Integrated Circuits for Digital Communications

Prof. David Johns
University of Toronto

(johns@eecg.toronto.edu)
(www.eecg.toronto.edu/~johns)

Motivation

• Many exciting new applications for digital comm (previously, modems and satellites)
• xDSL (x=H, A, or V), ethernet, ATM, cable-modems, firewire, disk-drives, wireless, infrared, etc.
• Want to push up bit rate using same channels

Mainly Digital

• Tolerant to large noise and channel variations
• Tolerant to analog IC freq and dc variations
• Easier to design and test

Mainly Analog

• Smaller size and power at high-speeds
Basic Challenge

- Channel attenuates and phase changes signal, also noise added

Channel Amplitude and Phase Variations

- Amplitude variation (usually lowpass)
- Phase variations (no longer linear phase)
- Linear phase keeps symmetric pulses (intuitively better for less interference)
Channel Noise

- Noise and bandwidth limits how much digital information can be sent through channel
- Usually model noise as being added after channel impairment
- Can use filtered noise if not white
- Usually assume Gaussian noise because of central limit theorem and mathematically tractable
- Note that receive equalizer might increase noise if it has boost (noise enhancement)

Low Noise - Large Bandwidth

- Not pushing maximum bits through channel
- Low noise — can use more levels for each symbol
- Large bandwidth — can increase symbol-rate
**Multi-Level — Low-Noise, Large Bandwidth**

- Twice the bit information over same bandwidth!
- More prone to noise causing errors
- Commonly called PAM (here 2B1Q — 4-PAM)

**High Noise - Low Bandwidth**

- Equalizing back to “square-wave” amplifies noise
- Particularly bad if most noise is at high frequencies
- Too many errors — need to equalize without much noise enhancement (or add coding)
- More errors with multi-level PAM (unless coded)
Noise Enhancement Example

Before Equalization

Signal power = 1.01 W
Noise power = 0.02 W
SNR = 10\log\left(\frac{1.01}{0.02}\right) = 17 \text{ dB}

After Equalization

Signal power = 2 W
Noise power = 1.01 W
SNR = 10\log\left(\frac{2}{1.01}\right) = 3 \text{ dB}
Basic Baseband PAM Concepts

General Data Communication System

- Source coder removes redundancy from source (i.e. MPEG, ADPCM, text compression, etc.)
- Channel coder introduces redundancy to maximize information rate over channel. (i.e. error-correcting codes, trellis coding, etc.)
- Our interest is in channel coding/decoding and channel transmission/reception.
• In 2B1Q, coder maps pairs of bits to one of four levels
  \[ A_k = \{-3, -1, 1, 3\} \]

Rectangular Transmit Filter

• The spectrum of \( A_k \) is flat if random.
• The spectrum of \( s(t) \) is same shape as \( H_t(f) \)
Nyquist Pulses

- $h(t)$ is the impulse response for transmit filter, channel and receive filter ($\otimes$ denotes convolution)
  
  $$h(t) = h_i(t) \otimes h_c(t) \otimes h_r(t)$$  

  \hspace{1cm} (1)

  $$q(t) = \sum_{m = -\infty}^{\infty} A_m h(t - mT) + n(t) \otimes h_r(t)$$  

  \hspace{1cm} (2)

  - The received signal, $q(t)$, is sampled at $kT$.
  
  $$q_k = \sum_{m = -\infty}^{\infty} A_m h(kT - mT) + u(kT), \ u(t) \equiv n(t) \otimes h_r(t)$$  

  \hspace{1cm} (3)

  - To have zero intersymbol interference (i.e. $q_k = A_k + u_k$)
    
    $$h(kT) = \delta_k \quad (\delta_k = 0, 1, 0, 0, …)$$  

  \hspace{1cm} (4)

Nyquist Pulses

- For zero ISI, the same criteria in the frequency domain is: ($f_s = 1/T$)
  
  $$\frac{1}{T} \sum_{m = -\infty}^{\infty} H(j2\pi f + jm2\pi f_s) = 1$$  

  \hspace{1cm} (5)

  - Known as Nyquist Criterion

Example Nyquist Pulses (in freq domain)

- Sinc pulse
- Raised-cosine pulse
Nyquist Pulses

Sinc pulse

Raised-cosine pulse

Raised-Cosine Pulse

\[
H(j2\pi f) = \begin{cases}
T; & 0 \leq |f| \leq (1 - \alpha)\left(\frac{f_s}{2}\right) \\
\frac{T}{2}\left[1 + \cos\left(\frac{\pi}{2\alpha}\left(\frac{f_s}{f_s} - (1 - \alpha)\right)\right)\right]; & (1 - \alpha)\left(\frac{f_s}{2}\right) \leq |f| \leq (1 + \alpha)\left(\frac{f_s}{2}\right) \\
0; & |f| > (1 + \alpha)\left(\frac{f_s}{2}\right)
\end{cases}
\]

- \(\alpha\) determines excess bandwidth
Raised-Cosine Pulses

- More excess bandwidth — impulse decays faster.

Raised-Cosine Pulse

- $\alpha$ determines amount of excess bandwidth past $f_s/2$

- Example: $\alpha = 0.25$ implies that bandwidth is 25 percent higher than $f_s/2$ while $\alpha = 1$ implies bandwidth extends up to $f_s$.

- Larger excess bandwidth — easier receiver

- Less excess bandwidth — more efficient channel use

Example

- Max symbol-rate if a 50% excess bandwidth is used and bandwidth is limited to 10kHz

- $1.5 \times (f_s/2) = 10$ kHz implies $f_s = 13.333 \times 10^3$ symbols/s
Eye Diagram

- "a" indicates immunity to noise
- "b" indicates immunity to errors in timing phase
- slope "c" indicates sensitivity to jitter in timing phase

Eye Diagram

- Zero crossing — NOT a good performance indicator
- 100% bandwidth has little zero crossing jitter
- 50% BW has a lot of zero crossing jitter but it is using less bandwidth

- Less excess BW — more intolerant to timing phase
Example Eye Diagrams

- $\alpha = 0$, 0\% excess bandwidth
- $\alpha = 0.25$, 25\% excess bandwidth
- $\alpha = 0.5$, 50\% excess bandwidth
- $\alpha = 1$, 100\% excess bandwidth

Example Eye Diagrams

- $\alpha = 0$, 0\% excess bandwidth
- $\alpha = 1$, 100\% excess bandwidth
- $\alpha = 0.5$, 50\% excess bandwidth
- $\alpha = 1$, 100\% excess bandwidth
**Matched-Filter**

- For zero-ISI, $h_{tc}(t) \ast h_r(t)$ satisfies Nyquist criterion.
- For optimum noise performance, $h_r(t)$ should be a matched-filter.
- A matched-filter has an impulse response which is time-reversed of $h_{tc}(t)$

$$h_r(t) = \kappa h_{tc}(-t)$$  \hspace{1cm} (6)

where $\kappa$ is an arbitrary constant.

---

**Matched-Filter (proof)**

- Consider isolated pulse case (so no worry about ISI)

$$r(t) = A_0 h_{tc}(t) + n(t)$$ \hspace{1cm} (7)

$$q_0 = \int_{-\infty}^{\infty} r(\tau) h_r(t-\tau) d\tau \bigg|_{t=0} = \int_{-\infty}^{\infty} r(\tau) h_r(-\tau) d\tau$$ \hspace{1cm} (8)

$$q_0 = A_0 \int_{-\infty}^{\infty} h_{tc}(\tau) h_r(-\tau) d\tau + \int_{-\infty}^{\infty} n(\tau) h_r(-\tau) d\tau$$ \hspace{1cm} (9)

- Want to maximize signal term to noise term
- Variance of noise is

$$\sigma_n^2 = N_0 \int_{-\infty}^{\infty} h_r^2(-\tau) d\tau$$ \hspace{1cm} (10)
**Matched-Filter (proof)**

- Assuming $A_0$ and $h_{tc}(t)$ fixed, want to maximize

\[
\text{SNR} = \frac{A_0^2 \int_{-\infty}^{\infty} h_{tc}(\tau)h_r(-\tau) d\tau}{N_0^2 \int_{-\infty}^{\infty} h_r^2(-\tau) d\tau} \tag{11}
\]

- Use Schwarz inequality

\[
\left[ \int_a^b f_1(x)f_2(x) dx \right]^2 \leq \left[ \int_a^b f_1^2(x) dx \right] \left[ \int_a^b f_2^2(x) dx \right] \tag{12}
\]

with equality if and only if $f_2(x) = K f_1(x)$

- Maximizing (11) results in $h_r(t) = K h_{tc}(-t)$ — QED

**Matched-Filter — Why optimum?**

- Transmit filter, channel and noise
- Too much noise, All of signal
- Too little signal, Less noise
- Just right — max SNR
ISI and Noise

- In general, we need the output of a **matched filter** to obey Nyquist criterion.
- Frequency response at output of matched filter is $|H_m(j\omega)|^2$ leading to criterion

$$\frac{1}{T} \sum_{m=\pm\infty} \left|H_m(j2\pi f + jm2\pi f_s)\right|^2 = 1$$

(13)

Example

- Assume a flat freq resp channel and raised-cosine pulse is desired at matched-filter output.
- Transmit filter should be $\sqrt{\text{raised-cosine}}$.
- Receive filter should be $\sqrt{\text{raised-cosine}}$.

---

**Gaussian Noise and SNR Requirement**
Probability Distribution Function

• Consider a random variable \( X \)
• Cumulative distribution function (c.d.f.) — \( F_x(x) \)

\[
F_x(x) \equiv P_r(X \leq x) - \infty < x < \infty
\]

\[
1 \geq F_x(x) \geq 0
\]

Example

• Consider a fair die

\[
F_x(x) = \begin{cases} 
0 & \text{if } x < 1 \\
\frac{1}{6} & \text{if } 1 \leq x < 2 \\
\frac{2}{6} & \text{if } 2 \leq x < 3 \\
\frac{3}{6} & \text{if } 3 \leq x < 4 \\
\frac{4}{6} & \text{if } 4 \leq x < 5 \\
\frac{5}{6} & \text{if } 5 \leq x < 6 \\
1 & \text{if } 6 \leq x 
\end{cases}
\]

Probability Density Function

• Derivative of \( F_x(x) \) is p.d.f. defined as \( f_x(x) \)

\[
f_x(x) \equiv \frac{dF_x(x)}{dx} \quad \text{or} \quad F_x(x) = \int_{-\infty}^{x} f_x(\alpha) d\alpha
\]

• To find prob that \( x \) is between \( x_1 \) and \( x_2 \)

\[
P_r(x_1 < X \leq x_2) = \int_{x_1}^{x_2} f_x(\alpha) d\alpha
\]

• It is the area under p.d.f. curve.

Example (fair die)

\[
f_x(x) = \begin{cases} 
\frac{1}{6} & \text{if } x = 1, 2, 3, 4, 5, 6 \\
0 & \text{otherwise}
\end{cases}
\]
Uniform Distribution

- p.d.f. is a constant
- Variance is given by: \( \sigma^2 = \frac{\Delta^2}{12} \) where \( \Delta \) is range of random variables

\[ f_x(x) = \frac{1}{\Delta} \]

\[ \sigma^2 = \frac{\Delta^2}{12} \]

- Crest factor: \( CF = \frac{\max}{\sigma} = \frac{\Delta/2}{\Delta/\sqrt{12}} = \sqrt{3} = 1.732 \)

Example

- A uniform random variable chosen between 0 and 1 has a mean, \( \mu = 0.5 \), and variance, \( \sigma^2 = 1/12 \)

Gaussian Random Variables

Probability Density Function

- Assuming \( \sigma^2 = 1 \) (i.e. variance is unity) and \( \mu = 0 \) (i.e. mean is zero) then

\[ f_x(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \] (18)
Gaussian Random Variables

- Often interested in how likely a random variable will be in tail of a Gaussian distribution

\[ Q(x) \equiv P_r(X > x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\alpha^2/2} d\alpha \]  

(19)

\[ Q(x) = \frac{1}{2} \text{erfc}(x/\sqrt{2}) \]  

(20)

Gaussian Random Variables

- Probability of \( x \) being in tail of Gaussian distribution

\[ Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\alpha^2/2} d\alpha \]

\[ \sigma^2 = 1 \]
\[ \mu = 0 \]

- If \( \sigma^2 \neq 1 \) or \( \mu \neq 0 \)

\[ P_r(X > x) = Q((x - \mu)/\sigma) \]  

(21)
Example SNR Calculation

- 100Base-T2 for fast-ethernet uses 5-PAM
- Want to calculate the receive SNR needed for a symbol-error-rate of $10^{-10}$ (assume rest is ideal).

\[
\begin{array}{cccccc}
1/8 & 1/4 & 1/4 & 1/4 & 1/8 \rightarrow \text{probability of send} \\
-4 & -2 & 0 & 2 & 4
\end{array}
\]

- Signal power, $P_s$

\[
P_s = \frac{1}{4} \times 0W + \frac{1}{2} \times 4W + \frac{1}{4} \times 16W = 6W
\] (22)

- Using a reference of 1W as 0dB,

\[
P_s = 10\log_{10}(6) = 7.78\text{dB}
\] (23)

Example SNR Calculation

- Assume Gaussian noise added to receive signal.
- Since symbols are distance 2 apart, a noise value greater than 1 will cause an error in receive symbol.
- Want to find $\sigma$ of Gaussian distribution such that likelihood of random variable greater than 1 is $10^{-10}$.
- Recall

\[
Q(x/\sigma) = 0.5\text{erfc}((x/\sigma)/\sqrt{2})
\] (24)

- Let $x = 1$ and set

\[
2Q(1/\sigma) = 10^{-10}
\] (25)

(2 value because variable might be $>1$ or $<-1$)

\[
0.5 \times 10^{-10} = Q(1/\sigma) = 0.5\text{erfc}(1/(\sigma\sqrt{2}))
\] (26)
Example SNR Calculation

- Trial and error gives $1/(\sigma \sqrt{2}) = 4.57$ implying that
  $\sigma = 0.1547 = 1/6.46$
- Noise with $\sigma = 0.1547$ has a power of (ref to $1W$)
  $$P_n = 10\log_{10}(\sigma^2) = -16.2\text{dB}$$  \hspace{1cm} (27)
- Finally, SNR needed at receive signal is
  $$\text{SNR} = 7.78\text{dB} - (-16.2\text{dB}) = 24\text{dB}$$ \hspace{1cm} (28)
- Does not account that large positive noise on $+4$ signal will \textbf{not} cause symbol error (same on $-4$).
- It is slightly conservative
- BER approx same as symbol error rate if Gray coded

m-PAM

- For $m$ bits/symbol $\Rightarrow$ $2^m$ levels
- Normalize distance between levels to 2 (so error of 1 causes a symbol error)
  - $(m = 1) \Rightarrow \pm 1$  \hspace{.5cm} $(m = 3) \Rightarrow \pm 1, \pm 3, \pm 5, \pm 7$ etc.
- Noise variance of $(\sigma = 0.1547) \Rightarrow \text{BER} = 10^{-10}$
- Symbols spaced $\pm 1, \pm 3, \pm 5, \ldots, \pm (2^m - 1)$
  — average power is: $S_m = (4^m - 1)/3$
  $$\text{SNR} = 10\log\left(\frac{S_m}{\sigma^2}\right) = 10\log\left(\frac{4^m - 1}{3\sigma^2}\right)$$ \hspace{1cm} (29)
m-PAM

\[
\text{SNR} = 10\log\left(\frac{S_m}{\sigma^2}\right) = 10\log\left(\frac{4^m - 1}{3\sigma^2}\right)
\]  

(30)

• equals 23.1 dB for \( m = 2 \), \( \text{BER} = 10^{-10} \)
• equals 28.2 dB for \( m = 3 \), \( \text{BER} = 10^{-10} \) (approx +6dB)
• Can show \( S_{m+1} = 4S_m + 1 \)
• Require 4 times more power to maintain same symbol error rate with same noise power (uncoded)
• In other words, — to send 1 more bit/symbol, need 6dB more SNR (but does not increase bandwidth)

Why Assume Gaussian Noise?

**Central-Limit Theorem**

• Justification for modelling many random signals as having a Gaussian distribution

  Sum of independent random variables approaches Gaussian as sum increases

• Assumes random variables have identical distributions.
• No restrictions on original distribution (except finite mean and variance).
• Sum of Gaussian random variables is also Gaussian.
Central Limit Theorem Example

- Consider averaging 10 uniformly random variables.

<table>
<thead>
<tr>
<th>Sample</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>14</td>
<td>68</td>
<td>42</td>
<td>-22</td>
<td>-18</td>
<td>26</td>
<td>80</td>
<td>-27</td>
<td>-57</td>
<td>-8</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>-68</td>
<td>88</td>
<td>0</td>
<td>-72</td>
<td>-75</td>
<td>-15</td>
<td>-49</td>
<td>36</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>-93</td>
<td>-57</td>
<td>-52</td>
<td>-70</td>
<td>13</td>
<td>30</td>
<td>-72</td>
<td>-73</td>
<td>82</td>
<td>27</td>
</tr>
<tr>
<td>Uniform Distribution between -100, 100</td>
<td>7</td>
<td>43</td>
<td>-64</td>
<td>17</td>
<td>-50</td>
<td>24</td>
<td>89</td>
<td>57</td>
<td>-50</td>
<td>-12</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>-74</td>
<td>-36</td>
<td>69</td>
<td>-2</td>
<td>61</td>
<td>-18</td>
<td>-9</td>
<td>72</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>91</td>
<td>-82</td>
<td>77</td>
<td>18</td>
<td>-7</td>
<td>-50</td>
<td>-74</td>
<td>-30</td>
<td>-6</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>-45</td>
<td>30</td>
<td>91</td>
<td>92</td>
<td>-5</td>
<td>77</td>
<td>-10</td>
<td>1</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>-99</td>
<td>-70</td>
<td>11</td>
<td>-75</td>
<td>-22</td>
<td>-82</td>
<td>62</td>
<td>20</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>78</td>
<td>-17</td>
<td>36</td>
<td>-70</td>
<td>-60</td>
<td>-59</td>
<td>-68</td>
<td>86</td>
<td>64</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>-95</td>
<td>-23</td>
<td>97</td>
<td>-36</td>
<td>-94</td>
<td>-86</td>
<td>30</td>
<td>51</td>
<td>70</td>
</tr>
<tr>
<td>Average</td>
<td>24</td>
<td>-43</td>
<td>3</td>
<td>14</td>
<td>-22</td>
<td>-16</td>
<td>-17</td>
<td>4</td>
<td>21</td>
<td>50</td>
</tr>
</tbody>
</table>

Uniform and Gaussian Signals

1000 samples of uniform random variables

1000 samples of Gaussian random variables

\[ \sigma^2 = 1 \]
Filtered Random Signals

Filtered with 3'rd order Butterworth lowpass with cutoff $f_s/200$

No longer independent from sample to sample

Channel Capacity
**Limits of Communications**

- Want to find max bits/sec for a given channel with arbitrary noise spectrum.

  \[ H_N(s) \]

  White Gaussian noise

  \[ H_C(s) \]

  Input Signal

  Channel

  max bits/sec

- Limited input signal power
- First find capacity for flat channels with white noise and then integrate to find overall capacity

---

**Channel Capacity of a Bandlimited Channel**

- Channel with bandwidth \( B \) (baseband or passband).

  \[ A(f) \]

  Input signal

  \(-B\) \( \quad \) \( B\) \( \quad \) \( 2B\)

  Receive signal

- Highest rate is \( 2B \) samples/s to maintain independent noise samples and independent signal samples.
- Receive signal is \( 2B \) symbols/s
- Thus, the capacity in bits per second is given by
  \[ C = 2BC_s \]  
  \[ (31) \]

  where \( C_s \) is the capacity in bits/symbol
Capacity, $C_s$ — bits/symbol

- $C_s$ is the max number of bits/symbol that can be transmitted through a channel
- In our case, each symbol has additive independent noise samples
- Need to define entropy and mutual information

- **Entropy**
  - Amount of information required on average to describe a random variable.
  - Also called average information

- **Mutual Information**
  - Measure of average information passed through a channel

Entropy (Average Information)

- Information in rare event higher than common event.
- Consider a discrete-valued random process $\{X_k\}$ with outcomes $\{a_1, a_2, \ldots, a_K\}$
- Entropy of $X$ is defined to be
  
  \[ H(X) = E[-\log_2(p_x(x))] = -\sum p_x(x)\log_2(p_x(x)) \]  
  \[ = 32 \]  

- Because of $\log_2$ — information measured in “bits”
- Note that actual values are unimportant — only probabilities.
- **Source coding theorem** — if independent trials of $X$ occur at $r$ trials/s then can encode source by a bit stream of rate $rH(X) + \varepsilon$ for any $\varepsilon > 0$. 


Entropy Example 1

• Consider a binary random variable with alphabet \{2, 7\}. If \( q = p_x(7) \) then

\[
H(X) = -q \log_2(q) - (1 - q) \log_2(1 - q)
\]

(33)

• Peaks at 1 bit and no information when \( q = 0 \) or \( q = 1 \).

• When \( q = 0.5 \), source coder is to just transmit 0 for 2 and 1 for 7.

\[
H(X) = 0.1 \log_2(0.1) - 0.9 \log_2(0.9) = 0.47
\]

(34)

• Tough to achieve optimum but one coding scheme is

<table>
<thead>
<tr>
<th>2 trials</th>
<th>bits</th>
<th>likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,2</td>
<td>0</td>
<td>0.81</td>
</tr>
<tr>
<td>2,7</td>
<td>10</td>
<td>0.09</td>
</tr>
<tr>
<td>7,2</td>
<td>110</td>
<td>0.09</td>
</tr>
<tr>
<td>7,7</td>
<td>111</td>
<td>0.01</td>
</tr>
</tbody>
</table>

• Average number of bits is \( \text{Rate} = 0.645 \text{ bits/trial} \)

\[
\text{Rate} = 0.5 \times (0.81 \times 1 + 0.09 \times 2 + 0.09 \times 3 + 0.01 \times 3)
\]

(35)

• 0.5 because 2 trials are determined at once.
**Entropy Example 2**

- Consider a 8 horse race with each horse having a probability of winning as:

\[
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{16} & \frac{1}{64} & \frac{1}{64} & \frac{1}{64} & \frac{1}{64}
\end{bmatrix}
\] (36)

\[
H(X) = -\frac{1}{2}\log_2(2^{-1}) - \frac{1}{4}\log_2(2^{-2}) - \frac{1}{8}\log_2(2^{-3}) - \frac{1}{16}\log_2(2^{-4}) - \frac{4}{16}\log_2(2^{-6})
\]

\[
H(X) = \frac{1}{2} + \frac{1}{2} + \frac{3}{8} + \frac{4}{16} + \frac{24}{64} = 2 \text{ bits}
\] (37)

- Want to send info of winning horse to bookie
- Code as: [0, 10, 110, 1110, 111100, 111101, 111110, 111111]
- Average bit length is:

\[
\frac{1}{2} \times 1 + \frac{1}{4} \times 2 + \frac{1}{8} \times 3 + \frac{1}{16} \times 4 + 4 \times \frac{1}{64} \times 6 = 2 \text{ bits}
\] (38)

**Min and Max Entropy**

- \(0 \leq H(X)\)

**Discrete Valued**

- If \(K\) is size of alphabet, then

\[
0 \leq H(X) \leq \log_2(K)
\] (39)

- If and only if all outcomes equally likely,

\[
H(X) = \log_2(K)
\] (40)

**Example:**

- If eight horses running all have an equal chance of winning, 3 bits needed to send info to bookie.
Differential Entropy

• All continuous-valued random variables have infinite entropy (try sending $\pi$ to someone)
• Use differential entropy for cont-valued variables.
• It is a measure on how fast the entropy goes to $\infty$

Min and Max

• If $Y$ has zero mean and variance $\sigma^2$
  \[ 0 \leq h(Y) \leq 0.5 \log_2(2\pi e \sigma^2) \]  (41)
• If and only if $Y$ is Gaussian, then
  \[ h(Y) = 0.5 \log_2(2\pi e \sigma^2) \]  (42)

Conditional Entropy

• Conditional entropy, $H(X|Y)$ — average info needed to represent $X$ after observing $Y$
• If $Y = X$, then $H(X|Y) = 0$
• If knowing $Y$ gives no info for $X$ then $H(X|Y) = H(X)$

Example: Binary symmetric channel

- $p = 0$ or $1$ implies $H(X|Y) = 0$
- $p = 0.5$ implies $H(X|Y) = H(X)$
Mutual Information

- **Average mutual information**: $I(X, Y)$
  
  $$I(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

- It is always positive and is a measure of the average information passed through channel.

**Example:**

- Binary symmetric channel

\[
\begin{array}{ccc}
  x = 0 & & 1-p \\
  & p & \downarrow \\
  x = 1 & & 1-p \\
\end{array}
\]

\[
\begin{array}{ccc}
  y = 0 & & \\
  & & \downarrow \\
  y = 1 & & \\
\end{array}
\]

BSC Example

- Assume $X = \{0.5, 0.5\}$ (i.e. 0 or 1 equally likely)

\[
\begin{array}{c|cc}
  X & 0 & 1 \\
  \hline
  0 & 0.5(1-p) & 0.5(p) \\
  1 & 0.5(p) & 0.5(1-p) \\
\end{array}
\]

\[
X = \{0.5, 0.5\} \quad Y = \{0.5, 0.5\}
\]

\[
H(X) = 1 \quad H(Y) = 1
\]
BSC Example

\[ H(X|Y) = \sum_{i=1}^{2} P(Y=y_i)H(X|Y=y_i) \]
\[ = 0.5H(p, 1-p) + 0.5H(p, 1-p) \]
\[ = H(p, 1-p) \]  
(44)

which is the entropy of 1 random variable with 2 levels having probabilities of \( p \) and \( 1-p \).

\[ H(Y|X) = H(p, 1-p) \]  
(45)

\[ I(X, Y) = H(X) - H(X|Y) = 1 - H(p, 1-p) \]  
(46)

• If \( p = 0.5 \), \( H(0.5, 0.5) = 1 \) and \( I(X, Y) = 0 \): no transmission

• If \( p = 1 \), \( H(1, 0) = 0 \) and \( I(X, Y) = 1 \): full transmission

Channel Capacity, \( C_s \)

• Channel capacity per symbol: \( C_s \)

\[ C_s = \max_{p(x)} I(X, Y) \]  
(47)

• \( C_s \) is in units of bits/symbol

• *It is the maximum* \( I(X, Y) \) *over all possible input probability distributions.*

• If channel is used \( s \) times per second, then capacity in bits/second: \( C \)

\[ C = sC_s \]  
(48)
Capacity Example

- Above is a binary erasure channel
- Receiver knows when a bit was erased
- However, sent bit is unknown
- What should the pdf of the input signal be for capacity to be reached in channel?

\[ x = 0 \quad 1 - \alpha \quad y = 0 \]
\[ \alpha \quad y = 0.5 \]
\[ x = 1 \quad \alpha \quad 1 - \alpha \quad y = 1 \]

Capacity Example

- Can show that capacity is achieved when X is equally likely to be a 0 or a 1
- Capacity is then equal to
  \[ C_s = 1 - \alpha \]  \hspace{1cm} (49) \]
- With feedback from Y to X, can easily build a channel to achieve capacity (ask for a re-transmit)
- But, capacity can be achieved without feedback!
- In general, feedback does not increase capacity of channels (though it often can make the design much simpler).
Channel Capacity with additive Gaussian

- Assume a channel with $Y = X + N$
- $N$ is an independent zero-mean Gaussian random variable with variance $\sigma_n^2$
- From (42), $h(N) = 0.5 \log_2(2\pi e \sigma_n^2)$
- Assume the variance of $X$ is $\sigma_x^2$
- From (47), we want to maximize
  \[ I(X, Y) = h(Y) - h(Y|X) \] (50)

Channel Capacity with additive Gaussian

- However, $h(Y|X) = h(N)$ since once $X$ is known, information left in $Y$ is the noise and this is fixed.
  \[ h(Y|X) = 0.5 \log_2(2\pi e \sigma_n^2) \] (51)
- Need to maximize $h(Y)$ and this is done if $Y$ is Gaussian which is possible if $X$ is Gaussian.
- Maximum of $h(Y)$ given by
  \[ h(Y) = 0.5 \log_2(2\pi e (\sigma_x^2 + \sigma_n^2)) \] (52)
Channel Capacity with additive Gaussian

- Thus, channel capacity is achieved with a Gaussian input resulting in

\[ C_s = 0.5 \log_2 \left( 2 \pi e \left( \sigma_x^2 + \sigma_n^2 \right) \right) - 0.5 \log_2 \left( 2 \pi e \sigma_n^2 \right) \]  

(53)

\[ C_s = 0.5 \log_2 \left( 1 + \frac{\sigma_x^2}{\sigma_n^2} \right) \]  

(54)

in bits/symbol
- This assumes a real-valued random signal.
- \( \sigma_x^2 / \sigma_n^2 \) is the signal-to-noise ratio of the received signal

m-PAM and Channel Capacity

- For \( (\text{BER} = 10^{-10}) \Rightarrow \sigma_n^2 = 0.1547^2 = 1/41.8 \)
- At m-bits per symbol, channel capacity is actually

\[ C_s = 0.5 \log_2 \left( 1 + S_m / \sigma_n^2 \right) \]
\[ = 0.5 \log_2 \left( 1 + \frac{(4^m - 1)}{3 \sigma_n^2} \right) \]
\[ = 0.5 \log_2 \left( \frac{(4^m)}{(3 \sigma_n^2)} \right) \]  

(55)
- which for \( \sigma_n^2 = 1/41.8 \)

\[ C_s = \log_2 \left( 2^m \right) + 0.5 \log_2(13.9) \]
\[ = m + 1.9 \]  

(56)
- So channel capacity is 1.9 more bits/symbol — almost 12 dB power savings (for \( m \) large)
Channel Capacity of a Bandlimited Channel

- Recall

\[ C = 2BC_s \]  

(57)

- Thus, the capacity in bits per second is given by

\[ C = B \log_2(1 + \sigma_x^2/\sigma_n^2) \]  

(58)

- \( \sigma_x^2/\sigma_n^2 \) is the signal-to-noise ratio of the received signal

Normalized Channel Capacity

- For example, a SNR of 20dB has a capacity of 6.66 (bits/s)/Hz.
### Capacity of a Telephone Channel

- The dominant noise source is PCM quantization error
- SNR approximately 35dB
- Bandwidth approximately 3kHz

\[
C = 3000 \times \log_2(1 + 10^{3.5}) = 3000 \times 11.63 \approx 34 \text{ kbits/s} \quad (59)
\]

- To achieve this rate, one needs 11.63 (bits/s)/Hz.
- X56 modems achieve higher rates since the quantization error is not random but known
- 8-bit A/D converters used at 8kS/s - potential for 64kb/s

### Capacity for Non-Flat Channels and Noise

- Break the channel into small \( \Delta f \) regions and integrate.

\[
C = \int_0^\infty \log_2(1 + \frac{S_X^2(f)|H_C(f)|^2}{S_N^2(f)})df \quad (60)
\]

![Diagram of the capacity formula for non-flat channels and noise](image)
Capacity for Non-Flat Channels

- Maximize capacity under fixed input power constraint
  \[ P_s = \int_{-\infty}^{\infty} S_X^2(f) df \]  
  (61)

- Can show that \( S_X^2(f) \) should be chosen such that
  \[ S_X^2(f) = \begin{cases} 
  L - \frac{S_N^2(f)}{|H_C(f)|^2}, & 0 \leq f \leq F \\
  0, & \text{otherwise}
  \end{cases} \]  
  (62)

where \( L \) is chosen to meet the power constraint and \( F \) is the set of freq where \( L > S_N^2/|H_C|^2 \)

- Above known as the \textit{water-pouring spectrum}.

Water-Pouring

- The area equals the transmit power, \( P_s \).
- Note that the spectrum shape of the transmit signal should be large where the SNR is large.
Adaptive Equalization

Adaptive Filter Introduction

- Adaptive filters are used in:
  - Noise cancellation
  - Echo cancellation
  - Sinusoidal enhancement (or rejection)
  - Beamforming
  - Equalization

- Adaptive equalization for data communications proposed by R.W. Lucky at Bell Labs in 1965.
- LMS algorithm developed by Widrow and Hoff in 60s for neural network adaptation
Adaptive Filter Introduction

- A typical adaptive system consists of the following two-input, two output system

\[
\begin{align*}
\delta(n) & \\
\text{adaptive} & \text{algorithm} \\
\end{align*}
\]

- \(u(n)\) and \(y(n)\) are the filter’s input and output
- \(\delta(n)\) and \(e(n)\) are the reference and error signals

Adaptive Filter Goal

- Find a set of filter coefficients to minimize the power of the error signal, \(e(n)\).
- Normally assume the time-constant of the adaptive algorithm is \textit{much slower} than those of the filter, \(H(z)\).
- If it were instantaneous, it could always set \(y(n)\) equal to \(\delta(n)\) and the error would be zero (this is useless)
- Think of adaptive algorithm as an optimizer which finds the best set of \textit{fixed} filter coefficients that minimizes the power of the error signal.
**Noise (and Echo) Cancellation**

- Useful in cockpit noise cancelling, fetal heart monitoring, acoustic noise cancelling, echo cancelling, etc.

\[
H(z) = H_1(z)/H_2(z)
\]

**Sinusoidal Enhancement (or Rejection)**

- The sinusoid's frequency and amplitude are unknown.
- If \( H(z) \) is adjusted such that its phase plus the delay equals 360 degrees at the sinusoid's frequency, the sinusoid is cancelled while the noise is passed.
- The “noise” might be a broadband signal which should be recovered.
Adaptation Algorithm

• Optimization might be performed by:
  • perturb some coefficient in $H(z)$ and check whether the power of
    the error signal increased or decreased.
  • If it decreased, go on to the next coefficient.
  • If it increased, switch the sign of the coefficient change and go on to
    the next coefficient.
  • Repeat this procedure until the error signal is minimized.

• This approach is a steepest-descent algorithm but is slow and not very accurate.

• The LMS (Least-Mean-Square) algorithm is also a steepest-descent algorithm but is more accurate and simpler to realize.
Steepest Descent Algorithm

• In the one-dimensional case

\[
E[e^2(n)]
\]

\[
\begin{align*}
&\Rightarrow \quad \partial E[e^2(n)] > 0 \\
p_i^* &\quad p_i(2) \quad p_i(1) \quad p_i(0)
\end{align*}
\]

Steepest-Descent Algorithm

• In the two-dimensional case

\[
E[e^2(n)]
\]

(\text{out of page})

\[
\begin{align*}
&\Rightarrow \quad \text{Steepest-descent path follows perpendicular to tangents of the contour lines.}
\end{align*}
\]
LMS Algorithm

- Replace expected error squared with instantaneous error squared. Let adaptation time smooth out result.

\[ p_i(n + 1) = p_i(n) - \mu \left( \frac{\partial e^2(n)}{\partial p_i} \right) \]

\[ p_i(n + 1) = p_i(n) - 2\mu e(n) \left( \frac{\partial e(n)}{\partial p_i} \right) \]

- and since \( e(n) = \delta(n) - y(n) \), we have

\[ p_i(n + 1) = p_i(n) + 2\mu e(n)\phi_i(n) \quad \text{where} \quad \phi_i = \frac{\partial y(n)}{\partial p_i} \]

- \( e(n) \) and \( \phi_i(n) \) are uncorrelated after convergence.

Variants of the LMS Algorithm

- To reduce implementation complexity, variants are taking the sign of \( e(n) \) and/or \( \phi_i(n) \).

- **LMS** — \[ p_i(n + 1) = p_i(n) + 2\mu e(n) \times \phi_i(n) \]

- **Sign-data LMS** — \[ p_i(n + 1) = p_i(n) + 2\mu e(n) \times \text{sgn}(\phi_i(n)) \]

- **Sign-error LMS** — \[ p_i(n + 1) = p_i(n) + 2\mu \text{sgn}(e(n)) \times \phi_i(n) \]

- **Sign-sign LMS** — \[ p_i(n + 1) = p_i(n) + 2\mu \text{sgn}(e(n)) \times \text{sgn}(\phi_i(n)) \]

- However, the sign-data and sign-sign algorithms have gradient misadjustment — **may not converge!**

- These LMS algorithms have different dc offset implications in analog realizations.
 Obtaining Gradient Signals

\[ \phi_i(n) = \frac{\partial y(n)}{\partial p_i} = h_{ny}(n) \otimes h_{um}(n) \otimes u(n) \]

- \( H(z) \) is a LTI system where the signal-flow-graph arm corresponding to coefficient \( p_i \) is shown explicitly.
- \( h_{um}(n) \) is the impulse response of from \( u \) to \( m \)
- The gradient signal with respect to element \( p_i \) is the convolution of \( u(n) \) with \( h_{um}(n) \) convolved with \( h_{ny}(n) \).

Gradient Example

\[ \frac{\partial y(t)}{\partial G_2} = -v_{lp}(t) \quad \frac{\partial y(t)}{\partial G_3} = -v_{bp}(t) \]
The performance surface becomes ill-conditioned as the state-signals become correlated (or have large variations).

\[ \frac{\partial y(n)}{\partial p_i} = x_i(n) \]

The gradient signals are simply the state signals

\[ p_i(n + 1) = p_i(n) + 2\mu e(n)x_i(n) \]  \hspace{1cm} (63)

- Only the zeros of the filter are being adjusted.
- There is no need to check that for filter stability (though the adaptive algorithm could go unstable if \( \mu \) is too large).
- The **performance surface is guaranteed unimodal** (i.e. there is only one minimum so no need to worry about being stuck in a local minimum).
- The performance surface becomes ill-conditioned as the state-signals become correlated (or have large power variations).
**Performance Surface**

- Correlation of two states is determined by multiplying the two signals together and averaging the output.
- Uncorrelated (and equal power) states result in a “hyper-paraboloid” performance surface — good adaptation rate.
- Highly-correlated states imply an ill-conditioned performance surface — more residual mean-square error and longer adaptation time.

![Diagram of Performance Surface]

**Adaptation Rate**

- Quantify performance surface — state-correlation matrix
  \[
  R = \begin{bmatrix}
  E[x_1x_1] & E[x_1x_2] & E[x_1x_3] \\
  E[x_2x_1] & E[x_2x_2] & E[x_2x_3] \\
  E[x_3x_1] & E[x_3x_2] & E[x_3x_3]
  \end{bmatrix}
  \]

- Eigenvalues, \( \lambda_i \), of \( R \) are all positive real — indicate curvature along the principle axes.
- For adaptation stability, \( 0 < \mu < \frac{1}{\lambda_{\text{max}}} \) but adaptation rate is determined by least steepest curvature, \( \lambda_{\text{min}} \).
- Eigenvalue spread indicates performance surface conditioning.
Adaptation Rate

- Adaptation rate might be 100 to 1000 times slower than time-constants in programmable filter.
- Typically use same $\mu$ for all coefficient parameters since orientation of performance surface not usually known.
- A large value of $\mu$ results in a larger coefficient “bounce”.
- A small value of $\mu$ results in slow adaptation
- Often “gear-shift” $\mu$ — use a large value at start-up then switch to a smaller value during steady-state.
- Might need to detect if one should “gear-shift” again.

Adaptive IIR Filtering

- The poles (and often the zeros) are adjusted — useful in applications with long impulse responses.
- A stability check might be needed for the adaptive filter itself to ensure the poles do not go outside the unit circle for too long a time (or perhaps at all).
- In general, a multi-modal performance surface occurs. Thus, one should try not be stuck in a local minimum too far from the global minimum
- However, if the order of the adaptive filter is greater than the order of the system being matched (and all poles and zeros are being adapted) — the performance surface is unimodal.
- To obtain the gradient signals for poles, extra filters are generally required.
Adaptive IIR Filtering

- Direct-form structure needs only one additional filter to obtain all the gradient signals.
- However, choice of structure for programmable filter is VERY important — sensitive structures tend to have ill-conditioned performance surfaces.
- Equation error structure has unimodal performance surface but has a bias.
- SHARF (simplified hyperstable adaptive recursive filter) — the error signal is filtered to guarantee adaptation — needs to meet a strictly-positive-real condition
- There are few commercial use of adaptive IIR filters

Digital Adaptive Filters

- FIR tapped delay line is the most common

\[
\begin{align*}
\sum p_i(n)x_i(n) &= y(n) \\
\frac{\partial y(n)}{\partial p_i} &= x_i(n)
\end{align*}
\]
FIR Adaptive Filters

- All poles at $z = 0$ and zeros only adapted.
- Special case of an adaptive linear combiner
- Unimodal performance surface
- States are uncorrelated and equal power if input signal is white — hyper-paraboloid
- If not sure about correlation matrix, can guarantee adaptation stability by choosing
  \[ 0 < \mu < \frac{1}{(\text{# of taps})(\text{input signal power})} \]
- Usually need an AGC so signal power is known.

FIR Adaptive Filter

- Coefficient word length typically $2 + 0.5\log_2(\text{# of taps})$ bits longer than “bit-equivalent” dynamic range
- Example: 6-bit input with 8-tap FIR might have 10-bit coefficient word lengths.
- Example: 12-bit input with 128-tap FIR might have 18-bit coefficient word lengths for 72 dB output SNR.
- Requires multiplies in filter and adaptation algorithm (unless an LMS variant used or slow adaptation rate) — twice the complexity of FIR fixed filter.
- **POWER:** typically 1 (mW/MHz)/tap
- Example [Abbot, ISSCC, 94]: 9 tap FIR filter with 6-bit input and 10b coeff — 500 mW at 72 MHz.
Equalization — Training Sequence

- The reference signal, \( \delta(n) \), is equal to a delayed version of the transmitted data.
- The training pattern should be chosen so as to ease adaptation — pseudorandom is common.
- Above is a feedforward equalizer (FFE) since \( y(n) \) is not directly created using derived output data.

FFE Example

- Suppose channel, \( H_{tc}(z) \), has impulse response 0.3, 1.0, -0.2, 0.1, 0.0, 0.0

- If FFE is a 3-tap FIR filter with \( y(n) = p_1 u(n) + p_2 u(n - 1) + p_3 u(n - 2) \) (64)
- Want to force \( y(1) = 0, y(2) = 1, y(3) = 0 \)
- Not possible to force all other \( y(n) = 0 \)
FFE Example

\begin{align*}
  y(1) &= 0 = 1.0p_1 + 0.3p_2 + 0.0p_3 \\
  y(2) &= 1 = -0.2p_1 + 1.0p_2 + 0.3p_3 \\
  y(3) &= 0 = 0.1p_1 + (-0.2)p_2 + 1.0p_3
\end{align*}

(65)

• Solving results in \( p_1 = -0.266, \ p_2 = 0.886, \ p_3 = 0.204 \)

• Now the impulse response through both channel and equalizer is: 0.0, -0.08, 0.0, 1.0, 0.0, 0.05, 0.02, ...

FFE Example

• Although ISI reduced around peak, introduction of slight ISI at other points (better overall)

• Above is a “zero-forcing” equalizer — usually boosts noise too much

• An LMS adaptive equalizer minimizes the mean squared error signal (i.e. find low ISI and low noise)

• In other words, do not boost noise at expense of leaving some residual ISI
Equalization — Decision-Directed

- After training, the channel might change during data transmission so adaptation should be continued.
- The reference signal is equal to the recovered output data.
- As much as 10% of decisions might be in error but correct adaptation will occur.

Equalization — Decision-Feedback

- **Decision-feedback equalizers** make use of $\delta(n)$ in directly creating $y(n)$.
- They enhance noise less as the **derived** input data is used to cancel ISI.
- The error signal can be obtained from either a training sequence or decision-directed.
DFE Example

- Assume signals 0 and 1 (rather than -1 and +1) (makes examples easier to explain)
- Suppose channel, $H_{mc}(z)$, has impulse response 0.0, 1.0, -0.2, 0.1, 0.0, 0.0
- If DFE is a 2-tap FIR filter with
  \[ y_{DFE}(n) = 0.2\delta(n-1) + (-0.1)\delta(n-2) \]  
  \[(66)\]
- Input to slicer is now 0.0, 1.0, 0.0, 0.0 0.0 0.0

FFE and DFE Combined

- Assuming correct operation, output data = input data
- $e(n)$ same for both FFE and DFE
- $e(n)$ can be either training or decision directed
FFE and DFE Combined

Model as:

\[ Y \frac{N}{X} = H_1 \]

\[ Y \frac{X}{Y} = H_{tc}H_1 + H_2 \]

- When \( H_{tc} \) small, make \( H_2 = 1 \) (rather than \( H_1 \to \infty \))

DFE and FFE Combined

- FFE can deal with precursor ISI and postcursor ISI
- DFE can only deal with postcursor ISI
- However, FFE enhances noise while DFE does not

**When both adapt**

- FFE tries to add little boost by pushing precursor into postcursor ISI (allpass)
Equalization — Decision-Feedback

• The multipliers in the decision feedback equalizer can be simple since received data is small number of levels (i.e. +1, 0, -1) — can use more taps if needed.
• An error in the decision will propagate in the ISI cancellation — error propagation.
• More difficult if Viterbi detection used since output not known until about 16 sample periods later (need early estimates).
• Performance surface might be multi-modal with local minimum if changing DFE affects output data.

Fractionally-Spaced FFE

• Feed forward filter is often a FFE sampled at 2 or 3 times symbol-rate — fractionally-spaced (i.e. sampled at $T/2$ or at $T/3$)
• Advantages:
  — Allows the matched filter to be realized digitally and also adapt for channel variations (not possible in symbol-rate sampling)
  — Also allows for simpler timing recovery schemes (FFE can take care of phase recovery)
• Disadvantage
  Costly to implement — full and higher speed multiplies, also higher speed A/D needed.
Equalization (Error Prediction)

- Error prediction filters also can be used to whiten error and thereby improve adaptation.

- Decorrelates the error at slicer due to FFE noise enhancement to make up for channel attenuation.

Wired Digital Communications
Wired Digital Transmission

Long Twisted-Pair Applications (1km - 6km)
- T1/E1 — 1.5/2Mb/s (2km)
- ISDN — Integrated Services Digital Network
- HDSL — High data-rate Digital Subscriber Line
- ADSL — Asymmetric DSL
- VDSL — Very high data-rate DSL

Short Twisted-Pair Applications (20m - 100m)
- 100Mb/s Fast-Ethernet — TX, T4, T2
- Gigabit Ethernet — Short haul, Long haul
- ATM — 155Mb/s

Short Coax (300m)
- Digital video delivery — 300Mb/s

Cable Modelling
- Modelled as a transmission line.

Twisted-Pair Typical Parameters:
- \[ R(f) = \frac{(1 + j) \sqrt{f/4}}{\Omega/km} \] due to the skin effect
- \[ L = 0.6 \text{ mH/km} \] (relatively constant above 100kHz)
- \[ C = 0.05 \mu F/km \] (relatively constant above 100kHz)
- \[ G = 0 \]
Skin Effect

- “Resistance” is not constant with frequency and is complex valued.
- Can be modelled as:

\[ R(\omega) = k_R(1 + j)\sqrt{\omega} \tag{69} \]

where \( k_R \) is a constant given by

\[ k_R = \frac{1}{\pi d_c} \left( \frac{\mu}{2\sigma} \right)^{1/2} \tag{70} \]

- \( d_c \) is conductor diameter, \( \mu \) is permeability, \( \sigma \) is conductivity
- Note resistance is inversely proportional to \( d_c \).

Characteristic Impedance

\[ Z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}} \tag{71} \]

- Making use of (69) and assuming \( G = 0 \)

\[ Z_0 = \left( \frac{k_R\sqrt{\omega}(1 + j) + j\omega L}{j\omega C} \right)^{1/2} \tag{72} \]

\[ Z_0 = \sqrt{\frac{L}{C}} \left( 1 + \frac{k_R}{L\sqrt{\omega}}(1 - j) \right)^{1/2} \tag{73} \]

Now using approx \( (1 + x)^{1/2} \approx 1 + x/2 \) for \( x \ll 1 \)

\[ Z_0 = \sqrt{\frac{L}{C}} + \frac{k_R}{2\sqrt{\omega LC}}(1 - j) \tag{74} \]

- At high freq, \( Z_0 \) appears as constant value \( \sqrt{L/C} \)
**Characteristic Impedance**

- From (71), when \( \omega L \gg R \) (typically \( \omega \gg 2\pi \times 16kHz \))

\[
Z_{0h} = \sqrt{\frac{L}{C}}
\]

resulting in

\[
Z_{0h} = 110 \ \Omega
\]

- Thus, when terminating a line, a resistance value around 110 \( \Omega \) should be used.

![Graph showing log Z0 vs log f with |Z0| (log) at 110 and 1/\sqrt{f} at 20kHz]

**Cable Transfer-Function**

- When properly terminated, a cable of length \( d \) has a transfer-function of

\[
H(d, \omega) = e^{-d\gamma(\omega)}
\]

where \( \gamma(\omega) \) is given by

\[
\gamma(\omega) = \sqrt{(R + j\omega L)(G + j\omega C)}
\]

- Breaking \( \gamma(\omega) \) into real and imaginary parts,

\[
\gamma(\omega) = \alpha(\omega) + j\beta(\omega)
\]

\[
H(d, \omega) = e^{-d\alpha(\omega)}e^{-jd\beta(\omega)}
\]

- \( \alpha(\omega) \) determines **attenuation**.
- \( \beta(\omega) \) determines **phase**.
Cable Transfer-Function

- Assuming $G = 0$, then from (78)
  \[ \gamma = \left(j \omega CR - \omega^2 LC\right)^{1/2} \]  
  (81)

- Substituting in (69)
  \[ \gamma = \left(j \omega^{1.5} k_R C (1 + j) - \omega^2 LC\right)^{1/2} \]  
  (82)

  \[ \gamma = j \omega \sqrt{LC} \left(1 + \frac{k_R}{L \sqrt{\omega}} (1 - j)\right)^{1/2} \]  
  (83)

Now using approx $\cdot (1 + x)^{1/2} \approx 1 + x/2 \quad \text{for} \quad x \ll 1$

\[ \gamma \approx \frac{k_R}{2 \sqrt{L}} \sqrt{\frac{C}{L}} + j \left(\omega \sqrt{LC} + \frac{k_R}{2 \sqrt{L}} \sqrt{\frac{C}{L}}\right) \]  
(84)

---

Cable Attenuation

- Equating (79) and (84)
  \[ \alpha(\omega) \approx \frac{k_R}{2 \sqrt{L}} \sqrt{\frac{C}{L}} \times \sqrt{\omega} \]  
  (85)

- Therefore gain in dB is
  \[ H_{dB}(d, \omega) \approx -8.68d \times \frac{k_R}{2 \sqrt{L}} \sqrt{\frac{C}{L}} \times \sqrt{\omega} \]  
  (86)

- Note that attenuation in dB is proportional to cable length (i.e. 2x distance doubles attenuation in dB)
- Can reduce attenuation by using a larger diameter cable
- Attenuation proportional to root-frequency
**Cable Attenuation**

- Gain in dB is proportional to $\sqrt{f}$ due to skin effect.

- Do not confuse with $1/f$ noise slow frequency roll-off.

![Graph showing the relationship between gain in dB and frequency for cable attenuation.](image)

**Cable Phase**

- Equating (79) and (84)

$$\beta(\omega) = \omega \sqrt{LC} + \frac{k_R}{2} \sqrt{C/L} \times \omega$$  \hspace{1cm} (87)

- The linear term usually dominates
- The linear term implies a constant group delay.
- In other words, the linear term simply accounts for the delay through the cable.
- Ignoring linear phase portion, remaining phase is proportional to $\sqrt{f}$.
- Note it has the same multiplying term as attenuation.
IIR Filter Cable Match using Matlab

% this program calculates an iir num/den transfer-function
% approx for a transmission line with exp(sqrt(s)) type response.
clear;

% Order of IIR filter to match to cable
% nz is numerator order and np is denominator order
nz = 9;
np = 10;

% important parameters of cable
C = 0.05e-6 % capacitance per unit length in farads/km
L = 0.6e-3 % inductance per unit length in henries/km
KR = 0.25 % resistance per unit length in ohms/km (times (1+j)*sqrt(omega))
D = 0.1 % cable length in km
% above values adjusted to obtain -20dB atten for 100m at 125MHz
K_CABLE = (KR/2)*sqrt(C/L);

% the frequency range for finding tf of cable
fmin=1;
fmax=1e9;

% specify frequency points to deal with
nmax=1000;
f=logspace(log10(fmin), log10(fmax), nmax);
w=2*pi*f;
s=j*w;

% 'cable' is desired outcome in exponential form
CABLE = exp(-D*K_CABLE*sqrt(2)*sqrt(s));

% Perform IIR approximate transfer-function match
% Since invfreqs miminizes (num-cable*den)
% first need an approximate den so that it can be used
% as a freq weighting to minimize (num/den - cable)
[num,den]=invfreqs(cable,w,nz,np, 1./w);
[denor]=freqs(den,1,w);
% re-iterate process with weighting for the denominator
% which now minimizes (num/den - cable)
[num,den]=invfreqs(cable,w,nz,np, (1./denor).^2);
[denor]=freqs(den,1,w);
[num,den]=invfreqs(cable,w,nz,np, (1./denor).^2);

% find approximate transfer function 'cable_approx' to 'cable'
[CABLE_APPROX]=freqs(num,den,w);
% also find pole-zero model
[Z,p,k]=tf2zp(num,den);

% PLOT RESULTS
clf;
figure(1);
subplot(211);
semilogx(f,20*log10(abs(cable)),'r');
hold on;
semilogx(f,20*log10(abs(cable_approx)),'b');
title('Cable Magnitude Response');
xlabel('Freq (Hz)');
ylabel('Gain (dB)');
grid;
hold off;
subplot(212);
semilogx(f,angle(cable)*180/pi,'r');
hold on;
semilogx(f,angle(cable_approx)*180/pi,'b');
title('Cable Phase Response');
xlabel('Freq (Hz)');
ylabel('Phase (degrees)');
grid;

hold off;
figure(2);
subplot(211);
semilogx(f,20*log10(abs(cable)./abs(cable_approx)));
title('Gain Error Between Cable and Cable_approx');
xlabel('Freq (Hz)');
ylabel('Gain Error (dB)');
subplot(212);
semilogx(f,(angle(cable)-angle(cable_approx))*180/pi);
title('Phase Error Between Cable and Cable_approx');
xlabel('Freq (Hz)');
ylabel('Phase Error (degrees)');
grid;
Near and Far End Crosstalk

- In FEXT, interferer and signal both attenuated by cable
- In NEXT, signal attenuated but interferer is coupled directly in.
- When present, NEXT almost always dominates.
- Can cancel NEXT if nearby interferer is known.
- Envelope of squared gain of NEXT increases with $f^{1.5}$

Twisted-Pair Crosstalk

- Crosstalk depends on turns/unit length, insulator, etc.
- Twisted-pairs should have different turns/unit length within same bundle

- 15dB/decade loss
- Typical cat 3 wiring
**Transformer Coupling**

- Almost all long wired channels (>10m) are AC coupled systems
- AC coupling introduces **baseline wander** if random PAM sent
- A long string of like symbols (for example, +1) will decay towards zero degrading performance
- Requires baseline wander correction (non-trivial)
- Can use passband modulation schemes (CAP, QAM, DMT)

**Why AC couple long wired channels??**

---

**Transformer Coupling**

Eliminates need for similar grounds

- If ground potentials not same — large ground currents

**Rejects common-mode signals**

- Transformer output only responds to differential signal current
- Insensitive to common-mode signal on both wires
**Generic Wired PAM Transceiver**

- Look at approaches for each block

**HDSL Application**

- 1.544Mb/s over 4.0km of existing telephone cables.
- Presently 4-level PAM code (2B1Q) over 2 pairs (a CAP implementation also exists).
- Symbol-rate is 386 ksymbols/s

**Possible Bridged-Taps**

- Can have unterminated taps on line
- Modelling becomes more complicated but DFE equalizes effectively
- Also causes a wide variation in input line impedance to which echo canceller must adapt — difficult to get much analog echo cancellation
HDSL Application

- Symbol-rate is 386 ksymbols/s

Received Signal

- For $d = 4km$, a 200kHz signal is attenuated by $40dB$.
- Thus, high-freq portion of a 5Vpp signal is received as a 50mVpp signal — Need effective echo cancellation

Transmit Path

- Due to large load variations, echo cancellation of analog hybrid is only 6dB
- To maintain 40dB SNR receive signal, linearity and noise of transmit path should be better than 74dB.

ISDN Application

- Similar difficulty to HDSL but lower frequency
- 160kb/s over 6km of 1 pair existing telephone cables
- 4-level PAM coding — 2B1Q
- Receive signal at 40kHz atten by 40dB
- Requires highly linear line-drivers + A/D converters for echo cancellation (similar to HDSL)
Fast-Ethernet Application

<table>
<thead>
<tr>
<th>CAT3</th>
<th>CAT5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_{dB}(f) = 2.32 \sqrt{f} + 0.238f$</td>
<td>$H_{dB}(f) = 1.967 \sqrt{f} + 0.023f + 0.05/\sqrt{f}$</td>
</tr>
<tr>
<td>12.5MHz ↔ 11dB</td>
<td>12.5MHz ↔ 7dB</td>
</tr>
<tr>
<td>crosstalk worse</td>
<td>crosstalk better</td>
</tr>
</tbody>
</table>

**100Base-T4**
- 4 pair CAT3 — 3 pair each way, 25MS/s with coding

**100Base-TX**
- 2 pair CAT5 — 3 level PAM to reduce radiation

**100Base-T2**
- 2 pair CAT3 — 5x5 code, 25MS/s on each pair

Typical Transmit D/A Block

- Polyphase filter to perform upsampling + filtering

**HDSL**
- D/A and filter needs better than 12-bit linearity
- Might be an oversampled 1-bit DAC
- One example: $\uparrow 16$; 48 tap FIR; $\uparrow 4$; ΔΣ DAC

**Fast-Ethernet**
- Typically around 35 dB linearity + noise requirement
- 100Base-T2 example: $\uparrow 3$; simple FIR; 75MHz 4-bit DAC; 3'rd-order LP cont-time filter
**Line Drivers**

- Line driver supplies drive current to cable.
- Commonly realized as voltage buffers.
- Often the most challenging part of analog design.
- Turns ratio of transformer determines equivalent line impedance.

\[
\begin{align*}
V_{ne} &= \frac{2}{n}V_2 \\
V_1 &= V_2/n \\
I_1 &= nI_2 \\
R_1 &= R_2/n^2
\end{align*}
\]

**Typical Values**
- \( R_2 = 100\Omega \)
- \( V_2 = \pm2.5V \)
- \( I_2 = \pm25mA \)

**Line Driver Efficiency**

- Efficiency improves as power supply increased

**Example (assume can drive within 1V of supplies)**

- From typical values, max power delivered by line driver is
  \[
  P_{\text{line}+R} = 2 \times 2.5 \times 25mA = 125mW
  \]

**12V Case**

- Consider 12V supply — use \( n = 0.5, V_{ne,\text{max}} = 10V, \)
  \( I_{L,\text{max}} = 12.5mA \) leading to \( P = 12 \times 12.5mA = 150mW \)
  (and drive an 800 ohm load)

**3V Case**

- Consider 3V supply — use \( n = 5, V_{ne,\text{max}} = 1V, \)
  \( I_{L,\text{max}} = 125mA \) leading to \( P = 3 \times 125mA = 375mW \)
  (and drive an 8 ohm load!!!
Line Driver

• In CMOS, W/L of output stage might have transistors on the order of 10,000!
• Large sizes needed to ensure some gain in final stage so that feedback can improve linearity — might be driving a 30 ohm load
• When designing, ensure that enough phase margin is used for the wide variation of bias currents
• Nested Miller compensation has been successfully used in HDSL application with class AB output stage
• Design difficulties will increase as power supplies decreased

2-4 Wire Hybrids

• Dual-duplex often used to reduce emission.
• However, dual-duplex requires hybrids and echo cancellation.
  
  ![Diagram](image)

  - If $R_L = R_T$, no echo through hybrid
  - Can be large impedance variation.
**Typical HDSL Line Impedances**

Loop Impedances

| Frequency [Hz] | \(|Z|\) |
|---------------|--------|
| 0             | 0      |
| 0.5           | 50     |
| 1             | 100    |
| 1.5           | 150    |
| 2             | 200    |
| 2.5           | 250    |
| 3             | 300    |
| 3.5           | 350    |
| 4             | 400    |
| 4.5           | 450    |
| 5 x 10^5      | 500    |

**Hybrid Issues**

- Note zero at dc and pole at 10kHz.
- Low frequency pole causes long echo tail (HDSL requires 120 tap FIR filter)

**Alternatives**

- Could eliminate \( R_1 \) circuit and rely on digital echo cancellation but more bits in A/D required.

**OR**

- Can make \( R_1 \) circuit more complex to ease A/D specs.
- Less echo return eases transmit linearity spec.
- Might be a trend towards active hybrids with or without extra A/D and D/A converters (particularly for higher speeds).
Typical Receive A/D

- Often, VGA is controlled from digital signal.
- Anti-aliasing can be simple in oversampled systems.
- Continuous-time filters are likely for fast-ethernet.
- Example: 100Base-T2 suggests a 5'th order continuous-time filter at 20MHz with a 6-bit A/D at 75MHz.
- Challenge here is to keep size and power of A/D small.

---

Echo Cancellation

- Typically realized as an adaptive FIR filter.
- Note input is transmit signal so delay lines and multiplies are trivial.
- HDSL uses about a 120 tap FIR filter.
- Coefficient accuracy might be around 20 bits for dynamic range of 13 bits.
Echo Cancellation

• Fast-ether net might be around 30 taps and smaller coefficient accuracy
• Can also perform some NEXT cancellation if signal of nearby transmitter is available (likely in 100Base-T2 and gigabit ethernet)

Alternatives

• Higher data rates may have longer echo tails.
• Might go to FIR/IIR hybrid to reduce complexity.
• Non-linear echo cancellation would be VERY useful in reducing transmit linearity spec.
• However, these non-linearities have memory and thus Volterra series expansions needed.

Equalization

HDSL

• Echo canceller required before equalization so fractional spaced equalizer not practical
• Typically 9 tap FFE and 120 tap DFE
• Long DFE also performs dc recovery (baseline wander)

Fast Ethernet

• Often fractional-spaced EQ - 30 taps
• DFE — 20 taps (dc recovery)
**dc Recovery (Baseline Wander)**

- Wired channels often ac coupled
- Reduces dynamic range of front-end circuitry and also requires some correction if not accounted for in transmission line-code

![Diagram of dc Recovery (Baseline Wander)]

- Front end may have to be able to accommodate twice the input range!
- DFE can restore baseline wander - lower frequency pole implies longer DFE
- Can use line codes with no dc content

---

**Baseline Wander Correction #1**

**DFE Based**

- Treat baseline wander as postcursor interference
- May require a long DFE

\[
\frac{z - 1}{z - 0.5} = 1 - \frac{1}{2}z^{-1} - \frac{1}{4}z^{-2} - \frac{1}{8}z^{-3} - \ldots
\]

**IMPULSE INPUT**

\[
\begin{align*}
0 & 1 -0.5 -0.25 -0.125 -0.06 \ldots \\
0 & 1 \ 0 \ 0 \ 0 \ 0 \ \ldots \\
0 & 0 \ 0 \ 0.5 \ 0.25 \ 0.125 \ 0.06 \ \ldots \\
& \frac{1}{2}z^{-1} + \frac{1}{4}z^{-2} + \frac{1}{8}z^{-3} + \ldots
\end{align*}
\]
Baseline Wander Correction #1

DFE Based

\[
\frac{z - 1}{z - 0.5} = 1 - \frac{1}{2} z^{-1} - \frac{1}{4} z^{-2} - \frac{1}{8} z^{-3} - \ldots
\]

**STEP INPUT**

0 1 0.5 0.25 0.125 0.06 ... → 0 1 1 1 1 1 ... → 0 1 1 1 1 1 ... → 0 1 1 1 1 1 ...

0 0 0.5 0.75 0.875 0.938 ...

Baseline Wander Correction #2

Analog dc restore

**STEP INPUT**

0 1 1 1 1 1 ... → 0 1 1 1 1 1 ... → 0 1 1 1 1 1 ...

• Equivalent to an analog DFE
• Needs to match RC time constants
Baseline Wander Correction #3

Error Feedback

- Integrator time-constant should be faster than ac coupling time-constant
- Effectively forces error to zero with feedback
- May be difficult to stabilize if too much in loop (i.e. AGC, A/D, FFE, etc)

Timing Recovery
Timing Recovery (two types)

- Timing more difficult with less excess bandwidth.

Deductive Timing Recovery

- Non-linear spectral line method most popular (linear spectral line method used if $f_s$ tone present).
- Apply a non-linearity to receive signal and bandpass filter to recover $f_s$ tone (usually with PLL).
- Works because receive signal is cyclostationary (i.e. its moments vary in time and are periodic).
- Common non-linearities used are squaring and absolute circuit (rectifier) (for low excess BW)
- *Ensemble average* of non-linear circuit output is periodic in T
- Thus, a $f_s$ component exists (scrambled data)
Example (100% excess BW)

-15 -10 -5 0 5 10 15

-1.5 -1 -0.5 0 0.5 1

receive signal

abs(receive signal)

average(abs(receive signal))

average NOT in time but over transmit sequences (100 sequences in this case)

Example (20% excess BW)

-15 -10 -5 0 5 10 15

-2 -1.5 -1 -0.5 0 0.5 1 1.5 2

-15 -10 -5 0 5 10 15

0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1.2 1.4 1.6 1.8

receiv e signal

abs(receive signal)

average(abs(receive signal))

average NOT in time but over transmit sequences (100 sequences in this case)
**Deductive Timing**

- Can pre-filter receive signal to only non-flat portion to reduce jitter — eliminate portion that does not contribute to timing tone.

![Diagram showing Deductive Timing](image)

**Inductive Timing — Early Late**

- Can sample at 2X and determine if clock is early or late when a transition occurs.

![Diagram showing Inductive Timing](image)

- **Early**
  - Sample at (a), (b), and (c)
  - If (a) = (b) ≠ (c), slow down clock

- **On-time**
  - Sample at (a), (b), and (c)

- **Late**
  - Sample at (a), (b), and (c)
  - If (a) ≠ (b) = (c), speed up clock

- If (a) = (b) = (c), do nothing

- However, (b) sample does not indicate how far away from zero crossing — can add dither to (b) to aid estimate.
Inductive Timing (MMSE)

- Commonly realized as minimum mean-square error (i.e. MMSE timing)
- Also called LMS timing.
- Assume sample times are \( kT + \tau_k \)
  \[ A_k = \pm 1 \]

Correct sampling phase  Late sampling phase  \( \tau_k > 0 \)

\[ E \left[ E_k^2(\tau_k) \right] = E \left[ (Q_k(\tau_k) - A_k)^2 \right] \]  \( (88) \)

where \( E[\bullet] \) denotes expectation, \( Q_k(\tau_k) \) is the sampled signal (it is a function of \( \tau_k \)) and \( A_k \) is the ideal symbol.

- Stochastic gradient (as in LMS algorithm) leads to
  \[ \tau_{k+1} = \tau_k - \mu \left( E_k(\tau_k) \times \frac{\partial Q_k(\tau_k)}{\partial \tau_k} \right) \]  \( (89) \)
Inductive Timing (MMSE)

- Can replace derivative wrt $\tau_k$ by derivative wrt time since sampled at $t = kT + \tau_k$

$$\frac{\partial Q_k(\tau_k)}{\partial \tau_k} = \left. \frac{\partial Q(t)}{\partial t} \right|_{t = kT + \tau_k}$$ (90)

Inductive Timing (MMSE)

- Can sample at 2X symbol-rate and perform derivative in discrete-time.
2X Timing Example

- Sample at twice symbol-rate

\[ \tau_{k+1} = \tau_k + \mu (Q_k - \hat{A}_k) \times (Q_{k+1} - Q_{k-1}) \]  \hspace{1cm} (91)

\[ A_k = \pm 1 \]

Correct sampling phase \hspace{1cm} Late sampling phase \hspace{1cm} \tau_k > 0

- At \( Q_k \), slope is neg, \( E_k \) is neg, so \( \tau_k \) is decreased.
- Use absolute values then 50% duty cycle not needed

Inductive Timing — Baud-Rate

- If all sampling done at symbol-rate, MMSE timing can still be used — base it on impulse response.

- Early-late — adjust so \( h_1 - h_{-1} = 0 \)
- Zero-crossing — adjust so \( h_1 = 0 \)
Inductive Timing — Baud-Rate

- To obtain impulse response estimates, cross correlate received signals with received symbols.
- Recall

\[ Q(t) = \sum_{m} A_m h(t - mT) + n(t) \]  

(92)

- Sampled at time \( kT + \tau \), we have

\[ Q_k = Q(kT + \tau) \]

\[ = \ldots + A_{k-1} h(kT + \tau - (k-1)T) + A_k h(kT + \tau - kT) + \ldots \]

\[ = \ldots + A_{k-1} h_1(\tau) + A_k h_0(\tau) + A_{k+1} h_{-1}(\tau) + \ldots \]  

(93)

where \( h_k(\tau) \equiv h(kT + \tau) \)

Inductive Timing — Baud-Rate

- To estimate \( h_1(\tau) \), use \( Q_k \times A_{k-1} \)

- All other terms go to zero since \( A_{k-1} \) is uncorrelated with \( A_j \) when \( k \neq j \)

- To estimate \( h_{-1}(\tau) \), we need to use a delayed version of \( Q_k \)

\[ Q_{k-1} = \ldots + A_{k-1} h_0(\tau) + A_k h_{-1}(\tau) + A_{k+1} h_{-2}(\tau) + \ldots \]  

(94)

- To estimate \( h_{-1}(\tau) \), use \( Q_{k-1} \times A_k \)
Inductive Timing — Baud-Rate

- To build early-late scheme,

\[ \begin{align*}
Q_k & \quad z^{-1} \\
A_k & \quad z^{-1}
\end{align*} \]

- Early-late is insensitive to amplitude distortion.
- Zero-crossing is better where phase distortion dominates
- \( h_0 \) factor should be known otherwise adaptation gain will vary (can divide it out in algorithm).

A Fractional-N Frequency Synthesizer

- Often need a low jitter clock that can have arbitrary frequency.
- A voltage-controlled crystal oscillator is expensive.
- Use oversampling within a PLL

\[ \begin{align*}
\text{crystal} & \quad f_{xt} \\
\div M & \quad \text{phase detect} \\
\div N & \quad \text{loop filter} \\
\div P & \quad \text{VCO} \\
\frac{Nf_{xt}}{PM} & \\
\end{align*} \]

\[ N = \{ k-1, k, k+1 \} \]

A digital controlled oscillator
Elastic Buffer

- Used to deal with low frequency input clock jitter
- Allows attenuation of clock jitter to next stage

Example

- Input clock rate — 1MHz but varies from 0.9MHz to 1.1MHz in sinusoidal fashion at 1kHz
- Output clock rate — fixed at 1MHz
- Input clock high — 16 extra bits stored in buffer
- Input clock low — 16 bits removed from buffer

- Keep elastic buffer half-full on-average through feedback

Other Modulation Schemes

QAM, CAP and DMT
Complex Signals

- Concept useful for describing a pair of real signals
- Let $j = \sqrt{-1}$

Two Important Properties of Real Signals

- Amplitude is symmetric ($A(j\omega) = |A(-j\omega)|$)
- Phase is anti-symmetric ($\angle A(j\omega) = -1 \times \angle A(-j\omega)$)

Two Important Complex Relationships

- Continuous-time
  \[ e^{j\omega t} = \cos(\omega t) + j\sin(\omega t) \]  
  (95)
- Discrete-time
  \[ e^{j\omega n T} = \cos(\omega n T) + j\sin(\omega n T) \]  
  (96)

Complex Transfer Function

- Let $h(t)$ be a complex impulse response
  \[ h(t) = \Re \{ h(t) \} + j \Im \{ h(t) \} \]  
  (97)
- 4 systems needed if both $h(t)$ and $u(t)$ complex
- 1 system needed if both $h(t)$ and $u(t)$ real
- 2 systems needed if one is complex and other real
Hilbert Transform

- Often need a complex signal with all negative frequency components zero — use Hilbert transform

- Hilbert transform is a real filter with response

\[ h_{bt}(t) = \frac{1}{\pi t} \quad (98) \]
\[ H_{bt}(j\omega) = -j \text{sgn}(\omega) \quad (99) \]

- The Hilbert transform of a signal \( x(t) \) is denoted as \( \hat{x}(t) \) and can be found using filter in (99)

\[ \hat{X}(j\omega) = -j \text{sgn}(\omega)X(j\omega) \quad (100) \]

- Shift phase of signal by -90 degrees at all frequencies — allpass filter with phase shift

- Recall \( j = e^{-j\pi/2} \)

Phase Splitter

- A complex system, \( \phi(t) \), that removes negative frequency components referred to as a phase splitter.

\[ \Phi(j\omega) = \begin{cases} 
1, & \omega \geq 0 \\
0, & \omega < 0 
\end{cases} \quad (101) \]

- A phase splitter is built using a Hilbert transform (hence the name phase splitter)
**Phase Splitter**

- To form a signal, $u(t)$, having only positive freq components from real signal, $x(t)$

\[ u(t) = 0.5(x(t) + j\dot{x}(t)) \]  

(102)

- $u(t)$ is two real signals where we think of signals as

\[ x(t) = \text{Re}\{2u(t)\} \]  

(103)

\[ \dot{x}(t) = \text{Im}\{2u(t)\} \]  

(104)

- To see that only positive frequency components remain — use (100) and (102)

\[ U(j\omega) = 0.5(X(j\omega) + j \times (-j \text{sgn}(\omega)X(j\omega))) \]  

(105)

\[ U(j\omega) = 0.5(X(j\omega) + \text{sgn}(\omega)X(j\omega)) \]  

(106)
Real-Valued Modulation

\[ y(t) = x(t) \cos(\omega_c t) \] (107)

- Multiplication by \( \cos(\omega_c t) \) results in convolution of frequency spectrum with two impulses at \( +\omega_c \) and \( -\omega_c \)

Complex Modulation

\[ y(t) = e^{j\omega_c t} x(t) \] (108)

- Mult a signal by \( e^{j\omega_c t} \) shifts spectrum by \( +\omega_c \)
Passband and Complex Baseband Signals

- Can represent a passband signal as a complex baseband signal.
- Need complex because passband signal may not be symmetric around $\omega_c$.

$$ Y(j\omega) \xrightarrow{\text{phase splitter}} y(t) \xrightarrow{\sqrt{2} \phi(t)} u(t) \xrightarrow{e^{-j\omega_c t}} U(j\omega) $$

- $\sqrt{2}$ factor needed to keep the same signal power.

Modulation of Complex Baseband

- It is only possible to send real signals along channel.
- Can obtain passband modulation from a complex baseband signal by complex modulation then taking real part.

$$ U(j\omega) \xrightarrow{u(t)} v(t) \xrightarrow{\sqrt{2} \text{Re}\{ \}} y(t) \xrightarrow{e^{j\omega_c t}} Y(j\omega) $$

- Works because $v(t)$ has only positive freq. therefore its imag part is its Hilbert transform and taking real part restores negative frequencies.
Double Sideband

- \( x(t) \) is a real signal so positive and negative frequencies symmetric
- Modulated signal, \( y(t) \), has symmetry above and below carrier freq, \( \omega_c \) — using twice minimum bandwidth necessary to send baseband signal.

\[
\begin{align*}
 v(t) &= x(t) \times (\cos(\omega_c t) + j \sin(\omega_c t)) \\
y(t) &= \sqrt{2} x(t) \cos(\omega_c t)
\end{align*}
\]

Single Sideband

- Twice as efficient as double sideband
- Disadvantage — requires a phase-splitter good to near dc (difficult since a phase discontinuity at dc)
Single Sideband

- If \( v_1(t) = a(t) + j b(t) \), then \( y(t) = \text{Re}\{e^{j\omega_c t} v_1(t)\} \) becomes

\[
y(t) = \sqrt{2} \text{Re}\{(\cos(\omega_c t) + j \sin(j \omega_c t)) \times (a(t) + j b(t))\}
\]

\[
y(t) = \sqrt{2} a(t) \cos(\omega_c t) - \sqrt{2} b(t) \sin(\omega_c t)
\]

Quadrature Amplitude Modulation (QAM)

- Start with two independent real signals

\[
u(t) = a(t) + j b(t)
\]

- In general, they will form a complex baseband signal
- Modulate as in single-sideband case

\[
y(t) = \sqrt{2} a(t) \cos(\omega_c t) - \sqrt{2} b(t) \sin(\omega_c t)
\]

- For data communications let \( a(t) \) and \( b(t) \) be the output of two pulse shaping filters with multilevel inputs, \( A_k \) and \( B_k \)
- While QAM and single sideband have same spectrum efficiency, QAM does not need a phase splitter
- Typically, spectrum is symmetrical around carrier but information is twice that of double-side band.
**QAM**

- Can draw signal constellations

QAM 4  
QAM 16  
QAM 64

- Can Gray encode so that if closest neighbor to correct symbol chosen, only 1 bit error occurs
QAM

- To receive a QAM signal, use correlation receiver

\[
\begin{align*}
\text{input} & \rightarrow g(-t) \\
\cos(\omega_c t) & \rightarrow \hat{A}_k \\
\sin(\omega_c t) & \rightarrow \hat{B}_k
\end{align*}
\]

- When transmitting a small bandwidth (say 20kHz) to a large carrier freq (say 100MHz), often little need for adaptive equalization — use fixed equalizer

CAP

- Carrierless AM-PM modulation
- Essentially QAM modulated to a low carrier, \( f_c \)

\[
\begin{align*}
A_k & \rightarrow g(t) \\
B_k & \rightarrow g(t) \\
a(t) & \rightarrow \cos(\omega_c t) \\
b(t) & \rightarrow \sin(\omega_c t) \\
u(t) & = a(t) + jb(t) \\
y(t) & = \sqrt{2} u(t)
\end{align*}
\]
CAP

- BIG implementation difference — can directly create impulse response of two modulated signals.

\[ A_k \xrightarrow{g_i(t)} g_i(t) \xrightarrow{+} \frac{\sqrt{2}}{y(t)} \]

\[ B_k \xrightarrow{g_q(t)} g_q(t) \xrightarrow{+} \frac{\sqrt{2}}{y(t)} \]

where

\[ g_i(t) = g(t) \cos(\omega_c t) \] (115)

\[ g_q(t) = g(t) \sin(\omega_c t) \] (116)

- Not feasible if \( \omega_c \) is much greater than symbol freq

- Two impulse responses are orthogonal

\[ \int_{-\infty}^{\infty} g_i(t)g_q(t)dt = 0 \] (117)

CAP

- The choice for \( \omega_c \) depends on excess bandwidth

- Excess bandwidth naturally gives a notch at dc

- For 100% excess bandwidth \( \omega_c = f_s \)

- For 0% excess bandwidth \( \omega_c = f_s/2 \)
Example — Baseband PAM

- Desired Rate of 4Mb/s — Freq limited to 1.5MHz
- Use 50% excess bandwidth ($\alpha = 0.5$)
- Use 4-level signal (2-bits) and send at 2MS/s

Example — CAP

- Desired Rate of 4Mb/s — Freq limited to 1.5MHz
- Use 50% excess bandwidth ($\alpha = 0.5$)
- Use CAP-16 signalling and send at 1MS/s

- Note faster roll-off above 1MHz
- Area under two curves the same
**CAP**

- Two matched filters used for receiver

\[
\begin{align*}
&\text{input} \\
&g_i(t) \xrightarrow{f_s} \hat{A}_k \\
&g_q(t) \xrightarrow{f_s} \hat{B}_k
\end{align*}
\]

- When adaptive, need to adapt each one to separate impulse — should ensure they do not converge to same impulse

**CAP vs. PAM**

- Both have same spectral efficiency
- Carrier recovery similar? (not sure)

- CAP is a passband scheme and does not rely on signals near dc
- More natural for channels with no dc transmission

- Can always map a PAM scheme into CAP
  - 2-PAM $\leftrightarrow$ 4-CAP
  - 4-PAM $\leftrightarrow$ 16-CAP
  - 8-PAM $\leftrightarrow$ 64-CAP

- Cannot always map a CAP scheme into PAM
  - cannot map $32$-CAP into PAM since $\sqrt{32}$ is not an integer number
**DMT Modulation**

- Discrete-MultiTone (DMT)
- A type of multi-level orthogonal multipulse modulation
- More tolerant to radio-freq interference
- More tolerant to impulse noise
- Can theoretically achieve closer to channel capacity
- Generally more complex demodulation
- Generally more latency

**ADSL (Asymmetric DSL)**

- 6Mb/s to home, 350kb/s back to central office over existing twisted-pair
- POTS splitter so telephone can coexist

---

**Multipulse Modulation**

- Consider the two orthogonal signals from CAP — one transmission scheme is to transmit $g_1(t)$ for a binary 1 and $g_2(t)$ for a binary 0.
- Use a correlation receiver to detect which one was sent.
- Spectral efficiency (if $\alpha = 0$) is only $1 \text{ (symbols/s)/Hz}$ rather than $2 \text{ (symbols/s)/Hz}$ in the case of PAM
- In general, need $N\pi/T$ bandwidth to send $N$ orthogonal pulses
- PAM, $N = 1$, minimum bandwidth: $\pi/T$
- QAM and CAP, $N = 2$, minimum bandwidth: $2\pi/T$
**Combined PAM and Multipulse**

- Changing scheme to sending $\pm g_i(t)$ and $\pm g_q(t)$ becomes a 2-level for each 2 orthogonal multipulses which is same as 4-CAP
- Multitone uses many orthogonal pulses as well as multi-levels on each (each pulse may have different and/or varying number of multi-levels)

- In discrete-form, it makes use of FFT — called Discrete MultiTone (DMT)
- Also called MultiCarrier Modulation (MCM)

---

**Bit Allocation**

- Allocate more bits where SNR is best

- A radio interferer causes low SNR at $f_x$
- Perhaps send only 1 b/s/Hz in those bands
- At high SNR send many b/s/Hz
FFT Review

- FFT is an efficient way to build a DFT (Discrete Fourier Transform) when number of samples $N = 2^M$
- If rectangular window used and time-domain signal periodic in $N$, then FFT has impulses in freq domain

\[ N \]

\[ N^2 \]

\[ +3 \]

\[ +1 \]

\[ -1 \]

\[ -3 \]

\[ \frac{4\pi}{N}, \pi, 2\pi \]

(freq (rad/sample))

DMT Generation

- Input to IFFT (inverse FFT) is quantized impulses at each freq (real and imag)
- Forced symmetric around $\pi$ (complex conjugate)
- Output is real and is sum of quantized amplitude sinusoids
- Quantized real - quantized amplitude cosine
- Quantized imag - quantized amplitude sine
- Symbol-rate is much lower than bandwidth used
Example — N=4

DMT Modulation
DMT Modulation

- Symbol Length, $T$
  - make symbol length as long as tolerable
  - typically need 3 symbol periods to decode
- If max channel bandwidth is $f_{\text{max}}$, sampling rate should be $f_{\text{samp}} > 2f_{\text{max}}$
- Choose $N = 2^M > f_{\text{samp}}T$ where $M$ is an integer

Example

- Max channel bandwidth is 1MHz,
- $f_{\text{samp}} = 2$MHz, $N = 512$ results in $M = 9$, $T = 1/3.9$kHz
- Channel bandwidths are $\Delta f = f_{\text{max}}/(N/2) = 3.9$kHz

Cyclic Prefix

- If channel is modelled as having a finite impulse response on length L, send last L samples at beginning to ignore transient portion of channel
- Could send much more but no need
- When receiving, ignore first L samples received (purge out transient part of channel)
- Each FFT bin will undergo phase and magnitude change, equalize out using a complex multiplication
- If channel model too long, pre-equalize to shorten significant part of channel impulse response
DMT Modulation

- Clock sent in one frequency bin
- More tolerant to impulse noise because of long symbol length
  - expect around $10 \log(N)$ dB improvement
  - $N = 512$ implies 27 dB improvement
- Longer latency
- Can place more bits in frequency bins where more dynamic range occurs (achieve closer to capacity)
- Transmit signal appears more Gaussian-like
  - a large Crest factor
  - more difficult line driver
  - need channel with less distortion or clipping
Coding

Scrambling (Spectrum control)
- “Whiten” data statistics
- Better for dc balance and timing recovery

Line Coding (Spectrum control)
- Examples: dc removal or notch

Hard-Decoding (Error Control)
- Error detection or correction — received bits used

Soft-Decoding (Error Control)
- Error prevention
- Most likely sequence — received samples used
**PN Sequence Generators**

- Use \( n \)-bit shift register with feedback
- If all-zero state occurs, it remains in that state forever
- Maximal length if period is \( 2^n - 1 \)

**Maximal-Length PN Sequences**

<table>
<thead>
<tr>
<th>Delay Length</th>
<th>Feedback Taps</th>
<th>Delay Length</th>
<th>Feedback Taps</th>
<th>Delay Length</th>
<th>Feedback Taps</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1,2</td>
<td>13</td>
<td>1,3,4,13</td>
<td>24</td>
<td>1,2,7,24</td>
</tr>
<tr>
<td>3</td>
<td>1,3</td>
<td>14</td>
<td>1,6,10,14</td>
<td>25</td>
<td>3,25</td>
</tr>
<tr>
<td>4</td>
<td>1,4</td>
<td>15</td>
<td>1,15</td>
<td>26</td>
<td>1,2,6,26</td>
</tr>
<tr>
<td>5</td>
<td>2,5</td>
<td>16</td>
<td>1,3,12,16</td>
<td>27</td>
<td>1,2,5,27</td>
</tr>
<tr>
<td>6</td>
<td>1,6</td>
<td>17</td>
<td>3,17</td>
<td>28</td>
<td>3,28</td>
</tr>
<tr>
<td>7</td>
<td>3,7</td>
<td>18</td>
<td>8,18</td>
<td>29</td>
<td>2,29</td>
</tr>
<tr>
<td>8</td>
<td>2,3,4,8</td>
<td>19</td>
<td>1,2,5,19</td>
<td>30</td>
<td>1,2,23,30</td>
</tr>
<tr>
<td>9</td>
<td>4,9</td>
<td>20</td>
<td>3,20</td>
<td>31</td>
<td>3,31</td>
</tr>
<tr>
<td>10</td>
<td>3,10</td>
<td>21</td>
<td>2,21</td>
<td>32</td>
<td>1,2,22,32</td>
</tr>
<tr>
<td>11</td>
<td>2,11</td>
<td>22</td>
<td>1,22</td>
<td>33</td>
<td>13,33</td>
</tr>
<tr>
<td>12</td>
<td>1,4,6,12</td>
<td>23</td>
<td>5,23</td>
<td>34</td>
<td>1,2,27,34</td>
</tr>
</tbody>
</table>
**Side-Stream Scrambler**

- Also called “frame-synchronized”
  \[ c_k = b_k \oplus x_k \]  (118)
  \[ c_k \oplus x_k = b_k \oplus x_k \oplus x_k = b_k \oplus 0 = b_k \]  (119)
- Advantage: no error propagation
- Disadvantage: need to synchronize scramblers
- Note that \( c_k \) would be all zeros if \( b_k = x_k \) (unlikely)

**Self-Synchronized Scrambler**

- Similar to side-stream, \( b_k \) recovered since \( y_k \oplus y_k = 0 \)
- Advantage: no need for alignment of scramblers.
- Disadvantage: one error in received value of \( c_k \) results in three errors (one for each XOR summation)
- Can also have more problems with periodic inputs.
Line Coding

Change pulse shape

- NRZ
- RTZ
- Biphas

- Remains a 2-level signal but more high-freq content

Filter data signal

Filter data signal

- Results in more signal levels than needed for bit transmission — “correlated level coding”
- Loose 3dB in performance unless maximal likelihood detector used.

Block Line Codes

- Map block of \( k \) bits into \( n \) data symbols drawn from alphabet of size \( L \).
- When \( 2^k < L^n \), redundancy occurs and can be used to shape spectrum.
- Example: blocks of 3 bits can be mapped to blocks of 2 3-level symbols.
**Hard-Decoding**

• Redundancy by adding extra bits
• Error detection and/or correction performed by looking *after* quantizer
• Examples: parity check, Reed-Solomon

**Soft-Decoding**

• Makes direct decisions on info bits without making intermediate decisions about transmitted symbols.
• Processes $Q_k$ directly — combines slicing and removal of redundancy
• Can achieve better performance than hard decoding
Partial-Response Motivation

Disadvantage — Feed-Forward Equalizer

- An FFE boosts the noise in areas where received signal power is low
- Example:

\[
A_k = \pm 1 \quad H_{te}(z) \quad \text{FFE} \quad H_1(z) \quad \text{output data} \quad \hat{A}_k = \pm 1
\]

- Noise is boosted at high frequencies.
Partial-Response Motivation

Disadvantage — Decision Feedback Equalizer

- A DFE does not make use of all the impulse response.

\[ H_{fe}(z) = 1 + 0.9z^{-1} + 0.1z^{-2} \]

\[ A_k = \pm 1 \]

\[ A_k = \hat{A}_k = \pm 1 \]

\[ H_2(z) = 0.9z^{-1} + 0.1z^{-2} \]

- Since \( \hat{A}_k = A_k \), impulse response is \( \delta(k) \)

- If single input, better to look for \( 1, 0.9, 0.1, 0, 0 \) than \( 1, 0, 0, 0, 0 \)

- Postcursor ISI may have significant signal power.

Partial-Response Motivation

- Rather than equalizing to a Nyquist pulse, equalize to a partial-response signal

- Equalize to \( 1 + z^{-1} \) in DFE example

- Less noise boost

- More of the impulse response used to determine transmitted signal

- Need to look at a string of received symbols rather than symbol-by-symbol detection — MLSD

- Disadvantage — extra complexity and may not recover full dynamic range loss
Nyquist Criterion for Zero ISI

- **Nyquist’s First Criterion** for zero ISI

\[ h(kT) = \begin{cases} 
1 & k = 0 \\
0 & k \neq 0
\end{cases} \]

\[ \sum_n H\left(\omega - \frac{2\pi n}{T}\right) = T \]

Minimum Bandwidth System with Zero ISI

\[ H(\omega) = \begin{cases} 
T & |\omega| \leq \pi/T \\
0 & \text{elsewhere}
\end{cases} \]

- A brickwall low-pass spectrum with a cutoff frequency of \(1/(2T)\) (“sinc” impulse response)

- However — impulse response decays at a rate of \(1/t\) due to the frequency discontinuity in \(H(f)\).

- Excessive ISI if *any* timing perturbation occurs
Non-Minimum Bandwidth System

- One way to overcome jitter problem is to use more than the minimum bandwidth.
- A popular class of non-minimum bandwidth solutions are — **Cosine Roll-Off Filters**
- Can still transmit and receive only one of two symbols.

- But are minimum bandwidth systems practical? **Yes.** — use **partial-response signaling.**

Partial-Response Signaling

- By relaxing the **zero-ISI criterion** of Nyquist, the maximum symbol rate of 2 symbols/hertz can be achieved.

- Allow a **controlled** amount of ISI by digitally FIR filtering the data — results in more signal levels.

- Three popular FIR filters:
  - $1 + z^{-1}$ duobinary - class 1 zero at $f_s/2$
  - $1 - z^{-1}$ dicode zero at dc
  - $1 - z^{-2}$ modified duobinary - class 4 zeros at dc, $f_s/2$
    (also called PR4 or PRIV)
**Duobinary (1+z⁻¹)**

- Impulse response decays at a rate of $1/t^2$ since $H(f)$ is continuous but its first derivative is not.
- However, it transmits signal power at dc.

**Dicode (1-z⁻¹)**

- Impulse response decays at a rate of $1/t$ due to the frequency discontinuity in $H(f)$.
- However, it does not transmit any signal power at dc.
Modified Duobinary \((1-z^{-2}) — \text{class 4}\)

\[ 1 - z^{-2} = (1 - z^{-1})(1 + z^{-1}) \]

- \(h(t)\) decays at a rate of \(1/t^2\) since \(H(f)\) is continuous.
- It does not transmit signal power at either dc or \(f_s/2\).

Class-4 Partial Response Signaling Scheme

- Spectral nulls at DC and \(f_s/2\)
- Can be encoded/decoded by two interleaved dicode encoder/decoder each operating at half the rate.

- Thus, \textit{we need only decode a dicode} and use two interleaved identical blocks to decode PRIV.
- If binary inputs, 3 level output — BPR4 or BPRIV
- (If 4 level inputs, 9 level output — QPR4 or QPRIV)
Magnetic Recording Similarities

- At low densities, a magnetic read signal is inherently 1-D encoded (i.e. a dicode).
  
  ![Example Waveform]

- At higher densities, high-frequency roll-off important (modelled as a Lorentzian pulse).
- If equalized to a 1-D channel, high-frequency noise is amplified.

- **Find a good approximation to channel so that the boost required by equalizer is kept small.**

Magnetic Recording Similarities

- Magnetic recording channel often modelled as Lorentzian pulse

![Frequency Response of a Lorentzian Magnetic Read Channel]
Magnetic Recording Similarities

- Similar to \((1 - z^{-1})(1 + z^{-1})^n\) partial-response channel.

\[
|H(w)| = \begin{cases} 
1 & \text{if } n=0 \\
0.5 & \text{if } n=1 \\
0.25 & \text{if } n=2 \\
0.125 & \text{if } n=5 
\end{cases}
\]

SNR Degradation for Dicode

- Now 3 levels being sent rather than just two.

- Thus, a bit-by-bit detection results in SNR performance degradation (about 2-3 dB loss).
- However, the 3 levels have some redundancy included.
- SNR performance can be recovered in detection by employing Maximum-Likelihood Sequence Estimation (MLSE) detection schemes
- The Viterbi Algorithm is an efficient way of realizing MLSE detection
### Trellis Introduction

- **Input**: 0, 1
- **Output**: ±1, 0

**State Diagram**

- States: 0/0, 0/-1, 1, 1/0, 1/+1

**Trellis Diagram**

- States: 0, 1, 0/0, 1/0, 1/+1, 0/-1
- Transitions:
  - k-1 to k
  - Data: 0, 1
  - Encoded Data: ±1, 0

### Trellis Representation of Dicode \((1-z^{-1})\)

- A trellis can be used to describe an encoder.
- Example:

**Trellis**

- States: 0, 1, 0/0, 1/0
- Transitions:
  - k-1 to k
  - Data: 0, 1
  - Encoded Data: ±1, 0

**Encoder**

- Input: 0, 1, 0
- Output: +1, 0, -1, +1, -1, 0
Transmit Trellis for Dicode (1-z⁻¹)

- Note that following a ‘+1’ output, there can be an arbitrary number of zeros followed by a ‘-1’.
- In other words, if two ‘+1’ symbols are detected with no ‘-1’ between them, an error occurred in transmission.
- Similar for a ‘-1’ output.

Conventional Bit-by-Bit Detection

- Note the error in bit 3 received.
- Error can be detected since a “-1” must be next non-zero symbol after a “+1”.
- Did the error most likely occur in symbol 2, 3 or 4?
Viterbi Algorithm (VA)

- VA is an iterative method for determining the most likely sequence sent — maximum likelihood detector.

- Accomplished by creating a receive trellis having branch metrics proportional to the difference squared between received signal and each ideal symbol value.

- The most likely sequence is the shortest path through the receive trellis.

- State metrics and path memory also stored to reduce search time through trellis — they are the length of shortest path and path taken at each node.

Viterbi Algorithm Example

Data 1 1 0 1 0 0
Encoded Data 1 0 -1 +1 -1 0
Channel 0.8 0.3 -0.4 0.9 -0.8 -0.1

large noise

Data Received 1 1 0 1 0 0
Encoded Data Received 0 +1 0 -1 +1 -1 0

University of Toronto
Viterbi Algorithm Example

Detailed Received Trellis Description (BPR4)

- Transmitted signal — one of three values, ±a, 0
- Received signal — $y_k$.

\[
\begin{align*}
\text{state metrics} & \rightarrow m_{0_{k-1}} & m_{0_k} \\
& \downarrow (y_k - 0)^2 & \\
& (y_k + a)^2 & (y_k - a)^2 \\
& \uparrow (y_k - 0)^2 & \text{branch metrics} \\
\text{state metrics} & \rightarrow m_{1_{k-1}} & m_{1_k}
\end{align*}
\]

- Equations:
\[
\begin{align*}
m_{0_k} &= \min\{ (m_{0_{k-1}} + y_k^2), (m_{1_{k-1}} + (y_k + a)^2) \} \\
m_{1_k} &= \min\{ (m_{1_{k-1}} + y_k^2), (m_{0_{k-1}} + (y_k - a)^2) \}
\end{align*}
\]
Simplifications to Remove Multiplications

- Remove $y_k^2$ terms since it occurs in both terms and we are only interested in finding the minimum path (don't need the absolute length of the path).
- State-metrics can now be either positive or negative.

\[
\begin{align*}
  m_{0_{k-1}} & \quad (y_k - 0)^2 \\
  (y_k + a)^2 & \quad (y_k - a)^2 \\
  m_{1_{k-1}} & \quad (y_k - 0)^2 \\
  m_{0_k} & \quad m_{0_k} \\
  m_{1_k} & \quad m_{1_k}
\end{align*}
\]

Simplifications to Remove Multiplications

- Divide all branches by $2a$ (assume $a > 0$)
- Simply scales state metrics.

\[
\begin{align*}
  m_{0_{k-1}} & \quad 0 \\
  (2ay_k + a^2) & \quad (-2ay_k + a^2) \\
  m_{1_{k-1}} & \quad 0 \\
  m_{0_k} & \quad m_{0_k} \\
  m_{1_k} & \quad m_{1_k}
\end{align*}
\]

- Equations:
\[
\begin{align*}
  m_0 &= \min\{m_{0_{k-1}}, (m_{1_{k-1}} + y_k + a)\} \\
  m_1 &= \min\{m_{1_{k-1}}, (m_{0_{k-1}} - y_k + a)\}
\end{align*}
\]
Difference Metric Algorithm

- [Wood and Peterson, Trans. on Comm., May 1986]
- We are not interested in absolute state-metric values — only which state-metric is smaller.
- Store only the difference in the state-metrics, $\Delta m_k$
  \[ \Delta m_k = m0_k - m1_k \]
- We shall see that while absolute state-metric values increase in time, their difference does not.
- This “difference metric algorithm” results in less complex realizations for both digital and analog realizations in cases where there are only two state-metrics.

Difference Metric Algorithm

- Subtract off $m1_{k-1}$ from input state-metrics and add it into each branch metric instead.
- Now, $m1_{k-1}$ can be subtracted off branch metrics since it is the same in all branches.
Difference Metric Algorithm

• Different path choices.

\[ \Delta m_{k-1} \]

\[ \begin{pmatrix} 0 \\ y_k + \frac{a}{2} \\ 0 \end{pmatrix} \]

\[ \begin{pmatrix} 0 \\ -y_k + \frac{a}{2} \end{pmatrix} \]

\[ \begin{pmatrix} \Delta m_{k-1} \leq y_k + \frac{a}{2} \) and \( \Delta m_{k-1} < y_k - \frac{a}{2} \) \]

\[ \Delta m_k = y_k - \frac{a}{2} \]

\[ \begin{pmatrix} 0 \\ m_0_k \end{pmatrix} \]

\[ \begin{pmatrix} m_1_k \end{pmatrix} \]

\[ \begin{pmatrix} \Delta m_{k-1} \geq y_k + \frac{a}{2} \) and \( \Delta m_{k-1} > y_k - \frac{a}{2} \) \]

\[ \Delta m_k = y_k + \frac{a}{2} \]
Difference Metric Algorithm

- Equations:
  \[
  \Delta m_k = \begin{cases} 
  y_k + \frac{a}{2} & \Delta m_{k-1} > y_k + \frac{a}{2} \\
  \Delta m_{k-1} & \frac{a}{2} \leq \Delta m_{k-1} < y_k + \frac{a}{2} \\
  y_k - \frac{a}{2} & \Delta m_{k-1} < y_k - \frac{a}{2}
  \end{cases}
  \]

- These equations describe an adjustable threshold device.
- Used in digital PR4 implementations.
- They are also simple to implement in analog.

Typical Digital Implementation

- 6-bit flash A/D requires 63 comparators + decoding logic.
- A/D converter might consume around 300mW (or more)
- FIR equalizer might be 1mW/MHz/tap — 8-tap at 100MHz = 800mW
**Typical Digital Implementation**

- Digital equalizer requires multi-bit multiplies in feed-forward equalizer (power hungry)
- If decision feedback is used, it will need to use early estimates of output (cannot wait for MLSE to finish).
- Digital difference algorithm requires some minor adders and digital comparators.
- Digital path memory logic consists of about 16 serial/parallel shift registers.

**Typical Analog Implementation**

- Analog equalizer needed (less power than digital but more challenging)
- Digital path memory is the same as in fully digital realization.
Typical Analog Implementation

- Analog difference algorithm is very small (about the size of 2 comparators)
- Thus, power is saved and speed can be increased over 6-bit A/D converter.
- Note that dynamic range in analog parts need only be around 6 bits (i.e. 40dB)

Analog Implementation

- Path memory consists of two serial/parallel in/out shift registers.

0 0 State "0"
1 1 State "1"
Some Practical Limitations

• In a digital implementation, performance is degraded by limiting the number of bits used in A/D conversion

• Typically use about 6-bit A/D converters (easily achievable in an all analog implementation).

• Truncating the trace-back length (path memory) also degrades the performance.

• Typically use length of 16 for little loss in performance.

Why an Analog Implementation?

• Avoids using a pre-stage A/D converter.
• Combines the A/D and VA into one stage with a complexity near to a 2-bit A/D (Special-purpose A/D converter).
• Consumes less power.
• Operates faster.
• 6-bit accuracy is enough (Moderate Precision Circuitry).
• Low-dynamic range requirement (Low-Voltage Operation).
• The difference algorithm updates only one sampled data without using previous samples (no accumulative analog errors)
**Simulation and Experimental Results**

- Simulation and experimental (discrete prototype) results confirm validity of the analog approach and its robustness against imperfections.

![Graphs showing Bit Error Rate (BER) vs. Signal to Noise Ratio (SNR) with different DC offsets and gain mismatches.](image)

Offsets and mismatches are described in percentage of $\alpha$.

---

**Input-Interleaved Algorithm**

- The implementation above updates $\Delta m_k$ once the comparator outputs are known.

- Thus, **critical speed path is 2 sample-and-holds**.
  (Sample input and compare, then, perhaps, update $\Delta m_k$ with another sample-and-hold).

- The input-interleaved algorithm reduces the critical speed path to a single sample-and-hold (i.e. can operate at twice the speed).

- It uses two sample-and-holds at the input and switches which one the input goes to if $\Delta m_k$ needs to be updated.
**Input Interleaved Algorithm**

- According to the update mechanism, $\Delta m_{k-1}$ is a DC-shifted version of a previously-sampled input signal

  $$\Delta m_{k-1} = y_{k-j} \pm 0.5$$

- We can use the previously held input signal plus appropriate sign of DC shift for $\Delta m_k$.

- When $\Delta m_k$ needs to be updated, switch input on to other sample-and-hold capacitor and use the just sampled input and a sign-bit for new $\Delta m_k$.

**Input-Interleaved Algorithm**

- Toggle between two S/Hs which store $y_k$ and $y_{k-j}$

- Use a flip-flop to properly switch the DC signal

- Speed improvement, as no additional S/H is required ($y_{k-j}$ has already been stored)
BiCMOS Integrated-Circuit Implementation
BiCMOS Integrated-Circuit Implementation

- Path memory consists of 2x12 multiplexed-input D flip-flops

```
D D ... D
```

State "0"

```
D D ... D
```

State "1"

- Clock phases

BiCMOS Integrated-Circuit Implementation

- Compared to other analog implementations
  [Matthews and Spencer, JSSC, Dec. 93]

  - Less complex (Individual state metrics are not calculated)
  - Less prone to imperfections (Feedback signals are only digital)
  - Fully differential
  - Faster (Master-slave S/Hs are not used)

[Yamasaki, ISSCC, 1994]

  - No details given
Integrated-Circuit Implementation

Experimental Results

- Process: 0.8 μm BiCMOS
- Area (dicode): ~0.25 mm²
  - Analog: ~0.06 mm²
  - Digital: ~0.1 mm²
  - Bypass capacitors, ...
- Power consumption (dicode):
  - 3.3V power supply
  - ~12mW at 50MHz
  - ~15mW at 100MHz
Experimental Results

• Setup

Experimental Results

• Measured Bit-Error-Rate (BER) performance

A General Implementation Approach

• Analog implementations are useful if the preceding signal processing is simple
  • Magnetic recording
  • Data transmission over unshielded cables
• Simplifications are only possible in some special cases (i.e. PR4)
• This general approach can be used in
  • More general PRS schemes (i.e. EPR4, EEPR4)
  • Convolutional codes
  • Multi-level digital communication
  • Irregular trellises
A General Implementation Approach

- This approach takes full advantage of the ability of simple analog circuits in realizing the ACS function

\[ m_i(k) = \text{Max}_j \{ m_j(k-1) - e_{ji}(k) \} \quad i = 1, 2, \ldots, N \]
\[ j = 1, 2, \ldots, M \]

- Branch metrics, \( e_{ji} \), are usually expressed in terms of linear combinations of the received samples and DC signals

Circuit Realization

- A generalized differential cell is employed to realize the ACS function

- Using degenerated differential pairs in V/I conversions makes the linear combinations simple to realize
- Optional DC currents added to the error signals reduce the unnecessary DC voltage drops across the resistors
Circuit Realization

- Branch currents in the differential cells are comparison results.
- To achieve high speeds, ping-pong S/Hs are preferred to master-slave S/Hs in feeding the state metrics back.
- Algorithmic growth of state metrics is overcome by a fast Common-Mode Feedback (CMFB) circuit.
- Fast CMFB minimizes the signal swings of the state metrics — this approach is usually not practical in digital realizations.

Design Example: Binary Dicode

The diagram shows a binary dicode circuit with various biases and input signals.
Integrated Circuit Implementation

• A chip containing Viterbi decoders for a binary dicode and an Extended PR4 (EPR4, \((1-D)(1+D)^2\)) has been fabricated in a 0.8\(\mu\)m BiCMOS process

• Based on simulations, fast speed (>100 MHz) can be achieved with \(~15\text{mW/state}\) (Excluding path memory)

• The area is \(~0.03\text{mm}^2/\text{state}\) (Excluding path memory)

• In a CMOS implementation, lower \(g_m\) causes some degradation in:
  - Obtaining simple low-impedance nodes
  - ACS performance due to the high dependency of \(v_{GS}\) to the drain current

---

Integrated Circuit Implementation

- MVJ\(n^+\) I 50 \(\Omega\) Drivers
- EPR4
- 50 \(\Omega\) Drivers
- Dicodex
- Test Dicodex
- Clock Generator
- 0.1mm
Preliminary Experimental Results

- Results on dicode

- High-frequency tests have been conducted up to 80 MHz (Off-chip path memory)

Summary

- The use of partial-response signals allows one to send closer to the maximum rate of 2 symbols/hertz.
- Making use of partial-response signalling reduces the need for large equalization boost.
- The difference algorithm is efficient for PR4 signals
- Analog realizations of the difference algorithm save silicon and power over digital realizations (however, an analog equalizer is needed)
- Input-interleaved algorithm increases the speed for an analog implementation
- A general analog Viterbi approach was discussed. (However, it makes use of bipolar transistors).
Infrared Channels

**Advantages**
- Free from regulation, low cost
- Blocked by walls — reduces eavesdropping and inter-cell interference
- Abundance of bandwidth in directed line-of-sight
- **Disadvantages** — lower range and bandwidth than radio

**Types of Links**
- Line-of-sight (does not rely on reflections) — higher bandwidth, easily blocked
- Diffuse (disperse Tx and wide-angle Rx) — lower bandwidth, very tolerant to shadowing
Infrared Channels

**Transmit LED**
- GaAs LED emission match peak of silicon photodiode sensitivity (850 nm wavelength)
- 10-20% efficient — transmit current up to 0.5A
- Laser diodes also used — 30-70% efficient
- Laser diodes must be rendered eye-safe — diffuser

**Receive Photodiode**
- Reverse-biased — light creates electron-hole pairs in depletion region
- R is responsivity of diode — 0.6 implies 60% of photons collected result in current flow
- Optical filters and concentrators can be used (reduces required size of photodiode)
- Usually off-chip photodiode used since received power is proportional to photodiode area
- Large photodiode results in large diode capacitance
- Large input dynamic range (perhaps 100 dB) required.
Infrared Channels

Eye safety
- Need to limit optical power for eye safety
- Optical power proportional to Tx signal current
- Receive signal current proportional to optical power
- Optical power proportional to square of receive signal power
- Makes design different than conventional channels

Example
- 10 dB loss in optical power (100μW → 10μW)
- 20 dB loss in Rx current (100nA → 10nA)
- If noise remains unchanged, twice dB loss
- Reason for low range operation

Noise
- Ambient light typically much larger than infrared light resulting in a large dc bias Rx current
- Main source of noise — shot noise from ambient light on photodiode
- Modeled as white but can be shaped by preamp (increases at higher freq)
- Independent of Rx signal
- Also noise from fluorescent lamp ballasts — 20kHz and harmonics
- Fluorescent lamp noise likely to become worse as ballast frequencies increase
Diffuse Channel

- Infrared behaves same as visible light
- No worry about multipath fading since wavelength is so small.
- Multipath dispersion does exist and limits channels to 10-50MHz
- Most reflective walls are modelled as Lambertian reflectors — incident light re-radiated in all directions
- Results in multipath that is hard to shadow
- Results in time-dispersion (i.e. lowpass filtering)
- Use an LED with spatial dispersion
- Use photodiode with optical filter + concentrator

Diffuse Channel

- Typical room response

  - Diffuse system with no LOS
  - Rolls off steadily at high frequencies
Pulse-Position Modulation

- In 1993, IrDA (Infrared Data Association) formed
- 4.0 Mb/s standard uses 4-PPM

- Type of orthogonal modulation
- In general, L-PPM has each symbol having L time slots
- Power transmitted in one time slot and zero otherwise

---

Advantage
- Average power requirement decreases with increasing L

Disadvantages
- Higher bandwidth required
- Increased peak-power requirement
- Requires both time-slot and symbol-level synchronization

Soft-Decoding
- Choose largest of L samples

Hard-Decoding
- Each sample quantized to 0 or 1 (1.5 dB penalty)
**Different Modulation Schemes**

- For optical channel (intensity modulation)

![Graph showing normalized power and bandwidth requirements for different modulation schemes relative to 2-PAM.](image)

**IR Systems**

**Present Systems**

- 4Mb/s 4-PPM over LOS systems
- dc rejection to combat fluorescent lighting noise
- Little equalization
- Silicon PIN diode roughly 1 cm²

**Future Trends**

- Lower cost laser diodes with diffusers
- PAM modulation for higher data-rates
- 50Mb/s diffuse system demonstrated but not integrated (2-PAM)
- DFE and Max Likelihood Sequence Detectors (MLSD) used to combat intersymbol interference
## Analog Equalization

### Analog Filters

**Switched-capacitor filters**
- Accurate transfer-functions
- High linearity, good noise performance
- Limited in speed
- Requires anti-aliasing filters

**Continuous-time filters**
- Moderate transfer-function accuracy (requires tuning circuitry)
- Moderate linearity
  + High-speed
  + Good noise performance
Adaptive Linear Combiner

- The gradient signals are simply the state signals
- If coeff are updated in discrete-time

\[ p_i(n+1) = p_i(n) + 2\mu e(n)x_i(n) \] (120)

- If coeff are updated in cont-time

\[ p_i(t) = \int_0^\infty 2\mu e(t)x_i(t)dt \] (121)

- Only the zeros of the filter are being adjusted.
- There is no need to check that for filter stability (though the adaptive algorithm could go unstable if \( \mu \) is too large).
Adaptive Linear Combiner

- The performance surface is guaranteed unimodal (i.e. there is only one minimum so no need to worry about being stuck in a local minimum).
- The performance surface becomes ill-conditioned as the state-signals become correlated (or have large power variations).

Analog Adaptive Linear Combiner

- Better to use input summing rather than output summing to maintain high speed operation
- Requires extra gradient filter to obtain gradients

Analog Adaptive Filters

Analog Equalization Advantages

- Can eliminate A/D converter
- Reduce A/D specs if partial equalization done first
- If continuous-time, no anti-aliasing filter needed
- Typically consumes less power and silicon for high-frequency low-resolution applications.

Disadvantages

- Long design time (difficult to “shrink” to new process)
- More difficult testing
- DC offsets can result in large MSE (discussed later).
Analog Adaptive Filter Structures

- Tapped delay lines are difficult to implement in analog.

**To obtain uncorrelated states:**

- Can use Laguerre structure — cascade of allpass first-order filters — poles all fixed at one location on real axis

- For arbitrary pole locations, can use orthonormal filter structure to obtain uncorrelated filter states [Johns, CAS, 1989].

---

**Orthonormal Lader Structure**

- For white noise input, all states are uncorrelated and have equal power.
**Analog’s Big Advantage**

- In digital filters, programmable filter has about same complexity as a fixed filter (if not power of 2 coeff).
- In analog, arbitrary fixed coeff come for free (use element sizing) but programming adds complexity.
- In continuous-time filters, frequency adjustment is required to account for process variations — relatively simple to implement.
  - *If channel has only frequency variation — use arbitrary fixed coefficient analog filter and adjust a single control line for frequency adjustment.*
- Also possible with switched-C filter by adjusting clock frequency.

---

**Analog Adaptive Filters**

- Usually digital control desired — can switch in caps and/or transconductance values
- Overlap of digital control is better than missed values

- In switched-C filters, some type of multiplying DAC needed.
- Best fully-programmable filter approach is not clear
Analog Adaptive Filters — DC Offsets

- DC offsets result in partial correlation of data and error signals (opposite to opposite DC offset)

\[
x_i(k) \xrightarrow{\Sigma} m_{xi} \xrightarrow{\times} \Sigma \xrightarrow{\triangle} w_i(k)
\]

- At high-speeds, offsets might even be larger than signals (say, 100 mV signals and 200mV offsets)
- DC offset effects worse for ill-conditioned performance surfaces

- Sufficient to zero offsets in either error or state-signals (easier with error since only one error signal)
- For integrator offset, need a high-gain on error signal
- Use **median-offset cancellation** — slice error signal and set the median of output to zero
- In most signals, its mean equals its median

- Experimentally verified (low-frequency) analog adaptive with DC offsets more than twice the size of the signal.
**DC Offset Effects for LMS Variants**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>LMS</th>
<th>SD-LMS</th>
<th>SE-LMS</th>
<th>SS-LMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>input power</td>
<td>$\sigma_e^2 \propto 1/\sigma_x^2$</td>
<td>no effect</td>
<td>$\sigma_e^2 \propto 1/</td>
<td>\mu</td>
</tr>
<tr>
<td>no offsets</td>
<td>$\sigma_e^2 \rightarrow 0$ for $\mu \rightarrow 0$</td>
<td>$\sigma_e^2 \rightarrow 0$ for $\mu \rightarrow 0$</td>
<td>$\sigma_e^2 \propto \mu^2 \sigma_x^2$</td>
<td>$\sigma_e^2 \propto \mu^2 \sigma_x^2$</td>
</tr>
<tr>
<td></td>
<td>$\sigma_e^2$ weakly depends on $\mu$</td>
<td>$\sigma_e^2$ strongly depends on $\mu$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>algorithm complexity</td>
<td>1 multiplier/tap</td>
<td>1 slicer/tap</td>
<td>1 trivial multiplier/tap</td>
<td>1 slicer/tap</td>
</tr>
<tr>
<td></td>
<td>1 integrator/tap</td>
<td>1 integrator/tap</td>
<td>1 integrator/tap</td>
<td>1 slicer/filter</td>
</tr>
<tr>
<td></td>
<td>1 XOR gate/tap</td>
<td>1 counter/tap</td>
<td>1 DAC/tap</td>
<td>1 slicer/filter</td>
</tr>
<tr>
<td>convergence</td>
<td>no gradient misalignment</td>
<td>gradients misaligned</td>
<td>no gradient misalignment</td>
<td>gradients misaligned</td>
</tr>
</tbody>
</table>

**Coax Cable Equalizer**

- Analog adaptive filter used to equalize up to 300m
- Cascade of two 3’rd order filters with a single tuning control

$$w_1 s/ (s + p_1)$$
$$w_2 s/ (s + p_2)$$
$$w_3 s/ (s + p_3)$$

Highpass filters

- Variable $\alpha$ is tuned to account for cable length
Coax Cable Equalizer

- Equalizer optimized for 300m
- Works well with shorter lengths by tuning $\alpha$
- Tuning control found by looking at slope of equalized waveform
- Max boost was 40 dB
- System included dc recovery circuitry
- Bipolar circuit used — operated up to 300Mb/s

Analog Adaptive Equalization Simulation

- Channel modelled by a 6'th-order Bessel filter with 3 different responses — 3MHz, 3.5MHz and 7MHz
- 20Mb/s data
- PR4 generator — 200 tap FIR filter used to find set of fixed poles of equalizer
- Equalizer — 6'th-order filter with fixed poles and 5 zeros adjusted (one left at infinity for high-freq roll-off)
**Analog Adaptive Equalization Simulation**

- Analog blocks simulated with a 200MHz clock and bilinear transform.
- Switch S1 closed (S2 open) and all poles and 5 zeros adapted to find a good set of fixed poles.
- Poles and zeros depicted in digital domain for equalizer filter.
- Residual MSE was -31dB

**Equalizer Simulation — Decision Directed**

- Switch S2 closed (S1 open), all poles fixed and 5 zeros adapted using
  - \( e(k) = 1 - y(t) \) if \( y(t) > 0.5 \)
  - \( e(k) = 0 - y(t) \) if \(-0.5 \leq y(t) \leq 0.5\)
  - \( e(k) = -1 - y(t) \) if \( y(t) < -0.5 \)
- all sampled at the decision time — assumes clock recovery perfect
- Potential problem — AGC failure might cause \( y(t) \) to always remain below \( \pm 0.5 \) and then adaptation will force all coefficients to zero (i.e. \( y(t) = 0 \)).
- Zeros initially mistuned to significant eye closure
Equalizer Simulation — Decision Directed

• 3.5MHz Bessel

Equalizer Simulation — Decision Directed

• Channel changed to 7MHz Bessel
• Keep same fixed poles (i.e. non-optimum pole placement) and adapt 5 zeros.

• Residual MSE = -29dB
• Note that no equalizer boost needed at high-freq.
Equalizer Simulation — Decision Directed

- Channel changed to 3MHz Bessel
- Keep same fixed poles and adapt 5 zeros.
- Residual MSE = -25dB
- Note that large equalizer boost needed at high-freq.
- Probably needs better equalization here (perhaps move all poles together and let zeros adapt)

BiCMOS Analog Adaptive Filter Example

- Demonstrates a method for tuning the pole-frequency and Q-factor of a 100MHz filter — adaptive analog
- Application is a pulse-shaping filter for data transmission.
- One of the fastest reported integrated adaptive filters — it is a Gm-C filter in 0.8um BiCMOS process
- Makes use of MOS input stage and translinear-multiplier for tuning
- Large tuning range (approx. 10:1)
- All analog components integrated (digital left off)
**BiCMOS Transconductor**

Two styles implemented:
- "2-quadrant" tuning (F-Cell),
- "4-quadrant" tuning by cross-coupling top input stage (Q-Cell)

**Biquad Filter**

- $f_0$ and $Q$ not independent due to finite output conductance
- Only use 4-quadrant transconductor where needed
**Experimental Results Summary**

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transconductor (T.) size</td>
<td>0.14mm x 0.05mm</td>
</tr>
<tr>
<td>T. power dissipation</td>
<td>10mW @ 5V</td>
</tr>
<tr>
<td>Biquad size</td>
<td>0.36mm x 0.164mm</td>
</tr>
<tr>
<td>Biquad worst case CMRR</td>
<td>20dB</td>
</tr>
<tr>
<td>Biquad $f_c$ tuning range</td>
<td>10MHz-230MHz @ 5V, 9MHz-135MHz @ 3V</td>
</tr>
<tr>
<td>Biquad $Q$ tuning range</td>
<td>1-Infinity</td>
</tr>
<tr>
<td>Bq. inpt. ref. noise dens.</td>
<td>$0.21 V_{PpS} / \sqrt{H}$</td>
</tr>
<tr>
<td>Biquad PSRR+</td>
<td>28dB</td>
</tr>
<tr>
<td>Biquad PSRR-</td>
<td>21dB</td>
</tr>
<tr>
<td>Filter Setting</td>
<td>Output 3rd Order Intercept Point</td>
</tr>
<tr>
<td>100MHz, $Q = 2$, Gain = 10.6dB</td>
<td>23dBm</td>
</tr>
<tr>
<td>20MHz, $Q = 2$, Gain = 30dB</td>
<td>20dBm</td>
</tr>
<tr>
<td>100MHz, $Q = 15$, Gain = 29.3dB</td>
<td>18dBm</td>
</tr>
<tr>
<td>227MHz, $Q = 35$, Gain = 31.7dB</td>
<td>10dBm</td>
</tr>
</tbody>
</table>

**Adaptive Pulse Shaping Algorithm**

- **Fo control**: sample output pulse shape at nominal zero-crossing and decide if early or late (cutoff frequency too fast or too slow respectively)
- **Q control**: sample bandpass output at lowpass nominal zero-crossing and decide if peak is too high or too small (Q too large or too small)
Experimental Setup

- Off-chip used an external 12 bit DAC.
- Input was 100Mb/s NRZI data 2Vpp differential.
- Comparator clock was data clock (100MHz) time delayed by 2.5ns

Pulse Shaper Responses

- Initial — high-freq. high-Q
- Initial — high-freq. low-Q
- Initial — low-freq. high-Q
- Initial — low-freq. low-Q
Summary

- Adaptive filters are relatively common
- LMS is the most widely used algorithm
- Adaptive linear combiners are almost always used.
- Use combiners that do not have poor performance surfaces.
- Most common digital combiner is tapped FIR

Digital Adaptive:
- more robust and well suited for programmable filtering

Analog Adaptive:
- best suited for high-speed, low dynamic range.
- less power
- very good at realizing arbitrary coeff with frequency only change.
- Be aware of DC offset effects

References

General References

Information Theory and Capacity

Adaptive Filters and Equalization
A. Shoal, W.M. Snelgrove and D.A. Johns, “A 100Mb/s BiCMOS adaptive pulse-shaping filter”, accepted for publication in *IEEE Journal on Selected Areas in Communications: Special issue on Copper Wire Access Technologies for High Performance Networks*.

**Wired Channels**


**Timing Recovery**


**Partial-Response and Viterbi Detection**

Infrared