LMS ALGORITHMIC VARIANTS IN ACTIVE NOISE AND VIBRATION CONTROL

Markus Rupp[*]

Vienna University of Technology
Institute of Telecommunications
Gusshausstr. 25/389, A-1040 Vienna, Austria
e-mail: mrupp@nt.tuwien.ca.at

Fabian Hausberg

I/EF-35, Engine/Gearbox Mountings
Audi AG, D-85045 Ingolstadt, Germany
e-mail: fabian.hausberg@audi.de

ABSTRACT

In this article we provide analyses of two low complexity LMS algorithmic variants as they typically appear in the context of FXLMS for active noise or vibration control in which the reference signal is not obtained by sensors but internally generated by the known engine speed. In particular we show that the algorithm with real valued error is robust and exhibits the same steady state quality as the original complex-valued LMS algorithm but at the expense of only achieving half the learning speed while its counterpart with real-valued regression vector behaves only equivalently in the statistical sense.

Index Terms— FXLMS algorithm, error bounds, l_2-stability, robustness, mean-square-convergence.

1. INTRODUCTION

In Figure 1 we depict the classical active noise suppression setup as it is common in literature [1]. Recently such algorithms have received more attention due to their applicability in active engine mounts to suppress disturbing engine vibrations [2–6]. Both have in common that the typical algorithm to solve the control problem is the well-known Filtered-X-Least Mean Square (FXLMS) algorithm [1, 7]. While classically the FXLMS algorithm requires measured sensor data as input, in car engines, the vibrations are solely generated by the engine not requiring additional sensors as the engine speed in terms of its revolutions ω_0 alone is sufficient knowledge to reconstruct the vibration signal $z(t)$:

$$\begin{aligned} z(t) &= \sum_{l=1}^{L} w_{R,l}\cos(l\omega_0 t) + w_{I,l}\sin(l\omega_0 t) \\ &= \Re\left\{\mathbf{w}^\mathsf{T}\mathbf{x}(t)\right\}, \end{aligned} \tag{1}$$

with $w_{R,l}$ and $w_{I,l}$ denoting the real and imaginary parts of the vector entries $\mathbf{w}$, corresponding to the linear system W in Figure 1. The upper scripts $^\mathsf{T}$,$^\mathsf{H}$, and '$*$' as used here and in the following equations stand for transpose, Hermitian and conjugate complex, respectively. The driving source signal $x(t)$ can thus artificially be generated. Due to its signal structure containing $\cos()$ and $\sin()$, the adaptive control algorithm is often formulated as complex-valued. The order of the adaptive filter M is selected to be L allowing to compensate the L harmonics in the input signal. After sampling $\mathbf{x}(t)$, the now complex-valued input vector $\mathbf{x}_k$ has the following entries

$$\mathbf{x}_k = [\exp(j\Omega_0 k), \exp(j2\Omega_0 k), ..., \exp(jL\Omega_0 k)]^\mathsf{T}. \tag{2}$$

The sampled output z_k is additionally disturbed by noise v_k and we observe

$$d_k = \Re\left\{\mathbf{x}_k^\mathsf{T}\mathbf{w} + v_k\right\}. \tag{3}$$

Together with the control signal y_k an error term $\tilde{e}_{f,k}$ is physically generated. Thus a physical implementation of such systems must rely on the real part of $\tilde{e}_{f,k}$, often further linearly filtered by a so called auxiliary path H before measured.

While typically in such context the FXLMS algorithm [1, 7, 8] is applied, with its specific problems of including an auxiliary path H, we will concentrate in this paper only on specific properties of the driving process $\mathbf{x}_k$ and the gradient updates so that we simply set $H = 1$ in Fig. 1, obtaining an LMS algorithm for which the filtered error signal $\tilde{e}_{f,k} = \tilde{e}_{a,k}$, that is the distorted but unfiltered a-priori error signal. Equivalently, we consider the LMS algorithm in its complex-valued form [9], further referred to as CLMS:

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_{k-1} + \mu\mathbf{x}_k^*\tilde{e}_{a,k}, \tag{4}$$

$$\tilde{e}_{a,k} = d_k - \mathbf{x}_k^\mathsf{T}\hat{\mathbf{w}}_{k-1}. \tag{5}$$

Due to the physical nature of the problem, a complex-valued error signal $\tilde{e}_{a,k}$ cannot be applied to control the vibrations and thus only the real part of the error signal $\Re\{\tilde{e}_{a,k}\}$ is being applied, modifying the CLMS algorithm.

1.1. Relation to prior work

While the FXLMS algorithm has been analyzed for many years [1, 7, 8, 10] and the CLMS is well known and understood [9] with many applications in data transmission, using

[*]This work has been funded by the NFN SISE project S10611 (National Research Network Signal and Information Processing in Science and Engineering).

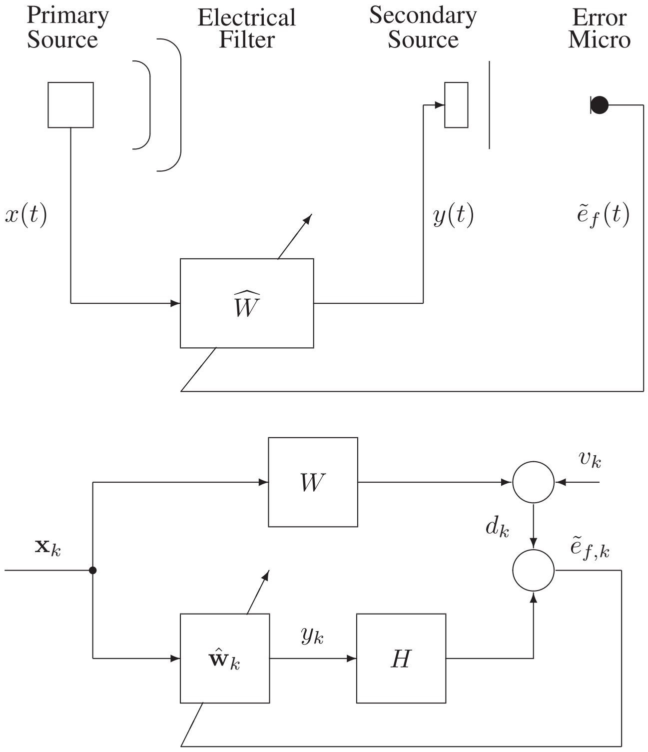

Fig. 1. *Upper: Active vibration suppression. Lower: Equivalent block diagram of the FXLMS algorithm.*

simply the real part of either the error or the regression vector in a CLMS algorithm is novel and has not been analyzed at all. In [2,4] the algorithm with real error signal was proposed but only its stability nature in the context of the FXLMS algorithm has been analyzed. Experiments on vehicles [6] have proven the algorithm to work but lack theoretical understanding and thus step-sizes are selected based on experimental data. This paper's contribution is intended to fill this gap.

1.2. Notation and paper structure

We denote vectors by bold face type small variables and matrices by capital letters. In the following Section 2 we investigate two variants of the CLMS algorithm, one with real-valued error terms and due to symmetry we also analyze its counterpart with real-valued regression vector. All together the following two variants are of interest:

1. RELMS: algorithm with real-valued error

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_{k-1} + \mu \mathbf{x}_k^* \Re\left\{\tilde{e}_{a,k}\right\}. \tag{6}$$

2. RXLMS: algorithm with real-valued regression vector

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_{k-1} + \mu \Re\left\{\mathbf{x}_k^*\right\} \tilde{e}_{a,k}. \tag{7}$$

Due to the multiplication with a real rather than a complex valued term, both algorithms result in considerably less complexity when compared to a conventional CLMS algorithm. We continue our analysis of the algorithm and show robustness only for the RELMS algorithm in Section 3 while its counterpart the RXLMS algorithm behaves non-robustly under worst case sequences. Finally, we present simulation results in Section 4 to support our theoretical findings and deliver some closing remarks in Section 5.

2. MSE ANALYSIS

Applying only the real part of the error term $e_{a,k}$ or alternatively the real part of the regression vector $\mathbf{x}_k$ results in reduced complexity. In the following we investigate the impact on the learning speed and steady state of such variants.

Theorem 2.1 *With respect to the Mean Squared Error (MSE) of the parameter error vector, the RELMS and RXLMS algorithm behave identically. Compared to CLMS they both*

- *exhibit the same steady-state values*
- *can only achieve half the learning speed.*

Proof: We introduce the parameter error vector $\tilde{\mathbf{w}}_k = \mathbf{w} - \hat{\mathbf{w}}_k$ with $\mathbf{w}$ denoting our reference system (Wiener solution) and formulate its updates explicitly into real and imaginary parts ($\Re\left\{\tilde{\mathbf{w}}_k\right\} = \tilde{\mathbf{w}}_{R,k}, \Im\left\{\tilde{\mathbf{w}}_k\right\} = \tilde{\mathbf{w}}_{I,k}$). We obtain for CLMS:

$$\begin{aligned}
\begin{bmatrix} \tilde{\mathbf{w}}_{R,k} \\ \tilde{\mathbf{w}}_{I,k} \end{bmatrix} &= \mathbf{B}_k^{(LMS)} \begin{bmatrix} \tilde{\mathbf{w}}_{R,k-1} \\ \tilde{\mathbf{w}}_{I,k-1} \end{bmatrix} \\
&\quad - \mu \begin{bmatrix} \mathbf{x}_{R,k} v_{R,k} + \mathbf{x}_{I,k} v_{I,k} \\ \mathbf{x}_{R,k} v_{I,k} - \mathbf{x}_{I,k} v_{R,k} \end{bmatrix} \\
\mathbf{B}_k^{(LMS)} &= \\
\mathbf{I} - \begin{bmatrix} \mu \mathbf{x}_{R,k}\mathbf{x}_{R,k}^\mathsf{T} + \mu \mathbf{x}_{I,k}\mathbf{x}_{I,k}^\mathsf{T} & -\mu \mathbf{x}_{R,k}\mathbf{x}_{I,k}^\mathsf{T} + \mu \mathbf{x}_{I,k}\mathbf{x}_{R,k}^\mathsf{T} \\ \mu \mathbf{x}_{R,k}\mathbf{x}_{I,k}^\mathsf{T} - \mu \mathbf{x}_{I,k}\mathbf{x}_{R,k}^\mathsf{T} & \mu \mathbf{x}_{R,k}\mathbf{x}_{R,k}^\mathsf{T} + \mu \mathbf{x}_{I,k}\mathbf{x}_{I,k}^\mathsf{T} \end{bmatrix}
\end{aligned} \tag{8}$$

where we explicitly have split the noise into real and imaginary part $v_k = v_{R,k} + j v_{I,k}$.

For the RELMS algorithm:

$$\begin{aligned}
\begin{bmatrix} \tilde{\mathbf{w}}_{R,k} \\ \tilde{\mathbf{w}}_{I,k} \end{bmatrix} &= \mathbf{B}_k^{(RE)} \begin{bmatrix} \tilde{\mathbf{w}}_{R,k-1} \\ \tilde{\mathbf{w}}_{I,k-1} \end{bmatrix} - \mu \begin{bmatrix} \mathbf{x}_{R,k} v_{R,k} \\ -\mathbf{x}_{I,k} v_{R,k} \end{bmatrix} \\
\mathbf{B}_k^{(RE)} &= \begin{bmatrix} \mathbf{I} - \mu \mathbf{x}_{R,k}\mathbf{x}_{R,k}^\mathsf{T} & \mu \mathbf{x}_{R,k}\mathbf{x}_{I,k}^\mathsf{T} \\ \mu \mathbf{x}_{I,k}\mathbf{x}_{R,k}^\mathsf{T} & \mathbf{I} - \mu \mathbf{x}_{I,k}\mathbf{x}_{I,k}^\mathsf{T} \end{bmatrix}
\end{aligned} \tag{9}$$

while for the RXLMS algorithm we obtain

$$\begin{aligned}
\begin{bmatrix} \tilde{\mathbf{w}}_{R,k} \\ \tilde{\mathbf{w}}_{I,k} \end{bmatrix} &= \mathbf{B}_k^{(RX)} \begin{bmatrix} \tilde{\mathbf{w}}_{R,k-1} \\ \tilde{\mathbf{w}}_{I,k-1} \end{bmatrix} - \mu \begin{bmatrix} \mathbf{x}_{R,k} v_{R,k} \\ \mathbf{x}_{R,k} v_{I,k} \end{bmatrix} \\
\mathbf{B}_k^{(RX)} &= \begin{bmatrix} \mathbf{I} - \mu \mathbf{x}_{R,k}\mathbf{x}_{R,k}^\mathsf{T} & \mu \mathbf{x}_{R,k}\mathbf{x}_{I,k}^\mathsf{T} \\ -\mu \mathbf{x}_{R,k}\mathbf{x}_{I,k}^\mathsf{T} & \mathbf{I} - \mu \mathbf{x}_{R,k}\mathbf{x}_{R,k}^\mathsf{T} \end{bmatrix}
\end{aligned} \tag{10}$$

Following a classical MSE analysis [11, 12], we need to compute the parameter error vector covariance matrix

$$\mathbf{K}_k = \mathbb{E}\left[\begin{bmatrix} \tilde{\mathbf{w}}_{R,k} \\ \tilde{\mathbf{w}}_{I,k} \end{bmatrix}\begin{bmatrix} \tilde{\mathbf{w}}_{R,k} \\ \tilde{\mathbf{w}}_{I,k} \end{bmatrix}^{\mathsf{T}}\right] = \begin{bmatrix} \mathbf{K}_{\mathrm{RR,k}} & \mathbf{K}_{\mathrm{RI},k} \\ \mathbf{K}_{\mathrm{IR},k} & \mathbf{K}_{\mathrm{II},k} \end{bmatrix} \quad (11)$$

in dependence to the autocorrelation (acf) matrix of the input signal $\mathbf{R}_{x,R} = \mathbb{E}[\mathbf{x}_{R,k}\mathbf{x}_{R,k}^{\mathsf{T}}]$ and $\mathbf{R}_{x,I} = \mathbb{E}[\mathbf{x}_{I,k}\mathbf{x}_{I,k}^{\mathsf{T}}]$. To simplify matters we limit our analysis to the circular Gaussian case for which $\mathbf{R}_{x,R} = \mathbf{R}_{x,I} = \mathbf{R}$. Let us introduce the following matrix operators:

$$\begin{aligned} \mathbf{P}_a[\mathbf{A}] &= \mathbf{A} - 2\mu\mathbf{R}\mathbf{A} - 2\mu\mathbf{A}\mathbf{R} \\ &\quad + 2\mu^2[3\mathbf{R}\mathbf{A}\mathbf{R} + \mathbf{R}\mathrm{tr}(\mathbf{A}\mathbf{R})], && (12) \\ \mathbf{P}_b[\mathbf{A}] &= 2\mu^2[\mathbf{R}\mathrm{tr}(\mathbf{R}\mathbf{A}) - \mathbf{R}\mathbf{A}\mathbf{R}], && (13) \\ \mathbf{P}_c[\mathbf{A}] &= \mathbf{A} - \mu\mathbf{R}\mathbf{A} - \mu\mathbf{A}\mathbf{R} \\ &\quad + \mu^2[2\mathbf{R}\mathbf{A}\mathbf{R} + \mathbf{R}\mathrm{tr}(\mathbf{A}\mathbf{R})]. && (14) \end{aligned}$$

For the CLMS algorithm we obtain:

$$\begin{aligned} \mathbf{K}_{\mathrm{RR,k}} &= \mathbf{P}_a[\mathbf{K}_{\mathrm{RR,k-1}}] + \mathbf{P}_b[\mathbf{K}_{\mathrm{II,k-1}}] + \mu^2\sigma_v^2\mathbf{R} && (15) \\ \mathbf{K}_{\mathrm{RI},k} &= \mathbf{P}_a[\mathbf{K}_{\mathrm{RI},k-1}] - \mathbf{P}_b[\mathbf{K}_{\mathrm{IR},k-1}] && (16) \\ \mathbf{K}_{\mathrm{IR},k} &= \mathbf{P}_a[\mathbf{K}_{\mathrm{IR},k-1}] - \mathbf{P}_b[\mathbf{K}_{\mathrm{RI},k-1}] && (17) \\ \mathbf{K}_{\mathrm{II},k} &= \mathbf{P}_a[\mathbf{K}_{\mathrm{II},k-1}] + \mathbf{P}_b[\mathbf{K}_{\mathrm{RR,k-1}}] + \mu^2\sigma_v^2\mathbf{R}. && (18) \end{aligned}$$

As $\mathbf{K}_{\mathrm{RI},k}$ and $\mathbf{K}_{\mathrm{IR},k}$ have no driving force, they are irrelevant and only the first and last term need to be considered. In the RELMS and RXLMS algorithm, the terms look slightly different; we find:

$$\begin{aligned} \mathbf{K}_{\mathrm{RR},k} &= \mathbf{P}_c[\mathbf{K}_{\mathrm{RR},k-1}] + \mu^2\mathbf{R}\mathrm{tr}(\mathbf{K}_{\mathrm{II},k-1}\mathbf{R}) \\ &\quad + \frac{1}{2}\mu^2\sigma_v^2\mathbf{R} && (19) \\ \mathbf{K}_{\mathrm{II},k} &= \mathbf{P}_c[\mathbf{K}_{\mathrm{II},k-1}] + \mu^2\mathbf{R}\mathrm{tr}(\mathbf{K}_{\mathrm{RR},k-1}\mathbf{R}) \\ &\quad + \frac{1}{2}\mu^2\sigma_v^2\mathbf{R}. && (20) \end{aligned}$$

As before, $\mathbf{K}_{\mathrm{IR},k}$ and $\mathbf{K}_{\mathrm{RI},k}$ only depend on each other and have no driving term, thus they are irrelevant to the algorithmic behavior.

The algorithmic learning behavior can now be evaluated, following classic approaches. Diagonalizing the acf matrices $\Lambda = \mathbf{Q}\mathbf{R}\mathbf{Q}^H$ (correspondingly $\mathbf{P}_a[\mathbf{I}] \to \Lambda_a, \mathbf{P}_b[\mathbf{I}] \to \Lambda_b$), and extracting the diagonal terms of the matrices in vector form $\Lambda \to \lambda = [\Lambda_{11}, \Lambda_{22}, ..., \Lambda_{MM},]^{\mathsf{T}}, \mathbf{K}_{\mathrm{RR,k}} \to \mathbf{c}_{\mathrm{RR,k}} = [(\mathbf{K}_{\mathrm{RR,k}})_{11}, (\mathbf{K}_{\mathrm{RR,k}})_{22}, ..., (\mathbf{K}_{\mathrm{RR,k}})_{MM}]^{\mathsf{T}}$, ..., we obtain for the CLMS algorithm

$$\begin{bmatrix} \mathbf{c}_{\mathrm{RR,k}} \\ \mathbf{c}_{\mathrm{II},k} \end{bmatrix} = \begin{bmatrix} \Lambda_a & \Lambda_b \\ \Lambda_b & \Lambda_a \end{bmatrix}\begin{bmatrix} \mathbf{c}_{\mathrm{RR,k-1}} \\ \mathbf{c}_{\mathrm{II},k-1} \end{bmatrix} + \mu^2\sigma_v^2\begin{bmatrix} \lambda \\ \lambda \end{bmatrix}. \quad (21)$$

The RELMS and RXLMS algorithms deliver similar terms:

$$\begin{bmatrix} \mathbf{c}_{\mathrm{RR},k} \\ \mathbf{c}_{\mathrm{II},k} \end{bmatrix} = \begin{bmatrix} \Lambda_c & \mu^2\lambda\lambda^{\mathsf{T}} \\ \mu^2\lambda\lambda^{\mathsf{T}} & \Lambda_c \end{bmatrix}\begin{bmatrix} \mathbf{c}_{\mathrm{RR},k-1} \\ \mathbf{c}_{\mathrm{II},k-1} \end{bmatrix} + \frac{\mu^2}{2}\sigma_v^2\begin{bmatrix} \lambda \\ \lambda \end{bmatrix}. \quad (22)$$

Learning: Compared to the CLMS algorithm, the two variants must behave slower in learning as the terms linear in μ appear only with half their amount. While there is no analytical closed form for arbitrary correlation, it is possible to compute the behavior under uncorrelated input, $\mathbf{R} = \frac{1}{2}\mathbf{I}$. In this case we find $\Lambda_a = \mathbf{I}(1 - 2\mu + \frac{3}{2}\mu^2) + \frac{1}{2}\mu^2\mathbf{1}\mathbf{1}^{\mathsf{T}}$ and $\Lambda_b = \frac{1}{2}\mu^2(\mathbf{1}\mathbf{1}^{\mathsf{T}} - \mathbf{I})$, while the two variants exhibit $\Lambda_c = \mathbf{I}(1 - \mu + \frac{1}{2}\mu^2) + \frac{1}{4}\mu^2\mathbf{1}\mathbf{1}^{\mathsf{T}}$ and off-diagonal terms in μ^2. All relevant eigenvalues can be computed. Important for learning speed is the largest decisive eigenvalue, that is the eigenvalue obtained for eigenvector $\mathbf{1}$. It depends on the step-size μ and its smallest value is obtained for

$$\mu_{\mathrm{opt}} = \frac{1}{1+M}, \quad (23)$$

for all three algorithms. Also the stability limit at roughly twice the optimal step-size does not differ. However the maximum speed of learning differs as the same optimal step-size causes different values for CLMS and its variants. We find the largest decisive eigenvalue to be

$$\begin{aligned} \lambda_{*,\mathrm{LMS}} &= 1 - \mu_{\mathrm{opt}} && (24) \\ \lambda_{*,\mathrm{RE+RX}} &= 1 - \frac{1}{2}\mu_{\mathrm{opt}}, && (25) \end{aligned}$$

explaining that the two variants RELMS and RXLMS can only achieve half of the learning speed of the CLMS algorithm.

Steady-state: All algorithms end in the same steady state quality. While in Eq. (22) we find only half the noise power driving the dynamics of the system when compared to the CLMS algorithm shown in Eq. (21), the contributions on the linear part of the step-size are also only half. In steady-state these terms compensate each other and thus all three algorithms end up in the same steady-state values.

We restricted the analysis to the simplest case of Gaussian driving processes. Methods to extend the results under arbitrary density functions and correlations can be found in [13].

3. ROBUSTNESS ANALYSIS

Theorem 3.2 *The RELMS algorithm is as robust as the CLMS algorithm. The RXLMS algorithm is non-robust.*

Proof: We follow the approach of robustness analysis [12, 14–16]. Consider the RELMS algorithm in parameter error vector form:

$$\begin{aligned} \tilde{\mathbf{w}}_k &= \tilde{\mathbf{w}}_{k-1} - \mu\mathbf{x}_k^*\Re\{\tilde{e}_{a,k}\} && (26) \\ &= \tilde{\mathbf{w}}_{k-1} - \mu\mathbf{x}_k^*\Re\left\{\mathbf{x}_k^{\mathsf{T}}\tilde{\mathbf{w}}_{k-1}\right\} - \mu\mathbf{x}_k^*\Re\{v_k\}. \end{aligned}$$

Building the squared l_2-norm on both sides leads to

$$\begin{aligned} \|\tilde{\mathbf{w}}_k\|_2^2 &= \|\tilde{\mathbf{w}}_{k-1}\|_2^2 + \mu^2 \|\mathbf{x}_k\|^2 \Re^2 \{\tilde{e}_{a,k}\} \\ &\quad -2\mu\Re\{e_{a,k}\}\Re\{\tilde{e}_{a,k}\} \qquad (27) \\ &= \|\tilde{\mathbf{w}}_{k-1}\|_2^2 + \mu^2 \|\mathbf{x}_k\|^2 \Re^2 \{\tilde{e}_{a,k}\} \\ &\quad -\mu\left[\Re^2\{\tilde{e}_{a,k}\} + \Re^2\{e_{a,k}\} - \Re^2\{v_k\}\right], \end{aligned}$$

where we also introduced the undisturbed a priori error $e_{a,k} = \mathbf{x}_k^\mathsf{T}\tilde{\mathbf{w}}_{k-1}$, $\tilde{e}_{a,k} = e_{a,k} + v_k$. As long as $\mu^2\|\mathbf{x}_k\|^2 - \mu < 0$ we can conclude local robustness

$$\frac{\|\tilde{\mathbf{w}}_k\|_2^2 + \mu\Re^2\{e_{a,k}\}}{\|\tilde{\mathbf{w}}_{k-1}\|_2^2 + \mu\Re^2\{v_k\}} \le 1, \qquad (28)$$

which can easily be extended to global robustness:

$$\frac{\|\tilde{\mathbf{w}}_N\|_2^2 + \mu\sum_{k=1}^N \Re^2\{e_{a,k}\}}{\|\tilde{\mathbf{w}}_0\|_2^2 + \mu\sum_{k=1}^N \Re^2\{v_k\}} \le 1. \qquad (29)$$

Applying the small gain theorem as in [12, 15, 16] allows to prove that the algorithm behaves robustly for step-sizes $0 < \mu < 2/\|\mathbf{x}_k\|_2^2$. A consequence of this behavior is that there cannot be a single input sequence causing divergence. The RXLMS algorithm however is non-robust. A similar result was already obtained in [17] where an LMS algorithm was considered with corrupted regression vector. Taking only the real-part of the regression vector can be interpreted as such. As a consequence there exists input sequences that cause the algorithm to diverge even if the step-size condition is satisfied. Even arbitrarily small step-sizes can result in divergence.

4. SIMULATION RESULTS

In our first experiment we compare RELMS and RXLMS algorithms with CLMS behavior. We run Monte Carlo (MC) simulations (averaged over 1 000 runs) for filters of order $M = 20$ with white Gaussian noise as input signal. In Fig. 2 the obtained system mismatch (continuous lines) and their predicted values (dashed lines) are depicted for step-sizes $\mu = \alpha/(1+M)$, $\alpha = \{0.2, 0.5, 0.8, 1, 1.2, 1.5, 1.8\}$. We find excellent agreement with the theoretical values (dashed linear are basically invisible as hidden by continuous lines). The RXLMS algorithm is not shown here as it behaves identically to the RELMS algorithm.

In our second MC experiment we compare RXLMS and RELMS algorithms under worst case situations. We constructed the worst case regression vectors by random search. Although this does not guarantee to find the worst case sequence, it allows to clearly indicate if an algorithm can diverge by finding some very destructive sequences. Fig. 3 depicts the results when repeating the previous experiment for the two LMS variants. As expected, only the RELMS algorithm behaves robustly. Its learning rate is greatly hampered

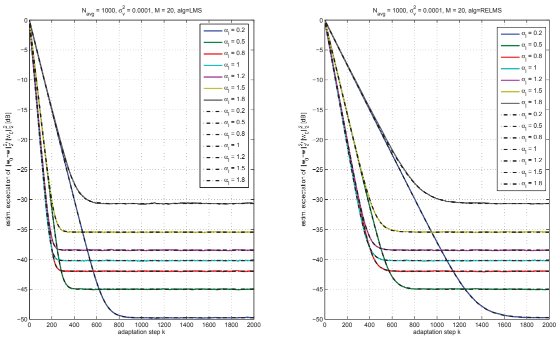

Fig. 2. *Learning curves of CLMS and RELMS (MC: continuous, theory: dashed). Both algorithms exhibit the same stability and steady-state, RELMS exhibits a slower learning rate.*

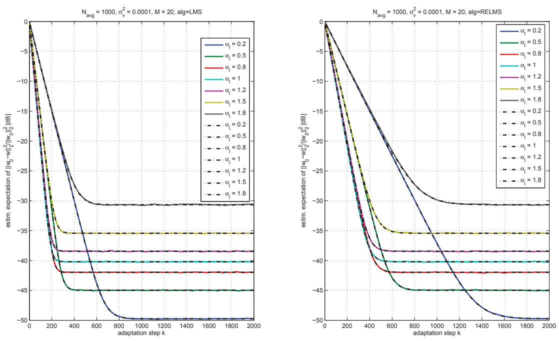

Fig. 3. *Worst case learning curves of RELMS (robust) and RXLMS (non robust).*

by the found worst case sequences but there is no indication of divergence. The situation is entirely different for the RXLMS algorithm. Finding sequences to make the RXLMS diverging was straightforward.

Experimental results in vehicles using an FXLMS algorithm with real-valued error are presented in [6], confirming our theoretical findings.

5. CONCLUSION

In this contribution we investigated two variants of the CLMS algorithm as they are being applied in control of active engine mounts to suppress disturbing engine vibrations. We clarified some principal behavior of applying a real-valued error term in complex-valued updates. Applying these variants to the FXLMS algorithm, that is the LMS algorithm with filtered-error term is expected to hamper the convergence speed but not to impact the robustness of the algorithm.

REFERENCES

[1] S.J. Elliott, *Signal Processing for Active Control*, Academic Press, London, San Diego, 2001.

[2] Sven Johansson, *Active Control of Propeller-Induced Noise in Aircraft*, Doctoral thesis, Blekinge Institute of Technology, Sweden, http://www.bth.se/fou/forskinfo.nsf/0/4c246ae6eb7bf99dc12569f10037a543/$FILE/PHDthesis_sjo.pdf, 2000.

[3] P.M.T. Fursdon, "The design and control of the avon active engine mount," in *Proc. of the 4th Styrian Noise, Vibration and Harshness Congress*, 2006, pp. 19/1–19/10.

[4] K. Tammi, "Active control of rotor vibrations by two feedforward control algorithms," *Journal of Dynamic Systems, Measurement and Control*, vol. 131, pp. 1–10, sep. 2009.

[5] A.J. Hillis, "Multi-input multi-output control of an automotive active engine mounting system," *Proc. of the Inst. of Mech. Eng., Part D: Journal of Automobile Engineering*, vol. 225, pp. 1492–1504, 2011.

[6] F. Hausberg, S. Vollmann, P. Pfeffer, S. Hecker, M. Ploechl, and T. Kolkhorst, "Improving the convergence behavior of active engine mounts in vehicles with cylinder-on-demand engines," in *Proc. of Internoise, Innsbruck, Austria*, Sep. 2013.

[7] M. Rupp and A.H. Sayed, "Robust FxLMS algorithm with improved convergence performance," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 1, pp. 78–85, Jan. 1998.

[8] S.J. Elliott, I.M. Stothers, and P.A. Nelson, "A multiple error LMS algorithm and its application to the active control of sound and vibration," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 35, pp. 1423–1434, 1987.

[9] B. Widrow, J. McCool, and M. Ball, "The complex LMS algorithm," *Proc. of the IEEE*, vol. 63, no. 4, pp. 719–720, 1975.

[10] E. Bjarnason, "Analysis of the filtered-X LMS algorithm," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP-93)*, 1993, vol. 3, pp. 511–514.

[11] S.Haykin, *Adaptive Filter Theory, 4.ed.*, Prentice Hall, 2002.

[12] A.H.Sayed, *Fundamentals of Adaptive Filtering*, Wiley, 2003.

[13] M. Rupp, "The LMS algorithm under arbitrary linearly filtered processes," in *Proc. EUSIPCO Conference, Barcelona*, Aug. 2011.

[14] 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.

[15] M. Rupp and A.H. Sayed, "A time-domain feedback analysis of filtered-error adaptive gradient algorithms," *IEEE Transactions on Signal Processing*, vol. 44, no. 6, pp. 1428–1440, Jun. 1996.

[16] A.H. Sayed and M. Rupp, *Chapter: Robustness Issues in Adaptive Filters*, CRC Press, 1998.

[17] R. Dallinger and M. Rupp, "A strict stability limit for adaptive gradient type algorithms," in *Conference Record of the Forty-Third Asilomar Conference on Signals, Systems, and Computers, Pacific Grove (CA)*, Nov. 2009, pp. 1370 – 1374.