DC OFFSETS IN ANALOGUE ADAPTIVE IIR FILTERS

D.A. Johns, W.M. Snelgrove and A.S. Sedra

University of Toronto, Canada

INTRODUCTION

Recently, experimental results for a discrete prototype analogue adaptive IIR filter were presented in Johns et al. (1). The discrete prototype consisted of a state-space filter where poles and zeros were adapted by adjusting state coefficients through the use of a gradient based LMS algorithm as described in Johns et al. (2). Although this analogue algorithm was previously presented for adapting the zeros of adaptive linear combiners in Widrow et al. (3), the results in (1) showed that a similar approach could successfully be applied to adapting the poles as well as the zeros of analogue adaptive IIR filters. The necessary gradient signals were obtained as the outputs of an extra filter allowing the adaptive algorithm to be implemented entirely with analogue components. However, as with any analogue system, DC offsets occur throughout the realization and thus their effects need to be addressed. Through experimentation with the prototype described in (1), it was determined that some of the more critical locations for DC offsets to occur are in the coefficient update algorithm integrators. Towards understanding the effects of these integrator offsets, this paper presents approximate formulae relating coefficient error and excess mean squared error to DC offsets and a gradient correlation matrix.

The effect of DC offsets present in analogue adaptive linear combiners (where only zeros are adapted) has been previously investigated in Compton (4) and a formula derived giving the coefficient error due to DC offsets in coefficient update integrators. One result in this paper is to show that this same DC offset formula can be used to give approximate results for adaptive IIR filters where both poles and zeros are adapted. In showing the validity of this approximate DC offset formula result, a different derivation is used in this paper so that an appropriate approximation can be made. Simulation results are then given verifying the approximation made. Another result in this paper is to modify the offset formulae to account for use of the LMS sign-data algorithm. Although, this sign-data algorithm is often used in practice due to its very low complexity, these authors have seen no analysis of DC offset effects on systems using this algorithm in the literature. Finally, experimental results using a discrete prototype are presented confirming the usefulness of the sign-data offset-induced excess error formulae.

DERIVATION OF THE DC OFFSET FORMULAE

Consider a general adaptive IIR filter shown in Figure 1. The output of the programmable filter, y(t), is subtracted from a reference signal, S(t), to create an error signal, e(t). The adaptive algorithm makes use of the error signal and filter states to adjust the programmable filter coefficients, [n], in such a way as to minimize the mean squared value of the error signal. In other words, the purpose of the adaptive algorithm is to locate a minimum in an error performance surface by adjusting the filter coefficients. The error performance surface is defined to be the surface which is created by measuring the mean squared value of the error signal as the filter coefficients are varied.

One approach to finding a minimum in an error performance surface is to use the method of steepest descent. Unfortunately, with the standard steepest descent method, one requires a partial derivative of the mean squared error with respect to the filter coefficients. To circumvent this problem, the least-mean-squared (LMS) algorithm was developed, Widrow and Hoff (5), where the instantaneous error squared signal is used to approximate the mean squared error. With this approach, the following LMS update formula for the coefficient p[i] is obtained

\[ p[i] = \mu \int \left[ e(t) \frac{\partial y(t)}{\partial p[i]} \right] dt \]

where \( \mu \) is a small positive parameter which controls the rate of convergence. All the derivative terms, \( \frac{\partial y(t)}{\partial p[i]} \), can be obtained using a gradient filter as described in (1).

Unfortunately, the above update formula can not be realized exactly and DC offset terms will always be present in analogue implementations. With a DC offset term present, the i'th coefficient update formula becomes

\[ p[i] = \mu \int \left[ e(t) \frac{\partial y(t)}{\partial p[i]} + m[i] \right] dt \]

where m[i] is the DC offset for the i'th update formula. A block diagram for this coefficient update formula is shown in Figure 2 where the DC offset is injected as a separate signal so that the integrator may be considered ideal. Note that with such a conceptualization, the DC offset term need not come from only the offset of the integrator. Specifically, if both the error and gradient signals have offsets, then these DC signals correlate with each other resulting in an additional DC offset. By defining m[i] to also include this offset signal, we may consider both the error and gradient signals to be ideal with respect to DC offsets.

When an adaptive filter is at steady state, the expected value of the coefficient signal p[i] is a constant value implying that the expected value of the signal into the integrator must be zero. Thus at steady state, the following equation holds

\[ E[e(t) \frac{\partial y(t)}{\partial p[i]} + m[i]] = 0 \]

where \( E[\cdot] \) denotes expectation. Since the expected value of a DC signal is the DC level, we can write

\[ E[e(t) \frac{\partial y(t)}{\partial p[i]}] = -m[i] \]

(1)

We now make use of the fact that with the LMS algorithm, the inside of the expectation operator in equation (1) is the
instantaneous estimate of the derivative of the mean squared error with respect to the parameter $p_i$, or in mathematical terms,

$$\epsilon(t) \frac{\partial y(t)}{\partial p_i} = -\frac{1}{2} \frac{\partial \epsilon^2(t)}{\partial p_i}$$  \hspace{1cm} (2)

Substituting equation (2) in equation (1) above and swapping the expectation and derivative operators, we also have the following condition at steady state.

$$\frac{\partial E[\epsilon^2(t)]}{\partial p_i} = 2m_i$$

This formula implies that when no DC offset is present in the $i$th update formula ($m_i = 0$), the adaptive filter settles at a point where the partial derivative of the performance surface with respect to the $i$th coefficient is zero. This is precisely the condition for finding a minimum. However, in the case of a non-zero DC offset, the adaptive filter settles at a point where the same partial derivative is at a value equal to twice the DC offset. In other words, the filter settles at a position where the error is slightly correlated with the gradient signal in order to cancel the effect of the DC offset, as seen from equation (1). Note that a DC offset forces the filter coefficients to be incorrect which implies an error in the programmable filter’s transfer function at all frequencies (not just at DC).

For the adaptive IIR filter shown in Figure 1, consider the case where there are $N$ filter coefficients, $(p_i)$. Now making the assumption that only small coefficient changes occur due to DC offsets, at steady state the error signal can be written as

$$\epsilon(t) = \delta(t) - y^*(t) = \sum_{i=1}^{N} \frac{\partial y(t)}{\partial p_i} \Delta p_i$$  \hspace{1cm} (3)

where $y^*(t)$ is defined as the optimum output which causes the minimum mean squared error and $\Delta p_i$ is defined to be the change in coefficients from their optimum values due to DC offsets. Making the use of vector notation, we can write the gradient signals as a vector, $\Delta p$, where

$$\Delta p_i = \frac{\partial y(t)}{\partial p_i}$$  \hspace{1cm} for $i = 1$ to $N$

and the change in coefficients as the vector $q$ where

$$q_i = \Delta p_i$$

With this notation, we can write equation (3) as

$$\epsilon(t) = \delta(t) - y^*(t) = \nabla^2(t) q$$

Since we are interested in finding the excess mean squared error and coefficient errors that result from DC offsets (as opposed to overall mean squared error), without loss of generality, we make the assumption that the mean squared error equals zero if all the DC offsets are zero. Making this assumption implies that the optimum filter output, $y^*(t)$, equals the reference signal, $\delta(t)$ and therefore the excess error signal can be reduced to simply

$$\epsilon(t) = -\nabla^2(t) q$$  \hspace{1cm} (4)

Now writing the DC offsets as a vector $m$, we can apply equation (1) above for the set of DC offsets to obtain

$$E[\epsilon(t)] = -m$$  \hspace{1cm} (5)

Combining equations (4) and (5) results in

$$E[\epsilon(t)] a^2(t) q = m$$  \hspace{1cm} (6)

Now defining a correlation matrix, $R$, as

$$R = E[\epsilon(t) \nabla^2(t)]$$

equation (6) can be solved for $q$ resulting in

$$q = R^{-1} m$$  \hspace{1cm} (7)

Thus, with the use of the gradient correlation matrix, $R$, this equation gives the error in coefficient values due to DC offsets. Note that this is the same equation derived for the adaptive linear combiner case in reference (4) except that this formula is exact in the linear combiner case. However in the IIR case, formula (7) is only an approximation due to the assumption made in writing the error signal equation (3).

To obtain the excess mean squared error, $|q|^2$, due to DC offsets, we use the definition for the mean squared error and perform the following manipulations,

$$|q|^2 = E[\epsilon(t) \epsilon(t)] = E[q^T \nabla^2(t) q] = E[m^T R^{-1} R q^T m] = m^T R^{-1} m$$  \hspace{1cm} (8)

Note from equation (8) that the value of the excess error due to DC offsets is proportional to the inverse of the correlation matrix, $R$. This fact implies that the excess error will increase as the states become more correlated since the matrix $R$ will become more ill-conditioned. This increased excess error is one reason to look for adaptive IIR structures with orthonormal gradients for analogue realizations. In the case where all the gradient signals are orthonormal (ie $R$ equals the identity matrix), the excess mean squared error is simply the sum of the squares of the DC offsets. Also note that in the adaptive linear combiner case, the correlation matrix is not a function of the coefficients, $p_i$, and thus the offset-induced excess error does not depend on the final transfer function of the adaptive filter. However, in the adaptive IIR case, the correlation matrix is a function of the adaptive filter’s transfer function and thus one requires a knowledge of the final transfer function to apply the offset-induced excess error formula. Since this exact transfer function is usually not known, one can only hope to obtain approximate results with this method by estimating the final transfer function. Finally note that the level of the input signal affects the excess error through the correlation matrix $R$.

**Simulation Examples**

As a test for this error formula, some simulations of adaptive digital filters were performed. Digital filters were used in the simulations since they are much easier to simulate than analogue equivalents. In the first example, a third order adaptive linear combiner with DC offsets was simulated where the reference transfer function was $z^{-2} + z^{-1} + 1$. The input correlation matrix $R$ for the simulation was chosen to be

$$R = \begin{bmatrix} 1 & 0.8 & 0.9 \\ 0.8 & 1 & 0.72 \\ 0.9 & 0.72 & 1 \end{bmatrix}$$

and the DC offset vector $m$ was arbitrarily set to

$$m = \begin{bmatrix} 0.01 \\ 0.02 \\ -0.03 \end{bmatrix}$$
For this example, the calculated excess root mean squared error (RMSE) using equation (8) was 0.09222 and the simulated RMSE was 0.092. This is certainly a close agreement as one expects for the adaptive linear combiner case.

To check the validity of this formula for the IIR case, a second order example with DC offsets present was simulated. In this simulation, a state-space system was used where only the bottom row of the A matrix was adapted while all other parameters remained equal to the optimum values. The reference filter had the state-space coefficients:

\[ A = \begin{bmatrix} 0 & 1 \\ -0.6 & 1.2 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0.41 \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad d = 0 \]

Defining \( x(n) \) to be the vector of gradient signals required to adapt the bottom row of \( A \), the correlation matrix, \( R \), was found to be

\[ R = \begin{bmatrix} 0.998 & 0.816 \\ 0.816 & 0.998 \end{bmatrix} \]

Note that this correlation matrix shows high correlation between gradients and therefore a large curvature in the performance surface. With offsets arbitrarily set equal to 0.01 and 0.01 for \( m_1 \) and \( m_2 \), respectively, the simulated and calculated RMS errors were 0.036 and 0.0332, respectively. The bottom row of the adaptive filter's A matrix settled at the coefficient values -0.575 and 1.175. This example shows a reasonable agreement between the calculated and simulated values. However, by decreasing the offset, the accuracy of the small change approximation in equation (8) is improved and therefore an even closer agreement should be obtained. Decreasing the offsets to 0.001 and -0.001, the simulated and calculated RMS errors were 0.0033 and 0.00332, respectively, which is certainly a close agreement. For this simulation, the bottom row of the A matrix settled at the coefficient values -0.598 and 1.196.

**EXCESS ERROR FOR THE LMS SIGN-DATA CASE**

In many practical realizations, the LMS algorithm is modified to the LMS sign-data algorithm to simplify the multiply operation between the error and the gradient signals. Treichler et al. (6) However, the effects of DC offsets on systems using this simplified algorithm has not yet been analyzed in the literature. In this section, the offset-induced excess error formula will be modified to account for use of the LMS sign-data algorithm.

In accounting for the use of the sign-data algorithm, we make use of the sign function, \( \text{sgn}(x) \), defined as follows:

\[ \text{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \end{cases} \]

In the sign-data algorithm, the following update equation is used for the coefficient \( p_i \):

\[ p_i(t) = 2\mu \int e(t) \text{sgn}(\frac{d y(t)}{dB}) dt \]

Note that this above equation is essentially the same as in the LMS case but with the introduction of the signum function on the gradient signal. It is not difficult to see that when DC offsets are applied to a system using the sign-data algorithm, equation (5) above becomes

\[ E[\text{sgn}(x(t))e(t)] = -m \]

where the signum of a vector is defined as applying the signum function to each of the vector elements. As before, we can write the error signal as a function of \( q \) and \( x(t) \) using equation (4) and therefore can write

\[ E[\text{sgn}(x(t))e(t)]q = m \]

This leads to the following equation for \( q \):

\[ q = \tilde{R}^{-1}m \]

where the signum correlation matrix, \( \tilde{R} \), is defined as

\[ \tilde{R} = E[\text{sgn}(x(t))x(t)^T] \]

or, equivalently, the element \( \tilde{R}_{ij} \) is defined as

\[ \tilde{R}_{ij} = E[\text{sgn}(x_i(t))x_j(t)] \]

Now, as before, we can use the definition for the mean squared error to obtain the following offset-induced excess error formula for the sign-data case.

\[ \mathbf{e}^T = m^T \tilde{R} - \tilde{R}^{-1}m \]

Note that in the above equation, the matrices \( R \) and \( R \) need to be obtained. For white noise inputs, the matrix \( R \) can be easily obtained using impulse responses. Unfortunately, it is not clear how to obtain the basic matrix \( R \) except by applying the defining equation (9). However, in the special case where the inputs have a Gaussian white noise zero mean characteristic, one can find a closed form expression for the elements of \( R \) in terms of the elements of \( \tilde{R} \). Specifically, if the input signal has a zero mean Gaussian distribution, the joint probability density function, \( \Phi_{x_1x_2}(x_1,x_2) \), between the signals \( x_1 \) and \( x_2 \) can be written as (Papoulis (7), p. 186).

\[ \Phi_{x_1x_2}(x_1,x_2) = \frac{1}{2\pi \sigma_1 \sigma_2 \rho_{x_1x_2}} \exp \left[-\frac{1}{2} \left( \frac{x_1^2}{\sigma_1^2} + \frac{x_2^2}{\sigma_2^2} - 2 \rho_{x_1x_2} \frac{x_1x_2}{\sigma_1 \sigma_2} \right) \right] \]

Using this joint probability function, the term \( \tilde{R}_{ij} \) can be found by integrating the weighted probability density function over both variables or mathematically,

\[ \tilde{R}_{ij} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \text{sgn}(x_1) \Phi_{x_1x_2}(x_1,x_2) dx_1 dx_2 \]

Performing this integration leads to the following closed form expression for the elements of \( R \).

\[ \tilde{R}_{ij} = \int \int \text{sgn}(x_1) \Phi_{x_1x_2}(x_1,x_2) dx_1 dx_2 \]

Therefore, in the case of zero mean Gaussian white noise inputs, equation (11) can be used to obtain \( R \) and equation (10) can be used to give the offset-induced excess error in adaptive FIR filters using the sign-data algorithm. The same formula can be used to give approximate results for adaptive IIR filters also using the sign-data algorithm.

**Experimental Results**

In this section, DC offset experimental results using a discrete prototype will be compared to theoretical values found by applying the offset-induced excess error formulae above. Since the discrete prototype uses the sign-data algorithm, equation (10) will be the formula used for comparison. The discrete prototype is a third-order adaptive filter where \( 6 \) filter coefficients are adjusted to adapt both the poles and zeros of the filter. A model matching application was used for comparison and a white noise generator was applied to both the reference and adaptive filter. The details of the discrete prototype are given in (1).
A DC offset signal was applied separately to each of the coefficient update integrators as shown in Figure 2. However, it should be mentioned here that the sign of the gradient signal was multiplied by an amplified version of the error signal rather than the error signal itself. Letting $k$ represent the amplification constant for the error signal, it is easily seen that introducing this gain factor reduces the offset effects since $k|e|^2$ replaces $|e|^2$ in equation (10) above. Since, the right hand side of equation (10) is unaffected by the addition of the gain factor, $k$, the resulting offset-induced excess rms error is reduced by the factor $k$. Experimentation confirms the reduction in offset-induced excess error when increasing the gain factor, $k$. It should be pointed out that this gain factor will be difficult to realize at high frequencies since high frequency gain circuits are not a trivial task to implement. This difficulty in implementation is one of the major reasons for developing these DC offset related formulae. With these formulae available and a known tolerance on DC offsets, a designer can choose the minimum error gain factor, $k$, necessary to meet specifications. For experimentation with the discrete prototype, a gain of 82 was used for $k$.

Referring again to Figure 2, positive and negative offset voltages of equal magnitude were applied to the $i$'th offset integrator. With this approach, two offset-induced excess rms error voltages are measured, $|e|^2$ and $|e|^2$, corresponding to the positive and negative offset voltages, respectively. If no other offset voltages are present in the circuit, then the above theory predicts that $|e|^2$ should equal $|e|^2$. However, this was not observed due to the fact that there already existed unknown equivalent offsets resulting from the non-idealities of the circuit realization. If we call the vector of unknown offsets $m$ and the measured rms error with no external offsets applied, $|e|^2$, then one can write

$$|e|^2 = m^T H m,$$

where $H$ is defined to be $R^{-1}$. Now letting the known vector of positive applied offsets be $m_0$, we can write equations for $|e|^2$ and $|e|^2$:

$$|e|^2 = (m_0 + m)^T H (m_1 + m_2)$$

and

$$|e|^2 = (m_1 - m) H (m_1 - m_2).$$

Letting $|e|^2$ be the offset-induced excess error due to $m_0$ only, from the above equations, it is not difficult to show the following equality.

$$|e|^2 = 2 |e|^2 + 2 |e|^2 - |e|^2.$$

Finally, we can now compare experimental results with theoretical ones. Table 1 shows a comparison of theoretical vs. experimental excess error voltages due to DC offsets. Each row in Table 1 corresponds to a DC offset voltage applied to a single coefficient update integrator. To measure the rms voltages of the amplified error, $|e|^2$, a digital-readout true rms meter was used. Unfortunately, it was not a simple matter to read the rms value of the output error signal since low frequency components were present and thus successive meter readings varied considerably. The value used was an estimate of the average of a few successive readings. We see from Table 1 that all measurements agree within 20 percent of the theoretical predictions. This degree of accuracy is reasonable considering that all circuit non-idealities other than integrator DC offsets have been ignored and that noise rms measurements were made. This degree of accuracy should be close enough for design purposes when one considers the variability of DC offsets in a given technology.

**CONCLUSIONS**

This paper presented approximate formulae relating coefficient errors and excess mean squared error to DC offsets in analogue adaptive IIR filters. The formulae were derived for both the LMS and LMS sign-data algorithms. Simulation and experimental results were presented verifying the derived formulae.

**REFERENCES**


TABLE 1 - DC offset experimental vs. theoretical results. When no DC offsets applied, $\mu e L_m$ equals 0.2 Vrms. The i'th row corresponds to DC offsets on the i'th update integrator.

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>$\mu e L_p$</th>
<th>$\mu e L_h$</th>
<th>Experimental $\mu e L_m$</th>
<th>Theoretical $\mu e L_m$</th>
<th>Percentage Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.1</td>
<td>0.7</td>
<td>0.9</td>
<td>0.77</td>
<td>17</td>
</tr>
<tr>
<td>2</td>
<td>0.31</td>
<td>0.36</td>
<td>0.27</td>
<td>0.281</td>
<td>-4</td>
</tr>
<tr>
<td>3</td>
<td>0.33</td>
<td>0.26</td>
<td>0.22</td>
<td>0.22</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0.4</td>
<td>0.34</td>
<td>0.31</td>
<td>0.275</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>0.7</td>
<td>0.51</td>
<td>0.58</td>
<td>0.575</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0.7</td>
<td>0.46</td>
<td>0.56</td>
<td>0.538</td>
<td>4</td>
</tr>
</tbody>
</table>