Adaptive Filtering

This lecture describes some applications of adaptive filters and explains the LMS algorithm.
After this lecture you should be able to: (1) apply an adaptive filter to solve several simple signal processing problems, (2) explain and implement the LMS algorithm, (3) explain the effects of changing λ and select an appropriate value.
This material is covered in Chapter 9 of the text.

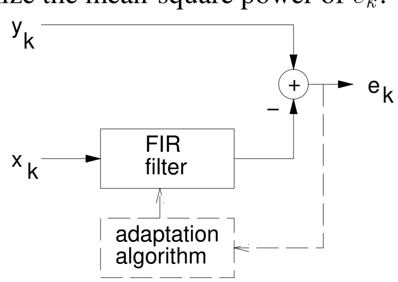
Introduction

Adaptive filters are filters whose characteristics change (adapt) with time to minimize an error criteria. Adaptive filtering is a powerful and flexible technique that has many applications. Some of these include:

- noise cancelation (e.g. for hands-free cellular phones),
- power spectrum estimation,
- interference-canceling antennas,
- adaptive equalizers for telephone-line modems, and
- low bit rate coding of speech.

Although the analysis of adaptive filter performance is difficult the implementation of many adaptive filters is quite simple. In this lecture we will concentrate on the principles and applications of adaptive filters and study only the simplest and most widely used filter adaptation algorithm, the LMS algorithm.

The diagram below shows the structure of an adaptive filter. One of two inputs, x_k, is passed through a filter and subtracted from the other, y_k, to produce an error signal, e_k. Some type of algorithm continuously monitors e_k and "adjusts" the filter coefficients to try to minimize the mean-square power of e_k.



We will restrict ourselves to FIR filters. The N filter coefficients, $w_k(i), i = 0 \ldots N-1$ change with time (k) and the output of the filter will be the FIR sum:

$$\sum_{i=0}^{N-1} w_k(i)x(k-i)$$

Applications

By connecting x, y, and e to different signals we can use adaptive filters for an amazingly wide range of applications.

A simple example is noise cancelation in a vehicle. If x is a noise signal (e.g. from a microphone in the engine compartment) and y contains a desired signal plus some of the noise (e.g. from a microphone in the passenger compartment) then the filter will attempt to minimize the power of e. If the noise and signal are uncorrelated their powers will add and so minimizing e also minimizes the noise component in the e. Thus e will be a "cleaned up" version of y.

Now consider the transfer function of this filter when it has adapted. In order to minimize e it must have the same transfer function as the audio path from the microphone in the engine compartment to microphone in the passenger compartment. An adaptive filter can thus be used to measure an unknown transfer function (this is "system identification").

Now consider a telephone modem where y is a known signal transmitted at the start of each phone call (a "training sequence") and x is the output of the phone line. The adaptive filter will minimize e by setting the filter to the (multiplicative) inverse of the telephone channel. After the training is finished the filter coefficients can be fixed and the modem can work with the output of the (now fixed) filter. This "adaptive equalizer" allows the modem to adapt to different telephone lines.

Finally, consider a signal composed of a noise-like signal plus a periodic component. This signal is fed into y and a delayed version is fed to x. The periodic signal component in both x and y will still be correlated but the noise-like signals in the two branches will be uncorrelated. In order to minimize the power of e the adaptive filter will shift the phase of its input so as to cancel out the periodic component.

The LMS Algorithm

The algorithm that adapts the filter coefficients attempts to minimize the mean-square error (MSE) of e. If the statistics of y and x are known it is possible to compute the coefficients for this optimum filter. This filter is known as a Weiner filter. However, the statistics of y and x are usually unknown and instead we use an algorithm that iteratively adjusts the filter coefficients to minimize the MSE.

This algorithm, the LMS (least mean square) algorithm, is very simple. At each iteration the value of each filter coefficient is updated using the following formula:

$$w_{k+1}(i) = w_k(i) + 2\mu e_k x(k-i)$$

where e_k is the difference between y and the output of the filter and μ is a constant that controls the convergence properties of the adaptive filter.

μ is a compromise between convergence speed and residual error. Large values of μ speed up the convergence but the filter coefficients will tend to oscillate around around the optimum solution. Smaller values of μ reduce this variance but increase the time it takes the filer to adapt.

Another consideration is that the filter may not converge at all if the value of μ is too large. To ensure convergence μ should be in the range

$$0 < \mu < \frac{1}{\lambda_{\text{max}}}$$

where λ_{max} is the largest eigenvalue of the input's (x's) NxN autocorrelation matrix. However, an easier to compute bound is

$$0 < \mu < \frac{1}{3NE\{x^2\}}$$

where N is the number of FIR filter coefficients. In most cases μ will be considerably smaller than this in order to obtain reasonably stable filter coefficients.

Other Algorithms and Filter Structures

There are other filter adaptation algorithms that converge more quickly or have lower residual variance than the LMS algorithm. The most common of these is the "recursive least squares" (RLS) algorithm. This algorithm is more complex but has better performance in many applications.

It's also possible to use filter structures other than the FIR transversal filter. The lattice filter structure is particularly popular because there is an efficient algorithm ("Levinson-Durbin") for computing it's tap weights.