ChannelScience.com
Enabling Technology by Enabling Understanding℠
7300 Cody; Plano TX 75024-3837 USA
csobey@channelscience.com
(972) 814-3441 Voice
(972) 208-9095 FAX

LMS Filter Adaptation in PRML Channels

by Charles H. Sobey and Barry S. Russell, ChannelScience.com

1. Equalization and LMS

Current state-of-the-art PRML (partial-response, maximum-likelihood) read channels detect user data accurately under very adverse signal conditions. PRML channels are used throughout the hard disk drive industry and are rapidly expanding to tape drives, optical drives, and even to Gbit Ethernet. An enabling technology at the core of the PRML channel is the Viterbi detector. This uses a sophisticated algorithm to determine the user's stored data pattern, based on a sequence of samples from the readback waveform. However, there is an equally important signal processing chain that the samples must go through before being presented to the detector.

In the classic read channel architecture [1], this signal chain includes a variable gain amplifier, a continuous-time filter and a sampler, such as an analog-to-digital converter (ADC). The sampling interval is controlled by a phase-locked loop. Samples are passed through a discrete-time filter whose output is processed by the Viterbi algorithm. This detector is designed to expect certain sample target values, which should occur only in particular sequences. Any deviation of the samples from these target values is a potential source of error.

Shaping the signal to the desired target values is one of the goals of the equalization provided by the continuous-time and discrete-time filters. Other equalization goals include limiting noise and setting the channel bandwidth to reduce aliasing due to sampling. To shape the analog signal, the continuous-time filter has a lowpass magnitude response with high frequency boost, and perhaps some time-asymmetry compensation. Typical implementations have four, five or seven poles, and usually one or two zeros. The sampled output of the continuous-time filter is processed by the finite impulse response (FIR) filter. The FIR provides precise control over the sample values.

Figure 1 shows a tapped delay line architecture of a 5-tap FIR. The Ds represent delay elements (registers) that store the input sample values. FIRs for data storage typically have from 5 to 10 taps, depending on the architecture and the division of signal shaping responsibilities between the continuous- and discrete-time domains. Some slower speed applications can use DSP chips, or even software, to perform FIR filtering. However, the data rates in hard disk drives are so fast, currently

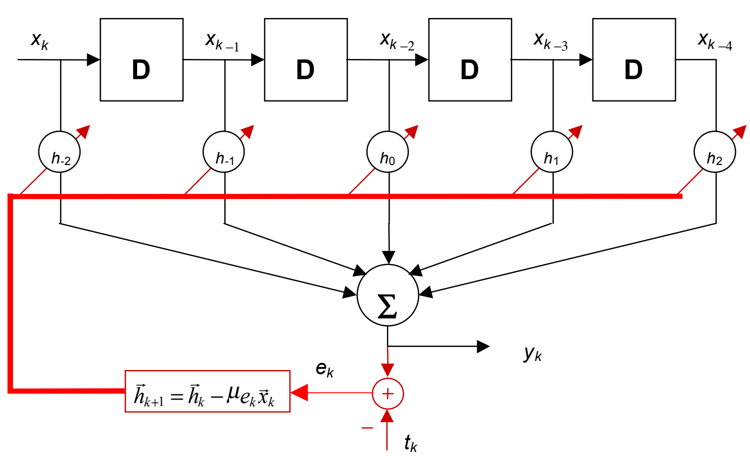


Figure 1. 5-Tap FIR filter with LMS adaptation of the tap weights.

approaching 1Gb/s in the lab, that custom FIR circuitry is needed in the read channel.

In Figure 1, noisy samples of a continuous-time waveform come into the filter from the left. Each time a new sample is input, the other samples in the filter are shifted to the right. This continues until there are no further delay elements left for a particular sample. Five samples are in the filter at any one time. Each sample that is in the filter is multiplied by a particular tap weight, or "coefficient." These products are summed together to provide the filter's output, y_k, at each clock cycle. The sequence of y_k's is passed to the Viterbi detector.

Figure 2 provides an example of the effectiveness of a 5-tap FIR in equalizing to EPR4 targets [2]. The top graph shows samples, x_k, of a noisy lorentzian waveform. Its density is 2.0, specified as the width of an isolated pulse, measured at 50% amplitude, divided by the duration of a channel bit (PW50/T). White gaussian noise has been added to the continuous-time waveform to achieve a signal-to-noise ratio (SNR) of 25dB over the bandwidth of interest. The bottom graph shows the output of the FIR, y_k, clustered around the desired EPR4 target values of -2, -1, 0, +1 and +2. Notice the single frequency "preamble," which lasts until about sample 80. The preamble is used for gain and timing acquisition by the read channel's AGC (automatic gain control) and PLL (phase-locked loop).

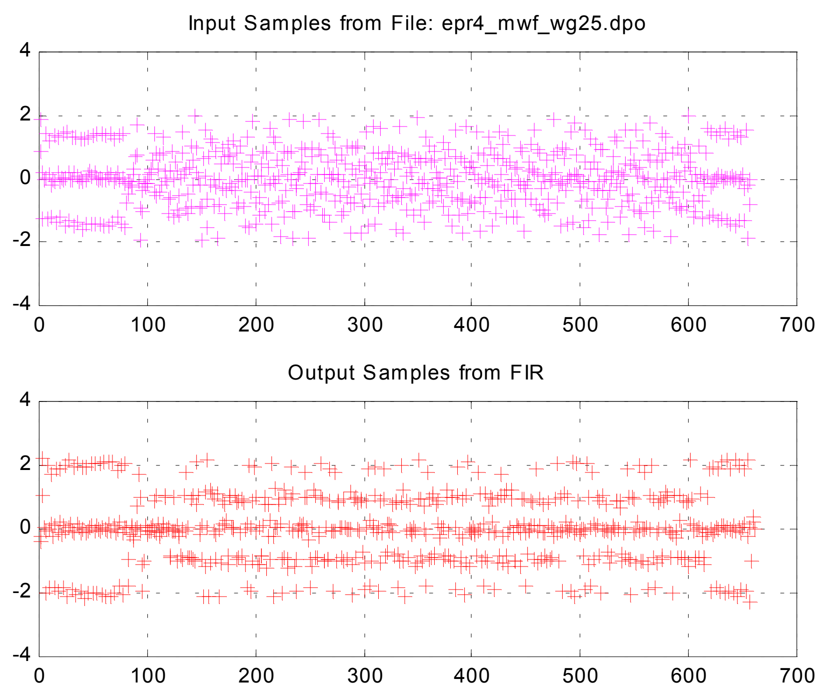


Figure 2. A 5-tap FIR, with properly set tap weights, effectively filters these noisy samples of a lorentzian waveform (top) to the desired EPR4 target values (bottom).

A critical question in setting up any FIR is "What are the best tap weights to use?" Fortunately, this question has been well answered by the least mean square (LMS) adaptation algorithm. Specifically, LMS adjusts the tap weights of the filter to minimize the *mean squared error* at the output of the FIR. The LMS update equation that does this is

$$\vec{h}_{k+1} = \vec{h}_k - \mu e_k \vec{x}_k \, , \tag{1}$$

where h_k is the vector of tap weights at time k, μ is the update parameter or "step-size," e_k is the error term and x_k is the vector of the most recent input samples.

The derivation of (1) and the theory of filter adaptation are well represented in the literature [3,4]. However, practical advice on using LMS adaptation is more difficult to find. In this article, we provide practical insight into the role of the input signal on LMS adaptation. To begin, what equation (1) says in words is this: "The tap weight values for the next sample time are equal to the current tap values, minus a correction factor. The correction factor is based on the input sequence, the error in the FIR output and a weighting term, μ."

This identifies four parameters that are important for robust LMS adaptation. First, the *initial* tap weights, at k=0, must be determined. Second, μ must be selected. This parameter controls how quickly the taps adapt. A large value results in quicker adaptation, but less precise tap weights, more noise at the output of the filter and possibly unstable behavior. A small μ results in slower adaptation, but finer control over the values that the taps take on and typically a lower squared error at the output of the FIR. Gearshift algorithms provide the best of both worlds by starting with a large μ value and later switching to a smaller value for fine-tuning of the tap weights.

Third, the error term must be calculated. Specifically this is $e_k = y_k - t_k$, where t_k is the noise-free target value that corresponds to the current output of the FIR, y_k. Clearly when the error is small, we probably don't need to change the tap weights as much as when the error is large. Not so clear is how we know the correct target value! If we knew that, why would we bother to filter the signal or even detect it at all?

There are two ways that the target value is determined for each output of the FIR: Data-directed and decision-directed target determination. Data-directed adaptation implies that a known training sequence is being read. Therefore the desired target sequence out of the FIR is known. This is a powerful tool that, when properly used, can adapt the FIR to optimal tap weights from almost any initial setting. Decision-directed adaptation works on unknown user data. It typically uses a quick decision device, such as a "slicer" (threshold detector), to determine the correct target value for each output of the FIR. In EPR4, for example, any sample in the range from 0.5 to 1.5 would correspond to a +1 target value.

Each of the LMS parameters mentioned above is important. In fact, they all correspond to channel registers that must be properly set. The remainder of this article, however, is devoted to a fourth LMS parameter that *cannot* be set by channel registers: the input samples, x_k. This variable distributes the error term, e_k, appropriately among all of the tap weights. As such, the input signal plays a key role in determining its own filtering.

2. Spectral Content of the Input Signal

The spectral content of a training pattern must reflect that of the expected signal itself. For example, if a simulated high-density signal *without noise* is input to the FIR, LMS will adjust the taps to equalize the signal to the target values. Typically this involves boosting high frequencies to slim the signal. When a real world signal is filtered using these tap settings, the signal *and* the noise are boosted. Excessive amplification of the noise is referred to as "noise enhancement." This occurs because adapting on the noise-free signal did not impose a mean squared error penalty that indicated that too much high frequency boost was being applied. A further cost of noise enhancement is that the spectrum of the noise at the output of the filter is no longer white. This coloration of the noise degrades the performance of any detector that is optimal in *white* noise.

2.1 Tuning-in to a Single-Frequency Signal

The previous example shows the importance of having the input signal reflect real world conditions. Now, suppose that LMS is adapting on a single-frequency pattern in noise. LMS will adjust the filter to a bandpass response. That is, the magnitude response of the filter will pass the frequency range that contains the signal and aggressively attenuate all other (noise-only) frequencies. If the expected range of inputs is only over this band, this is the right thing to do. However, if the input signal later contains energy at the frequencies that are attenuated, the signal at the output of the filter may be excessively distorted, causing bit errors.

Figure 3 uses the FIR and LMS optimization screens of ChannelScience.com's PRML*pro*™ [5] read channel model to illustrate the effect of adapting on single-frequencies. The top graph in Figure 3 shows the trajectories of the FIR's five taps during adaptation. The initial tap setting is 0, 0, 1, 0, 0. The trajectories show how (data-directed) LMS adjusts each tap, based on a noisy input waveform. The waveform starts with a low frequency pattern (0.125 f/f_s) and its 3rd harmonic (0.375 f/f_s). It ends with a mid-frequency pattern (0.25 f/f_s). The center-left graph is a plot of the final tap weights. The center-right graph displays the spectrum of the FIR output in red, the magnitude response of the final FIR filter in blue and the corresponding group delay in magenta. The bottom graph plots the FIR output samples.

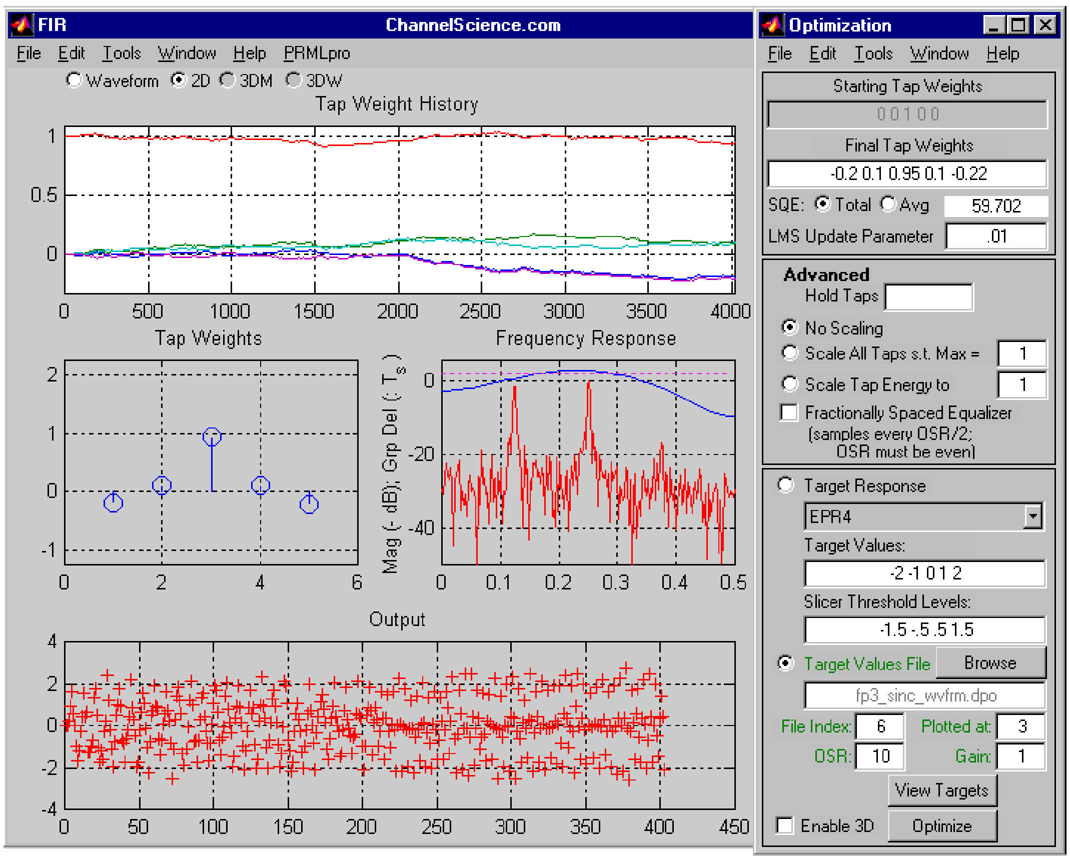
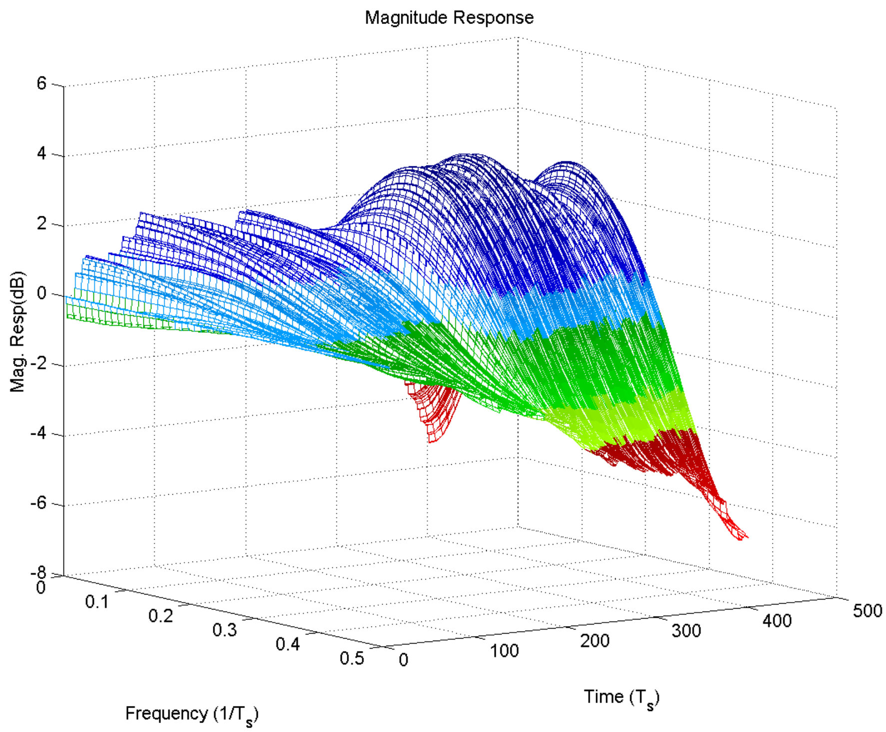


Figure 3. The top plot shows the trajectories of the FIR's five taps as LMS adapts them to selectively filter each of the two unique single-frequency portions of the input waveform. The input consists of lorentzian pulses at a signal to additive white gaussian noise ratio of 15dB. The LMS optimization screen of PRML*pro*™ is overlayed on the right hand side of the FIR screen in this figure.

Notice how LMS tries to tune the filter to each frequency. Specifically, the center tap, 3, stays around its initial value of 1. But the taps on either side of the center tap, 2 and 4, move from zero to slightly positive values, creating a lowpass filter. Around sample index 2000 (in the top graph, 200 in the bottom graph), the higher frequency portion of the waveform reaches the filter. At this time, taps 1 and 5 move to negative values, creating a bandpass magnitude response.

Additional insight is provided by the two novel graphs in Figure 4, which also are based on PRML*pro*™ output. The vertical axis in both graphs is the magnitude response of the FIR, as a function of the normalized "Frequency" axis. The "Time" axis indicates the procession of samples through the LMS



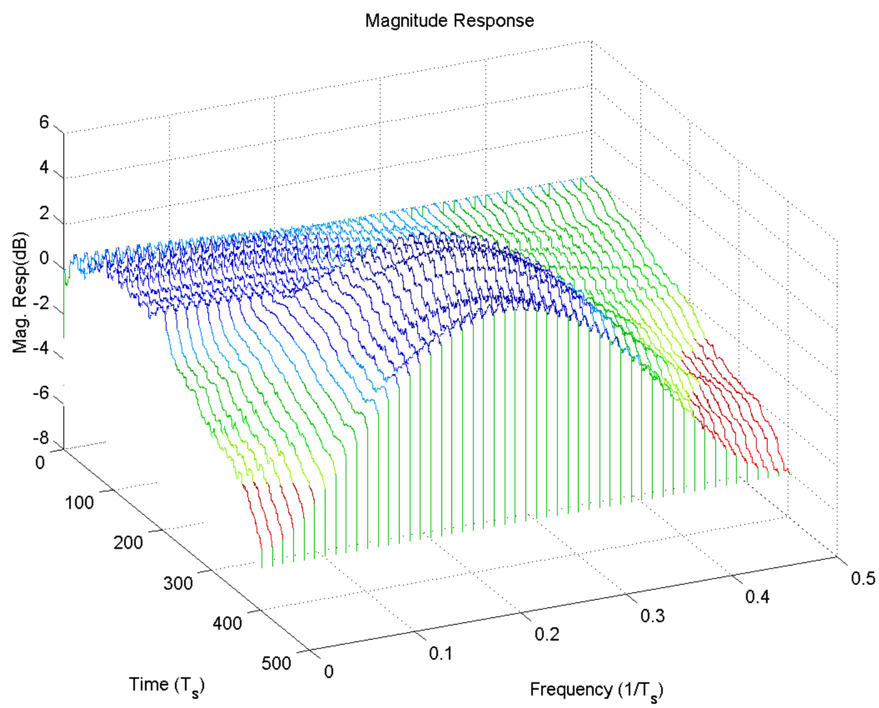


Figure 4. Two views of the magnitude response of the FIR, as LMS adapts it to "tune-in" to the spectral content of the input waveform.

algorithm. These plots provide two different views of the changing characteristics of the frequency response of the FIR as LMS adapts it to the changing spectral content in the input samples.

To examine the results in Figure 4 in more detail, start on either graph at time index 0. Recall that the taps are set to 0, 0, 1, 0, 0 initially. This is an all-pass filter characteristic. That is, the magnitude response is flat, providing a gain of 0dB at all frequencies. As the low frequency tone and its harmonics enter the FIR, LMS adapts the coefficients to pass the lower frequencies while attenuating the noise at the higher frequencies. The lowpass filter characteristic would be more pronounced if the 3^{rd} harmonic of the signal were not also in the passband of our channel.

At time index 200, the mid-frequency tone replaces the low frequency tone and enters the FIR. This tone’s harmonics are outside the passband, so LMS adapts the FIR to a bandbass frequency response, centered at a normalized frequency of 0.25. This attenuates the noise in the frequencies above and below the center tone.

From this example we can see the importance of selecting a data-directed training pattern whose spectral characteristics reflect those of the real world data. Further, in decision-directed mode, it is important that the *user’s* data regularly reflects the broadest possible spectral content. This is often accomplished by passing their data pattern through a deterministic scrambler before it is stored on the medium. The same scrambling operation is applied to the detected sequence to unscramble it back to the original data pattern.

2.2 Thermal Aspserities (TAs) and LMS

Thermal asperities [6] cause the output of a (G)MR head to quickly exhibit an offset of approximately 2 to 8 times the normal signal range. The output slowly returns to normal as the read stripe returns to thermal equilibrium. The waveform in the top graph of Figure 6 contains three TAs. It may appear that the effects of TAs are devastating to the PRML channel. However, analyzing the response of the LMS algorithm to these TAs shows a way to mitigate their effects in the channel.

The bottom graph in Figure 5 shows the output of the FIR when LMS is allowed to adapt to the TAs. The results are surprisingly good. There are two reasons for this. One is that data-directed adaptation was used for this example. Because we are adapting on known data, such as a training pattern, the correct error term is always supplied to the LMS algorithm.

The other reason for the good performance is suggested by the red spectrum plot in the center-right graph of Figure 5. The slow decay of the thermal asperities causes a concentration of power at low frequencies (around 0). This is in contrast to the ideal EPR4 spectrum that has a smooth roll-off to DC. In fact, class IV partial response channels are inherently DC-free. For this reason, ac-coupling (highpass filtering) is used many times in the readback signal chain to block very low frequency content. Figure 6 shows the changing highpass characteristics of the FIR as LMS adapts on the three TAs.

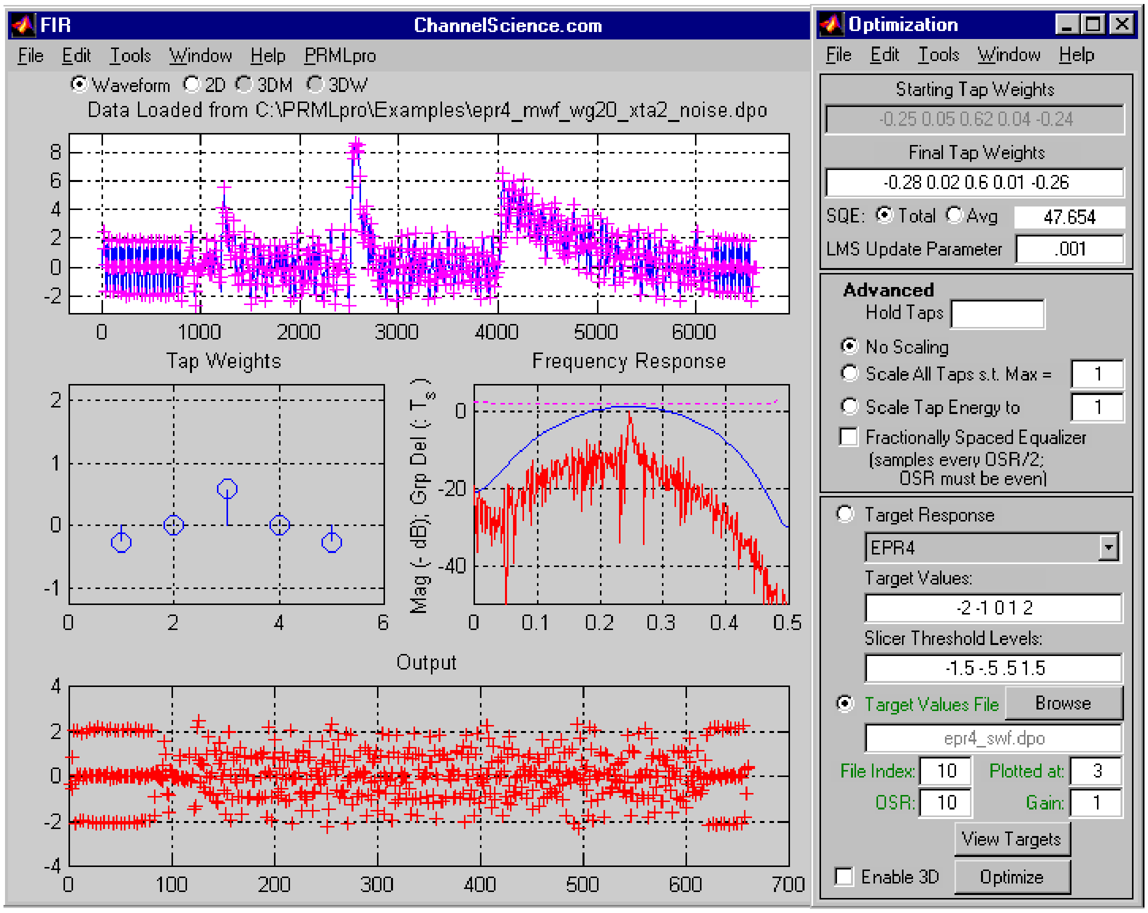


Figure 5. Data-directed LMS adaptation provides a much more controlled output.

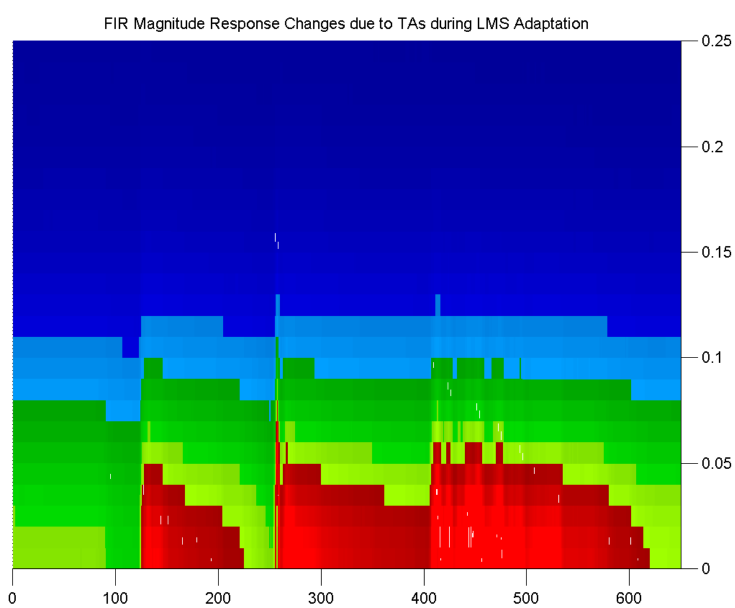


Figure 6. LMS responds to TAs by increasing the highpass frequency of the FIR.

In Figure 6, the horizontal axis is the bit index (time), the vertical axis is the normalized frequency from DC (0) to the center of the passband (0.25). The color of the plot indicates the magnitude of the FIR's response. Blue denotes amplification, green represents mild attenuation and red indicates strong attenuation. Each TA corresponds to a red area in the plot. The "height" of each red area indicates the frequency range that receives the strongest attenuation. The duration of this interval reflects the amplitude and the rates of rise and decay of the TA. The first TA is quick with moderate amplitude. The second TA is the same rate, but achieves twice the maximum amplitude. The third TA has the same maximum amplitude as first, but its rise and decay rates are one half and one tenth, respectively, of those of the other asperities.

From Figure 6 we see that the LMS algorithm indicates that we should increase the frequency of the read channel's highpass pole when a TA is encountered. In fact, this is a common feature of PRML read channel thermal asperity compensation circuits [7]. Regardless of the highest frequency content of the TA, there is a limit on how much the high-pass pole can be pushed before attenuating the data itself.

3. Conclusion

The input signal to the FIR/LMS combination must represent real world signals in terms of noise and spectral content. This is easy to control in a training pattern, but the user must be able to store *any* data pattern – even single frequencies. To avoid problems for LMS adaptation as well as for other channel functions, the user's data is scrambled before storage. This makes the stored data pattern more likely to have broad, representative spectral content.

From the TA results presented above, it may appear that the FIR/LMS combination is capable of fixing any problem that your data encounters. Of course, this is *not* the case. Remember that data-directed adaptation implies that we know what the data should be. When adapting on user data, we often use a simple threshold detector to make decisions about what the correct samples should be. Excessive mistakes in decision-directed adaptation can cause the LMS algorithm to become unstable, adapting to unusable FIR settings.

There are also practical circuit implementation issues that degrade performance during TAs. Specifically, large TAs can cause the output of amplifiers to distort and the output of the ADC to clip. This can eliminate the data signal on top of the TA. Even with good highpass filtering, the data signal is gone. This is one reason TA compensation typically takes place earlier in the read channel or preamp, in the continuous-time analog circuitry. Finite precision of the FIR tap weights is another concern. The fine-tuning of the floating-point taps, that provides such good TA performance in simulations, might not be available in the read channel chip.

5. References

[1] R.D. Cideciyan, *et al.*, "A PRML System for Digital Magnetic Recording," *IEEE Journal on Selected Areas in Communications*, Vol. 10, No. 1, January 1992, pp. 38-56.

[2] H.K. Thapar and A.M. Patel, "A Class of Partial Response Systems for Increasing Storage Density in Magnetic Recording," *IEEE Trans. Magn.*, Vol. MAG-23, No. 5, September 1987, pp. 3666-3668.

[3] S. Haykin, *Adaptive Filter Theory*, 3rd Edition, Publisher Prentice-Hall, Inc., 1995.

[4] E.A. Lee and D.G. Messerschmitt, *Digital Communication*, 2nd Edition, Kluwer Academic Publishers, 1994.

[5] PRML*pro*™, available for download at www.ChannelScience.com

[6] E. Sawatzky, "Thermal Asperities: MR Heads Face New Dangers," *Data Storage Magazine*, February 1998, pp. 49-54.

[7] B.Bloodworth, *et al.*, "A 450-Mb/s Analog Front End for PRML Read Channels," *IEEE Journal of Solid-State Circuits*, Vol. 34, No. 11, November 1999, pp. 1661-1675.