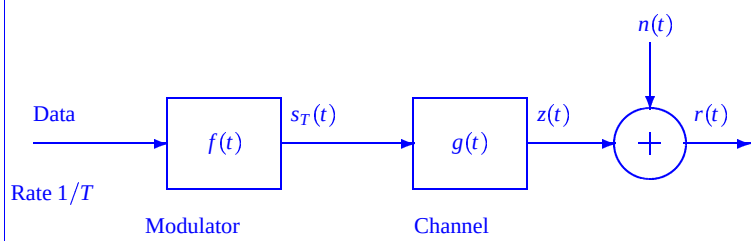
Lecture Notes 9: Intersymbol Interference

In this lecture we examine optimum demodulation when the transmitted signal is filtered by the channel and there is additive white Gaussian noise. The optimum demodulator chooses the possible transmitted vector that would result in the received vector (in the absence of noise) to be as close as possible (in Euclidean distance) to what was received. This we show can be implemented by a filter matched to the received signal for a given data symbol followed by a nonlinear processing via the Viterbi algorithm. The filter is sampled at the data rate. We also analyze the performance of such a system. The analysis is very similar to that of convolutional codes. Because the received signal is filtered and sampled, the output of the filter consists of two components. One due to the transmitted signal and one due to the noise. The output due to noise is, however, not white. However, in the next section we show that the output of the matched filter can be whitened. With a whitened matched filter the optimum receiver (Viterbi algorithm) becomes clear. Finally, in the last section we show how to design a system to eliminate intersymbol interference.

Optimum Demodulation

Consider transmitting data at rate $1/T$ through a channel with bandwidth W or through a distorting channel. We would like to find the optimum (minimum sequence error probability) receiver. Assume the modulator is a filter acting on a infinite sequence of impulses (at rate $1/T$ with impulse response $f(t)$. The channel is characterized by an impulse response of $g(t)$ and the receiver is a filter sampled at rate $1/T$ with impulse response $h(t)$.



The transmitted signal is of the form

$$s_{T,\mathbf{u}}(t) = \sum_{m=-N}^{N} u_m f(t-mT)$$

where u_m is the data symbol transmitted during the m-th signaling interval assumed to be in the alphabet A and $f(t)$ is the waveform used for transmission. We assume a transmission of $2N+1$ data symbols (think of N as being very large). The output of the channel filter is then

$$\begin{aligned} z(t) &= \int_{-\infty}^{\infty} g(t-\tau)s(\tau)d\tau \\ &= \int_{-\infty}^{\infty} g(t-\tau) \sum_{m=-N}^{N} u_m f(\tau-mT)d\tau \\ &= \sum_{m=-N}^{N} u_m \int_{-\infty}^{\infty} g(t-\tau) f(\tau-mT)d\tau \\ &= \sum_{m=-N}^{N} u_m h(t-mT) \end{aligned}$$

where

$$h(t) = \int_{-\infty}^{\infty} g(t-\tau) f(\tau)d\tau$$

The received signal consist of two terms. One due to signal and one due to noise.

$$\begin{aligned} r(t) &= \sum_{m=-N}^{N} u_m h(t-mT) + n(t) \\ &= s_{\mathbf{u}}(t) + n(t) \end{aligned}$$

where

$$s_{\mathbf{u}}(t) = \sum_{m=-N}^{N} u_m h(t-mT)$$

Since $n(t)$ is white Gaussian noise the optimum receiver computes for each data sequence $\mathbf{v}$

$$\begin{aligned} \Lambda_N(\mathbf{v}) &= 2(r(t), s_{\mathbf{v}}(t)) - \|s_{\mathbf{v}}\|^2 \\ &= 2\int_{-\infty}^{\infty} r(t)s_{\mathbf{v}}(t)dt - \int_{-\infty}^{\infty} s_{\mathbf{v}}^2(t)dt \\ &= 2\int_{-\infty}^{\infty} r(t) \sum_{k=-N}^{N} v_k h(t-kT)dt - \int_{-\infty}^{\infty} \sum_m \sum_k v_m v_k h(t-kT)h(t-mT)dt \\ &= 2\sum_{k=-N}^{N} v_k \int_{-\infty}^{\infty} r(t)h(t-kT)dt - \sum_m \sum_k v_m v_k \int_{-\infty}^{\infty} h(t-kT)h(t-mT)dt \\ &= 2\sum_{k=-N}^{N} v_k y_k - \sum_{k=-N}^{N} \sum_{m=-N}^{N} v_k v_m x_{m-k} \end{aligned}$$

where

$$y_k = \int_{-\infty}^{\infty} r(t)h(t-kT)dt$$

$$x_{m-k} = \int_{-\infty}^{\infty} h(t-kT)h(t-mT)dt$$

Thus the optimum decision rule is

$$\text{Chose } \mathbf{v} \text{ if } \Lambda_N(\mathbf{v}) = \max_{\mathbf{u}} \Lambda_N(\mathbf{u}).$$

Since the decision statistic depends on the received signal only through y_k it is clear that y_k is a sufficient statistic to implement optimal receiver.

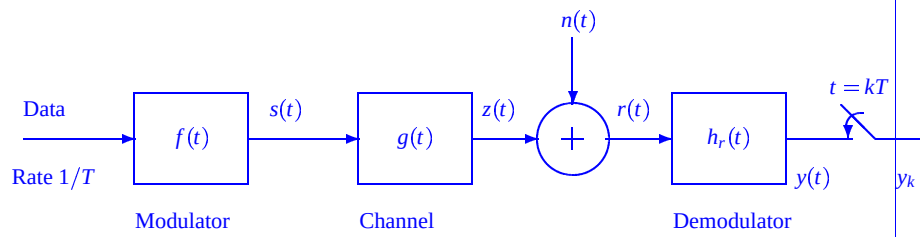
Consider a filter $h_r(t) = h(-t)$. Then if the received sequence is passed through this filter the output would be

$$\begin{aligned} y(s) &= \int_{-\infty}^{\infty} r(t)h_r(s-t)dt \\ &= \int_{-\infty}^{\infty} r(t)h(t-s)dt \end{aligned}$$

The sampled output would be

$$y(kT) = \int_{-\infty}^{\infty} r(t)h(t-kT)dt$$

Since this is y_k defined earlier the received signal should be filtered and sampled as shown below before doing some processing.

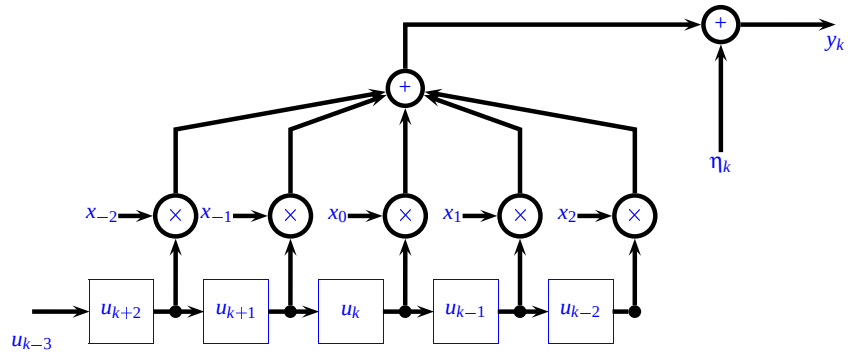


$$\begin{aligned} y_k &= \int_{-\infty}^{\infty} r(t)h(t-kT)dt = \int_{-\infty}^{\infty} \sum_{k=-N}^{N} u_m h(t-mT)h(t-kT)dt + \eta_k \\ \eta_k &= \int_{-\infty}^{\infty} n(t)h(t-kT)dt. \end{aligned}$$

Now the original continuous time detection problem can be replaced with a discrete time problem.

$$y_k = \sum_{m=-N}^{N} u_m x_{m-k} + \eta_k.$$

Model



Model

Note that η_k is Gaussian.

$$\begin{aligned}
E[\eta_k] &= 0 \\
Var[\eta_k] &= \frac{N_0}{2}\int_{-\infty}^{\infty} h^2(t)dt \\
E[\eta_k\eta_m] &= E\left[\int_{-\infty}^{\infty} n(t)h(t-kT)dt \int_{-\infty}^{\infty} n(s)h(s-mT)dt\right] \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(t-kT)h(s-mT)E\left[n(t)n(s)\right]dtds \\
&= \frac{N_0}{2}\int_{-\infty}^{\infty} h(t-kT)h(t-mT)dt = \frac{N_0}{2}x_{k-m}.
\end{aligned}$$

However, η_k is not an i.i.d. sequence.

Optimum Receiver

$$\begin{aligned}
\Lambda_N(\mathbf{v}) &= 2\sum_{k=-N}^{N} v_k y_k - \sum_{k=-N}^{N}\sum_{m=-N}^{N} v_k v_m x_{m-k} \\
&= 2\sum_{k=-N}^{N} v_k y_k - \sum_{k=-N}^{N} x_0|v_k|^2 - 2\sum_{k=-N}^{N} v_k \sum_{m=-N}^{k-1} v_m x_{m-k} \\
&= 2\sum_{k=-N}^{N} v_k y_k - \sum_{k=-N}^{N} x_0 v_k^2 - 2\sum_{k=-N}^{N} v_k \sum_{j=1}^{k+N} v_{k-j}x_{-j} \\
&= \sum_{k=-N}^{N} 2v_k y_k - \sum_{k=-N}^{N} x_0 v_k^2 - 2\sum_{k=-N}^{N} v_k \sum_{j=1}^{k+N} v_{k-j}x_j
\end{aligned}$$

(Note that $v_j = 0$, $j < -N$). Assume $x_m = 0$, $|m| > L$ (finite intersymbol interference)

$$\begin{aligned}
\Lambda_N(\mathbf{v}) &= 2\sum_{k=-N}^{N} v_k y_k - \sum_{k=-N}^{N} x_0 v_k^2 - 2\sum_{k=-N}^{N} v_k \sum_{j=1}^{\min(L,k+N)} v_{k-j}x_j \\
&= \sum_{k=-N}^{N}\left[2v_k y_k - x_0 v_k^2 - 2v_k \sum_{j=1}^{\min(L,k+N)} v_{k-j}x_j\right]
\end{aligned}$$

The decision rule can be implemented in a Viterbi algorithm like structure. Define the state at time k to be the last L data symbols. (These are the only symbols that affect the output at time k).

$$\sigma_k = (v_{k-L+1}, \ldots, v_k).$$

Let

$$\lambda(\sigma_k, \sigma_{k+1}) = 2v_{k+1}y_{k+1} - x_0 v_{k+1}^2 - 2v_{k+1}\sum_{m=1}^{L} v_{k+1-m}x_m$$

Then

$$\Lambda_N(\mathbf{v}) = \sum_{k=-N}^{N-1} \lambda(\sigma_k, \sigma_{k+1}) + 2v_{-N}y_{-N} - x_0 v_{-N}^2$$

Thus we can apply the Viterbi algorithm.

Viterbi Algorithm

Let $\Gamma(\sigma_m)$ be the length (optimization criteria) of the shortest (optimum) path to state σ_m at time m. Let $\Omega(\sigma_m)$ be the shortest path to state σ_m at time m. Let $\hat{\Gamma}(\sigma_{m+1}, \sigma_m)$ be the length of the path to state σ_{m+1} at time m that goes through state σ_m at time m. The algorithm works as follows.

Storage:

k, time index, $\hat{\Omega}(\sigma_m)$, $\sigma_m \in A^L$ $\Gamma(\sigma_m)$, $\sigma_m \in A^L$

Initialization

$k = -N$,

$\hat{\Omega}(\sigma_{-N}) = \sigma_{-N}$, $\sigma_{-N} \in \Phi^{L-1} \times A$ (Φ is the empty set).

$\Gamma(\sigma_{-N}) = 2v_{-N}y_{-N} - x_0 v_{-N}^2$

Recursion

$\hat{\Gamma}(\sigma_{m+1}, \sigma_m) = \Gamma(\sigma_m) + \lambda(\sigma_m, \sigma_{m+1})$

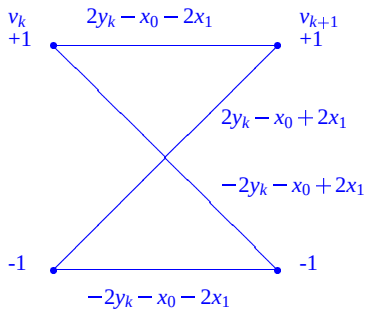
$\Gamma(\sigma_{m+1}) = \max_{\sigma_m} \hat{\Gamma}(\sigma_{m+1}, \sigma_m)$ for each σ_{m+1}

Let $\hat{\Omega}_m(\sigma_{m+1}) = \operatorname{argmax}_{\sigma_m} \hat{\Gamma}(\sigma_{m+1}, \sigma_m)$. $\hat{\Omega}(\sigma_{m+1}) = \hat{\Omega}(\sigma_m, \sigma_{m+1})$

Example: $L = 1$, $x_0 \neq 0$, $x_1 \neq 0$, $x_2 = 0$, $v_k \in \{\pm 1\}$

$$\Lambda_N(\mathbf{v}) = \sum_{k=-N}^{N} \lambda(\sigma_k, \sigma_{k+1}) + 2y_{-N}v_{-N} - x_0 v_{-N}^2$$

$$\lambda(\sigma_k, \sigma_{k+1}) = 2v_{k+1}y_{k+1} - x_0 v_{k+1}^2 - 2v_{k+1}v_k x_1$$



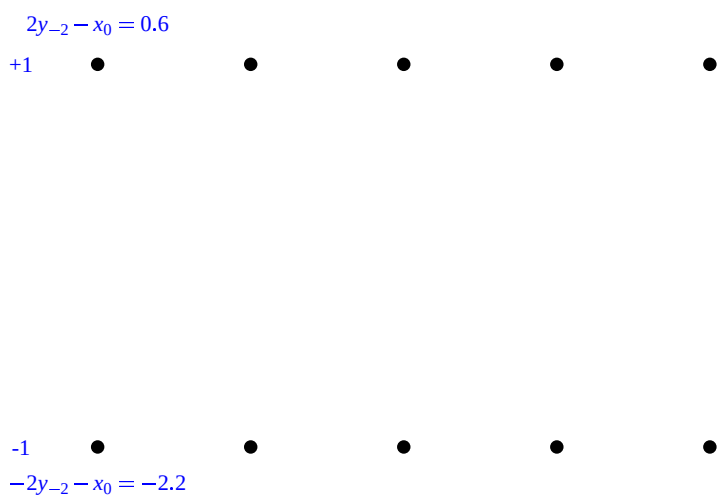
Example

Consider a channel with $x_0 = 0.8, x_1 = 0.2$. Consider the following received sequence of length 5 ($N = 2$).

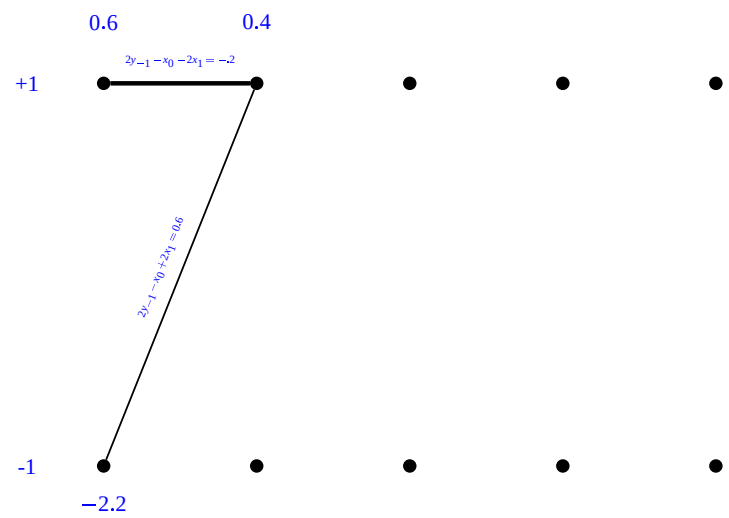
$$\mathbf{y} = (y_{-2}, y_{-1}, y_0, y_1, y_2) = (0.7, 0.5, -0.9, 0.3, -0.6)$$

Assume $v_k \in \{\pm 1\}$. Then the trellis is shown below.

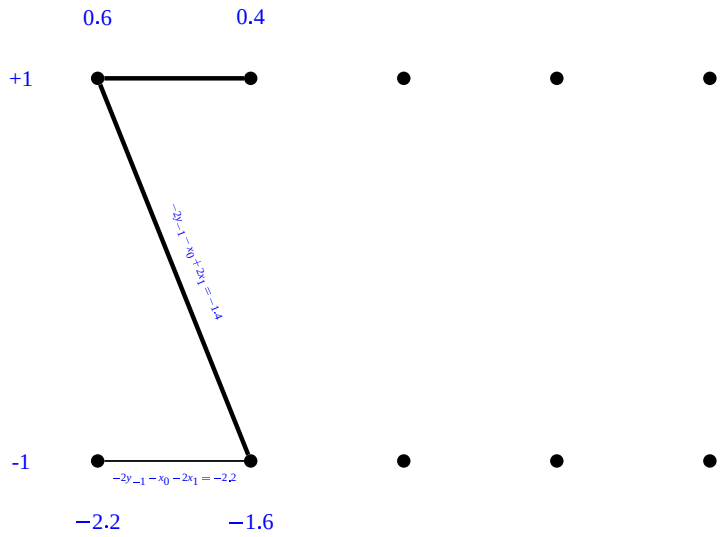
Trellis



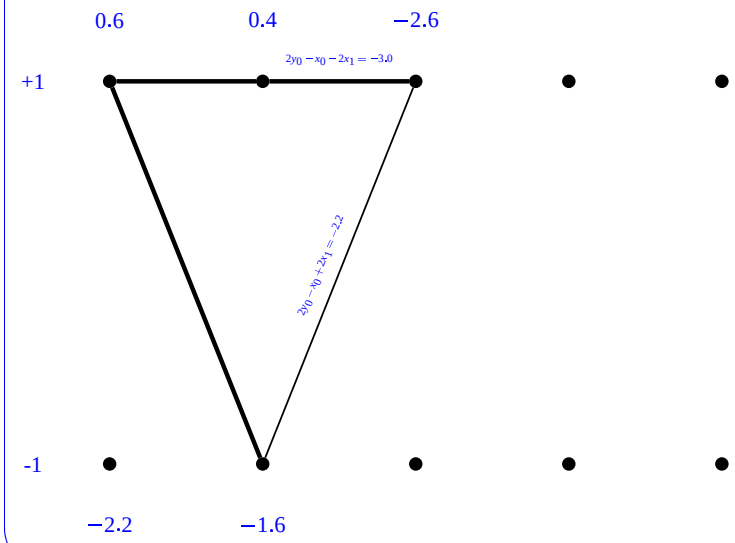
Trellis



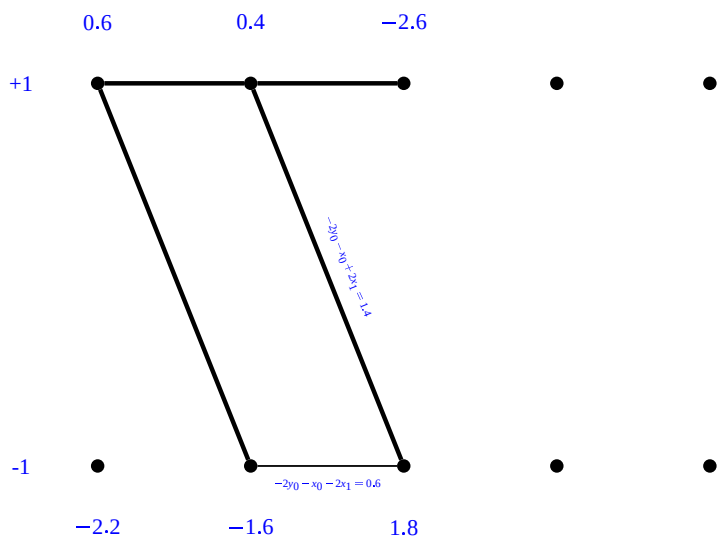
Trellis



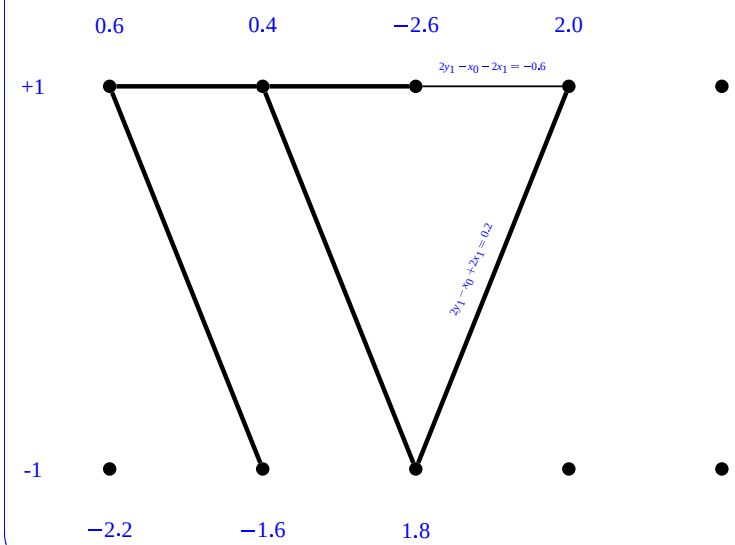
Trellis



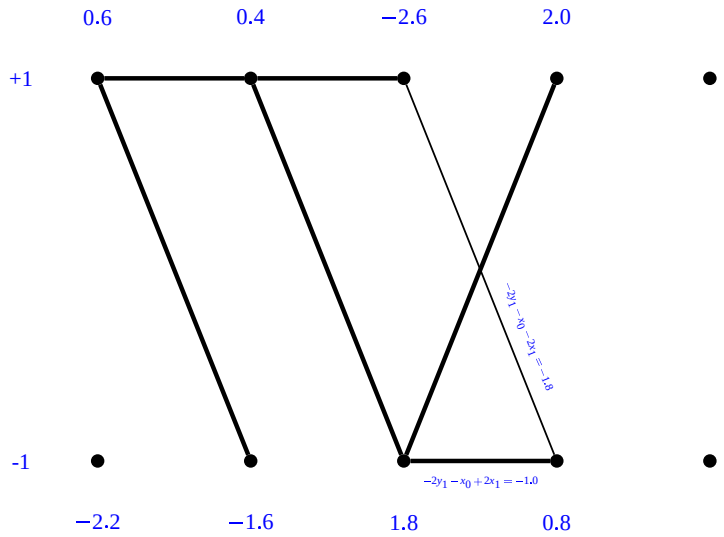
Trellis



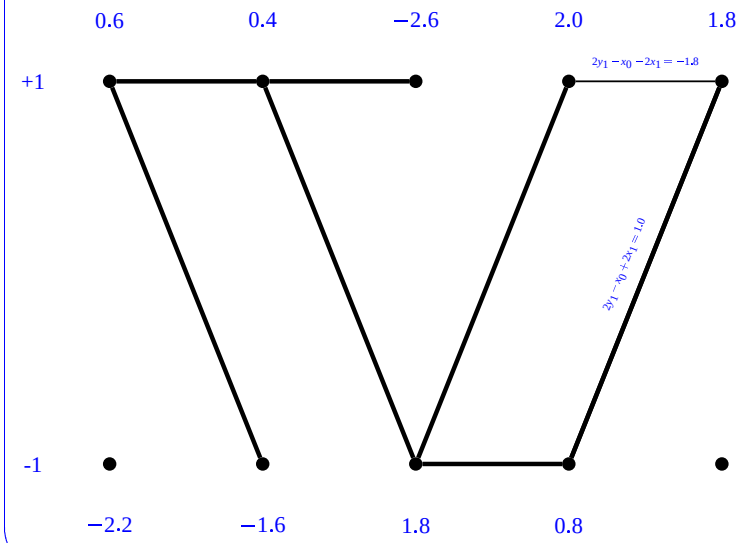
Trellis



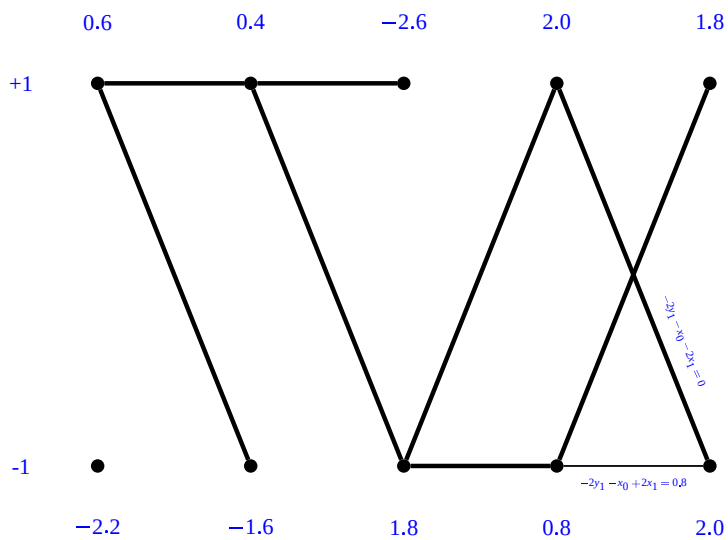
Trellis



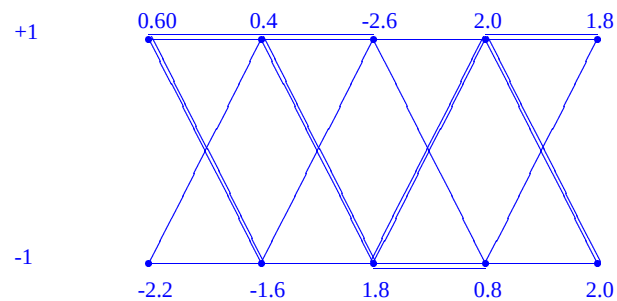
Trellis



Trellis



Trellis

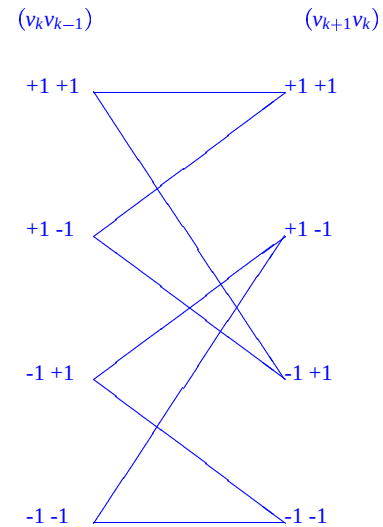


The double lines represent the path chosen by the Viterbi decoder. Thus the Viterbi decoder would output the sequence $\mathbf{v} = (+1, +1, -1, +1, -1)$.

Example: $L = 2$, $x_0 \neq 0$, $x_1 \neq 0$, $x_2 \neq 0$, $v_k \in \{\pm 1\}$

$$\Lambda_N(\mathbf{v}) = \sum_{k=-N}^{N-1} \lambda(\sigma_k, \sigma_{k+1}) + 2v_{-N}y_{-N} - x_0 v_{-N}^2$$

$$\lambda(\sigma_k, \sigma_{k+1}) = 2v_{k+1}y_{k+1} - x_0 v_{k+1}^2 - 2v_{k+1}v_k x_1 - 2v_{k+1}v_{k-1}x_2$$



v_k	v_{k-1}	v_{k+1}	v_k	Metric
$+1$	$+1$	$+1$	$+1$	$2y_{k+1} - x_0 - 2x_1 - 2x_2$
$+1$	$+1$	-1	$+1$	$-2y_{k+1} - x_0 + 2x_1 + 2x_2$
$+1$	-1	$+1$	$+1$	$2y_{k+1} - x_0 - 2x_1 + 2x_2$
$+1$	-1	-1	$+1$	$-2y_{k+1} - x_0 + 2x_1 - 2x_2$
-1	$+1$	$+1$	-1	$2y_{k+1} - x_0 + 2x_1 - 2x_2$
-1	$+1$	-1	-1	$-2y_{k+1} - x_0 - 2x_1 + 2x_2$
-1	-1	$+1$	-1	$2y_{k+1} - x_0 + 2x_1 + 2x_2$
-1	-1	-1	-1	$-2y_{k+1} - x_0 - 2x_1 - 2x_2$

Error Probability

Now consider the performance of the above maximum likelihood sequence detector (MLSD). We will evaluate the union upper bound to the error probability. To do this we need to determine the pairwise error probability between two sequences. This is the probability that sequence $\mathbf{v}$ is demodulated given sequence $\mathbf{u}$ is transmitted for a system with only two possible transmitted sequences $(\mathbf{u}, \mathbf{v})$. Let $P\{\mathbf{u} \to \mathbf{v}\}$ denote the conditional error probability given $\mathbf{u}$ transmitted. Then

$$\begin{aligned} P\{\mathbf{u} \to \mathbf{v}\} &= P\{\Lambda_N(\mathbf{v}) \geq \Lambda_N(\mathbf{u}) | \mathbf{u}\} \\ &= Q\left(\frac{\|s_{\mathbf{v}}(t) - s_{\mathbf{u}}(t)\|}{\sqrt{2N_0}}\right) \\ &\leq e^{-\|s_{\mathbf{v}}(t) - s_{\mathbf{u}}(t)\|^2/4N_0}. \end{aligned}$$

As expected the pairwise error probability depends only on the square Euclidean distance between the signals $s_{\mathbf{u}}$ and $s_{\mathbf{v}}$.

$$\|s_{\mathbf{v}}(t) - s_{\mathbf{u}}(t)\|^2 = \int_{-\infty}^{\infty} (s_{\mathbf{v}}(t) - s_{\mathbf{u}}(t))^2 \, dt$$

$$
\begin{aligned}
&= \int_{-\infty}^{\infty} \left(\sum_{k=-N}^{N} (v_k - u_k)\, h(t-kT) \right)^2 dt \\
&= \int_{-\infty}^{\infty} \sum_{k=-N}^{N} \sum_{m=-N}^{N} (v_k - u_k)(v_m - u_m)\, h(t-kT) h(t-mT)\, dt \\
&= \sum_{k=-N}^{N} \sum_{m=-N}^{N} (v_k - u_k)(v_m - u_m) \int_{-\infty}^{\infty} h(t-kT) h(t-mT)\, dt \\
&= \sum_{k=-N}^{N} \sum_{m=-N}^{N} (v_k - u_k)(v_m - u_m)\, x_{k-m} \,.
\end{aligned}
$$

Let $\varepsilon_k = \frac{1}{2}(v_k - u_k) = \begin{cases} 1 & v_k = 1, u_k = -1 \\ 0 & v_k = u_k \\ -1 & v_k = -1, u_k = 1 \,. \end{cases}$

$$
\begin{aligned}
\|s_{\mathbf{v}}(t) - s_{\mathbf{u}}(t)\|^2 &= 4 \sum_{k=-N}^{N} \sum_{m=-N}^{N} \varepsilon_k \varepsilon_m x_{k-m} \\
&= 4 \sum_{k=-N}^{N} \varepsilon_k^2 x_0 + 4 \sum_{k=-N+1}^{N} 2 \sum_{m=-N}^{k-1} \varepsilon_k \varepsilon_m x_{k-m}
\end{aligned}
$$

$$
\begin{aligned}
&= 4 \left[\sum_{k=-N}^{N} \left(\varepsilon_k^2 x_0 + 2 \sum_{j=1}^{k+N} \varepsilon_k \varepsilon_{k-j} x_j \right) \right] \\
&= 4 \left[\sum_{k=-N}^{N} \left(\varepsilon_k^2 x_0 + 2 \sum_{j=1}^{L} \varepsilon_k \varepsilon_{k-j} x_j \right) \right]
\end{aligned}
$$

$$
\begin{aligned}
P(\mathbf{u} \to \mathbf{v}) &\le \exp\left\{ -\frac{1}{N_0} \sum_{k=-N}^{N} \left(\varepsilon_k^2 x_0 + 2 \sum_{m=1}^{L} \varepsilon_k \varepsilon_{k-m} x_m \right) \right\} \\
&= P(\mathbf{e})
\end{aligned}
$$

Thus the incremental Euclidean distance between two paths for a given time index k is depends on the past L errors. The error state is defined to be the last L errors $(\varepsilon_k, \ldots - \varepsilon_{k-L+1})$. The all zero error state corresponds to the past L symbols being correctly demodulated.

An error event is defined to be an error sequence that diverges once from the all zero state and then remerges later. Since a necessary condition for an error of a particular type (first event error or symbol error) is that an error event occurs that causes the demodulator/decoder to follow a path that diverges and then at some later time remerges we can calculate the error probability for a particular node by counting the number of paths (and their distance) that diverge and remerge. We can use the state diagram to determine the number of error sequences with a particular distance.

Let $\mathbf{e} = (\varepsilon_{-N}, \ldots, \varepsilon_N)$ $w_H(\mathbf{e})$ = Hamming weight of $\mathbf{e}$ (number of nonzero terms)

$$
\begin{aligned}
P_{E,m} &= \text{First event error probability} \\
&= P\{\text{at time } m \text{ decoder is not at correct state for the first time}\} \\
P_b &= \text{Bit error probability} \\
&= P\{\text{bit error occurs for symbol } m\}
\end{aligned}
$$

The union bound on the probability of error at time 0 is

$$
\begin{aligned}
P_{E,m} &\le \sum_{\substack{\mathbf{u},\mathbf{v} \\ \varepsilon \neq 0}} P(\mathbf{u} \to \mathbf{v}) P(\mathbf{u}) \\
&= \sum_{\substack{\mathbf{e} \\ \varepsilon \neq 0}} \sum_{\substack{\mathbf{u},\mathbf{v} \\ \mathbf{e} = \frac{1}{2}(\mathbf{v}-\mathbf{u})}} P(\mathbf{u} \to \mathbf{v}) P(\mathbf{u})
\end{aligned}
$$

where the sum is over all sequences that diverge from the all zero state and then remerge later. Each of the 2^{2N+1} $\mathbf{u}$ sequences are equally likely. In each position where $\varepsilon_k \neq 0$ the components of the sequences $\mathbf{u}$ and $\mathbf{v}$ are determined. If $\varepsilon_k = 1$ then $v_k = 1$ and $u_k = -1$. Similarly if $\varepsilon_k = -1$ then $v_k = -1$ and $u_k = 1$. The components $\mathbf{e}$ where $\varepsilon_k = 0$ there are two choices for u_k and v_k ($u_k = v_k = 1$ or $u_k = v_k = -1$). Since there are $2N + 1 - w_H(\mathbf{e})$ places

where $\varepsilon_k = 0$ there are $2^{2N+1-w_H(\mathbf{e})}$ such sequences $\mathbf{u}$ and $\mathbf{v}$. Hence

$$
\begin{aligned}
P_{E,f} &\le \sum_{\substack{\mathbf{e} \\ \varepsilon_0 \neq 0}} P(\mathbf{e}) 2^{(2N+1)-w_H(\mathbf{e})} 2^{-(2N+1)} \\
&= \sum_{\substack{\mathbf{e} \\ \varepsilon_0 \neq 0}} 2^{-w_H(\mathbf{e})} P(\mathbf{e}) \,.
\end{aligned}
$$

The bit error probability is bounded by

$$
P_b \le \sum_{\mathbf{e}} \frac{w(\mathbf{e})}{2^{w(\mathbf{e})}} P(\mathbf{e}) \,.
$$

$$
\frac{P(\mathbf{e})}{2^{w(\mathbf{e})}} = \prod_{k=-N}^{N} \frac{1}{2^{w_H(\varepsilon_k)}} \exp\left\{ -\frac{1}{N_0} \left(\varepsilon_k^2 x_0 + 2 \sum_{m=1}^{L} \varepsilon_k \varepsilon_{k-m} x_m \right) \right\}
$$

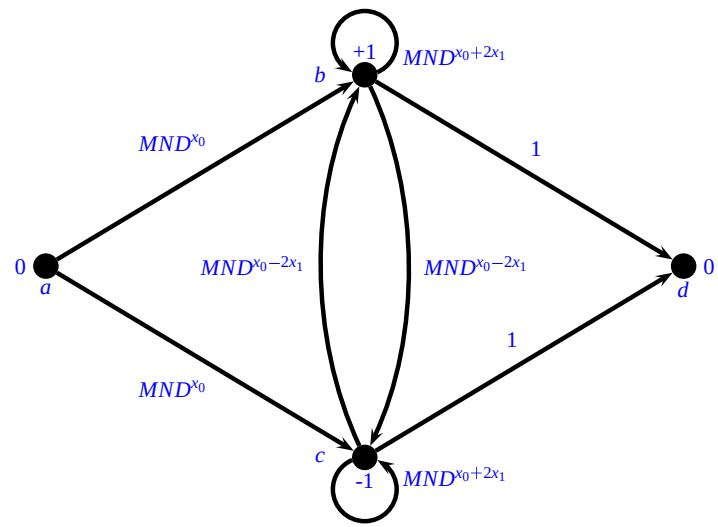
For $L = 1$

$$
\frac{P(\mathbf{e})}{2^{w_h(\mathbf{e})}} = \prod_{k=-N}^{N} \frac{1}{2^{w_H(\varepsilon_k)}} \exp\left\{ -\frac{1}{N_0} \left(\varepsilon_k^2 x_0 + 2 \varepsilon_k \varepsilon_{k-1} x_1 \right) \right\}
$$

We calculate these union bounds by enumerating the sequences that diverge from the all zero error state and remerge (error events) that correspond to a given Euclidean distance between two data sequences and has a given number of nonzero terms (or is a given length). To do this we draw a state diagram (similar to that for convolutional codes) and label each path with

$D^x N^l M^l$ where x is the incremental Euclidean distance squared (divided by 4) in going from one state to another and l is 1 if the error path is nonzero and is zero if the error is zero. (This redundant use of l will be explained when we determine the bit error probability).

Error State Diagram $L = 1$



Transfer Function

The transfer function is calculated by solving the following equations for T_d/T_a.

$$\begin{aligned} T_d &= T_c + T_b \\ T_b &= MND^{x_0-2x_1}T_b + MND^{x_0+2x_1}T_c + MND^{x_0}T_a \\ T_c &= MND^{x_0-2x_1}T_c + MND^{x_0+2x_1}T_b + MND^{x_0}T_a \end{aligned}$$

Adding the last two equations and solving for $T_b + T_c$ and substituting the result into the first equation yields

$$\begin{aligned} \frac{T_d}{T_a}(D,N,M) &= \frac{2NMD^{x_0}}{1 - NMD^{x_0+2x_1} - NMD^{x_0-2x_1}} \\ &= 2NMD^{x_0} + 2N^2M^2D^{2x_0-2x_1} + 2N^2M^2D^{2x_0+2x_1} + \ldots \end{aligned}$$

Thus there are two paths with 1 error and Euclidean distance squared of $4x_0$. There are two

paths with two errors and Euclidean distance squared of $4x_0 - 4x_1$ and so on.

$$\begin{aligned} P_E &\leq T(D,N,M)|_{D=e^{-1/N_0},N=1,M=1/2} = \frac{e^{-x_0/N_0}}{\left[1 - \frac{1}{2}e^{-x_0/N_0}\left(e^{2x_1/N_0} + e^{-2x_1/N_0}\right)\right]} \\ P_b &\leq \frac{\partial T(D,N,M)}{\partial N}|_{D=e^{-1/N_0},N=1,M=1/2} = \frac{2MD^{x_0}}{\left[1 - NM\left(D^{x_0-2x_1} + D^{x_0+2x_1}\right)\right]^2}|_{D=e^{-1/N_0},N=1,M=1/2} \\ &= \frac{e^{-x_0/N_0}}{\left[1 - \frac{1}{2}e^{-x_0/N_0}\left(e^{2x_1/N_0} + e^{-2x_1/N_0}\right)\right]^2} \end{aligned}$$

For large SNR this is the same as no ISI! Just as with convolutional codes this Union-Bhattacharyya bound can be improved by using the exact error probability for the first few terms and then upper bounding the error probability for higher order terms with the Bhattacharyya bound.

Union Bound

The union bound can be tightly bounded by

$$P_b < \sum_{j=5}^{J} w_j (Q(\sqrt{\frac{2E_b j}{5N_0}}) - D^{j/5}) + w(D)|_{D=\exp\{-x_0/N_0\}}$$

Energy

Notice that the energy of the signal at the output of the channel is

$$\begin{aligned} E &= E_x[\int_{-\infty}^{\infty} s^2(t)dt] \\ &= \int_{-\infty}^{\infty} E_x[(\sum_{k=-N}^{N} u_k h(t-kT))^2 dt \\ &= \int_{-\infty}^{\infty} E_x[\sum_{k=-N}^{N} u_k h(t-kT)) \sum_{l=-N}^{N} u_l h(t-lT))] \\ &= \int_{-\infty}^{\infty} \sum_{k=-N}^{N} \sum_{l=-N}^{N} E_x[u_k u_l] h(t-lT) h(t-kT) dt \end{aligned}$$

where $E_x[\cdot]$ denotes expectation with respect to the data bit. Because the data are assumed independent, identically distributed with $E_x[u_k u_l] = 1 if k = l$ and is 0 otherwise the energy of the signal is

$$E = \int_{-\infty}^{\infty} \sum_{k=-N}^{N} u_k h^2(t-kT) dt$$

$$\begin{aligned} &= \sum_{k=-N}^{N} \int_{-\infty}^{\infty} h^2(t-kT) dt \\ &= \sum_{k=-N}^{N} x_0 \\ &= (2N+1)x_0 \end{aligned}$$

The energy per bit is $E_b = E/(2N+1) = x_0$.

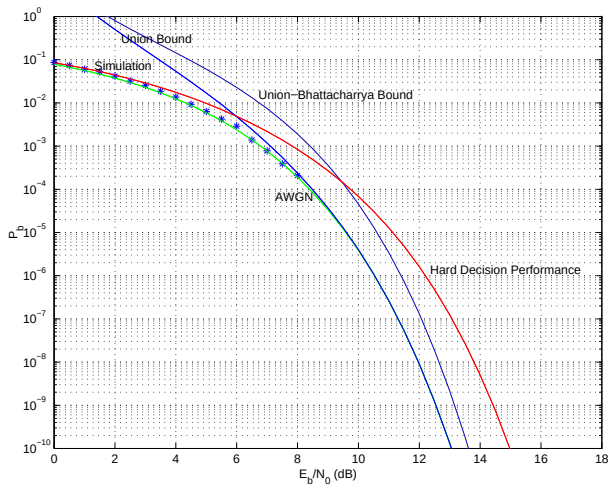


Figure 106: Performance of Optimal Receiver ($x_0 = 1.0, x_1 = 0.2$)

Maple Code

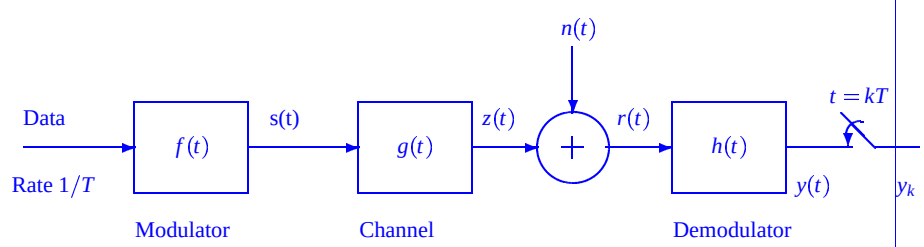
```
x0:=1.00;
x1:=0.20;
with(linalg);
a:=M*N*D^(x0-2*x1);
b:=M*N*D^(x0+2*x1);
c:=M*N*D^x0;
m:=matrix(3,3,[[-1, -1, 1],[1-a, -b, 0],[-b, 1-a, 0]]);
d:=[0, M*N*D^x0, M*N*D^x0];
fxx1:=linsolve(m,d);
fxx2:=fxx1[3];
fxx3:=diff(fxx2, N);
fxx4:=eval(fxx3, N=1);
fxx5:=eval(fxx4, M=0.5);
fxx6:=simplify(fxx5);
fxx7:=series(fxx6, D=0, 5);
```


Bounds ($x_0 = 1, x_1 = 0.2$)

$$
\begin{aligned}
P_b \;<\;& D + D^{\frac{8}{5}} + 0.75D^{\frac{11}{5}} + D^{\frac{12}{5}} + 0.5D^{\frac{14}{5}} + 1.5D^3 \\
& + 0.3125D^{\frac{17}{5}} + 1.5D^{\frac{18}{5}} + 0.75D^{\frac{19}{5}} + 0.1875D^4 + 1.25D^{\frac{21}{5}} + 1.5D^{\frac{22}{5}} + 0.109375D^{\frac{23}{5}} \\
& + 0.9375D^{\frac{24}{5}} + 1.875D^5 + 0.5625D^{\frac{26}{5}} + 0.65625D^{\frac{27}{5}} + 1.875D^{\frac{28}{5}} + 1.28515625D^{\frac{29}{5}} \\
& + 0.4375D^6 + 1.640625D^{\frac{31}{5}} + 1.89453125D^{\frac{32}{5}} + 0.59375D^{\frac{33}{5}} + 1.3125D^{\frac{34}{5}} \\
& + 2.198242188D^7 + \ldots|_{D=\exp\{-1/N_0\}} \\
=\;& \sum_{j=15}^{\infty} w_j D^{j/5}|_{D=\exp\{-1/N_0\}} \\
=\;& w(D)|_{D=\exp\{-1/N_0\}}
\end{aligned}
$$

Signal Design for Filtered Channels

Because the complexity of the Viterbi algorithm grows as the $|A|^L$ where $|A|$ is the alphabet size and L is the memory of the channel it is desirable to design a system with zero intersymbol interference. So consider transmitting data at rate $1/T$ through a channel with bandwidth W. At what rate is this possible *without* creating intersymbol interference? Assume the modulator is a filter acting on a infinite sequence of impulses (at rate $1/T$ with impulse response $f(t)$. The channel is characterized by an impulse response of $g(t)$ and the receiver is a filter sampled at rate $1/T$ with impulse response $h(t)$.



The transmitted signal is of the form

$$s(t) = \sum_{m=-\infty}^{\infty} u_m f(t - mT)$$

The output of the received filter is then

$$y_k = \sum_{m=-N}^{N} u_m x_{k-m} + \eta_k.$$

In order that there be no intersymbol interference we require that

$$x_m = \begin{cases} 1, & m = 0 \\ 0, & m \neq 0. \end{cases}$$

Let

$$\begin{aligned} x(t) &= \int_{-\infty}^{\infty} h(\tau)h(\tau - t)dt \\ &= \int_{-\infty}^{\infty} h(\tau)\tilde{h}(t-\tau)dt \end{aligned}$$

where $\tilde{h}(t) = h(-t)$. Then $x(t)$ is the convolution of $h(t)$ with $\tilde{h}(t)$ and $x(nT) = x_n$. Thus $X(f) = H(f)\tilde{H}(f) = |H(f)|^2$. If the (absolute) bandwidth of the channel is W then $X(f)$ also has bandwidth W. That is

$$X(f) = 0 \quad |f| > W$$

Thus by the sampling theorem

$$\begin{aligned} x(t) &= \sum_{n=-\infty}^{\infty} x(\frac{n}{2W}) \frac{\sin(2\pi W(t - \frac{n}{2W}))}{2\pi W(t - \frac{n}{2W})} \\ &= \sum_{n=-\infty}^{\infty} x(\frac{n}{2W}) \phi_{n,W}(t). \end{aligned}$$

where

$$\phi_{n,W}(t) = \frac{\sin(2\pi W(t - \frac{n}{2W}))}{2\pi W(t - \frac{n}{2W})}.$$

If $W = 1/2T$ then

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT) \frac{\sin(\pi(t - nT)/T)}{\pi(t - nT)/T}$$

For no intersymbol interference we require that $x(nT) = 0$ for $n \neq 0$. Let $x(0) = 1$ then

$$x(t) = \frac{\sin(\pi(t - nT)/T)}{\pi(t - nT)/T}.$$

which implies that

$$X(f) = \begin{cases} T, & |f| \leq \frac{1}{2T} \\ 0, & |f| > \frac{1}{2T} \end{cases}$$

Thus

$$|H(f)|^2 = \begin{cases} T, & |f| \leq \frac{1}{2T} \\ 0, & |f| > \frac{1}{2T} \end{cases}$$

Thus $2W$ pulses per second can yield zero intersymbol interference. It is easy to see that by signaling faster than rate $2W$ we can not guarantee that there is no intersymbol interference.

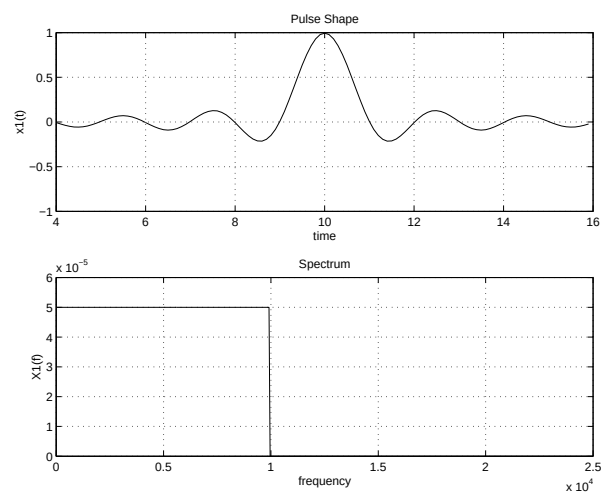


Figure 107: Nyquist Pulse Shape and Spectrum

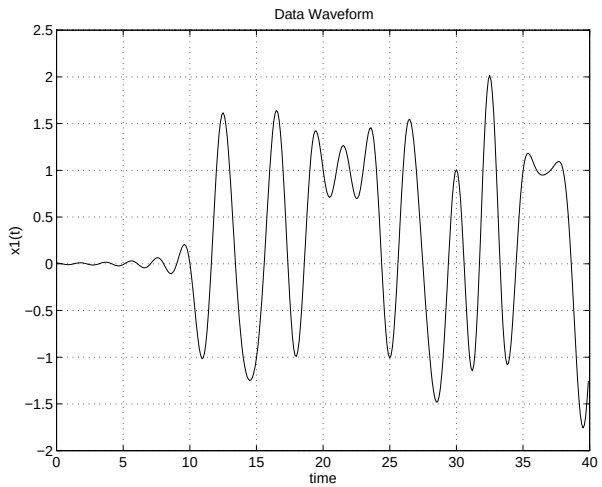


Figure 108: Nyquist Waveform

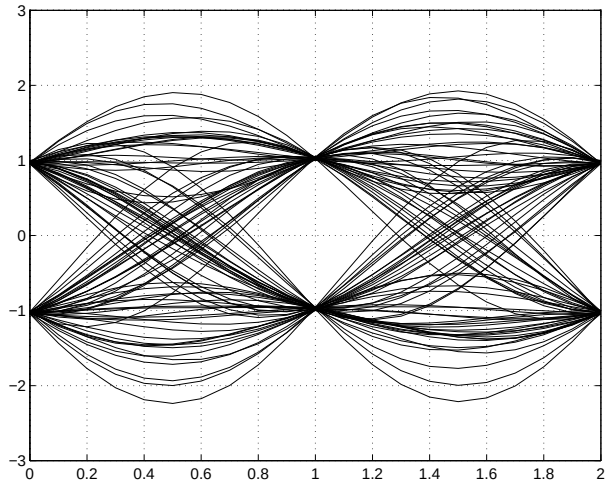


Figure 109: Nyquist Eye Diagram

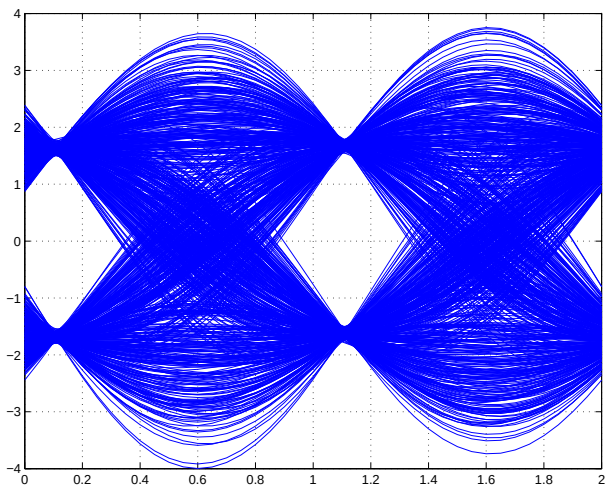


Figure 110: Nyquist Eye Diagram

Nyquist Pulses

Problems with this pulse shape are:

- It is hard to generate
- A slight timing error results in infinite series decaying as $1/t$ for intersymbol interference.

Solutions

- Signal slower
- Allow intersymbol interference in a controlled fashion.

Intersymbol-Interference Free Pulse Shapes

Consider slower signaling first. Consider $\frac{1}{T} < 2W$, $W > 1/2T$. (This implies aliasing at the receiver. Since $W > 1/2T$ we can divide the interval $[-W, W]$ into segments of length $1/T$. Let $N = \lceil (2WT-1)/2 \rceil$ be the number of such segments. When the signal is sampled these segments get moved to the origin and cause aliasing.

$$\begin{aligned} x(kT) &= \int_{-W}^{W} X(f) e^{j2\pi fkT} df \\ &= \sum_{n=-N}^{N} \int_{f=(2n-1)/T}^{(2n+1)/2T} X(f) e^{j2\pi fkT} df \\ &= \sum_{n=-N}^{N} \int_{f=-1/2T}^{1/2T} X(f+\frac{n}{T}) e^{j2\pi(f+\frac{n}{T})kT} df \\ &= \int_{f=-1/2T}^{1/2T} [\sum_{n=-N}^{N} X(f+\frac{n}{T})] e^{j2\pi fkT} df \end{aligned}$$

Let

$$X_{eq}(f) = \sum_{n=-N}^{N} X(f+\frac{n}{T})$$

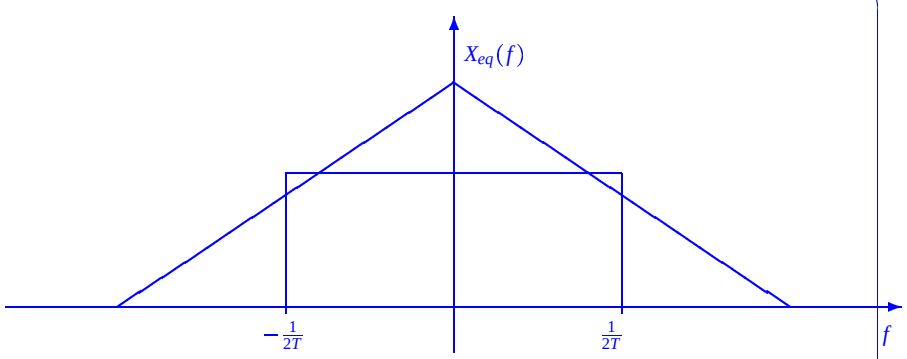
Then

$$x(kT) = \int_{f=-1/(2T)}^{1/(2T)} X_{eq}(f) e^{j2\pi ukT} df$$

Clearly we can have zero intersymbol interference if

$$X_{eq}(f) = \begin{cases} T, & |f| \le \frac{1}{2T} \\ 0, & |f| > \frac{1}{2T}, \end{cases}$$

Examples (1)



(2)

$$X(f) = \begin{cases} T, & 0 \le |f| \le \frac{1-\alpha}{2T} \\ \frac{T}{2}[1-\sin(\pi T(f-\frac{1}{2T})/\alpha)], & \frac{1-\alpha}{2T} \le |f| \le \frac{1+\alpha}{2T} \end{cases}$$

$$x(t) = \frac{\sin(\pi t)}{\pi t} \frac{\cos(\alpha\pi t/T)}{1-4\alpha^2 t^2/T^2}$$

Notice that the pulse decays as $1/t^3$ instead of $1/t$ for the Nyquist pulse. The parameter α is called the rolloff factor.

$$H(f) = \begin{cases} \sqrt{T}, & 0 \le |f| \le \frac{1-\alpha}{2T} \\ \sqrt{\frac{T}{2}[1-\sin(\pi t(f-\frac{1}{2T})/\alpha)]}, & \frac{1-\alpha}{2T} \le |f| \le \frac{1+\alpha}{2T} \end{cases}$$

$$h(t) = \frac{\sin(\pi(1-\alpha)t) + 4\alpha t\cos(\pi(1+\alpha)t)}{\pi[1-(4\alpha t)^2]t}$$

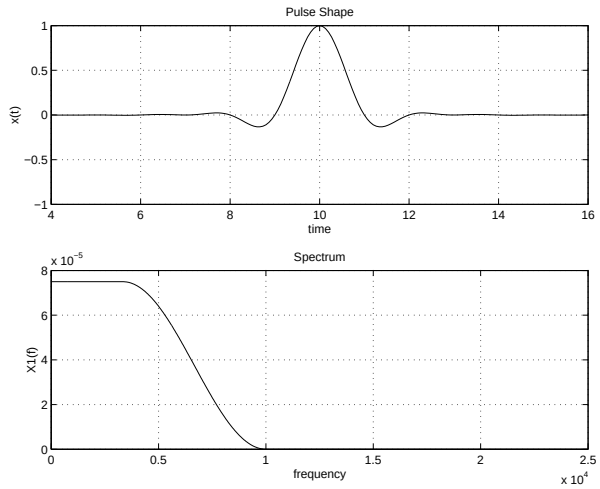


Figure 111: Raised Cosine Pulse Shape and Spectrum ($\alpha = 0.5$)

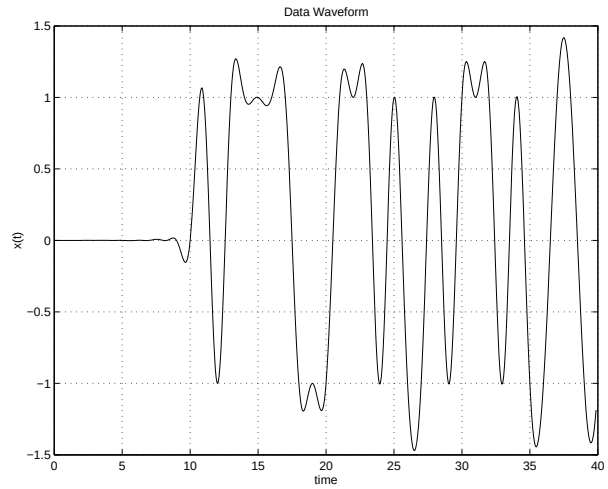


Figure 112: Raised Cosine Waveform ($\alpha = 0.5$)

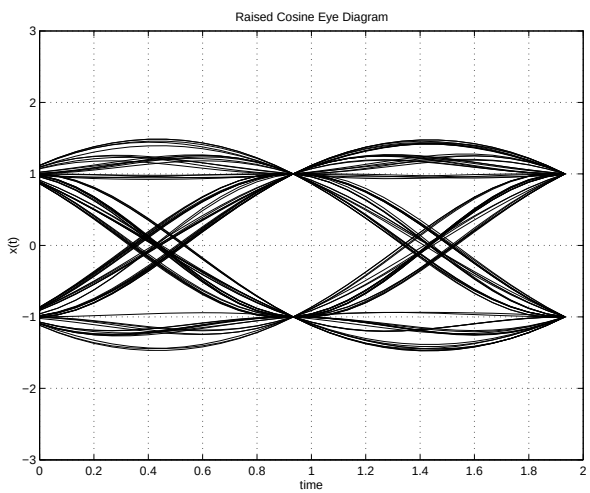


Figure 113: Raised Cosine Eye Diagram $\alpha = 0.5$

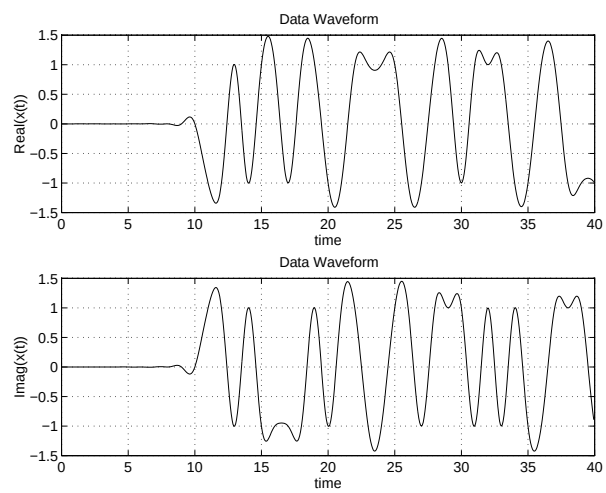


Figure 114: Raised Cosine Waveform ($\alpha = 0.5$)

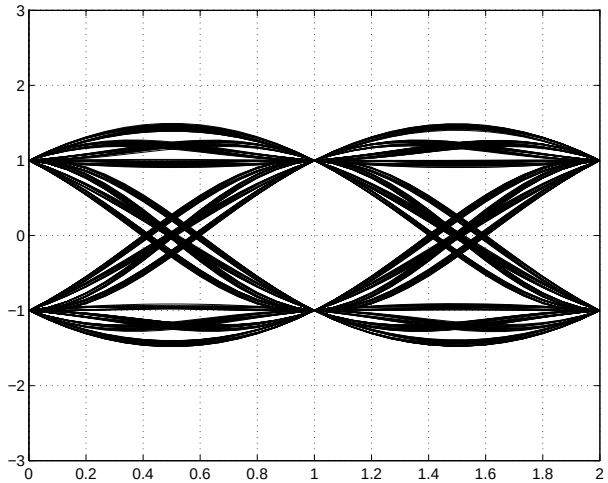


Figure 115: Raised Cosine Waveform Eye Diagram ($\alpha = 0.5$)

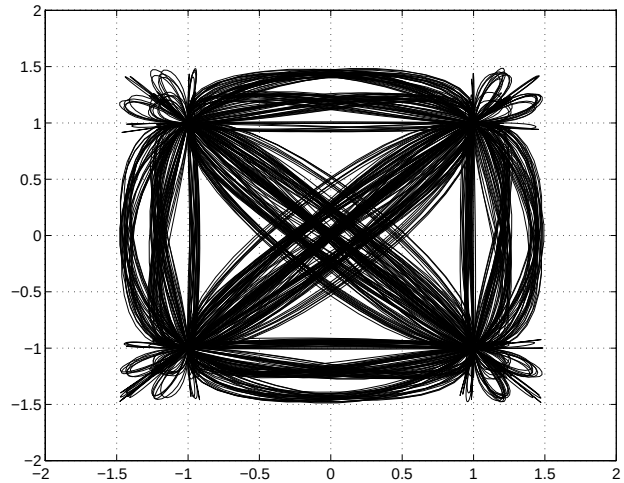


Figure 116: Raised Cosine Waveform ($\alpha = 0.5$)

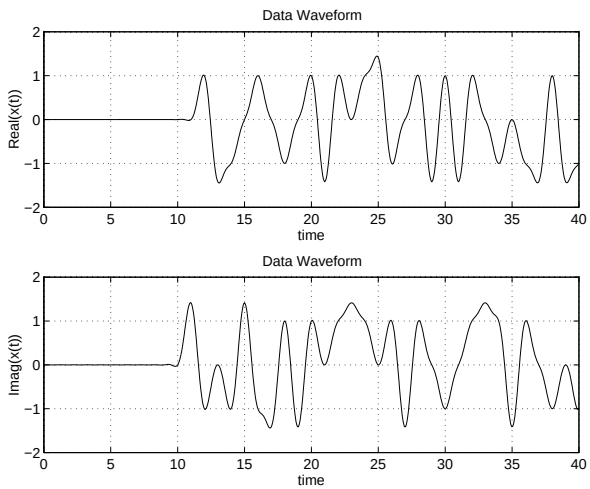


Figure 117: Raised Cosine Waveform $\pi/4$ QPSK ($\alpha = 1.0$)

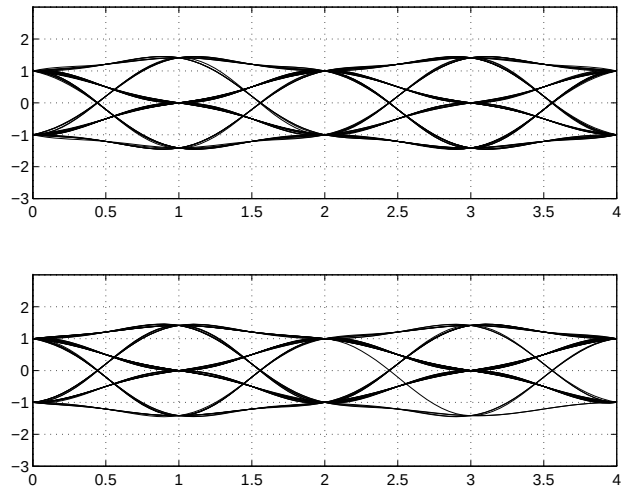


Figure 118: Raised Cosine Waveform Eye Diagram $\pi/4$ QPSK ($\alpha = 1.0$)

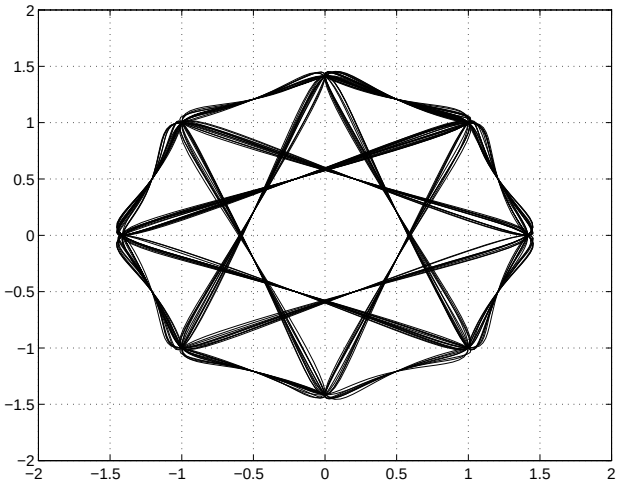


Figure 119: Raised Cosine Waveform $\pi/4$ QPSK ($\alpha = 1.0$)

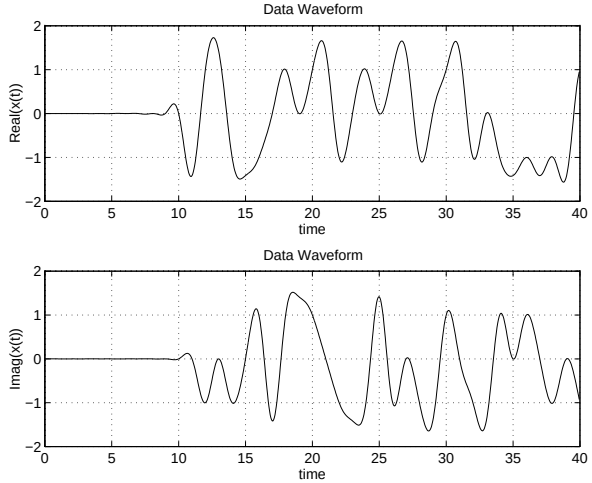


Figure 120: Raised Cosine Constellation $\pi/4$ QPSK ($\alpha = 0.5$)

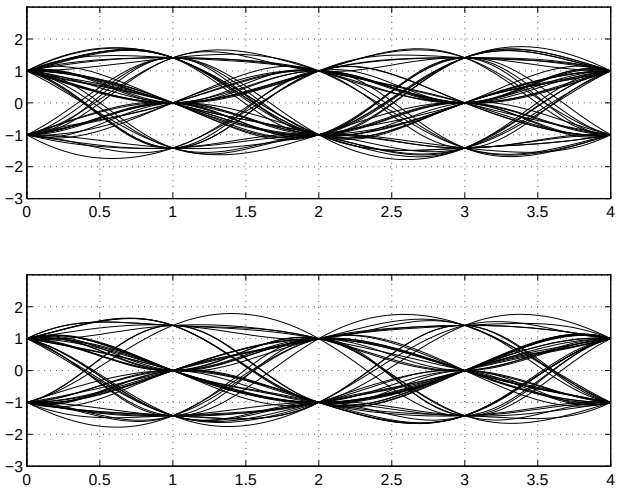


Figure 121: Raised Cosine Waveform Eye Diagram $\pi/4$ QPSK ($\alpha = 0.5$)

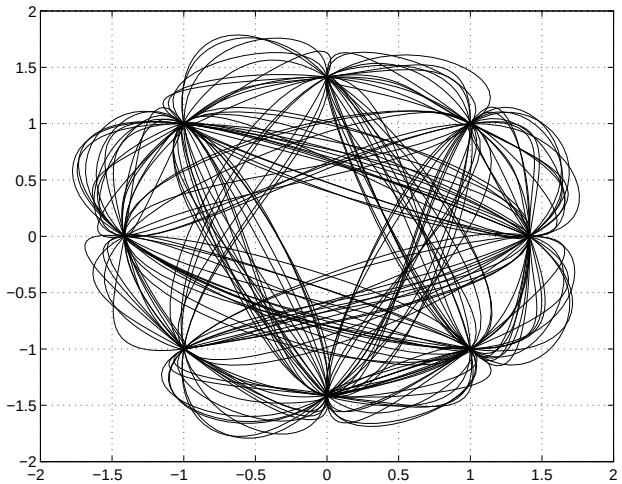


Figure 122: Raised Cosine Waveform $\pi/4$ QPSK ($\alpha = 0.5$)

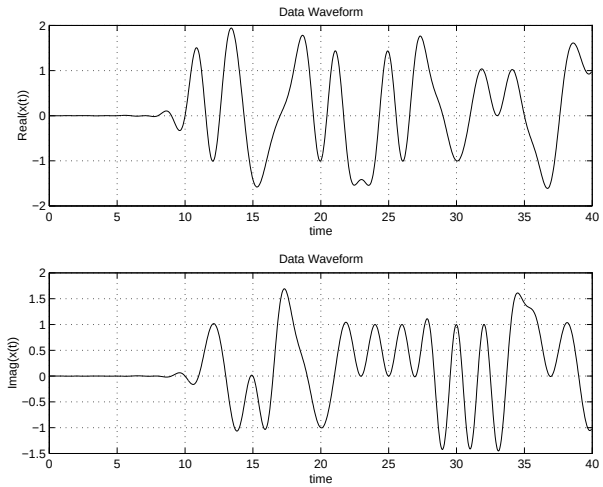


Figure 123: Raised Cosine Waveform $\pi/4$ QPSK ($\alpha = 0.35$)

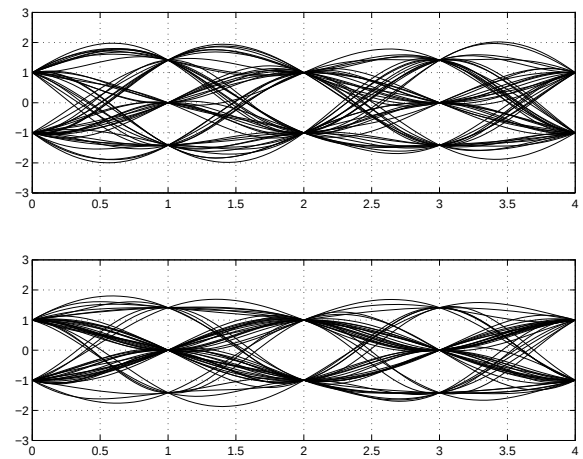


Figure 124: Raised Cosine Waveform Eye Diagram $\pi/4$ QPSK ($\alpha = 0.35$)

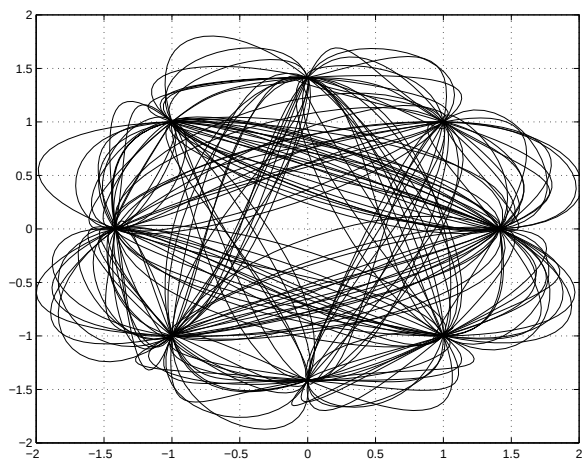


Figure 125: Raised Cosine Waveform $\pi/4$ QPSK ($\alpha = 0.35$)

Because we do not have any intersymbol interference the performance of this method for

avoiding intersymbol interference has the same performance as BPSK. The difference is that this modulation scheme requires absolute bandwidth of $W = \frac{1+\alpha}{2T}$. However, since this does not result in a constant envelope signal for applications requiring constant envelope transmission this modulation scheme is not acceptable.

Controlled Intersymbol Interference: Partial-Response

The second method is to allow some intersymbol interference. This intersymbol interference is allowed in a controlled fashion. We still signal at rate $1/T = 2W$ but do not require zero intersymbol interference.

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT)\frac{\sin(\pi(t-nT)/T)}{\pi(t-nT)/T}$$

$$X(f) = \begin{cases} \sum_{n=-\infty}^{\infty} x(nT)\frac{e^{-jn\pi f/W}}{2W}, & |f| \le W \\ 0, & |f| > W. \end{cases}$$

Example (1):

$$x(nT) = \begin{cases} 1, & n = 0,1 \\ 0, & n \ne 0,1 \end{cases}$$

This is called Duobinary Transmission (also called partial response class I).

$$X(f) = \begin{cases} \frac{e^{-j\pi f/2W}}{W}\cos(\frac{\pi f}{2W}), & |f| \le W \\ 0, & |f| > W. \end{cases}$$

Example (2):

$$x(nT) = \begin{cases} 1, & n = -1,1 \\ 0, & n \ne 0,1 \end{cases}$$

This is called Modified Duobinary Transmission (also called Partial Response Class IV).

$$X(f) = \begin{cases} \frac{j}{W}\sin(\frac{\pi f}{W}), & |f| \le W \\ 0, & |f| > W. \end{cases}$$

This is used in magnetic recording (with maximum likelihood decoding).

For these systems with controlled intersymbol interference we still need a way to detect the data. One method is decision feedback. The other method is precoding.

Ortogonal Frequency Division Multiplexing (OFDM)

Orthogonal frequency division multiplexing (OFDM) is an alternative method of eliminating intersymbol interferece.

Let $\mathbf{b} = (b_0, ..., b_{N_{bp}})$ denote the packet of encoded data. This packet is serial-to-parallel converted into M streams of data each of length $L = N_{bp}/M$. The sequences are $\mathbf{b}_0, ..., \mathbf{b}_{M-1}$ where

$$\begin{aligned} \mathbf{b_0} &= (b_0, b_M, b_{2M}, ..., b_{(L-1)M}) \\ \mathbf{b_1} &= (b_1, b_{M+1}, b_{2M+1}, ..., b_{(L-1)M+1}) \\ \mathbf{b_{M-1}} &= (b_{M-1}, b_{2M-1}, ..., b_{LM-1}) \end{aligned}$$

Equivalently the data streams can be written as a sequence of L column vectors of length M components.

$$\widetilde{\mathbf{b}}_0 = (b_0, b_1, ..., b_{M-1})^T$$

The data vector $\widetilde{\mathbf{b}}_0$ is augmented with zeros to generate the data column vector C_0 where

$$\mathbf{C_0} = (0, 0, ..., 0, b_0, ..., b_{M-1}, 0, 0, ..., 0)^T$$

where the leading and trailing zeros are used to form a guardband around the signal as will be described subsequently. The vector $\mathbf{C_0}$ has length M_d. The vector $\mathbf{C_0}$ is the input to an IFFT which produces a sequence of time samples of length M_d. These time domain samples correspond to samples in time with a sampling rate of T/M_d. Let c_0 denote the output of the IFFT when C_0 is the input. Now for purposes of mitigating intersymbol interference prepend a number of these samples.

$$\widetilde{c}_0 = (c_{M_d-\nu}, c_{M_d-\nu+1}, ..., c_{M_d-1}, c_0, ..., c_{M-1})$$

The length of $\widetilde{c}$ is $M_d + \nu$ where ν represents the length of the channel memory (including any filtering in the transmitter and receiver). Since the sample rate is T_s/M_d the duration corresponding to $\widetilde{c}$ is $T_s(M_d+\nu)/M_d$. The data rate is then $M * M_d/(T_s * (M_d+\nu))$ coded symbols/sec. For $M_d \gg \nu$ the data rate is approximately M/T_s symbols per second. The sequence of samples after the prefix insertion for a packet then is given by

$$d = (\widetilde{c}_0, \widetilde{c}_1, \widetilde{c}_2, \widetilde{c}_3, ...\widetilde{c}_L)$$

These samples then represent the signal at a sample rate of M/T. Thus

$$d(lT/M) = d_l, \quad l = 0, 1, ..., (M+\nu)L - 1$$

In the frequency domain this signal is

$$S(f) = \sum_{l=0}^{\tilde{M}-1} c_l \mathrm{sinc}((f - l/T)T)$$

The desired transmitted signal is

$$s(t) = \sum_{l=0}^{M-1} \cos(2\pi f_H(t)t + \phi_l(t))$$

where $f_H(t)$ is a frequency hopping pattern given by

$$f_H(t) = \sum_{m=0}^{N_{hp}-1} f_m p_{T_h}(t - mT_h).$$

Here N_{hp} denotes the number of hops per packet, T_h denotes the duration of a dwell (hop) interval and $p_T(t) = 1$ for $0 \le t \le T$ and zero elsewhere. The information bearing phase, $\phi_l(t)$, is given by

$$\phi_l(t) = \sum_{m=0}^{N_{sp}-1} 2\pi f_l t + b_{l,m} p_{T_s}(t - mT_s)$$

where f_l is the l-th carrier and $b_{l,m}$ is the data symbol for the $l-th$ carrier during the m-th symbol interval which takes values $k\pi/2 + \pi/4,\ k = 0,1,2,3$. The number of OFDM symbols

per packet is denoted by N_{sp}. The desired RF signal is obtained by mixing a baseband signal $s_b(t)$ up to a carrier frequency via

$$\begin{aligned} s(t) &= \mathrm{Re}[s_b(t)\exp(j2\pi f_c t)] \\ &= \mathrm{Re}[s_b(t)]\cos(j2\pi f_c t) - \mathrm{Im}[s_b(t)]\sin(j2\pi f_c t) \end{aligned}$$

The necessary baseband transmitted signal is

$$s_b(t) = \sum_{l=0}^{M-1} \exp\{\phi_l(t)\} \quad = \quad \sum_{l=0}^{M-1}\sum_{m=0}^{N_{sp}-1} p_T(t - mT_s)\exp\{j2\pi f_l t + jb_{l,m}\}$$

In order for the signal to have M orthogonal components it is necessary that any pair of frequencies used f_m and f_l satisfy

$$f_m - f_l = n/T_s$$

for some integer n. We will assume $f_l = l/T_s$. consider the signal transmitted during a single symbol interval. If the signal $s_b(t)$ is sampled at spacings $\Delta t = T_s/M$ then the samples are

$$\begin{aligned} s_b(m\Delta t) &= \sum_{l=0}^{M-1} b_l p_T(m\Delta t)\exp\{j2\pi f_l m\Delta t\} \\ &= \sum_{l=0}^{M-1} b_l p_{T_s}(m\Delta t)\exp\{j2\pi f_l m T_s/M\}, \quad 0 \le m \le M-1 \end{aligned}$$

$$\begin{aligned} &= \sum_{l=0}^{M-1} b_l \exp\{j2\pi l m/M\}, \quad 0 \le m \le M-1 \\ &= M\mathrm{IFFT}\{\mathbf{b}\}, \quad m = 0,1,\ldots,(M-1). \end{aligned}$$

where the IFFT is the inverse Fourier transform.

Receiver

At the receiver the received signal is

$$r_b(t) = s_b(t) + n(t)$$

The optimal receiver does a complex correlation to determine the data.

$$\begin{aligned} Z_l &= \int_0^T s_b(t)\exp\{-j2\pi f_l t\}dt \\ &= \int_0^T s_b(t)\exp\{-j2\pi f_l t\}dt \end{aligned}$$

Suppose that the sampling rate at the receiver is $\delta t = T/(ML)$. Then

$$\begin{aligned} Z_l &\approx \sum_{n=0}^{N-1} s(nT/(ML))\exp\{-j2\pi f_l nT/(ML)\}T/(ML) \\ &= \sum_{n=0}^{ML-1} s(nT/(ML))\exp\{-j2\pi l n/(ML)\}T/(ML) \\ &= \sum_{n=0}^{ML-1}\sum_{k=0}^{M-1} b_k \exp\{j2\pi f_k nT/(ML)\}\exp\{-j2\pi l n/(ML)\}T/(ML) \end{aligned}$$

$$
\begin{aligned}
&= \sum_{n=0}^{ML-1}\sum_{k=0}^{M-1} b_k \exp\{j2\pi(k/T)nT/(ML)\}\exp\{-j2\pi ln/(ML)\}T/(ML) \\
&= \sum_{n=0}^{ML-1}\sum_{k=0}^{M-1} b_k \exp\{j2\pi(k-l)n/(ML)\}T/(ML) \\
&= \sum_{n=0}^{ML-1} b_l T/(ML) \\
&= b_l T
\end{aligned}
$$

The calculation of Z_l can be done via the FFT.

$$
\begin{aligned}
Z_l &= \sum_{n=0}^{ML-1} s_n \exp\{-j2\pi ln/(ML)\}T/(ML) \\
&= \mathrm{FFT}\{\mathbf{s}_l\}T/(ML)
\end{aligned}
$$

where FFT is the fast Fourier transform. The first M values from the FFT are the desired decision variables.

Consider a channel with intersymbol interference. The time domain data signal is

$$
c_k = \sum_{l=0}^{M-1} b_l e^{j2\pi lk/M}, k = 0, 1, ..., M-1
$$

The cyclic prefix guard interval forces $c_{-l} = c_{M-l}, l = 1, ..., \nu$. The received signal (with intersymbol interference) is

$$
r_k = \sum_{m=0}^{\nu} c_{k-m} h_m
$$

where $h_m, m = 0, ..., \nu$ is the channel impulse response. The receiver processes the received data via

$$
\begin{aligned}
z_n &= \sum_{k=0}^{M-1} r_k e^{-j2\pi nk/M} \\
&= \sum_{k=0}^{M-1}\sum_{m=0}^{\nu} c_{k-m} h_m e^{-j2\pi nk/M} \\
&= \sum_{m=0}^{\nu} h_m \sum_{k=0}^{M-1} c_{k-m} e^{-j2\pi nk/M} \\
&= \sum_{m=0}^{\nu} h_m \sum_{u=-m}^{M-1-m} c_u e^{-j2\pi n(u+m)/M}
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{m=0}^{\nu} h_m e^{-j2\pi nm/M} \sum_{u=-m}^{M-1-m} \left[\sum_{l=0}^{M-1} b_l e^{j2\pi lu/M}\right] e^{-j2\pi nu/M} \\
&= \sum_{m=0}^{\nu} h_m e^{-j2\pi nm/M} \sum_{l=0}^{M-1} b_l \sum_{u=-m}^{M-1-m} e^{j2\pi(l-n)u/M} \\
&= \sum_{m=0}^{\nu} h_m e^{-j2\pi nm/M} b_n M \\
&= H_n b_n M
\end{aligned}
$$

where

$$
H_n = \sum_{m=0}^{\nu} h_m e^{-j2\pi nm/M}
$$

Design Example

As a possible design example consider a system where the carriers are spaced $1/T = 25$kHz apart and there are $M_d = 40$ carriers with a guard band on 12 channels on each side. In this case $M = 64$. Ignoring the prefix insertion we obtain a signal as shown in Figure 126. The signal is upsampled by inserting zeros between the samples of the original data signal and filtering. In Figure 127 we show the case where the oversampling is by a factor of 8. The resulting data signal is then filtered (using an ideal brick wall filter) and results in the desire spectrum as seen in Figure 128. The data rate possible can be calculated as follows. First, the data is encoded. Typically the code rate, r,is on the order of 1/4-1/2. Assume QPSK is used on each of the M_d carriers and each symbol has duration T seconds. The data rate is then

$$
R = 2 * M_d * r/T * \frac{M}{M+\nu}
$$

The bandwidth that is used depends on the filtering done. In theory the bandwidth could be as small as M/T Hz.

Spectrum

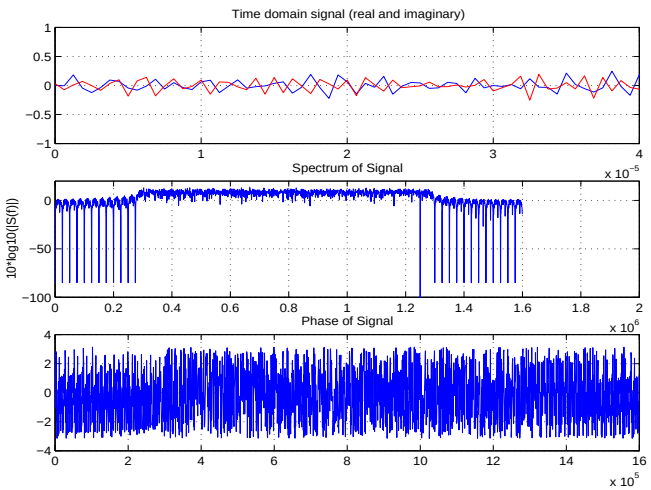


Figure 126: Output of IFFT

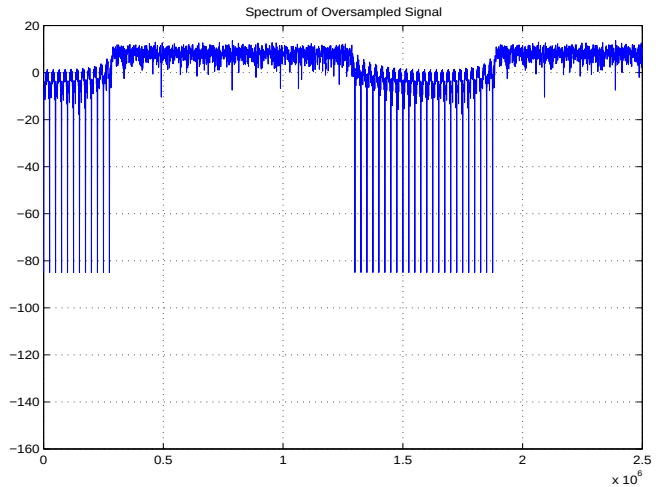


Figure 127: Output of IFFT

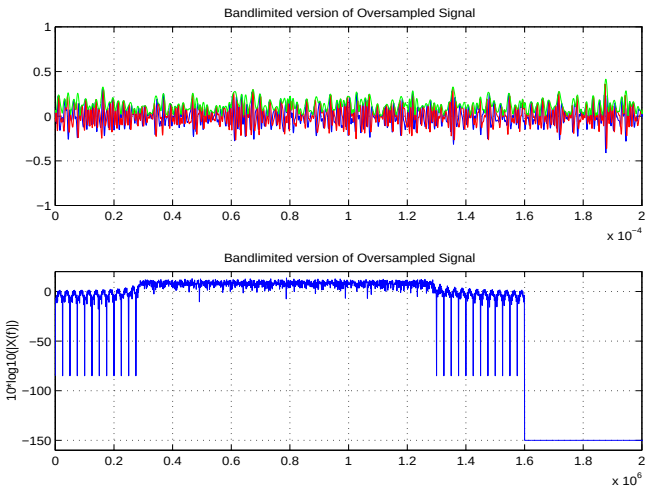
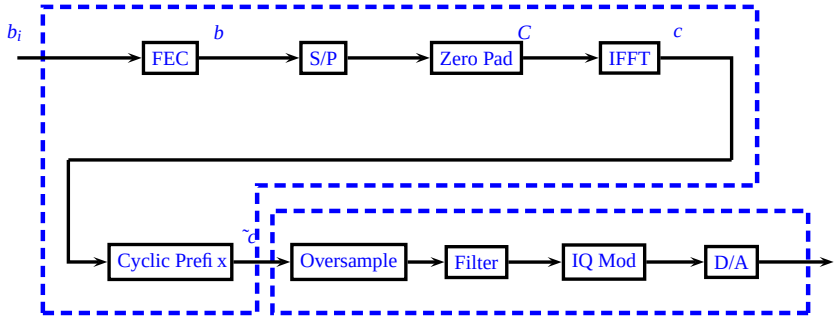


Figure 128: Output of IFFT

Transmitter Block Diagram



TI DSP
TMS 320C6711

Analog Devices 9857 (DDS/IQ Mod)

System Parameters

The DSP generates a packet of coded bits as follows. A packet of information bits of length 20480 is used to as the input to a rate 1/3 code (convolutional, turbo or LDPC) producing 61440 coded bits. Depending on the multipath of the channel the codeword is punctured to fewer bits increasing the effective rate of the code. Consider the case of no multipath initially. Then the coded bits are partitioned into blocks of length 64 which is used to determine 32 complex (frequency domain) samples. The frequency domain samples are padded with zeros on either side (16 on either side in this example). The resulting 64 complex samples are used as input to an IFFT which produces 64 complex samples. For multipath channels a cyclic prefix is added which allows us to deal with the intersymbol interference.

Eye Diagrams

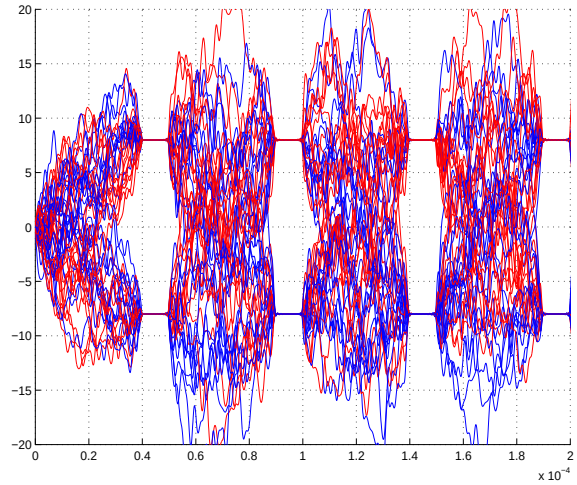


Figure 129: Eye Diagram

Eye Diagram

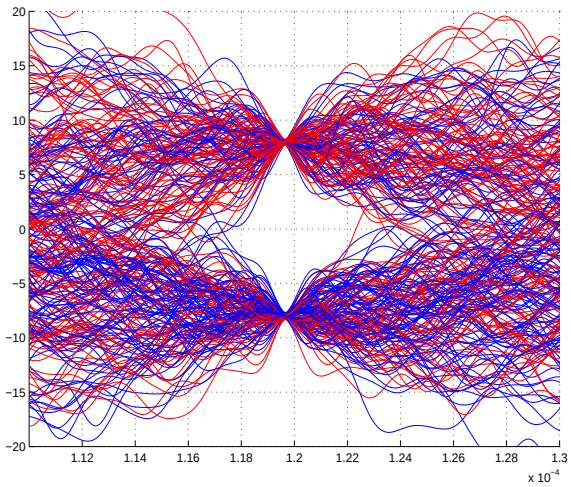


Figure 130: Eye Diagram