A Strict Stability Limit for Adaptive Gradient Type Algorithms

Robert Dallinger and Markus Rupp

Institute of Communications and Radio-Frequency Engineering
Vienna University of Technology
Gusshausstrasse 25/389, A-1040 Vienna, Austria
Email: {rdalling,mrupp}@nt.tuwien.ac.at

***Abstract*—This paper considers gradient type algorithms with a regression vector allowed to be different from the input vector. To cover the most general case, no restrictions are imposed on the dependency between the excitation vector and the regression vector. In terms of l_2–stability, for the real-valued domain, a convergence analysis is performed based on the singular value decomposition. It reveals that such algorithms are potentially unstable if the input vector and the regression vector do not have the same direction. For the conventional gradient type algorithm (for which latter vectors are parallel), an l_2–stability bound, known from literature to be sufficient, can be shown to be actually strict. Simulations demonstrate how the presented method can be used to discover unstable modes of an apparently stable algorithm.**

I. Introduction

Over the last few decades, adaptive gradient type algorithms have been studied on a broad scale. The probably most famous representative out of this family is the least mean square (LMS) algorithm [1]. In literature, stochastic as well as deterministic methods have been proposed to analyse stability and convergence behaviour of adaptive algorithms (an overview can be found in, e.g., [2]). In the case of lacking statistical a-priori knowledge about the analysed system, deterministic methods may lead to more reliable results than stochastic methods [3]. Accordingly, [4] states for the LMS an l_2–stability bound which is inherently conservative since it is obtained by the small-gain theorem [5]. The here presented approach, which is based on a convergence analysis using the singular value decomposition (SVD), allows to show that this bound is not only sufficient but actually also necessary. Moreover, it investigates a generalisation of the conventional gradient type algorithm in the sense that the regression vector is allowed to be an arbitrary vector which is not necessarily parallel to the input vector. This leads to the insight that for such algorithms, it is not possible to find a universal bound for the step-size which ensures stability. Instead, the decision on stability always requires to incorporate the specific relation between the input vector and the regression vector. Even though, in many cases it may not be possible to find stability conditions in closed form, numerical methods can be used to test the stability behaviour of a specific algorithm by simulations. This is demonstrated by simulations using an algorithm similar to the normalised LMS (NLMS), where the regression vector is disturbed by a noisy offset. For random offsets, the algorithm appears to be stable. However, the here presented method reveals hidden instabilities.

II. Generalised Gradient Type Algorithms

Throughout this work, a generalised gradient type algorithms is considered in the context of system identification in a real-valued environment. The reference system as well as the estimated system are assumed to be linear combiners with the same number M of parameters, contained in the vectors $\mathbf{w}$ and $\hat{\mathbf{w}}_k$ respectively. Starting at iteration step $k = 0$ with the initial estimate $\hat{\mathbf{w}}_{-1}$, the estimated system is adaptively updated following (cmp. Fig. 1)

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_{k-1} + \mu_k \mathbf{x}_k e_k, \tag{1}$$

where μ_k is the in general time-varying step-size. Assuming that the systems $\mathbf{w}$ and $\hat{\mathbf{w}}_{k-1}$ are both excited by the same input vector $\mathbf{u}_k$, the update error e_k is given by the difference between the observed response of the reference system and the output of the most recent estimate of the system, i.e.,

$$e_k = \mathbf{u}_k^{\mathsf{T}}\mathbf{w} + v_k - \mathbf{u}_k^{\mathsf{T}}\hat{\mathbf{w}}_{k-1}, \tag{2}$$

where v_k models additional disturbances and uncertainties in the observed signal. Introducing the parameter error vector $\tilde{\mathbf{w}}_k = \mathbf{w} - \hat{\mathbf{w}}_k$, (1) and (2) can be combined to

$$\tilde{\mathbf{w}}_k = \left[\mathbf{I} - \mu_k \mathbf{x}_k \mathbf{u}_k^{\mathsf{T}}\right] \tilde{\mathbf{w}}_{k-1} - \mu_k \mathbf{x}_k v_k, \tag{3}$$

with the $(M \times M)$ identity matrix $\mathbf{I}$. In practice, the regression vector $\mathbf{x}_k$ will depend on the input vector $\mathbf{u}_k$ by some certain function $f : \mathbb{R}^M \mapsto \mathbb{R}^M$. However, at this point, the regression vector $\mathbf{x}_k$ does not need to be specified further and is actually assumed to be independent from $\mathbf{u}_k$ (indicated by the dashed arrow in Fig. 1). To clarify two terms which are frequently used in the sequel, we present the following:

Definition 1. An adaptive algorithm with an update rule given by (1) and (2) with $M \geq 2$ is called *symmetric*, if the regression vector $\mathbf{x}_k$ is parallel to the input vector $\mathbf{u}_k$, i.e., $\mathbf{x}_k = f(\mathbf{u}_k) = \gamma_k \mathbf{u}_k$, with the real-valued and possibly time-varying scalar γ_k. If $\mathbf{x}_k$ contains a component orthogonal to $\mathbf{u}_k$, the algorithm is termed *asymmetric*.

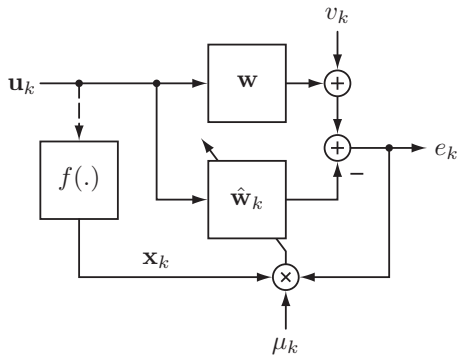

Fig. 1. The algorithmic structure considered in this paper.

III. Stability Analysis of Generalised Gradient Type Algorithms

In this section, the convergence behaviour of the parameter error vector $\tilde{\mathbf{w}}_k$ is considered. Starting in Sec. III-A with the investigation of the behaviour for one single time instant k, Sec. III-B provides global statements for symmetric and asymmetric algorithms. Following the lines of [3] and [4], we consider l_2–stability in the sense, that a finite normalised-noise energy and a bounded initial estimate, i.e., $\|\tilde{\mathbf{w}}_{-1}\|_2 < \infty$, results in a bounded parameter error vector, i.e., $\|\tilde{\mathbf{w}}_k\|_2 < \infty$.

A. Local Convergence

Since the normalised-noise energy is required to be finite, the term $\mu_k \mathbf{x}_k v_k$ in (3) has to vanish for $k \to \infty$. Therefore, by choosing k adequately large, the noise energy can be forced to become arbitrarily small which motivates (the common approach [4]) to confine the considerations in this section to the noiseless case (i.e., $v_k = 0$). Before global statements on convergence or divergence can be found, it is necessary to study the local mechanisms.

For the following derivations, it is convenient to rewrite (3) for the noiseless case in the form

$$\tilde{\mathbf{w}}_k = \mathbf{B}_k \tilde{\mathbf{w}}_{k-1}, \tag{4}$$

with the matrix

$$\mathbf{B}_k = \mathbf{I} - \mu_k \mathbf{x}_k \mathbf{u}_k^{\mathsf{T}}. \tag{5}$$

This allows us to define:

Definition 2. An adaptive algorithm given by (4) is called *potentially unstable*, if at least one singular value of the matrix $\mathbf{B}_k$ is larger than one. Otherwise it is called *locally convergent*.

With the induced l_2–norm $\|\mathbf{B}_k\|_2$ of the matrix $\mathbf{B}_k$ [6], the criterion for potential instability can be defined equivalently by $\|\mathbf{B}_k\|_2 > 1$. As a consequence, for a potentially unstable algorithm, $\tilde{\mathbf{w}}_{k-1}$ in (4) can always be chosen such that $\|\tilde{\mathbf{w}}_k\|_2 > \|\tilde{\mathbf{w}}_{k-1}\|_2$ (cf. also (8)), which is equivalent to local divergence.

The following two theorems, which are jointly proven further down, clarify the local convergence behaviour for asymmetric and symmetric algorithms.

Theorem 1. *In the noiseless case, real-valued asymmetric algorithms (cmp. Def. 1) are described by* (4) *and* (5)*. For such an algorithm,* $\|\mathbf{B}_k\|_2 = 1$ *iff* $\mu_k = 0$*. Otherwise, it is potentially unstable.*

In the corresponding proof, it will be seen that for $\mu_k \neq 0$, an asymmetric algorithm always has exactly one singular value larger than one and one singular value smaller than one. All remaining singular values are equal to one. For $\mu_k = 0$, $\mathbf{B}_k$ simplifies to the identity matrix and $\tilde{\mathbf{w}}_k = \tilde{\mathbf{w}}_{k-1}$.

Theorem 2. *According to Def. 1, real-valued symmetric algorithms perform in the noiseless case (i.e.,* $v_k = 0$*) a mapping given by* (4) *and* (5) *with* $\mathbf{x}_k = \gamma_k \mathbf{u}_k$*. For an arbitrary* $\mu_k \in \mathbb{R}$*,* $\|\mathbf{B}_k\|_2 = 1$ *iff one of the following conditions is satisfied:*

i.) μ_k *and* γ_k *have the same sign (i.e.,* $\mu_k \gamma_k > 0$*) and*

$$0 < |\mu_k| \leq \frac{2}{|\gamma_k| \|\mathbf{u}_k\|_2^2} = \frac{2|\gamma_k|}{\|\mathbf{x}_k\|_2^2} = \frac{2}{\|\mathbf{u}_k\|_2 \|\mathbf{x}_k\|_2}, \tag{6}$$

ii.) $\mu_k \gamma_k = 0$*.*

Otherwise, the algorithm is potentially unstable.

The following proof will reveal that symmetric algorithms have always only one singular value different from one, except for the case $\mu_k \gamma_k = 0$. Then, similarly to asymmetric algorithms, all singular values are equal to one and $\tilde{\mathbf{w}}_k = \tilde{\mathbf{w}}_{k-1}$.

Proof of Theorems 1 and 2: Since any matrix can be factorised using the SVD [7], also (the real-valued) $\mathbf{B}_k$ can be factorised by

$$\mathbf{B}_k = \mathbf{Q}_k \mathbf{\Sigma}_k \mathbf{V}_k^{\mathsf{T}}, \tag{7}$$

where the diagonal matrix $\mathbf{\Sigma}_k$ contains the singular values $\sigma_{i;k}$ of $\mathbf{B}_k$ (i.e., the positive roots of the eigenvalues of $\mathbf{B}_k \mathbf{B}_k^{\mathsf{T}}$ or equivalently $\mathbf{B}_k^{\mathsf{T}} \mathbf{B}_k$), and $\mathbf{Q}_k$ and $\mathbf{V}_k$ are unitary matrices formed by the normalised left-sided respectively right-sided singular vectors of $\mathbf{B}_k$. This allows to express the maximum dilatation introduced by $\mathbf{B}_k$ as

$$\sup_{\|\tilde{\mathbf{w}}_{k-1}\|_2 = 1} \|\tilde{\mathbf{w}}_k\|_2 = \sup_{\|\tilde{\mathbf{w}}_{k-1}\|_2 = 1} \|\mathbf{B}_k \tilde{\mathbf{w}}_{k-1}\|_2 = \max_{1 \leq i \leq M} \sigma_{i;k}. \tag{8}$$

The singular vectors of the matrix $\mathbf{B}_k$ will be considered in more detail in the proofs of Lemma 1 and 2. Accordingly, it can be verified that $\mathbf{B}_k$ has $M - 2$ unit singular values and two singular values given by

$$\sigma_{1,2;k}^2 = 1 + \frac{\alpha_k^2}{2} - \alpha_k \rho_k \pm \alpha_k \sqrt{1 + \frac{\alpha_k^2}{4} - \alpha_k \rho_k}, \tag{9}$$

for which $i = 1, 2$ was arbitrarily chosen without loss of generality (WOLOG). Additionally, in (9) the correlation coefficient ρ_k and the normalised step-size α_k were introduced, defined by:

$$\alpha_k = \mu_k \|\mathbf{x}_k\|_2 \|\mathbf{u}_k\|_2, \tag{10}$$

$$\rho_k = \frac{\mathbf{x}_k^{\mathsf{T}} \mathbf{u}_k}{\|\mathbf{x}_k\|_2 \|\mathbf{u}_k\|_2}. \tag{11}$$

1) Specific for Theorem 1: From (9), we obtain for asymmetric algorithms with $\mu_k \neq 0$

$$\frac{\sigma_{1,2;k}^2 - 1}{\alpha_k} = \frac{\alpha_k}{2} - \rho_k \pm \underbrace{\sqrt{1 + \frac{\alpha_k^2}{4} - \alpha_k \rho_k}}_{> \left|\frac{\alpha_k}{2} - \rho_k\right| \text{ since } |\rho_k| < 1}. \tag{12}$$

Therefore, except from the case $\mu_k = 0$, an asymmetric algorithm has always exactly one singular value larger than one and one singular value smaller than one, and thus, $\|\mathbf{B}_k\|_2 > 1$.

2) Specific for Theorem 2: For symmetric algorithms, in the non-trivial case (i.e., $\mu_k \neq 0, \gamma_k \neq 0$), the magnitude of the correlation coefficient is equal to one ($\rho_k = \operatorname{sign}\gamma_k$), and consequently,

$$\alpha_k \rho_k = \operatorname{sign}(\mu_k \gamma_k)\, |\alpha_k|. \tag{13}$$

In this case, it can easily be verified that one singular value in (9) equals to one, the other one (WOLOG, lets assume $\sigma_{1;k}$) is given by

$$\sigma_{1;k} = \left|1 - \operatorname{sign}(\gamma_k)\, \alpha_k\right|, \tag{14}$$

which directly allows to identify condition i.) in Theorem 2.

3) The trivial case: For $\mu_k = 0$ in the asymmetric case, respectively $\mu_k \gamma_k = 0$ in the symmetric case, the mapping in (4) simplifies to the identity, and $\|\mathbf{B}_k\|_2 = \|\mathbf{I}\|_2 = 1$. ■

It should be pointed out that Theorem 1 states that in general, for asymmetric algorithms, from any time instant k to the subsequent one, the norm of the parameter error vector can either grow, remain the same or shrink. This behaviour cannot be influenced by the choice of the step-size μ_k (excluding the trivial case). By which it is already revealed that asymmetric algorithms cannot be ensured to converge in general. The following section addresses this fact in more detail.

B. Global Convergence

In this section, the Theorems 1 and 2 are employed to analyse the global convergence behaviour of the generalised gradient type algorithms. In this context, it is necessary to specify the terms convergence and divergence more precisely.

Definition 3. An algorithm given by (4) is *convergent* if for all possible combinations of the pair of real-valued vector sequences $\{\mathbf{u}_k\}$ and $\{\mathbf{x}_k\}$, the norm of the parameter error vector does not increase for all but a finite number of k:

$$\max_{\{\mathbf{u}_k\},\{\mathbf{x}_k\}} \|\tilde{\mathbf{w}}_k\|_2 \leq \|\tilde{\mathbf{w}}_{k-1}\|_2 \;\forall\, k \in \mathbb{N}_0 \setminus \bar{\mathcal{K}}_{(>)}, \tag{15}$$

where $\bar{\mathcal{K}}_{(>)} \subset \mathbb{N}_0$ is the finite set of time instants for which $\|\tilde{\mathbf{w}}_k\|_2 > \|\tilde{\mathbf{w}}_{k-1}\|_2$.
Analogously, the algorithm is *divergent* if

$$\max_{\{\mathbf{u}_k\},\{\mathbf{x}_k\}} \|\tilde{\mathbf{w}}_k\|_2 > \|\tilde{\mathbf{w}}_{k-1}\|_2 \;\forall\, k \in \mathbb{N}_0 \setminus \bar{\mathcal{K}}_{(\leq)}. \tag{16}$$

The sequences solving the left side of (15) respectively (16) are here shortly referred to by *worst-case sequences*.

Note that in this work, the case of infinite sets $\bar{\mathcal{K}}_{(>)}$, respectively $\bar{\mathcal{K}}_{(\leq)}$, is excluded, since then either the converging or the diverging behaviour may dominate.

Lemma 1. *For an asymmetric algorithm with the vectors $\mathbf{u}_k$ and $\mathbf{x}_k$ being independent from each other, a pair of worst-case sequences, $\{\mathbf{u}_k\}$ and $\{\mathbf{x}_k\}$, leading to divergence, can always be found.*

Proof: Note that for the sake of brevity, in this proof, the subscripts k for the singular values are suppressed. Similarly, in the subscripts of the singular vectors, k is omitted. However, keeping in mind that they depend on the time index k.

In a first step, we consider the right-sided singular vectors of the matrix $\mathbf{B}_k$ in (4), which also provides us with the corresponding singular values. Based on these results, we derive rules for the construction of the worst-case sequences for $\{\mathbf{u}_k\}$ and $\{\mathbf{x}_k\}$.

It can easily be verified that each set of $M-2$ orthonormal vectors, which are all orthogonal to $\mathbf{u}_k$ and $\mathbf{x}_k$, is a set of singular vectors of $\mathbf{B}_k$. The corresponding singular values are $\sigma_i = 1$ with $i = 3, \ldots, M$, chosen WOLOG. Since the singular vectors are mutually orthogonal, the remaining two singular vectors are necessarily a linear combination of $\mathbf{u}_k$ and $\mathbf{x}_k$. Knowing that, the solution of $\mathbf{B}_k^\mathsf{T}\mathbf{B}_k\bar{\mathbf{v}}_{1,2} = \sigma_{1,2}^2\bar{\mathbf{v}}_{1,2}$ with $\bar{\mathbf{v}}_{1,2} \in \operatorname{span}\{\mathbf{u}_k, \mathbf{x}_k\}$ leads to the right-sided singular vectors (derivation skipped)

$$\bar{\mathbf{v}}_{1,2} = \left(\frac{1 - \sigma_{1,2}^2}{\alpha_k} - \rho_k\right) \frac{\mathbf{u}_k}{\|\mathbf{u}_k\|_2} + \frac{\mathbf{x}_k}{\|\mathbf{x}_k\|_2}. \tag{17}$$

For completeness, it should be mentioned that the left-sided singular vectors are obtained by (17), with the sign changed and all $\mathbf{u}_k$ replaced by $\mathbf{x}_k$ and vice versa. The derivation also provides us with the singular values $\sigma_{1,2}$ which were already presented in (9).

It holds that $\|\tilde{\mathbf{w}}_k\|_2 = \|\mathbf{Q}_k^\mathsf{T}\tilde{\mathbf{w}}_k\|_2$, since $\mathbf{Q}_k$ in (7) is unitary. Hence, using the fact that $\sigma_i = 1$ for $i = 3, \ldots, M$, the squared l_2–norm of (4) can be rewritten as

$$\begin{aligned}\|\tilde{\mathbf{w}}_k\|_2^2 &= \\ &= \|\mathbf{Q}_k^\mathsf{T}\tilde{\mathbf{w}}_k\|_2^2 = \|\Sigma_k\mathbf{V}_k^\mathsf{T}\tilde{\mathbf{w}}_{k-1}\|_2^2 = \sum_{i=1}^{M}\left(\sigma_i\mathbf{v}_i^\mathsf{T}\tilde{\mathbf{w}}_{k-1}\right)^2 \\ &= \|\tilde{\mathbf{w}}_{k-1}\|_2^2 + \left(\sigma_1^2 - 1\right)\left(\mathbf{v}_1^\mathsf{T}\tilde{\mathbf{w}}_{k-1}\right)^2 \\ &\quad + \left(\sigma_2^2 - 1\right)\left(\mathbf{v}_2^\mathsf{T}\tilde{\mathbf{w}}_{k-1}\right)^2,\end{aligned} \tag{18}$$

where the normalised singular vectors $\mathbf{v}_i = \|\bar{\mathbf{v}}_i\|_2^{-1}\bar{\mathbf{v}}_i$ with $i = 1, 2$ were introduced. Equ. (17) and (18) together with (9)–(11) form the rules to compose the worst-case sequences.

In the sequel, WOLOG, let $\sigma_1 > 1$ and $\sigma_2 < 1$. Then, it becomes obvious that with respect to local convergence, the worst-case is achieved if

$$\tilde{\mathbf{w}}_{k-1} \parallel \mathbf{v}_1 \quad \text{and} \quad \tilde{\mathbf{w}}_{k-1} \perp \mathbf{v}_2. \tag{19}$$

Note however that the parameter error $\|\tilde{\mathbf{w}}_k\|_2$ even grows as long as $\left(\sigma_1^2 - 1\right)\left(\mathbf{v}_1^\mathsf{T}\tilde{\mathbf{w}}_{k-1}\right)^2 > \left(1 - \sigma_2^2\right)\left(\mathbf{v}_2^\mathsf{T}\tilde{\mathbf{w}}_{k-1}\right)^2$.

If $\mathbf{u}_k$ and $\mathbf{x}_k$ can be chosen arbitrarily, (19) can always be satisfied. The easiest way to achieve this is to choose $\mathbf{u}_k \perp \mathbf{x}_k$. Then, the singular values only depend on α_k and become

independent from the direction of $\mathbf{u}_k$ and $\mathbf{x}_k$. The choice of $\mathbf{v}_1 \parallel \tilde{\mathbf{w}}_{k-1}$ (and thus, $\mathbf{v}_2 \perp \tilde{\mathbf{w}}_k$) leads to a linear system of equations which allows to determine $\mathbf{u}_k$ and $\mathbf{x}_k$ for the worst-case. Thus, for independent vectors $\mathbf{u}_k$ and $\mathbf{x}_k$, worst-case sequences leading to divergence can always be found. ■

For a specific algorithm, the vector $\mathbf{x}_k$ is normally *not* independent from $\mathbf{u}_k$. The dependency $\mathbf{x}_k = f(\mathbf{u}_k)$ can drastically restrict the space of sequence pairs in (15) and (16). This may cause that not a single pair of worst-case sequences exists which leads to divergence. As an example of an asymmetric algorithm with a region of convergence, we consider a gradient algorithm with a time-invariant positive definite real-valued step-size matrix $\mathbf{M}$, i.e., $\mathbf{x}_k = \mathbf{M}\mathbf{u}_k$ (cmp. also [3]). Obviously, $\mathbf{x}_k$ depends rigidly on $\mathbf{u}_k$. Due to the Cholesky factorisation [6], $\mathbf{M} = \mathbf{L}\mathbf{L}^\mathsf{T}$ with the lower triangular matrix $\mathbf{L}$, which is invertible since $\mathbf{M}$ is invertible. Then, it can easily be verified that by multiplication from the left with $\mathbf{L}^{-1}$, (4) with (5) can be mapped to

$$\tilde{\mathbf{w}}'_k = \left[\mathbf{I} - \mu_k \mathbf{u}'_k \mathbf{u}'^\mathsf{T}_k\right] \tilde{\mathbf{w}}'_{k-1}, \tag{20}$$

where $\tilde{\mathbf{w}}'_k = \mathbf{L}^{-1}\tilde{\mathbf{w}}_k$ and $\mathbf{u}'_k = \mathbf{L}^\mathsf{T}\mathbf{u}_k$. Consequently, the asymmetric algorithm with constant positive definite step-size matrix $\mathbf{M}$ can be mapped to an equivalent symmetric algorithm for which Theorem 2 and Lemma 2 (see below) apply. Though, the original algorithm has a singular value larger than one. Finally, the norm equivalence theorem (see [6], p. 96) allows to conclude that if the equivalent symmetric algorithm converges (diverges) also the original asymmetric algorithm converges (diverges).

Lemma 2. *Symmetric algorithms converge (diverge) in the sense of Def. 3 iff the conditions for local convergence in Theorem 2 are satisfied (violated) for all but a finite number of k.*

Proof: This proof follows the same line as the proof of Lemma 1. Again, the index k is omitted for singular values and singular vectors. In the symmetric case, $M-1$ mutually orthogonal vectors can be found which are orthogonal to $\mathbf{u}_k$ (and $\mathbf{x}_k$). Consequently, $\mathbf{B}_k$ has $M-1$ singular values equal to one. The remaining singular vector $\mathbf{v}_1$ necessarily has the same direction as $\mathbf{u}_k$ (and $\mathbf{x}_k$). The corresponding singular value, WOLOG, lets assume σ_1, can easily be verified to be given by (14). Similarly, as for the asymmetric case, we obtain

$$\|\tilde{\mathbf{w}}_k\|_2^2 = \|\tilde{\mathbf{w}}_{k-1}\|_2^2 + \left(\sigma_1^2 - 1\right)\left(\mathbf{v}_1^\mathsf{T}\tilde{\mathbf{w}}_{k-1}\right)^2. \tag{21}$$

This shows that $\|\tilde{\mathbf{w}}_k\|_2 \leq \|\tilde{\mathbf{w}}_{k-1}\|_2$ for any $\mathbf{v}_1$, as long as $\sigma_1 \leq 1$. Hence, (15) is satisfied if the conditions for local convergence in Theorem 2 are satisfied for all but a finite number of k.

On the other hand, if $\sigma_1 > 1$, $\|\tilde{\mathbf{w}}_k\|_2 > \|\tilde{\mathbf{w}}_{k-1}\|_2$ can always be achieved by choosing $\mathbf{u}_k \parallel \tilde{\mathbf{w}}_{k-1}$, which already states the rule for the generation of a worst-case sequence which satisfies (16). Thus, if the conditions for local convergence in Theorem 2 are violated infinitely often, the algorithm is divergent in the sense of Def. 3. ■

Lemma 2 provides a *strict* condition on convergence. The bound is the same as in [4]. However, there the condition is only shown to be sufficient since it is derived using the small-gain theorem [5].

IV. Application and Simulation Results

In this section, simulations are presented which apply the previous findings to a specific asymmetric algorithm and show that it is unstable although it seems to be stable. According to the structure in Fig. 1, the NLMS is obtained for

$$\mathbf{x}_k = f_{\mathrm{NLMS}}(\mathbf{u}_k) = \frac{\mathbf{u}_k}{\|\mathbf{u}_k\|_2^2}, \tag{22}$$

and the conditions for local convergence (which are also globally valid, due to Lemma 2) lead to the well known stability bound $0 \leq \mu_k \leq 2$. However, we now assume that the function f introduces random disturbances to the input vector. This leads to a regression vector of the same length as for the NLMS, but with a randomly modified direction:

$$\mathbf{x}_k = f_{\mathrm{noisy}}(\mathbf{u}_k) = \frac{\mathbf{z}_k}{\|\mathbf{u}_k\|_2\|\mathbf{z}_k\|_2}, \tag{23}$$

where the vector $\mathbf{z}_k$ is given by the direction of $\mathbf{u}_k$ with some random offset vector of constant length:

$$\mathbf{z}_k = \frac{\mathbf{u}_k}{\|\mathbf{u}_k\|_2} + \frac{\mathbf{r}_k}{3\|\mathbf{r}_k\|_2}. \tag{24}$$

In the sequel, we will term the algorithm which uses f_{noisy}, the 'noisy' NLMS. It belongs to the class of asymmetric algorithms since the (normalised) regression vector is not parallel to the (normalised) input vector.

In the performed simulations, systems of length $M = 5$ are considered. All random signals (vectorial or scalar) are zero-mean, white, Gaussian, and mutually statistically independent. The additive noise in Fig. 1 has a variance $\sigma_v^2 = 10^{-6}$ and the entries of the input vector $\mathbf{u}_k$ have unit variance $\sigma_u^2 = 1$. The results are averaged over 1000 separate runs. For each run, the reference system $\mathbf{w}$ is randomly generated as a zero-mean, white, Gaussian vector of unit norm. The algorithms are evaluated by the observation of the relative system misadjustment, defined by

$$m_{\mathbf{w}}(k) = \frac{\|\tilde{\mathbf{w}}_k\|_2^2}{\|\tilde{\mathbf{w}}_{-1}\|_2^2}. \tag{25}$$

Two separate simulations are performed. In the first one, for each time instant, $\mathbf{r}_k$ is independently generated as Gaussian random vector. The corresponding results for the constant step-sizes $\mu_k = \mu \in \{0.5, 1, 1.5\}$ are depicted in Fig. 2 and compared to the learning curves of the unmodified NLMS. Except from a slightly reduced convergence speed, the 'noisy' NLMS does not show any major differences.

In the second set of simulations, $\mathbf{r}_k$ is chosen with the objective of achieving the worst-case in the sense of Def. 3. At this point, it has to be mentioned that the relation between $\mathbf{x}_k$

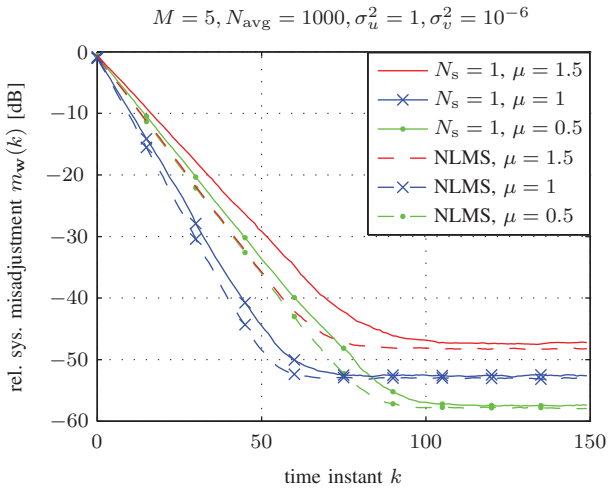

Fig. 2. 'Noisy' NLMS in comparison to the actual NLMS. $N_s = 1$ indicates that $\mathbf{r}_k$ was randomly generated and no worst-case search was performed.

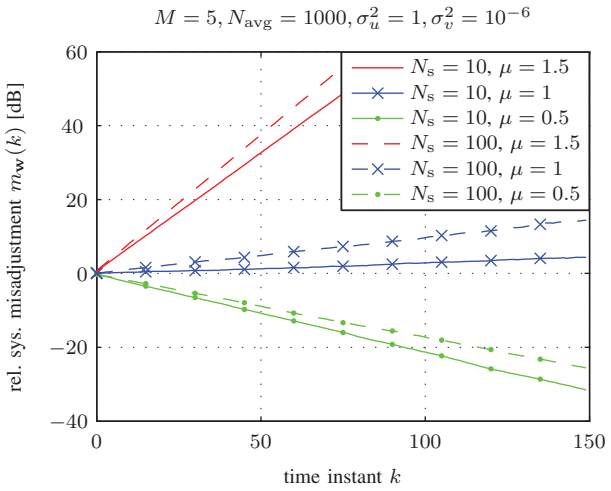

Fig. 3. 'Noisy' NLMS with $N_s = 10$ and $N_s = 100$ random trials to approximate the worst-case sequence of $\mathbf{x}_k$.

and $\mathbf{u}_k$ via the function f can extremely complicate the identification of worst-case sequences. In many cases, numerical methods will be necessary for their identification. Even for the rather artificial example of the here considered 'noisy' NLMS, a theoretical construction of worst-case sequences leads to nonlinear equations which are difficult to handle. Therefore, for a given random sequence $\{\mathbf{u}_k\}$, the worst-case sequence $\{\mathbf{x}_k\}$ was approximated by a random search in combination with the exploitation of the insights from Sec. III-B. There, it was already found that for the worst-case sequences, the singular vector corresponding to the largest singular value, has to be parallel to $\tilde{\mathbf{w}}_{k-1}$. Additionally, this singular vector is also a linear combination of $\mathbf{u}_k$ and $\mathbf{x}_k$. Vice versa, the regression vector $\mathbf{x}_k$ leading to the worst-case lies in the hyperplane spanned by the parameter error vector $\tilde{\mathbf{w}}_{k-1}$ and the corresponding input vector $\mathbf{u}_k$. Knowing this, the worst-case can be approximated by randomly searching the vector $\mathbf{x}_k \in \text{span}\{\mathbf{u}_k, \tilde{\mathbf{w}}_{k-1}\}$ which maximises the norm of $\tilde{\mathbf{w}}_k$ in (4). Since in this simulation $\mathbf{x}_k$ is given by (23), instead of $\mathbf{x}_k$, the vector $\mathbf{r}_k \in \text{span}\{\mathbf{u}_k, \tilde{\mathbf{w}}_{k-1}\}$ has to be found which maximises $\|\tilde{\mathbf{w}}_k\|_2$.

Fig. 3 shows the results of two simulation sets. One uses $N_s = 10$ trials for the worst-case search, the other one uses $N_s = 100$. Compared to Fig. 2, for $\mu_k = \mu = 0.5$, the algorithm shows a reduced convergence speed. It converges slower for $N_s = 100$ than for $N_s = 10$. For simulations performed with $N_s = 1000$, the convergence behaviour was almost the same as for $N_s = 100$. Thus, with $\mu = 0.5$ the algorithm seems to be actually stable. In other words, it seems that for this step-size (and presumably for smaller step-sizes as well), no worst-case sequence can be constructed which leads to divergence. On the other hand, for $\mu_k = \mu \in \{1, 1.5\}$ the algorithm obviously diverges. Similarly to the converging case, the speed of divergence is higher for the simulations using $N_s = 100$ than for those using $N_s = 10$. And again, experiments with $N_s = 1000$ led to an only marginally higher divergence speed than for $N_s = 100$.

These simulations show that on the first glance, the here considered 'noisy' NLMS seems to be stable. Nevertheless, the use of worst-case sequences allows to discover (perhaps unlikely but possible) cases for which it diverges. From another perspective, the approach allows to identify the range of step-sizes for which the algorithm is ensured to be stable.

V. Conclusions

Extending the findings of [3] and [4], l_2-stability conditions were developed which revealed the novel insight that adaptive gradient type algorithms of the asymmetric class are potentially unstable. Moreover, the l_2–stability bound from [4] was shown to be necessary and not only sufficient. It would be desirable to find a more systematic way to construct the worst-case sequences instead of using a random search. The here performed analysis is restricted to the real-valued case, an extension to the complex domain will be presented elsewhere.

Acknowledgements

This work has been funded by the NFN SISE (National Research Network "Signal and Information Processing in Science and Engineering").

References

[1] B. Widrow and M. E. Hoff Jr., "Adaptive switching circuits," *IRE WESCON conv. Rec.*, vol. Part 4, pp. 96–104, 1960.

[2] A. H. Sayed, *Fundamentals of adaptive filtering*, John Wiley & Sons, Inc., Hoboken (NJ), USA, 2003.

[3] A. H. Sayed and M. Rupp, "Error-energy bounds for adaptive gradient algorithms," *IEEE Trans. on Signal Processing*, vol. 44, no. 8, pp. 1982–1989, Aug. 1996.

[4] M. Rupp and A. H. Sayed, "A time-domain feedback analysis of filtered-error adaptive gradient algorithms," *IEEE Trans. on Signal Processing*, vol. 44, no. 6, pp. 1428–1439, Jun. 1996.

[5] A. J. van der Schaft, *L_2-gain and passivity techniques in nonlinear control*, Springer, London, UK, 2nd edition, 2000.

[6] T. K. Moon and W. C. Stirling, *Mathematical methods and algorithms for signal processing*, Prentice-Hall, Inc., Upper Saddle River (NJ), USA, 2000.

[7] G. H. Golub and C. F. Van Loan, *Matrix computations*, Johns Hopkins Univiversity Press, Baltimore (MD), USA, 3rd edition, 1996.