

PAM Equalization

Contents

- Slide 1** Introduction
- Slide 2** Introduction (cont. 1)
- Slide 3** System Description
- Slide 4** System Description (cont. 1)
- Slide 5** System Description (cont. 2)
- Slide 6** System Description (cont. 3)
- Slide 7** LMS Adaptive Equalization Theory
- Slide 8** LMS Equalization Theory (cont. 1)
- Slide 9** LMS Equalization Theory (cont. 2)
- Slide 10** LMS Equalization Theory (cont. 3)
- Slide 11** LMS Equalization Theory (cont. 4)
- Slide 12** Experiments for Adaptive Equalization
- Slide 13** A Handshaking Sequence
- Slide 14** Handshaking Sequence (cont. 1)
- Slide 15** Handshaking Sequence (cont. 2)
- Slide 16** Looped Back Experiments
- Slide 17** Looped Back Experiments (cont. 1)
- Slide 18** Looped Back Experiments (cont. 2)
- Slide 19** Looped Back Experiments (cont. 3)
- Slide 20** Looped Back Experiments (cont. 4)
- Slide 21** Looped Back Experiments (cont. 5)

Slide 22	Looped Back Experiments (cont. 6)	
Slide 23	Looped Back Experiments (cont. 7)	
Slide 24	Looped Back Experiments (cont. 8)	
Slide 25	Looped Back Experiments (cont. 9)	
Slide 26	Experiments with the Transmitter Connected to Another DSK	
Slide 27	Experiments with the Transmitter Connected to Another DSK (cont.	1)
Slide 28	Experiments with the Transmitter Connected to Another DSK (cont.	2)
Slide 29	Experiments with the Transmitter Connected to Another DSK (cont.	3)
Slide 30	Experiments with the Transmitter Connected to Another DSK (cont.	4)
Slide 31	References	

Introduction

See the class web site for the full text for this chapter.

- The PAM transmitter you created in “Chapter 11 Digital Data Transmission by Baseband Pulse Amplitude Modulation” will be used here.
- The transmitter output will be passed through a channel simulation filter to add intersymbol interference (ISI).
- An adaptive FIR filter using the least mean-square (LMS) algorithm will be used to eliminate most of the ISI.
- In the first experiments the transmitter output samples with ISI will be looped back to an adaptive equalizer in the same DSK program as the transmitter to avoid having to implement the interpolator of Chapter 12 and the symbol clock tracker of Chapter 11.

Introduction (cont. 1)

- Next, the transmitter output will be sent to the codec output and connected to the codec input in another PC. Then the interpolator and symbol clock tracker will have to be implemented also.
- Initial equalizer training will be accomplished by using a 15-symbol repeating sequence and the LMS algorithm. See Tretter [2, Chapter 11] for a rapid method of computing the initial equalizer tap values using the DFT and IDFT when a known repeating sequence is transmitted. See “Chapter 15 QAM Receiver II” for a detailed presentation of adaptive equalization for quadrature amplitude systems. A method for initially adjusting the equalizer called “blind equalization” when the transmitted sequence is unknown is also discussed in Chapter 15 and can be modified for PAM equalizers.

System Description

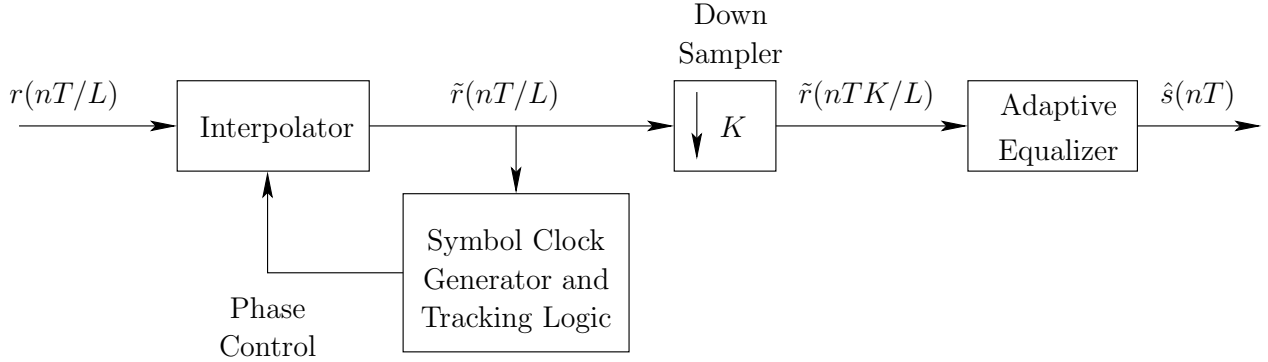


Figure 1: PAM Receiver

A block diagram of a PAM receiver with an adaptive equalizer is shown in the figure above.

- T is the symbol period. The received signal $r(t)$ is sampled L times per symbol, that is, with sampling period T/L resulting in the sequence $r(nT/L)$. $L = 4$ will be used for the experiments of this chapter.
- The symbol clocks in the transmitter and receiver will be slightly different because they are generated by hardware at different locations and, possibly, by relative motion between the transmitter and receiver. Therefore, the receiver

System Description (cont. 1)

must acquire the symbol clock in the received signal. A method for doing this is to pass the received sequence $r(nT/L)$ through a variable phase interpolator. Implementations of the interpolator are presented in Chapter 12.

- Let the interpolator output be $\tilde{r}(nT/L) = r[(n + \delta)T/L]$ where δ is the time shift introduced by the interpolator as a fraction of the sampling period. Then $\tilde{r}(nT/L)$ is applied to a symbol clock tone generator and phase tracking logic which is discussed in Section 12.3 and shown in Figure 12.1. This symbol clock tracking loop locks to the positive zero crossings of the clock tone generator output.

- The interpolator output is then down sampled by a factor of K to give the sequence $\tilde{r}(nTK/L)$ which is applied to the adaptive equalizer. K and L are chosen so that $L/K = Q$ is an integer.

System Description (cont. 2)

Thus the adaptive equalizer operates on samples taken with period $TK/L = T/Q$. For the experiments in this chapter, you will use $L = 4$ and $K = 2$ so $Q = 2$ and $TK/L = T/2$.

- $T/2$ spacing allows the equalizer to have a frequency response that can compensate for received signals like raised cosine signals whose bandwidth extends somewhat beyond the Nyquist frequency of $\omega_s/2 = \pi/T$.
- The equalizer is an FIR filter with a delay line and taps spaced $TK/L = T/Q$ which is $T/2$ for our experiments. A block diagram of the equalizer is shown in Figure 2. The blocks labeled $z^{-K/L}$ represent delays of $KT/L = T/Q$. It is said to be a *fractionally spaced* equalizer.
- The equalizer output is computed once per symbol and gives an estimate $\hat{s}(nT)$ of the transmitted symbol.

System Description (cont. 3)

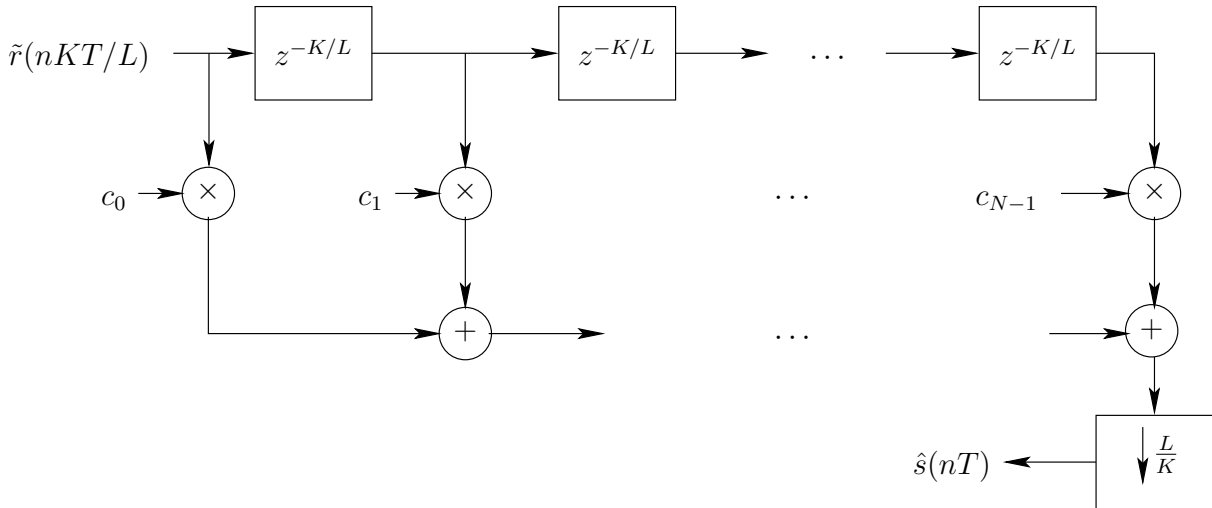


Figure 2: Equalizer for PAM

- A fractionally spaced equalizer also acts as an interpolator and automatically adjusts for signal time shifts. The receiver symbol clock must be locked in frequency to the transmitter clock. If there is a clock frequency difference, the equalizer will compensate for the drifting clocks and will fail when the correct timing falls off its ends.

LMS Adaptive Equalization Theory

The equalizer is an FIR filter with N adjustable tap values $c_0, c_1, \dots, c_{N-1}$.

- The taps are spaced by $\tau = TK/L$ which for the experiments in this chapter will be $\tau = T/2$.
- The taps are updated to minimize the mean-square-error between the equalizer output and the ideal transmitted symbol. They are updated only at the symbol instants nT .
- Samples are entered into the delay line every $TK/L = T/Q$ seconds but the equalizer output is only computed every T seconds. The intermediate equalizer outputs are not needed.
- The equalizer output at time nT is

$$\hat{s}(nT) = \sum_{k=0}^{N-1} c_k \tilde{r} \left(nT - k \frac{T}{Q} \right) = \sum_{k=0}^{N-1} c_k \tilde{r} (nT - k\tau) \quad (1)$$

- Suppose the actual transmitted sequence is a_n . Then the instantaneous equalizer output error is

$$e(nT) = a_n - \hat{s}(nT) \quad (2)$$

LMS Equalization Theory (cont. 1)

- The mean-squared output error is

$$\begin{aligned}\Lambda &= E\{e^2(nT)\} = E\{(a_n - \hat{s}(nT))^2\} \\ &= E\left\{\left[a_n - \sum_{k=0}^{N-1} c_k \tilde{r}(nT - k\tau)\right]^2\right\} \quad (3)\end{aligned}$$

- The tap values that minimize Λ can be found by setting the partial derivatives of Λ with respect to the tap values to zero. The partial derivate of Λ with respect to tap c_i is

$$\begin{aligned}\frac{\partial \Lambda}{\partial c_i} &= 2E\left\{e(nT) \frac{\partial e(nT)}{\partial c_i}\right\} \\ &= -2E\{e(nT) \tilde{r}(nT - i\tau)\} \quad (4)\end{aligned}$$

Setting the partial derivatives to zero results in a set of N linear equations in N unknowns which is essentially the same as the set for QAM presented in Section 15.1.1 and involves inverting the $N \times N$ correlation matrix for the delay line contents.

LMS Equalization Theory (cont. 2)

- An iterative technique for converging to the solution is used in practice. The partial derivative points in the direction of increase in Λ with respect to the tap value. Therefore, incrementing a tap value by a small step in the direction opposite to the partial derivative of Λ with respect to that tap value will decrease Λ .
- The expectation $E\{e(nT)\tilde{r}(nT - i\tau)\}$ in (4) can be approximated by just $e(nT)\tilde{r}(nT - i\tau)$. The factors $e(nT)$ and $\tilde{r}(nT - i\tau)$ can be directly measured in the receiver. In fact, $\tilde{r}(nT - i\tau)$ is just the equalizer input sample sitting at tap i .
- Let μ be a small positive scale factor. Then the LMS tap update algorithm is

$$\begin{aligned} c_i(n+1) &= c_i(n) + \mu e(nT)\tilde{r}(nT - i\tau) \quad (5) \\ &\text{for } i = 0, \dots, N-1 \end{aligned}$$

A block diagram illustrating the tap update algorithm is shown in Figure 3.

LMS Equalization Theory (cont. 3)

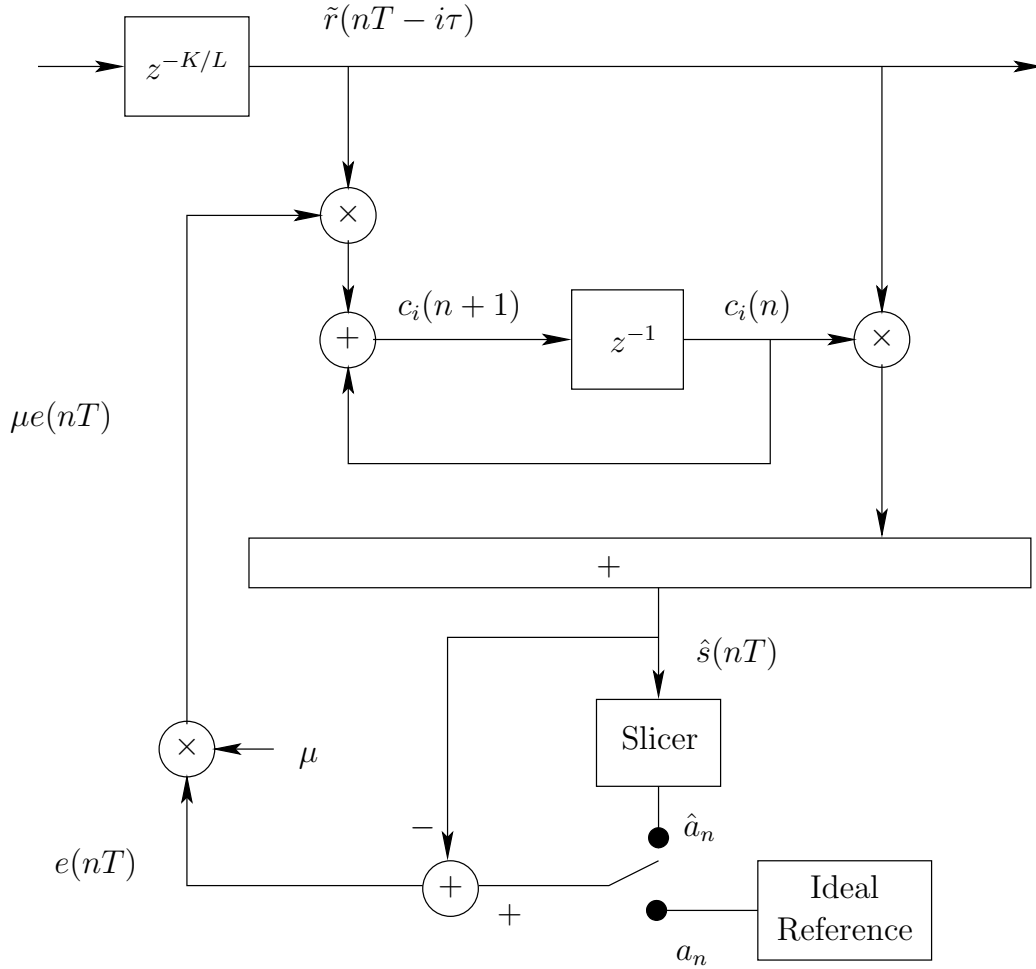


Figure 3: LMS Updating for a Tap at time nT

Tap updates are performed only at the symbol instants nT . The switch is initially connected to a_n for ideal reference training and changed to $\hat{a}_n$ for decision directed adaptation after convergence. Widrow [3] popularized the algorithm .

LMS Equalization Theory (cont. 4)

- The scale factor μ must be chosen small enough to guarantee stability of the LMS algorithm.

Let the eigenvalues of the correlation matrix for the delay line contents be $\{\lambda_i\}$. Then it can be shown that the criterion for stability of tap convergence is

$$0 < \mu < 1 / \max_i \{\lambda_i\} \quad (6)$$

See 15.1.2 for more details on convergence.

- The scale factor μ determines the speed and accuracy of convergence. The tap values hover about the optimum solution when steady-state is reached. A larger μ results in faster convergence but more tap jitter. Smaller values result in slower convergence but less tap jitter. An approach in practice is to use a large μ for initial training and then switch to a smaller one during data detection.

Experiments for Adaptive Equalization for PAM

- As usual, set the sampling rate to 16,000 Hz.

In all the experiments use an $N = 30$ tap $T/2$ spaced equalizer. Thus the equalizer delay line spans 15 symbols.

- You can use the PAM transmitter you created in Chapter 11 as a starting point. The transmitter you created there generates four output samples per symbol, so the symbol rate is $f_s = 4000$ symbols/second.

- Only two-level PAM will be investigated.

Internally in your program use the levels 3 and -3. Scale the interpolation filter bank outputs by an appropriate value to use a significant part of the dynamic range of the DAC, convert the samples to integers, and put them in the left output channel. Put a baud sync signal in the right channel. These steps should already be in the program you created for Chapter 11.

A Handshaking Sequence

- Include an integer variable in your program to count the number of transmitted symbols to determine when different parts of the handshaking sequence should run.
- Modify your PAM transmitter to generate the following handshaking sequence.
 1. First send $1/2$ second of silence by transmitted 2000 symbols with level 0. This will allow a receiver to decide that no signal is present and initiate code to detect signal presence.
 2. Next send $1/2$ second of symbols that alternate between 3 and -3 for 2000 symbols. This is called a *dotting sequence*. The transmitter output will be a sine wave at 2000 Hz which is half the symbol rate. It will provide a strong signal to allow a receiver to detect the presence of a signal, adjust its AGC, and lock its symbol clock tracking loop to the received symbol clock.

Handshaking Sequence (cont. 1)

3. Next transmit a two-level symbol sequence that repeats every 15 symbols for 2000 symbols. The sequence period is the same as the number of symbols spanned by the equalizer delay line. The receiver will use this known sequence for ideal reference equalizer training.

- Generate the sequence with a 4-stage feedback shift register with the connection polynomial $h(D) = 1 + D + D^4$ as explained in Chapter 9. Set the initial state of the shift register to any non-zero value of your choosing.

- If the binary sequence generated is $b(n)$, then

$$b(n) = b(n - 1) \oplus b(n - 4) \quad (7)$$

where $\oplus$ is modulo 2 addition. Include a listing of the sequence in your lab report.

Handshaking Sequence (cont. 2)

- Map the logical binary value 0 to symbol level +3 and value 1 to symbol level -3. An equation for this mapping if $b(n)$ is considered to be a real number is

$$a(n) = 3 - 6 b(n) \quad (8)$$

- Scale the interpolation filter bank outputs by the same value as before, convert them to 16-bit two's complement integers, and send them to the codec in the left channel along with a baud sync signal in the right channel.
4. After the periodic sequence, continually send a two-level pseudo-random symbol sequence based on a 23-stage maximal length feedback shift register generator with the connection polynomial $h(D) = 1 + D^{18} + D^{23}$. This will allow finer equalizer adjustment and also simulate binary random customer data.

Experiments with the Transmitter Output Looped Back Internally

For this experiment loop the unscaled transmitter output samples back to your receiver equalizer code internally in the same program as the transmitter. This will significantly simplify the receiver program because

- you can use the transmitter's symbol counter to determine the handshaking phase,
- you will not have to detect signal presence,
- and not have to implement an interpolator and symbol clock tracking logic. Clock tracking is not an issue because the transmitter and receiver code are running in the same DSP with the same clock.

Actually, you will filter the original interpolation filter bank output with a channel simulation filter to introduce inter-symbol-interference (ISI) in the transmitted signal and loop these filtered samples back to the receiver.

Looped Back Experiments (cont. 1)

Create a receiver program to perform the following tasks:

1. Introduce ISI by passing the original transmitter interpolation filter bank output samples through an IIR filter of the form

$$\begin{aligned} G(z) &= \frac{1.5(1 - b_1)(1 - b_2)}{(1 - b_1 z^{-1})(1 - b_2 z^{-1})} \\ &= \frac{c}{1 + d_1 z^{-1} + d_2 z^{-2}} \end{aligned} \quad (9)$$

where $c = 1.5(1 - b_1)(1 - b_2)$, $b_1 = 0.9$, $b_2 = 0.7$, $d_1 = -(b_1 + b_2)$, and $d_2 = b_1 b_2$.

Scale the filtered samples, which occur at a 16 kHz rate, for the codec, convert them to integers and send them to the left codec channel along with a baud clock sync signal in the right channel.

Looped Back Experiments (cont. 2)

Arrange your program so that the channel simulation filter can be included or not.

- First do not include the filter and observe the nearly ideal eye diagram on the oscilloscope. The two-level eyes will be almost completely open and the transmitted symbols can be determined without error by observing the polarity of the received signal at the symbol instants. You should be able to see the different phases of the hand shaking sequence.

- Next enable the channel simulation filter and you should see that the eye is completely closed and that the transmitted symbols cannot be determined from this signal.

2. Down sample the output of the channel simulation filter by a factor of two and put the resulting samples into the delay line of a $T/2$ spaced 30-tap adaptive equalizer. That is, put every other channel filter output into

Looped Back Experiments (cont. 3)

the equalizer delay line. The channel output samples occur at a 16 kHz rate, so the down sampled sequence samples occur at an 8 kHz rate. Do this for the floating point samples without scaling for the codec.

3. Cyclic Equalization

- Wait for the silence and dotting phases of the transmitted handshake sequence to end based on the transmitter's symbol counter. Then wait 30 more $T/2$ samples for the equalizer delay line to fill up with samples from the 15-symbol repeating phase.
- Make a replica in your receiver of the 15-symbol sequence generator. Use it as an ideal reference and update the equalizer taps once per symbol, that is, once every second delay line input sample, using the LMS algorithm specified by (6). Try $\mu = 0.002$.

Looped Back Experiments (cont. 4)

- Do not compute the equalizer output between symbol instants. Just shift a new $T/2$ sample into the delay line.
- After several cycles of the 15-point symbol sequence, the output of the channel filter will also repeat every 15 symbols. Let one period of the sequence be $a_0, \dots, a_{14}$ and its DFT be $A_0, \dots, A_{14}$. According to the IDFT formula

$$a_n = \frac{1}{15} \sum_{k=0}^{14} A_k e^{j \frac{2\pi}{15} nk} = \frac{1}{15} \sum_{k=0}^{14} A_k e^{j \left(k \frac{\omega_s}{15}\right) nT}$$

for $n = 0, \dots, 14$ (10)

where $\omega_s = 2\pi/T = 2\pi \times 4000$ is the radian symbol rate. Thus the repeating sequence probes the channel only at the discrete frequencies $k \times 4000/15$ for $k = 0, \dots, 14$.

- At the end of Cyclic Equalization and before Tap Rotation, extract and plot the equalizer tap sequence. Do this only once since the program must be stopped to read

Looped Back Experiments (cont. 5)

the tap values so the system will not run in real time. After extracting the tap values once, reload and restart the program without looking at the tap values at the end of Cyclic Equalization so the program runs in real time.

4. Tap Rotation

The periodic sequences in the transmitter and receiver will usually not be in phase with each other. However, the equalizer will set up to optimize for periodic sequence in the receiver. It will automatically shift the received sequence to align it in time with the local ideal reference.

- The position of the largest equalizer tap indicates the shift required for this alignment. The largest tap may not be near the center of the equalizer delay line and this will not be good when a random data signal with a distributed spectrum is received. A solution to this problem is given next.

Looped Back Experiments (cont. 6)

- At the end of the repeating 15-point sequence phase, determine the location of the tap with the largest magnitude. Then rotate the equalizer tap sequence an integer number of symbols, that is, by a multiple of two positions, to move the largest tap near the center of the delay line.

Cyclically rotating an N -point sequence ℓ positions to the right results in its DFT being multiplied by $\exp(-j2\pi\ell k/N)$ and just adds a linear phase shift to the equalizer frequency response at the probe frequencies. The effect is to delay the 15-point repeating equalizer output sequence by ℓ symbols.

- Also at the end of the periodic sequence phase, “gear shift” the update scale factor μ to a smaller value to achieve finer convergence.

Looped Back Experiments (cont. 7)

5. Decision Directed Equalization

By the end of cyclic ideal reference training, the equalizer outputs at the symbol instants should be close to the ideal symbol values and the ideal transmitted symbols can be correctly determined with high probability by quantizing, that is, *slicing*, the equalizer outputs to the nearest ideal symbol levels ± 3 .

- From the end of the periodic ideal reference phase onward, use the sliced equalizer outputs as the ideal symbol reference values. This is called *decision directed equalization*.
- The equalizer will continue to adapt some after the periodic training phase has ended and a random customer data sequence is transmitted because the spectrum of the transmitted signal becomes distributed over the signal bandwidth rather than discrete lines at the 15 probe frequencies.

Looped Back Experiments (cont. 8)

- The receiver has no idea what random sequence is transmitted and cannot use ideal reference training at this point. However, it can use decision directed training once the eye is open. In practice, the equalizer is continually adapted during data transmission to track small deviations in symbol clock timing and channel changes.
- Plot the equalizer coefficients after they have converged with Decision Directed Equalization and compare them with the coefficients at the end of Cyclic Equalization and Tap Rotation.

6. Observing the Equalizer Output

The equalizer output is computed only at the symbol instants, so the question of how to observe it in real-time using the lab equipment arises. You cannot write it to the console or to a file in the PC because the program will not run in real-time then.

Looped Back Experiments (cont. 9)

- A method to view the equalizer output on the oscilloscope is to apply it to an interpolation filter bank similar to the one in the transmitter to generate four samples per symbol and send the resulting samples at the 16 kHz rate to one output channel of the codec. The interpolation filter bank should be based on a raised cosine filter, not a square root of raised cosine filter, so it introduces no inter-symbol-interference. Send a baud sync signal to the other output channel. Make sure to scale the equalizer outputs appropriately to use a large portion of the dynamic range of the codec's DAC. You can then observe an eye diagram on the oscilloscope and see the equalizer converge in real-time.

- Experiment with different values of the equalizer update scale factor μ to see how it affects the convergence speed and accuracy.

Experiments with the Transmitter Output Connected to Another DSK

Now you will make a PAM receiver that works in a different DSK than the one containing the transmitter. It will contain most of the systems that a real world receiver requires.

- Use the transmitter code you created for the previous section in one DSK. The transmitter should use the handshaking sequence specified there.
- Connect the line out of the transmitter DSK to the line in of a DSK in another PC where you will make the receiver. Use a 16 kHz sampling rate for the codec in the receiver DSK.

Connected to Another DSK (cont. 1)

- Your receiver code should perform the following tasks:

1. The receiver should monitor the 16 kHz input samples to detect the presence of a PAM signal. Devise a method to relatively quickly detect received signal energy, the presence of a 2 kHz dotting tone, or a combination of both criteria. Once a signal is detected, continue to monitor the input energy to determine when the input signal stops.
2. Start a symbol counter when an input signal is detected to determine when different phases of the handshaking sequence are present.

Connected to Another DSK (cont. 2)

3. As soon as the dotting tone is detected, start your symbol clock tracking loop including a variable phase interpolator, clock tone generator, and phase correction logic. All these components should operate at a 16 kHz sampling rate.
 - Remember that the clock tracking loop locks to the positive zero crossings of the generated clock tone. You can get an initial rough estimate for the correct interpolator phase by finding the position of a positive zero crossing of the clock tone and using it to set the phase. Making a good guess for the initial phase results in quicker loop lock.
4. When the 15-symbol periodic phase starts,
 - down sample the interpolator output by a factor of two, and wait for at least $30 T/2$ samples to fill the equalizer delay line.

Connected to Another DSK (cont. 3)

- Then use ideal reference training to adapt the equalizer as you did in the loop back experiments.
 - Also send the equalizer outputs four times per symbol to the left codec channel and a baud sync signal to the right channel as before.
 - Continue to send the equalizer output to the codec from here on.
5. At the end of the periodic signal phase, perform tap rotation and switch to decision directed updating as before.
 6. Plot the equalizer coefficients after they have converged.
 7. Connect the baud sync signals from the transmitter and receiver to the oscilloscope and see if they are locked in frequency or drift relative to each other. You will see some clock jitter but they should be essentially locked in frequency.

Connected to Another DSK (cont. 4)

8. Turn off the interpolator phase updating and see if the two baud sync clocks drift relative to each other.

References

1. S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, 1986.
2. S. Tretter, *Constellation Shaping, Nonlinear Precoding, and Trellis Coding for Voiceband Telephone Channel Modems*, Kluwer Academic Publishers, 2002.
3. B. Widrow and S.D. Stearns, *Adaptive Signal Processing*, Prentice-Hall, 1985.