}(k)z_{2i}(k)}{z_{1i}^{\mathrm{T}}(k)z_{1i}(k)z_{2i}^{\mathrm{T}}(k)z_{2i}(k)}, \tag{5.6}$$

where the subscript i denotes the ith component of the vectors. Again, a condition in order to have the decisive eigenvalue $\lambda(k)$ inside the unit circle is obtained:

$$0 < \sum_{i=1}^{M} \alpha_i < 2. \tag{5.7}$$

A very similar result, but only for adaptive transversal filters, can be found in [16]. If anything is known about the parameter of the plant, a parameter update with individual step-sizes can be helpful. However, in this extreme case the homogeneous part is driven by the term $e_{\mathrm{m}}(k)/z_{2i}(k)$, $i = 1, \ldots, M$. If a very small value for any element of the vector $z_2(k)$ occurs even once, the whole system will burst

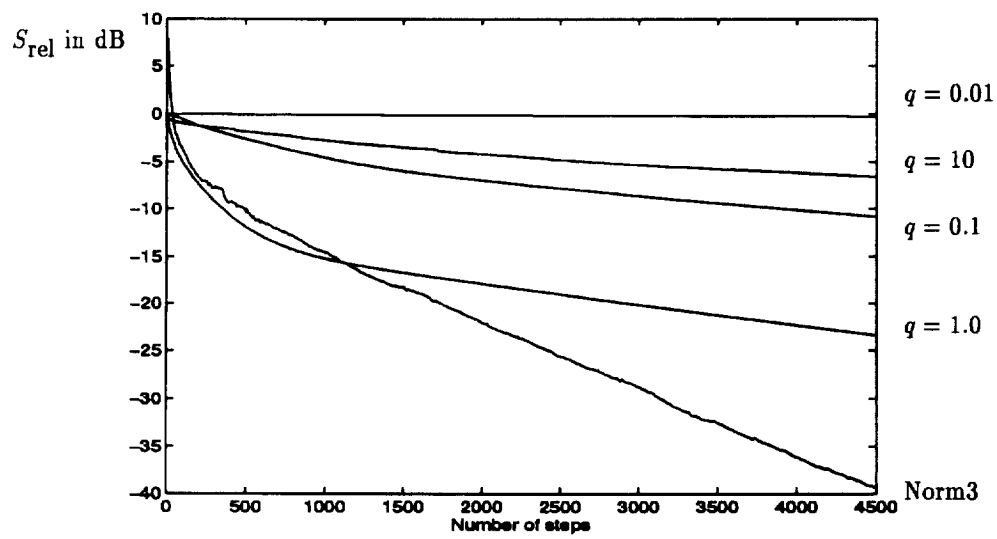

Fig. 7. NSHARF algorithm for several system gains q using Normalization 1 (= Normalization 2) and 3 and $\alpha = 0.1$.

out. Recently, an algorithm that uses this kind of normalization but updates only the coefficient for which $|e_m(k)/z_{2i}(k)|$ is maximal has been proposed [5] and showed even faster behavior than Normalization 1. If the normalization is positive, i.e. a norm, the squared L_1-norm can be useful in speech applications, since the typically low amplitudes of a speech signal are not markedly further lowered. Since $L_1 \geqslant L_2$, convergence is guaranteed as well.

These ideas have been tested with simulations for the NSHARF algorithm. The same conditions described for the former examples also applied here, but no disturbance was used. Fig. 7 depicts the relative system mismatch for several system gains q. As can be seen, if q varies, the algorithm can result in very low convergence speed. As simulations showed, the step-size for optimal convergence speed depends strongly on the unknown gain q. Only if q is known, an optimal step-size can be applied. We used $\alpha = 0.1$ in every run, but it should not be failed to mention that this step-size has not always been optimal. When Normalization 3 was applied, the algorithm behaved unchanged for every gain q. In Fig. 7 Normalization 3 resulted in a very quick system match.

6. Conclusion

Beginning with a generalization of gradient-based algorithms, two major groups have been distinguished: algorithms without and those with filtered adaptation errors. For both groups, two normalization rules have been proposed satisfying the condition to have all eigenvalues of the transition matrix inside (or on) the unit circle. Since these systems are time-varying, and have no eigenvalue outside the unit circle, it is a necessary but not sufficient condition to be convergent. For slowly time-varying systems we found a normalization rule that can satisfy all conditions for convergence. However, for general time-invariant systems, the more restrictive condition of being a contraction mapping must be fulfilled to guarantee convergence. Only for three (symmetric) algorithms the normalization can assure to have a contracting mapping and thus convergence for every input sequence is guaranteed. Real-time experiments as well as simulations validated the proposed normalization rules. However, convergence of an algorithm does not include stability of the calculated IIR filter function. This remains an open problem. Other possible normalization rules together with necessary conditions to make the behavior of the algorithms independent of the unknown system gain and to allow more freedom in the choice of the step-sizes were presented in the last section.

References

[1] M.G. Bellanger, *Adaptive Digital Filters and Signal Analysis*, Marcel Dekker, New York, 1986.

[2] H. Brehm and W. Stammler, "Description and generation of spherically invariant speech-model signals", *Signal Processing*, Vol. 12, No. 2, 1987, pp. 119–141.

[3] J. Chao, S. Kawabe and S Tsujii, "A new IIR adaptive echo canceler: GIVE", *Proc. IEEE Internat. Conf. System Engrg.*, Kobe, Japan, September 1992, pp. 547–551.

[4] J.E. Cousseau and P.S.R. Diniz, "A consistent Steiglitz–McBride algorithm", *Proc. Internat. Symp. Circuits Systems*, Chicago, May 1993, pp. 52–55.

[5] S.C. Douglas, "A family of normalized LMS algorithms", *IEEE Signal Processing Lett.*, Vol. 1, No. 3, 1994, pp. 49–51.

[6] L.J. Eriksson, "Development of the filtered-U algorithm for active noise control", *J. Acoust. Soc. Amer.*, Vol. 89, No. 1, April 1991, pp. 257–265.

[7] H. Fan, "A new adaptive IIR filter", *IEEE Trans. Circuits and Systems*, Vol. CAS-33, No. 10, October 1986, pp. 939–947.

[8] H. Fan and W.K. Jenkins, "An investigation of an adaptive IIR echo canceller: Advantages and problems", *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-36, No. 12, December 1988, pp. 1819–1833.

[9] P.L. Feintuch, "An adaptive recursive LMS filter", *Proc. IEEE*, Vol. 64, No. 11, November 1976, pp. 1622–1624.

[10] S. Gee and M. Rupp, "A comparison of adaptive IIR echo cancellers for hybrids", *Proc. Internat. Conf. Acoust. Speech Signal Process.*, Toronto, May 1991, pp. 1541–1544.

[11] I.S. Gradsteyn and I.M. Ryzhik, *Table of Integrals, Series and Products*, Academic, New York, 1980.

[12] C.R. Johnson Jr., "A convergence proof for a hyperstable adaptive recursive filter", *IEEE Trans. Inform. Theory*, Vol. IT-25, November 1979, pp. 745–749.

[13] M.G. Larimore, J.R. Treichler and C.R. Johnson, "SHARF: An algorithm for adapting IIR digital filters", *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-28, No. 4, August 1980, pp. 428–440.

[14] J. Lin and R. Unbehauen, "Modification of the least mean-square equation error algorithm for IIR system identification and adaptive filtering", *Proc. Internat. Symp. Circuits Systems*, New Orleans, 1990, pp. 3150–3153.

[15] J. Lin and R. Unbehauen, "Bias-remedy least mean-square equation error algorithm for IIR system parameter recursive estimation", *IEEE Trans. Signal Process.*, Vol. SP-40, No. 1, January 1992, pp. 62–69.

[16] S. Makino, Y. Kaneda and N. Koizumi, "Exponentially weighted step-size NLMS adaptive filter based on the statistics of a room impulse response", *IEEE Trans. Speech Audio Proc.*, Vol. 1, January 1993, pp. 101–108.

[17] W.B. Mikhael, F.H. Wu, L.G. Kazovsky, G.S. Kang and L.J. Fransen, "Adaptive filters with individual adaptation of parameters", *IEEE Trans. Circuits and Systems*, Vol. 33, No. 7, July 1986, pp. 677–685.

[18] M. Rupp, "Adaptive IIR echo cancellers for hybrids using the Motorola 56001", *Proc. Eusipco Signal Processing V*, Barcelona, 1990, pp. 1487–1490.

[19] M. Rupp, "The behavior of LMS and NLMS algorithms in the presence of spherically invariant processes", *IEEE Trans. Signal Process.*, Vol. SP-41, No. 3, March 1993, pp. 1149–1160.

[20] M. Rupp and A.H. Sayed, "On the stability and convergence of Feintuch's algorithm for adaptive IIR filtering", *Internat. Conf. Acoust. Speech Signal Process.*, Detroit, May 1995, pp. 1388–1391.

[21] A.H. Sayed and M. Rupp, "Optimality criteria for gradient-type algorithms", *Asilomar Conf.*, October 1994.

[22] J.J. Shynk, "Adaptive IIR filtering", *IEEE Acoust. Speech Signal Process., Mag.* Vol. 6, No. 2, April 1989, pp. 4–21.

[23] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms*, Prentice Hall Information and System Sciences Series, 1995.

[24] S.D. Stearns and G.R. Elliot, "On adaptive recursive filtering", *Proc. 10th Asilomar Conf. on Circuits, Systems, and Computers*, November 1976, pp. 5–11.

[25] K. Steiglitz and L.E. McBride, "A technique for the identification of linear systems", *IEEE Trans. Automat. Control*, Vol. AC-10, 1965, pp. 461–464.

[26] M. Tarrab and A. Feuer, "Convergence and performance analysis of the normalized LMS algorithm with uncorrelated Gaussian data", *IEEE Trans. Inform. Theory*, Vol. IT-34, No. 4, July 1988, pp. 680–691.