Chapter 11
Digital Data Transmission by Baseband Pulse Amplitude Modulation (PAM)

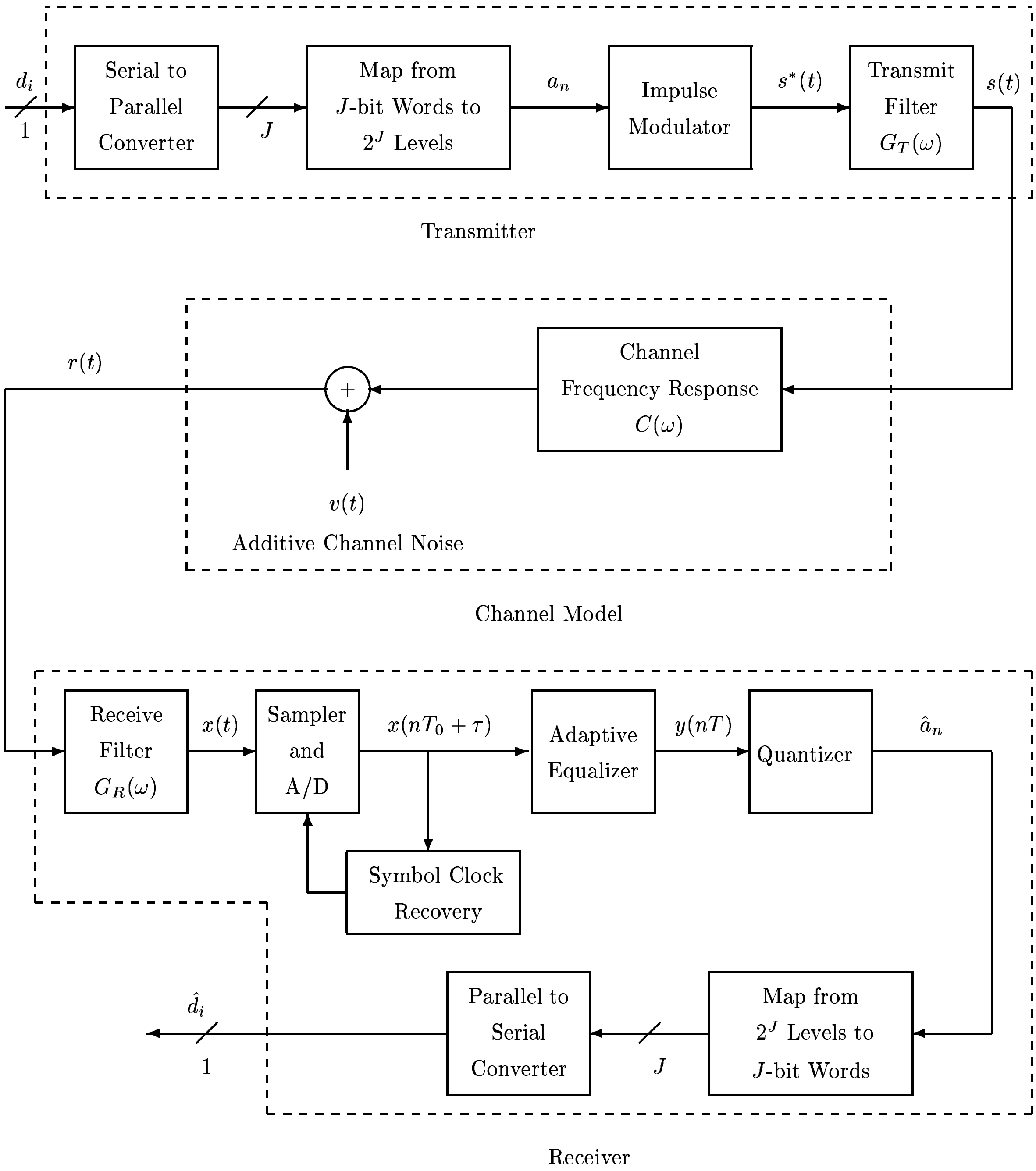
Contents

Chapter 11
Digital Data Transmission by Baseband Pulse Amplitude Modulation (PAM)

Goals

- Learn about baseband digital data transmission over bandlimited channels by PAM.
- Learn how to generate bandlimited PAM signals using *baseband shaping filters* realized by interpolation filter banks.
- Learn about *intersymbol interference* (ISI).
  - Eye diagrams to show ISI
  - The Nyquist criterion for no ISI
  - Raised cosine shaping filters for no ISI
- Derive a symbol error probability formula.
- Implement a symbol clock recovery method.

Block Diagram of a Baseband PAM System



Description of the PAM Block Diagram

- The transmitter input d_i is a serial binary data sequence with a bit rate of R_d bits/sec.
- Input bits are blocked into J-bit words by the serial-to-parallel converter.
- Input blocks are mapped into the sequence of symbols a_n which are selected from an alphabet of $M = 2^J$ distinct voltage levels. For example, the following levels uniformly spaced by $2d$ are commonly used:

  $$\ell_i = d(2i-1) \ \text{ for } \ i = -\frac{M}{2}+1,\ldots,0,\ldots,\frac{M}{2}$$

  The minimum level is $-(M-1)d$ and the maximum level is $(M-1)d$.
- The symbol rate is $f_s = 1/T = R_d/J$ symbols/sec or baud.

PAM Figure Description (cont. 1)

- The *impulse modulator* output is

$$s^*(t) = \sum_{k=-\infty}^{\infty} a_k \delta(t - kT)$$

- The bandlimiting transmit filter output is

$$s(t) = \sum_{k=-\infty}^{\infty} a_k g_T(t - kT)$$

  The combination of impulse modulator and transmit filter is a mathematical model for a D/A converter followed by a lowpass filter.

- The channel is modeled as a filter $C(\omega)$ followed by an additive noise source.

- The *receive filter* eliminates out-of-band noise and, in conjunction with the transmit filter, forms a properly shaped pulse.

PAM Figure Description (cont. 2)

The combined transmit filter, channel, and receive filter frequency response is

$$G(\omega) = G_T(\omega)C(\omega)G_R(\omega)$$

and the corresponding impulse response is

$$g(t) = g_T(t) * c(t) * g_R(t) = \mathcal{F}^{-1}\{G(\omega)\}$$

The combined filter $G(\omega)$ is called the *baseband shaping filter*.

The output of the receive filter is

$$x(t) = \sum_{k=-\infty}^{\infty} a_k g(t-kT) + v(t) * g_R(t)$$

- The output of the receive filter is sampled at a rate that is an integer multiple N of the symbol rate f_s. Typically, N might be 3 or 4. These samples are used by the Symbol Clock Recovery system to lock the receiver symbol clock to the transmitter clock.

PAM Figure Description (cont. 3)

- The Adaptive Equalizer is an FIR filter with adjustable taps that automatically compensates for channel amplitude and phase distortion. It also corrects for small deviations in the transmit and receive filter responses from their ideal nominal values. A least mean-square error (LMS) adaptation algorithm is used most often.
- The equalizer output is sampled at the symbol rate and quantized to the nearest ideal level.
- The Quantizer output is mapped to the corresponding J-bit binary word and converted back to a serial output data sequence.

Baseband Shaping and Intersymbol Interference (ISI)

An impulse response with the following property is said to have *no intersymbol interference*:

$$g(nT) = \delta[n] = \begin{cases} 1 & \text{for} \quad n = 0 \\ 0 & \text{otherwise} \end{cases}$$

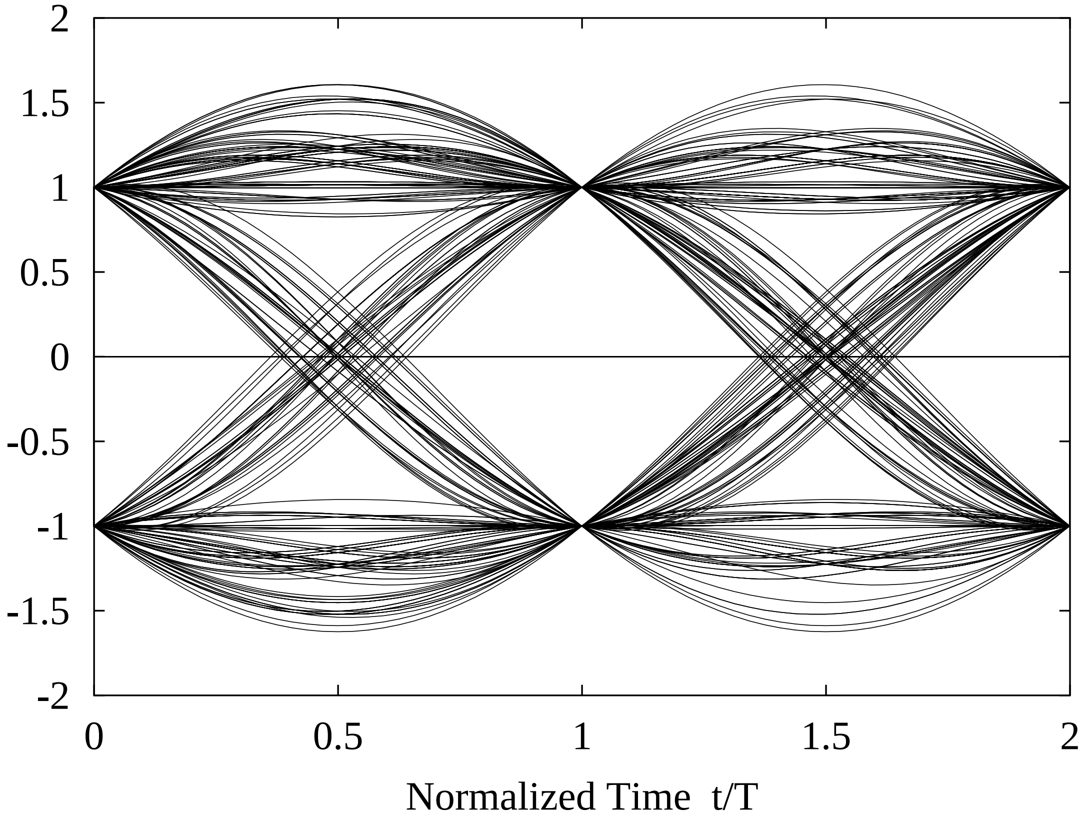
Then, if the additive noise is zero, the samples of the receive filter output at the symbol instants are

$$x(nT) = \sum_{k=-\infty}^{\infty} a_k \delta[n-k] = a_n$$

which are exactly the transmitted symbols.

The transmit and received filters are usually designed so their cascade, $G_T(\omega)G_R(\omega)$, has no ISI with, perhaps, some compromise equalization for the expected channel.

Eye Diagrams



Eye Diagram for a Raised Cosine Channel with 12% Excess Bandwidth

With zero noise, baud rate samples of the receive filter output are

$$x(nT) = \sum_{k=-\infty}^{\infty} a_k g(nT - kT)$$

$$x(nT) = g(0)\left[a_n + \sum_{\substack{k=-\infty \\ k\neq n}}^{\infty} a_k \frac{g(nT-kT)}{g(0)}\right]$$

The right-hand sum is the ISI for the received symbol. The worst case ISI occurs when the symbols a_k have their maximum magnitude $(M-1)d$ and the same sign as $g(nT-kT)$. Then

$$\begin{aligned} D &= (M-1)d \sum_{\substack{k=-\infty \\ k\neq n}}^{\infty} \left|\frac{g(nT-kT)}{g(0)}\right| \\ &= (M-1)d \sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} \left|\frac{g(kT)}{g(0)}\right| \end{aligned}$$

The peak fractional eye closure is defined to be

$$\eta = \frac{D}{d} = (M-1) \sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} \left|\frac{g(kT)}{g(0)}\right|$$

When η is less than 1, the eyes are open.

The Nyquist Criterion for No ISI

The impulse response samples are

$$g(nT) = \frac{1}{2\pi} \int_{-\infty}^{\infty} G(\omega) e^{j\omega nT} \, d\omega$$

Let $\omega_s = 2\pi f_s = 2\pi/T$. Then

$$g(nT) = \sum_{k=-\infty}^{\infty} \frac{1}{\omega_s} \int_{-\frac{\omega_s}{2}-k\omega_s}^{\frac{\omega_s}{2}-k\omega_s} \frac{1}{T} G(\omega) e^{j\omega nT} \, d\omega$$

$$= \sum_{k=-\infty}^{\infty} \frac{1}{\omega_s} \int_{-\frac{\omega_s}{2}}^{\frac{\omega_s}{2}} \frac{1}{T} G(\omega - k\omega_s) e^{j(\omega - k\omega_s)nT} \, d\omega$$

Recognizing that $e^{-jkn\omega_s T} = e^{-kn2\pi} = 1$ and taking the sum inside the integral gives

$$g(nT) = \frac{1}{\omega_s} \int_{-\frac{\omega_s}{2}}^{\frac{\omega_s}{2}} G^*(\omega) e^{j\omega nT} \, d\omega$$

Nyquist's Criterion (cont.)

where

$$G^*(\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} G(\omega - k\omega_s)$$

The function $G^*(\omega)$ is called the *aliased* or *folded* spectrum.

There is no ISI if and only if

$$G^*(\omega) = 1$$

because then the integral at the bottom of the previous slide becomes

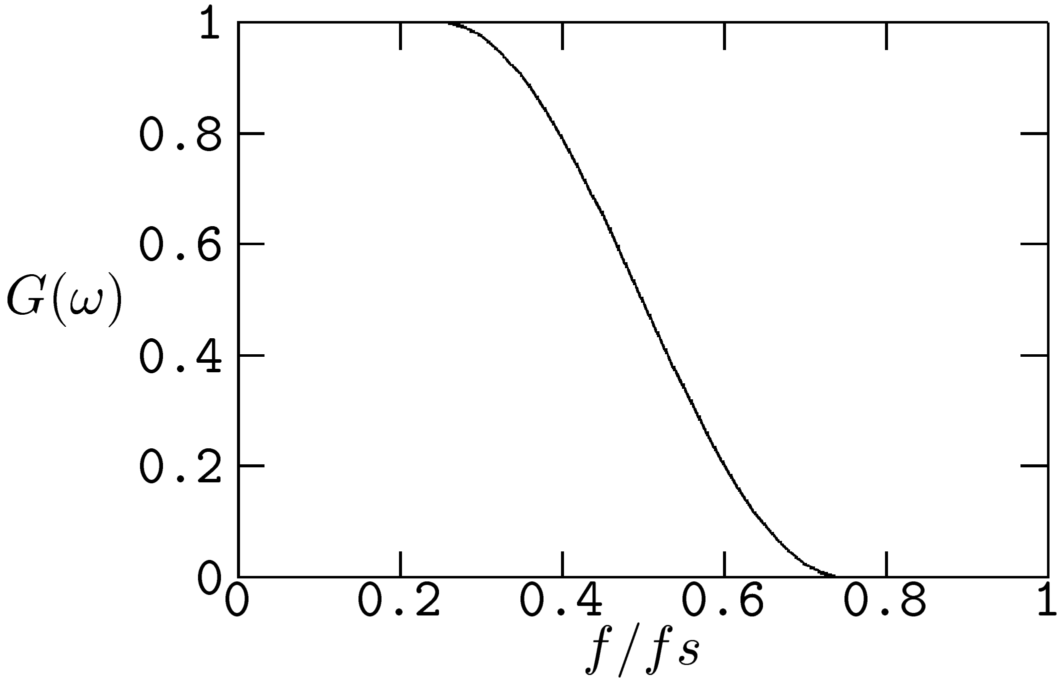
$$g(nT) = \frac{1}{\omega_s} \int_{-\frac{\omega_s}{2}}^{\frac{\omega_s}{2}} e^{j\omega nT}\, d\omega = \delta[n]$$

Raised Cosine Baseband Shaping Filters

The frequency response of a raised cosine filter is

$$G(\omega) = \begin{cases} T \quad \text{for} \quad |\omega| \le (1-\alpha)\frac{\omega_s}{2} \\ \frac{T}{2}\left\{1 - \sin\left[\frac{T}{2\alpha}\left(|\omega| - \frac{\omega_s}{2}\right)\right]\right\} \\ \qquad \text{for} \quad (1-\alpha)\frac{\omega_s}{2} \le |\omega| \le (1+\alpha)\frac{\omega_s}{2} \\ 0 \quad \text{elsewhere} \end{cases}$$

$\alpha \in [0, 1]$ is the *excess bandwidth factor*.



Example for $T = 1$ and $\alpha = 0.5$

Raised Cosine Filters (cont.)

The corresponding impulse response is

$$g(t) = \frac{\sin\left(\frac{\omega_s}{2}t\right)}{\frac{\omega_s}{2}t} \frac{\cos\left(\alpha\frac{\omega_s}{2}t\right)}{1-4(\alpha t/T)^2}$$

The program **C:\DIGFIL\RASCOS.EXE** computes samples of the impulse response modified by the Hamming window.

Special Cases

- If $\alpha = 0$, the raised cosine filter becomes an ideal flat lowpass filter with cutoff frequency $\omega_s/2$.
- If $\alpha = 1$, the frequency response has no flat region and is one cycle of a cosine function raised up so it becomes 0 at the cutoff frequency of ω_s.
- As the bandwidth is increased by making α closer to 1, the impulse response decays more rapidly.

Splitting the Shaping Between the Transmit and Receive Filters

If the channel amplitude response is flat across the signal passband and the noise is white, the amplitude response of the combined baseband shaping filter should be equally split between the transmit and receive filters to maximize the output signal-to-noise ratio, that is,

$$|G_T(\omega)| = |G_R(\omega)| = |G(\omega)|^{1/2}$$

Their phases can be arbitrary as long as the combined phase is linear.

When raised cosine shaping is used, the transmit and receive filters are called *square-root of raised cosine* filters.

C:\DIGFIL\SQRTRACO.EXE can be used to compute the impulse response of a square-root of raised cosine filter.

Implementing the Transmit Filter by an Interpolation Filter Bank

This approach places the computational burden on the DSP and allows a simple analog output lowpass filter to be used.

The transmit filter continuous-time output is

$$s(t) = \sum_{k=-\infty}^{\infty} a_k g_T(t - nT)$$

Let $t = nT + m(T/L)$ to get

$$s\left(nT + m\tfrac{T}{L}\right) = \sum_{k=-\infty}^{\infty} a_k g_T\left(nT + m\tfrac{T}{L} - kT\right)$$

$$\text{for} \quad m = 0, 1, \ldots, L-1$$

Now let L discrete-time *interpolation subfilters* be defined as

$$g_{T,m}(n) = g_T(nT + m\tfrac{T}{L}) \quad \text{for } m = 0, \ldots, L-1$$

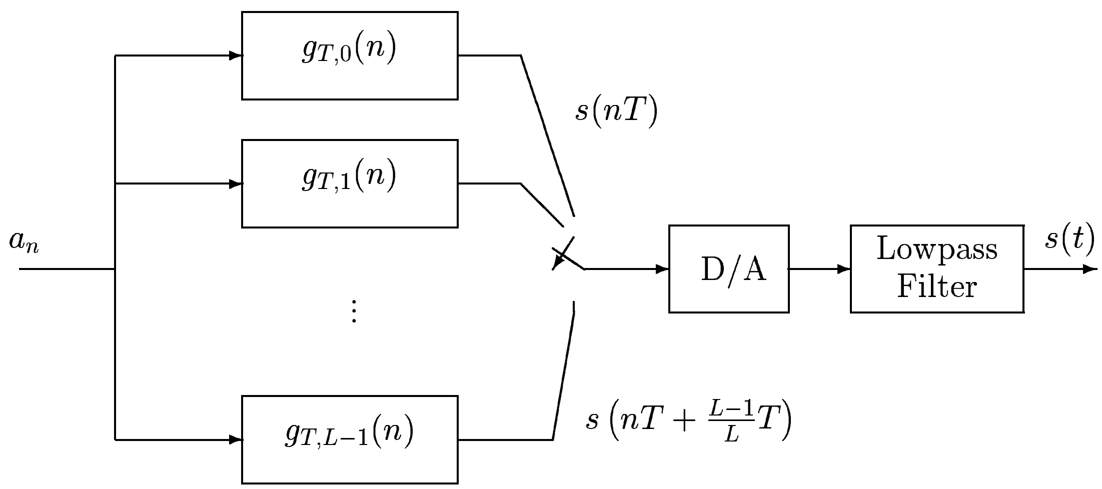
Interpolation Filters (cont.)

- Notice that the subfilters, $g_{T,m}(n)$, are FIR filters with T spaced taps.
- The L output samples for the symbol period starting at time nT are

$$s(nT + m\tfrac{T}{L}) = \sum_{k=-\infty}^{\infty} a_k g_{T,m}(n-k)$$
$$\text{for} \quad m = 0, 1, \ldots, L-1$$

- Next, the subfilter outputs are multiplexed to a D/A converter at the rate of Lf_s samples/second.
- Finally, the D/A output is passed through a simple analog lowpass filter.
- The interpolation filter bank is illustrated in the next slide. In practice, the transmit filter impulse response is truncated to a finite duration by a window function.

An Interpolation Filter Bank



Symbol Error Probability vs. SNR

Assumptions:

- The frequency response of the channel is a constant over the signal bandwidth.
- Symbols from the M level alphabet are used with equal probability.
- Symbols selected at different times are uncorrelated random variables.
- The additive noise is white and Gaussian with two-sided power spectral density $N_0/2$.

Symbol Error Probability (2)

- The combined baseband shaping filter has a raised cosine response with excess bandwidth factor α and the shaping is split equally between the transmit and receive filters. Therefore, the transmit and receive filters both have square-root of raised cosine responses.

It can be shown [Lucky, Salz, and Weldon, pp. 52-53] that the average transmitted power is

$$P_s = \frac{E\{a_n^2\}}{T} \frac{1}{2\pi} \int_{-\infty}^{\infty} |G_T(\omega)|^2 \, d\omega$$

With the square-root of raised cosine transmit filter, this reduces to

$$P_s = \frac{E\{a_n^2\}}{T}$$

Symbol Error Probability (3)

With equally likely levels, the expected squared symbol value is

$$a^2 = E\{a_n^2\} = \frac{2}{M}\sum_{k=1}^{M/2}[d(2k-1)]^2 = (M^2-1)\frac{d^2}{3}$$

and the transmitted power is

$$P_s = (M^2-1)\frac{d^2}{3T} \tag{1}$$

The noise at the output of the square-root of raised cosine receive filter has the variance

$$\sigma^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{N_0}{2}|G_R(\omega)|^2\,d\omega = \frac{N_0}{2} \tag{2}$$

Also, the channel noise power in the Nyquist band $(-\omega_s/2, \omega_s/2)$ is

$$P_N = \frac{1}{2\pi}\int_{-\omega_s/2}^{\omega_s/2}\frac{N_0}{2}\,d\omega = \frac{N_0}{2T} \tag{3}$$

Symbol Error Probability (4)

Since there is no ISI, the samples of the receive filter output have the form

$$x(nT) = a_n + v_R(nT)$$

where $v_R(nT)$ is a sample of the channel noise filtered by the receive filter. The received sample $x(nT)$ is quantized to the nearest ideal level $\hat{a}_n$.

Error Probability for Inner Points

For the $M - 2$ inner levels, an error is made if the noise magnitude exceeds d, half the distance between points. In this case, the symbol error probability is

$$P_I = P(|v_R(nT)| > d) = 2Q(d/\sigma)$$

where $Q(x)$ is the Gaussian tail probability

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}\, dt$$

Symbol Error Probability (5)

which is accurately approximated for $x > 2$ by

$$Q(x) \simeq \frac{1}{x\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

Error Probability for the Outer Points

For the level $(M-1)d$ the error probability is

$$P_{O+} = P(v_R(nT) < -d) = Q(d/\sigma)$$

For the outer level $-(M-1)d$ the error probability is

$$P_{O-} = P(v_R(nT)) > d) = Q(d/\sigma) = P_{O+}$$

The total symbol error probability is

$$\begin{aligned} P_e &= \frac{M-2}{M} P_I + \frac{1}{M} P_{O+} + \frac{1}{M} P_{O-} \\ &= 2\frac{M-1}{M} Q(d/\sigma) \end{aligned}$$

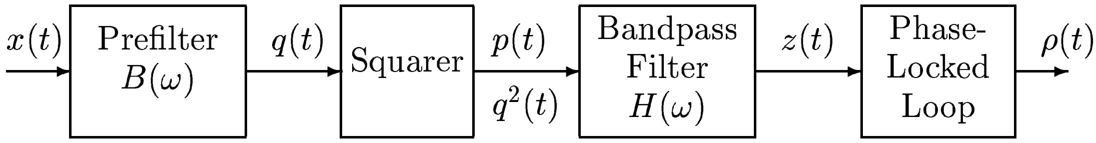
Symbol Error Probability (6)

Solving the transmitted power equation (1) for d and using formula (2) for the receive filter output noise power and formula (3) for the channel noise power, the error probability can be expressed in terms of the channel signal-to-noise ratio P_s/P_N as

$$P_e = 2\frac{M-1}{M} Q\left[\left(\frac{3}{M^2-1}\frac{P_s}{P_N}\right)^{1/2}\right]$$

Symbol Clock Recovery

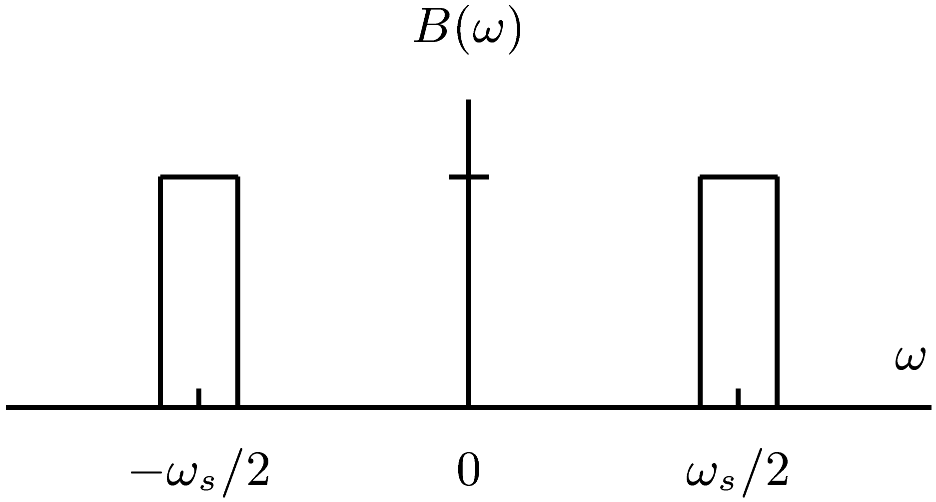
The receiver must lock its local symbol clock frequency and phase to those in the received signal. A method, good for bandlimited systems, for deriving the symbol clock from the received signal is illustrated below.



System to Generate a Symbol Clock Tone

Generating a Clock Tone (cont. 1)

- The receive filter output $x(t)$ is first passed through a prefilter with frequency response $B(\omega)$. This is a bandpass filter centered at $\omega_s/2$, half the symbol frequency.



Let the combined baseband shaping filter and prefilter frequency and impulse responses be

$$G_1(\omega) = G(\omega)B(\omega) \quad \text{and} \quad g_1(t) = g(t) * b(t)$$

Then, the prefilter output is

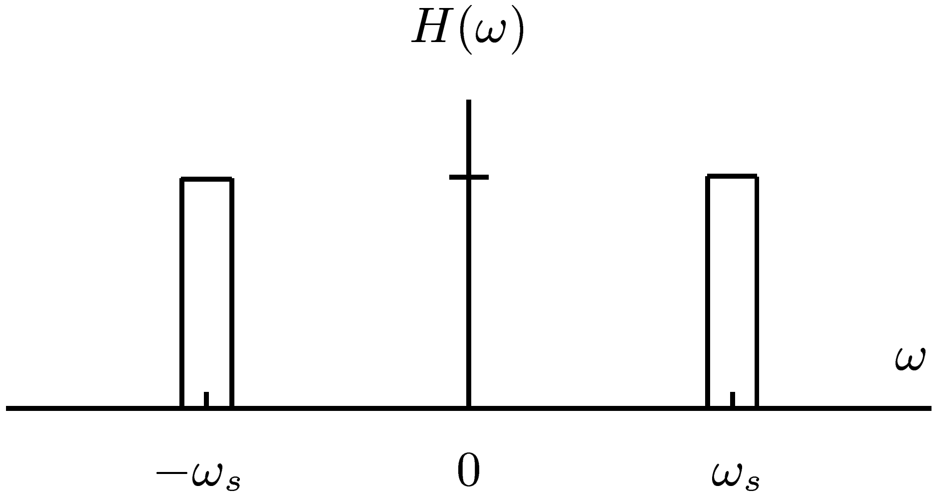
$$q(t) = \sum_{k=-\infty}^{\infty} a_k g_1(t - kT)$$

Generating a Clock Tone (cont. 2)

- The prefilter output is squared to get

$$p(t) = q^2(t)$$
$$= \sum_{k=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} a_k a_m g_1(t - kT) g_1(t - mT)$$

- $p(t)$ is passed through a bandpass filter $H(\omega)$ centered at the symbol rate ω_s.



$z(t)$ looks like a sinusoid at the clock frequency with slowly varying amplitude and phase. Its zero crossings cluster together. $z(t)$ is applied to a phase-locked loop to generate a stable symbol clock.

Generating a Clock Tone (cont. 3)

It will be assumed that the symbols are a sequence of zero-mean uncorrelated random variables. Therefore,

$$E\{a_k a_m\} = a^2 \delta_{k,m}$$

The expected value of the squarer output is

$$E\{p(t)\} = \sum_{k=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} E\{a_k a_m\} g_1(t-kT) g_1(t-mT)$$

This reduces to

$$\Lambda(t) = E\{p(t)\} = a^2 \sum_{k=-\infty}^{\infty} g_1^2(t-kT)$$

$\Lambda(t)$ is periodic with period equal to the symbol period T. It can be expressed as a Fourier series of the form

$$E\{p(t)\} = \sum_{k=-\infty}^{\infty} p_k e^{jk\omega_s t}$$

Generating a Clock Tone (cont. 4)

where

$$p_k = \frac{1}{T}\int_0^T E\{p(t)\}e^{-jk\omega_s t}\,dt$$

It can be shown that

$$\begin{aligned} p_k &= \frac{a^2}{T}\int_{-\infty}^{\infty} g_1^2(t)e^{-jk\omega_s t}\,dt \\ &= \frac{a^2}{T2\pi}\int_{-\infty}^{\infty} G_1(\omega)G_1(k\omega_s-\omega)\,d\omega \end{aligned}$$

The expected value of the output bandpass filter is also periodic with the Fourier series expansion

$$E\{z(t)\} = \sum_{k=-\infty}^{\infty} z_k e^{jk\omega_s t}$$

where

$$\begin{aligned} z_k &= p_k H(k\omega_s) \\ &= H(k\omega_s)\frac{a^2}{T2\pi}\int_{-\infty}^{\infty} G_1(\omega)G_1(k\omega_s-\omega)\,d\omega \end{aligned}$$

Generating a Clock Tone (cont. 5)

- By selecting the prefilter $B(\omega)$ to be a narrow band filter that passes components only near $\pm\omega_s/2$, it can be seen that $p_k = 0$ except for $k = -1, 0$, or 1.
- By selecting the output bandpass filter $H(\omega)$ so that it only passes spectral components near $\pm\omega_s$, the $k = 0$ term is removed and only the symbol frequency components for $k = \pm 1$ remain.
- When the baseband shaping filter has zero excess bandwidth, that is, when $G(\omega) = 0$ for $|\omega| \geq \omega_s/2$, all the Fourier coefficients p_k are zero for $k \neq 0$ since the nonzero portions of $G(\omega)$ and $G(k\omega_s - \omega)$ do not overlap. This timing recovery method then fails.

Generating a Clock Tone (cont. 6)

Condition for Good Timing Recovery

When $G_1(\omega)$ is symmetric about $\omega_s/2$ and is bandlimited to the interval

$$\omega_s/4 < |\omega| < 3\omega_s/4$$

and $H(\omega)$ is symmetric about ω_s, it can be shown that the variance of $z(t)$ is zero and perfect timing recovery is possible. When these symmetry conditions are nearly met, the variations in the zero crossings of the timing wave $z(t)$ are very small and the receiver can track the symbol clock frequency by locking to the zero crossings.

Theoretical Exercises

1. Write a C function to generate pseudo-random four-level symbols.
   - The function should use a 23-stage self synchronizing shift register sequence generator with the connection polynomial

     $$h(D) = 1 + D^{18} + D^{23} \quad \text{or} \quad 1 + D^5 + D^{23}$$

     as discussed in Chapter 9, to generate the binary sequence d_n.
   - Generate the four-level sequence a_n from pairs of binary symbols (d_{2n}, d_{2n+1}) according to the rule

     $$a_n = (-1)^{d_{2n}}(1 + 2d_{2n+1})d$$

     where d is a desired scale factor.
2. Draw a vertical axis and show the 4 levels. Label each level with its analog value and also the corresponding pair of binary digits.

Theoretical Exercises (cont. 1)

3. Design an interpolation filter bank.
   - Use the program **C:\DIGFIL\RASCOS.EXE** to generate the subfilter coefficients.
   - Use a symbol rate of $f_s = 1/T = 2.4$ kHz.
   - First, use an excess bandwidth factor of $\alpha = 1.0$.
   - Truncate the shaping filter impulse response to the interval $[-4T, 4T]$ with a Hamming window.
   - Generate L=16 samples of the PAM signal per symbol interval, that is, generate the sequence $s(kT/16)$.
4. Generate data for an eye diagram that extends over two symbol intervals.
   - Write enough pairs $(\text{mod}(k, 32), s(kT/16))$ to a file to form a reasonably filled out 4-level eye diagram.

Theoretical Exercises (cont. 2)

- The function, mod$(k, 32)$, is the remainder when k is divided by 32 and ranges from 0 to 31. As k increases, mod$(k, 32)$ cycles throught the values $0, 1, \ldots, 31$. This performs the function of resetting the trace to the left-hand side every two symbols.
- When k reaches a multiple of 32 and mod$(k, 32) = 0$, you should write the three extra points $(32, s(kT/32))$, $(32, 0)$ and $(0, 0)$ to the file before writing $(0, s(kT/32)$. These three points accomplish the following:
  - $(32, s(kT/32))$ continues the trace to the right edge of the plot.
  - $(32, 0)$ moves the trace vertically to 0.
  - $(0, 0)$ moves the trace horizontally at 0 from the right edge back to the origin.

Theoretical Exercises (cont. 3)

Otherwise, retrace lines will be drawn through the eye diagram.

5. Print the eye diagram for $\alpha = 1$.
6. Change α to 0.125 and generate a new eye diagram. Print the new diagram. Discuss differences in the two eye diagrams and comment on how the excess bandwidth factor affects the required symbol sampling time accuracy in the receiver.
7. Eye Diagram Using Square-Root of Raised Cosine Shaping

   Repeat the raised cosine exercises for a square-root of raised cosine baseband shaping filter, but only for $\alpha = 0.125$. Also, compute the peak fractional eye closure defined on Slide 11-9 from the shaping filter impulse response.

8. **Theoretical Error Probability**

The error probability can be expressed in terms of the channel signal-to-noise ratio P_s/P_N as

$$P_e = 2\frac{M-1}{M} Q\left[\left(\frac{3}{M^2-1}\frac{P_s}{P_N}\right)^{1/2}\right]$$

under the following conditions:

- A flat channel frequency response.
- Symbols from the M level alphabet are used with equal probability.
- Symbols selected at different times are uncorrelated random variables.
- The noise is white and Gaussian with power spectrum $N_0/2$.
- The transmit and receive filters have square-root of raised cosine responses.

Plot P_e for $M = 2, 4$, and 8 as a function of the channel signal-to-noise ratio P_s/P_N. Plot P_e on a logarithmic scale and $10\log_{10}(P_s/P_N)$ dB on a linear scale.

Making a PAM Signal with the EVM

- Generate a four-level PAM signal using a raised cosine baseband shaping filter with $\alpha = 0.125$.
- Truncate the shaping filter impulse response to the interval $[-4T, 4T]$ by a Hamming window.
- Use symbol rate $f_s = 2.4$ kHz. Generate $L = 4$ PAM signal samples per symbol with an interpolation filter bank. The D/A sampling rate should be set to $2.4 \times 4 = 9.6$ kHz.
- Use the same 23-stage scrambler you used for the theortical exercises.
- Write the output samples to the left channel.
- You may write your program entirely in C or in combined C and assembly. Use -o3 optimization.

Frequency Response of the Shaping Filter

Let the impulse response of the baseband shaping filter, viewed as an FIR filter with $T/4$ tap spacing, be

$$g_n = \begin{cases} g(nT/4) & \text{for } \; n = -16, -15, \ldots, 16 \\ 0 & \text{elsewhere} \end{cases}$$

The frequency response of this filter is

$$\begin{aligned} G(\omega) &= \sum_{n=-16}^{16} g_n e^{-j\omega nT/4} \\ &= g_0 + 2\sum_{n=1}^{16} g_n \cos(\omega nT/4) \end{aligned}$$

- Compute and plot the amplitude response of your filter in dB over the frequency range of 0 to 4.8 kHz.

Using a Mailbox to Store and Output Samples

There are a variety of ways to structure the program. Here is one approach to try.

- Write output samples to the McBSP DXR with an interrupt routine triggered by the serial port transmit interrupts (XINT). Write the samples to the left channel.
- Determine the symbol timing by counting interrupts modulo 4.
- Set up an 8-word circular buffer as a "mail box." One half of the buffer (4 words) will be used to hold the output samples for the current symbol period, and the remaining half will be used to store the four samples for the next symbol period. Each symbol period, the input and output halves will be swapped. These are sometimes called *ping pong buffers*.

Output							
0	1	2	3	0	1	2	3
				Input			

Using a Mailbox (cont.)

- Initialize the output pointer to the address of the first word in the buffer and the input pointer to fifth word.
- Before starting data transmission, set the interrupt count to 0 to indicate the start of a symbol.
- At the start of each symbol period, generate four output samples, and write them to the mailbox. Do this in the main routine.
- The input pointer should be incremented circularly after each sample is written to the mailbox.
- After the four samples are written to the mailbox, the main routine should wait for the interrupt count to become 0. Remember, the interrupt routine should count interrupts modulo 4.

- The transmit interrupt service routine should write the sample addressed by the output pointer to the DXR, increment the output pointer circularly modulo 8, and increment the interrupt count modulo 4.

Generating a Baud Synch Signal

- Test your program by observing the eye diagram on the oscilloscope. Use DC coupling for the scope probe.

- You will need a signal to synchronize the sweeps with the symbol period to get a display like your theoretical plot. One way to generate a synch signal is to run the codec in stereo mode and create a 2400 Hz square-wave on the right channel codec output. You can do this by putting an integer like $A = 16000$ in the lower half of the word sent to the codec for first two samples in a baud and $-A$ in the second two samples.

Making a Clock Tone Generator

Write a program for the DSP to implement the symbol clock recovery system discussed in Slides 11-22 – 11-28 to the point labelled $z(t)$ on Slide 11-22. Do not implement the phase-locked loop.

- Use the same raised cosine baseband shaping filter you designed for the C6x PAM signal generator.
- The sampling rate for all operations in the symbol clock tone generator should be $4f_s = 9.6$ kHz.
- Write the PAM output samples to the left channel D/A converter. Connect the left channel line output to the left channel line input. Send the input samples to your clock tone generation function.
- For the prefilter, $B(\omega)$, design a second-order IIR filter with a center frequency of $f_s/2 = 1.2$ kHz and roughly a 100 Hz 3 dB bandwidth.

Making a Tone Generator (cont.)

- For the postfilter, $H(\omega)$, design a second-order bandpass IIR filter with a center frequency of 2.4 kHz and a 3 dB bandwidth of roughly 25 Hz. You should experiment with these bandwidths and observe how they affect the system performance.
- Write the tone generator system output samples $z(nT/4)$ to the right channel output.

Testing the Clock Tone Generator

1. First drive your baseband shaping filter with the alternating two-level symbol sequence $a_n = (-1)^n d$.
   - This is called a *dotting* sequence in the modem jargon.
   - Send the shaping filter output samples to the left channel output and the clock tone generator output samples to the right channel output.

Testing the Tone Generator (cont. 1)

Observe the left and right channel outputs simultaneously on the oscilloscope.

- Notice that $a_n = (-1)^n = \cos\left(\frac{\omega_s}{2} nT\right)$ which are symbol rate samples of a cosine wave that has a frequency of half the symbol rate.

  The sampled signal has spectral components at the set of frequencies

  $$\{\omega_s/2 + k\omega_s; k = -\infty, \ldots, \infty\}$$

  The shaping filter will pass the 1.2 kHz component and heavily attenuate the other components in the $[0, 4.8)$ kHz band. In other words, the PAM signal should be very close to a 1.2 kHz sine wave.

Testing the Tone Generator (cont. 2)

- Check that the tone generator output is a 2.4 kHz sinewave locked to the PAM signal.

2. Next, use a two-level pseudo-random symbol sequence having values $\pm d$. Use the shift register generator to select the levels. Observe the PAM signal and clock tone generator output on the oscilloscope. They should still be locked together. Comment on how the tone generator output looks compared to the output with the dotting sequence.
3. Finally, use the full four-level pseudo-random input symbol sequence. Observe the output of the clock recovery system on the oscilloscope and compare it with the previous cases.

Optional Team Exercise

If you are interested in doing more with PAM, team up with an adjacent group. Make one setup a PAM transmitter and the other a PAM receiver.

- Transmit a two-level PAM signal. The levels should be selected by the output of the 23-stage scrambler with an input of 0. Use a 2.4 kHz symbol rate.
- Sample the received signal at 9.6 kHz.
- Even though the transmitter and receiver both use a sampling frequency of 9.6 kHz, there will be slight differences due to small physical and temperature differences in the oscillator crystals and circuit components.

  You will have to devise a method for synchronizing the symbol clock in the receiver to the symbol clock in the transmitter.

Optional Team Exercise (cont.)

The sampling phase of the codec cannot be altered, so you will have to pass the received samples through a variable phase interpolator that compensates for the phase difference between the transmit and receive clocks. Variable phase interpolators are discussed in the next chapter. You can lock the phase of the receiver symbol clock to the positive zero crossings of the symbol clock tone generator. In addition you will have to compensate for any delays in the system so that samples are taken at the symbol instants, that is, at the point where the eye has its maximum opening.

- Quantize the selected symbol rate samples to a binary sequence. Descramble this sequence and check that the output is all 0's.
- Add Gaussian noise to the received samples in the DSP and make a plot of the bit-error rate vs. SNR.