

ANALOG FILTER ADAPTATION USING A DITHERED LINEAR SEARCH ALGORITHM

Anthony Chan Carusone and David A. Johns

Email: tcc@eecg.utoronto.ca

Dept. of Electrical and Computer Engineering, University of Toronto
10 King's College Rd., Toronto, CANADA, M5S 3G4

ABSTRACT

A variation of the differential steepest descent algorithm, here called the dithered linear search (DLS), is examined and applied to analog filter adaptation. The DLS algorithm is a gradient descent optimizer with a straightforward and robust hardware implementation. Gradient estimates are obtained by applying independent additive dither to all of the filter's parameters simultaneously and correlating the resulting changes in the output squared error to the dither signals. Unlike the popular LMS algorithm, the DLS algorithm does not require access to the filter's internal states. No additional analog hardware is required making it ideal for adaptive analog filters in mixed-signal systems. A theoretical analysis shows no gradient misalignment. The algorithm is verified on an integrated analog filter. The effects of dc offsets are also examined.

1. INTRODUCTION

The LMS algorithm is currently the most popular technique for digital filter adaptation. Unfortunately, implementation of the LMS algorithm in analog adaptive filters is challenging. Dc offsets on the analog signals can prevent accurate convergence [1], [2], and significant additional analog hardware may be required to obtain the gradient signals [3].

In this paper, a general technique for analog filter adaptation is discussed with a straightforward hardware implementation that is robust with respect to dc offsets. The dithered linear search (DLS) technique does not require access to the filter's internal states and no additional analog hardware is required. All filter parameters may be adapted simultaneously and independently of one another.

In Section 2 of this paper, some background theory is presented. In Section 3, the DLS algorithm is introduced. A theoretical analysis showing that there is no gradient misalignment using the DLS algorithm is performed in Section 4. Behavioral simulation results are then presented in Section 5. Dc offset effects are considered in Section 6 and experimental results from a continuous-time integrated analog filter are presented in Section 7.

2. BACKGROUND

Consider a completely general analog filter (continuous-time or discrete-time) with N variable parameters $\mathbf{p} = [p_1 \ p_2 \ \dots \ p_N]^T$. Let ε denote an error function used to quantify the filter's performance so that ε is smallest when the filter parameters are equal to their optimal values. It is common to use the mean squared error (MSE) in the filter output, y , with respect to some desired or reference signal, d , as the error function:

$$\varepsilon(\mathbf{p}) = E[(d - y)^2] = E[e^2] \quad (1)$$

Gradient descent optimizers proceed by updating the filter's parameters iteratively in a direction opposite the gradient $\nabla_{\mathbf{p}}\varepsilon = [\partial\varepsilon/\partial p_1 \ \partial\varepsilon/\partial p_2 \ \dots \ \partial\varepsilon/\partial p_N]^T$. The iterative update rule is,

$$\mathbf{p}(k+1) = \mathbf{p}(k) - \mu \cdot \nabla_{\mathbf{p}}\varepsilon, \quad (2)$$

where μ is a parameter controlling the rate of adaptation. The only problem is, how to obtain the gradient $\nabla_{\mathbf{p}}\varepsilon$. Generally the exact value of $\nabla_{\mathbf{p}}\varepsilon$ can not be determined, so an unbiased estimate of the gradient is used instead,

$$\hat{\nabla}_{\mathbf{p}}\varepsilon = \left[\begin{matrix} \hat{\partial\varepsilon} \\ \hat{\partial p_1} \\ \hat{\partial p_2} \\ \dots \\ \hat{\partial p_N} \end{matrix} \right]^T \quad (3)$$

Different approaches to obtaining the gradient estimates define different gradient descent algorithms. Perhaps the most obvious way to estimate the gradient is to make a direct measurement of each component by perturbing the parameters one at a time symmetrically around their current values and measuring the resulting change in the error function. The derivative is approximated by a finite difference expression,

$$\frac{\partial\varepsilon}{\partial p_i} \approx \frac{\varepsilon([p_1 \ p_2 \ \dots \ p_i + \Delta \ \dots]^T) - \varepsilon([p_1 \ p_2 \ \dots \ p_i - \Delta \ \dots]^T)}{2\Delta} \quad (4)$$

In order to use (4) as a gradient estimate, ε is estimated by setting $\mathbf{p} = [p_1 \ p_2 \ \dots \ p_i + \Delta \ \dots]^T$ and averaging $e^2(k)$ over $L/2$ data samples, then setting $\mathbf{p} = [p_1 \ p_2 \ \dots \ p_i - \Delta \ \dots]^T$ and averaging $e^2(k)$ over the following $L/2$ data samples.

$$\hat{\partial\varepsilon} = \frac{1}{2\Delta} \left[\frac{2}{L} \left(\sum_{k=l}^{l+L/2} e^2(k) \right) - \frac{2}{L} \left(\sum_{k=l+L/2}^{l+L} e^2(k) \right) \right] \quad (5)$$

Using the gradient estimate in (5) together with the parameter update rule in (2) results in the differential steepest descent (DSD) algorithm. An advantage of the DSD algorithm over LMS adaptation is that it does not require access to the filter's internal states. It is simple and intuitive. However, its hardware implementation is somewhat complicated by the fact that the gradient components must be estimated one at a time. During their discussion of the DSD algorithm, the authors of [4] point out that, "All weights can be simultaneously dithered at individual frequencies and the gradient components obtained by cross correlation." That idea is the basis of the "dithered linear search" algorithm described below.

3. THE DITHERED LINEAR SEARCH

The term "dither" refers to a signal with small amplitude and zero mean that is intentionally injected into a system. The dithered linear search (DLS) algorithm, like all linear search algo-

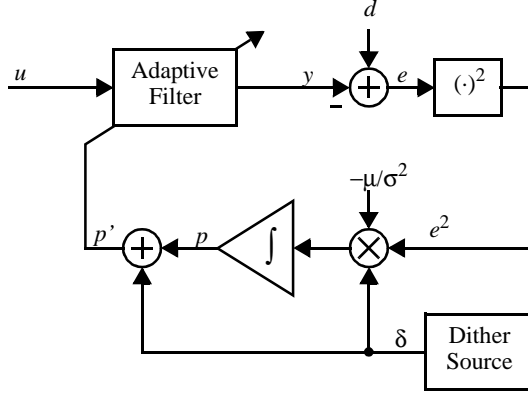


Figure 1. Block diagram of the DLS algorithm. Notice that no knowledge of the filter's internal states is required.

rithms, is a gradient descent optimizer. The DLS proceeds by simultaneously applying independent additive dither, $\delta_i(k)$, to all of the filter parameters being adapted.

$$p_i'(k) = p_i(k) + \delta_i(k) \quad (6)$$

In (6), p is the algorithm's current estimate of the optimal parameter values whereas p' is the parameter vector including dither. Each gradient component is then inferred by correlating changes in the corresponding dither signal to changes in $\epsilon(p')$.

$$p_i(k+1) = p_i(k) - \frac{\mu}{\sigma^2} \cdot \delta_i(k) \cdot \epsilon(p_i') \quad (7)$$

In (8), σ^2 is the variance of the dither signal. Again, instantaneous estimates can be substituted for $\epsilon(p_i)$ in which case (7) becomes,

$$p_i(k+1) = p_i(k) - \frac{\mu}{\sigma^2} \cdot \delta_i(k) \cdot e^2(k) \quad (8)$$

The gradient estimate is given by,

$$\frac{\partial \epsilon}{\partial p_i} = \frac{1}{\sigma^2} \cdot \delta_i(k) \cdot e^2(k) \quad (9)$$

Multiple filter parameters can be adapted simultaneously using independent (i.e. uncorrelated) dither signals. The adaptation of any one parameter, p_i , will not be effected by the dither on the others since, over time, only those changes in ϵ which are correlated with δ_i will influence the parameter p_i .

The DLS algorithm has a straightforward hardware implementation. A block diagram is shown in Fig. 1 for one parameter. The algorithm is easily scaled to adapt more parameters using more copies of the same hardware. If the dither signals are binary, the hardware is even simpler. The correlator becomes trivial and the dither signal can be directly connected to the LSB of the parameter control word. Uncorrelated binary dither signals are also very easy to generate in hardware.

4. THEORETICAL ANALYSIS

In this section, it will be shown that the gradient estimates in (9) are unbiased and, hence, the DLS algorithm is guaranteed to converge as long as the rate of adaptation, μ , is taken small enough. In order to simplify the theoretical analysis only adaptive linear combiners and binary dither signals are analyzed in detail.

An adaptive linear combiner with N parameters and state signals x_i , $1 \leq i \leq N$ has a state correlation matrix \mathbf{R} ,

$$\mathbf{R} = E \begin{bmatrix} x_1 x_1 & x_1 x_2 & \dots \\ x_2 x_1 & x_2 x_2 & \dots \\ \vdots & \vdots & \ddots \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & \dots \\ r_{21} & r_{22} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix} \quad (10)$$

Mathematically, the requirement of unbiased gradient estimates can be written as follows:

$$E[\nabla_{\hat{p}} \epsilon] = \nabla_p \epsilon \quad (11)$$

$$\Leftrightarrow E\left[\frac{\partial \epsilon}{\partial p_i}\right] = \frac{\partial \epsilon}{\partial p_i}, \text{ for all } i \quad (12)$$

For a binary dither signal, it is known that,

$$\delta_i(k) = \pm \Delta \quad (13)$$

Hence,

$$\sigma^2 \equiv E[\delta_i^2(k)] = \Delta^2 \quad (14)$$

Taking the expectation of both sides of (9),

$$\begin{aligned} E\left[\frac{\partial \epsilon}{\partial p_i}\right] &= E\left[\frac{1}{\sigma^2} \cdot \delta_i(k) \cdot e^2(k)\right] \\ &= \frac{1}{\Delta^2} \cdot E[\delta_i(k) \cdot e^2(k)] \end{aligned} \quad (15)$$

Since a dither signal has zero mean,

$$E[\delta_i(k)] = 0 \quad (16)$$

$$\Rightarrow \begin{cases} \delta_i(k) = +\Delta, & \text{one-half of the time} \\ \delta_i(k) = -\Delta, & \text{one-half of the time} \end{cases} \quad (17)$$

Combining (17) with (15) yields,

$$\begin{aligned} E\left[\frac{\partial \epsilon}{\partial p_i}\right] &= \frac{1}{\Delta^2} \left(\left(\frac{1}{2} \cdot E[\delta_i \cdot e^2] \Big|_{\delta_i = +\Delta} \right) + \left(\frac{1}{2} \cdot E[\delta_i \cdot e^2] \Big|_{\delta_i = -\Delta} \right) \right) \\ &= \frac{1}{\Delta^2} \left(\left(\frac{\Delta}{2} E[e^2] \Big|_{\delta_i = +\Delta} \right) + \left(\frac{-\Delta}{2} E[e^2] \Big|_{\delta_i = -\Delta} \right) \right) \\ &= \frac{1}{2\Delta} \left(E[e^2] \Big|_{\delta_i = +\Delta} - E[e^2] \Big|_{\delta_i = -\Delta} \right) \end{aligned} \quad (18)$$

In the two-parameter case ($N = 2$), when $\delta_1 = -\Delta$ the filter operates one-half of the time at each of the two points $\mathbf{p}' = [p_1 - \Delta \ p_2 \pm \Delta]^T$. Therefore,

$$\begin{aligned} E[e^2] \Big|_{\delta_1 = -\Delta} &= \frac{1}{2} \cdot \left(E[e^2] \Big|_{\mathbf{p}' = [p_1 - \Delta \ p_2 + \Delta]} + E[e^2] \Big|_{\mathbf{p}' = [p_1 - \Delta \ p_2 - \Delta]} \right) \\ &= \frac{1}{2} \left(\epsilon([p_1 - \Delta \ p_2 + \Delta]) + \epsilon([p_1 - \Delta \ p_2 - \Delta]) \right) \end{aligned} \quad (19)$$

Substituting the quadratic performance surface of a general adaptive linear combiner into (19) results in,

$$E[e^2] \Big|_{\delta_1 = -\Delta} = \epsilon(p_1 - \Delta, p_2) + \Delta^2 r_{22} \quad (20)$$

This result is easily generalized to the i th gradient estimate in an N -parameter case,

$$E[e^2] \Big|_{\delta_i = \pm \Delta} = \epsilon([p_1 \ p_2 \ \dots \ p_i \pm \Delta \ \dots]^T) + \Delta^2 (\text{tr}(\mathbf{R}) - r_{ii}) \quad (21)$$

Substituting (21) into (18) gives,

$$\begin{aligned}
 E\left[\frac{\partial \hat{\epsilon}}{\partial p_i}\right] &= \frac{1}{2\Delta} \left(\left(\epsilon\left([p_1 \dots p_{i+\Delta} \dots]^T\right) + \Delta^2(\text{tr}(\mathbf{R}) - r_{ii}) \right) \right. \\
 &\quad \left. - \left(\epsilon\left([p_1 \dots p_{i-\Delta} \dots]^T\right) + \Delta^2(\text{tr}(\mathbf{R}) - r_{ii}) \right) \right) \\
 &= \frac{\epsilon\left([p_1 \dots p_{i+\Delta} \dots]^T\right) - \epsilon\left([p_1 \dots p_{i-\Delta} \dots]^T\right)}{2\Delta} \quad (22)
 \end{aligned}$$

In order to complete the proof of (12), it is required that,

$$\frac{\epsilon\left([p_1 \dots p_{i+\Delta} \dots]^T\right) - \epsilon\left([p_1 \dots p_{i-\Delta} \dots]^T\right)}{2\Delta} = \frac{\partial \epsilon}{\partial p_i} \quad (23)$$

Equation (23) is true by definition of the gradient in the limit $\Delta \rightarrow 0$, and is also true for any value of Δ when the error function ϵ is a quadratic function of \mathbf{p} , as in adaptive linear combiners. Therefore, the DLS algorithm provides unbiased gradient estimates under those circumstances.

Since the gradient estimates are unbiased, when averaged over a long time they approach the true value of the gradient resulting in a perfect gradient descent optimizer. This limit is approached by taking μ very small. So, the DLS is guaranteed stable as long as μ is taken “small enough”. In general, the range of stable values for μ will depend upon the filter structure, the type of dither signal, and the statistics of the input data (which may not be precisely known *a priori*). Therefore, stability of the adaptive process should always be verified via simulation.

Note that (22) is a finite difference gradient estimate, just like the one used for the DSD algorithm in (4). The similarity is not surprising since the DSD algorithm can be considered a special case of the DLS algorithm where the parameters are dithered one at a time.

5. BEHAVIORAL SIMULATIONS

Behavioral simulations were performed using a 3rd order continuous-time orthonormal ladder filter in a “system identification” or “model matching” application. The structure of the adapted filter is shown in Fig. 2 [5]. With the feedback parameters α_i fixed and the feedin terms β_i adapted, the filter is an adaptive linear combiner. However, the state signals are not available at any internal nodes in the structure of Fig. 2, so the gradient signals required for LMS adaptation would be difficult to obtain. Specifically, it would be necessary to operate a second 3rd order continuous-time filter in parallel just to generate the gradient signals [3]. Therefore, the DLS algorithm is particularly desirable for this structure.

A finite steady-state error was introduced by including an additive independent noise source in the simulations at the filter output with a power of $\epsilon_{\min} = 0.01$. The input signal was white with a power of 0.1. The reference filter is a 3rd order elliptic lowpass transfer function with 0.5 dB ripple in the passband extending to a frequency of 10 (normalized with respect to the sampling frequency) and 40 dB of stopband attenuation. In the adapted filter, the feedback parameters, α_i , were chosen to provide the same pole placement as the reference filter and the feedin parameters, β_i , were adapted. The values of δ and μ are design parameters. In these simulations, they were chosen to yield a total

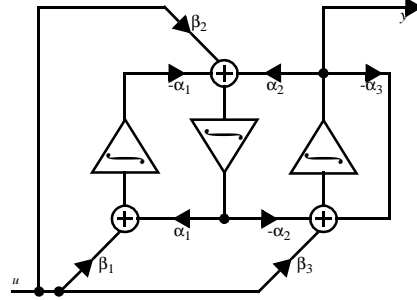


Figure 2. A 3rd-order orthonormal ladder filter using multiple feedins of the input signal.

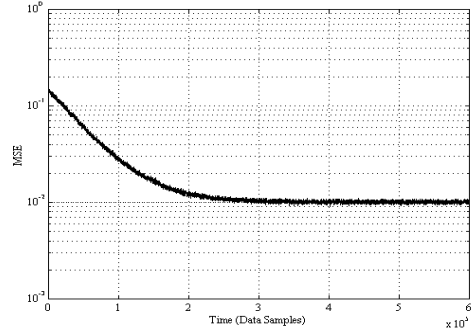


Figure 3. Behavioral simulation results of the DLS algorithm over an ensemble of 25 simulation runs.

excess MSE 1% of ϵ_{\min} in steady-state. The dither signals were Hadamard sequences of length 4 [6]. Results from an ensemble of 25 simulation runs are plotted in Fig. 3. All 25 simulation runs converged.

6. DC OFFSET EFFECTS

It is well known that dc offsets can limit the performance of the LMS algorithm for analog filters [1], [2], [3]. Offsets on the state and error signals cause excess MSE in steady-state. Therefore, dc offsets represent a significant performance limitation for analog adaptive filters and much research has been done to reduce the effect of dc offsets on LMS adaptation. Offsets on the error signal, $e(k)$ are usually eliminated using an adaptive dc tap at the filter output. Furthermore, since the DLS and DSD algorithms estimate the gradient from observations of only the squared error, $e^2(k)$, they are not susceptible to dc offsets on the state signals.

In order to highlight the effect of dc offsets, behavioral simulations were performed for the same model matching system described in Section 5, this time with dc offsets introduced on each of the filter’s internal states and on the error signal. The dc offsets have a mean squared value 1/10th that of the state and error signals respectively. No steady-state error is introduced at n (i.e. $\epsilon_{\min} = 0$). Using LMS adaptation (Fig. 4A) a residual error of approximately -15 dB relative to the reference signal persists due to the dc offsets. Using the same dc offsets and the DLS algorithm with an adaptive dc tap at the filter output, the only residual steady-state error is due to the perturbation caused by the dither itself, in this case approximately -45 dB relative to the reference signal (Fig. 4B).

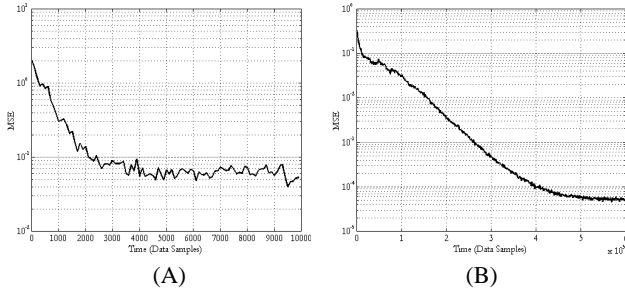


Figure 4. Mean squared error simulated with dc offsets on the state and error signals using (A) the LMS algorithm; (B) the DLS algorithm with an adaptive dc tap.

7. EXPERIMENTAL RESULTS

Model matching experiments were performed using the DLS algorithm and a continuous-time analog integrated filter. A die photo of the test chip is shown in Fig. 5 [7]. Only the digitally programmable analog signal path was integrated. The adaptation algorithm was implemented in software to provide greater flexibility. The filter is a 5th order orthonormal ladder filter with variable feedins, so the state signals are completely unavailable. The shape of the frequency response was fixed, but the cutoff frequency was programmable up to around 70 MHz by scaling the feedback gains. The dc gain was also programmable by scaling the feedin parameters. These two scaling factors were used as the adapted parameters.

In the model matching experiment, the same integrated filter was used alternately as both the adapted filter and the reference filter in order to avoid mismatches between the two paths in the experimental setup. Again, Hadamard sequences were used for the dither. The filter parameters and the relative MSE are plotted over time in Fig. 6. The non-zero MSE observed in steady-state is due to measurement noise.

8. CONCLUSION

A generalization of the differential steepest descent (DSD) algorithm, here called the dithered linear search (DLS), was described and analyzed. Its performance is the same as the DSD algorithm, however its hardware implementation can be somewhat simpler since only the parameters' LSB must be dithered during adaptation. When compared with the popular LMS algorithm, the DLS algorithm is much slower because it does not make use of the filter's internal state signals. However, the LMS algorithm has a number of serious drawbacks, particularly for analog continuous-time integrated filters. If implemented digitally, the state signals required for LMS adaptation are difficult to obtain or completely unavailable. If implemented with analog circuits, the LMS algorithm is susceptible to dc offsets. The DLS algorithm does not require access to any internal state signals and, by introducing an adaptive dc tap, the DLS algorithm is robust with respect to dc offsets. Furthermore, the rate of convergence is not a limiting factor in many digital communications applications where analog adaptive filters are often used. The adaptive algorithm was tested using both simulations and in hardware. In the hardware tests, the DLS algorithm successfully adapted a continuous-time 5th order analog filter.

9. REFERENCES

- [1] C.-P.J. Tzeng, "An Adaptive Offset Cancellation Technique for Adaptive Filters," *IEEE Trans. on Acoust., Speech, Signal Proc.*, pp. 799-803, May 1990.
- [2] D.A. Johns, W.M. Snelgrove, and A.S. Sedra, "Continuous-Time LMS Adaptive Recursive Filters," *IEEE Trans. Circuits Syst.*, pp. 769-778, July 1991.
- [3] K.A. Kozma, D.A. Johns, A.S. Sedra, "Automatic Tuning of Continuous-Time Integrated Filters Using an Adaptive Filter Technique," *IEEE Trans. Circuits Syst.*, pp. 1241-8, Nov. 1991.
- [4] B. Widrow, J.M. McCool, "A Comparison of Adaptive Algorithms Based on the Methods of Steepest Descent and Random Search," *IEEE Trans. Ant. & Prop.*, pp. 615-637, Sept. 1976.
- [5] D.A. Johns, W.M. Snelgrove, A.S. Sedra, "Orthonormal Ladder Filters," *IEEE Trans. Circuits Syst.*, pp. 337-343, March 1989.
- [6] J.H. van Lint, R.M. Wilson, *A Course in Combinatorics*, New York, NY: Cambridge University Press, 1992.
- [7] A. Chan Carusone and D. A. Johns, "A 5th Order Gm-C Filter in 0.25 μm CMOS with Digitally Programmable Poles & Zeros," *2002 IEEE Int. Symp. Circuits and Syst.*, May 2002.

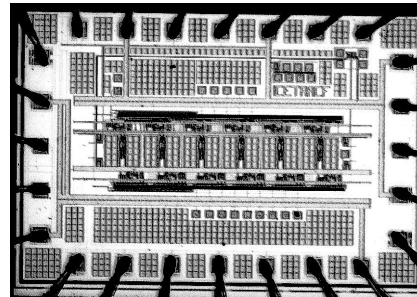


Figure 5. Die photo of the 5th order orthonormal ladder analog filter test chip.

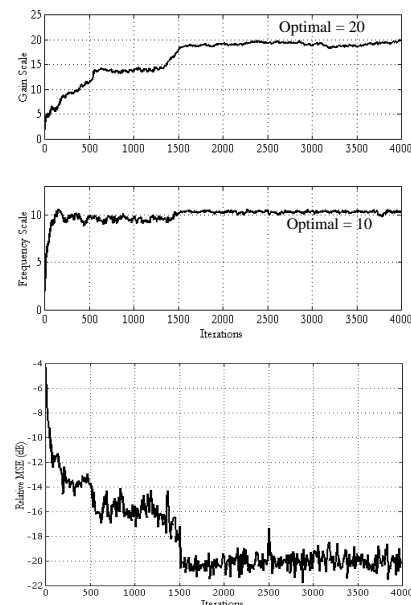


Figure 6. Experimental results from the adaptation of a 5th order analog integrated filter using the DLS algorithm.