ADAPTIVE LINEAR PREDICTION FILTERS BASED ON MAXIMUM A POSTERIORI ESTIMATION

Kristian T. Andersen[*†], *Toon van Waterschoot*[*], *Marc Moonen*[*]

[*]KU Leuven, ESAT/STADIUS, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
[†]Widex A/S, Nymøllevej 6, DK-3540 Lynge, Denmark

ABSTRACT

In this paper, we develop adaptive linear prediction filters in the framework of maximum a posteriori (MAP) estimation. It is shown how priors can be used to regularize the solution and references to known algorithms are made. The adaptive filters are suitable for implementation in real-time and by simulation with an adaptive line enhancer (ALE), it is shown how the parameters of the estimation problem affect the convergence of the adaptive filter. The adaptive line enhancer (ALE) is a widely used adaptive filter to separate periodic signals from additive background noise where it has traditionally been implemented using the least-mean-square (LMS) or recursive-least-square (RLS) filter. The derived algorithms can generally be used in any adaptive filter application with a desired target signal.

Index Terms— Maximum a posteriori; adaptive filters; linear prediction; regularization; adaptive line enhancer.

1. INTRODUCTION

In this paper, the problem of finding the optimal linear prediction coefficients in an adaptive filter of $N-1$ order is considered in the context of maximum a posteriori (MAP) estimation. The adaptive filters are recursively estimated over a frame of M samples and are derived for both the overdetermined ($M > N$) and underdetermined ($M < N$) case where priors are used to regularize the solution. The proposed algorithms can more generally be applied to any adaptive filter and MAP adaptive filters were also considered in [1], where a uniform prior was assumed and only the case $M = 1$ was considered. Optimally regularized adaptive filters has been widely studied, see e.g. [2] and the references therein. The main novelty of the proposed algorithms w.r.t. those reported in [2] is the use of two regularization terms. Throughout the paper, references will be made to well-known adaptive filter algorithms.

The proposed algorithms are applied in an adaptive line enhancer (ALE) [3]. The ALE is a type of adaptive filter used to separate a periodic signal $y(n)$ from additive background noise $e(n)$ and has found applications in a range of different fields. The adaptive filter predicts the $y(n)$ component in the observed signal $x(n) = y(n)+e(n)$ over a prediction horizon of K samples, leading to a system of equations given as:

$$\boldsymbol{x}_M = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{e} \tag{1}$$

where $\boldsymbol{x}_M \triangleq [x(n), x(n-1), ..., x(n-M+1)]^T$, $\boldsymbol{X} \triangleq [\boldsymbol{x}_N(n-K), \boldsymbol{x}_N(n-K-1), ..., \boldsymbol{x}_N(n-K-M+1)]^T$, $\boldsymbol{w}$ is the filter that will be used to predict $y(n)$ and $\boldsymbol{e} \triangleq [e(n), e(x-1), ..., e(x-M+1)]^T$.

The paper is organized as follows. In section 2, the linear prediction problem is formulated as MAP estimation in the framework of Bayesian learning. Section 3 contains the optimization of the linear prediction problem using the assumption of Gaussian distributions, while section 4 describes how the optimization can be changed to instead assume a Laplace distribution. Section 5 contains some experimental results using the derived equations and the conclusion can be found in section 6.

2. LINEAR PREDICTION AS MAP ESTIMATION

Following Bayesian learning [4], we will consider observations and filter coefficients to be stochastic variables and estimate the current filter coefficients by maximizing the *a posteriori* probability given by Bayes rule:

$$p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{x}_M, \boldsymbol{w}_{old}) = \frac{p(\boldsymbol{X}, \boldsymbol{x}_M, \boldsymbol{w}_{old}|\boldsymbol{w})p(\boldsymbol{w})}{p(\boldsymbol{X}, \boldsymbol{x}_M, \boldsymbol{w}_{old})} \tag{2}$$

where $\boldsymbol{w}_{old}$ is a set of old filter coefficients, $p(\boldsymbol{X}, \boldsymbol{x}_M, \boldsymbol{w}_{old}|\boldsymbol{w})$ is the likelihood, $p(\boldsymbol{w})$ is the prior and $p(\boldsymbol{X}, \boldsymbol{x}_M, \boldsymbol{w}_{old})$ is the evidence. In the context of adaptive filters, the old filter coefficients are the MAP estimates from the previous sample or frame depending on how often the MAP estimates are updated.

Assuming that the old filter coefficients and the current observations are independent given the new filter coefficients, we can factorize the likelihood:

$$p(\boldsymbol{X}, \boldsymbol{x}_M, \boldsymbol{w}_{old}|\boldsymbol{w}) = p(\boldsymbol{X}, \boldsymbol{x}_M|\boldsymbol{w})p(\boldsymbol{w}_{old}|\boldsymbol{w}) \tag{3}$$

Also, as the normalization $p(\boldsymbol{X}, \boldsymbol{x}_M, \boldsymbol{w}_{old})$ is independent of $\boldsymbol{w}$, it has no significance for the maximum of the posterior:

$$p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{x}_M, \boldsymbol{w}_{old}) \propto p(\boldsymbol{X}, \boldsymbol{x}_M|\boldsymbol{w})p(\boldsymbol{w}_{old}|\boldsymbol{w})p(\boldsymbol{w}) \tag{4}$$

Since the logarithm is a monotonically increasing function, the MAP estimates can be found by solving the following unconstrained minimization problem:

$$\underset{\boldsymbol{w}}{\text{minimize}} \quad \ell(\boldsymbol{w}) \tag{5}$$

where:

$$\begin{aligned}\ell(\boldsymbol{w}) &= -log\big[p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{x}_M, \boldsymbol{w}_{old})\big] \\ &\propto -log\big[p(\boldsymbol{X}, \boldsymbol{x}_M|\boldsymbol{w})\big] - log\big[p(\boldsymbol{w}_{old}|\boldsymbol{w})\big] - log\big[p(\boldsymbol{w})\big] \\ &= \ell_{\boldsymbol{X},\boldsymbol{x}_M|\boldsymbol{w}}(\boldsymbol{w}) + \ell_{\boldsymbol{w}_{old}|\boldsymbol{w}}(\boldsymbol{w}) + \ell_{\boldsymbol{w}}(\boldsymbol{w}) \end{aligned} \tag{6}$$

3. MAP USING GAUSSIAN DISTRIBUTIONS

Using (1) and assuming that $e(n)$ has a zero-mean Gaussian distribution, the first factor in the factorized likelihood in (4) is given by:

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{x}_M) &= p(\boldsymbol{X}, \boldsymbol{x}_M|\boldsymbol{w}) \\ &= \frac{1}{\sqrt{(2\pi)^M|\Sigma_e|}} \exp^{-\frac{1}{2}(\boldsymbol{x}_M - \boldsymbol{X}\boldsymbol{w})^T\Sigma_e^{-1}(\boldsymbol{x}_M - \boldsymbol{X}\boldsymbol{w})} \end{aligned} \tag{7}$$

where $\Sigma_e = \mathbb{E}[\boldsymbol{e}\boldsymbol{e}^T] = \text{diag}[\sigma_{\boldsymbol{e}}(n)^2, \sigma_{\boldsymbol{e}}(n-1)^2, ..., \sigma_{\boldsymbol{e}}(n-M+1)^2]$ is the covariance matrix of $\boldsymbol{e}$, $\mathbb{E}[\cdot]$ is the expectation operator and $|\cdot|$ is the determinant, which is equal to the trace for a diagonal matrix.

The first factor in the negative log-likelihood is:

$$\begin{aligned}\ell_{\boldsymbol{X},\boldsymbol{x}_M|\boldsymbol{w}}(\boldsymbol{w}) &= -\log \mathcal{L}(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{x}_M) \\ = \frac{1}{2}\log\big[(2\pi)^M|\Sigma_e|\big] &+ \frac{1}{2}(\boldsymbol{x}_M - \boldsymbol{X}\boldsymbol{w})^T\Sigma_e^{-1}(\boldsymbol{x}_M - \boldsymbol{X}\boldsymbol{w}) \\ &\propto \frac{1}{2}(\boldsymbol{x}_M - \boldsymbol{X}\boldsymbol{w})^T\Sigma_e^{-1}(\boldsymbol{x}_M - \boldsymbol{X}\boldsymbol{w}) \end{aligned} \tag{8}$$

Similarly, again assuming Gaussian distributions, we define the following:

$$\ell_{\boldsymbol{w}_{old}|\boldsymbol{w}}(\boldsymbol{w}) \propto \frac{1}{2}(\boldsymbol{w} - \boldsymbol{w}_{old})^T\Sigma_\Delta^{-1}(\boldsymbol{w} - \boldsymbol{w}_{old}) \tag{9}$$

$$\ell_{\boldsymbol{w}}(\boldsymbol{w}) \propto \frac{1}{2}\boldsymbol{w}^T\Sigma_w^{-1}\boldsymbol{w} \tag{10}$$

where $\Sigma_\Delta = \mathbb{E}[(\boldsymbol{w} - \boldsymbol{w}_{old})(\boldsymbol{w} - \boldsymbol{w}_{old})^T]$ and $\Sigma_w = \mathbb{E}[\boldsymbol{w}\boldsymbol{w}^T]$ are covariance matrices that determine how far from $\boldsymbol{w}_{old}$ the new filter coefficients $\boldsymbol{w}$ are expected to be and the prior expectation of $\boldsymbol{w}$, respectively. Since $\ell_{\boldsymbol{X},\boldsymbol{x}_M|\boldsymbol{w}}(\boldsymbol{w})$, $\ell_{\boldsymbol{w}_{old}|\boldsymbol{w}}(\boldsymbol{w})$ and $\ell_{\boldsymbol{w}}(\boldsymbol{w})$ are quadratic functions in $\boldsymbol{w}$ and the covariance matrices are positive semi-definite, the problem in (5) with Gaussian distributions is an unconstrained convex quadratic optimization problem.

A necessary and sufficient condition for the global minimum of an unconstrained convex quadratic optimization problem is that the gradient is equal to zero:

$$\frac{\partial \ell(\boldsymbol{w})}{\partial \boldsymbol{w}} = 0 \tag{11}$$

A closed form solution to this equation will be given in the following.

3.1. Minimization of $\ell(\boldsymbol{w})$ using Gaussian distributions

The solution to (11) with Gaussian distributions is given by:

$$\Big(\Sigma_w^{-1} + \Sigma_\Delta^{-1} + \boldsymbol{X}^T\Sigma_e^{-1}\boldsymbol{X}\Big)\boldsymbol{w} = \Sigma_\Delta^{-1}\boldsymbol{w}_{old} + \boldsymbol{X}^T\Sigma_e^{-1}\boldsymbol{x}_M \tag{12}$$

In the following, we consider the two cases $M > N$ and $M < N$ separately, as they have different interpretations. When $M > N$, $\tilde{\Phi} = \boldsymbol{X}^T\Sigma_e^{-1}\boldsymbol{X}$ has full rank and it becomes meaningful to interpret $\tilde{\Phi}$ and $\tilde{\gamma} = \boldsymbol{X}^T\Sigma_e^{-1}\boldsymbol{x}_M$ as the weighted covariance matrix and cross-covariance vector respectively. This makes it possible to write the solution as a function of $\tilde{\Phi}$ and $\tilde{\gamma}$, which can then be estimated using different assumptions, for instance using the assumption that $\tilde{\Phi}$ is a Toeplitz matrix as in the auto-correlation method [5]. When $M < N$, $\tilde{\Phi}$ is a singular matrix and the interpretation of a weighted covariance matrix and cross-covariance vector becomes less meaningful. In this case there are also more computationally efficient ways to calculate $\boldsymbol{w}$. We assume that Σ_w and Σ_Δ have full rank, which is often the case in practice. If this is not the case, the inverse of the singular matrices can be calculated as the pseudo-inverse using singular value decomposition (SVD), which will increase the computational complexity.

Case $M > N$: Isolating $\boldsymbol{w}$ in (12) and performing some matrix manipulations gives:

$$\boldsymbol{w} \;=\; \boldsymbol{w}_\delta \;+\; \big(\Sigma_w^{-1} \;+\; \Sigma_\Delta^{-1} \;+\; \tilde{\Phi}\big)^{-1}\big(\tilde{\gamma} \;-\; \tilde{\Phi}\boldsymbol{w}_\delta\big) \tag{13}$$

where $\boldsymbol{w}_\delta = \Sigma_w\big(\Sigma_w + \Sigma_\Delta\big)^{-1}\boldsymbol{w}_{old}$. The equation is a recursive equation that updates $\boldsymbol{w}$ to be somewhere on the line from $\boldsymbol{w}_\delta$ to a solution that is regularized by both Σ_w and Σ_Δ. $\boldsymbol{w}_\delta$ is a leaky version of $\boldsymbol{w}_{old}$ where the leak depends on the ratio between Σ_w and Σ_Δ. Note that the leak is not only on the first $\boldsymbol{w}_{old}$ as in the usual leaky-LMS but also inside the error term $\big(\tilde{\gamma} - \tilde{\Phi}\boldsymbol{w}_\delta\big)$. This form reveals that the MAP optimization problem with Gaussian distributions can be seen to be a Newton-type optimization. To determine the effect of Σ_w alone, we write (13) for $\Sigma_\Delta \to \infty I$, leading to $\boldsymbol{w}_\delta = 0$ and hence:

$$\boldsymbol{w} = \big(\Sigma_w^{-1} + \tilde{\Phi}\big)^{-1}\tilde{\gamma} \tag{14}$$

It is seen that Σ_w^{-1} acts as a regularization, where the lower the prior variance of $\boldsymbol{w}$, the more the solution is regularized towards zero.

Conversely, to determine the effect of Σ_Δ alone, we write (13) for $\Sigma_w \to \infty I$:

$$\boldsymbol{w} = \boldsymbol{w}_{old} + \big(\Sigma_\Delta^{-1} + \tilde{\Phi}\big)^{-1}\big(\tilde{\gamma} - \tilde{\Phi}\boldsymbol{w}_{old}\big) \tag{15}$$

It is seen that in this form, Σ_Δ^{-1} acts as a damping factor to the iterative solution as in the Levenberg-Marquardt optimization algorithm [6].

When letting both $\Sigma_\Delta \to \infty I$ and $\Sigma_w \to \infty I$, the solution in (13) is a weighted linear least-squares (LLS) solution:

$$\boldsymbol{w} = \tilde{\Phi}^{-1}\tilde{\gamma} = \Big(\boldsymbol{X}^T\Sigma_e^{-1}\boldsymbol{X}\Big)^{-1}\boldsymbol{X}^T\Sigma_e^{-1}\boldsymbol{x}_M \tag{16}$$

Case $M < N$: Using the matrix inversion lemma in (12), the solution can also be written as:

$$\boldsymbol{w} = \boldsymbol{w}_\delta + \Sigma_\delta \boldsymbol{X}^T (\Sigma_e + \boldsymbol{X}\Sigma_\delta \boldsymbol{X}^T)^{-1}(\boldsymbol{x}_M - \boldsymbol{X}\boldsymbol{w}_\delta) \quad (17)$$

where $\boldsymbol{w}_\delta = \Sigma_w \big(\Sigma_w + \Sigma_\Delta\big)^{-1} \boldsymbol{w}_{old}$ and $\Sigma_\delta = (\Sigma_w^{-1} + \Sigma_\Delta^{-1})^{-1}$. When $M < N$, this solution is more computationally efficient than (13) and leads to a generalized type of affine projection adaptive filter [7].

As before, to determine the effect of Σ_w, we write (17) for $\Sigma_\Delta \rightarrow \infty I$:

$$\boldsymbol{w} = \Sigma_w \boldsymbol{X}^T (\Sigma_e + \boldsymbol{X}\Sigma_w \boldsymbol{X}^T)^{-1} \boldsymbol{x}_M \quad (18)$$

It is seen that for this case, Σ_e is a regularization towards zero to the underdetermined solution and is weighted against Σ_w.

To determine the effect of Σ_Δ, we write (17) for $\Sigma_w \rightarrow \infty I$:

$$\boldsymbol{w} = \boldsymbol{w}_{old} + \Sigma_\Delta \boldsymbol{X}^T (\Sigma_e + \boldsymbol{X}\Sigma_\Delta \boldsymbol{X}^T)^{-1}(\boldsymbol{x}_M - \boldsymbol{X}\boldsymbol{w}_{old}) \quad (19)$$

which is a weighting between $\boldsymbol{w}_{old}$ and the underdetermined solution.

Of particular interest is the solution for $M = 1$:

$$\boldsymbol{w} = \boldsymbol{w}_{old} + \frac{\Sigma_\Delta \boldsymbol{x}_N}{\sigma_e^2 + \boldsymbol{x}_N^T \Sigma_\Delta \boldsymbol{x}_N} \big(x(n) - \boldsymbol{x}_N^T \boldsymbol{w}_{old}\big) \quad (20)$$

It is seen that this has the form of the RLS algorithm but with a different interpretation based on the statistical properties of $\boldsymbol{w}$ and $\boldsymbol{e}$. The equation also shows strong similarities to the Kalman filter [8]. The matrix Σ_Δ determines how fast the filter coefficients can be adapted similar to proportionate adaptive filters [9]. If instead, $\Sigma_\Delta \triangleq \Delta^2 I$, where $\Delta^2 >> \sigma_e^2$ so σ_e^2 can be ignored, and a step-size parameter μ is introduced to the update equation, then (20) becomes:

$$\boldsymbol{w} = \boldsymbol{w}_{old} + \frac{\mu \boldsymbol{x}_N}{\boldsymbol{x}_N^T \boldsymbol{x}_N} \big(x(n) - \boldsymbol{x}_N^T \boldsymbol{w}_{old}\big) \quad (21)$$

which is the normalized least-mean squares (NLMS) algorithm [7]. The step-size $/mu$ is a proportional factor to the update step and for $0 < \mu \leq 1$ the update equation takes a step that is somewhere between $\boldsymbol{w}_{old}$ and the solution to the MAP estimation problem.

4. MAP USING LAPLACE DISTRIBUTIONS

In section 3, the filter coefficients were derived under the assumption of Gaussian distributions. Other distributions can be used in a similar fashion and in this section, we consider the Laplace distribution for the error. The Laplace distribution is univariate, and although different generalizations exist to convert it to a multivariate distribution, in the following we consider each variable to be independent for the sake of simplicity, so the Laplace distribution of a vector $\boldsymbol{e}$ can be written as:

$$p(\boldsymbol{e}) = \prod_i \frac{1}{2b_i} \exp^{-\frac{|e_i - a_i|}{b_i}} \quad (22)$$

The Laplace distribution has heavier tails than the Gaussian distribution and is therefore often used to promote more sparse or robust estimates. Using the same procedure as in section 3, the negative log-likelihood can be written as:

$$\ell_{\boldsymbol{e}|\boldsymbol{a}} \propto \boldsymbol{B}^{-1} |\boldsymbol{e} - \boldsymbol{a}| \quad (23)$$

where $\boldsymbol{B} = \text{diag}[b_0, ..., b_{M-1}]$. The negative log-likelihood of the Laplace distribution is convex, except at the minimum $\boldsymbol{e} = \boldsymbol{a}$ where the gradient is not defined. Since it only contains first-order information, there is no local information about how close a variable $\boldsymbol{e}^*$ is to the minimum and therefore the solution to the optimization problem can not be written in closed form. The Laplace distribution corresponds to an L1-norm (and/or L1-regularized) estimation problem and in the context of linear prediction, the use of an L1-norm prior has been termed sparse linear prediction, see e.g. [10, 11]. A gradient descent algorithm could be used, coupled with a search along the gradient direction:

$$\boldsymbol{w} = \boldsymbol{w}_{old} - \mu \frac{\partial \ell(\hat{\boldsymbol{w}})}{\partial \hat{\boldsymbol{w}}} |_{\hat{\boldsymbol{w}} = \boldsymbol{w}_{old}} \quad (24)$$

where μ is the step-size. Assuming a Laplace distribution for $\ell_{\boldsymbol{X}, \boldsymbol{x}_M | \boldsymbol{w}}(\boldsymbol{w})$ would in this case give:

$$\boldsymbol{w} = \boldsymbol{w}_{old} + \mu \boldsymbol{X}^T \boldsymbol{B}^{-1} sgn\big(\boldsymbol{x}_M - \boldsymbol{X}\boldsymbol{w}_{old}\big) \quad (25)$$

where $sgn(*)$ is the sign function. For $M = 1$, this is the sign-LMS algorithm [12] where the minimum of the optimization problem is only reached if μ is chosen appropriately.

Assuming a Gaussian distribution for $\ell_{\boldsymbol{w}_{old} | \boldsymbol{w}}(\boldsymbol{w})$ and a Laplace distribution for $\ell_{\boldsymbol{X}, \boldsymbol{x}_M | \boldsymbol{w}}(\boldsymbol{w})$, and solving the corresponding MAP optimization problem, by setting the gradient equal to zero and using the matrix inversion lemma gives:

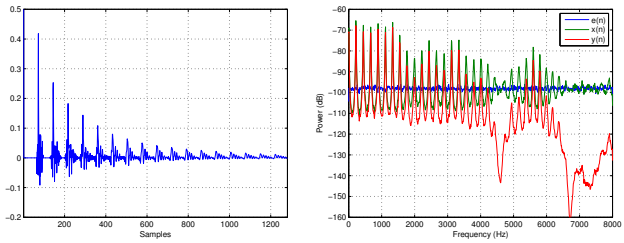
$$\boldsymbol{w} = \boldsymbol{w}_{old} + \Sigma_\Delta \boldsymbol{X}^T \boldsymbol{B}^{-1} sgn\big(\boldsymbol{x}_M - \boldsymbol{X}\boldsymbol{w}\big) \quad (26)$$

Assuming that the filter has converged, so we can approximate $\boldsymbol{w} \approx \boldsymbol{w}_{old}$, and that $\Sigma_\Delta = \Delta I$ so $\mu = \Delta$ is a scalar, then the sign-LMS follows.

5. EXPERIMENTAL RESULTS

In this section, we provide some simulations of an ALE. We estimate the prediction coefficients from a voiced speech sample and then excite a filter with Gaussian white noise using the estimated coefficients to generate a signal that is then used in the ALE. First, we consider a case where the used model fits the data and convergence to the true filter coefficients can be measured. Secondly, we investigate a case where the filter is undermodelled using a higher-order AR model, which is a more realistic scenario for a speech signal. As $M = 1$ is the most popular case in adaptive filters, it will be used in all simulations in this section.

To create a test signal $x(n)$ with known filter coefficients, we first estimate the coefficients from a voiced speech sample



(a) Impulse response of filter. (b) Power spectrum of signals.

Fig. 1. Test signal with known filter coefficients.

with a sample rate of 16kHz using (16) with $N = 32$ and $K = 64$ and then create $x(n)$ as:

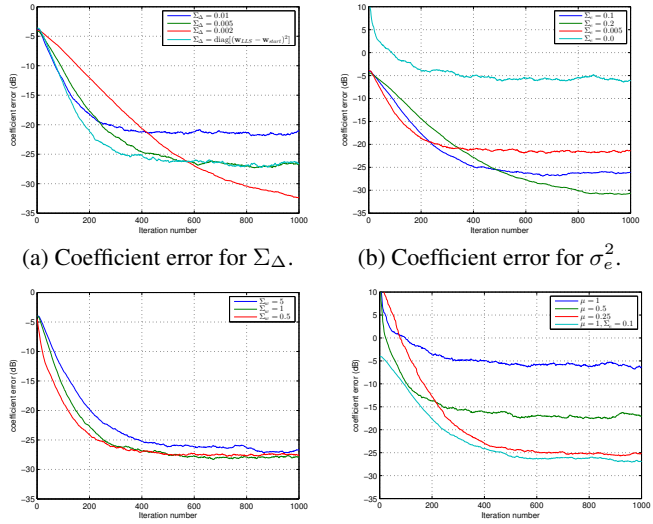
$$x(n) = y(n) + e(n) = \boldsymbol{w}_{true}^T \boldsymbol{x}_N(n-K) + e(n) \quad (27)$$

where $e(n)$ is a white noise signal with variance $\sigma_e^2 = 0.1$ and $\boldsymbol{w}_{true}$ are the true filter coefficients. $\boldsymbol{w}_{true}$ is scaled so the input signal-to-noise ratio $SNR_i = 0$, where SNR_i is given by:

$$SNR_i = 10 \log \frac{\sum_n y(n)^2}{\sum_n e(n)^2} \quad (28)$$

The impulse response of the filter that generates $x(n)$ and the power spectrum of the signals can be seen in Figure 1.

The adaptive filter was evaluated by initializing the coefficients to random values with a standard deviation of 0.1 and then measuring the energy difference between $\boldsymbol{w}_{true}$ and the coefficients of the adaptive filter for each sample. The experiment was repeated for 100 trials with different starting coefficients and noise $e(n)$ for each trial and the error was averaged over the trials. The parameters were initially set to $\Sigma_\Delta = 0.005I$, $\sigma_e^2 = 0.1$ and $\Sigma_w = 5I$. A simple test was done by changing the diagonal values and the experimental results can be seen in Figure 2. Testing with Σ_Δ revealed that this parameter could be used to tradeoff convergence speed with final coefficient error. This is expected as using larger values means the filter can adapt faster, while also making the filter having larger oscillations around the optimal coefficients. Choosing values that are relative to how far the current coefficients are to the true values makes the filter converge faster as seen for $\Sigma_\Delta = \text{diag}[(\boldsymbol{w}_{LLS} - \boldsymbol{w}_{start})^2]$ where $\boldsymbol{w}_{start}$ are the coefficients the filter is initialized with. Σ_Δ had in this case an upper bound of 0.005. A similar pattern is observed for σ_e^2, which acts as a kind of regularization to the MAP estimate. This can slow down the adaptation but eventually results in a more accurate result. Completely disregarding the noise, by setting $\sigma_e^2 = 0$ as in the NLMS, however, severely degrades the performance. Reducing all the diagonal values in Σ_w reduces the performance, since some coefficients should have large values. In the plot of coefficient error for Σ_w, the 3 diagonal values for the coefficients near the fundamental pitch were fixed at 5, while the rest of the diagonal values were reduced, which is seen to result in a



(a) Coefficient error for Σ_Δ. (b) Coefficient error for σ_e^2.

(c) Coefficient error for Σ_w. (d) Coefficient error for μ.

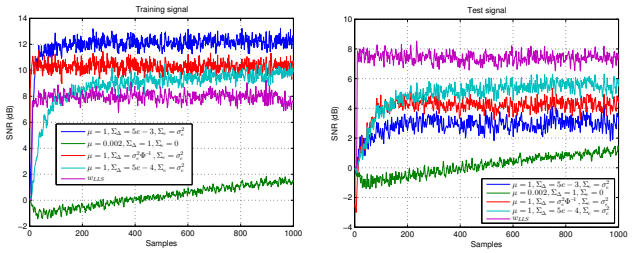
Fig. 2. The coefficient error for the adaptive filter for different parameters.

faster convergence. To test the NLMS algorithm, a step-size μ was introduced and different values tested while keeping $\sigma_e^2 = 0$. It is seen that reducing μ greatly improves the convergence, but in all cases the start of the error curve is higher than when the noise is taken into account by setting $\sigma_e^2 = 0.1$.

In a second experiment a 96th order AR filter is fitted to the same speech sample with $K = 1$ and then applied to a white noise signal. The signal is then mixed with an additional white noise signal at a SNR of 10dB. This resulting signal is then used as input to the adaptive filter estimating 64 samples ahead with a 31st order filter, i.e. $K = 64$ and $N = 32$. Due to the fact that the adaptive filter is undermodelled, it can not predict the AR model perfectly and the ideal performance of the adaptive filter is therefore less than the 10dB SNR that the signal is mixed with. To get an upper bound on the performance of the adaptive filter, the LLS solution $\boldsymbol{w}_{LLS}$ is calculated over the entire signal using (16) and used as comparison. The performance is measured by calculating the energy of the true sample $x(n)^2$ and the energy of the difference between the true and predicted sample $(x(n) - \hat{x}(n))^2$, repeating the prediction for 1000 trials and calculating the average SNR for each sample as:

$$SNR(n) = \frac{\sum_i x_i(n)^2}{\sum_i (x_i(n) - \hat{x}_i(n))^2} \quad (29)$$

where the index i corresponds to the trial number. Different adaptive filters evaluated on this signal can be seen in Figure 3. The blue curve is the MAP adaptive filter that was used in the previous experiment and it is seen that it has the highest SNR. The red curve corresponds to a RLS filter, where Φ has been estimated from the data and has a faster con-



(a) Training SNR. (b) Test SNR.

Fig. 3. Training and test results for the undermodelled case.

vergence. The green curve corresponds to the NLMS filter and $\mu = 0.002$ was set experimentally to the highest possible value without the filter diverging. The cyan curve is a MAP filter with lower Σ_Δ, which has a slower convergence. It is seen that all MAP filters have a higher SNR than the $\boldsymbol{w}_{LLS}$ filter for the training case, which indicates that they are overfitting the model to the data. Using the adaptive filter coefficients from this set of signal trials on a different set of signal trials calculated with the same AR model reveals that this is indeed the case as seen in the test case. The $\boldsymbol{w}_{LLS}$ filter has a similar SNR for this test case, which indicates that it has captured the underlying model, while the MAP filters have much lower SNR values than for the set of training signals. The blue curve performs the worst of the MAP filters, while the green curve performs better than the red RLS filter.

6. CONCLUSION

Adaptive filters have been derived in this paper corresponding to maximum a posteriori estimates using Gaussian or Laplacian distributions. The filters have been derived in the context of linear prediction, but can be applied in any adaptive filtering application. Two regularization terms have been included as probability distributions on the filter coefficients. References to known algorithms have been made and it has been shown how the derived filters relate to well known adaptive filters. Using experimental simulations it has been shown how the parameters affect the convergence.

7. ACKNOWLEDGEMENTS

The authors would like to thank Jens Brehm Nielsen at Widex for insightful discussions and feedback concerning Bayesian learning.

REFERENCES

[1] D. Y. Huang, S. Rahardja, and H. Huang, "Maximum a posteriori based adaptive algorithms," in *Signals, Systems and Computers, 2007. ACSSC 2007. Conference Record of the Forty-First Asilomar Conference on*, Nov 2007, pp. 1628–1632.

[2] T. van Waterschoot, G. Rombouts, and M. Moonen, "Optimally regularized adaptive filtering algorithms for room acoustic signal enhancement," *Signal Processing*, vol. 88, no. 3, pp. 594–611, Mar 2008.

[3] B. Widrow, Jr. Glover, J.R., J.M. McCool, J. Kaunitz, C.S. Williams, R.H. Hearn, J.R. Zeidler, Jr. Eugene Dong, and R.C. Goodlin, "Adaptive noise cancelling: Principles and applications," *Proceedings of the IEEE*, vol. 63, no. 12, pp. 1692–1716, Dec 1975.

[4] Christopher M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[5] John R Deller, John H L Hansen, and John G Proakis, *Discrete-time processing of speech signals*, Institute of Electrical and Electronics Engineers, New York, 2000, Originally published: New York : Macmillan, 1993.

[6] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer series in operations research and financial engineering. Springer, New York, NY, 2. ed. edition, 2006.

[7] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall information and system sciences series. Prentice Hall, 2002.

[8] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice-Hall, Upper Saddle River, NJ, 1993.

[9] Zhe Chen, Steven L Gay, and Simon Haykin, "Proportionate adaptation: New paradigms in adaptive filters," *Least-Mean-Square Adaptive Filters*, pp. 293–334, 2003.

[10] D. Giacobello, M.G. Christensen, M.N. Murthi, S.H. Jensen, and M. Moonen, "Sparse linear prediction and its applications to speech processing," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 5, pp. 1644–1657, July 2012.

[11] T. L. Jensen, D. Giacobello, T. van Waterschoot, and M. G. Christensen, "Fast algorithms for high-order sparse linear prediction with applications to speech processing," *Speech Communication, submitted for publication*, Jan 2015.

[12] Ali H. Sayed, *Fundamentals of Adaptive Filtering*, Wiley-IEEE Press, 1 edition, June 2003.