THIS APPLICATION CAN BE DONE USING:

PRODUCTS
Block Diagram
RIDE
ImPro Lab
VIDSP Studio
VIDSP Suite
OORVL Design Studio

Adaptive Filter Design - Method 1

Overview

An adaptive filter is a filter containing coefficients that are updated by some type of adaptive algorithm to improve or somehow optimize the filter's response to a desired performance criterion. In general, adaptive filters consist of two basic parts: the filter which applies the required processing on the incoming signal which is to be filtered; and an adaptive algorithm, which adjusts the coefficients of that filter to somehow improve its performance. This application note makes use of a self-contained LMS Adaptive Filter component, or block function, to achieve both the filtering and the adaptive algorithm for coefficient update; note that Application Note HSAP2023 achieves the adaptive filter without using the LMS Adaptive Filter block, and instead uses very low-level components such that the inner structure of the adaptive filter is more readily seen.

The overall structure for an adaptive filter is shown below in Figure 1. The incoming signal, x(n), is filtered (or weighted) in a digital filter to produce an output, y(n). The adaptive algorithm will continuously adjust the coefficients, or tap weights, in the filter to minimize the error, e(n), between the filtered output, y(n), and a signal representing the desired response of the filter, d(n).

Note: Selection of the desired response, d(n), of the filter is sometimes the most difficult step and can dramatically affect the success of the adaptive filter.

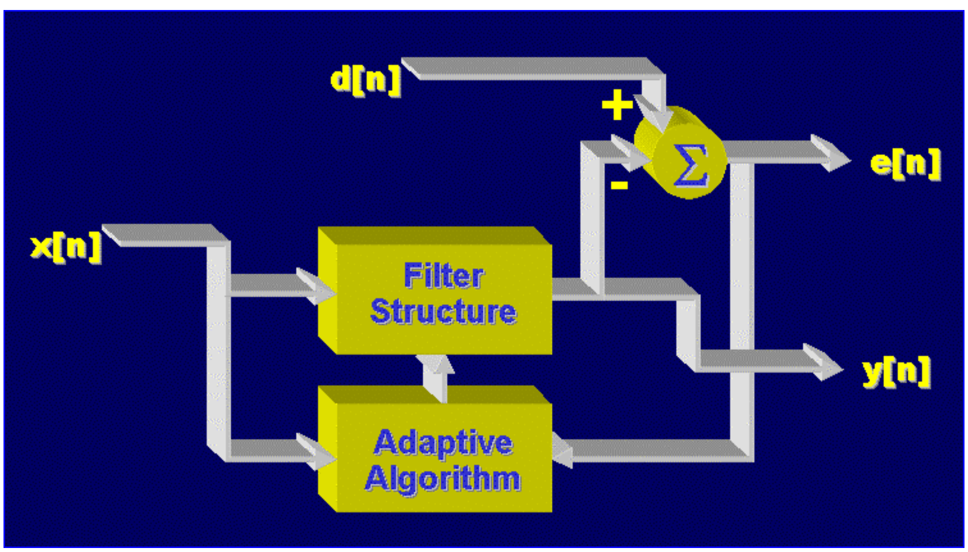

Figure 1 - General Form of an Adaptive Filter

9550 Skillman LB125 * Dallas, TX 75243 * Tel: (214) 343-8525 * Fax: (214) 343-2457 * www.hyperception.com * E-mail: info@hyperception.com

Product Specific Information

The Adaptive Filter implementation described in this application note was created with Hypersignal RIDE Enterprise Edition (HSWN8200). The primary advantage of this package is that all modes of operation are supported: simulation, direct operation on DSP, object code generation, and ANSI C source code generation. Other Hypersignal Block Diagram/RIDE Editions (Standard, Professional, or Enterprise) could also be used to accomplish this same application, although Block Diagram could only support PC-based operation (no direct DSP).

Detailed Description

Transversal Structure Filter

Several types of filter structures can be implemented in the design of the adaptive filters such as Infinite Impulse Response (IIR) or Finite Impulse Response (FIR). In this application note, only FIR structure is implemented. An adaptive FIR filter can be realized using a transversal, symmetric transversal, or a lattice structure. The most common implementation of an adaptive filter is the transversal structure (tapped delay line). The filter output signal y(n) is given by:

$$y(n) = \underline{w}^T(n)\underline{x}(n) = \sum_{1=0}^{N-1} w_i(n)\, x(n-i) \qquad (1)$$

- where $\underline{x}(n) = [x(n)\ x(n-1)\ \ldots\ x(n-N+1)]^T$ is the input vector
- $\underline{w}(n) = [w_0(n)\ w_1(n)\ \ldots\ w_{N-1}(n)]^T$ is the weight vector
- T denotes Transpose
- n is the time index
- N is the order of filter

This example is in the form of a finite impulse response filter as well as the convolution (inner product) of the two vectors $\underline{x}(n)$ and $\underline{w}(n)$.

LMS Adaptation Algorithm

The adaptation algorithm uses the error signal

$$e(n) = d(n) - y(n) \qquad (2)$$

where d(n) is the desired signal and y(n) is the filter output. The input vector x(n) and e(n) are used to update the adaptive coefficients according to a criterion that is to be minimized. The criterion employed in this section is the mean-square error (MSE) ε:

$$\varepsilon = E[e^2(n)] \qquad (3)$$

9550 Skillman LB125 * Dallas, TX 75243 * Tel: (214) 343-8525 * Fax: (214) 343-2457 * www.hyperception.com * E-mail: info@hyperception.com

where E[.] denotes the expectation operator. if y(n) from Equation (1) is substituted into Equation (2), then Equation (3) can be expressed as

$$\varepsilon = E[d^2(n)] + \underline{w}^T(n)R\underline{w}(n) - 2\,\underline{w}^T(n)\underline{p} \tag{4}$$

where $R = E[\underline{x}(n)\underline{x}(n)^T]$ is the NxN autocorrelation matrix, which indicates the sample-to-sample correlation within a signal, and $\underline{p} = E[d(n)\underline{x}(n)]$ is the Nx1 cross-correlation vector, which indicates the correlation between the desired signal d(n) and the input signal vector $\underline{x}(n)$. In order to avoid the complicated computation of R^{-1} and $\underline{p}$, a widely used LMS algorithm is used as an alternative algorithm that adapts the weights on a sample-by-sample basis. This algorithm is a more practical method for finding close approximate solutions to the weights in real time. The LMS algorithm uses the steepest descent method in which the next weight vector $\underline{w}(n+1)$ is increased by a change proportional to the negative gradient of mean-square-error performance surface:

$$\underline{w}(n+1) = \underline{w}(n) - u*\underline{\nabla}(n) \tag{5}$$

where u is the convergence factor which controls the stability of filter, usually small value is tried first for the stability. For the LMS algorithm, the gradient at the nth iteration, $\underline{\nabla}(n)$, is estimated by assuming squared error $e(n)^2$ as an estimate of the MSE in Equation (3). Thus, the expression for the gradient estimate can be simplified to

$$\underline{\nabla}(n) = \delta[e^2(n)]/\delta[\underline{w}(n)] = -2*e(n)*\,\underline{x}(n) \tag{6}$$

Substitution of this instantaneous gradient estimate into Equation (5) yields the Widrow-Hoff LMS algorithm

$$\underline{w}(n+1) = \underline{w}(n) + 2*u*e(n)*\,\underline{x}(n) \tag{7}$$

Implementation

There are two examples in this application note. In the first example, the desired signal is given by a reference signal (Cosine waveform); this design will attempt to extract a desired signal from noise using two adaptive filter (LMS) blocks (the desired signal might be thought to represent a particular anomaly such as an interfering tone in an otherwise good signal). In the second example, the desired signal is the noise signal, and the adaptive filter will attempt to extract the noise (which could then be subtracted from the input signal for noise removal).

9550 Skillman LB125 * Dallas, TX 75243 * Tel: (214) 343-8525 * Fax: (214) 343-2457 * www.hyperception.com * E-mail: info@hyperception.com

Example One – extracting an interfering tone

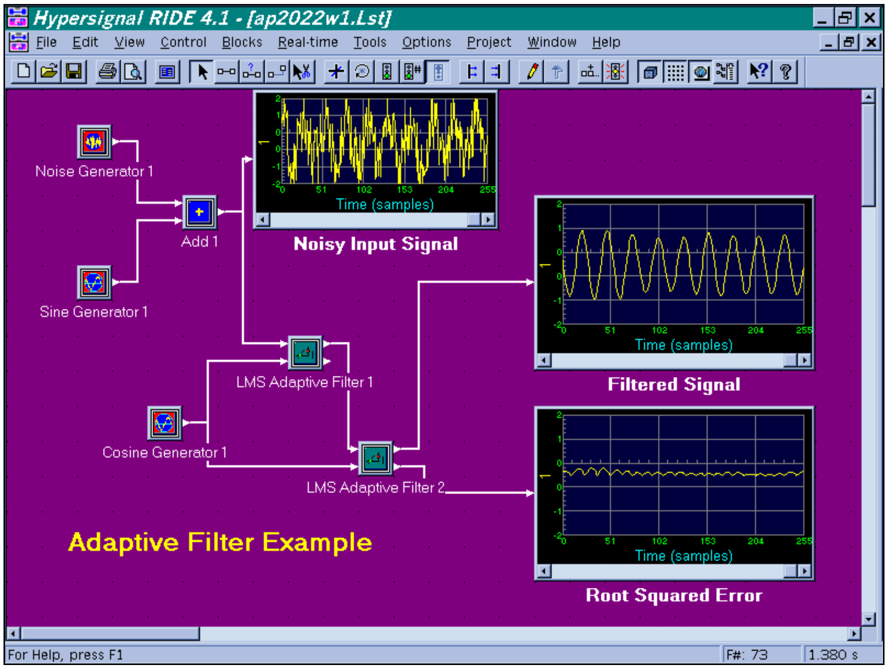

Figure 2 – Adaptive filtering to recover desired signal

As can be seen in Figure 2 above, an input sine waveform is added together with a noise source. The Sine Generator block provides a 300 Hz sine waveform output with a sample rate of 8000 Hz and the Noise Generator block provides the noise source. These two signals are simply combined by adding them together. The resulting signal is then connected to the first input connection of the first adaptive filter block. Note that this example demonstrates how two adaptive filters (LMS) may be cascaded to perform better adaptive filtering.

The Cosine Generator provides a 300 Hz cosine waveform with a sample rate 8000 Hz to the bottom input connection of adaptive filter as the desired signal, or goal. The top input is the raw signal input, while the first output channel is the filtered signal, and the second output is the root squared error.

It can be seen from resulting displays that the adaptive filter is indeed doing a reasonable job of extracting the desired signal. Both the input signal and the resulting filtered signal are displayed. In addition, the root squared error is displayed in the lower right display.

9550 Skillman LB125 * Dallas, TX 75243 * Tel: (214) 343-8525 * Fax: (214) 343-2457 * www.hyperception.com * E-mail: info@hyperception.com

Example Two – extracting a noise source

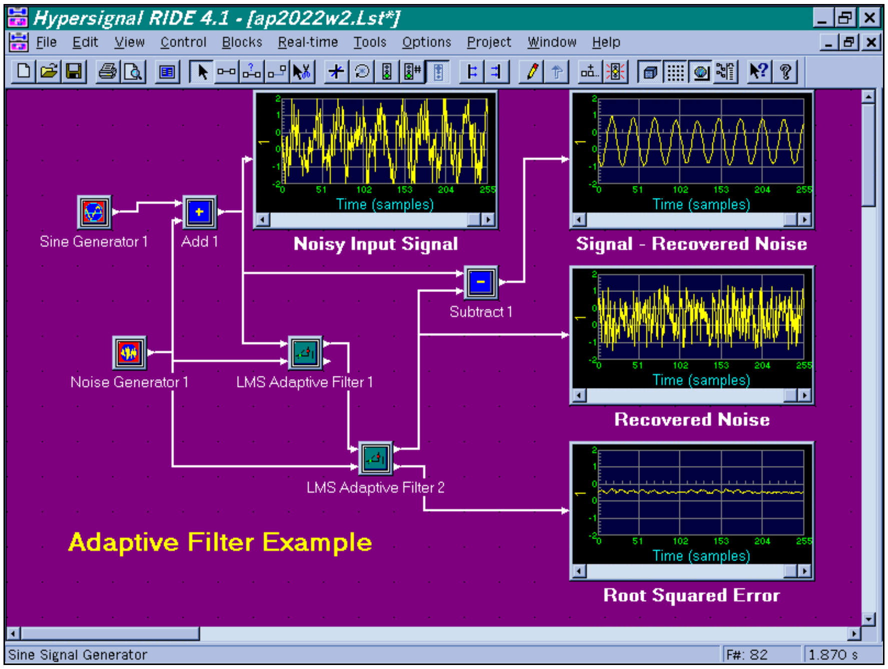

Figure 3 – Extracting Noise

The second implementation of adaptive filter is adaptive noise cancellation. In this case, the desired signal is the noise signal. As can be seen above in Figure 3, the Sine Generator provides a 300 Hz signal with a sample rate of 8000 Hz. The Noise Generator provides a noise signal which is effectively white noise. The two signals are added together using the add block. The resulting signal, as shown in the top left display, is connected to the first connection of the Adaptive Filter (LMS) block. Two Adaptive Filter (LMS) blocks are again cascaded to improve the filtering. The subtract block performs the subtraction of the output of the second Adaptive Filter (extracted noise, shown in the middle right display) from the original noisy signal. The top right display shows the final recovered signal after subtracting the estimated noise signal; it can be seen that the adaptive filtering in this case has been helpful in reducing the overall noise. In actual implementation, selection of the 'desired' signal is not as simple as shown here – but it is quite important and can be the make-or-break factor in an adaptive design.

Applications

Because of their robust performance in the unknown and time-variant environment, adaptive filters have been used widely in telecommunication, radar, sonar, control, and image processing applications. In the first example, a tone was extracted and could have been removed from the original signal; this could be quite useful for eliminating 60 Hz hum since it is so common on AC powered products. In addition, many communication systems may induce particular interfering periodic signals which are not wanted; adaptive algorithms may be

9550 Skillman LB125 * Dallas, TX 75243 * Tel: (214) 343-8525 * Fax: (214) 343-2457 * www.hyperception.com * E-mail: info@hyperception.com

used to reduce or eliminate these signals. The second example showed a simple case of unwanted noise in a signal which is common to many systems, especially communication systems. Other systems such as automotive (both the internal [quiet environment for driver], and external such as adaptive noise cancellation for a new type of engine muffler) could benefit from adaptive filter methods.

Adaptive filter methods are not even limited by one-dimensional processing. Digital Image Processing applications could also make use of these methods. The flexibility of this powerful technique combined with the fast real-time capabilities of today's DSP's allows its uses to be limited only to the creativity of the engineer.

References

Hyperception, Inc. *Hypersignal RIDE Users Manual and Reference Guide*, 1999.

Simon Haykin, *Adaptive Filter Theory*, Prentice Hall, Englewood Cliffs, 1991.

Panos Papamichalis, *Digital Signal Processing Applications With the TMS320 Family, Volume 3*, Texas Instruments, 1990.

Hyperception
The Leader in DSP

9550 Skillman LB125
Dallas, Texas 75243
Tel: (214) 343-8525 * Fax: (214) 343-2457
E-Mail: info@hyperception.com
www.hyperception.com