SYSC 5504
Principles of Digital Communication

Course Notes

Fall 2017/18

Department of Systems & Computer Engineering
Carleton University

Contents

Review of Probability Theory
Random Variables

A random variable models the outcome of an experiment which is not deterministic in nature.

Example: Coin Toss
Let X denote the outcome from flipping a coin.

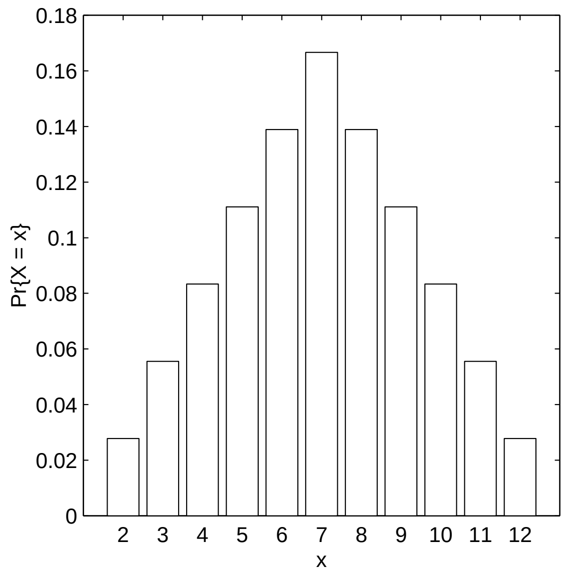
$$\Pr\{X = \text{`heads'}\} = \tfrac{1}{2}$$
$$\Pr\{X = \text{`tails'}\} = \tfrac{1}{2}$$

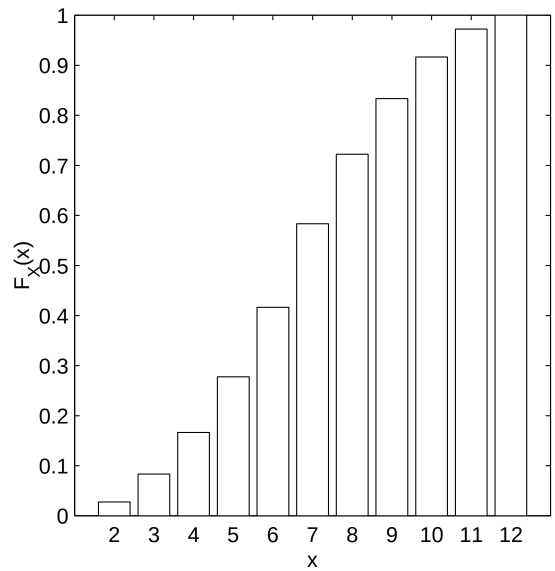
Heads or tails can occur with equal probability.

Discrete Random Variables

- takes on values from a set that is either finite or countably infinite
- takes each value with a certain probability, $\Pr\{X = x\}$
- one of the outcomes must occur, so
$$\sum_{\text{all } x} \Pr\{X = x\} = 1$$
- cumulative distribution function
$$F_X(x) = \Pr\{X \leq x\} = \sum_{\text{all } a \leq x} \Pr\{X = a\}$$

Example: Sum of the value of two dice





Joint and Conditional Probabilities

Consider two random variables, X and Y.
The *joint probability* of event $X = x$ and event $Y = y$ both occuring is denoted by $\Pr\{X = x, Y = y\}$.

Example: Flip a coin twice
Let X be the outcome of the first toss, and Y be the outcome of the second toss.

$$\Pr\{X = \text{`heads'}, Y = \text{`heads'}\} = \frac{1}{4}$$

Example: Balls in an urn

Consider an urn containing two red balls and two black balls.

Let X denote the colour of a ball drawn randomly. Then

$$\Pr\{X = \text{`red'}\} = \Pr\{X = \text{`black'}\} = \frac{1}{2}$$

Now let X and Y denote the colours of two balls drawn randomly without replacement. Then

$$\Pr\{X = \text{`red'}, Y = \text{`red'}\} = \frac{1}{6} \qquad \Pr\{X = \text{`black'}, Y = \text{`red'}\} = \frac{2}{6}$$

$$\Pr\{X = \text{`red'}, Y = \text{`black'}\} = \frac{2}{6} \qquad \Pr\{X = \text{`black'}, Y = \text{`black'}\} = \frac{1}{6}$$

The *joint cdf* of X and Y is

$$F_{X,Y}(x,y) = \Pr\{X \le x, Y \le y\} = \sum_{\text{all } a \le x} \sum_{\text{all } b \le y} \Pr\{X = a, Y = b\}$$

The *conditional probability* of event $Y = y$ occuring given that event $X = x$ has occurred is $\Pr\{Y = y | X = x\}$

Example: Coins

$$\Pr\{Y = \text{`heads'} | X = \text{`heads'}\} = \Pr\{Y = \text{`heads'}\} = \frac{1}{2}$$

Example: Balls

$$\Pr\{Y = \text{`red'} | X = \text{`red'}\} = \frac{1}{3}$$

X and Y are *independent* if

$$\Pr\{X = x, Y = y\} = \Pr\{X = x\} \Pr\{Y = y\}$$

Marginal probability:

$$\Pr\{X = x\} = \sum_{\text{all } y} \Pr\{X = x, Y = y\} \qquad \Pr\{Y = y\} = \sum_{\text{all } x} \Pr\{X = x, Y = y\}$$

Bayes' Rule:

$$\Pr\{Y = y | X = x\} = \frac{\Pr\{X = x, Y = y\}}{\Pr\{X = x\}} = \frac{\Pr\{X = x | Y = y\} \Pr\{Y = y\}}{\Pr\{X = x\}}$$

If X and Y are independendent, then

$$\Pr\{Y = y | X = x\} = \Pr\{Y = y\}$$

Moments

Expectation of a function of a random variable

$$\mathbf{E}[g(X)] = \sum_{\text{all } x} g(x) \Pr\{X = x\}$$

Mean: (Average value, expected value)

$$\mu_X = \mathbf{E}[X] = \sum_{\text{all } x} x \Pr\{X = x\}$$

Variance:

$$\sigma_X^2 = \text{Var}(X) = \mathbf{E}\left[(X - \mu_X)^2\right] = \sum_{\text{all } x} (x - \mu_X)^2 \Pr\{X = x\}$$

$$\sigma_X^2 = \mathbf{E}\left[(X - \mu_X)^2\right] = \mathbf{E}\left[X^2 - 2X\mu_X + \mu_X^2\right] = \mathbf{E}\left[X^2\right] - 2\mathbf{E}[X]\mu_X + \mu_X^2 = \mathbf{E}\left[X^2\right] - \mu_X^2$$

Covariance:

$$\text{Cov}(X,Y) = \mathbf{E}[(X - \mu_X)(Y - \mu_Y)] = \sum_{\text{all } x} \sum_{\text{all } y} (x - \mu_X)(y - \mu_Y) \Pr\{X = x, Y = y\}$$

Correlation:

$$\rho(X,Y) = \frac{\text{Cov}(X,Y)}{\sqrt{\text{Var}(X)\,\text{Var}(Y)}}$$

Some common distributions for discrete random variables

Bernoulli Distribution

Consider an experiment that can result in either a success or a failure.
e.g. transmission of a single bit – either the bit is received correctly (success), or an error occurs (failure).
Let $X = 0$ if the experiment is a success, and $X = 1$ if the experiment fails.
If p, with $0 \leq p \leq 1$, is the probability of a failure, then the probability distribution of X is:

$$\Pr\{X = 0\} = 1 - p \qquad\qquad \Pr\{X = 1\} = p$$

The random variable X is said to be a *Bernoulli* random variable.

Mean: $\mathbf{E}[X] = p$ Variance: $\mathrm{Var}(X) = p(1-p)$

NOTE: A coin toss is an example of a Bernoulli random variable with $p = \frac{1}{2}$.

Binomial Distribution

Suppose that N independent experiments are performed, each resulting in a failure with probability p and in success with probability $1-p$. If X represents the number of failures that occur in N trials, then X is said to be a *binomial* random variable with parameters (N, p). The probability distribution of X is

$$\Pr\{X = i\} = \binom{N}{i} p^i (1-p)^{N-i} \qquad i = 0, 1, 2, \ldots, N$$

Mean: $\mathbf{E}[X] = Np$ Variance: $\mathrm{Var}(X) = Np(1-p)$

Continuous Random Variables

- takes on values from a set that is uncountable
- defined by its *probability density function* (pdf), which is a non-negative function such that

$$\Pr\{X \in B\} = \int_B f_X(\alpha)\, d\alpha$$

with

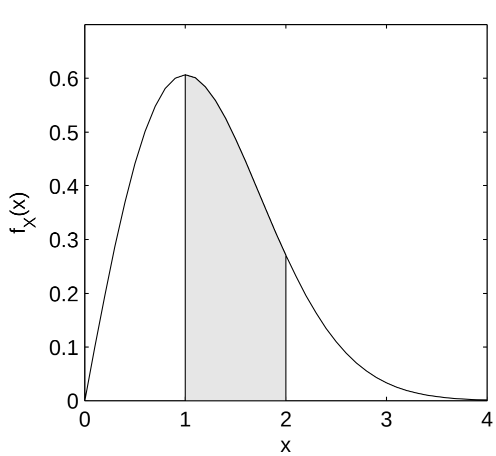
$$\int_{-\infty}^{\infty} f_X(\alpha)\, d\alpha = 1$$

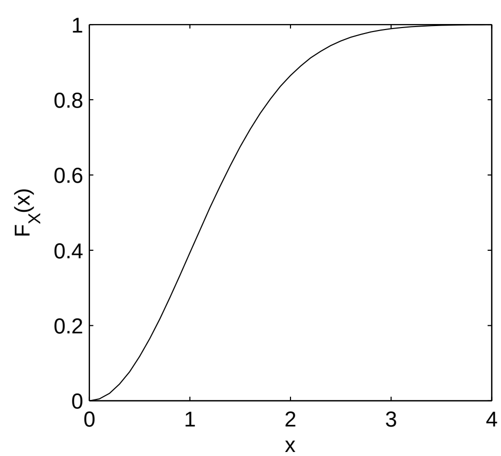
- cumulative distribution function (cdf)

$$F_X(x) = \Pr\{X \leq x\} = \int_{-\infty}^{x} f_X(\alpha)\, d\alpha$$

Example: Rayleigh distribution

$$f_X(x) = xe^{-x^2/2} \qquad\qquad F_X(x) = 1 - e^{-x^2/2}$$





Joint and Conditional Probabilities

Consider two continuous random variables, X and Y, with pdf's $f_X(x)$ and $f_Y(y)$.
The *joint pdf* of X and Y is denoted by $f_{X,Y}(x,y)$.
The *joint cdf* of X and Y is denoted by

$$F_{X,Y}(x,y) = \Pr\{X \le x, Y \le y\} = \int_{-\infty}^{x} \int_{-\infty}^{y} f_{X,Y}(\alpha,\beta)\, d\beta\, d\alpha$$

The *conditional pdf* of Y given that $X = x$ is $f_{Y|X}(y \mid X = x)$
X and Y are *independent* if

$$f_{X,Y}(x,y) = f_X(x) f_Y(y)$$

Marginal pdf:

$$f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x,y)\, dy$$

$$f_Y(y) = \int_{-\infty}^{\infty} f_{X,Y}(x,y)\, dx$$

Bayes' Rule:

$$f_{Y|X}(y \mid X = x) = \frac{f_{X,Y}(x,y)}{f_X(x)} = \frac{f_{X|Y}(x \mid Y = y) f_Y(y)}{f_X(x)}$$

If X and Y are independendent, then

$$f_{Y|X}(y \mid X = x) = f_Y(y)$$

Moments

Expectation of a function of a continuous random variable

$$\mathbf{E}[g(X)] = \int_{-\infty}^{\infty} g(x)\, f_X(x)\, dx$$

Mean: (Average value, expected value)

$$\mu_X = \mathbf{E}[X] = \int_{-\infty}^{\infty} x\, f_X(x)\, dx$$

Variance:

$$\sigma_X^2 = \mathrm{Var}(X) = \mathbf{E}\left[(X - \mu_X)^2\right] = \int_{-\infty}^{\infty} (x - \mu_X)^2\, f_X(x)\, dx$$

$$\sigma_X^2 = \mathbf{E}\left[X^2\right] - \mu_X^2$$

Covariance:

$$\mathrm{Cov}(X,Y) = \mathbf{E}[(X - \mu_X)(Y - \mu_Y)] = \int_{-\infty}^{\infty} (x - \mu_X)(y - \mu_Y)\, f_{X,Y}(x,y)\, dy\, dx$$

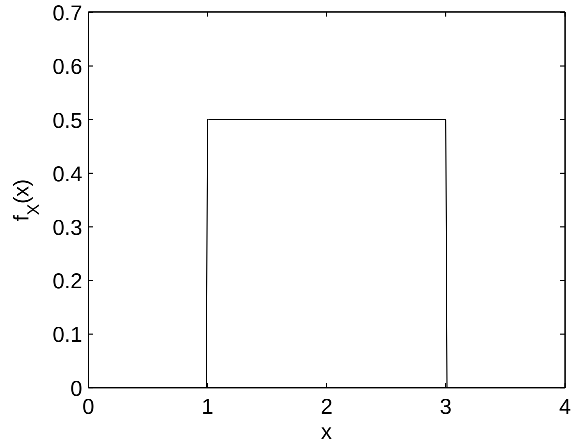
Correlation:

$$\rho(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)\,\mathrm{Var}(Y)}}$$

Some common distributions for continuous random variables

Uniform Distribution:

– A uniformly distributed random variable is defined over some range $[a, b]$.



– probability density function:

$$f_X(x) = \begin{cases} \frac{1}{b-a} & a \le x \le b \\ 0 & \text{otherwise} \end{cases}$$

– cumulative distribution function:

$$F_X(x) = \int_a^x f_X(\alpha)\, d\alpha = \frac{x-a}{b-a}$$

– mean:

$$\mathbf{E}[X] = a + \frac{b-a}{2}$$

– variance:

$$\mathrm{Var}(X) = \frac{(b-a)^2}{12}$$

Gaussian (Normal) Distribution:

– probability density function:

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma_X^2}} \exp\left\{ -\frac{(x-\mu_X)^2}{2\sigma_X^2} \right\}$$

– mean: $\mathbf{E}[X] = \mu_X$

– variance: $\mathrm{Var}(X) = \sigma_X^2$

– cumulative distribution function:

$$F_X(x) = \frac{1}{2} + \frac{1}{2}\mathrm{erf}\left(\frac{x-\mu_X}{\sqrt{2\sigma_X^2}} \right)$$

– error function

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-u^2}\, du$$

– Notes:

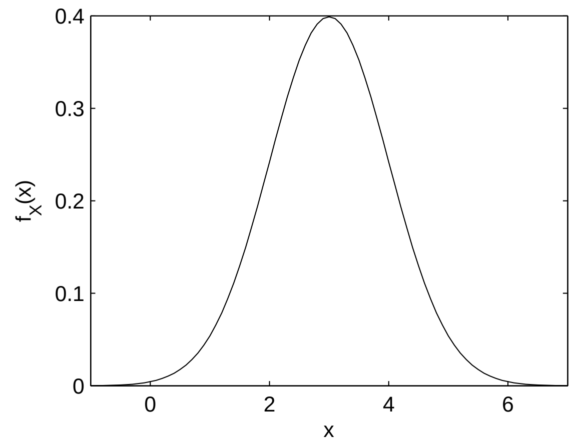
  – If two Gaussian random variables are uncorrelated, they are also independent.

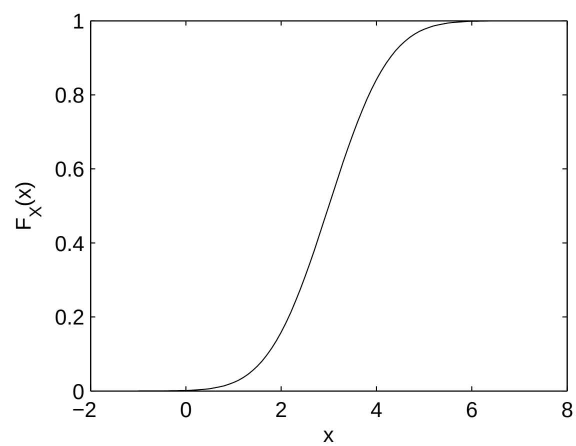
  – Any linear combination of two or more Gaussian random variables results in another Gaussian random variable.

    e.g.: If X and Y are Gaussian random variables, then

    $$Z = \alpha_1 X + \alpha_2 Y$$

    is also a Gaussian random variable, for any constants α_1 and α_2.





Notes on Complex-valued Random Variables

Consider the complex-valued random variable, Z, generated from two real-valued random variables, X and Y, by

$Z = X + jY$

- The pdf of Z is given by the joint pdf of X and Y:

  $$f_Z(z) = f_{X,Y}(\mathbf{Re}\{z\}, \mathbf{Im}\{z\})$$

- The mean of Z is

  $$\mu_Z = \mathbf{E}[Z] = \mathbf{E}[X + jY] = \mathbf{E}[X] + j\mathbf{E}[Y] = \mu_X + j\mu_Y$$

  where μ_X and μ_Y are the means of X and Y, respectively.

- The variance of Z is

  $$\sigma_Z^2 = \frac{1}{2}\mathbf{E}\left[\left|Z - \mu_Z\right|^2\right] = \frac{1}{2}\mathbf{E}\left[\left|(X - \mu_X) + j(Y - \mu_Y)\right|^2\right] = \frac{1}{2}\mathbf{E}\left[(X - \mu_X)^2 + (Y - \mu_Y)^2\right] = \frac{1}{2}(\sigma_X^2 + \sigma_Y^2)$$

  where σ_X^2 and σ_Y^2 are the variances of X and Y, respectively.

  NOTE: There is some dispute about the factor of one-half in the definition of the variance of complex random variables. Some authors (such as Proakis) use the factor, while others omit it. For analysis of communication systems inclusion of the factor is often useful. Stick with one approach and BE CONSISTENT.

- The pdf of a complex Gaussian r.v. is:

  $$f_Z(z) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} \exp\left\{-\frac{\sigma_Y^2\mathbf{Re}\{z - \mu_Z\}^2 - 2\rho\sigma_X\sigma_Y\mathbf{Re}\{z - \mu_Z\}\mathbf{Im}\{z - \mu_Z\} + \sigma_X^2\mathbf{Im}\{z - \mu_Z\}^2}{2\sigma_X^2\sigma_Y^2(1 - \rho^2)}\right\}$$

  where ρ is the correlation of X and Y,

  $$\rho = \frac{\mathrm{Cov}(X, Y)}{\sigma_X\sigma_Y}$$

  If X and Y are uncorrelated (i.e. $\rho = 0$) with equal variances (i.e. $\sigma_X^2 = \sigma_Y^2$), the pdf of Z reduces to

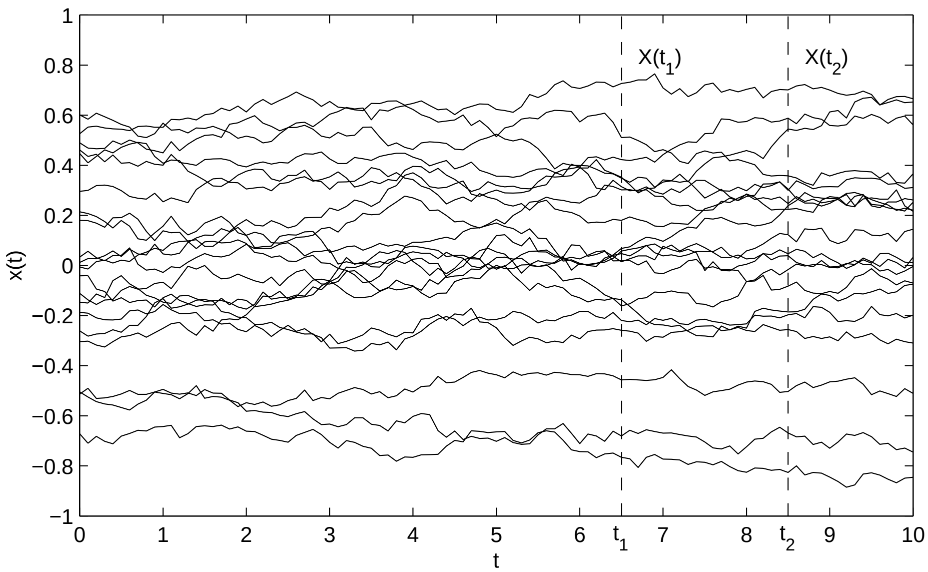
  $$f_Z(z) = \frac{1}{2\pi\sigma_Z^2} \exp\left\{-\frac{1}{2\sigma_Z^2}\left|Z - \mu_Z\right|^2\right\}$$

Review of Probability Theory
Stochastic Processes (Random Processes)

Def^n: A random process is an ensemble of waveforms (time functions) together with a probability rule that assigns a probability to each possible waveform.

- A random variable, X, can take on a single value from a set of values.
- A random process, $X(t)$, can take on a waveform from a set of waveforms.

Example:



Probability Distribution:

The value of the random process at any time, t_1, is a random variable.
i.e., $X(t_1)$ is a random variable, with some pdf $f_{X(t_1)}(x)$.
Let $X(t_1), X(t_2), \ldots, X(t_n)$ denote the random variables obtained by observing the random process $X(t)$ at times $t_1, t_2, \ldots, t_n$, for any n. The joint pdf of these observations is

$$f_{X(t_1),X(t_2),\ldots,X(t_n)}(x_1, x_2, \ldots, x_n)$$

Def^n: A random process is *stationary in the strict sense* if and only if

$$f_{X(t_1+\tau),X(t_2+\tau),\ldots,X(t_n+\tau)}(x_1, x_2, \ldots, x_n) = f_{X(t_1),X(t_2),\ldots,X(t_n)}(x_1, x_2, \ldots, x_n)$$

for any time offset, τ.

Note: For a stationary random process, the pdf does not change over time:

$$f_{X(t_1)}(x) = f_{X(t_2)}(x) \qquad \forall\, t_1, t_2$$

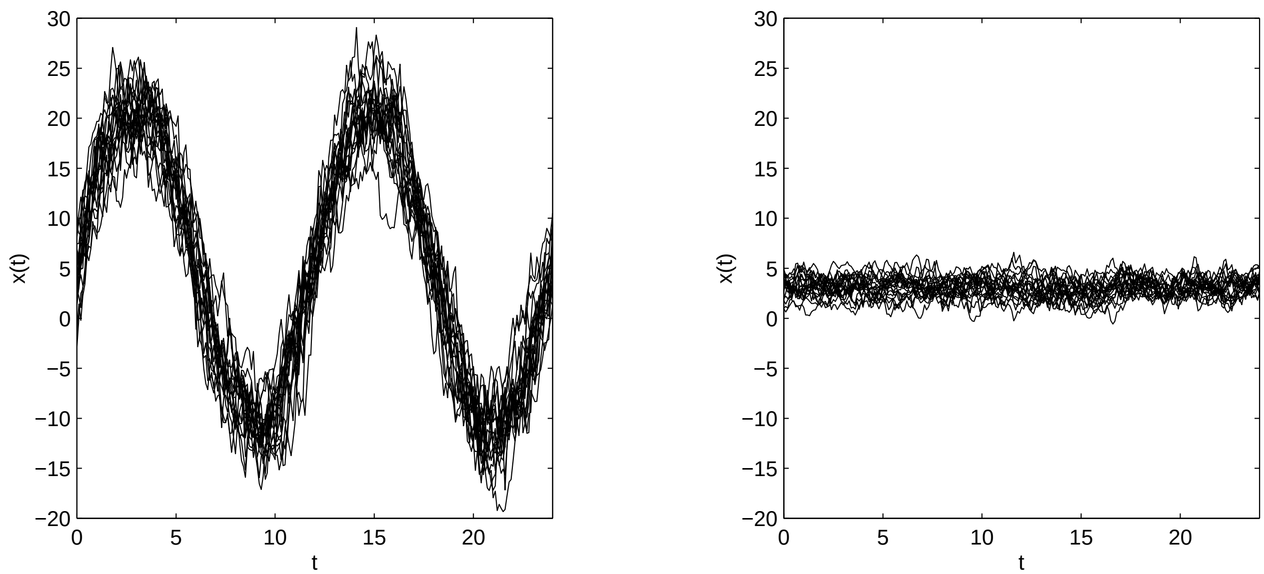
Mean: The mean of a random process at any time t_1 is

$$\mu_{X(t_1)} = \mathbf{E}[X(t_1)] = \int_{-\infty}^{\infty} x \, f_{X(t_1)}(x) \; dx$$

For a stationary random process $f_{X(t_1)}(x) = f_{X(t_2)}(x)$ for all t_1, t_2, so the mean is constant (independent of time):

$$\mu_{X(t_1)} = \mu_X$$

Example: Stationary vs. Nonstationary
The outside temperature in Ottawa is an example of a nonstationary random process, as the expected temperature in the summer is warmer than in the winter.
The temperature in your refrigerator can be modelled as a stationary random process.



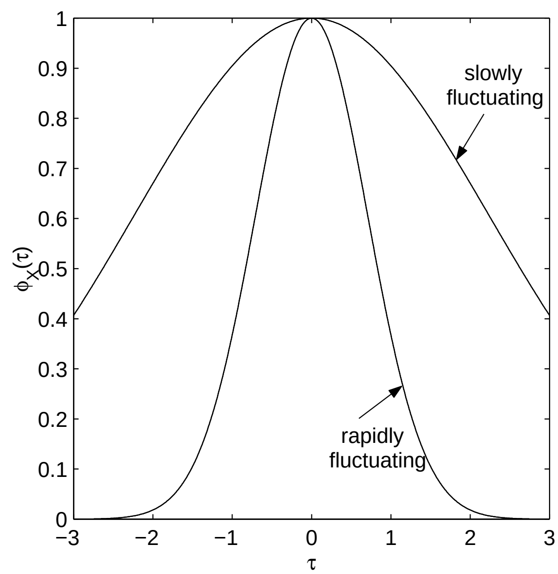
Autocorrelation function:
The autocorrelation function measures how much two observations taken at different times can vary. The autocorrelation function of a random process is defined as:

$$\phi_X(t_1, t_2) = \mathbf{E}[X(t_1)X(t_2)]$$

For a stationary random process the autocorrelation function is independent of time:

$$\phi_X(t_1, t_1 + \tau) = \phi_X(\tau)$$

Interpretation of the autocorrelation function



Note: For a complex-valued random process, $Z(t)$, the autocorrelation function is defined as:

$$\phi_Z(t_1, t_2) = \frac{1}{2}\mathbf{E}\left[Z^*(t_1)Z(t_2)\right]$$

For stationary complex random processes:

$$\phi_Z(\tau) = \frac{1}{2}\mathbf{E}\left[Z^*(t_1)Z(t_1+\tau)\right]$$

Properties of the autocorrelation function

For a stationary random process, $X(t)$:

1. The average power of the process is $\phi_X(0) = \mathbf{E}\left[X^2(t)\right]$
2. The autocorrelation function has even symmetry:

$$\phi_X(-\tau) = \phi_X(\tau)$$

3. The autocorrelation function is at a maximum at $\tau = 0$:

$$|\phi_X(\tau)| \leq \phi_X(0)$$

Proof: Note that

$$\Big(X(t+\tau) \pm X(t)\Big)^2 \geq 0$$

Taking the expected value yields

$$\mathbf{E}\left[\Big(X(t+\tau) \pm X(t)\Big)^2\right] \geq 0$$

$$\mathbf{E}\left[X^2(t+\tau) \pm 2X(t+\tau)X(t) + X^2(t)\right] \geq 0$$

$$\phi_X(0) \pm 2\phi_X(\tau) + \phi_X(0) \geq 0$$

$$\phi_X(0) \pm \phi_X(\tau) \geq 0$$

Therefore

$$-\phi_X(0) \leq \phi_X(\tau) \leq \phi_X(0)$$

Cross-correlation function

For two random processes, $X(t)$ and $Y(t)$, the cross-correlation function is

$$\phi_{XY}(t_1, t_2) = \mathbf{E}\left[X(t_1)Y(t_2)\right]$$

Wide-sense stationary

Many nonstationary random processes have the property that the mean and autocorrelation functions are independent of time. i.e.,

$$\mu_{X(t_1)} = \mu_X \text{ and } \phi_X(t_1, t_1+\tau) = \phi_X(\tau)$$

for all t_1.
Such random processes are referred to as wide-sense stationary (WSS).

Note: All strict-sense stationary random processes are also wide-sense stationary, but not all wide-sense stationary random processes are strict-sense stationary.

Gaussian random processes

When the pdf of the observations of a random process have a joint Gaussian distribution, the process is called a Gaussian random process. The pdf of the observation at time t_1 is

$$f_{X(t_1)}(x) = \frac{1}{\sqrt{2\pi\sigma_X^2(t_1)}} \exp\left\{-\frac{1}{2\sigma_X^2(t_1)}(x-\mu_X(t_1))^2\right\}$$

For WSS Gaussian random processes, this pdf is

$$f_{X(t_1)}(x) = \frac{1}{\sqrt{2\pi\sigma_X^2}} \exp\left\{-\frac{1}{2\sigma_X^2}(x-\mu_X)^2\right\}$$

Note: All WSS Gaussian random processes are also stationary in the strict sense.

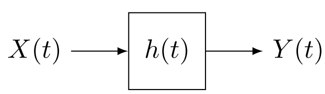
Power Spectral Density

For deterministic signals, the power spectrum is usually found by taking the Fourier transform of the signal. For stationary random processes, the power spectrum is found by taking the Fourier transform of the autocorrelation function:

$$\Phi_X(f) = \int_{-\infty}^{\infty} \phi_X(\tau) e^{-j2\pi f\tau}\, d\tau$$

Transmission of a random process through a linear filter

Consider a linear time-invariant filter with impulse response $h(t)$ and frequency response $H(f)$.



The input is the WSS random process $X(t)$, and the output is $Y(t)$. The output of the filter is related to the input through

$$Y(t) = \int_{-\infty}^{\infty} X(t-\alpha)h(\alpha)\, d\alpha$$

Mean:

$$\begin{aligned} \mathbf{E}[Y(t)] &= \int_{-\infty}^{\infty} \mathbf{E}[X(t-\alpha)]\, h(\alpha)\, d\alpha \\ &= \int_{-\infty}^{\infty} \mu_X h(\alpha)\, d\alpha \\ &= \mu_X H(0) \end{aligned}$$

Note: The mean does not depend on time.

Autocorrelation function:

$$
\begin{aligned}
\mathbf{E}\left[Y(t_1)Y(t_1+\tau)\right] &= \mathbf{E}\left[\int_{-\infty}^{\infty} X(t_1-\alpha_1)h(\alpha_1)\,d\alpha_1 \int_{-\infty}^{\infty} X(t_1+\tau-\alpha_2)h(\alpha_2)\,d\alpha_2\right] \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathbf{E}\left[X(t_1-\alpha_1)X(t_1+\tau-\alpha_2)\right] h(\alpha_1)h(\alpha_2)\,d\alpha_1\,d\alpha_2 \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \phi_X(\tau-\alpha_2+\alpha_1)h(\alpha_1)h(\alpha_2)\,d\alpha_1\,d\alpha_2 \\
&= \phi_Y(\tau)
\end{aligned}
$$

Note: The autocorrelation function does not depend on t_1. Therefore, the output of a linear time-invariant filter in response to a WSS random process is also WSS.

Power Spectral Density

$$
\begin{aligned}
\Phi_Y(f) &= \int_{-\infty}^{\infty} \phi_Y(\tau)e^{-j2\pi f\tau}\,d\tau \\
&= \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \phi_X(\tau-\alpha_2+\alpha_1)h(\alpha_1)h(\alpha_2)\,d\alpha_1\,d\alpha_2\right] e^{-j2\pi f\tau}\,d\tau \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \phi_X(\tau-\alpha_2+\alpha_1)e^{-j2\pi f\tau}\,d\tau\; h(\alpha_1)h(\alpha_2)\,d\alpha_1\,d\alpha_2 \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \Phi_X(f)e^{-j2\pi f(\alpha_2-\alpha_1)}h(\alpha_1)h(\alpha_2)\,d\alpha_1\,d\alpha_2 \\
&= \Phi_X(f)\int_{-\infty}^{\infty} h(\alpha_1)e^{j2\pi f\alpha_1}\,d\alpha_1 \int_{-\infty}^{\infty} h(\alpha_2)e^{-j2\pi f\alpha_2}\,d\alpha_2 \\
&= \Phi_X(f)H^*(f)H(f) \\
&= \Phi_X(f)\left|H(f)\right|^2
\end{aligned}
$$

White Gaussian Noise

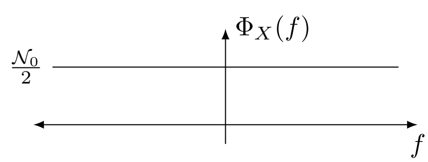
Thermal noise in communication systems is often modelled as a white Gaussian random process. This process is stationary, with a mean of zero and a double-sided power spectral density of $\mathcal{N}_0/2$. i.e.,

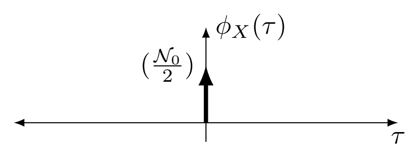
$$\Phi_X(f) = \frac{\mathcal{N}_0}{2}$$

The corresponding autocorrelation function is

$$
\begin{aligned}
\phi_X(\tau) &= \int_{-\infty}^{\infty} \Phi_X(f)e^{j2\pi f\tau}\,df \\
&= \int_{-\infty}^{\infty}\left[\frac{\mathcal{N}_0}{2}\right]e^{j2\pi f\tau}\,df \\
&= \frac{\mathcal{N}_0}{2}\delta(\tau)
\end{aligned}
$$

Observations at different times, no matter how close, are uncorrelated (and therefore independent, since the process is Gaussian).





Discrete-time Random Processes

The concept of random processes can be extended to discrete-time signals, such as a sampled version of a continuous-time random process. A discrete-time random process, X_n, is a sequence of random variables, one for each time index, n. That is, for any n_1, X_{n_1} is a random variable. This random variable itself can be either discrete or continuous (in its amplitude).

Example: The temperature at noon every day is modelled as a discrete-time random process with continuously distributed amplitude. A sequence of randomly generated bits can be modelled as a discrete-time random process with discrete amplitudes.

Mean: For a WSS discrete-time random process, the mean is

$$\mu_X = \mathbf{E}[X_n] \qquad \forall\, n$$

Autocorrlation Sequence:

For a WSS discrete-time random process, the autocorrelation sequence is

$$\phi_X(m) = \mathbf{E}[X_n X_{n+m}]$$

Note: For complex-valued discrete-time random processes, this is

$$\phi_X(m) = \tfrac{1}{2}\mathbf{E}[X_n^* X_{n+m}]$$

Power Spectral Density:

The PSD for a WSS process is

$$\Phi_X(f) = \sum_{n=-\infty}^{\infty} \phi_X(n) e^{-j2\pi f n}$$

The inverse transform is

$$\phi_X(n) = \int_{-\frac{1}{2}}^{\frac{1}{2}} \Phi_X(f) e^{j2\pi f n}\, df$$

Note: The PSD is periodic with a period of 1.

Bandpass Signal Modulation Schemes

Typically, data is communicated by modulating a carrier wave.
Either the amplitude, the phase, or the frequency is modulated, or some combination of the three.

<u>Amplitude Shift Keying (ASK)</u>

- modulate the amplitude of the carrier wave
- transmit $s_0(t)$ to represent a "0", and $s_1(t)$ to represent a "1":

$$s_0(t) = \begin{cases} 0\cos(2\pi f_c t), & 0 \le t \le T \\ 0 \quad , & \text{elsewhere} \end{cases}$$

$$s_1(t) = \begin{cases} A\cos(2\pi f_c t), & 0 \le t \le T \\ 0 \quad , & \text{elsewhere} \end{cases}$$

  f_c is the frequency of the carrier wave, and T is the duration of the symbol interval.
- ASK is also known as pulse amplitude modulation (PAM)

<u>M-ary Amplitude Shift Keying (M-ASK, M-PAM)</u>

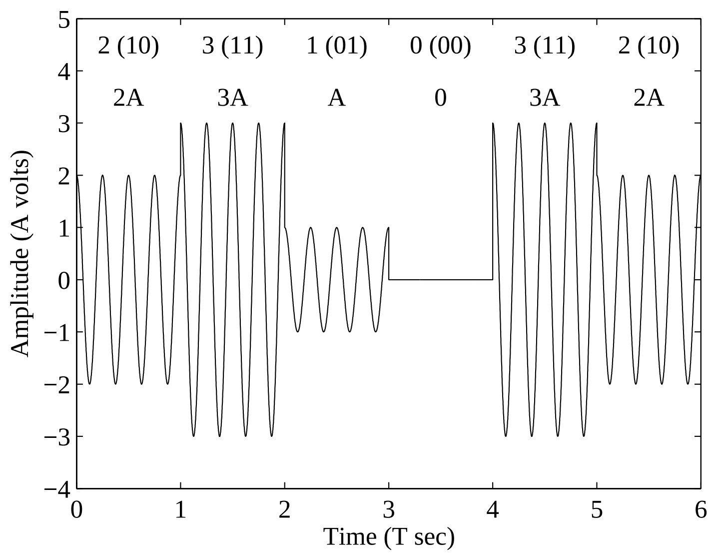
- M-ary modulation schemes are used to transmit multiple bits during a single symbol interval.
- $\log_2 M$ bits are transmitted at once. M is an integer power of 2.
- transmit $s_m(t)$ to represent $m \in \{0, 1, \ldots, M-1\}$

$$s_m(t) = \begin{cases} A_m\cos(2\pi f_c t), & 0 \le t \le T \\ 0 \quad , & \text{elsewhere} \end{cases}$$

- the amplitude of the carrier wave is

$$A_m = mA$$

<u>Example:</u> For $M = 4$, $A_m \in \{0, A, 2A, 3A\}$
The transmitted signal for the sequence 10,11,01,00,11,10 is



- M is the total number of possible signals, one of which is transmitted during each symbol interval. The transmission during each symbol interval conveys $\log_2 M$ bits of information.
- Note that a different set of amplitudes, such as $\{-3A, -A, A, 3A\}$ could be used instead, as long as the receiver knows which set of amplitudes the transmitter is using.

Phase Shift Keying (PSK)

– modulate the phase of the carrier wave

– transmit $s_0(t)$ to represent a "0", and $s_1(t)$ to represent a "1":

$$s_0(t) = \begin{cases} A\cos(2\pi f_c t), & 0 \le t \le T \\ 0, & \text{elsewhere} \end{cases}$$

$$s_1(t) = \begin{cases} A\cos(2\pi f_c t + \pi), & 0 \le t \le T \\ 0, & \text{elsewhere} \end{cases}$$

f_c is the frequency of the carrier wave, and T is the duration of the symbol interval.

– Note: binary PSK is the same as binary ASK

M-ary Phase Shift Keying (M-PSK)

– transmit $s_m(t)$ to represent $m \in \{0, 1, \ldots, M-1\}$

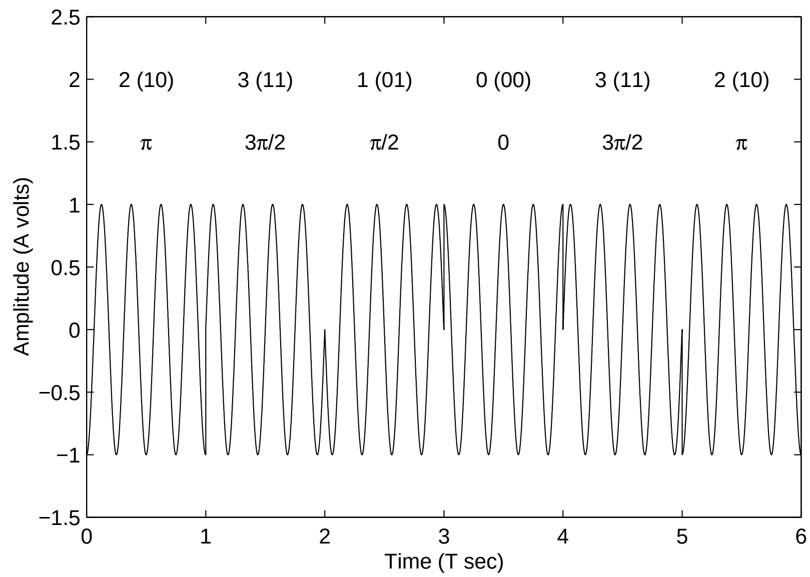
$$s_m(t) = \begin{cases} A\cos(2\pi f_c t + \theta_m), & 0 \le t \le T \\ 0, & \text{elsewhere} \end{cases}$$

– the phase of the carrier wave is

$$\theta_m = \frac{2\pi}{M} m$$

Example: For $M = 4$, $\theta_m \in \{0, \frac{1}{2}\pi, \pi, \frac{3}{2}\pi\}$

The transmitted signal for the sequence 10,11,01,00,11,10 is



Frequency Shift Keying (FSK)

– modulate the frequency of the carrier wave

– transmit $s_0(t)$ to represent a "0", and $s_1(t)$ to represent a "1":

$$s_0(t) = \begin{cases} A\cos(2\pi[f_c]t), & 0 \le t \le T \\ 0 & , \text{elsewhere} \end{cases}$$

$$s_1(t) = \begin{cases} A\cos(2\pi[f_c + \Delta f_c]t), & 0 \le t \le T \\ 0 & , \text{elsewhere} \end{cases}$$

f_c is the frequency of the carrier wave, T is the duration of the symbol interval, and Δf_c is the frequency separation.

– a good choice for the frequency separation is $\Delta f_c = \frac{1}{2T}$

M-ary Frequency Shift Keying (M-FSK)

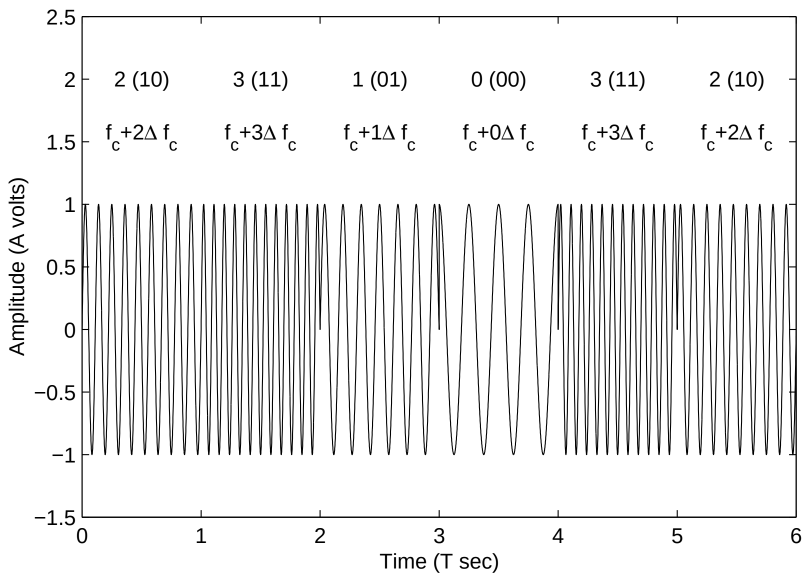
– transmit $s_m(t)$ to represent $m \in \{0, 1, \ldots, M-1\}$

$$s_m(t) = \begin{cases} A\cos(2\pi f_m t), & 0 \le t \le T \\ 0 & , \text{elsewhere} \end{cases}$$

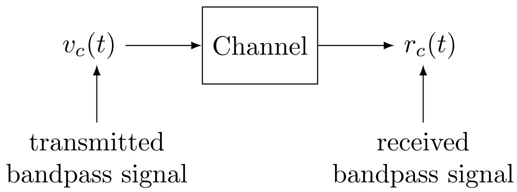
– the frequency of the carrier wave is

$$f_m = f_c + m\Delta f_c$$

Example: For $M = 4$, $f_m \in \{f_c, f_c + \Delta f_c, f_c + 2\Delta f_c, f_c + 3\Delta f_c\}$.
The transmitted signal for the sequence 10,11,01,00,11,10 is

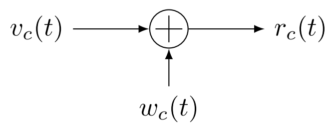


Channel Models



Ideally, $r_c(t) = v_c(t)$, but in practice the channel corrupts the transmitted signal, so $r_c(t) \neq v_c(t)$. The nature of the corruption depends on the channel model. Some common channel models are:

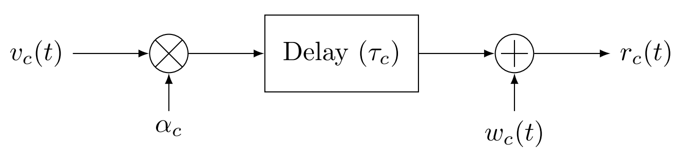
1. Additive White Gaussian Noise (AWGN) Channel



$$r_c(t) = v_c(t) + w_c(t)$$

   - $w_c(t)$ is additive noise, caused by random motion of charged particles within a resistive material (thermal noise). May be from both internal and external sources.
   - $w_c(t)$ = white, zero-mean stationary Gaussian random process
     - mean: $\mathbf{E}[w_c(t)] = 0 \; \forall \; t$
     - autocorrelation: $\phi_{w_c}(t;\tau) = \mathbf{E}[w_c(t)w_c(t+\tau)] = \frac{\mathcal{N}_0}{2}\delta(\tau)$
     - $\mathcal{N}_0$ = single-sided noise power spectral density
       - $\mathcal{N}_0 = kT_e$, where k = Boltzmann's constant (1.38×10^{-23} J/K), and T_e = equivalent noise temperature (K).

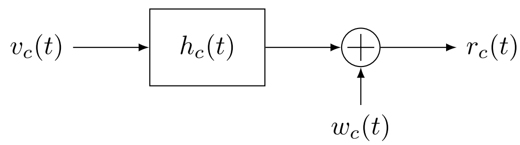
2. Propagation Delay and Attenuation



$$r_c(t) = \alpha_c v_c(t - \tau_c) + w_c(t)$$

   - signals take a finite (non-zero) amount of time to traverse the channel, and typically energy is lost in transmission, so the received signal is weaker than the transmitted one (attenuated).
   - α_c = signal attenuation
   - τ_c = signal propagation delay
   - both α_c and τ_c are a function of the propagation distance.
   - not much can be done about the attenuation, but it may need to be estimated at the receiver
   - the receiver will need to compensate for the delay, by means of symbol synchronization and carrier synchronization.
   - particularly in mobile environments α_c and τ_c can be time-variant, so
     $r_c(t) = \alpha_c(t) v_c\big(t - \tau_c(t)\big) + w_c(t)$
     e.g. frequency-flat fading due to multipath interference

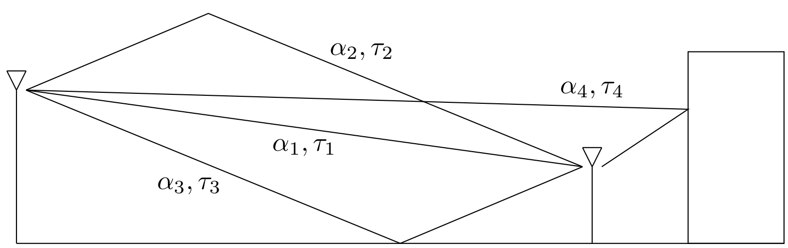
3. Linear Time-invariant Channel



$$r_c(t) = \int_{-\infty}^{\infty} v_c(t-\tau)h_c(\tau)\,d\tau + w_c(t)$$

 - $h_c(t)$ = channel impulse response
 - $H_c(f) = \mathcal{F}\{h_c(t)\}$ = channel frequency response
 - Ideally, $h_c(t) = \delta(t)$, so
 $$r_c(t) = \int_{-\infty}^{\infty} v_c(t-\tau)\delta(\tau)\,d\tau + w_c(t) = v_c(t) + w_c(t)$$
 In this case $H_c(f) = \mathcal{F}\{\delta(t)\} = 1$ (flat frequency response)

4. Multipath Interference



$$r_c(t) = \sum_i \alpha_i v_c(t-\tau_i) + w_c(t)$$

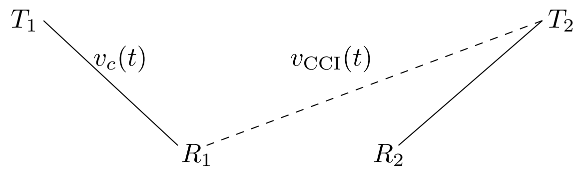
 - multiple transmission paths between transmitter and receiver
 - each path has a different attenuation (α_i) and delay (τ_i)
 - typically, these change over time
 $$r_c(t) = \sum_i \alpha_i(t) v_c\big(t - \tau_i(t)\big) + w_c(t)$$
 - Usually, there are lots of paths:
 $$r_c(t) = \int_{-\infty}^{\infty} v_c(t-\tau)h_c(\tau;t)\,d\tau + w_c(t)$$
 where
 $$h_c(\tau;t) = \sum_i \alpha_i\big(t + \tau_i(t) - \tau\big)\delta\big(\tau_i(t) - \tau\big)$$
 is the time-variant channel impulse response.

5. Adjacent Channel Interference

$$r_c(t) = v_c(t) + v_{\mathrm{ACI}}(t) + w_c(t)$$

 - $v_{\mathrm{ACI}}(t)$ = data signal used in an adjacent channel (e.g., different frequency or time slot)

6. Co-Channel Interference



$$r_c(t) = v_c(t) + v_{\mathrm{CCI}}(t) + w_c(t)$$

 - $v_{\mathrm{CCI}}(t)$ = data signal using the same channel, but usually far away.

Vector Space Concepts

Let $\{\underline{v}_0, \underline{v}_1, \ldots, \underline{v}_{M-1}\}$ be a set of K-dimensional vectors, with each vector characterized by its K components,
$\underline{v}_m = (v_{m,0}, v_{m,1}, \ldots, v_{m,K-1})$.
Definitions:

1. Inner product (dot product)

$$\underline{v}_m \bullet \underline{v}_n \triangleq \sum_{k=0}^{K-1} v_{m,k} v_{n,k}$$

2. Orthogonality

   Two vectors are *orthogonal* if their inner product is zero.

3. Norm (length)

$$\|\underline{v}_m\| \triangleq \sqrt{\underline{v}_m \bullet \underline{v}_m} = \sqrt{\sum_{k=0}^{K-1} v_{m,k}^2}$$

4. Orthonormal

   A set of vectors are *orthonormal* if the vectors are orthogonal and each vector has a norm of unity.

5. Linear independence

   A set of vectors are *linearly independent* if no one vector can be represented as a linear combination of the other vectors.

   NOTE: Orthonormal vectors are linearly independent.

6. Triangle inequality

$$\|\underline{v}_m + \underline{v}_n\| \leq \|\underline{v}_m\| + \|\underline{v}_n\|$$

7. Cauchy-Schwartz inequality

$$|\underline{v}_m \bullet \underline{v}_n| \leq \|\underline{v}_m\| \; \|\underline{v}_n\|$$

8. Vector space

   A set of K orthonormal vectors, $\{\underline{e}_k | 0 \leq k \leq K-1\}$ can be used to define a K-dimensional *vector space*. In this case the orthonormal vectors are referred to as the *basis vectors*. The *span* of the vector space is the set of all possible vectors that can be generated as a linear combination of the basis vectors. That is, the set of all vectors $\{\underline{v}\}$ given by

$$\underline{v} = \sum_{k=0}^{K-1} v_k \underline{e}_k$$

   for all $v_k \in (-\infty, \infty)$.

Signal Space Concepts

Let $\{s_0(t), s_1(t), \ldots, s_{M-1}(t)\}$ be a set of real- or complex-valued signals defined on some interval $[a, b]$.

Definitions:

1. Inner product

$$\langle s_m(t), s_n(t)\rangle \triangleq \int_a^b s_m(t) s_n^*(t)\, dt$$

2. Orthogonality

   Two signals are *orthogonal* if their inner product is zero.

3. Norm

$$\|s_m(t)\| \triangleq \sqrt{\langle s_m(t), s_m(t)\rangle} = \sqrt{\int_a^b \left|s_m(t)\right|^2 dt}$$

4. Orthonormal

   A set of signals are *orthonormal* if they are all orthogonal and their norms are all unity.

5. Linear independence

   A set of signals are *linearly independent* if no one signal can be represented as a linear combination of the other signals.

   NOTE: Orthonormal signals are linearly independent.

6. Triangle inequality

$$\|s_m(t) + s_n(t)\| \leq \|s_m(t)\| + \|s_n(t)\|$$

7. Cauchy-Schwartz inequality

$$|\langle s_m(t), s_n(t)\rangle| \leq \|s_m(t)\| \;\|s_n(t)\|$$

   or

$$\left|\int_a^b s_m(t) s_n^*(t)\, dt\right|^2 \leq \int_a^b \left|s_m(t)\right|^2 dt \;\int_a^b \left|s_n(t)\right|^2 dt$$

8. Signal space

   A set of K orthonormal signals, $\{\phi_k(t) | 0 \leq k \leq K-1\}$ can be used to define a K-dimensional *signal space*. In this case the orthonormal signals are referred to as the *basis signals*. The *span* of the signal space is the set of all possible signals that can be generated as a linear combination of the basis signals. That is, the set of all signals $\{s(t)\}$ given by

$$s(t) = \sum_{k=0}^{K-1} s_k \phi_k(t)$$

   for all $s_k \in (-\infty, \infty)$.

Gram-Schmidt Orthogonalization Procedure

For a set of M signals, $\{s_0(t), s_1(t), \ldots, s_{M-1}(t)\}$, defined on some interval $[a,b]$, find a set of K orthonormal basis signals, $\{\phi_0(t), \phi_1(t), \ldots, \phi_{K-1}(t)\}$, such that each $s_m(t)$ can be represented as a linear combination of the basis signals. Also, find the appropriate weights $\{s_{m,k}\}$ so that

$$s_m(t) = \sum_{k=0}^{K-1} s_{m,k}\phi_k(t)$$

for all $0 \le m \le M-1$.

Note: K is the minimum number of signals required. If $\{s_m(t)\}$ are linearly independent then $K = M$. Otherwise $K < M$.

Description of Algorithm

Let E_m be the energy of $s_m(t)$:

$$E_m = \left\| s_m(t) \right\|^2 = \int_a^b \left| s_m(t) \right|^2 \, dt$$

1. Find the first basis signal:

$$\phi_0(t) = \frac{s_0(t)}{\left\| s_0(t) \right\|} = \frac{s_0(t)}{\sqrt{E_0}}$$

Clearly,

$$\begin{aligned} s_0(t) &= \sqrt{E_0}\phi_0(t) \\ &= s_{0,0}\phi_0(t) \end{aligned} \qquad s_{0,k} = \begin{cases} \sqrt{E_0}\ , & \text{for } k = 0 \\ 0\ , & \text{for } 1 \le k \le K-1 \end{cases}$$

2. Find the second basis signal:
Calculate the projection of $s_1(t)$ onto $\phi_0(t)$:

$$s_{1,0} = \langle s_1(t), \phi_0(t) \rangle = \int_a^b s_1(t)\phi_0^*(t) \, dt$$

Calculate the error signal:

$$g_1(t) = s_1(t) - s_{1,0}\phi_0(t)$$

Note: $g_1(t)$ is orthogonal to $\phi_0(t)$.

The second basis signal is

$$\phi_1(t) = \frac{g_1(t)}{\left\| g_1(t) \right\|}$$

Clearly,

$$\begin{aligned} s_1(t) &= s_{1,0}\phi_0(t) + g_1(t) \\ &= s_{1,0}\phi_0(t) + s_{1,1}\phi_1(t) \end{aligned} \qquad s_{1,k} = \begin{cases} \langle s_1(t), \phi_0(t) \rangle\ , & \text{for } k = 0 \\ \left\| g_1(t) \right\|\ , & \text{for } k = 1 \\ 0\ , & \text{for } 2 \le k \le K-1 \end{cases}$$

3. Find the m^{th} basis signal:
Calculate the projection of $s_m(t)$ onto $\phi_k(t)$ for $k = 0, 1, \ldots, m-1$:

$$s_{m,k} = \langle s_m(t), \phi_k(t) \rangle = \int_a^b s_m(t)\phi_k^*(t) \, dt$$

Calculate the error signal:

$$g_m(t) = s_m(t) - \sum_{k=0}^{m-1} s_{m,k}\phi_k(t)$$

Note: $g_m(t)$ is orthogonal to $\{\phi_0(t), \phi_1(t), \ldots, \phi_{m-1}(t)\}$. i.e. $\langle g_m(t), \phi_k(t) \rangle = 0$ for all $0 \le k \le m-1$.

The m^{th} basis signal is

$$\phi_m(t) = \frac{g_m(t)}{\left\|g_m(t)\right\|}$$

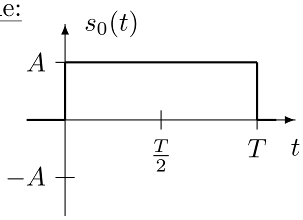
Clearly,

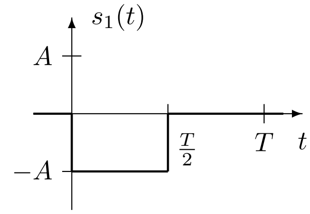
$$\begin{aligned} s_m(t) &= \sum_{k=0}^{m-1} s_{m,k}\phi_k(t) + g_m(t) \\ &= \sum_{k=0}^{m} s_{m,k}\phi_k(t) \end{aligned}$$

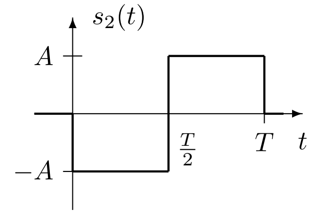
$$s_{m,k} = \begin{cases} \langle s_m(t), \phi_k(t) \rangle \, , & \text{for } 0 \le k \le m-1 \\ \left\|g_m(t)\right\| \, , & \text{for } k = m \\ 0 \, , & \text{for } m+1 \le k \le K-1 \end{cases}$$

Note: If $g_m(t) = 0$ for all $t \in [a, b]$, then $s_m(t)$ can be expressed as a linear combination of $\{s_0(t), s_1(t), \ldots, s_{m-1}(t)\}$.
Therefore $\{s_m(t) \mid 0 \le m \le M-1\}$ are not linearly independent. In this case no additional basis signal is required to represent $s_m(t)$, so $K \le M$.

Example:







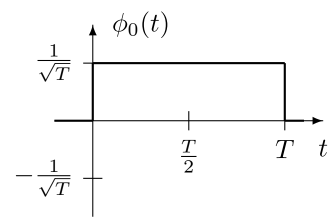
$$\begin{aligned} E_0 &= \left\|s_0(t)\right\|^2 = \int_a^b \left|s_0(t)\right|^2 \, dt \\ &= A^2 T \end{aligned}$$

$$E_1 = A^2 \frac{T}{2}$$

$$E_2 = A^2 T$$

1. Find the first basis signal:

$$\phi_0(t) = \frac{s_0(t)}{\sqrt{E_0}} = \frac{s_0(t)}{A\sqrt{T}}$$



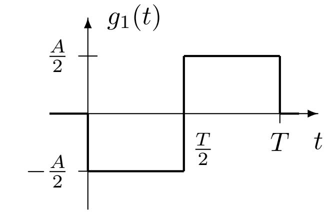
$$\begin{aligned} s_0(t) &= s_{0,0}\phi_0(t) \\ s_{0,0} &= \sqrt{E_0} = A\sqrt{T} \end{aligned}$$

$$\left\|\phi_0(t)\right\| = \sqrt{\left(\frac{1}{\sqrt{T}}\right)^2 T} = 1 \qquad \text{(normal)}$$

2. Find the second basis signal:
   Calculate the projection of $s_1(t)$ onto $\phi_0(t)$:

$$\begin{aligned} s_{1,0} &= \langle s_1(t), \phi_0(t) \rangle = \int_0^T s_1(t)\phi_0^*(t) \, dt \\ &= \int_0^{\frac{T}{2}} (-A)\frac{1}{\sqrt{T}} \, dt = \frac{-A}{\sqrt{T}}\frac{T}{2} = -\frac{A}{2}\sqrt{T} \end{aligned}$$

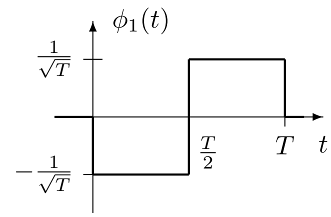
Calculate the error signal:



$$\begin{aligned} g_1(t) &= s_1(t) - s_{1,0}\phi_0(t) \\ &= \begin{cases} -A - \left(\frac{-A}{2}\sqrt{T}\right)\frac{1}{\sqrt{T}} , & 0 \le t < \frac{T}{2} \\ 0 - \left(\frac{-A}{2}\sqrt{T}\right)\frac{1}{\sqrt{T}} , & \frac{T}{2} \le t < T \end{cases} \\ &= \begin{cases} -\frac{A}{2} , & 0 \le t < \frac{T}{2} \\ \frac{A}{2} , & \frac{T}{2} \le t < T \end{cases} \end{aligned}$$

$$\left\|g_1(t)\right\| = \sqrt{\left(-\frac{A}{2}\right)^2 \frac{T}{2} + \left(\frac{A}{2}\right)^2 \frac{T}{2}} = \sqrt{\frac{A^2T}{4}} = \frac{A}{2}\sqrt{T}$$

The second basis signal is



$$\phi_1(t) = \frac{g_1(t)}{\left\|g_1(t)\right\|}$$

$$\left\|\phi_1(t)\right\| = \sqrt{\left(\frac{-1}{\sqrt{T}}\right)^2 \frac{T}{2} + \left(\frac{1}{\sqrt{T}}\right)^2 \frac{T}{2}} = 1 \qquad \text{(normal)}$$

Note: $\phi_0(t)$ and $\phi_1(t)$ are orthonormal:

$$\langle \phi_0(t), \phi_1(t) \rangle = \int_0^T \phi_0(t)\phi_1^*(t)\,dt = 0$$

$$s_1(t) = s_{1,0}\phi_0(t) + s_{1,1}\phi_1(t) \qquad \begin{aligned} s_{1,0} &= -\frac{A}{2}\sqrt{T} \\ s_{1,1} &= \frac{A}{2}\sqrt{T} \end{aligned}$$

3. Find the third basis signal:
   Calculate the projection of $s_2(t)$ onto $\phi_0(t)$:

$$\begin{aligned} s_{2,0} &= \langle s_2(t), \phi_0(t) \rangle = \int_0^T s_2(t)\phi_0^*(t)\,dt \\ &= \int_0^{\frac{T}{2}} (-A)\frac{1}{\sqrt{T}}\,dt + \int_{\frac{T}{2}}^T (A)\frac{1}{\sqrt{T}}\,dt = \frac{-A}{\sqrt{T}}\frac{T}{2} + \frac{A}{\sqrt{T}}\frac{T}{2} = 0 \end{aligned}$$

   Calculate the projection of $s_2(t)$ onto $\phi_1(t)$:

$$\begin{aligned} s_{2,1} &= \langle s_2(t), \phi_1(t) \rangle = \int_0^T s_2(t)\phi_1^*(t)\,dt \\ &= \int_0^{\frac{T}{2}} (-A)\left(\frac{-1}{\sqrt{T}}\right)\,dt + \int_{\frac{T}{2}}^T (A)\left(\frac{1}{\sqrt{T}}\right)\,dt = \frac{A}{\sqrt{T}}\frac{T}{2} + \frac{A}{\sqrt{T}}\frac{T}{2} = A\sqrt{T} \end{aligned}$$

   Calculate the error signal

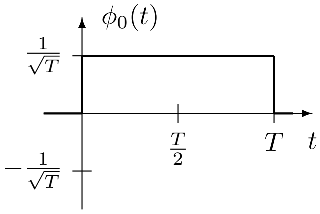
$$\begin{aligned} g_2(t) &= s_2(t) - s_{2,0}\phi_0(t) - s_{2,1}\phi_1(t) = s_2(t) - A\sqrt{T}\phi_1(t) \\ &= \begin{cases} -A - A\sqrt{T}\frac{-1}{\sqrt{T}} , & 0 \le t < \frac{T}{2} \\ A - A\sqrt{T}\frac{1}{\sqrt{T}} , & \frac{T}{2} \le t < T \end{cases} \\ &= \begin{cases} 0 , & 0 \le t < \frac{T}{2} \\ 0 , & \frac{T}{2} \le t < T \end{cases} \\ &= 0 \end{aligned}$$

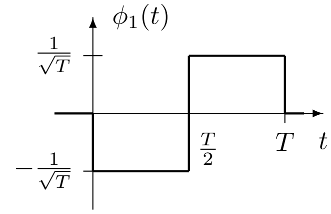
   Therefore

$$s_2(t) = s_{2,0}\phi_0(t) + s_{2,1}\phi_1(t) = A\sqrt{T}\phi_1(t)$$

   so a third basis signal is not required to represent $s_2(t)$.

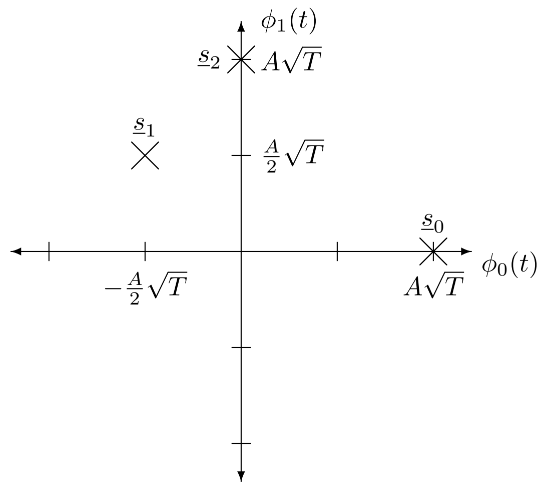
Summary:





$$
\begin{aligned}
s_0(t) &= A\sqrt{T}\phi_0(t) \\
s_1(t) &= -\tfrac{A}{2}\sqrt{T}\phi_0(t) + \tfrac{A}{2}\sqrt{T}\phi_1(t) \\
s_2(t) &= A\sqrt{T}\phi_1(t)
\end{aligned}
\qquad
\begin{aligned}
\underline{s}_0 &= \left(A\sqrt{T}, 0\right) \\
\underline{s}_1 &= \left(-\tfrac{A}{2}\sqrt{T}, \tfrac{A}{2}\sqrt{T}\right) \\
\underline{s}_2 &= \left(0, A\sqrt{T}\right)
\end{aligned}
$$

Signal Space Diagram:



Proof of the Gram-Schmidt Orthogonalization Procedure

Show that $\{\phi_0(t), \phi_1(t), \ldots, \phi_{K-1}(t)\}$ are orthonormal when determined by the Gram-Schmidt Procedure.

Assume that $\{\phi_0(t), \phi_1(t), \ldots, \phi_{k-1}(t)\}$ are orthonormal for some $k < K-1$.

Then, for any l, $0 \le l \le k-1$,

$$
\begin{aligned}
\langle \phi_k(t), \phi_l(t)\rangle &= \frac{1}{\left\|g_k(t)\right\|}\langle g_k(t), \phi_l(t)\rangle && \left(\phi_k(t) = \frac{g_k(t)}{\left\|g_k(t)\right\|}\right) \\
&= \frac{1}{\left\|g_k(t)\right\|}\left\langle \left[s_k(t) - \sum_{i=0}^{k-1} s_{k,i}\phi_i(t)\right], \phi_l(t)\right\rangle && \left(g_k(t) = s_k(t) - \sum_{i=0}^{k-1} s_{k,i}\phi_i(t)\right) \\
&= \frac{1}{\left\|g_k(t)\right\|}\left[\langle s_k(t), \phi_l(t)\rangle - \sum_{i=0}^{k-1} s_{k,i}\langle \phi_i(t), \phi_l(t)\rangle\right] \\
&= \frac{1}{\left\|g_k(t)\right\|}\left[\langle s_k(t), \phi_l(t)\rangle - \sum_{i=0}^{k-1} s_{k,i}\delta_{l-i}\right] && \left(\langle \phi_i(t), \phi_l(t)\rangle = \delta_{l-i}\right) \\
&= \frac{1}{\left\|g_k(t)\right\|}\left[s_{k,l} - s_{k,l}\right] && \left(\langle s_k(t), \phi_l(t)\rangle = s_{k,l}\right) \\
&= 0
\end{aligned}
$$

Therefore $\phi_k(t)$ is orthogonal with $\{\phi_l(t) | 0 \le l \le k-1\}$. Since

$$
\phi_k(t) = \frac{g_k(t)}{\left\|g_k(t)\right\|} \qquad \Longrightarrow \qquad \left\|\phi_k(t)\right\| = \frac{\left\|g_k(t)\right\|}{\left\|g_k(t)\right\|} = 1 ,
$$

$\{\phi_0(t), \phi_1(t), \ldots, \phi_k(t)\}$ are orthonormal.

Since $\{\phi_0(t)\}$ forms a one-dimensional orthonormal basis, it follows by recursion that

$$\{\phi_0(t), \phi_1(t), \ldots, \phi_{K-1}(t)\}$$

are orthonormal.

Also, since

$$g_m(t) = s_m(t) - \sum_{k=0}^{m-1} s_{m,k}\phi_k(t)$$

the signals are given by

$$\begin{aligned}
s_m(t) &= g_m(t) + \sum_{k=0}^{m-1} s_{m,k}\phi_k(t) \\
&= \left\| g_m(t) \right\| \phi_m(t) + \sum_{k=0}^{m-1} s_{m,k}\phi_k(t) \\
&= \sum_{k=0}^{m} s_{m,k}\phi_k(t) \qquad \text{with } s_{m,m} = \left\| g_m(t) \right\|
\end{aligned}$$

Note: When using the Gram-Schmidt Procedure it is sometimes helpful to use the following property:

$$\begin{aligned}
\left\| g_m(t) \right\|^2 &= \left\| s_m(t) - \sum_{k=0}^{m-1} s_{m,k}\phi_k(t) \right\|^2 \\
&= \int_a^b \left| s_m(t) - \sum_{k=0}^{m-1} s_{m,k}\phi_k(t) \right|^2 \, dt \\
&= \int_a^b \left| s_m(t) \right|^2 dt - \int_a^b s_m(t) \sum_{k=0}^{m-1} s_{m,k}^* \phi_k^*(t) \, dt - \int_a^b s_m^*(t) \sum_{k=0}^{m-1} s_{m,k}\phi_k(t) \, dt \\
&\qquad + \int_a^b \left| \sum_{k=0}^{m-1} s_{m,k}\phi_k(t) \right|^2 dt \\
&= \left\| s_m(t) \right\|^2 - \sum_{k=0}^{m-1} s_{m,k}^* \left\langle s_m(t), \phi_k(t) \right\rangle - \sum_{k=0}^{m-1} s_{m,k} \left\langle s_m^*(t) \phi_k^*(t) \right\rangle \\
&\qquad + \sum_{k=0}^{m-1} \sum_{l=0}^{m-1} s_{m,k} s_{m,l}^* \int_a^b \phi_k(t)\phi_l^*(t) \, dt \\
&= \left\| s_m(t) \right\|^2 - \sum_{k=0}^{m-1} s_{m,k}^* s_{m,k} - \sum_{k=0}^{m-1} s_{m,k} s_{m,k}^* + \sum_{k=0}^{m-1} \sum_{l=0}^{m-1} s_{m,k} s_{m,l}^* \delta_{l-k} \\
&= \left\| s_m(t) \right\|^2 - 2 \sum_{k=0}^{m-1} \left| s_{m,k} \right|^2 + \sum_{k=0}^{m-1} \left| s_{m,k} \right|^2 \\
&= E_m - \sum_{k=0}^{m-1} \left| s_{m,k} \right|^2
\end{aligned}$$

Geometric Representation of Bandpass Signals

M-ary Amplitude Shift Keying (Pulse Amplitude Modulation)

Transmitted Signals:

For $m \in \{0, 1, \ldots, M-1\}$

$$s_m(t) = \begin{cases} A_m \cos(2\pi f_c t), & \text{for } 0 \le t \le T \\ 0, & \text{elsewhere} \end{cases} \qquad A_m = mA$$

It is useful to express $s_m(t)$ in terms of a unit energy rectangular pulse,

$$h_T(t) = \begin{cases} \frac{1}{\sqrt{T}}, & \text{for } 0 \le t \le T \\ 0, & \text{elsewhere} \end{cases}$$

so

$$s_m(t) = A_m \sqrt{T}\, h_T(t) \cos(2\pi f_c t)$$

Basis Signals:

Only one basis signal is required to represent $\{s_m(t) \mid m = 0, 1, \ldots, M-1\}$:

$$\phi_0(t) = h_T(t)\sqrt{2}\cos(2\pi f_c t)$$

Note: $\phi_0(t)$ is normalized:

$$\begin{aligned} \left\| \phi_0(t) \right\|^2 &= \int_{-\infty}^{\infty} \left| \phi_0(t) \right|^2 dt = \int_{-\infty}^{\infty} \left| h_T(t)\sqrt{2}\cos(2\pi f_c t) \right|^2 dt \\ &= \int_0^T \left(\frac{1}{T}\right)(2)\cos^2(2\pi f_c t)\, dt = \frac{2}{T}\left[\frac{t}{2} + \frac{\sin(4\pi f_c t)}{8\pi f_c}\right]_0^T \\ &= \frac{2}{T}\left[\frac{T}{2} + \frac{\sin(4\pi f_c T)}{8\pi f_c}\right] = 1 + \frac{\sin(4\pi f_c T)}{4\pi f_c T} \\ &\cong 1 \end{aligned}$$

since $f_c T \gg 1$ (e.g., $f_c = 1$ Ghz, $T = 1\mu$s, $f_c T = 10^3$). Equality holds if $f_c T = \frac{n}{4}$ for some integer n.

Signals:

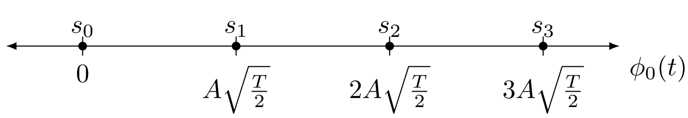
For $m \in \{0, 1, \ldots, M-1\}$

$$\begin{aligned} s_m(t) &= A_m \sqrt{\frac{T}{2}} \phi_0(t) \\ &= s_{m,0}\phi_0(t) \end{aligned}$$

with

$$s_{m,0} = A_m \sqrt{\frac{T}{2}}$$

Signal Space Diagram (M=4):



Equivalent lowpass signal:

$$\begin{aligned} s_m(t) &= A_m \sqrt{T}\, h_T(t)\cos(2\pi f_c t) \\ &= A_m \sqrt{T}\, h_T(t)\, \mathbf{Re}\left\{e^{j2\pi f_c t}\right\} \\ &= \mathbf{Re}\left\{A_m \sqrt{\frac{T}{2}} h_T(t)\sqrt{2} e^{j2\pi f_c t}\right\} \\ &= \mathbf{Re}\left\{s_{l,m}(t)\sqrt{2} e^{j2\pi f_c t}\right\} \end{aligned}$$

where

$$s_{l,m}(t) = A_m \sqrt{\frac{T}{2}} h_T(t)$$

is the equivalent lowpass signal for $s_m(t)$. Also, define

$$s_m = A_m\sqrt{\frac{T}{2}}$$

so

$$s_{l,m}(t) = s_m h_T(t)$$

M-ary Phase Shift Keying

Transmitted Signals:

For $m \in \{0, 1, \ldots, M-1\}$

$$s_m(t) = A\sqrt{T}\, h_T(t)\cos\left(2\pi f_c t + \theta_m\right) \qquad \theta_m = \frac{2\pi}{M}m$$

Basis Signals:

Note: $s_m(t)$ can also be written as

$$s_m(t) = A\sqrt{T}\, h_T(t)\cos\theta_m\cos\left(2\pi f_c t\right) - A\sqrt{T}\, h_T(t)\sin\theta_m\sin\left(2\pi f_c t\right)$$

Note: $\cos(2\pi f_c t)$ and $\sin(2\pi f_c t)$ are orthogonal over $[0, T]$:

$$\begin{aligned}
\langle\cos(2\pi f_c t), \sin(2\pi f_c t)\rangle &= \int_0^T \cos(2\pi f_c t)\sin(2\pi f_c t)\, dt \\
&= \left[\frac{\sin^2\left(2\pi f_c t\right)}{4\pi f_c}\right]_0^T \\
&= \left[\frac{\sin^2\left(2\pi f_c T\right)}{4\pi f_c}\right] \\
&\cong 0
\end{aligned}$$

since f_c is typically very large.

Therefore, only two basis signals are required to represent $\{s_m(t) \mid m = 0, 1, \ldots, M-1\}$:

$$\phi_0(t) = h_T(t)\sqrt{2}\cos\left(2\pi f_c t\right)$$
$$\phi_1(t) = -h_T(t)\sqrt{2}\sin\left(2\pi f_c t\right)$$

Signals:

For $m \in \{0, 1, \ldots, M-1\}$

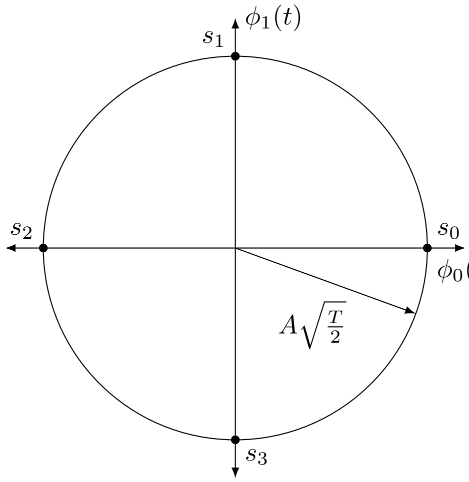
$$\begin{aligned}
s_m(t) &= A\sqrt{\frac{T}{2}}\cos\theta_m\phi_0(t) + A\sqrt{\frac{T}{2}}\sin\theta_m\phi_1(t) \\
&= s_{m,0}\phi_0(t) + s_{m,1}\phi_1(t)
\end{aligned}$$

with

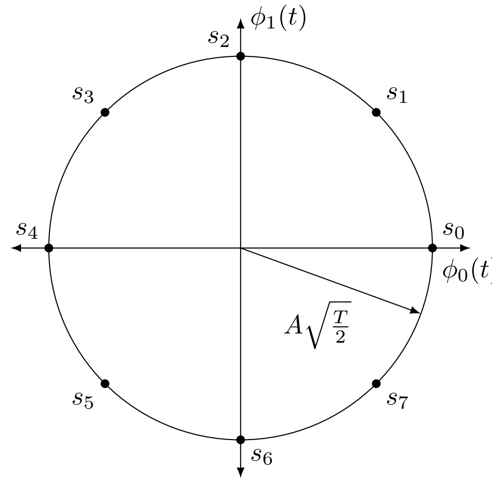
$$s_{m,0} = A\sqrt{\frac{T}{2}}\cos\theta_m \qquad s_{m,1} = A\sqrt{\frac{T}{2}}\sin\theta_m$$

Signal Space Diagram:

$M = 4$



$M = 8$



Quaternary Phase Shift Keying (QPSK)

Equivalent lowpass signal:

$$\begin{aligned} s_m(t) &= A\sqrt{T}\, h_T(t) \cos\left(2\pi f_c t + \theta_m\right) \\ &= A\sqrt{T}\, h_T(t) \mathbf{Re}\left\{ e^{j(2\pi f_c t + \theta_m)} \right\} \\ &= \mathbf{Re}\left\{ A\sqrt{\frac{T}{2}} e^{j\theta_m} h_T(t) \sqrt{2} e^{j2\pi f_c t} \right\} \\ &= \mathbf{Re}\left\{ s_{l,m}(t) \sqrt{2} e^{j2\pi f_c t} \right\} \end{aligned}$$

where

$$s_{l,m}(t) = A\sqrt{\frac{T}{2}} e^{j\theta_m} h_T(t)$$

is the equivalent lowpass signal for $s_m(t)$. Also, define

$$s_m = A\sqrt{\frac{T}{2}} e^{j\theta_m}$$

so

$$s_{l,m}(t) = s_m h_T(t)$$

Quadrature Amplitude Modulation (QAM)

Because $\cos(2\pi f_c t)$ and $\sin(2\pi f_c t)$ are orthogonal, it is possible to double the capacity of PAM.

Transmitted Signals:

For $m \in \{0, 1, \ldots, M-1\}$

$$s_m(t) = A_{c,m}\sqrt{T}\, h_T(t) \cos\left(2\pi f_c t\right) - A_{s,m}\sqrt{T}\, h_T(t) \sin\left(2\pi f_c t\right)$$

where $A_{c,m}$ is the amplitude of the cos carrier (the in-phase component), and $A_{s,m}$ is the amplitude of the sin carrier (the quadrature phase component).

Basis Signals:

$$\phi_0(t) = h_T(t)\sqrt{2}\cos\left(2\pi f_c t\right)$$
$$\phi_1(t) = -h_T(t)\sqrt{2}\sin\left(2\pi f_c t\right)$$

Signals:

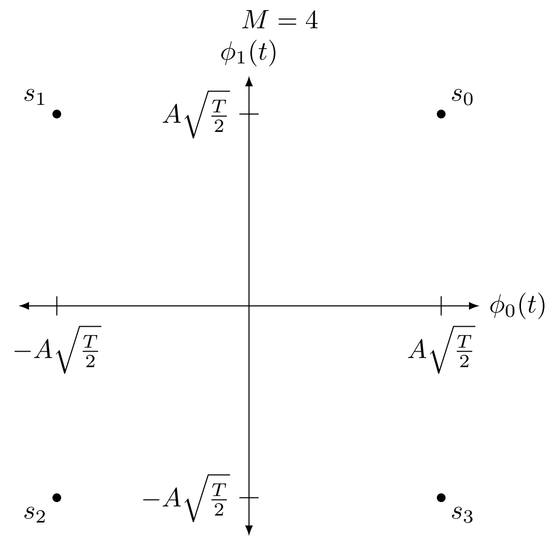
For $m \in \{0, 1, \ldots, M-1\}$

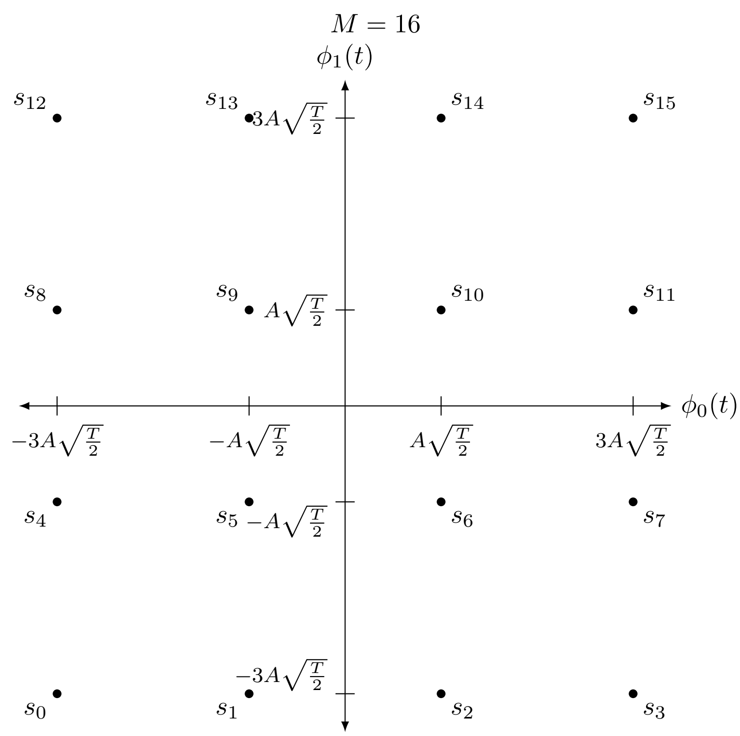
$$\begin{aligned} s_m(t) &= A_{c,m}\sqrt{\frac{T}{2}}\phi_0(t) + A_{s,m}\sqrt{\frac{T}{2}}\phi_1(t) \\ &= s_{m,0}\phi_0(t) + s_{m,1}\phi_1(t) \end{aligned}$$

with

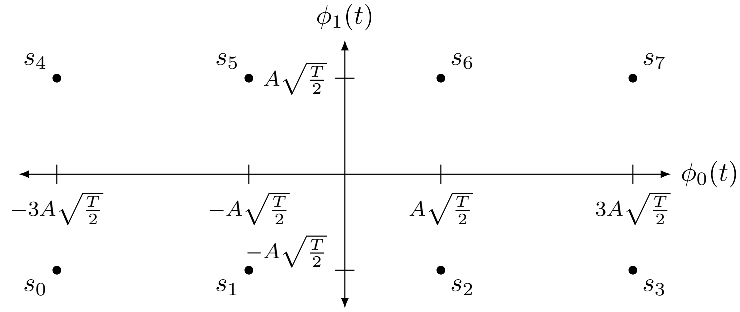
$$s_{m,0} = A_{c,m}\sqrt{\frac{T}{2}} \qquad\qquad s_{m,1} = A_{s,m}\sqrt{\frac{T}{2}}$$

Signal Space Diagram:





$M = 8$



Equivalent lowpass signal:

$$\begin{aligned} s_m(t) &= A_{c,m}\sqrt{T}\,h_T(t)\cos(2\pi f_c t) - A_{s,m}\sqrt{T}\,h_T(t)\sin(2\pi f_c t) \\ &= \mathbf{Re}\left\{[A_{c,m} + jA_{s,m}]\sqrt{T}\,h_T(t)\,[\cos(2\pi f_c t) + j\sin(2\pi f_c t)]\right\} \\ &= \mathbf{Re}\left\{[A_{c,m} + jA_{s,m}]\sqrt{\frac{T}{2}}\,h_T(t)\sqrt{2}e^{j2\pi f_c t}\right\} \\ &= \mathbf{Re}\left\{s_{l,m}(t)\sqrt{2}e^{j2\pi f_c t}\right\} \end{aligned}$$

where

$$s_{l,m}(t) = [A_{c,m} + jA_{s,m}]\sqrt{\frac{T}{2}}h_T(t)$$

is the equivalent lowpass signal for $s_m(t)$. Also, define

$$s_m = [A_{c,m} + jA_{s,m}]\sqrt{\frac{T}{2}}$$

so

$$s_{l,m}(t) = s_m h_T(t)$$

M-ary Frequency Shift Keying (FSK)

Transmitted Signals:

For $m \in \{0, 1, \ldots, M-1\}$

$$s_m(t) = A\sqrt{T}\,h_T(t)\cos(2\pi f_m t) \qquad f_m = f_c + m\Delta f_c$$

Basis Signals:

One basis signal is required for each transmitted signal:

$$\phi_m(t) = h_T(t)\sqrt{2}\cos(2\pi f_m t)$$

For proper operation, the transmitted signals should all be orthogonal:

$$\begin{aligned} \langle\phi_k(t), \phi_l(t)\rangle &= \int_{-\infty}^{\infty}\phi_k(t)\phi_l^*(t)\,dt \\ &= \int_{-\infty}^{\infty} h_T(t)\sqrt{2}\cos(2\pi f_k t)h_T^*(t)\sqrt{2}\cos(2\pi f_l t)\,dt \\ &= \int_{-\infty}^{\infty}\Big|h_T(t)\Big|^2 2\cos(2\pi f_k t)\cos(2\pi f_l t)\,dt \\ &= \frac{2}{T}\int_0^T \cos(2\pi f_k t)\cos(2\pi f_l t)\,dt \\ &= \frac{2}{T}\left[\frac{\sin(2\pi[f_k - f_l]t)}{4\pi[f_k - f_l]} + \frac{\sin(2\pi[f_k + f_l]t)}{4\pi[f_k + f_l]}\right]_0^T \\ &= \frac{2}{T}\left[\frac{\sin(2\pi[f_k - f_l]T)}{4\pi[f_k - f_l]} + \frac{\sin(2\pi[f_k + f_l]T)}{4\pi[f_k + f_l]}\right] \\ &= \frac{\sin(2\pi[f_k - f_l]T)}{2\pi[f_k - f_l]T} + \frac{\sin(2\pi[f_k + f_l]T)}{2\pi[f_k + f_l]T} \end{aligned}$$

$$
\begin{aligned}
&= \frac{\sin(2\pi\Delta f_c[k-l]T)}{2\pi\Delta f_c[k-l]T} + \frac{\sin(2\pi[2f_c+\Delta f_c(k+l)]T)}{2\pi[2f_c+\Delta f_c(k+l)]T} \\
&\cong \frac{\sin(2\pi\Delta f_c[k-l]T)}{2\pi\Delta f_c[k-l]T}
\end{aligned}
$$

To ensure orthogonality, $\Delta f_c T$ must be an integer multiple of $1/2$.

Signals:

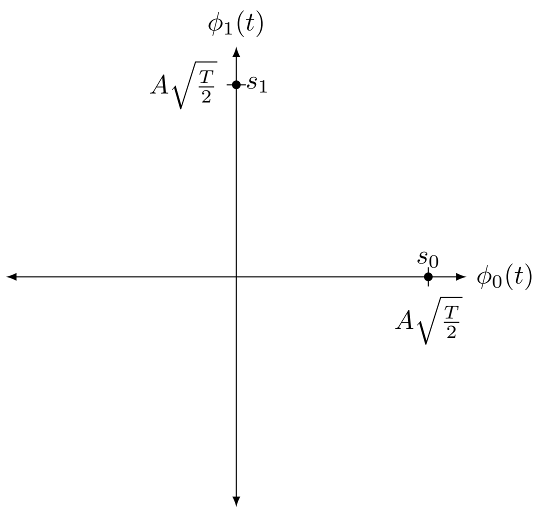
For $m \in \{0, 1, \ldots, M-1\}$

$$
\begin{aligned}
s_m(t) &= A\sqrt{\frac{T}{2}}\phi_m(t) \\
&= s_{m,m}\phi_m(t)
\end{aligned}
$$

with

$$
s_{m,n} = A\sqrt{\frac{T}{2}}\delta_{m-n} = \begin{cases} A\sqrt{\frac{T}{2}}\ , & \text{if } m = n \\ 0 \ , & \text{otherwise} \end{cases}
$$

Signal Space Diagram ($M = 2$):



Equivalent lowpass signal:

$$
\begin{aligned}
s_m(t) &= A\sqrt{T}\, h_T(t)\cos(2\pi f_m t) \\
&= A\sqrt{T}\, h_T(t)\mathbf{Re}\left\{e^{j2\pi(f_c+\Delta f_c m)t}\right\} \\
&= \mathbf{Re}\left\{A\sqrt{\frac{T}{2}}e^{j2\pi m\Delta f_c t}h_T(t)\sqrt{2}e^{j2\pi f_c t}\right\} \\
&= \mathbf{Re}\left\{s_{l,m}(t)\sqrt{2}e^{j2\pi f_c t}\right\}
\end{aligned}
$$

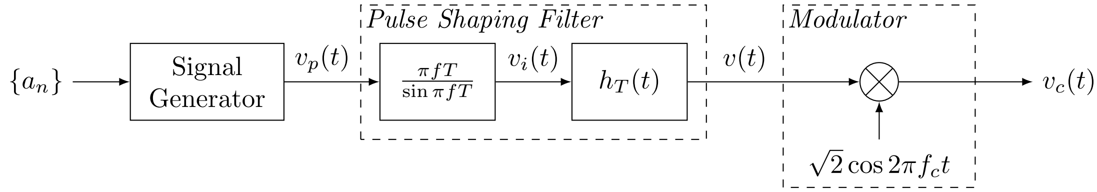
where

$$
s_{l,m}(t) = A\sqrt{\frac{T}{2}}e^{j2\pi m\Delta f_c t}h_T(t)
$$

is the equivalent lowpass signal for $s_m(t)$.

Bandpass Transmitter Structures

Amplidute-only Modulation Schemes



Data Source:

– transmit a block of N_a symbols,

$$\underline{a} = a_0\ a_1\ a_2\ \ldots\ a_{N_a-1}$$

with $a_n \in \{0, 1, \ldots, M-1\}$.

– $\underline{a}$ is referred to as the transmitted message.

– a_n is referred to as a message symbol.

$\underline{a} =$ 0 1 1 0 1

Signal Generator:

$\underline{v} =$ 0 A A 0 A

– converts message symbols into amplitudes

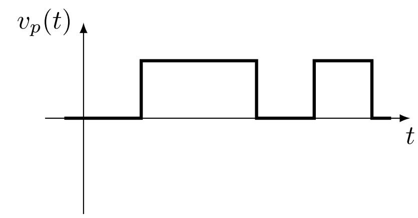
– generates rectangular pulse train

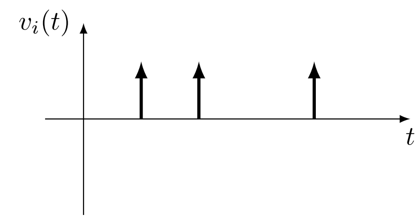
$$v_p(t) = \sum_{n=0}^{N_a-1} v_n p(t - nT)$$

– $\{v_n\}$ = transmitted amplitudes

– $p(t)$ = rectangular pulse (with unit energy)

– $1/T$ = symbol transmission rate





Pulse Shaping Filter:

– converts rectangular pulses into desired pulse shape

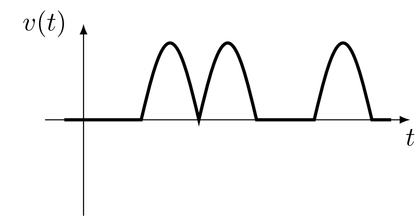
– first stage converts pulses into impulses

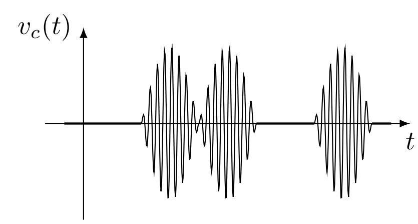
$$v_i(t) = \sum_{n=0}^{N_a-1} v_n \delta(t - nT)$$

– second stage applies pulse shape

$$v(t) = \sum_{n=0}^{N_a-1} v_n h_T(t - nT)$$

– $h_T(t)$ = desired pulse shape





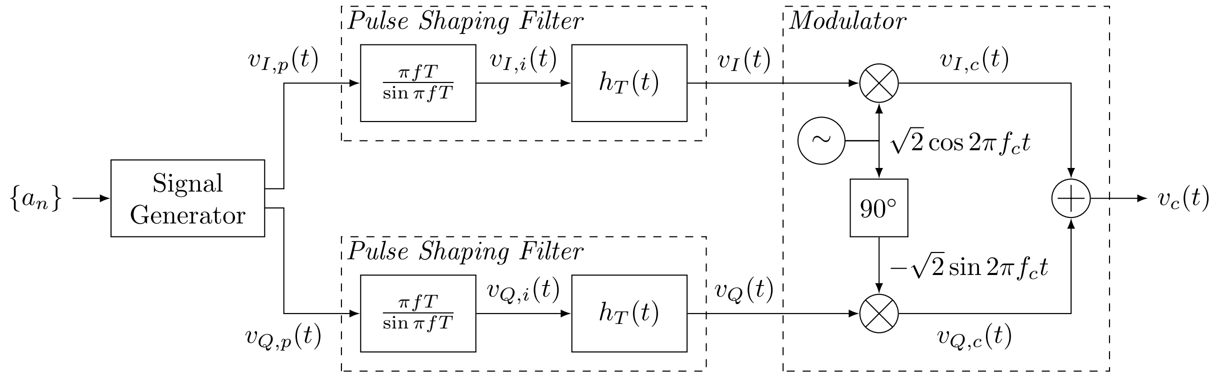
Modulator:

– modulates the amplitude of the carrier wave

$$v_c(t) = \sum_{n=0}^{N_a-1} v_n h_T(t - nT)\sqrt{2}\cos 2\pi f_c t$$

– f_c = carrier frequency

Amplitude and Phase Modulation Schemes



Signal Generator:

– generates separate amplitudes for in-phase and quadrature-phase channels

$$v_{I,p}(t) = \sum_{n=0}^{N_a-1} v_{I,n} p(t - nT) \qquad v_{Q,p}(t) = \sum_{n=0}^{N_a-1} v_{Q,n} p(t - nT)$$

Pulse Shaping Filters:

– applies desired pulse shape

$$v_I(t) = \sum_{n=0}^{N_a-1} v_{I,n} h_T(t - nT) \qquad v_Q(t) = \sum_{n=0}^{N_a-1} v_{Q,n} h_T(t - nT)$$

Modulator:

– generates in-phase and quadrature-phase carriers

$$v_{I,c}(t) = \sum_{n=0}^{N_a-1} v_{I,n} h_T(t - nT)\sqrt{2}\cos 2\pi f_c t \qquad v_{Q,c}(t) = -\sum_{n=0}^{N_a-1} v_{Q,n} h_T(t - nT)\sqrt{2}\sin 2\pi f_c t$$

– combines carriers

$$\begin{aligned} v_c(t) &= v_{I,c}(t) + v_{Q,c}(t) \\ &= \sum_{n=0}^{N_a-1} v_{I,n} h_T(t - nT)\sqrt{2}\cos 2\pi f_c t - \sum_{n=0}^{N_a-1} v_{Q,n} h_T(t - nT)\sqrt{2}\sin 2\pi f_c t \\ &= \mathbf{Re}\left\{\sum_{n=0}^{N_a-1} v_n h_T(t - nT)\sqrt{2}e^{j2\pi f_c t}\right\} \\ &= \mathbf{Re}\left\{v(t)\sqrt{2}e^{j2\pi f_c t}\right\} \end{aligned}$$

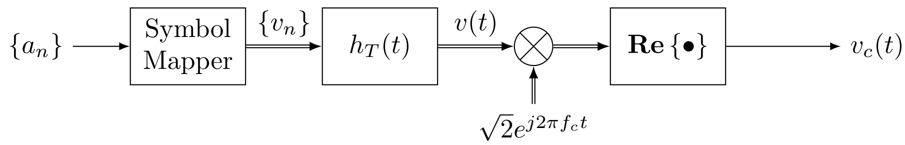
where

$$v_n = v_{I,n} + j v_{Q,n}$$

and

$$v(t) = v_I(t) + j v_Q(t) = \sum_{n=0}^{N_a-1} v_n h_T(t - nT)$$

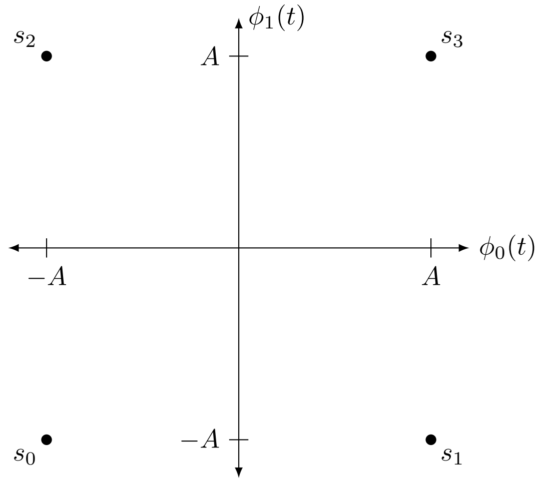
Complex lowpass equivalent representation:



Symbol Mapper:

– converts data into complex-valued points in the signal constellation

Example: 4-QAM



$$v_n \in \left\{ \begin{array}{rr} -A & -jA \\ A & -jA \\ -A & +jA \\ A & +jA \end{array} \right\}$$

Spectral Characteristics of Baseband Signals

Consider the transmitted pulse train

$$v(t) = \sum_{n=-\infty}^{\infty} v_n h_T(t - nT)$$

where T is the symbol duration (symbol period), $h_T(t)$ is the transmitted pulse shape, and $\{v_n\}$ are the transmitted symbols.

<u>Autocorrelation of $v(t)$:</u>

The autocorrelation function of $v(t)$ is:

$$\begin{aligned}
\phi_v(t;\tau) &\triangleq \mathbf{E}\left[v^*(t)v(t+\tau)\right] \\
&= \mathbf{E}\left[\left(\sum_{n=-\infty}^{\infty} v_n^* h_T^*(t-nT)\right)\left(\sum_{b=-\infty}^{\infty} v_b h_T(t+\tau-bT)\right)\right] \\
&= \sum_{n=-\infty}^{\infty}\sum_{b=-\infty}^{\infty} \mathbf{E}\left[v_n^* v_b\right] h_T^*(t-nT)h_T(t+\tau-bT) \\
&= \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} \mathbf{E}\left[v_n^* v_{n+m}\right] h_T^*(t-nT)h_T(t+\tau-(n+m)T) \\
&= \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} \phi_v(n;m) h_T^*(t-nT)h_T(t+\tau-(n+m)T)
\end{aligned}$$

Since $\underline{v}$ is typically a *stationary* discrete random process, we have

$$\phi_v(n;m) \triangleq \mathbf{E}\left[v_n^* v_{n+m}\right] = \phi_v(m)$$

That is, the autocorrelation function of $\underline{v}$ does not depend on n, so

$$\phi_v(t;\tau) = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} \phi_v(m) h_T^*(t-nT)h_T(t+\tau-(n+m)T)$$

Note that $\phi_v(t;\tau)$ depends on t, so $v(t)$ is not stationary. However, because

$$\begin{aligned}
\phi_v(t+T;\tau) &= \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} \phi_v(m) h_T^*(t+T-nT)h_T(t+T+\tau-(n+m)T) \\
&= \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} \phi_v(m) h_T^*(t-(n-1)T)h_T(t+\tau-(n-1+m)T) \\
&= \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} \phi_v(m) h_T^*(t-nT)h_T(t+\tau-(n+m)T) \\
&= \phi_v(t;\tau)
\end{aligned}$$

we observe that $v(t)$ is *cyclostationary.* To find the PSD of a cyclostationary random process, we need the time-averaged autocorrelation function:

$$\begin{aligned}
\overline{\phi}_v(\tau) &\triangleq \frac{1}{T}\int_0^T \phi_v(t;\tau)\,dt \\
&= \frac{1}{T}\int_0^T \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} \phi_v(m) h_T^*(t-nT)h_T(t+\tau-(n+m)T)\,dt \\
&= \frac{1}{T}\sum_{m=-\infty}^{\infty} \phi_v(m) \sum_{n=-\infty}^{\infty}\int_0^T h_T^*(t-nT)h_T(t+\tau-(n+m)T)\,dt \\
&= \frac{1}{T}\sum_{m=-\infty}^{\infty} \phi_v(m) \sum_{n=-\infty}^{\infty}\int_{-nT}^{-nT+T} h_T^*(u)h_T(u+\tau-mT)\,du \\
&= \frac{1}{T}\sum_{m=-\infty}^{\infty} \phi_v(m) \int_{-\infty}^{\infty} h_T^*(t)h_T(t+\tau-mT)\,dt
\end{aligned}$$

Power Spectral Density:

The power spectral density of $v(t)$ is:

$$\begin{aligned}
\overline{\Phi}_v(f) &\triangleq \int_{-\infty}^{\infty} \overline{\phi}_v(\tau) e^{-j2\pi f\tau}\, d\tau \\
&= \int_{-\infty}^{\infty} \frac{1}{T} \sum_{m=-\infty}^{\infty} \phi_v(m) \int_{-\infty}^{\infty} h_T^*(t) h_T(t+\tau-mT)\, dt\; e^{-j2\pi f\tau}\, d\tau \\
&= \frac{1}{T} \sum_{m=-\infty}^{\infty} \phi_v(m) \int_{-\infty}^{\infty} h_T^*(t) \int_{-\infty}^{\infty} h_T(t+\tau-mT) e^{-j2\pi f\tau}\, d\tau\, dt \\
&= \frac{1}{T} \sum_{m=-\infty}^{\infty} \phi_v(m) \int_{-\infty}^{\infty} h_T^*(t) \int_{-\infty}^{\infty} h_T(\alpha) e^{-j2\pi f(\alpha-t+mT)}\, d\alpha\, dt \\
&= \frac{1}{T} \sum_{m=-\infty}^{\infty} \phi_v(m) \int_{-\infty}^{\infty} h_T^*(t) \int_{-\infty}^{\infty} h_T(\alpha) e^{-j2\pi f\alpha}\, d\alpha\; e^{j2\pi f(t-mT)}\, dt \\
&= \frac{1}{T} \sum_{m=-\infty}^{\infty} \phi_v(m) \int_{-\infty}^{\infty} h_T^*(t) e^{j2\pi ft}\, dt\; H_T(f) e^{-j2\pi fmT} \\
&= \frac{1}{T} \sum_{m=-\infty}^{\infty} \phi_v(m) H_T^*(f) H_T(f) e^{-j2\pi fmT} \\
&= \left(\frac{1}{T} \sum_{m=-\infty}^{\infty} \phi_v(m) e^{-j2\pi fmT} \right) \Big| H_T(f) \Big|^2
\end{aligned}$$

From the equation for the PSD given above we see that the PSD of the transmitted signal depends not only on the pulse shape ($|H_T(f)|^2$), but also on the autocorrelation function of the transmitted symbols, $\phi_v(m)$. Some line-coding techniques, such as *alternate mark inversion* (AMI) exploit this properly by artificially introducing correlation between the symbols as part of the encoding process.

For *memoryless* coding schemes, where the symbols in $\underline{v}$ are independent (that is, v_n and v_{n+m} are independent for all $m \neq 0$), a more simple expression for the PSD can be found. Let $\mu_v = \mathbf{E}[v_n]$ be the mean of v_n, and $\sigma_v^2 = \mathbf{E}\left[|v_n|^2\right] - |\mu_v|^2$ be its variance. Then the autocorrelation function of $\underline{v}$ can be expressed as

$$\begin{aligned}
\phi_v(m) &\triangleq \mathbf{E}[v_n^* v_{n+m}] \\
&= \begin{cases} \mathbf{E}\left[|v_n|^2\right] & \text{if } m = 0 \\ \mathbf{E}[v_n^*]\, \mathbf{E}[v_{n+m}] & \text{if } m \neq 0 \end{cases} \\
&= \begin{cases} \sigma_v^2 + |\mu_v|^2 & \text{if } m = 0 \\ |\mu_v|^2 & \text{if } m \neq 0 \end{cases} \\
&= \left(\sigma_v^2 + |\mu_v|^2\right) \delta_m + |\mu_v|^2 (1 - \delta_m) \\
&= \sigma_v^2 \delta_m + |\mu_v|^2
\end{aligned}$$

The power spectral density is then given by

$$\begin{aligned}
\overline{\Phi}_v(f) &= \left(\frac{1}{T} \sum_{m=-\infty}^{\infty} \phi_v(m) e^{-j2\pi fmT} \right) \Big| H_T(f) \Big|^2 \\
&= \frac{1}{T} \sum_{m=-\infty}^{\infty} \left(\sigma_v^2 \delta_m + |\mu_v|^2\right) e^{-j2\pi fmT} \Big| H_T(f) \Big|^2 \\
&= \frac{\sigma_v^2}{T} \Big| H_T(f) \Big|^2 + \frac{|\mu_v|^2}{T} \Big| H_T(f) \Big|^2 \sum_{m=-\infty}^{\infty} e^{-j2\pi fmT} \\
&= \frac{\sigma_v^2}{T} \Big| H_T(f) \Big|^2 + \frac{|\mu_v|^2}{T} \Big| H_T(f) \Big|^2 \frac{1}{T} \sum_{m=-\infty}^{\infty} \delta\left(f - \frac{m}{T}\right) \\
&= \frac{\sigma_v^2}{T} \Big| H_T(f) \Big|^2 + \frac{|\mu_v|^2}{T^2} \sum_{m=-\infty}^{\infty} \Big| H_T\left(\frac{m}{T}\right) \Big|^2 \delta\left(f - \frac{m}{T}\right)
\end{aligned}$$

Spectral Characteristics of Bandpass Signals

The transmitted baseband signal, $v_c(t)$, can be expressed in terms of the complex lowpass equivalent signal, $v(t)$, as

$$v_c(t) = \mathbf{Re}\left\{v(t)\sqrt{2}e^{j2\pi f_c t}\right\}$$

where

$$v(t) = \sum_{n=-\infty}^{\infty} v_n h_T(t - nT)$$

To find the PSD of $v_c(t)$, we start with the autocorrelation function,

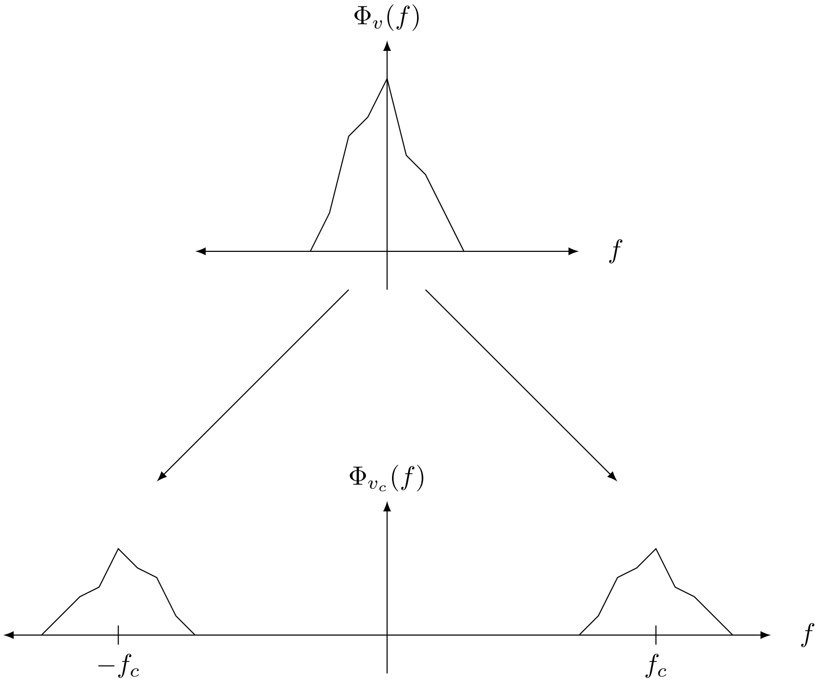
$$\begin{aligned}
\phi_{v_c}(t;\tau) &\triangleq \mathbf{E}\left[v_c(t)v_c(t+\tau)\right] \\
&= \mathbf{E}\left[\mathbf{Re}\left\{v(t)\sqrt{2}e^{j2\pi f_c t}\right\}\mathbf{Re}\left\{v(t+\tau)\sqrt{2}e^{j2\pi f_c (t+\tau)}\right\}\right] \\
&= \mathbf{E}\left[\frac{\sqrt{2}}{2}\left(v(t)e^{j2\pi f_c t} + v^*(t)e^{-j2\pi f_c t}\right)\frac{\sqrt{2}}{2}\left(v(t+\tau)e^{j2\pi f_c (t+\tau)} + v^*(t+\tau)e^{-j2\pi f_c (t+\tau)}\right)\right] \\
&= \tfrac{1}{2}\mathbf{E}\left[v(t)v(t+\tau)\right]e^{j2\pi f_c (2t+\tau)} + \tfrac{1}{2}\mathbf{E}\left[v(t)v^*(t+\tau)\right]e^{-j2\pi f_c \tau} \\
&\quad + \tfrac{1}{2}\mathbf{E}\left[v^*(t)v(t+\tau)\right]e^{j2\pi f_c \tau} + \tfrac{1}{2}\mathbf{E}\left[v^*(t)v^*(t+\tau)\right]e^{-j2\pi f_c (2t+\tau)} \\
&= \tfrac{1}{2}\phi_v(t;\tau)e^{j2\pi f_c \tau} + \tfrac{1}{2}\phi_v^*(t;\tau)e^{-j2\pi f_c \tau} \\
&\quad + \tfrac{1}{2}\mathbf{E}\left[v(t)v(t+\tau)\right]e^{j2\pi f_c (2t+\tau)} + \tfrac{1}{2}\mathbf{E}\left[v^*(t)v^*(t+\tau)\right]e^{-j2\pi f_c (2t+\tau)}
\end{aligned}$$

since the complex sinusoids vary much more quickly than $v(t)$.
Averaging the autocorrelation function over one period gives

$$\begin{aligned}
\overline{\phi}_{v_c}(\tau) &= \frac{1}{T}\int_0^T \phi_{v_c}(t;\tau)\,dt \\
&= \frac{1}{2T}\int_0^T \phi_v(t;\tau)\,dt\,e^{j2\pi f_c \tau} + \frac{1}{2T}\int_0^T \phi_v^*(t;\tau)\,dt\,e^{-j2\pi f_c \tau} \\
&\quad + \frac{1}{2T}\int_0^T \mathbf{E}\left[v(t)v(t+\tau)\right]e^{j2\pi f_c (2t+\tau)}\,dt + \frac{1}{2T}\int_0^T \mathbf{E}\left[v^*(t)v^*(t+\tau)\right]e^{-j2\pi f_c (2t+\tau)}\,dt \\
&\cong \tfrac{1}{2}\overline{\phi}_v(\tau)e^{j2\pi f_c \tau} + \tfrac{1}{2}\overline{\phi}_v^*(\tau)e^{-j2\pi f_c \tau}
\end{aligned}$$

The PSD can then be found by taking the Fourier transform of the time-averaged autocorrelation function:

$$\begin{aligned}
\overline{\Phi}_{v_c}(f) &= \int_{-\infty}^{\infty} \overline{\phi}_{v_c}(\tau)e^{-j2\pi f\tau}\,d\tau \\
&= \int_{-\infty}^{\infty}\left[\tfrac{1}{2}\overline{\phi}_v(\tau)e^{j2\pi f_c \tau} + \tfrac{1}{2}\overline{\phi}_v^*(t;\tau)e^{-j2\pi f_c \tau}\right]e^{-j2\pi f\tau}\,d\tau \\
&= \tfrac{1}{2}\int_{-\infty}^{\infty}\overline{\phi}_v(\tau)e^{-j2\pi (f-f_c)\tau}\,d\tau + \tfrac{1}{2}\int_{-\infty}^{\infty}\overline{\phi}_v^*(\tau)e^{-j2\pi (f+f_c)\tau}\,d\tau \\
&= \tfrac{1}{2}\overline{\Phi}_v(f-f_c) + \tfrac{1}{2}\overline{\Phi}_v^*(-f-f_c)
\end{aligned}$$



If the data symbols are independent (i.e. v_n and v_m are independent for all $m \neq n$) and stationary, then

$$\overline{\Phi}_v(f) = \frac{\sigma_v^2}{T}\left|H_T(f)\right|^2 + \frac{|\mu_v|^2}{T^2}\sum_{m=-\infty}^{\infty}\left|H_T\left(\frac{m}{T}\right)\right|^2 \delta\left(f - \frac{m}{T}\right)$$

where

$$\begin{aligned} H_T(f) &= \mathcal{F}\left\{h_T(t)\right\} \\ \mu_v &= \mathbf{E}\left[v_n\right] \\ \sigma_v^2 &= \mathbf{E}\left[\left|v_n\right|^2\right] - \left|\mu_v\right|^2 \end{aligned}$$

Typically $\mu_v = 0$, in which case

$$\overline{\Phi}_v(f) = \frac{\sigma_v^2}{T}\left|H_T(f)\right|^2 = \frac{E_s}{T}\left|H_T(f)\right|^2$$

where $E_s = \mathbf{E}\left[\left|v_n\right|^2\right]$ is the average transmitted energy per symbol. Therefore

$$\overline{\Phi}_{v_c}(f) = \tfrac{1}{2}\frac{E_s}{T}\left|H_T(f - f_c)\right|^2 + \tfrac{1}{2}\frac{E_s}{T}\left|H_T(-f - f_c)\right|^2$$

If $h_T(t)$ is real, then $H_T^*(-f) = H_T(f)$, so

$$\overline{\Phi}_{v_c}(f) = \tfrac{1}{2}\frac{E_s}{T}\left|H_T(f - f_c)\right|^2 + \tfrac{1}{2}\frac{E_s}{T}\left|H_T(f + f_c)\right|^2$$

Notes:
1. The PSD does not depend on the actual locations of the points in the constellation (v_v), just on the average energy, E_s.
2. The PSD does not depend on the number of points (M) in the constellation. For example, the PSD of 64-QAM is the same as BPSK (if E_s is the same).

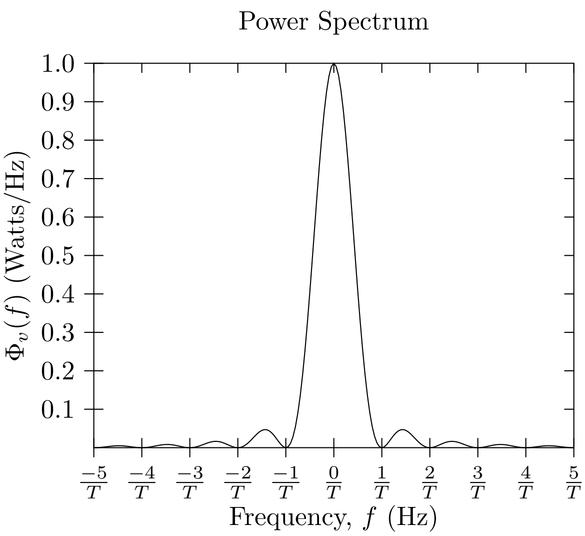
Example: For the rectangular pulse:

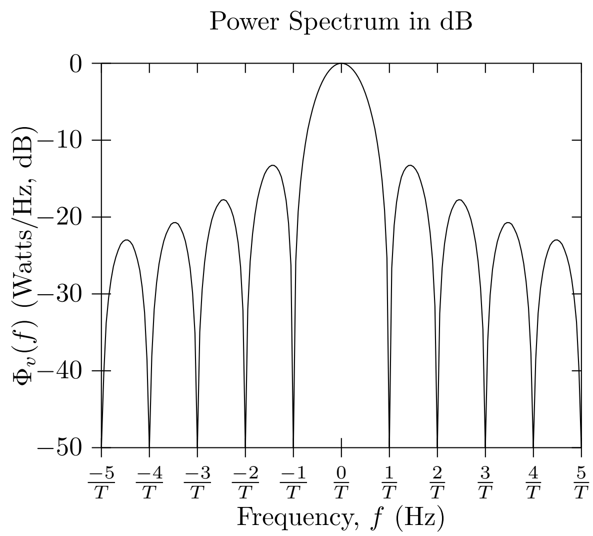
$$\begin{aligned} H_T(f) &= \int_{-\infty}^{\infty} h_T(t)e^{-j2\pi ft}\,dt \\ &= \int_0^T \frac{1}{\sqrt{T}}e^{-j2\pi ft}\,dt \\ &= \frac{1}{\sqrt{T}}\left[\frac{1}{-j2\pi f}e^{-j2\pi ft}\right]_0^T \\ &= \frac{1}{-j2\pi f\sqrt{T}}\left[e^{-j2\pi fT} - 1\right] \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{-j2\pi f\sqrt{T}} \left[e^{-j\pi f T} - e^{j\pi f T} \right] e^{-j\pi f T} \\
&= \frac{\sin(\pi f T)}{\pi f \sqrt{T}} e^{-j\pi f T} \\
&= \sqrt{T}\, \frac{\sin(\pi f T)}{\pi f T} e^{-j\pi f T}
\end{aligned}
$$

and the power spectrum is

$$
\overline{\Phi}_v(f) = E_s \left[\frac{\sin(\pi f T)}{\pi f T} \right]^2
$$





Bandwidth: Strictly speaking, the bandwidth of a signal $v(t)$, is the size of the range of frequencies for which $\Phi_v(f)$ is non-zero. Since all time-limited signals have infinite bandwidth, this strict definition is somewhat irrelevant. However, since most time-limited signals are essentially bandlimited, a number of arbitrary alternative definitions are used instead:

1. The 3dB bandwidth is the separation between the two points around f_c where the power spectrum drops below 1/2 its maximum value.
2. The null-to-null bandwidth is the separation between the two points around f_c where the power spectrum first drops to zero.
3. The 99% power bandwidth is the range of frequencies which contain 99% of the total signal power.

Example: For the rectangular pulse

1. 3dB Bandwidth $\cong \frac{0.886}{T}$ Hz
2. Null-to-Null Bandwidth $= \frac{2}{T}$ Hz
3. 99% Power Bandwidth $\cong \frac{20}{T}$ Hz

Spectral Efficiency:

Each symbol transmission last T seconds, requires a bandwidth of B Hz, and conveys $\log_2 M$ bits of information. The *spectral efficiency* of these modulation schemes is

$$\eta = \log_2 M/T/B$$

which has units of bits/second/Hz. Using the null-to-null bandwidth definition, the spectral efficiency is

$$\eta = \frac{\log_2 M}{2} \text{ bits/second/Hz}$$

For binary PSK ($M = 2$) the spectral efficiency is 1/2 bits/second/Hz, and for quaternary ($M = 4$) PSK the spectral efficiency is 1 bit/second/Hz. PAM and QAM have the same spectral efficency as PSK for the same values of M.

To reduce the bandwidth, different pulse shapes can be used. Consider this pulse:

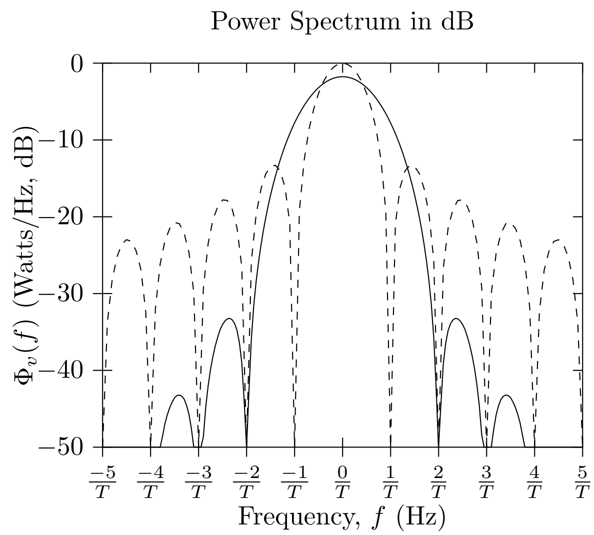
$$h_T(t) = \sqrt{\frac{2}{3T}}\left[1 + \cos\frac{2\pi}{T}\left(t - \frac{T}{2}\right)\right] \qquad 0 \le t \le T$$

Its Fourier transform is

$$H_T(f) = \sqrt{\frac{2T}{3}}\frac{\sin(\pi f T)}{\pi f T(1 - f^2T^2)}e^{-j\pi f T}$$

and its power spectrum is

$$\left|H_T(f)\right|^2 = \frac{2T}{3}\frac{\sin^2(\pi f T)}{[\pi f T(1 - f^2T^2)]^2}$$



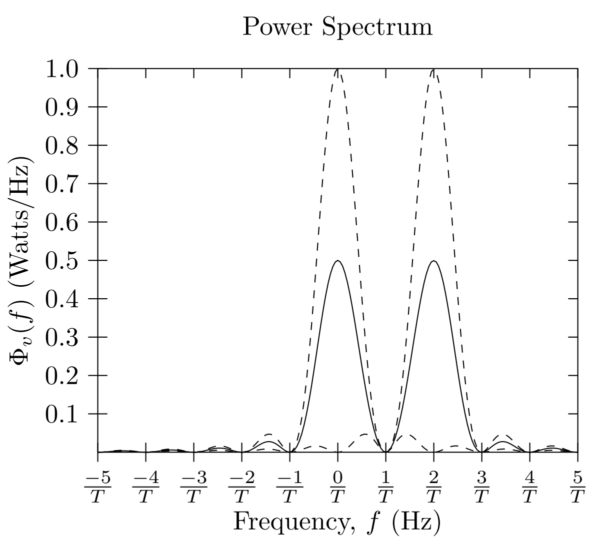
Bandwidth:

1. 3dB Bandwidth $\cong \frac{1.446}{T}$ Hz
2. Null-to-Null Bandwidth $= \frac{4}{T}$ Hz
3. 99% Power Bandwidth $= \frac{2\sqrt{2}}{T}$ Hz

Frequency Shift Keying:

When FSK is used, the PSD is well-approximated by

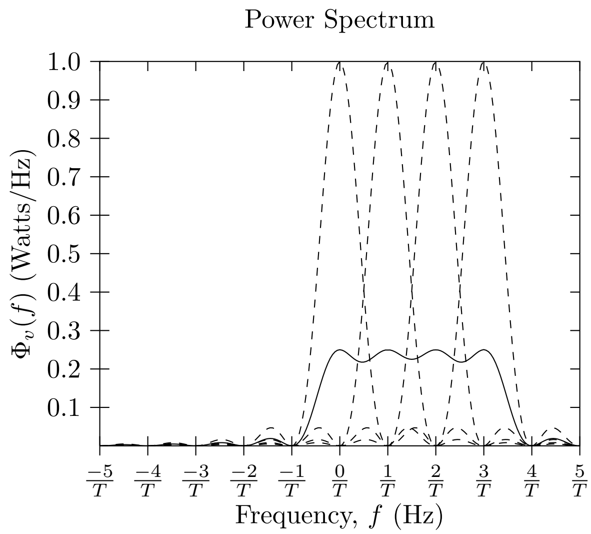
$$\left|\Phi_v(f)\right|^2 \cong \frac{E_s}{MT}\sum_{m=0}^{M-1}\left|H_T(f - m\Delta f_c)\right|^2$$



Bandwidth: The bandwidth depends on the frequency separation, Δf_c, and on the number of signals, M. For the spectrum shown above with $\Delta f_c = 2/T$ and $M = 2$, the null-to-null bandwidth is $4/T$. In general, the null-to-null bandwidth is

$$B = \frac{1}{T}\left[2 + (M-1)\Delta f_c T\right]$$

Although the smallest choice for Δf_c is $\frac{1}{2T}$ to ensure the signals are orthogonal, simple, noncoherent receivers require Δf_c to be at least as large as $1/T$. For $M = 4$, the bandwidth is $5/T$ and the resulting spectrum is



Spectral Efficiency:

The spectral efficiency of FSK is

$$\begin{aligned}\eta &= \log_2 M \text{ bits per symbol } / \; T \text{ seconds per symbol } / \; B \text{ Hz} \\ &= \log_2 M/T/\left(\frac{1}{T}\left[2 + (M-1)\Delta f_c T\right]\right) \\ &= \frac{\log_2 M}{2 + (M-1)\Delta f_c T} \text{ bits/second/Hz}\end{aligned}$$

When $\Delta f_c = 1/T$, the spectral efficiency of binary FSK is $1/3$ bits/second/Hz, and of 4-FSK is $2/5$ bits/second/Hz.

The Matched Filter

Suppose a signal, $s_c(t)$, is transmitted over an additive white Gaussian noise (AWGN) channel. The received signal is

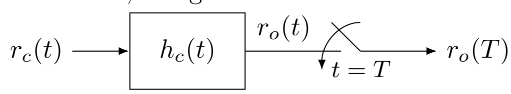
$$r_c(t) = s_c(t) + w_c(t)$$

where $w_c(t)$ is the noise component.

<u>AWGN:</u> By definition, $w_c(t)$ is a stationary random process with the following properties:

- Gaussian probability distribution at all time, t.
- zero mean $\Rightarrow \mu_{w_c} = \mathbf{E}\left[w_c(t)\right] = 0$
- white noise $\Rightarrow \phi_{w_c}(\tau) = \mathbf{E}\left[w_c(t)w_c(t+\tau)\right] = \frac{\mathcal{N}_0}{2}\delta(\tau)$

The objective is to detect the signal in the noise, using the receiver shown below:



The received signal is filtered then sampled at some observation time, T. The output of the filter, which has an impulse response of $h_c(t)$, is

$$\begin{aligned} r_o(t) &= r_c(t) \circledast h_c(t) \\ &= \left[s_c(t) + w_c(t)\right] \circledast h_c(t) \\ &= s_c(t) \circledast h_c(t) + w_c(t) \circledast h_c(t) \\ &= s_o(t) + w_o(t) \end{aligned}$$

where $\circledast$ denotes convolution, $s_o(t)$ is the data signal component of the filtered signal, and $w_o(t)$ is the noise component of the filtered signal. To reduce the effects of the noise, we want the filter to make the signal power to be considerably greater than the noise power at the output. In particular, we want to maximize the signal-to-noise ratio (SNR) of the output at time $t = T$. Therefore, we must find $h_c(t)$ such that the SNR

$$\gamma = \frac{\left|s_o(T)\right|^2}{\mathbf{E}\left[\left|w_o(T)\right|^2\right]}$$

is maximized.

<u>Signal Power:</u>

The filtered data signal is

$$s_o(t) = \int_{-\infty}^{\infty} s_c(t-\tau)h_c(\tau)\,d\tau = \int_{-\infty}^{\infty} S_c(f)H_c(f)e^{j2\pi ft}\,df$$

where $H_c(f)$ is the frequency response of the filter. The instantaneous signal power at time $t = T$ is

$$\left|s_o(T)\right|^2 = \left|\int_{-\infty}^{\infty} S_c(f)H_c(f)e^{j2\pi fT}\,df\right|^2$$

<u>Noise Power:</u>

The filtered noise component is

$$w_o(t) = \int_{-\infty}^{\infty} w_c(t-\tau)h_c(\tau)\,d\tau$$

The average noise power at time $t = T$ is

$$\begin{aligned} \mathbf{E}\left[\left|w_o(T)\right|^2\right] &= \mathbf{E}\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} w_c(T-\tau_1)w_c^*(T-\tau_2)h_c(\tau_1)h_c^*(\tau_2)\,d\tau_1\,d\tau_2\right] \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathbf{E}\left[w_c(T-\tau_1)w_c^*(T-\tau_2)\right]h_c(\tau_1)h_c^*(\tau_2)\,d\tau_1\,d\tau_2 \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left[\frac{\mathcal{N}_0}{2}\delta(\tau_1-\tau_2)\right]h_c(\tau_1)h_c^*(\tau_2)\,d\tau_1\,d\tau_2 \\ &= \frac{\mathcal{N}_0}{2}\int_{-\infty}^{\infty} h_c(\tau_2)h_c^*(\tau_2)\,d\tau_2 \\ &= \frac{\mathcal{N}_0}{2}\int_{-\infty}^{\infty} \left|h_c(\tau)\right|^2\,d\tau \\ &= \frac{\mathcal{N}_0}{2}\int_{-\infty}^{\infty} \left|H_c(f)\right|^2\,df \end{aligned}$$

The signal-to-noise ratio is then

$$\gamma = \frac{\left|\int_{-\infty}^{\infty} S_c(f)H_c(f)e^{j2\pi fT}\,df\right|^2}{\frac{\mathcal{N}_0}{2}\int_{-\infty}^{\infty}\left|H_c(f)\right|^2\,df}$$

We must find $H_c(f)$ which maximizes γ.

Cauchy-Schwarz Inequality:

For any two finite-energy functions, $g_1(x)$ and $g_2(x)$,

$$\left|\int_{-\infty}^{\infty} g_1(x)g_2(x)\,dx\right|^2 \le \int_{-\infty}^{\infty}\left|g_1(x)\right|^2\,dx \int_{-\infty}^{\infty}\left|g_2(x)\right|^2\,dx$$

with equality holding if and only if $g_2(x) = Kg_1^*(x)$ where K is an arbitrary constant.

Using the Cauchy-Schwarz inequality with $g_1(f) = S_c(f)e^{j2\pi fT}$ and $g_2(f) = H_c(f)$, we have

$$\left|\int_{-\infty}^{\infty} S_c(f)H_c(f)e^{j2\pi fT}\,df\right|^2 \le \int_{-\infty}^{\infty}\left|S_c(f)\right|^2\,df \int_{-\infty}^{\infty}\left|H_c(f)\right|^2\,df\ .$$

Therefore

$$\gamma \le \frac{\int_{-\infty}^{\infty}\left|S_c(f)\right|^2\,df\int_{-\infty}^{\infty}\left|H_c(f)\right|^2\,df}{\frac{\mathcal{N}_0}{2}\int_{-\infty}^{\infty}\left|H_c(f)\right|^2\,df} = \frac{2}{\mathcal{N}_0}\int_{-\infty}^{\infty}\left|S_c(f)\right|^2\,df$$

To find the transfer function of the filter which maximizes γ, it is necessary to find $H_{c,\text{opt}}(f)$ which satisfies the equality condition of the Cauchy-Schwarz inequality. This is

$$H_{c,\text{opt}}(f) = KS_c^*(f)e^{-j2\pi fT}$$

The corresponding impulse response is:

$$\begin{aligned} h_{c,\text{opt}}(t) &= K\int_{-\infty}^{\infty} S_c^*(f)e^{-j2\pi f(T-t)}\,df \\ &= Ks_c^*(T-t) \end{aligned}$$

For a real signal $s_c(t)$, the impulse response is

$$h_{c,\text{opt}}(t) = Ks_c(T-t)$$

The impulse response of the optimum filter is a time-reversed and delayed version of $s_c(t)$, with an arbitrary scaling factor. Such a filter is called a *matched filter*. When a matched filter is used, the maximum possible SNR is realized:

$$\gamma_{\max} = \frac{2}{\mathcal{N}_0}\int_{-\infty}^{\infty}\left|S_c(f)\right|^2\,df = \frac{E_S}{\mathcal{N}_0/2}$$

where E_S is the energy of $s_c(t)$.

Optimal Receivers for the AWGN Channel

Transmitter Model:

Consider a generic M-ary communication system where signal $s_m(t)$ is used to convey symbol $m \in \mathcal{M}$, where $\mathcal{M} = \{0, 1, \ldots, M-1\}$ is the symbol alphabet. The set of signals, $\{s_m(t) \mid m \in \mathcal{M}\}$, can be represented with K orthonormal basis signals $\{\phi_k(t) \mid k = 0, 1, \ldots, K-1\}$, with

$$s_m(t) = \sum_{k=0}^{K-1} s_{m,k}\phi_k(t)$$

where the weights are

$$s_{m,k} = \langle s_m(t), \phi_k(t) \rangle = \int_0^T s_m(t)\phi_k(t)\,dt$$

for all $m \in \mathcal{M}$ and $k \in \{0, 1, \ldots, K-1\}$.

Note: The Gram-Schmidt procedure not only describes how to find the basis signals and the weights, but also proves that a set of basis signals exists for any set of finite-energy data signals.

Additive White Gaussian Noise (AWGN) Channel Model:

Suppose symbol $m \in \mathcal{M}$ was transmitted. The received signal is represented by

$$r_c(t) = s_m(t) + w_c(t)$$

where

$s_m(t) =$ transmitted data signal

$w_c(t) =$ additive white Gaussian noise signal

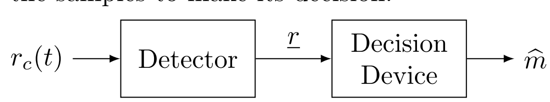
- stationary random process
- Gaussian distribution
- zero mean $\Rightarrow \mu_w = \mathbf{E}\left[w_c(t)\right] = 0$
- white noise $\Rightarrow \phi_w(\tau) = \mathbf{E}\left[w_c(t)w_c(t+\tau)\right] = \frac{\mathcal{N}_0}{2}\delta(\tau)$

Receiver:

The purpose of the receiver is to determine the transmitted symbol, m, based on observations of $r_c(t)$. Because of uncertainty introduced by the noise it is impossible to guarantee that the receiver will be able to correctly determine the transmitted symbol.

Optimal Receiver: An *optimal receiver* is one that is designed to minimize the probability that a decision error occurs. There exists no other receiver structure that can provide a lower probability of error.

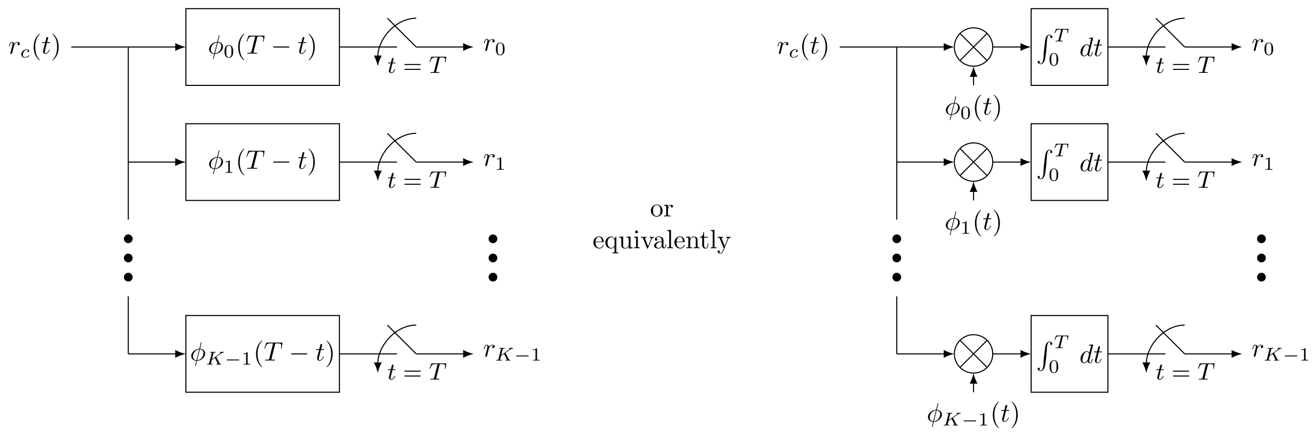
The optimal receiver can be separated into two stages, a detector, which filters and samples the received signal, and a decision device, which uses the samples to make its decision.



Detector – extracts a set of "sufficient statistics" from $r_c(t)$.
Decision Device – attempts to determine the transmitted symbol, m, based on $\underline{r} = [r_0\; r_1\; \cdots\; r_{K-1}]$.

Optimal Detector:

- filters and samples the received signal.
- optimal detector is composed of a bank of K matched filters (or correlators).
- filters are matched to the basis signals, $\{\phi_k(t) \mid k = 0, 1, \ldots, K-1\}$.



For $k \in \{0, 1, \ldots, K-1\}$, the received samples are

$$\begin{aligned} r_k &= \int_0^T r_c(t)\phi_k(t)\,dt \qquad\qquad [\text{but } r_c(t) = s_m(t) + w_c(t)] \\ &= \int_0^T s_m(t)\phi_k(t)\,dt + \int_0^T w_c(t)\phi_k(t)\,dt \\ &= s_{m,k} + w_k\ , \end{aligned}$$

where

$$w_k = \int_0^T w_c(t)\phi_k(t)\,dt$$

represents a sample of noise, and $\{s_{m,k}\}$ are the weights for signal $s_m(t)$. That is,

$$s_{m,k} = \langle s_m(t), \phi_k(t)\rangle = \int_0^T s_m(t)\phi_k(t)\,dt$$

The samples $\underline{r}$ represent the projection of the received signal onto the signal space defined by $\{\phi_k(t) | k = 0, 1, \ldots, K-1\}$.

Properties of w_k:

- Since $w_c(t)$ is a Gaussian random process, w_k is a Gaussian random variable.
- Mean:

$$\mathbf{E}[w_k] = \mathbf{E}\left[\int_0^T w_c(t)\phi_k(t)\,dt\right] = \int_0^T \mathbf{E}[w_c(t)]\,\phi_k(t)\,dt = 0\ .$$

- Covariance:

$$\begin{aligned} \mathbf{E}[w_k w_l] &= \mathbf{E}\left[\int_0^T w_c(t_1)\phi_k(t_1)\,dt_1 \int_0^T w_c(t_2)\phi_l(t_2)\,dt_2\right] \\ &= \int_0^T \int_0^T \mathbf{E}[w_c(t_1)w_c(t_2)]\,\phi_k(t_1)\phi_l(t_2)\,dt_1\,dt_2 \\ &= \int_0^T \int_0^T \frac{\mathcal{N}_0}{2}\delta(t_2 - t_1)\phi_k(t_1)\phi_l(t_2)\,dt_1\,dt_2 \\ &= \frac{\mathcal{N}_0}{2}\int_0^T \phi_k(t_2)\phi_l(t_2)\,dt_2 \\ &= \begin{cases} \mathcal{N}_0/2, & \text{if } k = l \\ 0, & \text{if } k \neq l \end{cases} \\ &= \frac{\mathcal{N}_0}{2}\delta_{l-k}\ . \end{aligned}$$

Properties of r_k:

- Mean:
$$\mathbf{E}[r_k] = \mathbf{E}[s_{m,k} + w_k] \;=\; s_{m,k} \;+\; \mathbf{E}[w_k] \;=\; s_{m,k} \;.$$
- Covariance:
$$\begin{aligned}\mathbf{E}\Big[\Big(r_k - \mathbf{E}[r_k]\Big)\Big(r_l - \mathbf{E}[r_l]\Big)\Big] &= \mathbf{E}[w_k w_l] \\ &= \frac{\mathcal{N}_0}{2}\delta_{l-k} \\ &= \begin{cases} \mathcal{N}_0/2, & \text{if } k = l \\ 0, & \text{if } k \neq l \end{cases} \;.\end{aligned}$$
- Distribution: $\{r_k\}$ are a set of independent Gaussian random variables, with $r_k \sim N(s_{m,k}, \mathcal{N}_0/2)$.

Residue:

In general, we cannot perfectly reconstruct $r_c(t)$ from the samples $\underline{r}$, so by sampling the filter outputs, some information has been lost. This lost information is the residual error,
$$r_e(t) = r_c(t) \;-\; \sum_{k=0}^{K-1} r_k \phi_k(t) \;.$$
However, $r_e(t)$ contains no relevant information to help in determining m.

Proof:
$$\begin{aligned} r_e(t) &= s_m(t) + w_c(t) - \sum_{k=0}^{K-1}[s_{m,k} + w_k]\,\phi_k(t) \\ &= s_m(t) - \sum_{k=0}^{K-1} s_{m,k}\phi_k(t) + w_c(t) - \sum_{k=0}^{K-1} w_k\phi_k(t) \\ &= s_m(t) - s_m(t) + w_c(t) - \sum_{k=0}^{K-1} w_k\phi_k(t) \\ &= w_c(t) - \sum_{k=0}^{K-1} w_k\phi_k(t) \\ &= w_e(t) \;, \end{aligned}$$
where
$$w_e(t) = w_c(t) - \sum_{k=0}^{K-1} w_k\phi_k(t) \;.$$
Since $w_c(t)$ and w_k are not based on m, $w_e(t)$ has the same value regardless of the transmitted signal. Therefore it will be of no direct assistance in determining m. However, $w_e(t)$ may provide some information about w_k, which could be used indirectly to determine m. But
$$\begin{aligned} \mathbf{E}[w_k w_e(t)] &= \mathbf{E}\left[w_k\left(w_c(t) - \sum_{l=0}^{K-1} w_l\phi_l(t)\right)\right] \\ &= \mathbf{E}[w_k w_c(t)] - \mathbf{E}\left[w_k \sum_{l=0}^{K-1} w_l\phi_l(t)\right] \\ &= \mathbf{E}\left[\int_0^T w_c(\tau)\phi_k(\tau)\,d\tau\; w_c(t)\right] - \sum_{l=0}^{K-1}\mathbf{E}[w_k w_l]\,\phi_l(t) \\ &= \int_0^T \mathbf{E}[w_c(\tau)w_c(t)]\,\phi_k(\tau)\,d\tau - \sum_{l=0}^{K-1}\frac{\mathcal{N}_0}{2}\delta_{l-k}\phi_l(t) \\ &= \int_0^T \frac{\mathcal{N}_0}{2}\delta(t-\tau)\phi_k(\tau)\,d\tau - \frac{\mathcal{N}_0}{2}\phi_k(t) \\ &= \frac{\mathcal{N}_0}{2}\phi_k(t) - \frac{\mathcal{N}_0}{2}\phi_k(t) \\ &= 0\;. \end{aligned}$$

– Therefore, $w_e(t)$ and w_k are uncorrelated for all $t \in [0,T]$ and $k \in \{0,1,\ldots,K-1\}$.
– Therefore, $w_e(t)$ is independent of w_k for all $t \in [0,T]$ and $k \in \{0,1,\ldots,K-1\}$.
– Therefore, $w_e(t)$ contains no information about w_k.
– Therefore, knowledge of $w_e(t)$ is of no assistance in determining m.
– The samples $\underline{r}$ are a set of sufficient statistics for determining m. There is no additional information in $r_c(t)$ that is relevant.

Optimal Decision Device

The decision device must make a decision about which symbol was transmitted based on the received observations, $\underline{r}$. An optimal decision device is one that makes this decision in such a manner that the probability of a symbol error is minimized. Let $\widehat{m}$ be the decision made by the device.

Def^n: The *a priori probability distribution* is the probability distribution of the transmitted symbols before any data has been received. It is denoted by $\Pr\{m \text{ sent}\}$. Typically, each symbol is equally likely to have been transmitted, so $\Pr\{m \text{ sent}\} = 1/M$.

Def^n: The *a posteriori probability distribution* (APP) is the probability distribution of the transmitted symbols after the received signal has been observed. It is denoted by $\Pr\{m \text{ sent} \mid \underline{r} \text{ received}\}$.

Def^n: The conditional probability density function $f_{\underline{r}}(\underline{r} \mid m \text{ sent})$ is the pdf of observing $\underline{r}$ at the output of the detector, given that symbol m was transmitted. This is referred to as the *likelihood function*.

Maximum A Posteriori Probability (MAP) Decision Rule

To minimize the probability of an error, the decision device must maximize the probability that its decision is correct. It chooses $\widehat{m} = m$, for the value of m with the largest *a posteriori* probability. That is, choose $\widehat{m} = m$ if

$$\Pr\{m \text{ sent} \mid \underline{r} \text{ received}\} \;\geq\; \Pr\{l \text{ sent} \mid \underline{r} \text{ received}\} \;\text{ for all } l \neq m \;,$$

or equivalently

$$\widehat{m} = \arg\max_m \Pr\{m \text{ sent} \mid \underline{r} \text{ received}\} \;.$$

This is known as the *maximum a posteriori probability* (MAP) decision rule.

Example: Consider a system where one of $M = 4$ possible values could have been transmitted. Suppose, based on the received signal, the receiver calculates the following APP's

$$\begin{aligned}
\Pr\{0 \text{ sent} \mid \underline{r} \text{ received}\} &= 0.2 \\
\Pr\{1 \text{ sent} \mid \underline{r} \text{ received}\} &= 0.1 \\
\Pr\{2 \text{ sent} \mid \underline{r} \text{ received}\} &= 0.4 \\
\Pr\{3 \text{ sent} \mid \underline{r} \text{ received}\} &= 0.3
\end{aligned}$$

According to the MAP decision rule, the receiver would chose $\widehat{m} = 2$, since it is most likely to have been transmitted based on the observations of the received signal.

Note: The probability of error in this case is 0.6, but any other choice for $\widehat{m}$ would lead to a higher probability of error.

The APP's can be calculated from the *likelihood function*, $f_{\underline{r}}(\underline{r} \mid m \text{ sent})$, with

$$\Pr\{m \text{ sent} \mid \underline{r} \text{ received}\} = \frac{f_{\underline{r}}(\underline{r} \mid m \text{ sent})\Pr\{m \text{ sent}\}}{f_{\underline{r}}(\underline{r})} = \frac{f_{\underline{r}}(\underline{r} \mid m \text{ sent}) \Pr\{m \text{ sent}\}}{\sum_{m'=0}^{M-1} f_{\underline{r}}(\underline{r} \mid m' \text{ sent}) \Pr\{m' \text{ sent}\}}$$

For the AWGN channel, since the components of $\underline{r} = [r_0\, r_1\, \cdots\, r_{K-1}]$ are independent, and each r_k has a Gaussian distribution with a mean of $s_{m,k}$ and a variance of $\mathcal{N}_0/2$, the likelihood function is

$$\begin{aligned}
f_{\underline{r}}(\underline{r} \mid m \text{ sent}) &= \prod_{k=0}^{K-1} f_{r_k}(r_k|m \text{ sent}) \\
&= \prod_{k=0}^{K-1} \frac{1}{\sqrt{2\pi(\mathcal{N}_0/2)}} \exp\left\{-\frac{(r_k - s_{m,k})^2}{2(\mathcal{N}_0/2)}\right\} \\
&= \frac{1}{(\sqrt{\pi\mathcal{N}_0})^K} \exp\left\{-\frac{1}{\mathcal{N}_0} \sum_{k=0}^{K-1} (r_k - s_{m,k})^2\right\} \;.
\end{aligned}$$

Maximum Likelihood (ML) Decision Rule

Under certain conditions, the MAP decision rule can be simplified. Usually, all the symbols are equally likely to be transmitted, so the *a priori* probabilities $\Pr\{m \text{ sent}\} = 1/M$, so the MAP decision rule can be expressed as:

$$\widehat{m} = \arg\max_m \frac{f_{\underline{r}}(\underline{r} \mid m \text{ sent}) 1/M}{f_{\underline{r}}(\underline{r})}$$

or

$$\widehat{m} = \arg\max_m f_{\underline{r}}(\underline{r} \mid m \text{ sent}) \ .$$

This is know as the maximum likelihood (ML) decision rule. Note that the ML decision rule is equivalent to the MAP decision rule if the *a priori* probabilities are all equal.

Simplifications to the ML Decision Rule

Using the expression given above for the likelihood function, the ML decision rule can then be expressed as:

$$\widehat{m} = \arg\max_m \frac{1}{(\sqrt{\pi \mathcal{N}_0})^K} \exp\left\{ -\frac{1}{\mathcal{N}_0} \sum_{k=0}^{K-1} (r_k - s_{m,k})^2 \right\}$$

or

$$\widehat{m} = \arg\max_m \exp\left\{ -\frac{1}{\mathcal{N}_0} \sum_{k=0}^{K-1} (r_k - s_{m,k})^2 \right\}$$

or (by taking the log)

$$\widehat{m} = \arg\max_m \left(-\frac{1}{\mathcal{N}_0} \sum_{k=0}^{K-1} (r_k - s_{m,k})^2 \right)$$

or

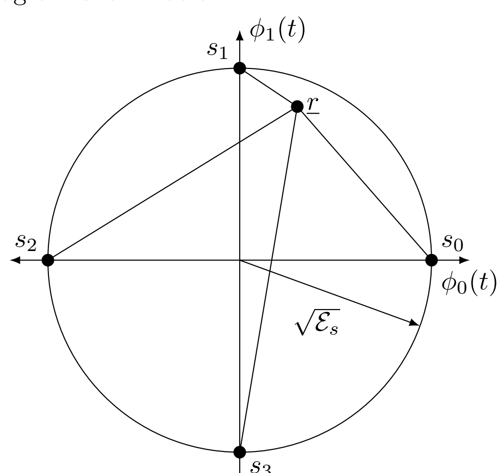
$$\widehat{m} = \arg\min_m \sum_{k=0}^{K-1} (r_k - s_{m,k})^2 \ .$$

But, $\sum_{k=0}^{K-1} (r_k - s_{m,k})^2 = \left\| \underline{r} - \underline{s}_m \right\|^2$, the square of the distance between $\underline{r}$ and the point in the signal space diagram corresponding to $s_m(t)$. Therefore the ML decision rule reduces to:

$$\widehat{m} = \arg\min_m \left\| \underline{r} - \underline{s}_m \right\| \ .$$

In other words, the optimal decision is that symbol that is "closest" to $\underline{r}$ in the signal space.

<u>Example:</u> Suppose that $\underline{r} = [0.3,\ 0.8]\sqrt{\mathcal{E}_s}$ is observed when QPSK is used. The observation is marked in the signal space diagram shown below:

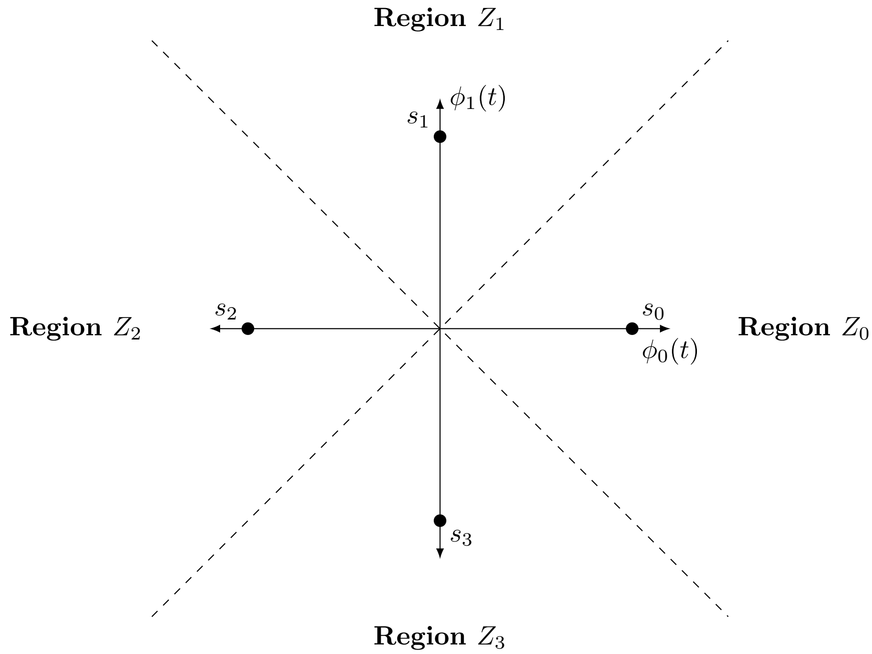


$$\left\| \underline{r} - \underline{s}_0 \right\| = \sqrt{1.13 \mathcal{E}_s}$$
$$\left\| \underline{r} - \underline{s}_1 \right\| = \sqrt{0.13 \mathcal{E}_s}$$
$$\left\| \underline{r} - \underline{s}_2 \right\| = \sqrt{2.33 \mathcal{E}_s}$$
$$\left\| \underline{r} - \underline{s}_3 \right\| = \sqrt{3.33 \mathcal{E}_s}$$

The decoder would choose $\widehat{m} = 1$.

Decision Regions:

Each possible received observation will be closest to one of the points in the signal constellation. For each signalling scheme, it is useful to draw the boundaries of the decision regions on the signal space diagram



Further Simplifications to the ML Decision Rule

The ML decision rule can also be expressed as:

$$\widehat{m} = \arg\min_m \sum_{k=0}^{K-1} (r_k^2 - 2r_k s_{m,k} + s_{m,k}^2)$$

or

$$\widehat{m} = \arg\min_m \left(\sum_{k=0}^{K-1} r_k^2 - 2 \sum_{k=0}^{K-1} r_k s_{m,k} + \sum_{k=0}^{K-1} s_{m,k}^2 \right)$$

or

$$\widehat{m} = \arg\min_m \left(-2 \sum_{k=0}^{K-1} r_k s_{m,k} + E_m \right)$$

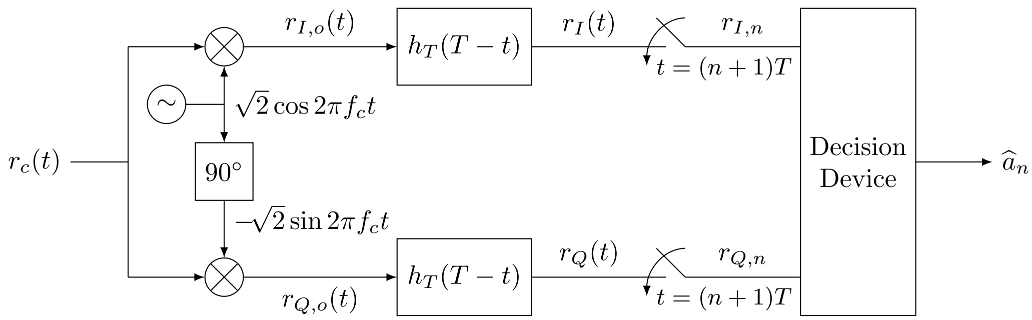
or

$$\widehat{m} = \arg\max_m \left(\sum_{k=0}^{K-1} r_k s_{m,k} - E_m/2 \right) .$$

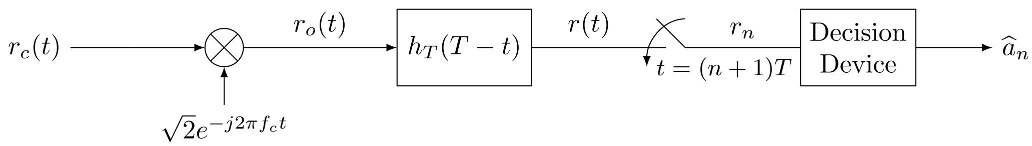
If all signals have equal energy (i.e., $E_m = E_l \; \forall \; m, l$), then the ML decision rule can be expressed as:

$$\widehat{m} = \arg\max_m \sum_{k=0}^{K-1} r_k s_{m,k} .$$

Bandpass Receiver Structures



or equivalently



Demodulator:

- converts received signal to baseband

$$\begin{aligned} r_o(t) &= r_{I,o}(t) + j r_{Q,o}(t) \\ &= r_c(t)\sqrt{2}\cos 2\pi f_c t - j r_c(t)\sqrt{2}\sin 2\pi f_c t \\ &= r_c(t)\sqrt{2}e^{-j2\pi f_c t} \end{aligned}$$

- received bandpass signal is

$$r_c(t) = v_c(t) + w_c(t)$$

so

$$\begin{aligned} r_o(t) &= v_c(t)\sqrt{2}e^{-j2\pi f_c t} + w_c(t)\sqrt{2}e^{-j2\pi f_c t} \\ &= \mathbf{Re}\left\{v(t)\sqrt{2}e^{j2\pi f_c t}\right\}\sqrt{2}e^{-j2\pi f_c t} + w_o(t) \qquad \left(\text{since } v_c(t) = \mathbf{Re}\left\{v(t)\sqrt{2}e^{j2\pi f_c t}\right\}\right) \\ &= \tfrac{1}{2}\left[v(t)\sqrt{2}e^{j2\pi f_c t} + v^*(t)\sqrt{2}e^{-j2\pi f_c t}\right]\sqrt{2}e^{-j2\pi f_c t} + w_o(t) \\ &= v(t) + v^*(t)e^{-j4\pi f_c t} + w_o(t) \end{aligned}$$

where $w_o(t)$ is the demodulated noise.

- the high frequency component will be removed by the receive filter.

Receive Filter:

- matched to the transmitted pulse shape

$$h_R(t) = h_T(T-t)$$

- maximizes signal-to-noise ratio
- filtered received signal is

$$\begin{aligned} r(t) &= r_I(t) + j r_Q(t) \\ &= \int_{-\infty}^{\infty} r_o(t-\tau) h_R(\tau)\, d\tau \\ &= \int_{-\infty}^{\infty} \left[v(t-\tau) + w_o(t-\tau)\right] h_T(T-\tau)\, d\tau \\ &= \int_{-\infty}^{\infty} v(t-\tau) h_T(T-\tau)\, d\tau + \int_{-\infty}^{\infty} w_o(t-\tau) h_T(T-\tau)\, d\tau \\ &= \int_{-\infty}^{\infty}\left[\sum_{n=0}^{N_a-1} v_n h_T(t-\tau-nT)\right] h_T(T-\tau)\, d\tau + w(t) \\ &= \sum_{n=0}^{N_a-1} v_n \int_{-\infty}^{\infty} h_T(t-\tau-nT) h_T(T-\tau)\, d\tau + w(t) \end{aligned}$$

$$= \sum_{n=0}^{N_a-1} v_n \int_{-\infty}^{\infty} h_T(t+\tau-[n+1]T)h_T(\tau)\,d\tau + w(t)$$
$$= \sum_{n=0}^{N_a-1} v_n h_{TR}(t-[n+1]T) + w(t)$$

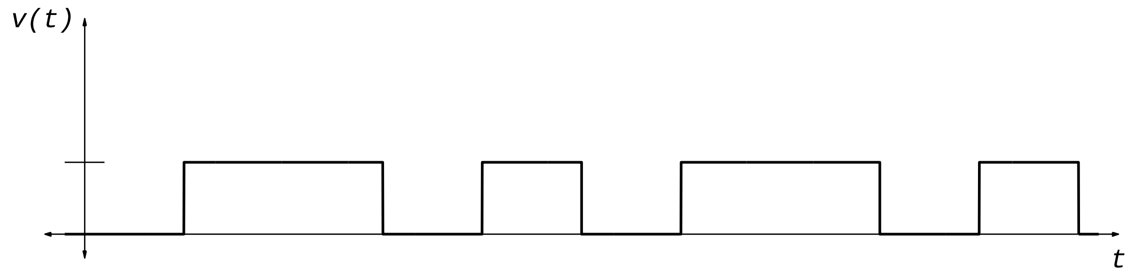
where $w(t)$ is the filtered noise, and

$$h_{TR}(t) = \int_{-\infty}^{\infty} h_T(t+\tau)h_T(\tau)\,d\tau$$

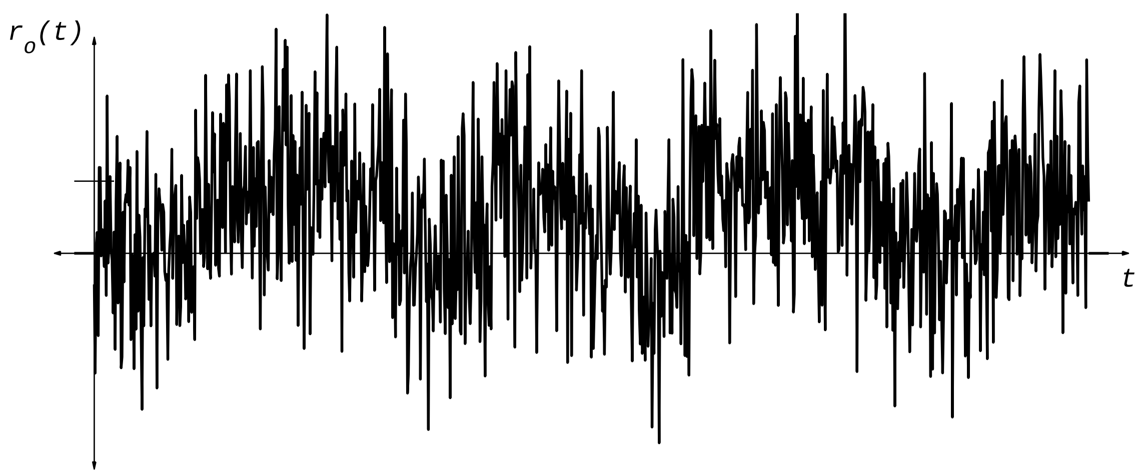
is the combined impulse response of the transmit and receive filters.

Eye Diagrams:

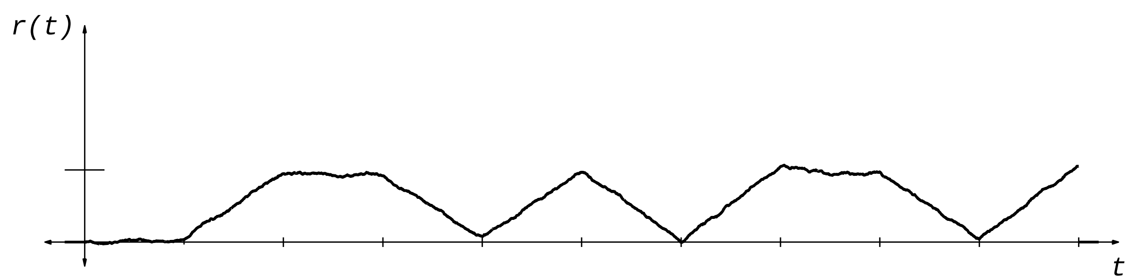
Suppose a rectangular pulse shape is used for $h_T(t)$. For a transmitted lowpass signal, $v(t)$, given by



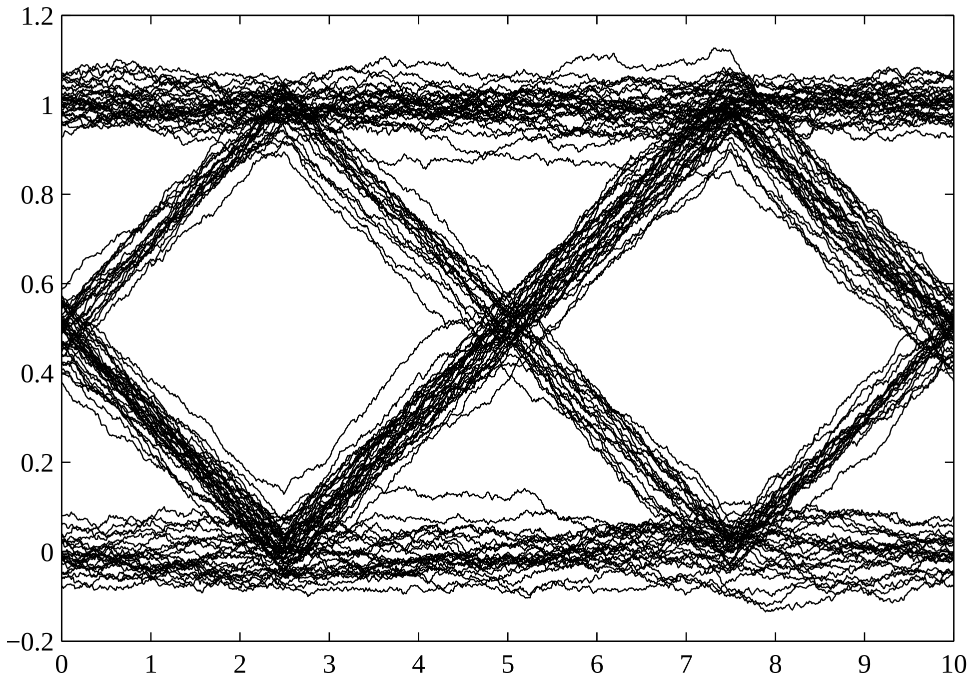
and a received demodulated signal, $r_o(t)$, of



the filter output would be



By overlaying segments of duration $2T$ from the above figure, an *eye diagram* can be created



Signal Sampler:

– sample signal at the symbol rate

$$\begin{aligned} r_n &= r_{I,n} + j r_{Q,n} \\ &= r([n+1]T) \\ &= \sum_{m=0}^{N_a-1} v_m h_{TR}([n+1]T - [m+1]T) + w([n+1]T) \\ &= \sum_{m=0}^{N_a-1} v_m h_{TR}([n-m]T) + w_n \end{aligned}$$

where w_n is a noise sample.

– to prevent intersymbol interference (ISI), it is necessary that

$$h_{TR}(nT) = \delta_n$$

so that

$$\begin{aligned} r_n &= \sum_{m=0}^{N_a-1} v_m \delta_{n-m} + w_n \\ &= v_n + w_n \end{aligned}$$

All unit-energy pulse shapes that are non-zero over only the interval $[0, T]$ possess this property.

Note: There are many other pulse shapes of longer duration that also fulfill this requirement. Although the transmitted symbols will overlap in time as the signal is transmitted, as long as the pulse shape has the property

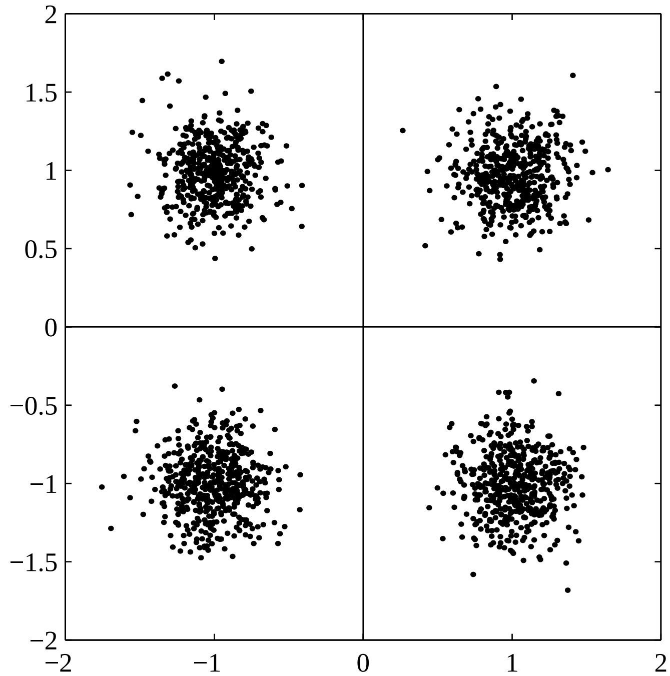
$$\int_{-\infty}^{\infty} h_T(nT + \tau) h_T(\tau)\, d\tau = \delta_n$$

then ISI will not occur.

Longer durations for the pulse shape can lead to narrower bandwidth signals.

Signal Space Diagram:

The received samples can be plotted on the signal space diagram:



Decision Device:

– select the point in the signal constellation closest to the received sample to estimate the transmitted symbol

Probability of a Symbol Error

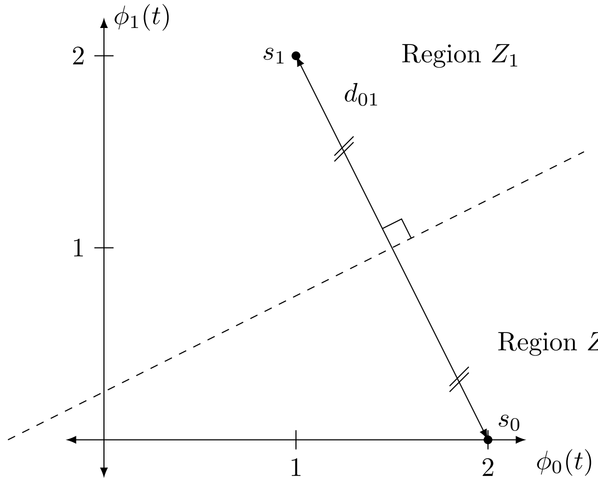
Binary Signalling

Consider the case of binary signalling with signals $\underline{s}_0 = (s_{0,0}, s_{0,1})$ and $\underline{s}_1 = (s_{1,0}, s_{1,1})$. where $\underline{s}_0$ and $\underline{s}_1$ are equally likely to be transmitted (that is, they have equal *a priori* probabilities, $\Pr\{\text{"0" sent}\} = \Pr\{\text{"1" sent}\} = \frac{1}{2}$).
The optimal maximum likelihood decision rule is:

Choose $\widehat{m} = 0$ if

$$\left\|\underline{r} - \underline{s}_0\right\|^2 < \left\|\underline{r} - \underline{s}_1\right\|^2$$

Example:



$$\begin{aligned} \underline{s}_0 &= (s_{0,0}, s_{0,1}) = (2,0) \\ \underline{s}_1 &= (s_{1,0}, s_{1,1}) = (1,2) \\ \\ \mathcal{E}_0 &= 4 \\ \mathcal{E}_1 &= 5 \\ d_{01} &= \sqrt{5} \end{aligned}$$

Suppose that a "0" is sent. An error occurs if $\underline{r}$ is closer to $\underline{s}_1$ than to $\underline{s}_0$. The probability of error is

$$P_{\varepsilon|0} = \Pr\left\{\left\|\underline{r} - \underline{s}_0\right\|^2 > \left\|\underline{r} - \underline{s}_1\right\|^2 \;\middle|\; \text{"0" sent}\right\}$$

If "0" is sent then $\underline{r} = \underline{s}_0 + \underline{w}$, where $\underline{w} = (w_0, w_1)$. w_0 and w_1 are independent Gaussian random variables, each with zero mean and variance of $\mathcal{N}_0/2$. Therefore

$$\begin{aligned} P_{\varepsilon|0} &= \Pr\left\{\left\|\underline{s}_0 + \underline{w} - \underline{s}_0\right\|^2 > \left\|\underline{s}_0 + \underline{w} - \underline{s}_1\right\|^2\right\} \\ &= \Pr\left\{\left\|\underline{w}\right\|^2 > \left\|\underline{w} + \underline{s}_0 - \underline{s}_1)\right\|^2\right\} \\ &= \Pr\left\{w_0^2 + w_1^2 > (w_0 + s_{0,0} - s_{1,0})^2 + (w_1 + s_{0,1} - s_{1,1})^2\right\} \\ &= \Pr\left\{w_0^2 + w_1^2 > w_0^2 + 2w_0(s_{0,0} - s_{1,0}) + (s_{0,0} - s_{1,0})^2 + w_1^2 + 2w_1(s_{0,1} - s_{1,1}) + (s_{0,1} - s_{1,1})^2\right\} \\ &= \Pr\left\{-2w_0(s_{0,0} - s_{1,0}) - 2w_1(s_{0,1} - s_{1,1}) > (s_{0,0} - s_{1,0})^2 + (s_{0,1} - s_{1,1})^2\right\} \\ &= \Pr\left\{w_0(s_{0,0} - s_{1,0}) + w_1(s_{0,1} - s_{1,1}) < -\tfrac{1}{2}\left\|\underline{s}_0 - \underline{s}_1\right\|^2\right\} \\ &= \Pr\left\{w_0(s_{0,0} - s_{1,0}) + w_1(s_{0,1} - s_{1,1}) < -\tfrac{1}{2}d_{01}^2\right\} \end{aligned}$$

where d_{01} is the distance between $\underline{s}_0$ and $\underline{s}_1$ in the signal space diagram.
Define

$$X = w_0(s_{0,0} - s_{1,0}) + w_1(s_{0,1} - s_{1,1})$$

Since w_0 and w_1 are Gaussian, X is also Gaussian, with a mean of

$$\begin{aligned} \mu_X &= \mathbf{E}[X] \\ &= \mathbf{E}[w_0](s_{0,0} - s_{1,0}) + \mathbf{E}[w_1](s_{0,1} - s_{1,1}) \\ &= (0)(s_{0,0} - s_{1,0}) + (0)(s_{0,1} - s_{1,1}) \\ &= 0 \end{aligned}$$

and a variance of

$$\begin{aligned} \sigma_X^2 &= \mathbf{E}\left[X^2\right] - \mu_X^2 \\ &= \mathbf{E}\left[\left[w_0(s_{0,0} - s_{1,0}) + w_1(s_{0,1} - s_{1,1})\right]^2\right] - (0)^2 \\ &= \mathbf{E}\left[w_0^2\right](s_{0,0} - s_{1,0})^2 + 2\mathbf{E}[w_0 w_1](s_{0,0} - s_{1,0})(s_{0,1} - s_{1,1}) + \mathbf{E}\left[w_1^2\right](s_{0,1} - s_{1,1})^2 \end{aligned}$$

$$= \frac{\mathcal{N}_0}{2}(s_{0,0} - s_{1,0})^2 + \frac{\mathcal{N}_0}{2}(s_{0,1} - s_{1,1})^2 = \frac{\mathcal{N}_0}{2} d_{01}^2$$

The probability of error, given that "0" was sent, is

$$\begin{aligned}
P_{\varepsilon|0} &= \Pr\left\{X < -\tfrac{1}{2}d_{01}^2\right\} \\
&= \int_{-\infty}^{-\frac{1}{2}d_{01}^2} f_X(x)\,dx \\
&= \int_{-\infty}^{-\frac{1}{2}d_{01}^2} \frac{1}{\sqrt{2\pi\sigma_X^2}} \exp\left\{-\frac{1}{2\sigma_X^2}(x-\mu_X)^2\right\} dx \\
&= \int_{-\infty}^{-\frac{1}{2}d_{01}^2} \frac{1}{\sqrt{\pi\mathcal{N}_0 d_{01}^2}} \exp\left\{-\frac{1}{\mathcal{N}_0 d_{01}^2}x^2\right\} dx \qquad \left(\text{let } u = \frac{-x}{\sqrt{\mathcal{N}_0 d_{01}^2}}\right) \\
&= \int_{\frac{d_{01}^2}{2\sqrt{\mathcal{N}_0 d_{01}^2}}}^{\infty} \frac{1}{\sqrt{\pi}} e^{-u^2}\,du \\
&= \tfrac{1}{2}\text{erfc}\left(\frac{d_{01}^2}{2\sqrt{\mathcal{N}_0 d_{01}^2}}\right) \\
&= \tfrac{1}{2}\text{erfc}\left(\frac{d_{01}}{2\sqrt{\mathcal{N}_0}}\right)
\end{aligned}$$

Now assume that a "1" was sent. The probability of error is

$$P_{\varepsilon|1} = \Pr\left\{\left\|\underline{r} - \underline{s}_0\right\|^2 < \left\|\underline{r} - \underline{s}_1\right\|^2 \mid \text{"1" sent}\right\}$$

By symmetry,

$$\begin{aligned}
P_{\varepsilon|1} &= P_{\varepsilon|0} \\
&= \tfrac{1}{2}\text{erfc}\left(\frac{d_{01}}{2\sqrt{\mathcal{N}_0}}\right)
\end{aligned}$$

The average probability of error is

$$\begin{aligned}
P_\varepsilon &= \Pr\{\text{"0" sent}\}\,P_{\varepsilon|0} + \Pr\{\text{"1" sent}\}\,P_{\varepsilon|1} \\
&= \tfrac{1}{2}P_{\varepsilon|0} + \tfrac{1}{2}P_{\varepsilon|1} \\
&= \tfrac{1}{2}\text{erfc}\left(\frac{d_{01}}{2\sqrt{\mathcal{N}_0}}\right)
\end{aligned}$$

which depends only on the distance between the two signals in the signal space diagram, and not on their actual positions.

M-ary Signalling

If symbol m is transmitted, a correct decision is made only if the sample, $\underline{r}$, falls within the decision region for symbol m in the signal space diagram. The probability of making a correct decision is therefore the probability that $\underline{r}$ falls within the decision region, which is denoted by Z_m.

$$P_{C|m} = \int_{Z_m} f_{\underline{r}}(\underline{r} \mid m \text{ sent})\, d\underline{r} \,,$$

where $f_{\underline{r}}(\underline{r} \mid m \text{ sent})$ is the joint conditional pdf of $\underline{r}$ given that m was sent (the likelihood function). For the AWGN channel

$$f_{\underline{r}}(\underline{r} \mid m \text{ sent}) = \frac{1}{\left(\sqrt{\pi \mathcal{N}_0}\right)^K} \exp\left\{ -\frac{1}{\mathcal{N}_0} \sum_{k=0}^{K-1} \left(r_k - s_{m,k}\right)^2 \right\}$$

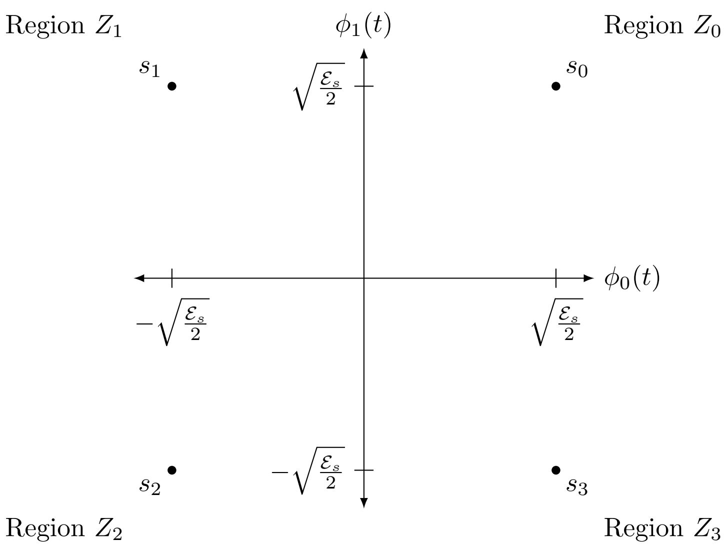
where $\underline{s}_m = [s_{m,0}\ s_{m,1}\ \ldots\ s_{m,K-1}]$ is the point for symbol m in the signal space diagram. The average probability of a correct decision is:

$$\begin{aligned} P_C &= \sum_{m=0}^{M-1} P_{C|m} \Pr\{m \text{ sent}\} \\ &= \frac{1}{M} \sum_{m=0}^{M-1} P_{C|m} \end{aligned}$$

for symbols with equal *a priori* probabilities.
The probability of a symbol error is:

$$P_\varepsilon = 1 - P_C = 1 - \frac{1}{M} \sum_{m=0}^{M-1} P_{C|m}$$

Example: 4-QAM



$$\begin{aligned} \underline{s}_0 &= \left(\sqrt{\frac{\mathcal{E}_s}{2}}, \sqrt{\frac{\mathcal{E}_s}{2}}\right) \\ \underline{s}_1 &= \left(-\sqrt{\frac{\mathcal{E}_s}{2}}, \sqrt{\frac{\mathcal{E}_s}{2}}\right) \\ \underline{s}_2 &= \left(-\sqrt{\frac{\mathcal{E}_s}{2}}, -\sqrt{\frac{\mathcal{E}_s}{2}}\right) \\ \underline{s}_3 &= \left(\sqrt{\frac{\mathcal{E}_s}{2}}, -\sqrt{\frac{\mathcal{E}_s}{2}}\right) \end{aligned}$$

$\mathcal{E}_s$ = Energy per symbol

$$\begin{aligned} P_{C|0} &= \int_{Z_0} f_{\underline{r}}(\underline{r} \mid \text{“0” sent})\, d\underline{r} \\ &= \int_0^\infty \int_0^\infty f_{r_0,r_1}(r_0, r_1 \mid \text{“0” sent})\, dr_0\, dr_1 \\ &= \int_0^\infty \int_0^\infty \frac{1}{\pi \mathcal{N}_0} \exp\left\{ -\frac{1}{\mathcal{N}_0} \left[\left(r_0 - \sqrt{\frac{\mathcal{E}_s}{2}}\right)^2 + \left(r_1 - \sqrt{\frac{\mathcal{E}_s}{2}}\right)^2 \right] \right\} dr_0\, dr_1 \\ &= \int_0^\infty \frac{1}{\sqrt{\pi \mathcal{N}_0}} \exp\left\{ -\frac{1}{\mathcal{N}_0} \left(r_0 - \sqrt{\frac{\mathcal{E}_s}{2}}\right)^2 \right\} dr_0 \int_0^\infty \frac{1}{\sqrt{\pi \mathcal{N}_0}} \exp\left\{ -\frac{1}{\mathcal{N}_0} \left(r_1 - \sqrt{\frac{\mathcal{E}_s}{2}}\right)^2 \right\} dr_1 \\ &= \left[\frac{1}{\sqrt{\pi}} \int_{-\infty}^{\sqrt{\frac{\mathcal{E}_s}{2\mathcal{N}_0}}} e^{-u_0^2}\, du_0 \right] \left[\frac{1}{\sqrt{\pi}} \int_{-\infty}^{\sqrt{\frac{\mathcal{E}_s}{2\mathcal{N}_0}}} e^{-u_1^2}\, du_1 \right] \qquad \left(\text{with } \begin{aligned} u_0 &= -\frac{r_0 - \sqrt{\frac{\mathcal{E}_s}{2}}}{\sqrt{\mathcal{N}_0}} \\ u_1 &= -\frac{r_1 - \sqrt{\frac{\mathcal{E}_s}{2}}}{\sqrt{\mathcal{N}_0}} \end{aligned} \right) \end{aligned}$$

$$= \left[1 - \frac{1}{\sqrt{\pi}} \int_{\sqrt{\frac{\mathcal{E}_s}{2\mathcal{N}_0}}}^{\infty} e^{-u^2}\, du\right]^2$$

$$= \left[1 - \tfrac{1}{2}\text{erfc}\left(\sqrt{\frac{\mathcal{E}_s}{2\mathcal{N}_0}}\right)\right]^2$$

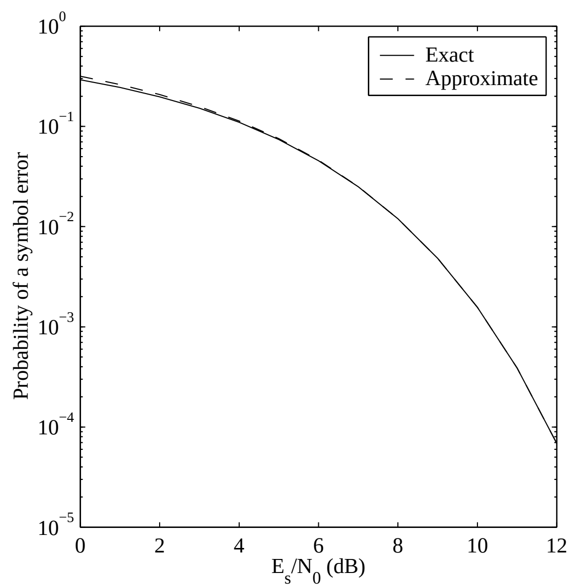
Because of the symmetric nature of the points in the signal space, $P_{C|0}$, $P_{C|1}$, $P_{C|2}$, and $P_{C|3}$ are all equal. Therefore

$$P_{C|m} = \left[1 - \tfrac{1}{2}\text{erfc}\left(\sqrt{\frac{\mathcal{E}_s}{2\mathcal{N}_0}}\right)\right]^2$$

for all $m \in \{0, 1, 2, 3\}$.

The probability of a symbol error is

$$\begin{aligned} P_\varepsilon &= 1 - \frac{1}{4}\sum_{m=0}^{3} P_{C|m} \\ &= 1 - \left[1 - \tfrac{1}{2}\text{erfc}\left(\sqrt{\frac{\mathcal{E}_s}{2\mathcal{N}_0}}\right)\right]^2 \\ &= \text{erfc}\left(\sqrt{\frac{\mathcal{E}_s}{2\mathcal{N}_0}}\right) - \frac{1}{4}\text{erfc}^2\left(\sqrt{\frac{\mathcal{E}_s}{2\mathcal{N}_0}}\right) \\ &\cong \text{erfc}\left(\sqrt{\frac{\mathcal{E}_s}{2\mathcal{N}_0}}\right) \end{aligned}$$



Note: In general, the integral to find $P_{C|m}$ is not always easy to evaluate.

Union Bound on the Error Probability

Def^n: Union Bound
The probability of a finite union of events is upper-bounded by the sum of the probabilities of the constituent events.

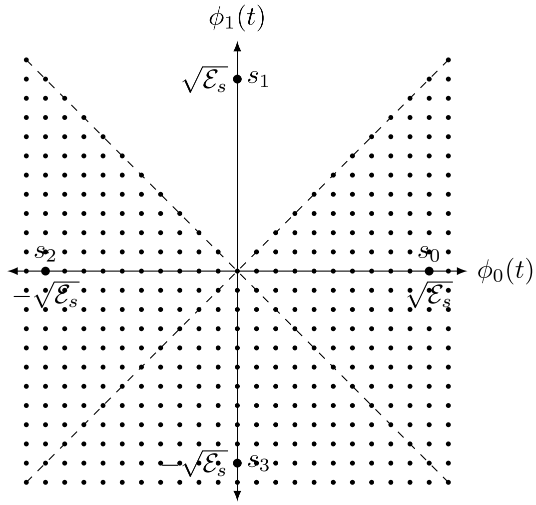
Let

$A_{m,l} =$ the event that $\underline{r}$ is closer to $\underline{s}_l$ than $\underline{s}_m$ when symbol m is sent.

The probability of a symbol error when m is sent is

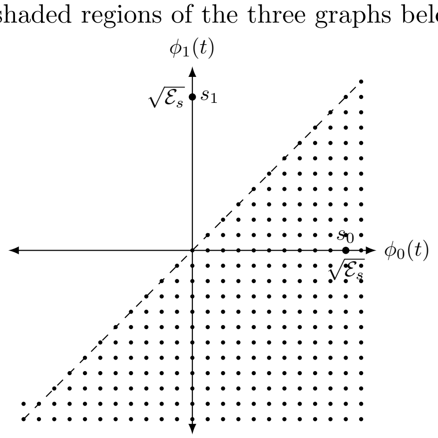
$$\begin{aligned} P_{\varepsilon|m} &= \Pr\{\underline{r} \text{ is closer to } \underline{s}_l \text{ than } \underline{s}_m \text{ for some } l \neq m \mid m \text{ sent}\} \\ &= \Pr\left\{ \bigcup_{\substack{l=0 \\ l\neq m}}^{M-1} A_{m,l} \right\} \\ &\leq \sum_{\substack{l=0 \\ l\neq m}}^{M-1} \Pr\{A_{m,l}\} \end{aligned}$$

Example:

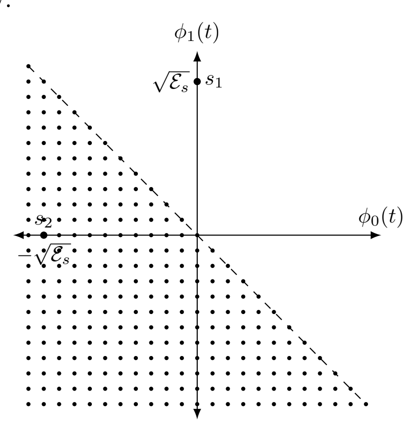


If symbol "1" is sent, an error occurs if $\underline{r}$ falls in the shaded region.

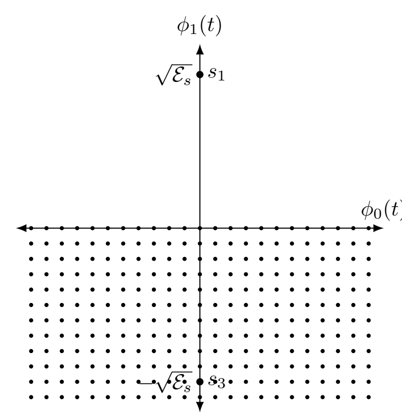
The probability of an error if "1" is sent is less than or equal to the sum of the probabilities that $\underline{r}$ falls in the shaded regions of the three graphs below:



Event $A_{1,0}$ occurs if $\underline{r}$ falls in this region.



Event $A_{1,2}$ occurs if $\underline{r}$ falls in this region.



Event $A_{1,3}$ occurs if $\underline{r}$ falls in this region.

From the discussion on error probability for binary signalling,

$$\Pr\{A_{m,l}\} = \tfrac{1}{2}\text{erfc}\left(\frac{d_{m,l}}{2\sqrt{\mathcal{N}_0}}\right)$$

where $d_{m,l}$ is the distance between points $\underline{s}_m$ and $\underline{s}_l$ in the signal space diagram.
Using the union bound, the probability of a symbol error is upper-bounded by

$$P_{\varepsilon|m} \le \sum_{\substack{l=0\\ l\neq m}}^{M-1} \tfrac{1}{2}\text{erfc}\left(\frac{d_{m,l}}{2\sqrt{\mathcal{N}_0}}\right)$$

so

$$P_{\varepsilon} \le \frac{1}{M}\sum_{m=0}^{M-1}\sum_{\substack{l=0\\ l\neq m}}^{M-1} \tfrac{1}{2}\text{erfc}\left(\frac{d_{m,l}}{2\sqrt{\mathcal{N}_0}}\right)$$

Example: For the QPSK example:

$$d_{1,0} = \sqrt{2\mathcal{E}_s} \qquad d_{1,2} = \sqrt{2\mathcal{E}_s} \qquad d_{1,3} = 2\sqrt{\mathcal{E}_s}$$

$$\begin{aligned} P_{\varepsilon|1} &\le \Pr\{A_{1,0}\} + \Pr\{A_{1,2}\} + \Pr\{A_{1,3}\} \\ &= \tfrac{1}{2}\text{erfc}\left(\frac{\sqrt{2\mathcal{E}_s}}{2\sqrt{\mathcal{N}_0}}\right) + \tfrac{1}{2}\text{erfc}\left(\frac{\sqrt{2\mathcal{E}_s}}{2\sqrt{\mathcal{N}_0}}\right) + \tfrac{1}{2}\text{erfc}\left(\frac{2\sqrt{\mathcal{E}_s}}{2\sqrt{\mathcal{N}_0}}\right) \\ &= \text{erfc}\left(\sqrt{\frac{\mathcal{E}_s}{2\mathcal{N}_0}}\right) + \frac{1}{2}\text{erfc}\left(\sqrt{\frac{\mathcal{E}_s}{\mathcal{N}_0}}\right) \end{aligned}$$

By symmetry

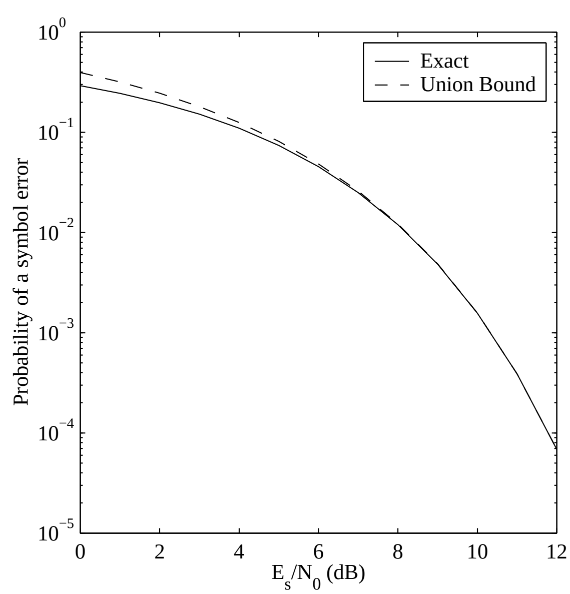
$$P_{\varepsilon|m} = P_{\varepsilon|1} \le \text{erfc}\left(\sqrt{\frac{\mathcal{E}_s}{2\mathcal{N}_0}}\right) + \frac{1}{2}\text{erfc}\left(\sqrt{\frac{\mathcal{E}_s}{\mathcal{N}_0}}\right)$$

Therefore

$$P_{\varepsilon} = \frac{1}{4}\sum_{m=0}^{3} P_{\varepsilon|m} \le \text{erfc}\left(\sqrt{\frac{\mathcal{E}_s}{2\mathcal{N}_0}}\right) + \frac{1}{2}\text{erfc}\left(\sqrt{\frac{\mathcal{E}_s}{\mathcal{N}_0}}\right)$$

Note: Since QPSK differs from 4-QAM only by a rotation of the bases, the error probability for QPSK is the same as for 4-QAM. That is,

$$P_{\varepsilon} = \text{erfc}\left(\sqrt{\frac{\mathcal{E}_s}{2\mathcal{N}_0}}\right) - \frac{1}{4}\text{erfc}^2\left(\sqrt{\frac{\mathcal{E}_s}{2\mathcal{N}_0}}\right)$$



Probability of a Bit Error

Usually, the probability of a bit error is of greater interest than the probability of a symbol error. The probability of a bit error given that symbol m was transmitted is

$$P_{b|m} = \sum_{l=0}^{M-1} \Pr\{\widehat{m} = l \mid m \text{ sent}\} \times \left(\frac{\#\text{ of bit positions in which } m \text{ and } l \text{ differ}}{\#\text{ of bits per symbol}} \right)$$

where

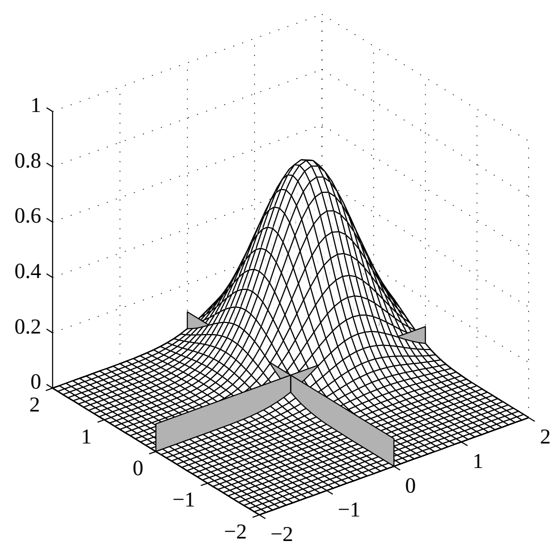
$$\Pr\{\widehat{m} = l \mid \text{m sent}\} = \int_{Z_l} f_{\underline{r}}(\underline{r} \mid m \text{ sent})\, d\underline{r}$$

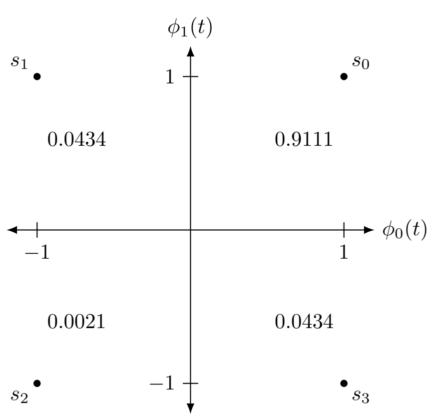
is the probability that the decision device decides in favour of symbol l when m was transmitted.
The average probability of a bit error is

$$P_b = \sum_{m=0}^{M-1} P_{b|m} \Pr\{m \text{ sent}\}$$

Example: 4-QAM
Likelihood function:

$$f_{\underline{r}}(\underline{r} \mid 0 \text{ sent}) = \frac{1}{\pi \mathcal{N}_0} \exp\left\{ -\frac{1}{\mathcal{N}_0} \left[(r_0 - 1)^2 + (r_1 - 1)^2 \right] \right\}$$





$$P_{0|0} = \Pr\{\widehat{m} = 0 \mid \text{“0” sent}\} = 0.9111$$
$$P_{1|0} = \Pr\{\widehat{m} = 1 \mid \text{“0” sent}\} = 0.0434$$
$$P_{2|0} = \Pr\{\widehat{m} = 2 \mid \text{“0” sent}\} = 0.0021$$
$$P_{3|0} = \Pr\{\widehat{m} = 3 \mid \text{“0” sent}\} = 0.0434$$

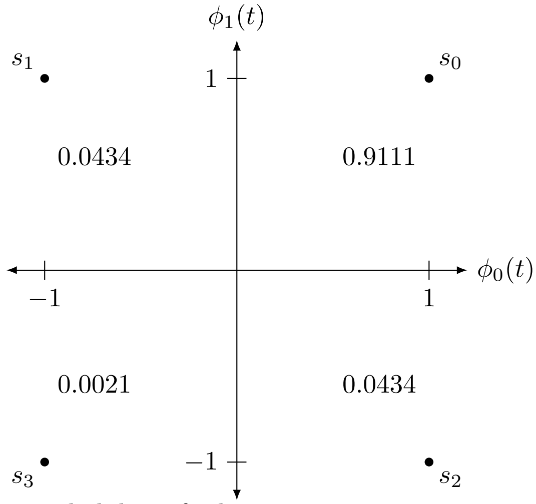
If a “0” (00) is sent, one bit error occurs if a “1” (01) is received, one bit error occurs if a “2” (10) is received, and two bit errors occur if a “3” (11) is received. The probability of a bit error is:

$$\begin{aligned} P_{b|0} &= \tfrac{0}{2} P_{0|0} + \tfrac{1}{2} P_{1|0} + \tfrac{1}{2} P_{2|0} + \tfrac{2}{2} P_{3|0}] \\ &= 0.06615 \end{aligned}$$

Note: The probability of a bit error depends on the order in which symbols are mapped to signals.

Example: 4-QAM

Now, suppose the symbols are mapped to signals as follows:



$$P_{0|0} = \Pr\{\widehat{m} = 0 \mid \text{“0” sent}\} = 0.9111$$
$$P_{1|0} = \Pr\{\widehat{m} = 1 \mid \text{“0” sent}\} = 0.0433$$
$$P_{2|0} = \Pr\{\widehat{m} = 2 \mid \text{“0” sent}\} = 0.0433$$
$$P_{3|0} = \Pr\{\widehat{m} = 3 \mid \text{“0” sent}\} = 0.0021$$

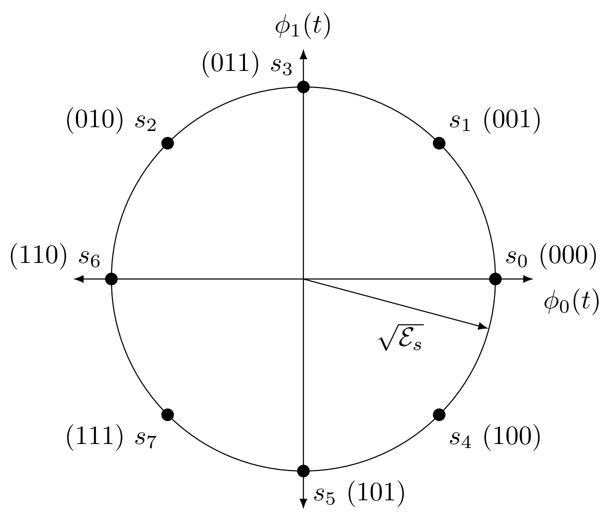
The probability of a bit error is:

$$\begin{aligned} P_{b|0} &= \tfrac{0}{2}P_{0|0} + \tfrac{1}{2}P_{1|0} + \tfrac{1}{2}P_{2|0} + \tfrac{2}{2}P_{3|0}] \\ &= 0.0455 \end{aligned}$$

Gray Mapping:

To reduce the probability of a bit error, care must be taken when assigning symbols to points in the signal constellation. When symbol errors occur, the received sample usually falls within one of the adjacent decision regions, and not in one that is far-removed from the transmitted signal. Symbols that occupy adjacent regions in the signal space diagram should differ by only one bit, if possible.

Example: 8-PSK

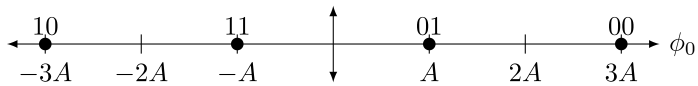


When Gray Mapping is used, usually there is only a single bit error if a symbol error occurs. The probability of a bit error is therefore

$$P_b \cong \frac{P_\varepsilon}{\log_2 M}$$

Example: Probability of a Bit Error of 4-PAM

Signal Space Diagram:

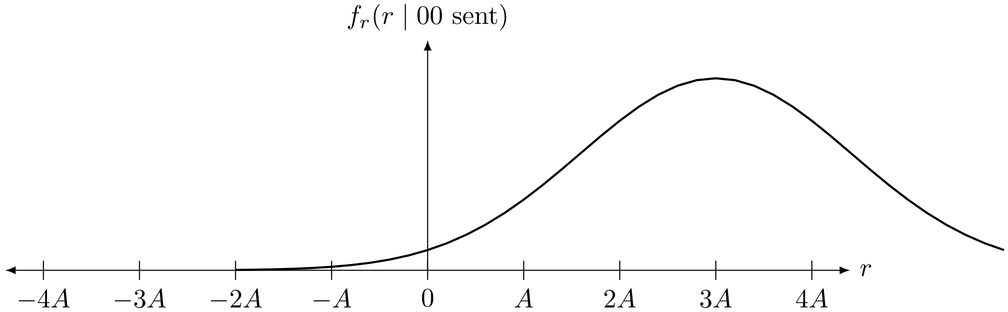


First, calculate the probability of receiving 00, 01, 10, and 11 if a 00 was transmitted. If 00 was transmitted then

$$r = 3A + w$$

where w is the AWGN, with a Gaussian distribution with zero mean and a variance of $\mathcal{N}_0/2$. The likelihood function is

$$f_r(r \mid 00 \text{ sent}) = \frac{1}{\sqrt{\pi \mathcal{N}_0}} \exp\left\{-\frac{1}{\mathcal{N}_0}(r-3A)^2\right\}$$

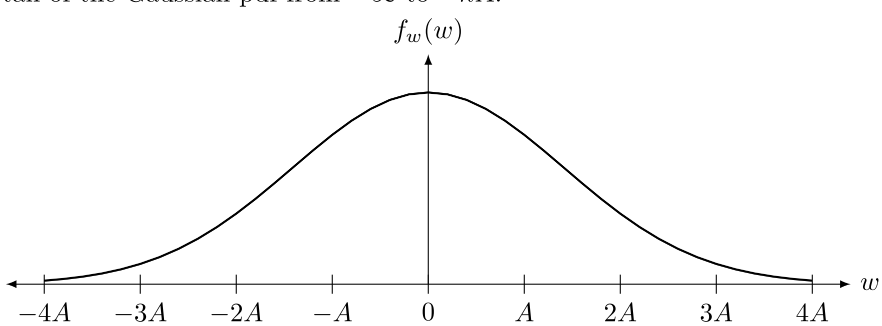


$$\begin{aligned}
P_{00|00} &= \Pr\{r > 2A \mid 00 \text{ sent}\} \\
&= \Pr\{3A + w > 2A\} \\
&= \Pr\{w > -A\} \\
&= 1 - \Pr\{w < -A\} \\
&= 1 - \Pr\{w > A\} \\
&= 1 - \tfrac{1}{2}\text{erfc}\left(\frac{A}{\sqrt{\mathcal{N}_0}}\right) \\
&= 1 - Q_1
\end{aligned}$$

where, to simplify notation, Q_n is defined as

$$Q_n \triangleq \tfrac{1}{2}\text{erfc}\left(\frac{nA}{\sqrt{\mathcal{N}_0}}\right) = \Pr\{w < -nA\} = \Pr\{w > nA\}$$

is the area under the tail of the Gaussian pdf from $-\infty$ to $-nA$.



$$P_{01|00} = \Pr\{0 < r \le 2A \mid 00 \text{ sent}\} = \Pr\{-3A < w \le -A\} = Q_1 - Q_3$$
$$P_{11|00} = \Pr\{-2A < r \le 0 \mid 00 \text{ sent}\} = \Pr\{-5A < w \le -3A\} = Q_3 - Q_5$$
$$P_{10|00} = \Pr\{r \le -2A \mid 00 \text{ sent}\} = \Pr\{w \le -5A\} = Q_5$$

The probability of a bit error given that 00 was sent is

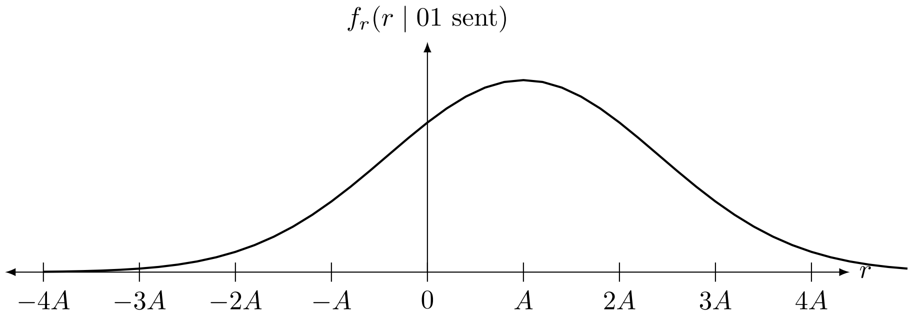
$$\begin{aligned}
P_{b|00} &= \sum_{l=0}^{M-1} \Pr\left\{\widehat{k} = l \mid 00 \text{ sent}\right\} \times \left(\frac{\#\text{ of bit positions in which 00 and } l \text{ differ}}{\#\text{ of bits per symbol}}\right) \\
&= \tfrac{0}{2}P_{00|00} + \tfrac{1}{2}P_{01|00} + \tfrac{1}{2}P_{10|00} + \tfrac{2}{2}P_{11|00} \\
&= \frac{1}{2}\left[0(1-Q_1) + 1(Q_1 - Q_3) + 1(Q_5) + 2(Q_3 - Q_5)\right] \\
&= \frac{1}{2}\left[Q_1 + Q_3 - Q_5\right]
\end{aligned}$$

If 01 was transmitted then

$r = A + w$

The likelihood function is

$$f_r(r \mid 01 \text{ sent}) = \frac{1}{\sqrt{\pi \mathcal{N}_0}} \exp\left\{-\frac{1}{\mathcal{N}_0}(r - A)^2\right\}$$



$P_{00|01} = \Pr\{r > 2A \mid 01 \text{ sent}\} = \Pr\{w > A\} = Q_1$

$P_{01|01} = \Pr\{0 < r \le 2A \mid 01 \text{ sent}\} = \Pr\{-A < w \le A\} = 1 - 2Q_1$

$P_{11|01} = \Pr\{-2A < r \le 0 \mid 01 \text{ sent}\} = \Pr\{-3A < w \le -A\} = Q_1 - Q_3$

$P_{10|01} = \Pr\{r \le -2A \mid 01 \text{ sent}\} = \Pr\{w \le -3A\} = Q_3$

The probability of a bit error given that 01 was sent is

$$\begin{aligned} P_{b|01} &= \tfrac{1}{2}P_{00|01} + \tfrac{0}{2}P_{01|01} + \tfrac{2}{2}P_{10|01} + \tfrac{1}{2}P_{11|01} \\ &= \frac{1}{2}\left[1(Q_1) + 0(1 - 2Q_1) + 2(Q_3) + 1(Q_1 - Q_3)\right] \\ &= \frac{1}{2}\left[2Q_1 + Q_3\right] \end{aligned}$$

By symmetry, if 10 is transmitted,

$$P_{b|10} = P_{b|00} = \frac{1}{2}\left[Q_1 + Q_3 - Q_5\right]$$

and if 11 is transmitted

$$P_{b|11} = P_{b|01} = \frac{1}{2}\left[2Q_1 + Q_3\right]$$

The average probability of a bit error is

$$\begin{aligned} P_b &= \sum_{m=0}^{3} P_{b|m} \Pr\{m \text{ sent}\} \\ &= \frac{1}{4}\left[P_{b|00} + P_{b|01} + P_{b|10} + P_{b|11}\right] \\ &= \frac{1}{8}\left[(Q_1 + Q_3 - Q_5) + (2Q_1 + Q_3) + (Q_1 + Q_3 - Q_5) + (2Q_1 + Q_3)\right] \\ &= \frac{1}{4}\left[3Q_1 + 2Q_3 - Q_5\right] \end{aligned}$$

Substituting for Q_n yields

$$P_b = \frac{1}{8}\left[3\text{erfc}\left(\frac{A}{\sqrt{\mathcal{N}_0}}\right) + 2\text{erfc}\left(\frac{3A}{\sqrt{\mathcal{N}_0}}\right) - \text{erfc}\left(\frac{5A}{\sqrt{\mathcal{N}_0}}\right)\right]$$

The average transmitted energy per bit is

$$\begin{aligned} \mathcal{E}_b &= \tfrac{1}{2}\mathcal{E}_s = \frac{1}{2}\sum_{m=0}^{3} \mathcal{E}_m \Pr\{m \text{ sent}\} = \frac{1}{8}\sum_{m=0}^{3} \mathcal{E}_m \\ &= \frac{1}{8}\left[(3A)^2 + (A)^2 + (-3A)^2 + (-A)^2\right] \\ &= \frac{1}{8}20A^2 = \frac{5}{2}A^2 \end{aligned}$$

The average probability of a bit error, expressed in terms of $\mathcal{E}_b$, is

$$P_b = \frac{1}{8}\left[3\text{erfc}\left(\sqrt{\frac{2\mathcal{E}_b}{5\mathcal{N}_0}}\right) + 2\text{erfc}\left(3\sqrt{\frac{2\mathcal{E}_b}{5\mathcal{N}_0}}\right) - \text{erfc}\left(5\sqrt{\frac{2\mathcal{E}_b}{5\mathcal{N}_0}}\right)\right]$$

Synchronization

Consider the more general AWGN channel model

$$r_c(t) = \alpha v_c(t-\tau) + w_c(t)$$

$v_c(t)$ – transmitted bandpass signal
$w_c(t)$ – bandpass AWGN signal
$r_c(t)$ – received bandpass signal
τ – transmission delay
α – channel attenuation

In general, the transmission delay and channel attenuation are unknown, and their effects must be taken into consideration.

Carrier Recovery

Suppose

$$v_c(t) = \mathbf{Re}\left\{v(t)\sqrt{2}e^{j2\pi f_c t}\right\}$$

is transmitted, where $v(t)$ is the complex lowpass equivalent transmitted signal. The received signal is

$$\begin{aligned} r_c(t) &= \alpha v_c(t-\tau) + w_c(t) \\ &= \alpha\mathbf{Re}\left\{v(t-\tau)\sqrt{2}e^{j2\pi f_c(t-\tau)}\right\} + w_c(t) \\ &= \alpha\mathbf{Re}\left\{v(t-\tau)\sqrt{2}e^{j2\pi f_c t}e^{j\phi_c}\right\} + w_c(t) \end{aligned}$$

where $\phi_c = -2\pi f_c \tau$ is the carrier phase uncertainty introduced by the unknown transmission delay.
If the transmission delay could be estimated extremely accurately, then it would be possible to estimate the carrier phase uncertainty. However, since the carrier frequency is usually very large, any slight error in estimating τ will lead to great uncertainty about ϕ_c.

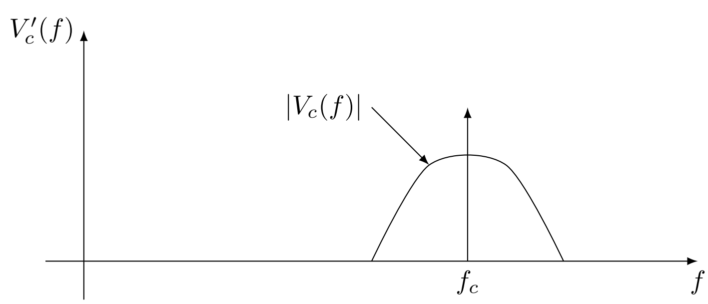
Example: Suppose the carrier frequency is $f_c = 1$ GHz, and the transmission delay is $\tau = 2$ μsec. The carrier phase uncertainty is $\phi_c = -2\pi f_c \tau = -4000\pi = 0$ radians. If the estimate of the transmission delay is $\widehat{\tau} = 2.0005$ μsec, the estimate of the phase uncertainty would be $\widehat{\phi}_c = -2\pi f_c \widehat{\tau} = -4001\pi = \pi$ radians, for an error of π radians.

Furthermore, a carrier phase uncertainty will result if there is a carrier frequency mismatch between the transmitter and receiver. If the carrier phase uncertainty is neglected, reliable data transmission is impossible.
Two basic approaches to carrier recovery are:

1. Pilot signal insertion
   - an unmodulated carrier is transmitted with the data-bearing signal
   - receiver uses narrowband filter to extract the pilot tone
   - use pilot tone for demodulation
   - requires additional power to transmit the pilot.

$$v_c'(t) = v_c(t) + K\cos(2\pi f_c t)$$

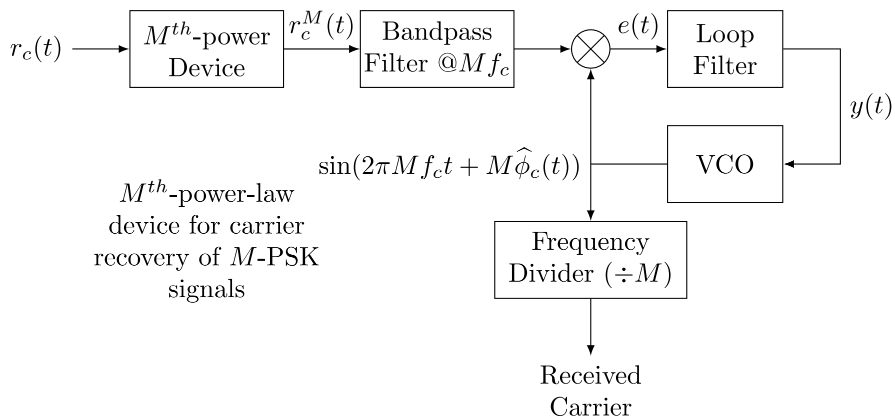
$$V_c'(f) = V_c(f) + \frac{K}{2}\delta(f-f_c) + \frac{K}{2}\delta(f+f_c)$$



2. Suppressed Carrier Extraction
   - extract carrier reference signal from data-bearing signal
   - use phase locked loop (PLL), squaring loop, Costas loop, etc ...

   Example:

   For M-PSK



M^{th}-power-law device for carrier recovery of M-PSK signals

Ignoring the noise component,

$$\begin{aligned} r_c(t) &= v_c(t) \\ &= \sum_{n=0}^{N_a-1} h_T(t-nT) A\cos(2\pi f_c t + \phi_c + \theta_n) \end{aligned}$$

where $\theta_n = \frac{2\pi}{M} a_n$ is the transmitted phase, ϕ_c is the phase uncertainty, $h_T(t)$ is the transmitted pulse shape, A is an arbitrary signal amplitude, and N_a is the number of symbols transmitted in a packet. The problem is to estimate ϕ_c, which is obscured by the transmitted data.

- M^{th}-power-law device generates harmonics of f_c.

$$\begin{aligned} r_c^2(t) &= \sum_{n=0}^{N_a-1}\sum_{m=0}^{N_a-1} h_T(t-nT)h_T(t-mT)A^2\cos(2\pi f_c t + \phi_c + \theta_n)\cos(2\pi f_c t + \phi_c + \theta_m) \\ &= \sum_{n=0}^{N_a-1} h_T^2(t-nT)A^2\cos^2(2\pi f_c t + \phi_c + \theta_n) \\ &= \sum_{n=0}^{N_a-1} h_T^2(t-nT)\frac{A^2}{2}\left[1+\cos(4\pi f_c t + 2\phi_c + 2\theta_n)\right] \\ r_c^4(t) &= \sum_{n=0}^{N_a-1}\sum_{m=0}^{N_a-1} h_T^2(t-nT)h_T^2(t-mT)\frac{A^4}{4}\left[1+\cos(4\pi f_c t + 2\phi_c + 2\theta_n)\right] \\ &\qquad \times\left[1+\cos(4\pi f_c t + 2\phi_c + 2\theta_m)\right] \\ &= \sum_{n=0}^{N_a-1} h_T^4(t-nT)\frac{A^4}{4}\left[1+\cos(4\pi f_c t + 2\phi_c + 2\theta_n)\right]^2 \\ &= \sum_{n=0}^{N_a-1} h_T^4(t-nT)\frac{A^4}{4}\left[1+2\cos(4\pi f_c t + 2\phi_c + 2\theta_n)+\cos^2(4\pi f_c t + 2\phi_c + 2\theta_n)\right] \\ &= \sum_{n=0}^{N_a-1} h_T^4(t-nT)\frac{A^4}{4}\left[1+2\cos(4\pi f_c t + 2\phi_c + 2\theta_n)+\tfrac{1}{2}+\tfrac{1}{2}\cos(8\pi f_c t + 4\phi_c + 4\theta_n)\right] \end{aligned}$$

In general,

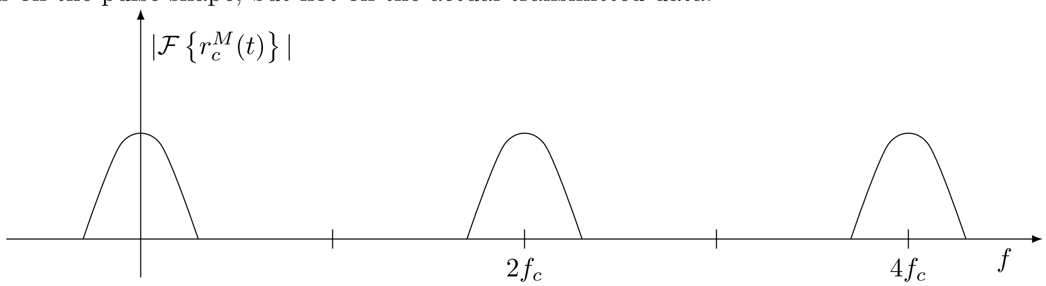
$$\begin{aligned} r_c^M(t) &= \cdots + \sum_{n=0}^{N_a-1} h_T^M(t-nT)2\left(\frac{A}{2}\right)^M\cos(2\pi M f_c t + M\phi_c + M\theta_n) \\ &= \cdots + \sum_{n=0}^{N_a-1} h_T^M(t-nT)2\left(\frac{A}{2}\right)^M\cos(2\pi M f_c t + M\phi_c + M\frac{2\pi}{M}a_n) \end{aligned}$$

$$= \cdots + \sum_{n=0}^{N_a-1} h_T^M(t-nT) 2 \left(\frac{A}{2}\right)^M \cos(2\pi M f_c t + M\phi_c)$$
$$= \cdots + x(t)\cos(2\pi M f_c t + M\phi_c)$$

where

$$x(t) = 2\left(\frac{A}{2}\right)^M \sum_{n=0}^{N_a} h_T^M(t-nT)$$

depends on the pulse shape, but not on the actual transmitted data.



– the bandpass filter isolates the harmonic
$$x(t)\cos(2\pi M f_c t + M\phi_c)$$
– the output of the multiplier is
$$\begin{aligned} e(t) &= x(t)\cos(2\pi M f_c t + M\phi_c)\sin(2\pi M f_c t + M\widehat{\phi}_c(t)) \\ &= x(t)\tfrac{1}{2}\left[\sin(4\pi M f_c t + M[\phi_c + \widehat{\phi}_c(t)]) + \sin(M[\phi_c - \widehat{\phi}_c(t)])\right] \end{aligned}$$
where $\widehat{\phi}_c(t)$ is an estimate of the carrier phase uncertainty.

– the loop filter (a narrowband filter) removes the high-frequency component, and most of $x(t)$, leaving
$$y(t) = K \sin M[\phi_c - \widehat{\phi}_c(t)] \cong KM[\phi_c - \widehat{\phi}_c(t)]$$
– the voltage controlled oscillator (VCO) produces a sinusoid $\sin[\alpha(t)]$, whose instantaneous phase is
$$\alpha(t) = 2\pi M f_c t + \frac{1}{K}\int_{-\infty}^{t} y(\tau)\, d\tau$$
where K is a gain constant.

If at time $t = t_1$,
$$\alpha(t_1) = 2\pi M f_c t_1 + M\widehat{\phi}_c(t_1)$$
then at time $t_2 > t_1$
$$\begin{aligned} \alpha(t_2) &= 2\pi M f_c t_2 + M\widehat{\phi}_c(t_1) + \frac{1}{K}\int_{t_1}^{t_2} y(\tau)\, d\tau \\ &\cong 2\pi M f_c t_2 + M\widehat{\phi}_c(t_1) + \frac{1}{K}\int_{t_1}^{t_2} KM[\phi_c - \widehat{\phi}_c(\tau)]\, d\tau \\ &\cong 2\pi M f_c t_2 + M\widehat{\phi}_c(t_1) + M[\phi_c - \widehat{\phi}_c(t_1)](t_2 - t_1) \end{aligned}$$
As $t \to \infty$, $\alpha(t) \to 2\pi M f_c t + M\phi_c$.

– the frequency divider output is
$$\sin\left(2\pi f_c t + \widehat{\phi}_c(t)\right)$$
– as $t \to \infty$, $\widehat{\phi}_c(t) \to \phi_c$, (mod $\frac{2\pi}{M}$)

– the phase tracking loop can lock onto ϕ_c with an offset of any integer multiple of $\frac{2\pi}{M}$.

– these carrier extraction techniques introduce a phase ambiguity that is an integer multiple of $\frac{2\pi}{M}$.

Note: this problem can be overcome by differentially encoding the signal prior to transmission, and differentially decoding the signal at the receiver.

Differentially Encoded M-PSK (M-DPSK)

To overcome the phase ambiguity introduced by the phase tracking loop, differential phase encoding is often used.

- For traditional (absolutely-encoded) M-PSK, if a_n is the symbol transmitted in the n^{th} symbol interval, with $a_n \in \{0, 1, \ldots, M-1\}$, then the phase transmitted in the n^{th} symbol interval is
$$\theta_n = \frac{2\pi}{M} a_n$$
- For differentially encoded M-PSK, the phase transmitted in the n^{th} symbol interval is
$$\theta_n = \theta_{n-1} + \frac{2\pi}{M} a_n$$
where θ_{n-1} is the phase transmitted in the previous symbol interval.
- Because of the phase ambiguity, the receiver decides that
$$\begin{aligned} \widehat{\theta}_{n-1} &= \theta_{n-1} + \theta_\epsilon \\ \widehat{\theta}_n &= \theta_n + \theta_\epsilon \end{aligned} \qquad \begin{pmatrix} \text{ignoring error} \\ \text{due to noise} \end{pmatrix}$$
where θ_ϵ is the phase ambiguity, with $\theta_\epsilon \in \left\{ \frac{2\pi}{M} k \;\middle|\; k = 0, 1, \ldots, M-1 \right\}$
- The difference between these two is
$$\begin{aligned} \widehat{\theta}_n - \widehat{\theta}_{n-1} &= \theta_n + \theta_\epsilon - \theta_{n-1} - \theta_\epsilon \\ &= \theta_n - \theta_{n-1} \\ &= \frac{2\pi}{M} a_n \end{aligned}$$
- Taking errors due to noise into account, note that an error in $\widehat{\theta}_n$ will cause not only an error in $\widehat{a}_n$, but also in $\widehat{a}_{n+1}$. As a result, the probability of error for M-DPSK is about twice that of M-PSK.

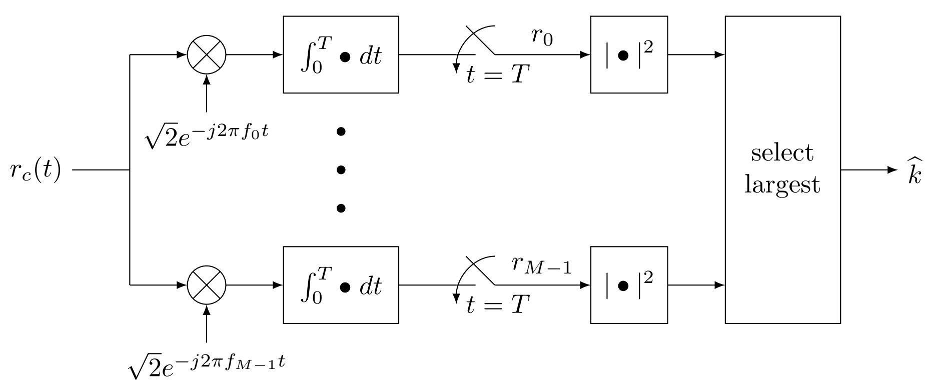
Noncoherent Receivers

An alternative to carrier recovery

- Coherent Receivers - perform carrier recovery
- Noncoherent Receivers - do not perform carrier recovery

Noncoherent receivers can only be used with certain signalling schemes

1. Noncoherent detection of frequency shift keying



$$\begin{aligned} r_m &= \int_0^T r_c(t)\sqrt{2}e^{-j2\pi f_m t}\, dt \\ &= \int_0^T \left[A\sqrt{2}\cos(2\pi f_k t + \phi_c) + w_c(t) \right] \sqrt{2}e^{-j2\pi f_m t}\, dt \\ &= 2A \int_0^T \cos(2\pi f_k t + \phi_c) e^{-j2\pi f_m t}\, dt + w_m \\ &= A \int_0^T \left[e^{j2\pi f_k t} e^{j\phi_c} + e^{-j2\pi f_k t} e^{-j\phi_c} \right] e^{-j2\pi f_m t}\, dt + w_m \\ &= A \int_0^T e^{j2\pi (f_k - f_m)t} e^{j\phi_c}\, dt + A \int_0^T e^{-j2\pi (f_k + f_m)t} e^{-j\phi_c}\, dt + w_m \end{aligned}$$

$$\begin{aligned}
&= A\frac{e^{j\phi_c}}{j2\pi(f_k-f_m)}\left[e^{j2\pi(f_k-f_m)t}\right]_0^T + A\frac{e^{-j\phi_c}}{-j2\pi(f_k+f_m)}\left[e^{-j2\pi(f_k+f_m)t}\right]_0^T + w_m \\
&= A\frac{e^{j\phi_c}}{j2\pi(f_k-f_m)}\left[e^{j2\pi(f_k-f_m)T}-1\right] + A\frac{e^{-j\phi_c}}{-j2\pi(f_k+f_m)}\left[e^{-j2\pi(f_k+f_m)T}-1\right] + w_m \\
&= A\frac{e^{j\phi_c}}{j2\pi(f_k-f_m)}\left[e^{j\pi(f_k-f_m)T}-e^{-j\pi(f_k-f_m)T}\right]e^{j\pi(f_k-f_m)T} \\
&\qquad + A\frac{e^{-j\phi_c}}{-j2\pi(f_k+f_m)}\left[e^{-j\pi(f_k+f_m)T}-e^{j\pi(f_k+f_m)T}\right]e^{-j\pi(f_k+f_m)T} + w_m \\
&= Ae^{j\phi_c}\frac{\sin(\pi[f_k-f_m]T)}{\pi(f_k-f_m)}e^{j\pi(f_k-f_m)T} + Ae^{-j\phi_c}\frac{\sin(\pi[f_k+f_m]T)}{\pi(f_k+f_m)}e^{-j\pi(f_k+f_m)T} + w_m \\
&= Ae^{j\phi_c}\frac{\sin(\pi[f_k-f_m]T)}{\pi(f_k-f_m)}e^{j\pi(f_k-f_m)T} + w_m
\end{aligned}$$

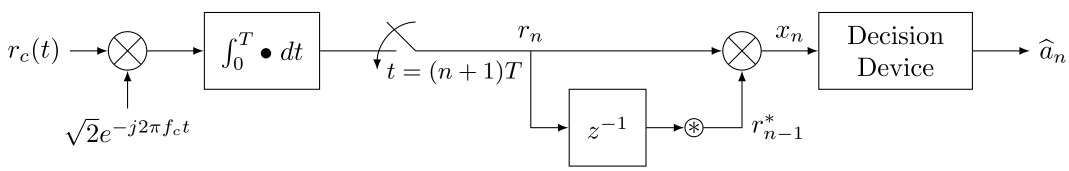
If $f_k - f_m = (k-m)\Delta f_c = \frac{k-m}{T}$ then

$$r_m = Ae^{j\phi_c}\delta_{k-m} + w_m$$

and

$$\left|r_m\right|^2 = A^2\delta_{k-m} + \text{ noise terms}$$

2. Noncoherent M-DPSK



$$r_n = \sqrt{\mathcal{E}_s}e^{j\theta_n}e^{j\phi_c} + w_n$$

$$\begin{aligned}
x_n &= r_n r_{n-1}^* \\
&= \mathcal{E}_s e^{j(\theta_n+\phi_c-\theta_{n-1}-\phi_c)} + \text{ noise terms} \\
&= \mathcal{E}_s e^{j\frac{2\pi}{M}a_n} + \text{ noise terms}
\end{aligned}$$

Timing Recovery (Symbol Synchronization)

The receiver must sample the matched filter outputs at the precise sampling instants, $t_n = nT + \tau$.
The receiver must know

- the symbol rate, $1/T$, and
- the transmission delay τ.

For correct sampling, the receiver requires a synchronized clock signal. Some approaches include:

1. Master Clock
   - transmitter and receiver are synchronized to an external master clock
   - receiver must still estimate and compensate for transmission delay
     - OK if $\tau << T$

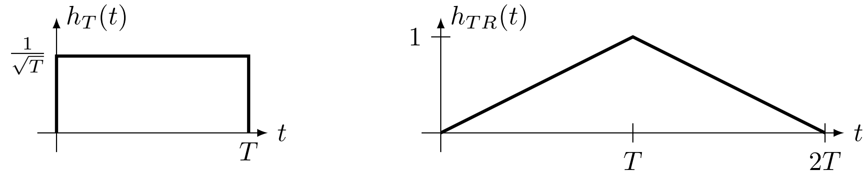
2. Transmitted Clock
   - clock signal is transmitted along with the data
   - receiver uses narrowband filter to extract clock signal from data
   - good timing recovery since clock signal has same delay as data signal
   - requires power to transmit clock, reducing power available for data
   - clock signal requires additional bandwidth

3. Extract clock signal from received data signal

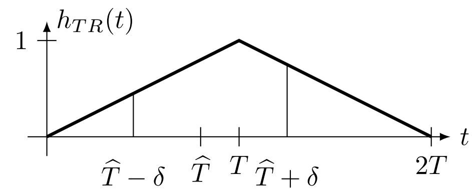
   Early-Late Gate Synchronization

   - one possible approach
   - based on the fact that the matched filter output is at a maximum at the correct sampling instant:

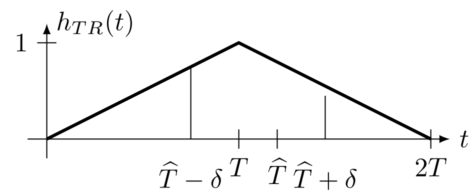
     Example: rectangular pulse



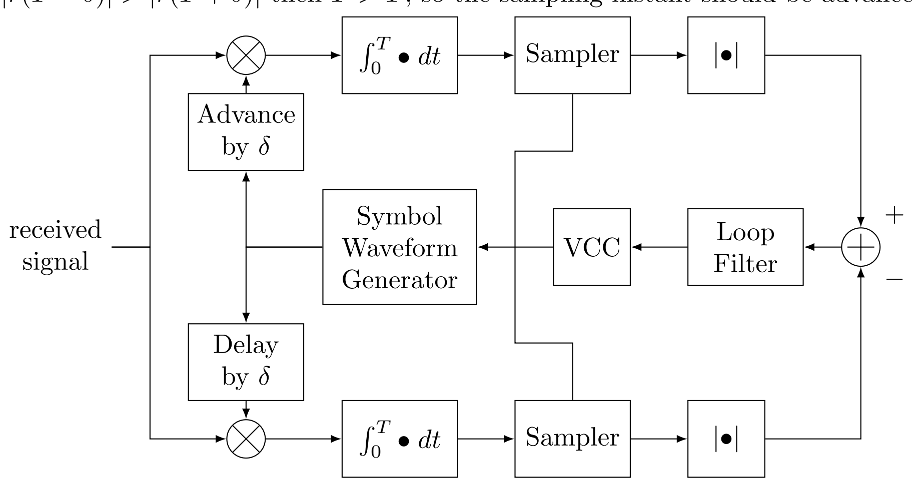
   - if $\widehat{T}$ is an estimate of the correct sampling instant, take two additional samples, one at $\widehat{T} - \delta$ and one at $\widehat{T} + \delta$
   - if $\widehat{T} < T$ then



   - if $\widehat{T} > T$ then



   - If $|r(\widehat{T} - \delta)| < |r(\widehat{T} + \delta)|$ then $\widehat{T} < T$, so the sampling instant should be delayed.
   - If $|r(\widehat{T} - \delta)| > |r(\widehat{T} + \delta)|$ then $\widehat{T} > T$, so the sampling instant should be advanced.



VCC: Voltage Controlled Clock

Summary of Bandpass Signalling

Notation:

$\mathcal{E}_s$ = transmitted energy per symbol $\quad \gamma_s = \frac{\mathcal{E}_s}{\mathcal{N}_0}$

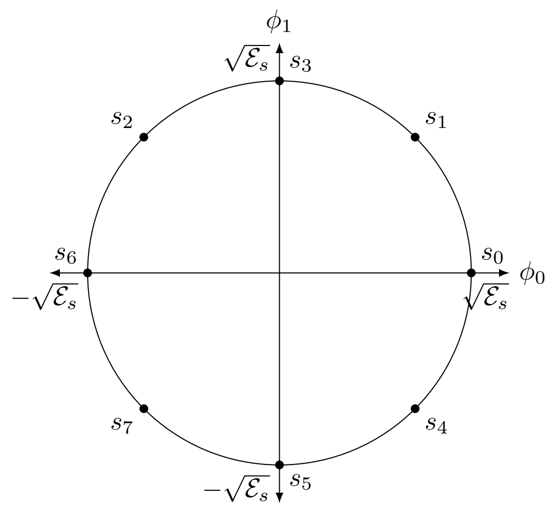
$\mathcal{E}_b$ = transmitted energy per bit $\quad \gamma_b = \frac{\mathcal{E}_b}{\mathcal{N}_0}$

Coherent M-ary Phase Shift Keying (M-PSK)

$$s_m(t) = A\cos(2\pi f_c t + \theta_m) \quad \text{with} \quad \theta_m \in \left\{0, \frac{2\pi}{M}, 2\left(\frac{2\pi}{M}\right), \ldots, (M-1)\frac{2\pi}{M}\right\}$$

$$\mathcal{E}_s = \tfrac{1}{2}A^2T$$

Signal Space Diagram ($M = 8$):



$$\phi_0(t) = \sqrt{\frac{2}{T}}\cos(2\pi f_c t) \qquad \phi_1(t) = -\sqrt{\frac{2}{T}}\sin(2\pi f_c t)$$

$$P_\varepsilon = \frac{M-1}{M} - \tfrac{1}{2}\text{erf}\left(\sqrt{\gamma_s}\sin\frac{\pi}{M}\right) - \frac{1}{\sqrt{\pi}}\int_0^{\sqrt{\gamma_s}\sin\frac{\pi}{M}} \text{erf}\left(u\cot\frac{\pi}{M}\right)\exp\{-u^2\}\,du$$

$$\cong \text{erfc}\left(\sqrt{\gamma_s}\sin\frac{\pi}{M}\right)$$

$$P_b \cong \frac{P_\varepsilon}{\log_2 M}$$

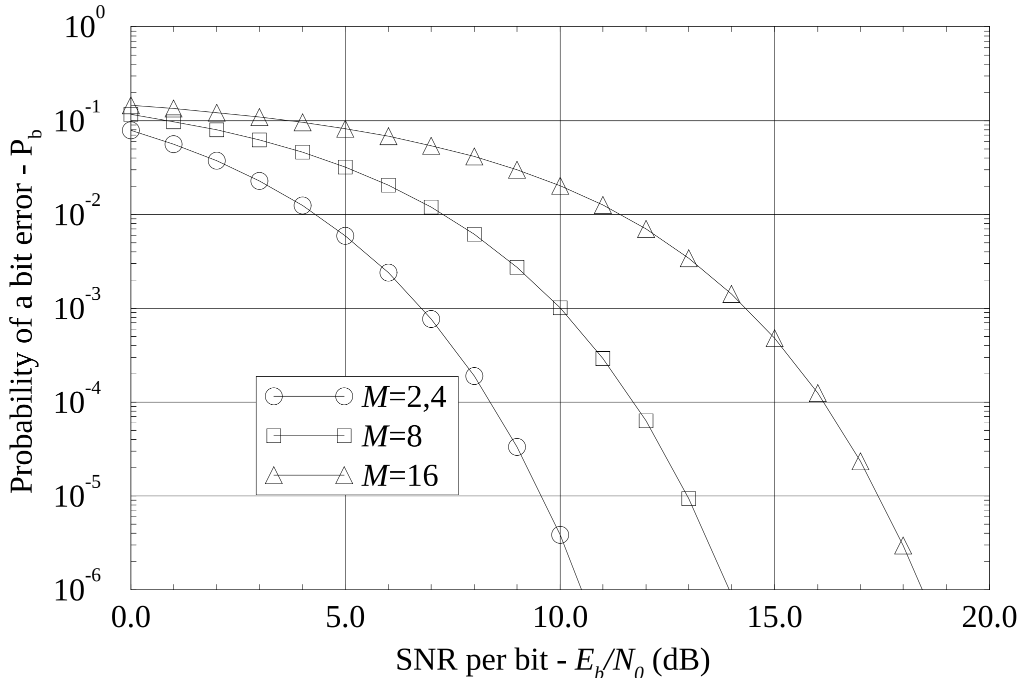
- $M = 2$

  $P_b = \tfrac{1}{2}\text{erfc}\left(\sqrt{\gamma_b}\right)$

- $M = 4$

  $$P_\varepsilon = \text{erfc}\left(\sqrt{\frac{\gamma_s}{2}}\right) - \frac{1}{4}\text{erfc}^2\left(\sqrt{\frac{\gamma_s}{2}}\right)$$

  $P_b = \tfrac{1}{2}\text{erfc}\left(\sqrt{\gamma_b}\right)$



Coherent M-ary Differential Phase Shift Keying (M-DPSK)

M-DPSK uses the same signalling scheme as M-PSK, except data is transmitted as the relative difference in the carrier phase between two consecutive symbol intervals, as opposed to the absolute carrier phase in each symbol interval.

- $M = 2$

$$P_b = \text{erfc}\left(\sqrt{\gamma_b}\right)\left[1 - \tfrac{1}{2}\text{erfc}\left(\sqrt{\gamma_b}\right)\right]$$

- $M = 4$

$$P_\varepsilon = 2\text{erfc}\left(\sqrt{\frac{\gamma_s}{2}}\right) - 2\text{erfc}^2\left(\sqrt{\frac{\gamma_s}{2}}\right) + \text{erfc}^3\left(\sqrt{\frac{\gamma_s}{2}}\right) - \frac{1}{4}\text{erfc}^4\left(\sqrt{\frac{\gamma_s}{2}}\right)$$

$$P_b = \text{erfc}\left(\sqrt{\gamma_b}\right)\left[1 - \tfrac{1}{2}\text{erfc}\left(\sqrt{\gamma_b}\right)\right]$$

Noncoherent M-ary Differential Phase Shift Keying (M-DPSK)

Noncoherent M-DPSK is similar to coherent M-DPSK, except no attempt is made to estimate the carrier phase uncertainty.

$$P_\varepsilon = \frac{\sin\frac{2\pi}{M}}{2\pi}\int_{-\pi/2}^{\pi/2}\frac{\exp\left\{-\gamma_s\left[1-\cos\frac{2\pi}{M}\cos u\right]\right\}}{1-\cos\frac{2\pi}{M}\cos u}\,du$$

$$\cong \text{erfc}\left(\sqrt{\gamma_s}\sin\frac{\pi}{\sqrt{2}M}\right)$$

$$P_b \cong \frac{P_\varepsilon}{\log_2 M}$$

- $M = 2$

$$P_b = \tfrac{1}{2}\exp\{-\gamma_b\}$$

- $M = 4$

$$P_b = Q_1(a,b) - \tfrac{1}{2}I_0(\sqrt{2}\gamma_b)\exp\{-2\gamma_b\}$$

with $a = \sqrt{2\gamma_b(1-\sqrt{\tfrac{1}{2}})}$ and $b = \sqrt{2\gamma_b(1+\sqrt{\tfrac{1}{2}})}$, where $Q_1(a,b)$ is the Markum Q function, given by

$$Q_1(a,b) = \exp\{-(a^2+b^2)/2\}\sum_{k=0}^{\infty}(\frac{a}{b})^k I_k(ab)\ ,$$

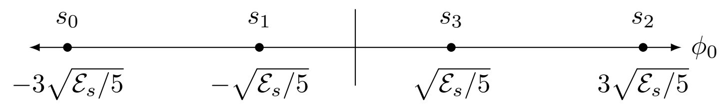
and $I_k(\cdot)$ is the modified Bessel function of order k.

Pulse Amplitue Modulation (M-PAM)

$s_m(t) = A_m \cos(2\pi f_c t)$ with $A_m \in \{-(M-1)A, \ldots, -3A, -A, A, 3A, \ldots, (M-1)A\}$

$\mathcal{E}_s = \frac{1}{6}(M^2 - 1)A^2 T$

Signal Space Diagram ($M = 4$):



$$\phi_0(t) = \sqrt{\frac{2}{T}}\cos(2\pi f_c t)$$

$$P_\varepsilon = \frac{M-1}{M}\text{erfc}\left(\sqrt{\frac{3}{(M^2-1)}\gamma_s}\right)$$

$$P_b \cong \frac{P_\varepsilon}{\log_2 M}$$

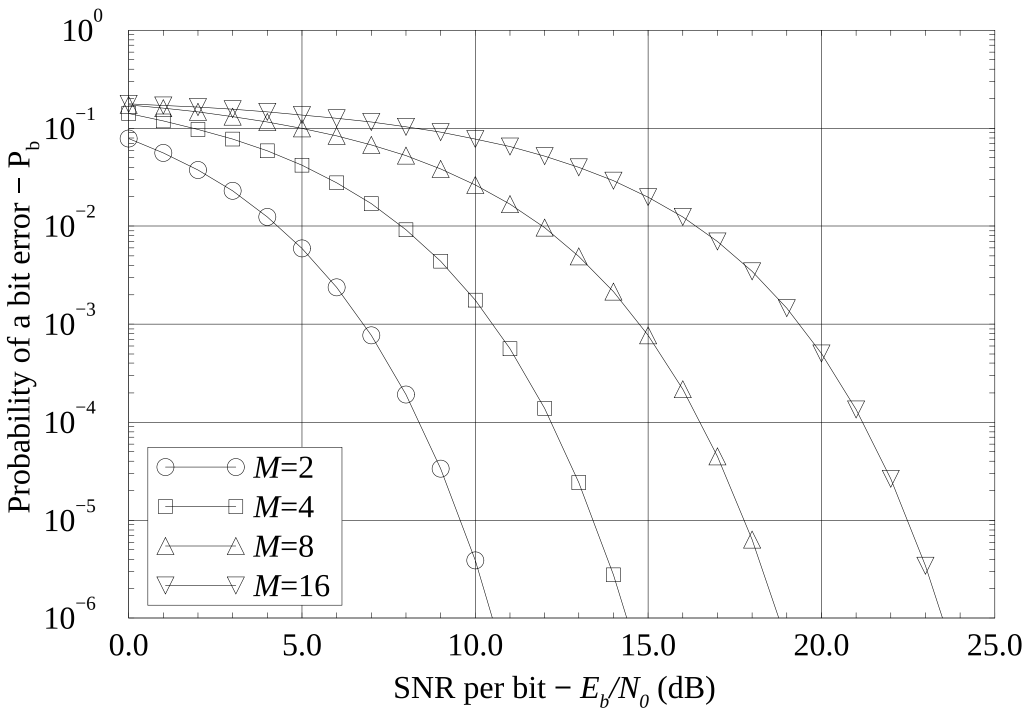
- $M = 2$

  $P_b = \frac{1}{2}\text{erfc}\left(\sqrt{\gamma_b}\right)$

- $M = 4$

  $$P_\varepsilon = \frac{3}{4}\text{erfc}\left(\sqrt{\frac{\gamma_s}{5}}\right)$$

  $$P_b = \frac{3}{8}\text{erfc}\left(\sqrt{\frac{2}{5}\gamma_b}\right) + \frac{1}{4}\text{erfc}\left(3\sqrt{\frac{2}{5}\gamma_b}\right) - \frac{1}{8}\text{erfc}\left(5\sqrt{\frac{2}{5}\gamma_b}\right)$$



M-ary Quadrature Amplitude Modulation (M-QAM)

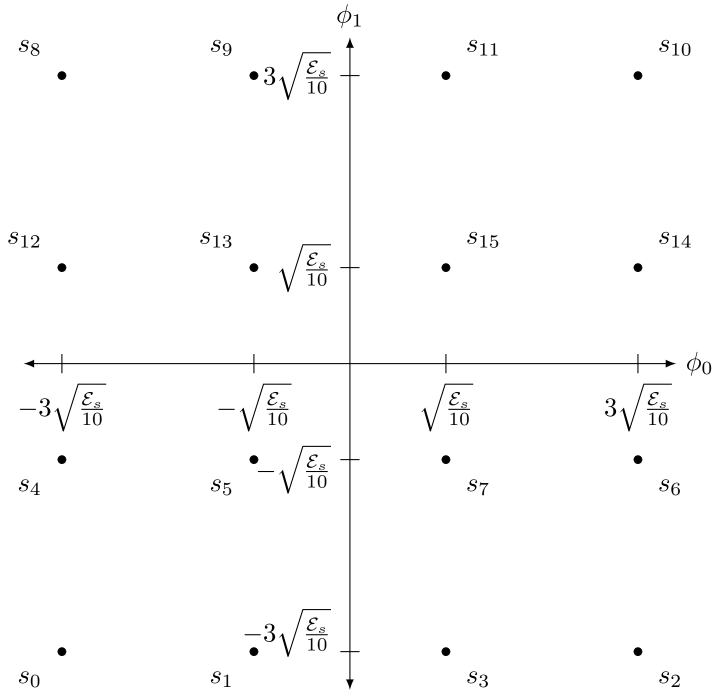
$$s_m(t) = A_{cm}\cos(2\pi f_c t) + A_{sm}\sin(2\pi f_c t)$$

with

$$A_{cm}, A_{sm} \in \{-(\sqrt{M}-1)A, \ldots, -3A, -A, A, 3A, \ldots, (\sqrt{M}-1)A\}$$

$$\mathcal{E}_s = \frac{1}{3}(M-1)A^2 T$$

Signal Space Diagram ($M = 16$):



$$\phi_0(t) = \sqrt{\frac{2}{T}}\cos(2\pi f_c t) \qquad \phi_1(t) = -\sqrt{\frac{2}{T}}\sin(2\pi f_c t)$$

$$P_\varepsilon = 2\left(1 - \frac{1}{\sqrt{M}}\right)\mathrm{erfc}\left(\sqrt{\left(\frac{3}{2(M-1)}\right)\gamma_s}\right)\left[1 - \tfrac{1}{2}\left(1 - \frac{1}{\sqrt{M}}\right)\mathrm{erfc}\left(\sqrt{\left(\frac{3}{2(M-1)}\right)\gamma_s}\right)\right]$$

$$P_b \cong \frac{P_\varepsilon}{\log_2 M}$$

- $M = 4$

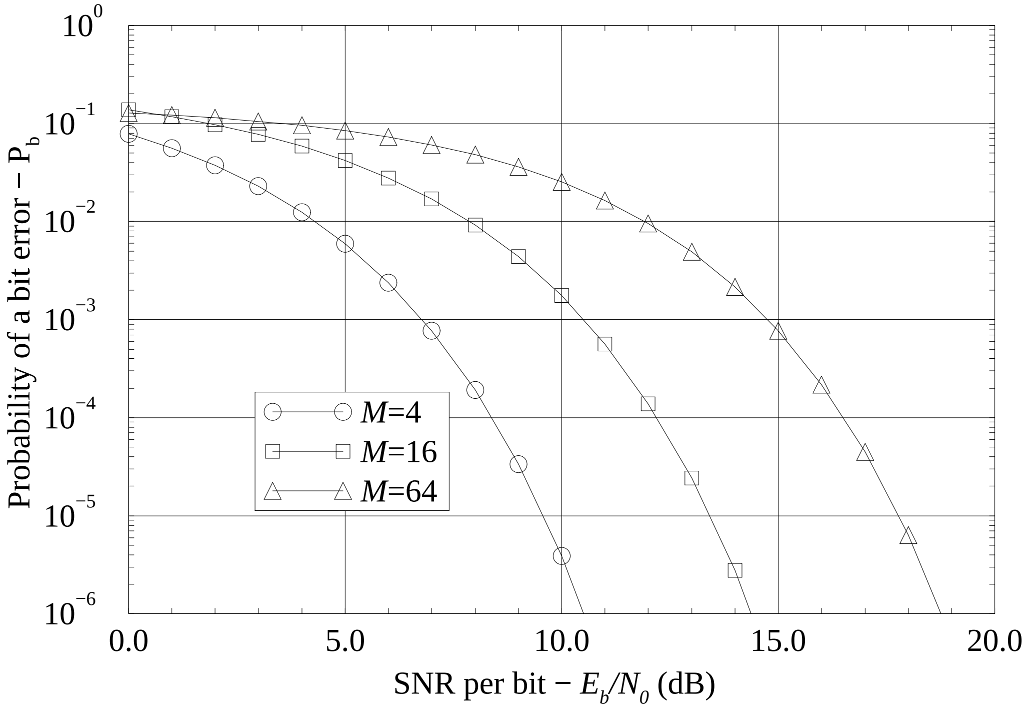
$$P_\varepsilon = \mathrm{erfc}\left(\sqrt{\frac{\gamma_s}{2}}\right) - \frac{1}{4}\mathrm{erfc}^2\left(\sqrt{\frac{\gamma_s}{2}}\right)$$

$$P_b = \tfrac{1}{2}\mathrm{erfc}\left(\sqrt{\gamma_b}\right)$$

- $M = 16$

$$P_\varepsilon = \frac{3}{2}\mathrm{erfc}\left(\sqrt{\frac{\gamma_s}{10}}\right) - \frac{9}{16}\mathrm{erfc}^2\left(\sqrt{\frac{\gamma_s}{10}}\right)$$

$$P_b = \frac{3}{8}\mathrm{erfc}\left(\sqrt{\frac{2}{5}\gamma_b}\right) + \frac{1}{4}\mathrm{erfc}\left(3\sqrt{\frac{2}{5}\gamma_b}\right) - \frac{1}{8}\mathrm{erfc}\left(5\sqrt{\frac{2}{5}\gamma_b}\right)$$



10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
Probability of a bit error − P_b
0.0
5.0
10.0
15.0
20.0
SNR per bit − E_b/N_0 (dB)
M=4
M=16
M=64

Coherent M-ary Frequency Shift Keying (M-FSK)

$$s_m(t) = A\cos[2\pi(f_c + m\Delta f_c)t] \quad \text{with} \quad \Delta f_c = \frac{1}{2T} \qquad \mathcal{E}_s = \tfrac{1}{2}A^2T$$

$$\phi_k(t) = \sqrt{\frac{2}{T}}\cos[2\pi(f_c + k\Delta f_c)t]$$

$$P_\varepsilon = 1 - \frac{1}{\sqrt{\pi}}\int_{-\infty}^{\infty}\left[1 - \tfrac{1}{2}\text{erfc}\left(u + \sqrt{\gamma_s}\right)\right]^{M-1}\exp\{-u^2\}\,du$$

$$\cong \tfrac{1}{2}(M-1)\text{erfc}\left(\sqrt{\tfrac{1}{2}\gamma_s}\right)$$

$$P_b = \frac{M/2}{M-1}P_\varepsilon$$

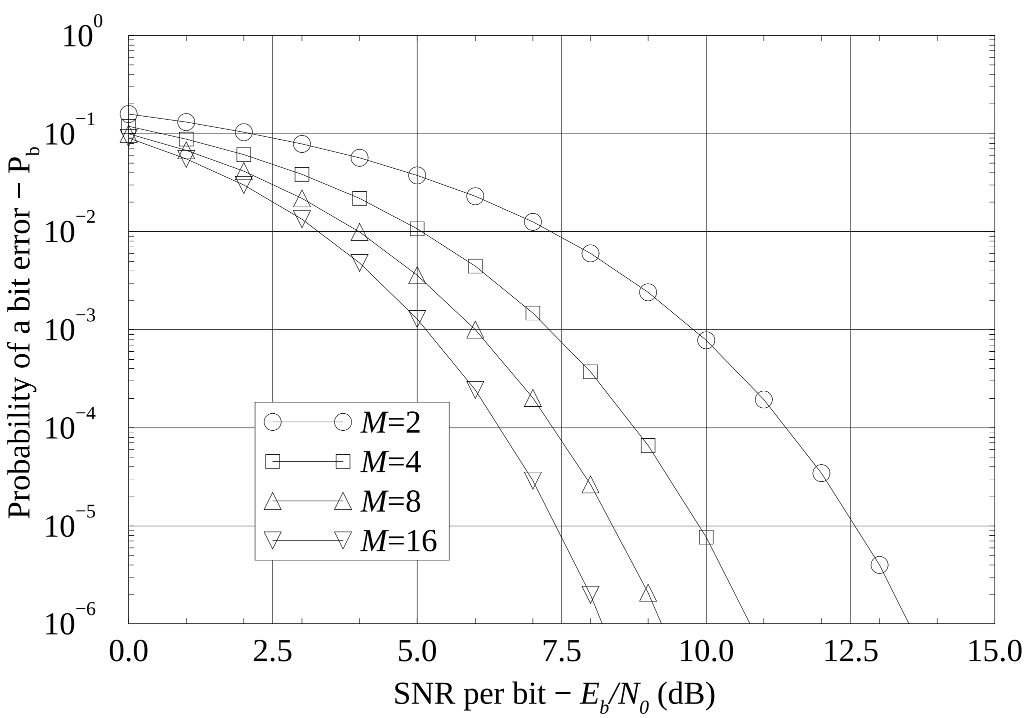
- $M = 2$

$$P_b = \tfrac{1}{2}\text{erfc}\left(\sqrt{\tfrac{1}{2}\gamma_b}\right)$$

- $M = 4$

$$P_\varepsilon \cong \frac{3}{2}\text{erfc}\left(\sqrt{\tfrac{1}{2}\gamma_s}\right)$$

$$P_b \cong \text{erfc}\left(\sqrt{\gamma_b}\right)$$



Noncoherent M-ary Frequency Shift Keying (M-FSK)

Noncoherent M-FSK is the same as M-FSK, except that no attempt at carrier phase recover is made by the receiver.

$$P_\varepsilon = \sum_{m=1}^{M-1} (-1)^{m+1} \binom{M-1}{m} \frac{1}{m+1} \exp\left\{-\frac{m}{m+1}\gamma_s\right\}$$

$$P_b = \frac{M/2}{M-1} P_\varepsilon$$

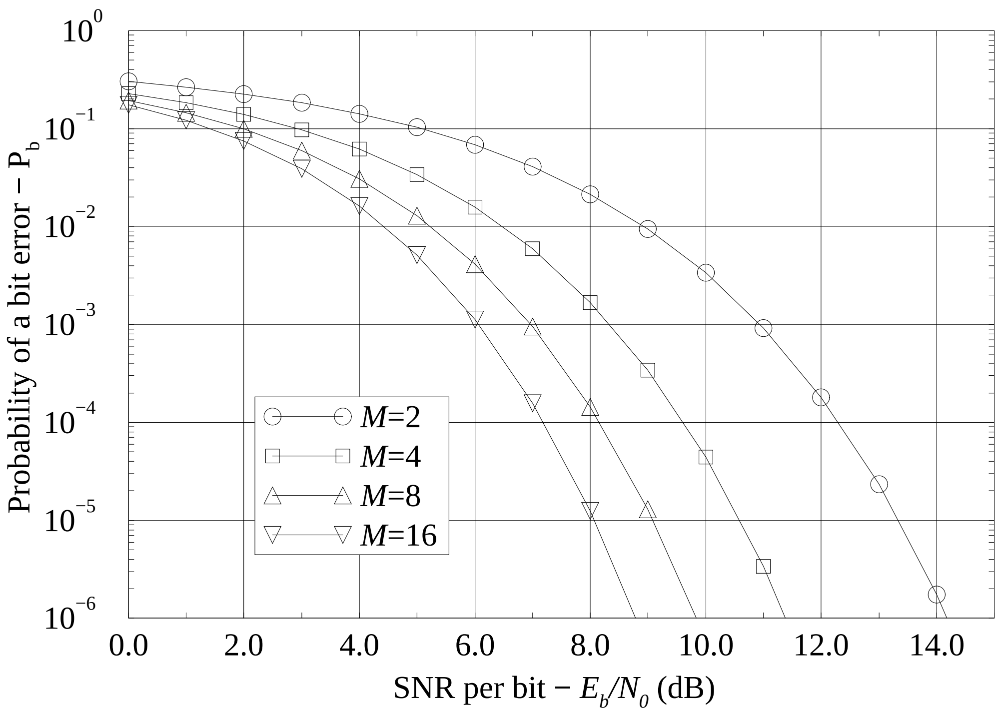
- $M = 2$

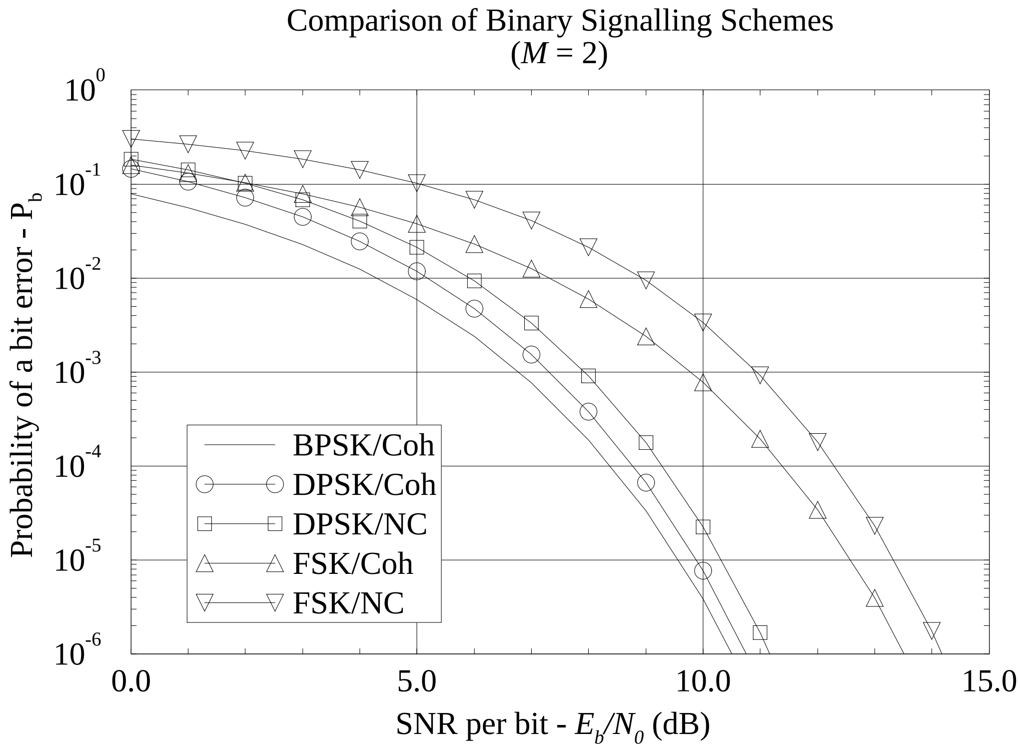
$$P_b = \tfrac{1}{2} \exp\left\{-\tfrac{1}{2}\gamma_b\right\}$$

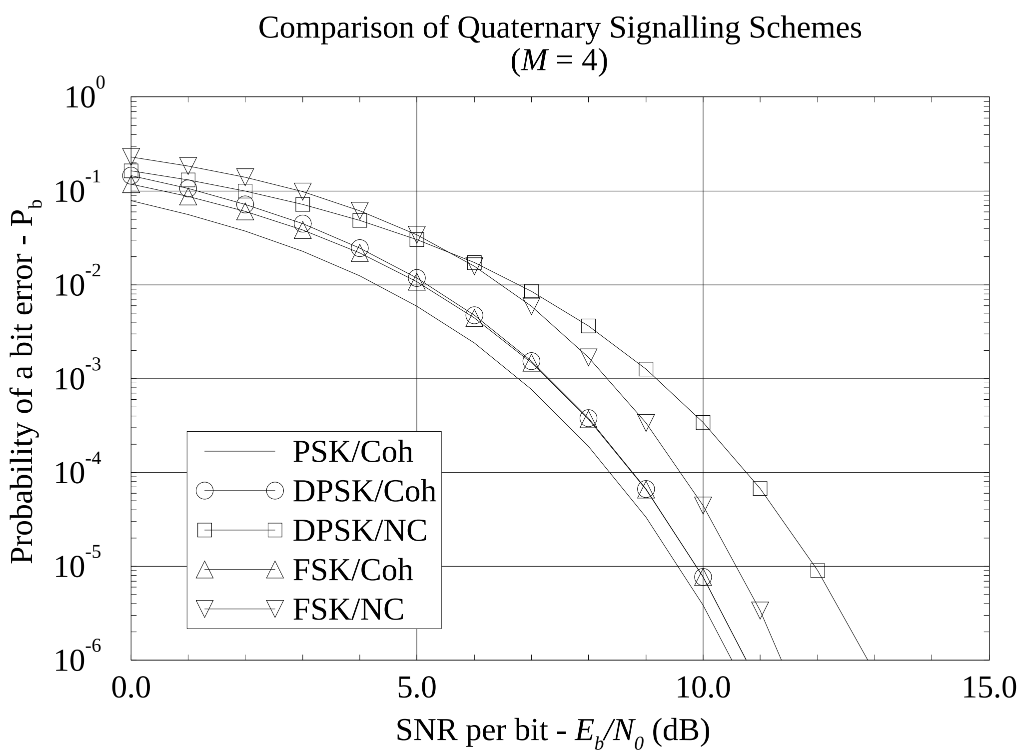
- $M = 4$

$$P_\varepsilon = \frac{3}{2} \exp\{-\tfrac{1}{2}\gamma_s\} - \exp\{-\tfrac{2}{3}\gamma_s\} + \tfrac{1}{4} \exp\{-\tfrac{3}{4}\gamma_s\}$$

$$P_b = \exp\{-\gamma_b\} - \frac{2}{3} \exp\{-\tfrac{4}{3}\gamma_b\} + \frac{1}{6} \exp\{-\tfrac{3}{2}\gamma_b\}$$




Comparison of Binary Signalling Schemes
(M = 2)
Probability of a bit error - Pb
SNR per bit - Eb/N0 (dB)
BPSK/Coh
DPSK/Coh
DPSK/NC
FSK/Coh
FSK/NC
0.0
5.0
10.0
15.0


Comparison of Quaternary Signalling Schemes
(M = 4)
Probability of a bit error - Pb
SNR per bit - Eb/N0 (dB)
PSK/Coh
DPSK/Coh
DPSK/NC
FSK/Coh
FSK/NC
0.0
5.0
10.0
15.0

Information Theory and Channel Capacity

Motivation:

Consider M-FSK. According to the union bound, the probability of a symbol error is

$$P_\varepsilon = P_{\varepsilon|k} \leq \sum_{\substack{l=0 \\ l\neq k}}^{M-1} \tfrac{1}{2}\text{erfc}\left(\frac{d_{k,l}}{2\sqrt{\mathcal{N}_0}}\right) \qquad \left(d_{k,l} = \sqrt{2\mathcal{E}_s}\right)$$

$$= (M-1)\tfrac{1}{2}\text{erfc}\left(\sqrt{\frac{\mathcal{E}_s}{2\mathcal{N}_0}}\right) \qquad \left(\mathcal{E}_b = \frac{\mathcal{E}_s}{\log_2 M}\right)$$

$$= (M-1)\tfrac{1}{2}\text{erfc}\left(\sqrt{\frac{\log_2 M}{2}\frac{\mathcal{E}_b}{\mathcal{N}_0}}\right) .$$

Question: What happens to P_ε as $M \to \infty$?

Fact: $\tfrac{1}{2}\text{erfc}\,(x) \leq e^{-x^2}$ for $x \geq 0$.

Therefore

$$P_\varepsilon \leq (M-1)\exp\left\{-\frac{\log_2 M}{2}\left(\frac{\mathcal{E}_b}{\mathcal{N}_0}\right)\right\}$$

$$< M\exp\left\{-\frac{\log_2 M}{2}\left(\frac{\mathcal{E}_b}{\mathcal{N}_0}\right)\right\} \qquad \left(M = e^{\ln M} = e^{\log_2 M \ln 2}\right)$$

$$= \exp\{\log_2 M \ln 2\}\exp\left\{-\frac{\log_2 M}{2}\left(\frac{\mathcal{E}_b}{\mathcal{N}_0}\right)\right\}$$

$$= \exp\left\{-\frac{\log_2 M}{2}\left(\frac{\mathcal{E}_b}{\mathcal{N}_0} - 2\ln 2\right)\right\} .$$

This is an upper bound on the probability of a symbol error.

If $\dfrac{\mathcal{E}_b}{\mathcal{N}_0} > 2\ln 2 = 1.39$ (1.42 dB) then $P_\varepsilon \to 0$ as $M \to \infty$.

Note: The union bound is not very tight for small $\frac{\mathcal{E}_b}{\mathcal{N}_0}$

The actual probability of error is

$$P_\varepsilon = \int_{-\infty}^{\infty}\left(1 - \left[1 - \tfrac{1}{2}\text{erfc}\,(u)\right]^{M-1}\right)\frac{1}{\sqrt{\pi}}\exp\left\{-\left(u - \sqrt{\frac{\mathcal{E}_s}{\mathcal{N}_0}}\right)^2\right\} du$$

A tighter bound (for $\mathcal{E}_b/\mathcal{N}_0 < 4\ln 2$) is

$$P_\varepsilon < 2\exp\left\{-\log_2 M\left(\frac{\mathcal{E}_b}{\mathcal{N}_0} - \ln 2\right)\right\}$$

If $\dfrac{\mathcal{E}_b}{\mathcal{N}_0} > \ln 2 = 0.693$ (-1.6 dB) then $P_\varepsilon \to 0$ as $M \to \infty$.

Note: This minimum SNR is the *Shannon limit* for the AWGN channel.

If the SNR is greater than the Shannon limit it is possible to achieve an arbitrarily low probability of error. To do so using M-FSK signalling, however, requires unlimited bandwidth.

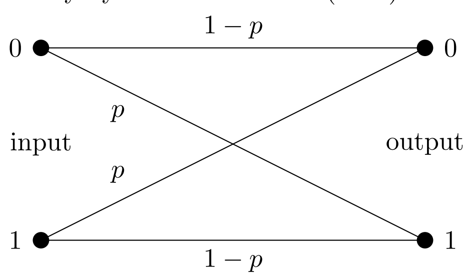
Question: Is it possible to achieve an arbitrarily low probability of error with finite bandwidth?

An alternative approach to achieving an arbitrarily low probability of error is through the use of channel coding (error correction).
Consider the L-repetition code, where each bit is transmitted L times (for some odd-valued L).

$$0 \rightarrow 000\cdots 0$$
$$1 \rightarrow \underbrace{111\cdots 1}_{L}$$

Suppose each code bit is transmitted over a binary symmetric channel with a cross-over probability of $p < \frac{1}{2}$.

Note: Binary Symmetric Channel (BSC)



Channel Transition Probabilities:

$$\Pr\{0 \text{ received} \mid 0 \text{ sent}\} = 1 - p$$
$$\Pr\{1 \text{ received} \mid 0 \text{ sent}\} = p$$
$$\Pr\{0 \text{ received} \mid 1 \text{ sent}\} = p$$
$$\Pr\{1 \text{ received} \mid 1 \text{ sent}\} = 1 - p$$

Use the *majority logic decision rule*: If more 0's are received than 1's, decide that the message bit was a 0, otherwise decide that the message bit was a 1.

Let $Y_k \in \{0, 1\}$ denote the value of the k^{th} received code bit, for $1 \leq k \leq L$, and let

$$D = \sum_{k=1}^{L} Y_k$$

be the total number of 1's received. D is a binomial random variable.
The majority logic decision rule is:

If $D < L/2$ decide that the message bit was a 0.
If $D > L/2$ decide that the message bit was a 1.

The probability of a message bit error is

$$\begin{aligned} P_{\text{mb}} = P_{\text{mb}|0} &= \Pr\{D > L/2 \mid \text{message 0 sent}\} \\ &= \sum_{l=\lceil \frac{L}{2} \rceil}^{L} \Pr\{D = l \mid \text{message 0 sent}\} \\ &= \sum_{l=\lceil \frac{L}{2} \rceil}^{L} \binom{L}{l} p^l (1-p)^{L-l} \end{aligned}$$

For large L, use the Chernoff bound:

$$\begin{aligned} P_{\text{mb}} &= \Pr\{D > L/2 \mid \text{message 0 sent}\} \\ &\leq [4p(1-p)]^{L/2} \end{aligned}$$

For $p < \frac{1}{2}$, $P_{\text{mb}} \rightarrow 0$ as $L \rightarrow \infty$.
However, since L channel uses are required to transmit a single message bit, the transmission rate is

$$R = \frac{1}{L} \text{ message bits per channel use}$$

As $L \rightarrow \infty$, $R \rightarrow 0$.

Question: Is it possible to achieve an arbitrarily low probability of error without reducing the transmission rate to zero?

Channel Coding Theorem:

It is possible to transmit information reliably (with an arbitrarily low probability of error) at any rate R which is less than the channel capacity, C. Furthermore, it is not possible to transmit information reliably at a rate greater than the channel capacity.

Introduction to Information Theory

Discrete Memoryless Source:

A discrete memoryless source draws symbols randomly from an alphabet

$$\mathcal{X} = \{x_0, x_1, \ldots, x_{K-1}\}$$

containing K symbols. The output is modelled as a random variable, X, with probabilities

$$P(x_k) = \Pr\{X = x_k\} \qquad \forall\, 0 \le k \le K-1$$

That is, $P(x_k)$ is the probability that the source emits symbol $x_k \in \mathcal{X}$.

Discrete $\rightarrow$ symbols are drawn from a discrete set
Memoryless $\rightarrow$ output at time n does not depend on outputs at other times.

Information:

How much information does a certain event represent?

- information is related to uncertainty or surprise.
- the occurrence of an unlikely event provides more information than the occurrence of a likely event.
- information is inversely proportional to probability.

Logarithmic Measure of Information:

The information gained after observation of event $X = x_k$ is

$$I(x_k) = \log_b \frac{1}{P(x_k)}$$

If $b = 2$ then units are *bits*
If $b = e$ then units are *nats*

Properties of $I(x_k)$:

1. $I(x_k) = 0$ if $P(x_k) = 1$.
   No information is gained if we are certain of the outcome of an event before it occurs.
2. $I(x_k) \ge 0$ since $0 < P(x_k) \le 1$.
   The observation of an event never leads to a loss of information.
3. $I(x_k) > I(x_l)$ if $P(x_k) < P(x_l)$.
   More information is gained when low-probability events occur.

Entropy:

The entropy of a discrete memoryless source is

$$\begin{aligned} H(X) &= \mathbf{E}\left[I(x_k)\right] \\ &= \sum_{k=0}^{K-1} I(x_k)\Pr\{X = x_k\} \\ &= \sum_{k=0}^{K-1} P(x_k)\log_2 \frac{1}{P(x_k)} \end{aligned}$$

This is a measure of the *average information* produced by the source per symbol.

Example: Fair Coin Toss

$$X = \begin{cases} H, & \text{with a probability of } \frac{1}{2} \\ T, & \text{with a probability of } \frac{1}{2} \end{cases}$$

The entropy is

$$H(X) = \tfrac{1}{2}\log_2 \frac{1}{1/2} + \tfrac{1}{2}\log_2 \frac{1}{1/2} = \tfrac{1}{2} + \tfrac{1}{2} = 1 \text{ bit}$$

Example:

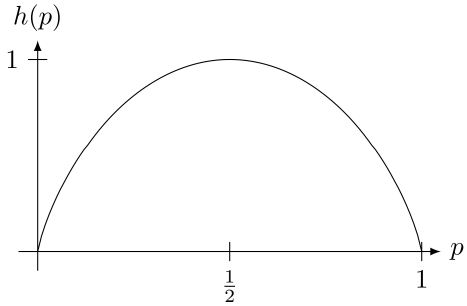
Binary Memoryless Source

$$X = \begin{cases} 0, & \text{with probability } 1-p \\ 1, & \text{with probability } p \end{cases}$$

The entropy is

$$H(X) = (1-p)\log_2 \frac{1}{(1-p)} + p\log_2 \frac{1}{p} \triangleq h(p)$$

where $h(p)$ is the *binary entropy function.*



Discrete Memoryless Channel (DMC):

– input and output alphabets are discrete.
– output at time n depends only on input at time n.

The channel is specified by an input alphabet

$\mathcal{X} = \{x_0, x_1, \ldots, x_{K-1}\}$,

an output alphabet

$\mathcal{Y} = \{y_0, y_1, \ldots, y_{L-1}\}$,

and a set of transition probabilities

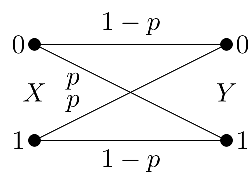
$P(y_l|x_k) = \Pr\{Y = y_l \mid X = x_k\}$

for $0 \le k \le K-1$ and $0 \le l \le L-1$. The input and output are modelled as discrete random variables.

Example: Binary Symmetric Channel (BSC)

$\mathcal{X} = \{0,1\} \qquad (K=2)$

$\mathcal{Y} = \{0,1\} \qquad (L=2)$



$P(0|0) = 1-p \qquad P(0|1) = p \qquad P(1|0) = p \qquad P(1|1) = 1-p$

Note: For the AWGN channel with BPSK, the crossover probability is

$$p = \tfrac{1}{2}\text{erfc}\left(\sqrt{\frac{\mathcal{E}_b}{\mathcal{N}_0}}\right)$$

Conditional Information:

The *conditional information* is defined as

$$I(x_k|y_l) = \log_2 \frac{1}{P(x_k|y_l)}$$

This is information about the event $X = x_k$ after event $Y = y_l$ has been observed.

Conditional Entropy:

The *conditional entropy*, which is based on the conditional information, is

$$\begin{aligned} H(X|Y) &= \mathbf{E}\left[I(x_k|y_l)\right] \\ &= \sum_{k=0}^{K-1}\sum_{l=0}^{L-1} P(x_k, y_l)\log_2 \frac{1}{P(x_k|y_l)} \end{aligned}$$

This measures the amount of uncertainty remaining in X after observing Y.

Theorem: $0 \le H(X|Y) \le H(X)$.

Proof: Since $0 \le P(x_k|y_l) \le 1$,

$$\log_2 \frac{1}{P(x_k|y_l)} \ge 0$$

Since $P(x_k, y_l)$ is also non-negative, $H(X|Y) \ge 0$.
To show the upper bound, we use the following lemma.

Lemma: The Fundamental Inequality

Let $\{p_i\}$ and $\{s_i\}$ be any two sets of N real numbers with the properties

$$\sum_{i=0}^{N-1} p_i = 1\ , \qquad \sum_{i=0}^{N-1} s_i = 1\ , \qquad p_i \geq 0, \quad s_i \geq 0\ .$$

Then

$$\sum_{i=0}^{N-1} p_i \ln \frac{s_i}{p_i} \leq 0 \qquad \text{(equal iff } s_i = p_i \ \forall\ i)$$

As a result,

$$\sum_{i=0}^{N-1} p_i \ln \frac{1}{p_i} \leq \sum_{i=0}^{N-1} p_i \ln \frac{1}{s_i} \qquad \text{(equal iff } s_i = p_i \ \forall\ i)$$

Proof:

$$\begin{aligned}
\ln x &\leq x - 1 && \text{(equal iff } x = 1) \\
\ln \frac{s_i}{p_i} &\leq \frac{s_i}{p_i} - 1 && \text{(equal iff } \tfrac{s_i}{p_i} = 1 \text{ or } s_i = p_i \ \forall\ i) \\
p_i \ln \frac{s_i}{p_i} &\leq s_i - p_i && \text{(equal iff } s_i = p_i \ \forall\ i) \\
\sum_{i=0}^{N-1} p_i \ln \frac{s_i}{p_i} &\leq \sum_{i=0}^{N-1} (s_i - p_i) = \sum_{i=0}^{N-1} s_i - \sum_{i=0}^{N-1} p_i = 0 && \text{(equal iff } s_i = p_i \ \forall\ i)
\end{aligned}$$

Since

$$H(X) = \sum_{k=0}^{K-1} P(x_k) \log_2 \frac{1}{P(x_k)} = \sum_{k=0}^{K-1} P(x_k) \log_2 \frac{1}{P(x_k)} \sum_{l=0}^{L-1} P(y_l|x_k) = \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} P(x_k, y_l) \log_2 \frac{1}{P(x_k)}$$

We have

$$\begin{aligned}
H(X|Y) - H(X) &= \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} P(x_k, y_l) \left[\log_2 \frac{1}{P(x_k|y_l)} - \log_2 \frac{1}{P(x_k)} \right] \\
&= \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} P(x_k, y_l) \log_2 \frac{P(x_k)}{P(x_k|y_l)} \\
&= \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} P(x_k, y_l) \log_2 \frac{P(x_k)P(y_l)}{P(x_k, y_l)}
\end{aligned}$$

Using the fundamental inequality, with $p_{k,l} = P(x_k, y_l)$ and $s_{k,l} = P(x_k)P(y_l)$, we have

$$H(X|Y) - H(X) \leq 0$$

with equality iff $P(x_k, y_l) = P(x_k)P(y_l)$ for all k, l. Therefore

$$H(X|Y) \leq H(X)$$

with equality iff X and Y are independent.

Since $0 \leq H(X|Y) \leq H(X)$, the uncertainty remaining in X after observing Y will never exceed the uncertainty in X prior to observing Y.

If X and Y are independent then observing Y will not change the uncertainty in X.

If $Y = X$ then observing Y will leave no uncertainty in X.

Average Mutual Information:

The *average mutual information* is

$$I(X;Y) = H(X) - H(X|Y)$$

This is the amount of information about X that is conveyed by observing Y.

Note: $I(X;Y) \geq 0$

Some information is conveyed, unless X and Y are independent.

The average mutual information can be expressed as

$$I(X;Y) = \sum_{k=0}^{K-1}\sum_{l=0}^{L-1} P(x_k, y_l)\log_2 \frac{P(x_k|y_l)}{P(x_k)} = \sum_{k=0}^{K-1}\sum_{l=0}^{L-1} P(x_k, y_l)\log_2 \frac{P(y_l|x_k)}{P(y_l)} = I(Y;X)$$

Note: The average mutual information is symmetric. That is, the information gained about X by observing Y equals the information gained about Y by observing X.

The average mutual information can also be expressed as

$$I(X;Y) = \sum_{k=0}^{K-1}\sum_{l=0}^{L-1} P(x_k)P(y_l|x_k)\log_2 \frac{P(y_l|x_k)}{\sum_{i=0}^{K-1} P(x_i)P(y_l|x_i)}$$

which depends only on the channel transition probabilities, $P(y_l|x_k)$, and on the input probability distribution, $P(x_k)$.

Note: The amount of information conveyed depends not only on the channel, but also on the input distribution.

Channel Capacity:

The *channel capacity* is the maximum average mutual information that can be conveyed per use of the channel, where the maximization is performed over the input probability distribution.

$$C = \max_{\{P(X_k)\}} I(X;Y) \qquad \text{bits per channel use}$$

Example:

Binary Symmetric Channel

Source	Channel
$\Pr\{X=0\} = 1-\alpha$	$\Pr\{Y=0\|X=0\} = 1-p$
$\Pr\{X=1\} = \alpha$	$\Pr\{Y=0\|X=1\} = p$
	$\Pr\{Y=1\|X=0\} = p$
	$\Pr\{Y=1\|X=1\} = 1-p$

For a given cross-over probability, p, find α which maximizes the average mutual information

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \end{aligned}$$

Note: The conditional entropy $H(Y|X)$ can be expressed as

$$\begin{aligned} H(Y|X) &= \sum_{k=0}^{K-1}\sum_{l=0}^{L-1} P(x_k, y_l)\log_2 \frac{1}{P(y_l|x_k)} \\ &= \sum_{k=0}^{K-1}\left(\sum_{l=0}^{L-1} P(y_l|x_k)\log_2 \frac{1}{P(y_l|x_k)}\right)P(x_k) \\ &= \sum_{k=0}^{K-1} H(Y|X=x_k)P(x_k) \end{aligned}$$

where

$$H(Y|X=x_k) = \sum_{l=0}^{L-1} P(y_l|x_k)\log_2 \frac{1}{P(y_l|x_k)}$$

For the BSC,

$$\begin{aligned} H(Y|X=0) &= \sum_{l=0}^{1} \Pr\{Y=l \mid X=0\}\log_2 \frac{1}{\Pr\{Y=l \mid X=0\}} \\ &= p\log_2 \frac{1}{p} + (1-p)\log_2 \frac{1}{1-p} \\ &= h(p) \end{aligned}$$

and

$$\begin{aligned} H(Y|X=1) &= \sum_{l=0}^{1} \Pr\{Y=l \mid X=1\}\log_2 \frac{1}{\Pr\{Y=l \mid X=1\}} \\ &= (1-p)\log_2 \frac{1}{(1-p)} + p\log_2 \frac{1}{p} \\ &= h(p) \end{aligned}$$

so

$$\begin{aligned} H(Y|X) &= H(Y|X=0)\Pr\{X=0\} + H(Y|X=1)\Pr\{X=1\} \\ &= (1-\alpha)h(p) + \alpha h(p) \\ &= h(p) \end{aligned}$$

which does not depend on the input probability distribution (α). The average mutual information is therefore

$$I(X;Y) = H(Y) - h(p)$$

To find the channel capacity it is necessary to find α which maximizes $H(Y)$.

Since

$$\begin{aligned} \Pr\{Y=0\} &= \Pr\{Y=0|X=0\}\Pr\{X=0\} + \Pr\{Y=0|X=1\}\Pr\{X=1\} \\ &= (1-p)(1-\alpha) + p\alpha \\ &= q\ , \end{aligned}$$

where $q = (1-p)(1-\alpha) + p\alpha$, and

$$\Pr\{Y=1\} = 1 - \Pr\{Y=0\} = 1 - q\ ,$$

the entropy of Y is

$$H(Y) = q\log_2\frac{1}{q} + (1-q)\log_2\frac{1}{1-q} = h(q)\ ,$$

which is the binary entropy function evaluated at q.

Since $h(q)$ is at a maximum at $q = \frac{1}{2}$, it is necessary to find α such that

$$(1-p)(1-\alpha) + p\alpha = \tfrac{1}{2}$$

Case 1: $p = \frac{1}{2}$

Then $\frac{1}{2}(1-\alpha) + \frac{1}{2}\alpha = \frac{1}{2}$, or $1 = 1$, so α can be arbitrarily between 0 and 1.

Case 2: $p \neq \frac{1}{2}$

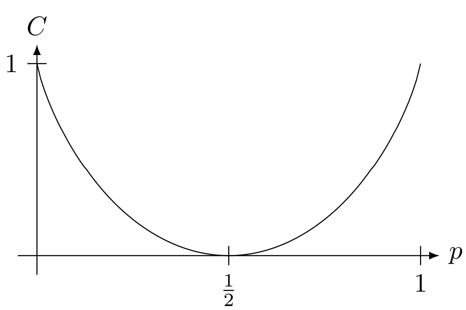
$$\begin{aligned} 1 - p - \alpha + p\alpha + p\alpha &= \tfrac{1}{2} \\ \alpha(2p-1) &= \tfrac{1}{2} - 1 + p \\ \alpha &= \frac{p - \frac{1}{2}}{2p-1} = \tfrac{1}{2} \end{aligned}$$

Therefore, $H(Y)$ is maximized when $\alpha = \frac{1}{2}$.

Therefore, $I(X;Y)$ is maximized when $\alpha = \frac{1}{2}$.

The channel capacity for the binary symmetric channel is

$$C = h(\tfrac{1}{2}) - h(p) = 1 - h(p) \qquad \text{bits per channel use}$$

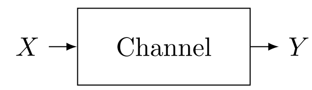


Channel Coding Theorem:

It is possible to transmit information reliably at any rate R which is less than the channel capacity, C. Furthermore, it is not possible to transmit information reliably at a rate greater than the channel capacity.

Continuously-Distributed Discrete-Time Channels

Now suppose we wish to convey a continuously distributed random variable over a continuously defined channel. X is the input to the channel and Y is the output, where X and Y are continuous random variables with a joint pdf of $f_{X,Y}(x,y)$, marginal pdf's of $f_X(x)$ and $f_Y(y)$, and conditional pdf's of $f_{X|Y}(x|y)$ and $f_{Y|X}(y|x)$.



To calculate the capacity of the channel it is necessary to extend the concepts of information theory to continuous random variables.

Differential Entropy:

For a continuous random variable, X, with pdf $f_X(x)$, the differential entropy is

$$H(X) = \mathbf{E}\left[\log_2 \frac{1}{f_X(x)}\right] = \int_{-\infty}^{\infty} f_X(x)\log_2 \frac{1}{f_X(x)}\,dx$$

This definition is similar to ordinary entropy of a discrete random variable

Note: Differential entropy is NOT a measure of information. Since X can take on an infinite number of values, an infinite number of bits is required to represent X.

Example: Uniform Distribution

$$f_X(x) = \begin{cases} \frac{1}{b-a}, & a \le x \le b \\ 0\ , & \text{otherwise} \end{cases}$$

$$H(X) = \int_a^b \frac{1}{b-a}\log_2(b-a)\,dx = \log_2(b-a)$$

Note: For $b-a<1$, $H(X)<0$.
Differential entropy can be negative.

Example: Gaussian Distribution, $X \sim N(\mu_X, \sigma_X^2)$

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma_X^2}}\exp\left\{-\frac{(x-\mu_X)^2}{2\sigma_X^2}\right\}$$

$$\begin{aligned} H(X) &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma_X^2}}\exp\left\{-\frac{(x-\mu_X)^2}{2\sigma_X^2}\right\}\log_2\left(\sqrt{2\pi\sigma_X^2}\exp\left\{\frac{(x-\mu_X)^2}{2\sigma_X^2}\right\}\right)dx \\ &= \frac{1}{\sqrt{\pi}}\int_{-\infty}^{\infty} e^{-u^2}\log_2\left(\sqrt{2\pi\sigma_X^2}e^{u^2}\right)du \qquad \left(\text{with } u = \frac{x-\mu_X}{\sqrt{2\sigma_X^2}}\right) \\ &= \frac{1}{\sqrt{\pi}}\int_{-\infty}^{\infty} e^{-u^2}\left(\log_2\sqrt{2\pi\sigma_X^2} + u^2\log_2 e\right)du \\ &= \frac{1}{\sqrt{\pi}}\int_{-\infty}^{\infty} e^{-u^2}\,du\,\tfrac{1}{2}\log_2(2\pi\sigma_X^2) + \frac{1}{\sqrt{\pi}}\int_{-\infty}^{\infty} u^2 e^{-u^2}\,du\,\log_2 e \\ &= \tfrac{1}{2}\log_2(2\pi\sigma_X^2) + \tfrac{1}{2}\log_2 e \\ &= \tfrac{1}{2}\log_2(2\pi e\sigma_X^2) \end{aligned}$$

Theorem: Gaussian random variables have a larger differential entropy than all other random variables with the same mean and variance.

Proof: Consider two random variables, U and V, where $U \sim N(\mu_U, \sigma_U^2)$, and V has the same mean and variance as U, but not necessarily the same distribution. From the fundamental inequality

$$H(V) = \int_{-\infty}^{\infty} f_V(v)\log_2\frac{1}{f_V(v)}\,dv \le \int_{-\infty}^{\infty} f_V(v)\log_2\frac{1}{f_U(v)}\,dv$$

with equality iff $f_U(v) = f_V(v)$ (i.e., iff U and V are identically distributed).
Since U is Gaussian,

$$f_U(u) = \frac{1}{\sqrt{2\pi\sigma_U^2}}\exp\left\{-\frac{1}{2\sigma_U^2}(u-\mu_U)^2\right\}$$

so

$$
\begin{aligned}
H(V) &\le \int_{-\infty}^{\infty} f_V(v)\left(\log_2\sqrt{2\pi\sigma_U^2}+\frac{1}{2\sigma_U^2}(v-\mu_U)^2\log_2 e\right)dv \\
&= \int_{-\infty}^{\infty} f_V(v)\,dv\,\tfrac{1}{2}\log_2(2\pi\sigma_U^2)+\frac{1}{2\sigma_U^2}\int_{-\infty}^{\infty}(v-\mu_U)^2 f_V(v)\,dv\,\log_2 e
\end{aligned}
$$

Since $\mu_U=\mu_V$ and $\sigma_U^2=\sigma_V^2$,

$$
\begin{aligned}
H(V) &\le \tfrac{1}{2}\log_2(2\pi\sigma_U^2)+\frac{1}{2\sigma_V^2}\int_{-\infty}^{\infty}(v-\mu_V)^2 f_V(v)\,dv\log_2 e \\
&= \tfrac{1}{2}\log_2(2\pi\sigma_U^2)+\frac{1}{2\sigma_V^2}\mathbf{E}\left[(V-\mu_V)^2\right]\log_2 e \\
&= \tfrac{1}{2}\log_2(2\pi\sigma_U^2)+\frac{1}{2\sigma_V^2}\sigma_V^2\log_2 e \\
&= \tfrac{1}{2}\log_2(2\pi\sigma_U^2)+\tfrac{1}{2}\log_2 e \\
&= \tfrac{1}{2}\log_2(2\pi e\sigma_U^2) \\
&= H(U)
\end{aligned}
$$

Therefore the differential entropy of V is always less than or equal to the differential entropy of a Gaussian random variable with the same mean and variance, with equality if and only if V is also Gaussian.

Conditional Differential Entropy:

The conditional differential entropy is

$$H(X|Y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{X,Y}(x,y)\log_2\frac{1}{f_{X|Y}(x|y)}\,dx\,dy$$

which has the same form as differential entropy.

Average Mutual Information:

The average mutual information is

$$I(X;Y)=H(X)-H(X|Y)$$

or

$$I(X;Y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{X,Y}(x,y)\log_2\frac{f_{X|Y}(x|y)}{f_X(x)}\,dx\,dy$$

This DOES represent the amount of information gained about X by observing Y.

Channel Capacity

Consider an additive Gaussian channel, with

$$Y=X+W$$

where W is a Gaussian random variable, with zero mean and variance σ_W^2. The channel capacity is

$$C=\max_{\{f_X(x)\}} I(X;Y)$$

where

$$I(X;Y)=H(X)-H(X|Y)=H(Y)-H(Y|X)$$

Theorem: $H(Y|X)=H(W)$

Proof:

$$
\begin{aligned}
H(Y|X) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{Y|X}(y|x)f_X(x)\log_2\frac{1}{f_{Y|X}(y|x)}\,dy\,dx \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_W(y-x)f_X(x)\log_2\frac{1}{f_W(y-x)}\,dy\,dx \qquad \big(\text{since } f_{Y|X}(y|x)=f_W(y-x)\big) \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_W(w)\log_2\frac{1}{f_W(w)}\,dw f_X(x)\,dx \qquad (\text{let } w=y-x) \\
&= \int_{-\infty}^{\infty} H(W)f_X(x)\,dx \\
&= H(W)
\end{aligned}
$$

Therefore

$$I(X;Y) = H(Y) - H(W)$$

To find the channel capacity we must find $f_X(x)$ which maximizes $H(Y)$. To produce meaningful results, we restrict our attention to the case where X has a fixed variance, σ_X^2.

Note: $H(Y)$ is maximized if Y has a Gaussian distribution.

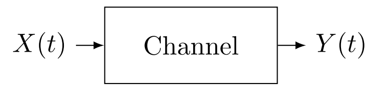
If Y must be Gaussian, then X must also be Gaussian, since W is Gaussian and $X = Y - W$.
If σ_X^2 is the variance of X, then the variance of Y is $\sigma_Y^2 = \sigma_X^2 + \sigma_W^2$.

Note: $H(Y) = \frac{1}{2}\log_2(2\pi e \sigma_Y^2)$ and $H(W) = \frac{1}{2}\log_2(2\pi e \sigma_W^2)$

For fixed σ_X^2, the channel capacity is

$$\begin{aligned} C &= \tfrac{1}{2}\log_2(2\pi e \sigma_Y^2) - \tfrac{1}{2}\log_2(2\pi e \sigma_W^2) \\ &= \tfrac{1}{2}\log_2(\frac{\sigma_Y^2}{\sigma_W^2}) \\ &= \tfrac{1}{2}\log_2(1 + \frac{\sigma_X^2}{\sigma_W^2}) \qquad \text{(bits per channel use)} \end{aligned}$$

Continuous-Time Waveform Channels



Suppose a random process is transmitted over a band-limited AWGN channel. The input, $X(t)$, is band-limited to W Hz, and the output, $Y(t)$, is related to the input through

$$Y(t) = X(t) + W(t)$$

where $W(t)$ is the result of passing an AWGN process through a lowpass filter with bandwidth W.
Since $Y(t)$ is band-limited, it can be sampled at the Nyquist rate. The samples are

$$Y_n = Y(nT) = X(nT) + W(nT) = X_n + W_n$$

where $T = \frac{1}{2W}$. The noise samples are independent Gaussian random variables with zero mean and variance

$$\sigma_W^2 = \mathcal{N}_0 W$$

For a block of N received samples, $\underline{Y}_N = [Y_1\ Y_2\ \ldots\ Y_N]$ for the transmitted samples $\underline{X}_N = [X_1\ X_2\ \ldots\ X_N]$, the channel capacity is

$$C = \lim_{N\to\infty} \frac{1}{N} \max_{\{f_{\underline{X}_N}(\underline{x}_N)\}} I(\underline{X}_N; \underline{Y}_N)$$

<u>Fact:</u> $I(\underline{X}_N;\underline{Y}_N) = H(\underline{Y}_N) - H(\underline{Y}_N|\underline{X}_N) = H(\underline{Y}_N) - H(\underline{W}_N)$
where $H(\underline{W}_N) = N\frac{1}{2}\log_2(2\pi e\sigma_W^2)$

<u>Theorem:</u> The differential entropy of a vector of random variables is largest only if the component random variables are independent.

<u>Proof:</u> From the fundamental inequality

$$H(\underline{Y}_N) = \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} f_{\underline{Y}_N}(\underline{y}_N)\log_2\frac{1}{f_{\underline{Y}_N}(\underline{y}_N)}\,d\underline{y}_N \le \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} f_{\underline{Y}_N}(\underline{y}_N)\log_2\frac{1}{\prod_{n=1}^{N} f_{Y_n}(y_n)}\,d\underline{y}_N$$

with equality iff $\prod_{n=1}^{N} f_{Y_n}(y_n) = f_{\underline{Y}_N}(\underline{y}_N)$ (i.e., iff the components of $\underline{Y}_N$ are independent). Therefore

$$\begin{aligned}
H(\underline{Y}_N) &\le \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} f_{\underline{Y}_N}(\underline{y}_N)\log_2\frac{1}{\prod_{n=1}^{N} f_{Y_n}(y_n)}\,d\underline{y}_N \\
&= \sum_{n=1}^{N}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} f_{\underline{Y}_N}(\underline{y}_N)\log_2\frac{1}{f_{Y_n}(y_n)}\,d\underline{y}_N \\
&= \sum_{n=1}^{N}\int_{-\infty}^{\infty} f_{Y_n}(y_n)\log_2\frac{1}{f_{Y_n}(y_n)}\,dy_n \\
&= \sum_{n=1}^{N} H(Y_n)
\end{aligned}$$

Therefore, the received samples $\underline{Y}_N$ must be independent. Since the noise samples are independent this implies that the components of $\underline{X}_N$ must also be independent. In this case the average mutual information is

$$I(\underline{X}_N;\underline{Y}_N) = \sum_{n=1}^{N} H(Y_n) - \sum_{n=1}^{N} H(W_n) = \sum_{n=1}^{N}[H(Y_n) - H(W_n)] = \sum_{n=1}^{N} I(X_n;Y_n)$$

As was shown earlier, $I(X_n;Y_n)$ is maximized when X_n has a Gaussian distribution, in which case

$$I(X_n;Y_n) = \tfrac{1}{2}\log_2\left(1+\frac{\sigma_X^2}{\sigma_W^2}\right)$$

The channel capacity is therefore

$$\begin{aligned}
C &= \lim_{N\to\infty}\frac{1}{N}\sum_{n=1}^{N}\tfrac{1}{2}\log_2\left(1+\frac{\sigma_X^2}{\sigma_W^2}\right) \\
&= \lim_{N\to\infty}\tfrac{1}{2}\log_2\left(1+\frac{\sigma_X^2}{\sigma_W^2}\right) \\
&= \tfrac{1}{2}\log_2\left(1+\frac{\sigma_X^2}{\sigma_W^2}\right)
\end{aligned}$$

for fixed σ_X^2. Since the channel capacity does not depend on the mean of $X(t)$, we can restrict our attention to zero-mean input random processes with fixed average power,

$$P_{\text{av}} = \mathbf{E}\left[X^2(t)\right] = \sigma_X^2$$

$$C = \tfrac{1}{2}\log_2\left(1+\frac{P_{\text{av}}}{\mathcal{N}_0 W}\right) \qquad \text{(bits per sample)}$$

Since one sample is transmitted every T seconds, the channel capacity per unit time is

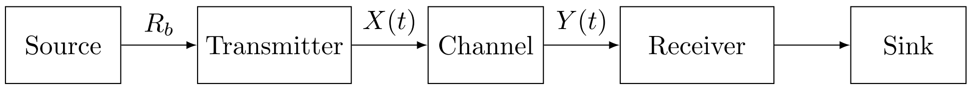
$$C = \frac{1}{2T}\log_2\left(1+\frac{P_{\text{av}}}{\mathcal{N}_0 W}\right) \qquad \text{(bits per second)}$$

Since $T = 1/(2W)$, the information capacity of an AWGN channel bandlimited to W Hz is

$$C = W\log_2\left(1+\frac{P_{\text{av}}}{\mathcal{N}_0 W}\right) \qquad \text{(bits per second)}$$

where P_{av} is the average transmitted signal power and $\mathcal{N}_0$ is the single-sided noise power spectral density.

To illustrate the importance of the channel capacity and the channel coding theorem, consider a source that produces information at a rate of R_b bits per second.



Provided that the information rate, R_b, is less than the channel capacity, C, then there exists a coding/modulation scheme which will provide arbitrarily low probability of error. If $R_b > C$, no such scheme exists. In this case either the source information rate must be slowed, or the channel capacity must be increased, either by increasing the bandwidth or increasing the average transmitted power.

It is often more useful to consider bandwidth efficiency instead of the information rate. If information is transmitted at a rate R_b over a channel with bandwidth W, the bandwidth efficiency is R_b/W bits per second per Hz.

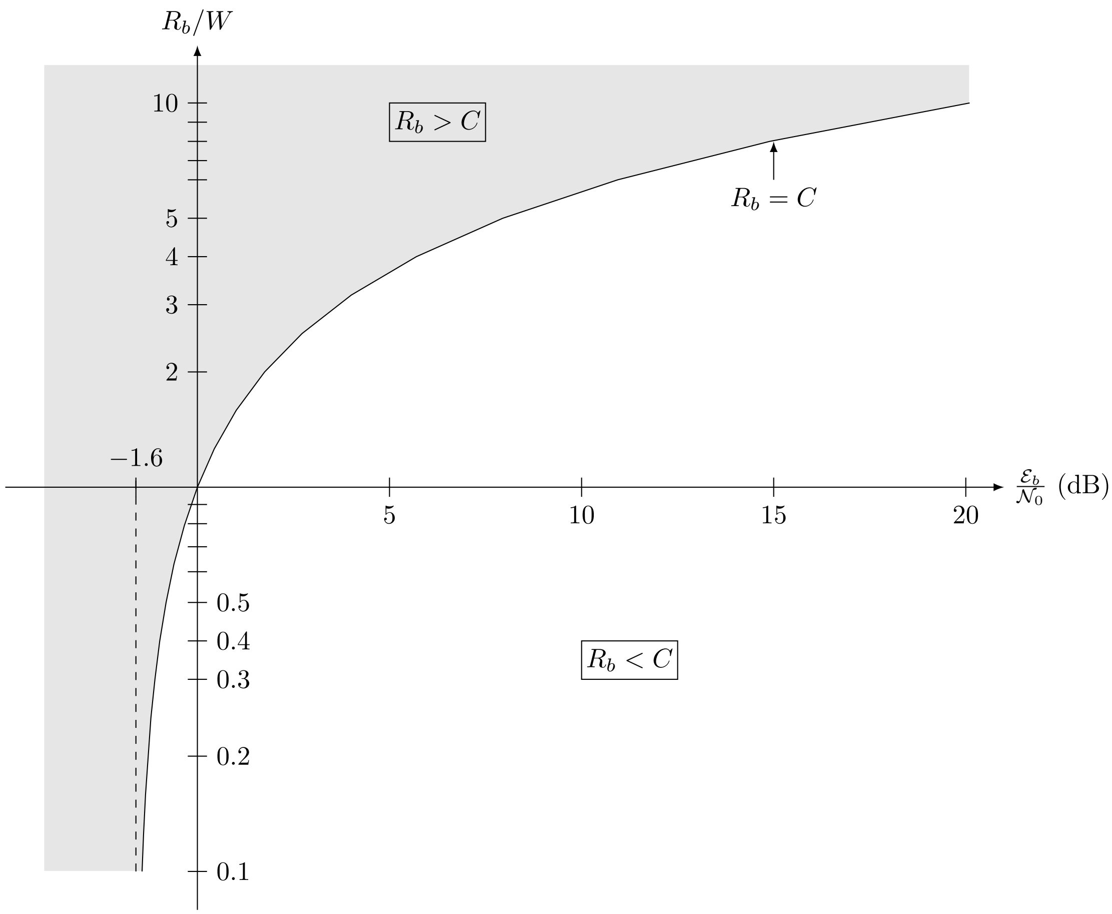
If the channel is operating at capacity it can convey C bits per second with an average power of P_{av}, so the energy per information bit is

$$\mathcal{E}_b = \frac{P_{\text{av}}}{C}\ .$$

As a result,

$$\frac{C}{W} = \log_2\left(1+\frac{C}{W}\frac{\mathcal{E}_b}{\mathcal{N}_0}\right) \qquad \text{bits per second per Hz}$$

This can be plotted in a *bandwidth efficiency diagram*.



Note that at capacity,

$$\frac{\mathcal{E}_b}{\mathcal{N}_0} = \frac{2^{C/W} - 1}{C/W}$$

As $W \to \infty$,

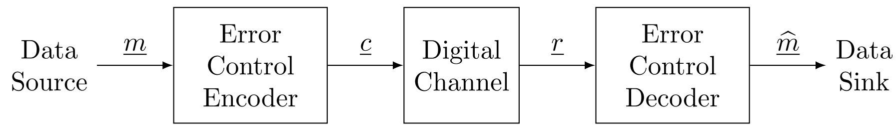
$$\frac{\mathcal{E}_b}{\mathcal{N}_0} \to \ln 2 = 0.693 = -1.6 \text{ dB}$$

which is the Shannon limit.

Remark: Although the channel coding theorem guarantees us that a suitable coding/modulation scheme exists to ensure reliable communication at information rates less than the channel capacity, it does not tell us how to do it.

Error Control Techniques

To improve the reliability of digital communication, error control techniques are typically employed. Generally, these involve inserting carefully controlled redundancy into the transmitted data, and exploiting this redundancy at the receiver to detect and/or correct transmission errors.



Notation:

$\underline{m} \Rightarrow$ message word (message block)

$\underline{m} = [m_0\ m_1\ \ldots\ m_{k-1}]$

$k \Rightarrow$ # of message bits per message word

$\underline{c} \Rightarrow$ code word

$\underline{c} = [c_0\ c_1\ \ldots\ c_{n-1}]$

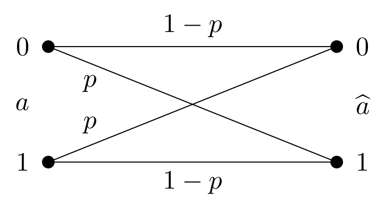
$n \Rightarrow$ # of code bits per code word

$R \Rightarrow$ code rate

$R = \frac{k}{n}$

Examples:

Consider the binary symmetric channel (BSC) with cross-over probability p.



For the AWGN channel with antipodal signalling,

$$p = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{\mathcal{E}_b}{\mathcal{N}_0}}\right)$$

No error control

Each message bit is sent without any encoding, so $k = 1$, $n = 1$, and $\underline{c} = \underline{m}$. The probability of a message bit error is

$$P_\varepsilon = p$$

Double-repitition code

Send each bit twice, and if the received bits differ, an error is detected. In this case, $k = 1$, $n = 2$, and the code is defined by

$\underline{m}$		$\underline{c}$
0	$\rightarrow$	00
1	$\rightarrow$	11

Suppose the message bit is $\underline{m} = [0]$, so the transmitted codeword is $\underline{c} = [00]$. The received codeword is

$$\underline{r} = \begin{cases} 00 \text{ with probability } (1-p)^2 \\ 01 \text{ with probability } (1-p)p \\ 10 \text{ with probability } p(1-p) \\ 11 \text{ with probability } p^2 \end{cases}$$

The probability that an error is detected is

$$P_d = 2p(1-p)$$

In this event, the receiver requests the transmitter to resend the codeword. This type of error control technique is known as *automatic repeat request (ARQ)*.

The probability of an undetected error is

$$P_\varepsilon = p^2$$

Triple-repitition code

Send each message bit three times, so $k = 1$, $n = 3$, and

$$\begin{array}{ccc} \underline{m} & & \underline{c} \\ 0 & \rightarrow & 000 \\ 1 & \rightarrow & 111 \end{array}$$

If all the received bits are not the same, the receiver can detect that an error occured. The probability of detecting an error is

$$\begin{aligned} P_d &= \Pr\{\text{one or two code bits are in error}\} \\ &= 3p(1-p)^2 + 3p^2(1-p) \end{aligned}$$

The probability of an undetected error is

$$\begin{aligned} P_\varepsilon &= \Pr\{\text{all three code bits are in error}\} \\ &= p^3 \end{aligned}$$

This code has a lower probability of undetected errors than the double-repitition code.

This code could also be used for *forward error correction (FEC)*, where instead of merely detecting the occurance of errors, the receiver tries to correct them. For the triple-repitition code the receiver would make a *majority rule* decision:

> If more zeros are received than ones, assume the message bit was a zero. Otherwise the receiver assumes the message bit was a one.

The possibility exists that the receiver could make a mistake by following this decision rule if errors occur in too many code bits. The probability of a message bit error is:

$$\begin{aligned} P_\varepsilon &= \Pr\{\text{2 or 3 code bits are in error}\} \\ &= 3p^2(1-p) + p^3 \\ &= 3p^2 - 2p^3 \end{aligned}$$

Warning

The discussion above is somewhat flawed because it neglects some important constraints on coded systems.

Because of the increased number of bits that need to be transmitted, a higher channel bandwidth is required.

In addition, to provide a fair comparison between different coding schemes, the comparison must be made for equal transmitted energy per *message* bit, $\mathcal{E}_{\text{mb}}$. The transmitted energy per code bit is then $\mathcal{E}_{\text{cb}} = R\mathcal{E}_{\text{mb}}$. Since the cross-over probability depends on the transmitted energy per code bit, different codes will operate with different cross-over probabilities. It is important to verify that the increase in the cross-over probability due to the decrease in the transmitted energy does not negate the improvements in performance due to coding.

Linear Block Codes

Def^n: An (n,k) binary code, $\mathcal{C}$, consists of a set of 2^k binary code words, each of length n bits, and a mapping function between message words and code words.

eg. Binary $(5,2)$ code with rate $R = \frac{2}{5}$.
Code $\mathcal{C} = \{01100, 10101, 10111, 11000\}$

$\underline{m}$	$\underline{c}$
00	01100
01	10101
10	10111
11	11000

Def^n: The *rate* of the code is defined as $R = \frac{k}{n}$.

Objective: Pick the code words so they are as far apart as possible. In general, this is a difficult task. Code $\mathcal{C}$ above is a bad code because the code words for 01 and 10 differ by only one bit.

Def^n: An (n,k) *linear block code* is defined by a generator matrix, $\underline{\underline{G}}$, such that the code word $\underline{c}$ for message $\underline{m}$ is obtained from

$$\underline{c} = \underline{m}\,\underline{\underline{G}}$$

or

$$[c_1\ c_2\ \cdots\ c_n] = [m_1\ m_2\ \cdots\ m_k] \begin{bmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,n} \\ g_{2,1} & g_{2,2} & \cdots & g_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k,1} & g_{k,2} & \cdots & g_{k,n} \end{bmatrix}$$

where modulo-2 arithmetic is used.

Note: Each codeword of a linear code is just some linear combination of the rows of $\underline{\underline{G}}$.

Note: The rows of $\underline{\underline{G}}$ must be linearly independent. Otherwise, two or more messages may map to the same code word.

Example: n-repititon code – an (n,1) code.

$$\mathcal{C} = \{00\cdots0, 11\cdots1\}$$
$$\underline{\underline{G}} = [11\cdots1] \qquad R = \frac{1}{n}$$

Example: Single parity check code (even parity) – a (k+1,k) code.

$$c_{k+1} = m_1 \oplus m_2 \oplus \cdots \oplus m_k$$

$$\underline{\underline{G}} = \left[\begin{array}{cccc|c} 1 & 0 & \cdots & 0 & 1 \\ 0 & 1 & \cdots & 0 & 1 \\ \vdots & \vdots & \ddots & \vdots & 1 \\ 0 & 0 & \cdots & 1 & 1 \end{array}\right] \qquad R = \frac{k}{k+1}$$

Def^n: The *Hamming weight* of a word is the number of 1's in the word.

eg. $w_H(110110) = 4$

Def^n: The *minimum weight* of a code, $\mathcal{C}$, is the Hamming weight of the smallest weight non-zero codeword of $\mathcal{C}$.

ie. $w_{\min} = \min\{w_H(\underline{c}) \mid \underline{c} \in \mathcal{C}, \underline{c} \neq \underline{0}\}$

eg. For code $\mathcal{C} = \{00000, 01011, 10101, 11110\}$

$$w_{\min} = w_H(01011) = 3$$

Def^n: The *Hamming distance* between two words, $\underline{a}$ and $\underline{b}$, is the number of positions in which $\underline{a}$ and $\underline{b}$ differ.

eg. $d_H(01011, 11110) = 3$

Fact: The Hamming distance between words $\underline{a}$ and $\underline{b}$ is equal to the Hamming weight of $\underline{a} \oplus \underline{b}$.

ie. $d_H(\underline{a}, \underline{b}) = w_H(\underline{a} \oplus \underline{b})$

Def^n: The *minimum distance* of a code, $\mathcal{C}$, is the minimum Hamming distance between any two different codewords in $\mathcal{C}$.

ie. $d_{\min} = \min\{d_H(\underline{a}, \underline{b}) \mid \underline{a}, \underline{b} \in \mathcal{C}, \underline{a} \neq \underline{b}\}$

eg. For code $\mathcal{C} = \{00000, 01011, 10101, 11110\}$

$$d_{\min} = 3$$

Theorem: For all linear block codes, $d_{\min} = w_{\min}$.

Proof: Suppose the minimum distance for code $\mathcal{C}$ is between code words $\underline{c}_1 = \underline{m}_1\underline{\underline{G}}$ and $\underline{c}_2 = \underline{m}_2\underline{\underline{G}}$. The minimum distance is

$$d_H(\underline{c}_1, \underline{c}_2) = w_H(\underline{c}_1 \oplus \underline{c}_2) = w_H(\underline{m}_1\underline{\underline{G}} \oplus \underline{m}_2\underline{\underline{G}}) = w_H([\underline{m}_1 \oplus \underline{m}_2]\underline{\underline{G}})$$

Since $\underline{m}_1 \oplus \underline{m}_2$ is a valid message word, $[\underline{m}_1 \oplus \underline{m}_2]\underline{\underline{G}}$ is a valid code word in $\mathcal{C}$.

Def^n: A *systematic code* is one in which the first k bits in each codeword are equal to the k message bits. i.e.

$$\underline{c} = \boxed{k \text{ message bits}}\boxed{(n-k) \text{ parity bits}}$$

For a systematic code,

$$\underline{\underline{G}} = \left[\begin{array}{c|cccc} & p_{1,1} & p_{1,2} & \cdots & p_{1,n-k} \\ & p_{2,1} & p_{2,2} & \cdots & p_{2,n-k} \\ \underline{\underline{I}}_k & \vdots & \vdots & \ddots & \vdots \\ & p_{k,1} & p_{k,2} & \cdots & p_{k,n-k} \end{array}\right] = \left[\,\underline{\underline{I}}_k \mid \underline{\underline{P}}\,\right]$$

where $\underline{\underline{I}}_k$ is the $k \times k$ identity matrix.

Def^n: Associated with every linear (n, k) code is a $(n-k) \times n$ *parity check matrix*, $\underline{\underline{H}}$, with the property

$$\underline{\underline{G}}\,\underline{\underline{H}}^T = \underline{\underline{0}}$$

For systematic codes,

$$\underline{\underline{H}} = \left[\underline{\underline{P}}^T \mid \underline{\underline{I}}_{n-k}\right]$$

Def^n: The *syndrome* of a received word, $\underline{r}$, is $\underline{s} = \underline{r}\,\underline{\underline{H}}^T$. The syndrome is of length $n-k$ bits.

Theorem: If $\underline{c}$ is a codeword in $\mathcal{C}$, then $\underline{c}\,\underline{\underline{H}}^T = \underline{0}$.

Proof: Since $\underline{c} = \underline{m}\,\underline{\underline{G}}$ for some message word $\underline{m}$ if $\underline{c} \in \mathcal{C}$

$$\underline{c}\,\underline{\underline{H}}^T = \underline{m}\,\underline{\underline{G}}\,\underline{\underline{H}}^T = \underline{m}\,\underline{\underline{0}} = \underline{0}\,.$$

Fact: The minimum distance of a code is equal to the minimum number of columns in $\underline{\underline{H}}$ that add to the zero vector.

Note: If codeword $\underline{c}$ is transmitted across a binary symmetric channel, the received word is

$$\underline{r} = \underline{c} \oplus \underline{e}$$

where $\underline{e}$ is the error pattern.

eg.

$$\begin{aligned} \underline{c} &= [10101] \\ \underline{e} &= [11000] \\ \underline{r} &= [01101] \end{aligned}$$

Fact: If the crossover probability of a BSC is p, then the probability of error pattern $\underline{e}$ occurring is

$$\Pr\{\underline{e}\} = p^{n_e}(1-p)^{(n-n_e)}$$

where n is the length of $\underline{e}$, and n_e is the Hamming weight of $\underline{e}$ (i.e., the number of code bit errors).

Example: A (6,3) systematic code.

$$\underline{\underline{G}} = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right] \qquad \begin{aligned} k &= 3 \\ n &= 6 \end{aligned} \qquad R = \tfrac{3}{6}$$

$\underline{m}$	$\underline{c}$
000	000000
001	001101
010	010011
011	011110
100	100110
101	101011
110	110101
111	111000

$$d_{\min} = 3$$

The parity check matrix is:

$$\underline{\underline{H}} = \left[\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

Suppose $\underline{r} = [110101]$ is received. The syndrome is

$$\underline{s} = \underline{r}\,\underline{\underline{H}}^T = [110101]\left[\begin{array}{ccc} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ \hline 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right] = [000]$$

Therefore, $\underline{r}$ is a valid codeword (no errors are detected).
Now, suppose $\underline{r} = [100101]$ is received.

$$\underline{s} = \underline{r}\,\underline{\underline{H}}^T = [011]$$

Therefore, $\underline{r}$ is not a valid codeword, so an error is detected.

Note: If an error is detected (i.e., the syndrome is non-zero), the decoder can either take actions to locate the errors and correct them (FEC), or request the retransmission of the codeword (ARQ).

Error Detection and Automatic Repeat Request (ARQ)

Error Detection:

- First compute the syndrome, $\underline{s}$ of the received word, $\underline{r}$ as

$$\begin{aligned}\underline{s} &= \underline{r}\,\underline{\underline{\mathrm{H}}}^T = (\underline{c} \oplus \underline{e})\,\underline{\underline{\mathrm{H}}}^T \\ &= \underline{c}\,\underline{\underline{\mathrm{H}}}^T \oplus \underline{e}\,\underline{\underline{\mathrm{H}}}^T \\ &= \underline{e}\,\underline{\underline{\mathrm{H}}}^T\end{aligned}$$

- If $\underline{s} = \underline{0}$ then $\underline{r}$ is assumed error-free and is accepted.
- If $\underline{s} \neq \underline{0}$ then an error is detected.

Note: If the error pattern, $\underline{e}$, matches a code word then an error will not be detected. If the error pattern does not match a code word then an error will be detected.
A linear block code can detect all error patterns that are not valid code words.

Theorem: A code with minimum distance $d_{\min}$ can detect all combinations of $d_{\min} - 1$ or fewer errors.

Proof: Any error pattern, $\underline{e}$, of weight $d_{\min} - 1$ or less is not a codeword and therefore yields a non-zero syndrome.

Def^n: The *random-error detecting capability* of a linear block code is $d_{\min} - 1$.

Example:

Suppose we would like to transmit message words of $k = 3$ bits, with a signal-to-noise ratio of $\mathcal{E}_{\mathrm{mb}}/\mathcal{N}_{0\|dB} = 6$ dB, or $\mathcal{E}_{\mathrm{mb}}/\mathcal{N}_0 \cong 3.98$.

Scheme 1: No coding, just send the 3 bits directly over the channel. The probability of a bit error is

$$p = \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{\mathcal{E}_{\mathrm{mb}}}{\mathcal{N}_0}}\right) = \frac{1}{2}\mathrm{erfc}\left(\sqrt{10^{0.6}}\right) \cong 0.00239$$

The probability of a message being correctly transmitted is $P_C = (1-p)^3 \cong 0.993$.
The probability of a message error is $P_\varepsilon = 1 - P_C \cong 0.00715$.

Scheme 2: Use the $(6,3)$ systematic code given by

$$\underline{\underline{\mathrm{G}}} = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right]$$

for error detection. Since $R = \frac{3}{6}$ the probability of a code bit error is

$$p = \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{\mathcal{E}_{\mathrm{cb}}}{\mathcal{N}_0}}\right) = \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{R\mathcal{E}_{\mathrm{mb}}}{\mathcal{N}_0}}\right) = \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{3}{6} \times 10^{0.6}}\right) \cong 0.0230$$

Event 1: No errors. The syndrome is zero and the received word is the transmitted codeword. The probability of the message being transmitted correctly is $P_C = (1-p)^6 = 0.870$

Event 2: Undetected errors. The syndrome is zero but the received word is not the transmitted codeword. This occurs if the error pattern is the same as a valid non-zero codeword. The undetected error probability is

$$\begin{aligned}P_\varepsilon &= \Pr\{\underline{e} \in \mathcal{C}\} \\ &= \Pr\{\underline{e} = 001101 \text{ or } \underline{e} = 010011 \text{ or } \underline{e} = 011110 \text{ or } \underline{e} = 100110 \\ &\qquad\quad \text{or } \underline{e} = 101011 \text{ or } \underline{e} = 110101 \text{ or } \underline{e} = 111000\} \\ &= 4p^3(1-p)^3 + 3p^4(1-p)^2 \\ &\cong 0.00004623\end{aligned}$$

Event 3: Detected errors. The syndrome is non-zero. The erasure or rejection probability (rate) is

$$\begin{aligned}P_d &= 1 - P_C - P_\varepsilon \\ &\cong 0.130\end{aligned}$$

Summary: Although the probability of receiving the message correctly is slightly lower in Scheme 2, the probability of undetected error is considerably lower.

Automatic Repeat Request (ARQ) Protocols

– Upon receipt of a block, the receiver detects whether or not a block error has occurred.

– If an error IS NOT detected, the receiver sends a positive acknowledgement (ACK) back to the transmitter.

– If an error IS detected, the receiver sends a negative acknowledgement (NACK) back to the transmitter.

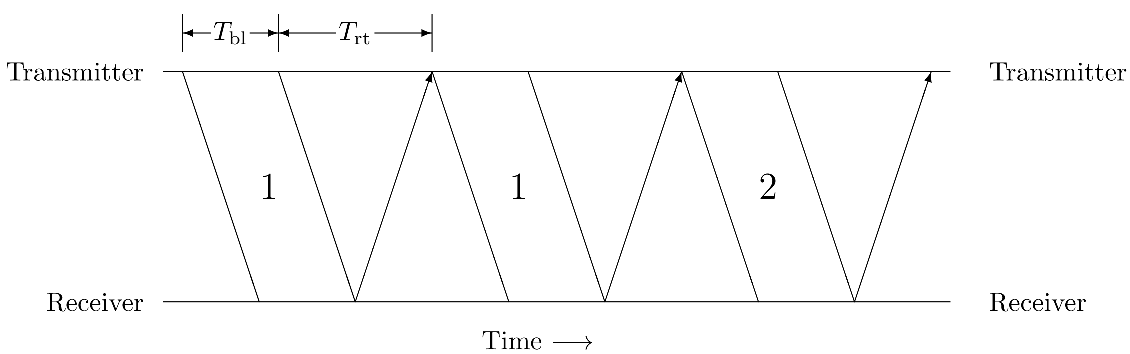
– The transmitter re-sends blocks that are not received correctly according to a repeat protocol:

1. stop-and-wait ARQ
2. continuous ARQ (go-back-N)
3. selective repeat ARQ

– Important parameters of ARQ schemes:
  - P_d = the probability of a detected block error.
  - T_{bl} = the block transmission time
    $T_{\text{bl}} = n/R_c = k/R_m$
  - T_{rt} = the round-trip transmission delay. This is twice the time it takes for a signal to propagate from the transmitter to the receiver.
  - $\overline{N}$ = the average number of times a block must be transmitted before it is correctly received.
  - $\overline{T}$ = the average total time it takes for a block to be successfully transmitted.
  - η = the effective transmission rate (message bits per second).

Stop-and-wait ARQ

– Transmit one block and wait for acknowledgment.
  - If an ACK is received, transmit a new block.
  - If a NACK is received, retransmit the same block.



– Average number of transmitted blocks per correctly received block:

$$\begin{aligned}\overline{N} &= 1(1-P_d) + 2(1-P_d)P_d + 3(1-P_d)P_d^2 + \cdots + m(1-P_d)P_d^{m-1} + \cdots \\ &= \sum_{m=0}^{\infty} m(1-P_d)P_d^{m-1} = (1-P_d)\sum_{m=0}^{\infty}\frac{d}{dP_d}\left(P_d^m\right) = (1-P_d)\frac{d}{dP_d}\left(\sum_{m=0}^{\infty}P_d^m\right) \\ &= (1-P_d)\frac{d}{dP_d}\left(\frac{1}{1-P_d}\right) = (1-P_d)\frac{1}{(1-P_d)^2} \qquad \left(\text{with } \sum_{n=0}^{\infty}\alpha^n = \frac{1}{1-\alpha} \text{ for } \alpha < 1\right) \\ &= \frac{1}{1-P_d}\end{aligned}$$

– Average time to successfully transmit one block:

$$\overline{T} = (T_{\text{bl}} + T_{\text{rt}})\overline{N}$$

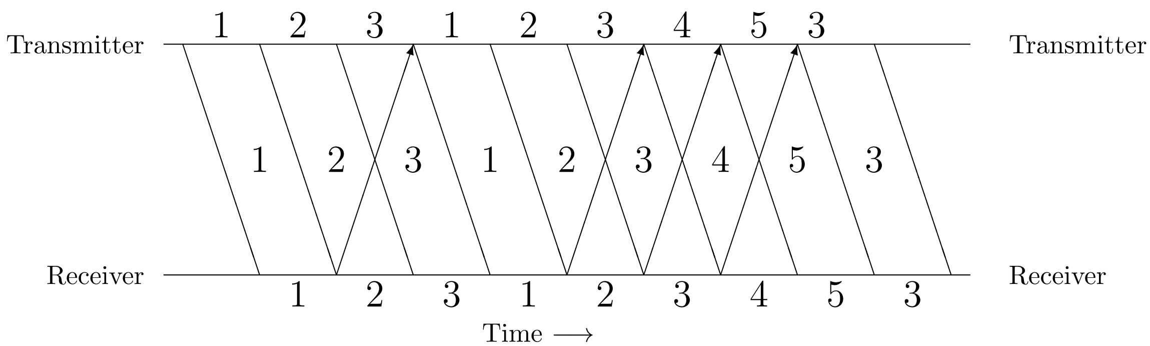
– Effective transmission rate

$$\eta = \frac{k}{\overline{T}} = \frac{k}{(T_{\text{bl}} + T_{\text{rt}})\overline{N}} = R_m\frac{k}{(k + T_{\text{rt}}R_m)\overline{N}} = R_m\frac{k}{(k + T_{\text{rt}}R_m)}(1-P_d)$$

$$= R_m(1 - P_d)\frac{1}{1 + \frac{T_{\rm rt}}{T_{\rm bl}}}$$

Continuous ARQ (Go-back-N)

- Blocks are transmitted continuously
- As soon as the transmitter has completed sending one block, it begins sending the next.
- The acknowledgment for a block arrives after a round-trip delay.
- Assume that $(N-1)$ blocks can be transmitted during a round-trip delay, so $N - 1 = \left\lceil \frac{T_{\rm rt}}{T_{\rm bl}} \right\rceil$.
- When a NACK is received, the transmitter backs up to the block that was rejected, and retransmits that block followed by the $(N-1)$ blocks previously transmitted during the round-trip delay.
- The receiver discards the $(N-1)$ blocks received after an incorrectly received one, so there is no need for buffering at the receiver.



- Average number of transmitted blocks per correctly received block:

$$\begin{aligned}\overline{N} &= 1(1-P_d) + (N+1)(1-P_d)P_d + (2N+1)(1-P_d)P_d^2 + \cdots + (mN+1)(1-P_d)P_d^m + \cdots \\ &= \sum_{m=0}^{\infty}(mN+1)(1-P_d)P_d^m \\ &= \frac{1+(N-1)P_d}{1-P_d}\end{aligned}$$

- Average time to successfully transmit one block:

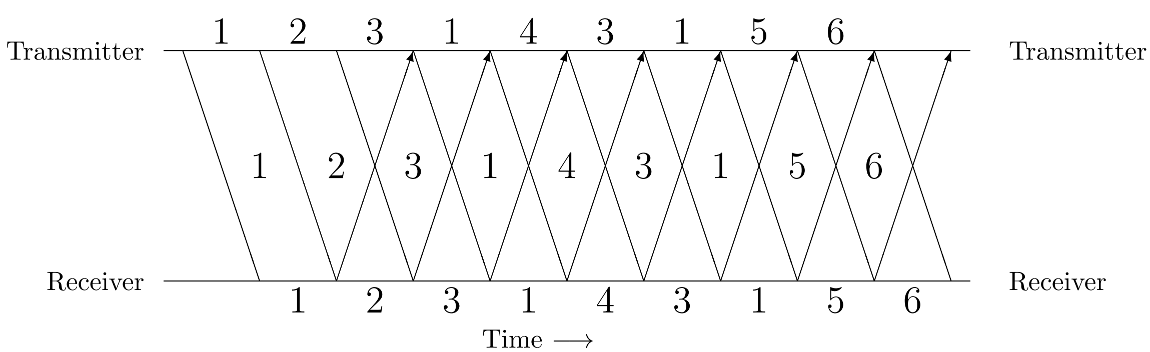
$$\overline{T} = \frac{n\overline{N}}{R_c} = T_{\rm bl}\overline{N}$$

- Effective transmission rate

$$\eta = \frac{k}{\overline{T}} = \frac{kR_c}{n\overline{N}} = R_m\frac{1}{\overline{N}} = R_m\frac{1-P_d}{1+(N-1)P_d}$$

Selective Repeat ARQ

- Variation of continuous ARQ
- Only the block received in error is retransmitted.
- A buffer is needed at the receiver to store blocks correctly received after a rejected block.



– Average number of transmitted blocks per correctly received block:

$$\begin{aligned}\overline{N} &= 1(1-P_d) + 2(1-P_d)P_d + 3(1-P_d)P_d^2 + \cdots + m(1-P_d)P_d^{m-1} + \cdots \\ &= \frac{1}{1-P_d}\end{aligned}$$

– Average time to successfully transmit one block:

$$\overline{T} = \frac{n\overline{N}}{R_c} = T_{\mathrm{bl}}\overline{N}$$

– Effective transmission rate:

$$\begin{aligned}\eta &= \frac{k}{\overline{T}} = \frac{kR_c}{n\overline{N}} = R_m\frac{1}{\overline{N}} \\ &= R_m(1-P_d)\end{aligned}$$

Forward Error Correction (FEC)

Example: A (6,3) systematic code.

$$\underline{\underline{G}} = \left[\begin{array}{ccc|ccc} 1&0&0&1&1&0 \\ 0&1&0&0&1&1 \\ 0&0&1&1&0&1 \end{array}\right] \qquad \underline{\underline{H}} = \left[\begin{array}{ccc|ccc} 1&0&1&1&0&0 \\ 1&1&0&0&1&0 \\ 0&1&1&0&0&1 \end{array}\right]$$

$\underline{m}$	$\underline{c}$
000	000000
001	001101
010	010011
011	011110
100	100110
101	101011
110	110101
111	111000

$k = 3$
$n = 6$
$R = \frac{3}{6} = \frac{1}{2}$
$d_{\min} = 3$

Suppose we send message $\underline{m} = 110$, so the transmitted code word is $\underline{c} = \underline{m}\,\underline{\underline{G}} = 110101$.
Suppose error pattern $\underline{e} = 010000$ occurs during transmission, so the received word is $\underline{r} = 100101$.
The decoder computes the syndrome $\underline{s} = \underline{r}\,\underline{\underline{H}}^T = 011$. Since the syndrome is non-zero, the decoder knows that some errors have occurred during transmission. Now it must try to determine the transmitted message.

Objective: Determine the transmitted message from $\underline{r}$.
Since $\underline{r} = \underline{c} \oplus \underline{e}$, then $\underline{r} \oplus \underline{e} = \underline{c} \oplus \underline{e} \oplus \underline{e} = \underline{c}$. So, if the decoder can determine $\underline{e}$, then it can find $\underline{c}$. For systematic codes the first k bits of $\underline{c}$ give the message, $\underline{m}$. Therefore, the decoder only needs to find $\underline{e}$ to determine the transmitted message.

Solution: Find $\underline{e}$ from $\underline{s}$.
For systematic codes,

$$\underline{\underline{H}} = \left[\, \underline{\underline{P}}^T \mid \underline{\underline{I}}_{n-k} \,\right] = \left[\begin{array}{cccc|cccc} p_{1,1} & p_{2,1} & \cdots & p_{k,1} & 1 & 0 & \cdots & 0 \\ p_{1,2} & p_{2,2} & \cdots & p_{k,2} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{1,n-k} & p_{2,n-k} & \cdots & p_{k,n-k} & 0 & 0 & \cdots & 1 \end{array}\right]$$

Since $\underline{s} = \underline{r}\,\underline{\underline{H}}^T = \underline{e}\,\underline{\underline{H}}^T$, $\underline{s} = [s_1\ s_2\ \ldots\ s_{n-k}]$ can be expressed as:

$$\begin{array}{rcl} s_1 &=& e_1p_{1,1} \oplus e_2p_{2,1} \oplus \cdots \oplus e_kp_{k,1} \oplus e_{k+1} \\ s_2 &=& e_1p_{1,2} \oplus e_2p_{2,2} \oplus \cdots \oplus e_kp_{k,2} \oplus e_{k+2} \\ \vdots && \vdots \qquad \vdots \qquad \ddots \qquad \vdots \qquad \vdots \\ s_{n-k} &=& e_1p_{1,n-k} \oplus e_2p_{2,n-k} \oplus \cdots \oplus e_kp_{k,n-k} \oplus e_n \end{array}$$

We would like the decoder to solve these equations for $\underline{e} = [e_1\ e_2\ \ldots\ e_n]$.

Example: The decoder must solve the following for $\underline{e}$:

$$\begin{array}{l} 0 = e_1 \oplus e_3 \oplus e_4 \\ 1 = e_1 \oplus e_2 \oplus e_5 \\ 1 = e_2 \oplus e_3 \oplus e_6 \end{array}$$

Problem: There are n unknowns but only $n-k$ equations. Therefore there is not a unique solution.

Fact: For each syndrome there are exactly 2^k possible solutions. i.e., there are 2^k error possible patterns that result in each syndrome. The true error pattern is one of them.

Example: The following eight error patterns all yield the same syndrome $\underline{s} = 011$.

$$\{010000, 011101, 000011, 001110, 110110, 111011, 100101, 101000\}$$

Problem: The decoder must select one of the 2^k error patterns, but if the wrong one is selected a *correction error* will result.

Solution: Minimize the probability of a correction error.

Of the 2^k possible error patterns for a given syndrome, the decoder assumes that the most probable error pattern is the true error pattern.

For a BSC, this corresponds to choosing the error pattern with the smallest Hamming weight.

Example:

Since $\underline{\widehat{e}} = 010000$ has the smallest weight of the eight error patterns listed above, the decoder assumes this is the true error pattern. Decoding continues with $\underline{\widehat{c}} = \underline{r} \oplus \underline{\widehat{e}} = 100101 \oplus 010000 = 110101$, and $\underline{\widehat{m}}$ is just the first three bits of $\underline{\widehat{c}}$, so $\underline{\widehat{m}} = 110$. Since $\underline{\widehat{m}} = \underline{m}$, the transmission error has been corrected.

Standard Array Decoding

Def^n: The *standard array* is a $2^{n-k} \times 2^k$ matrix containing all 2^n possible received words. From the position of a received word in the standard array, the most probable transmitted code word can be determined.

Figure: Standard Array

$$\begin{array}{cccc} \underline{c}_1 = \underline{0} & \underline{c}_2 & \cdots & \underline{c}_{2^k} \\ \underline{e}_2 & \underline{e}_2 \oplus \underline{c}_2 & \cdots & \underline{e}_2 \oplus \underline{c}_{2^k} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{e}_{2^{n-k}} & \underline{e}_{2^{n-k}} \oplus \underline{c}_2 & \cdots & \underline{e}_{2^{n-k}} \oplus \underline{c}_{2^k} \end{array} \qquad \underline{c}_i \in \mathcal{C} \;\; \forall\, i \in \{1, 2, \ldots, 2^k\}$$

Def^n: Each row of the standard array is a *coset*. There are 2^{n-k} cosets.

Def^n: The first element of a coset is the *coset leader*.

Algorithm: How to construct the standard array.

1. Put the all-zero code word as the first element of the first row.
2. Put the remaining $2^k - 1$ code words in the rest of the first row (the order is not important).
3. Select the coset leader $\underline{e}_2$ as any word that is not a code word and has the least possible weight. Put $\underline{e}_2$ as the first element of the second row. Fill in the rest of the row by adding $\underline{e}_2$ to the code word at the top of each column.
4. Select the coset leader $\underline{e}_3$ as any word that is not already in the table and has the least possible weight. Fill in the third row as above.
5. Repeat step (4) for all remaining cosets.

Demonstration:

Find the standard array for the code defined by

$$\underline{\underline{G}} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

The code words are $C = \{00000, 01101, 10110, 11011\}$.

Example:

The standard array for the main example in this section is:

000000	001101	010011	011110	100110	101011	110101	111000
000001	001100	010010	011111	100111	101010	110100	111001
000010	001111	010001	011100	100100	101001	110111	111010
000100	001001	010111	011010	100010	101111	110001	111100
001000	000101	011011	010110	101110	100011	111101	110000
010000	011101	000011	001110	110110	111011	100101	101000
100000	101101	110011	111110	000110	001011	010101	011000
001010	000111	011001	010100	101100	100001	111111	110010

Decoding: Find the received word, $\underline{r}$ in the standard array. The code word at the top of the column is the most likely transmitted one.

Example: $\underline{r} = 100101$ is found in the seventh column of the sixth coset, so the most likely code word is $\widehat{\underline{c}} = 110101$. This corresponds to message $\widehat{\underline{m}} = 110$.

Note: The coset leader gives the most probable error pattern.

Example: For $\underline{r} = 100101$ the coset leader is $\widehat{\underline{e}} = 010000$.

Syndrome Decoding

Theorem: All words in a coset have the same syndrome.

Proof: Let $\underline{r}_{i,j}$ be the word in the j^{th} column of the i^{th} coset, so

$$\underline{r}_{i,j} = \underline{e}_i \oplus \underline{c}_j$$

The syndrome of $\underline{r}_{i,j}$ is

$$\underline{s}_{i,j} = \underline{r}_{i,j}\ \underline{\underline{\mathrm{H}}}^T = \underline{e}_i\ \underline{\underline{\mathrm{H}}}^T \oplus \underline{c}_j\ \underline{\underline{\mathrm{H}}}^T = \underline{e}_i\ \underline{\underline{\mathrm{H}}}^T$$

which does not depend on the column, j.

Fact: Each coset leader has a different syndrome.

Def^n: The *syndrome table* gives the most probable error pattern for each possible syndrome.

Algorithm: How to construct the syndrome table.

1. For each coset leader in the standard array, compute the syndrome.
2. Sort the table by syndrome.

Example: Syndrome Table

coset leader	syndrome
000000	000
000001	001
000010	010
000100	100
001000	101
010000	011
100000	110
001010	111

$\Longrightarrow$

syndrome	coset leader
000	000000
001	000001
010	000010
011	010000
100	000100
101	001000
110	100000
111	001010

Algorithm: Alternate algorithm for constructing the syndrome table.

Let $T[\cdot]$ denote the syndrome table, containing 2^{n-k} entries, where $T[\underline{s}]$ is the most likely error pattern for syndrome $\underline{s}$. To construct the table without first finding the standard array, do the following:

1. Set $T[\underline{0}] = \underline{0}$.
2. For all error patterns, $\underline{e}$, of weight 1,
   (a) compute the syndrome for that error pattern, $\underline{s} = \underline{e}\ \underline{\underline{\mathrm{H}}}^T$.
   (b) if the table entry for the syndrome is empty, set $T[\underline{s}] = \underline{e}$. Otherwise, this error pattern can not be corrected.
3. Repeat step (2), but for all error patterns of weight 2.
4. Continue this process until the table is full.

Decoding: Syndrome Decoding

1. Compute the syndrome $\underline{s} = \underline{r}\ \underline{\underline{\mathrm{H}}}^T$.
2. Look up the corresponding error pattern, $\widehat{\underline{e}}$, from the syndrome table.
3. Estimate the transmitted code word as $\widehat{\underline{c}} = \underline{r} \oplus \widehat{\underline{e}}$.
4. Estimate the transmitted message, $\widehat{\underline{m}}$, as the first k bits of $\widehat{\underline{c}}$.

Error Correcting Capability

Fact: A block code with minimum distance $d_{\min}$ guarantees correcting all patterns of t or fewer errors, where $t = \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor$ (Note: $\lfloor x \rfloor$ is the largest integer $\leq x$).

Def^n: t is the *random-error correcting capability* of the code.

Note: A block code with random-error correcting capability t is usually capable of correcting some patterns of $t+1$ errors. It is capable of correcting $2^{n-k}-1$ different error patterns.

Def^n: A block code with minimum distance $d_{\min}$ that can correct all error patterns of weight t or less, but can not correct any patterns of weight $t+1$ or more is referred to as a *perfect code*.

Some Examples of Block Codes

Hamming Codes

For any $m \geq 3$ there exists a Hamming code with

$$n = 2^m - 1 \qquad d_{\min} = 3$$
$$k = 2^m - m - 1$$
$$p = n - k = m$$

The parity check matrix of a Hamming code consists of all $2^m - 1$ non-zero words of length m.

Example: The (7,4) Hamming code ($m = 3$)

$$\underline{\underline{H}} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad \text{or} \quad \underline{\underline{H}} = \left[\begin{array}{cccc|ccc} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

$$\underline{\underline{G}} = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right]$$

Example: The (15,11) Hamming code ($m = 4$)

$$\underline{\underline{H}} = \left[\begin{array}{ccccccccccc|cccc} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

All Hamming codes can detect all error patterns of weight 2 or less, and can correct all single errors. Hamming codes are *perfect codes*.

Shortened Hamming Codes

Reduce the number of message and code bits in a Hamming code by l, without reducing the number of parity bits.

$$n = 2^m - 1 - l \qquad d_{\min} \geq 3$$
$$k = 2^m - m - 1 - l$$
$$p = n - k = m$$

Example: The (6,3) shortened Hamming code ($m = 3$, $l = 1$)

$$\underline{\underline{G}} = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right]$$

Cyclic Codes

Cyclic codes are linear block codes with the additional constraint that every cyclic shift of a codeword is also a codeword. If

$\underline{c} = (c_0, c_1, c_2, \ldots, c_{n-1}) \in \mathcal{C}$

then

$\underline{c}^{(1)} = (c_{n-1}, c_0, c_1, \ldots, c_{n-2}) \in \mathcal{C}$

where $\underline{c}^{(1)}$ is the right *cyclic shift* of $\underline{c}$.

Example: If 0100111 is a code word of a code, so is 1010011.

Codes with this structure allow for simple implementations of the encoder and the syndrome calculator using shift registers. There is no need for complex matrix multiplications.

Note: Cyclic codes are generally discussed in terms of polynomials.

Note: Every codeword can be represented by a polynomial

$$\text{i.e. } \underline{c} = (c_0, c_1, c_2, \ldots, c_{n-1}) \Longleftrightarrow c(X) = c_0 + c_1X + c_2X^2 + \cdots c_{n-1}X^{n-1}$$

where $c_i \in \{0, 1\}$ for binary cyclic codes.

Note: Cyclic shifts can be expressed in terms of polynomials

$$\begin{aligned} c^{(1)}(X) &= c_{n-1} + c_0X + c_1X^2 + \cdots + c_{n-2}X^{n-1} \\ &= c_{n-1} + Xc(X) - c_{n-1}X^n \\ &= c_{n-1} + Xc(X) + c_{n-1}X^n \\ &= Xc(X) + c_{n-1}(X^n + 1) \end{aligned}$$

i.e., $c^{(1)}(X)$ is the remainder from dividing $Xc(X)$ by $(X^n + 1)$.
In general,

$$c^{(i)}(X) \text{ is the remainder from dividing } X^ic(X) \text{ by } (X^n + 1)$$

$$c^{(i)}(X) = X^ic(X) + (c_{n-i} + Xc_{n-i+1} + \cdots + X^{i-1}c_{n-1})(X^n + 1)$$

Def^n: Cyclic codes are defined by a generator polynomial

$$g(X) = 1 + g_1X + g_2X^2 + \cdots g_{n-k-1}X^{n-k-1} + X^{n-k}$$

of degree $n - k$, with $g_i \in \{0, 1\}$ for binary cyclic codes.

Encoding of cyclic codes:

Message polynomial $m(X)$ is encoded to code polynomial $c(X)$ with

$c(X) = m(X)g(X)$

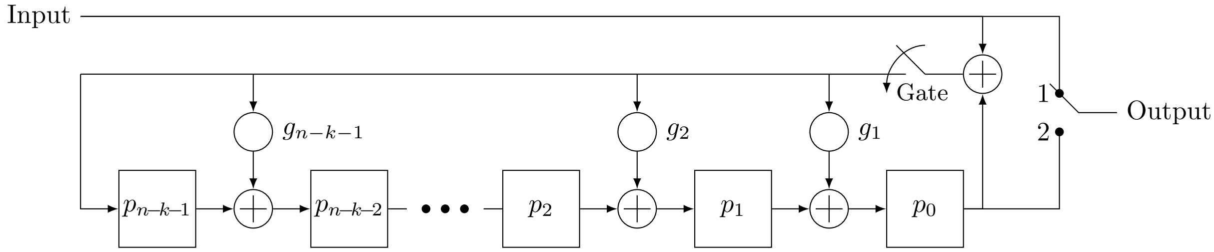
The generator matrix for a cyclic code can be expressed (in non-systematic form) as

$$\underline{\underline{G}} = \begin{bmatrix} 1 & g_1 & g_2 & \cdots & g_{n-k-1} & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & g_1 & \cdots & g_{n-k-2} & g_{n-k-1} & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & g_1 & g_2 & g_3 & \cdots & g_{n-k-1} & 1 \end{bmatrix}$$

Note: It is possible to find an equivalent cyclic code in systematic form.

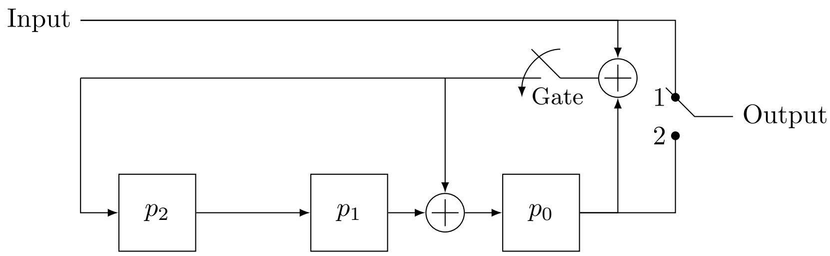
Systematic encoding of cyclic codes

Systematic cyclic codes can be encoded using a shift register.



1. Close gate, set switch to position 1.
2. Shift in the k message bits
3. Open gate, set switch to position 2.
4. Shift out contents of shift register.

Example: $n = 7$, $k = 4$, $n - k = 3$.
$g(X) = 1 + X + X^3 \iff \underline{g} = 1101$



$\underline{m} = 1100$

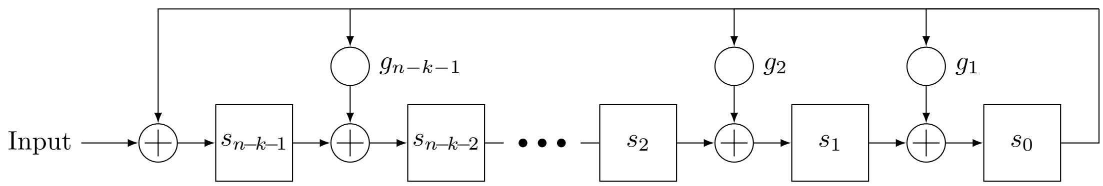
message bit	feedback bit	p_2	p_1	p_0	code bit
		0	0	0	
1	1	1	0	1	1
1	0	0	1	0	1
0	0	0	0	1	0
0	1	1	0	1	0
	0	0	1	0	1
	0	0	0	1	0
	0	0	0	0	1

$\underline{c} = 1100101$

$$\underline{\underline{G}}_{\text{sys}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

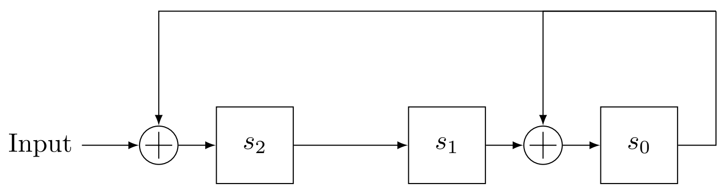
Syndrome Calculation:

The syndrome of systematic cyclic codes can also be calculated easily using a shift register.



1. Shift in the n received bits.
2. Syndrome is stored as the contents of shift register.

Example: $n = 7$, $k = 4$, $n - k = 3$.
$g(X) = 1 + X + X^3 \Longleftrightarrow \underline{g} = 1101$



$\underline{r} = 1100110$

received bit	s_2	s_1	s_0
	0	0	0
1	1	0	0
1	1	1	0
0	0	1	1
0	1	0	0
1	1	1	0
1	1	1	1
0	1	1	0

$\underline{s} = 011$

Cyclic Redundancy Check (CRC) Codes

Often used for error detection with ARQ schemes. One commonly used generator is

$$g(X) = 1 + X^2 + X^{15} + X^{16}$$

$$\underline{g} = 10100000000000011$$

Bose-Chaudhuri-Hocqunghem (BCH) Codes

A large class of cyclic codes. For any $m \geq 3$ and $t \geq 1$ there is a BCH code with

$$\begin{aligned} n &= 2^m - 1 \\ p &= n - k \leq mt \\ d_{\min} &= 2t + 1 \end{aligned}$$

These codes can correct all combinations of t or fewer errors.

Reed-Solomon Codes

Non-binary BCH codes, which work with symbols of k bits each. Message words consist of K m-bit symbols, and codewords consists of N m-bit symbols, where

$$N = 2^m - 1$$

The code rate is

$$R = \frac{K}{N}$$

Reed-Solomon codes can correct up to

$$t = \left\lfloor \tfrac{1}{2}(N - K) \right\rfloor$$

symbol errors. Good for correcting error bursts.

Example: (31,15) Reed-Solomon Code

$m = 5$ bits per symbol

$K = 15$ symbols, or 75 bits

$N = 31$ symbols, or 155 bits

This code can correct up to 8 symbol errors.

Convolutional Codes

As opposed to block codes, which operate on finite-length blocks of message bits, a convolutional encoder operates on a continuous sequence of message symbols. Let

$\underline{a} = a_1\ a_2\ a_3 \ldots$

denote the message sequence and

$\underline{c} = c_1\ c_2\ c_3 \ldots$

denote the code sequence.

At each clock cycle, a (n, k, m) convolutional encoder takes one message symbol of k message bits and produces one code symbol of n code bits. Typically k and n are small integers (less than 5), with $k < n$. The parameter m refers to the memory requirement of the encoder. Increasing m improves the performance of the code, but at increased decoder complexity (typically $m \leq 8$).

The basis for generating convolutional codes is the convolution of the message sequence with a set of generator sequences. Let

$\underline{g} = g_0\ g_1\ g_2 \ldots g_L$

denote a generator sequence of length $L + 1$ bits, and let the convolution of $\underline{a}$ and $\underline{g}$ be $\underline{b} = b_1\ b_2\ b_3 \ldots$, with each output bit given by

$$b_i = \sum_{l=0}^{L} a_{i-l}\ g_l\ .$$

$(2, 1, m)$ Convolutional Codes

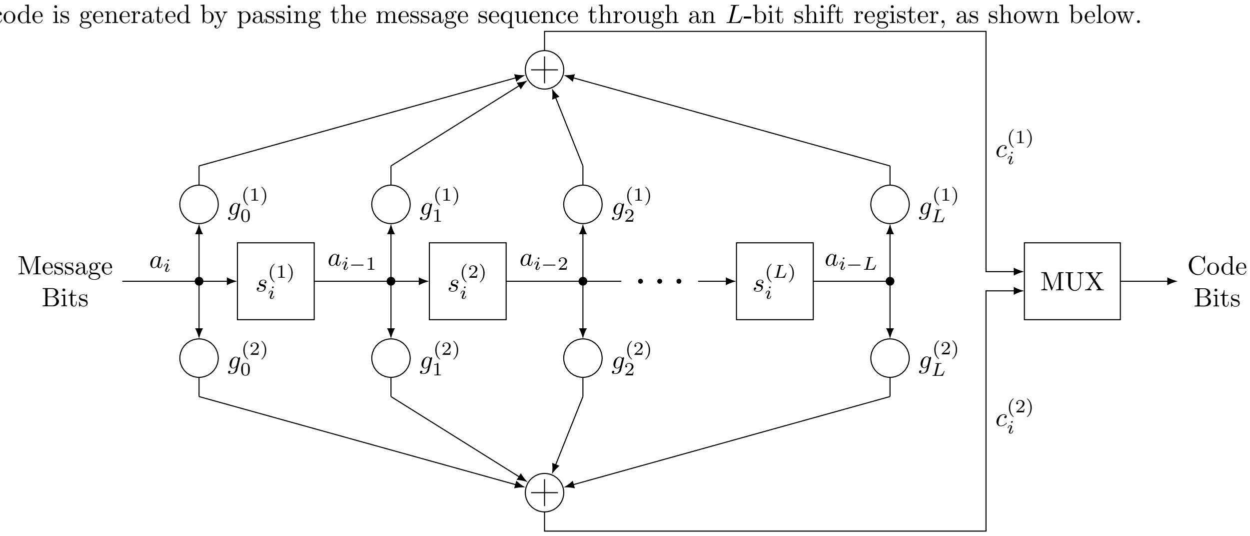
For a rate 1/2 convolutional code, two generator sequences, denoted by $\underline{g}^{(1)}$ and $\underline{g}^{(2)}$, are used. The two convolution output sequences are $\underline{c}^{(1)}$ and $\underline{c}^{(2)}$, with

$$c_i^{(1)} = \sum_{l=0}^{L} a_{i-l}\ g_l^{(1)} \qquad\qquad c_i^{(2)} = \sum_{l=0}^{L} a_{i-l}\ g_l^{(2)}\ .$$

These two sequences are multiplexed together, so the resulting code sequence is

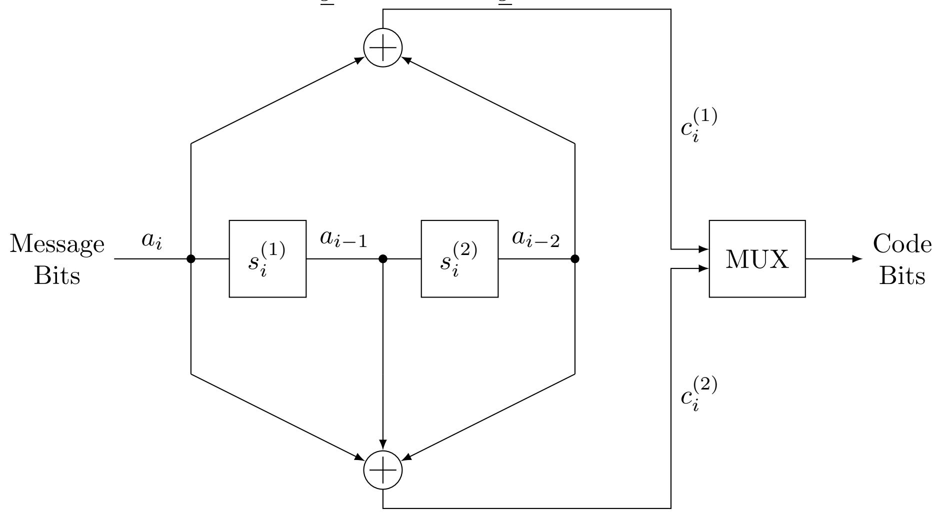
$\underline{c} = c_1^{(1)}\ c_1^{(2)}\ c_2^{(1)}\ c_2^{(2)}\ c_3^{(1)}\ c_3^{(2)}\ \ldots\ .$

The code is generated by passing the message sequence through an L-bit shift register, as shown below.



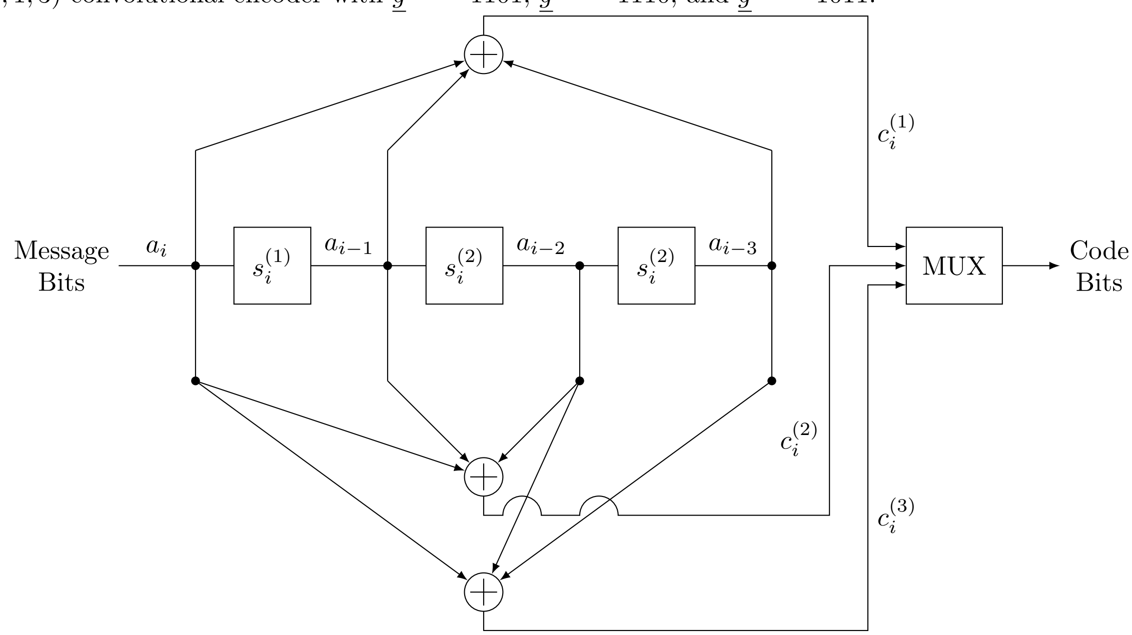
This is a rate 1/2 encoder because for each encoder clock cycle one message bit $(k = 1)$ enters the encoder and two code bits $(n = 2)$ are produced. The memory, m, is simply equal to L.

Example: A $(2,1,2)$ convolutional encoder with $\underline{g}^{(1)} = 101$ and $\underline{g}^{(2)} = 111$.



$(n,1,m)$ Convolutional Codes

Rate $1/n$ codes can be constructed by using n different generators.
Example: $(3,1,3)$ convolutional encoder with $\underline{g}^{(1)} = 1101$, $\underline{g}^{(2)} = 1110$, and $\underline{g}^{(3)} = 1011$.



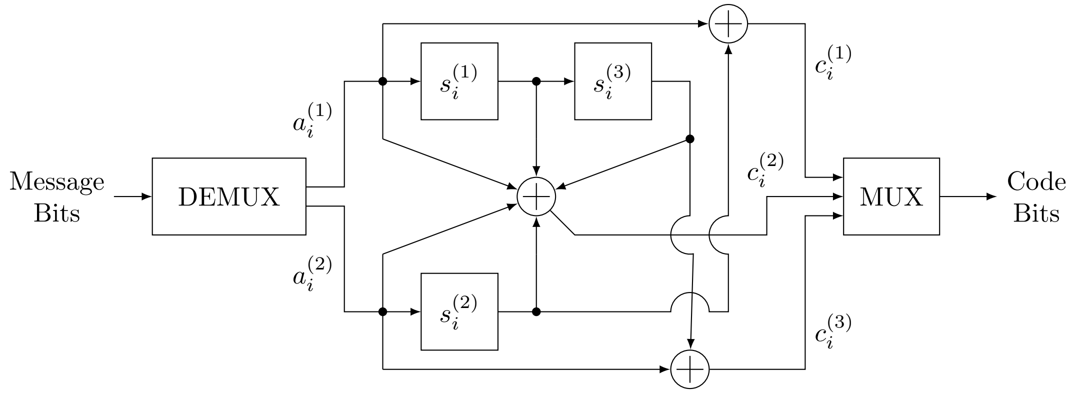
This is a rate 1/3 encoder because at each clock cycle one message bit $(k = 1)$ enters the encoder and three code bits $(n = 3)$ are produced. The memory, m, is three bits.

(n, k, m) Convolutional Codes

By using multiple shift registers, arbitrary rate k/n codes can be constructed. The input sequence is demultiplexed into k separate streams, which are passed through k shift registers. Thus one message symbol of k message bits enters the encoder with each encoder clock cycle, and one code symbol of n code bits is produced.

Example: A $(3, 2, 3)$ convolutional encoder, with generators

$$\underline{g}^{(1,1)} = 100 \qquad \underline{g}^{(2,1)} = 01$$
$$\underline{g}^{(1,2)} = 111 \qquad \underline{g}^{(2,2)} = 11$$
$$\underline{g}^{(1,3)} = 001 \qquad \underline{g}^{(2,3)} = 10$$



This is a rate 2/3 encoder because at each clock cycle two message bits ($k = 2$) enter the encoder and three code bits ($n = 3$) are produced. The total memory, m, is three bits.

Def^n: Let L_j denote the length of the j^{th} shift register, for $j \in \{1, 2, \ldots, k\}$, so the *total memory* of the encoder is

$$m = \sum_{j=1}^{k} L_j \ .$$

The length of the longest shift register is

$$L = \max_j \{L_j\} \ .$$

The *constraint length* of a convolutional code is defined as

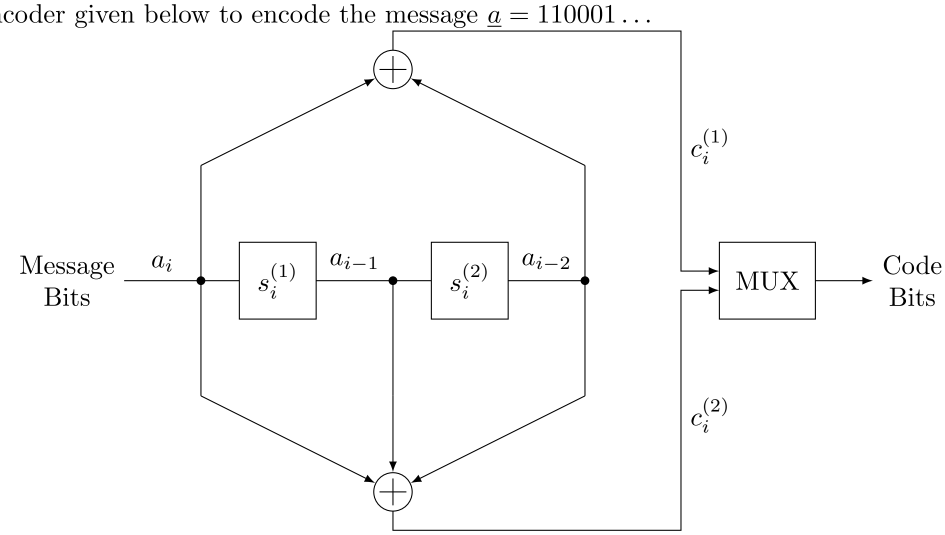
$$K = \max_j \{L_j\} + 1 \ .$$

The constraint length specifies the total number of message symbols that are used in determining each code symbol.

Def^n: The *state* of the encoder at any time is given by the values of the memory bits, $s_1 \ s_2 \ \ldots \ s_m$. The total number of different states the encoder can be in is $N_S = 2^m$.

Fact: The encoder output for any clock cycle is determined only by the k input bits currently entering the encoder and the m state bits of the encoder.

Example Use the encoder given below to encode the message $\underline{a} = 110001\ldots$



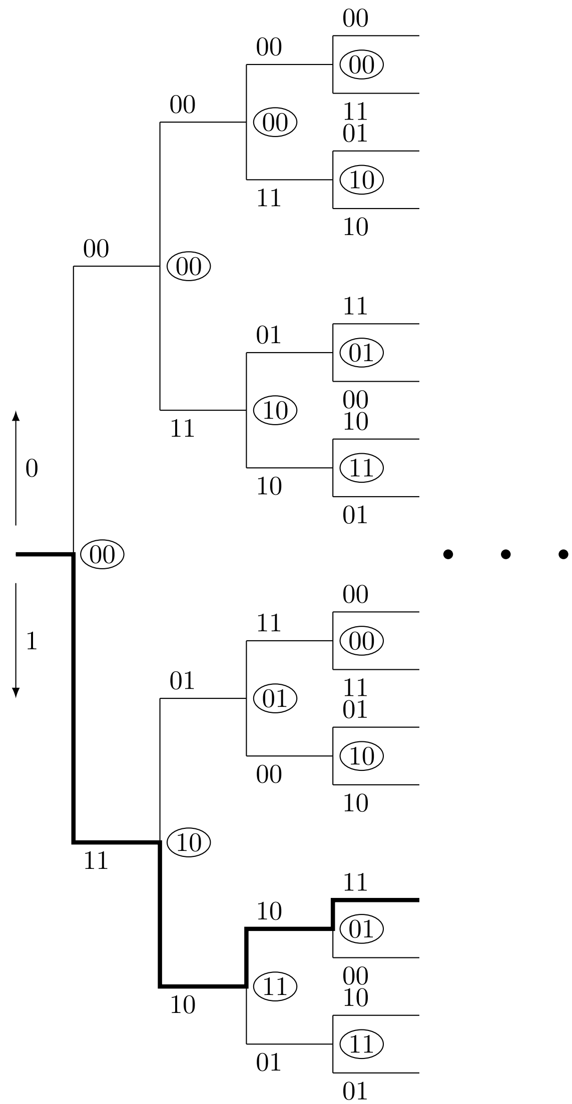
i	a_i	$s_i^{(1)}$	$s_i^{(2)}$	$c_i^{(1)}$	$c_i^{(2)}$
1	1	0	0	1	1
2	1	1	0	1	0
3	0	1	1	1	0
4	0	0	1	1	1
5	0	0	0	0	0
6	1	0	0	1	1
⋮	⋮	⋮	⋮	⋮	⋮

$$\underline{c}^{(1)} = 111101\ldots$$
$$\underline{c}^{(2)} = 100101\ldots$$

The transmitted code word is $\underline{c} = 111010110011\ldots$

Code Representations

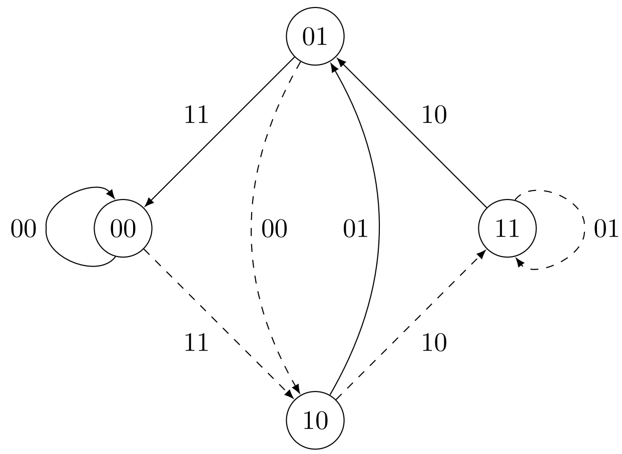
Tree Diagram: The relationship between message sequences and code sequences can be represented in a tree diagram. For the $(2,1,2)$ code given above, the tree diagram is:



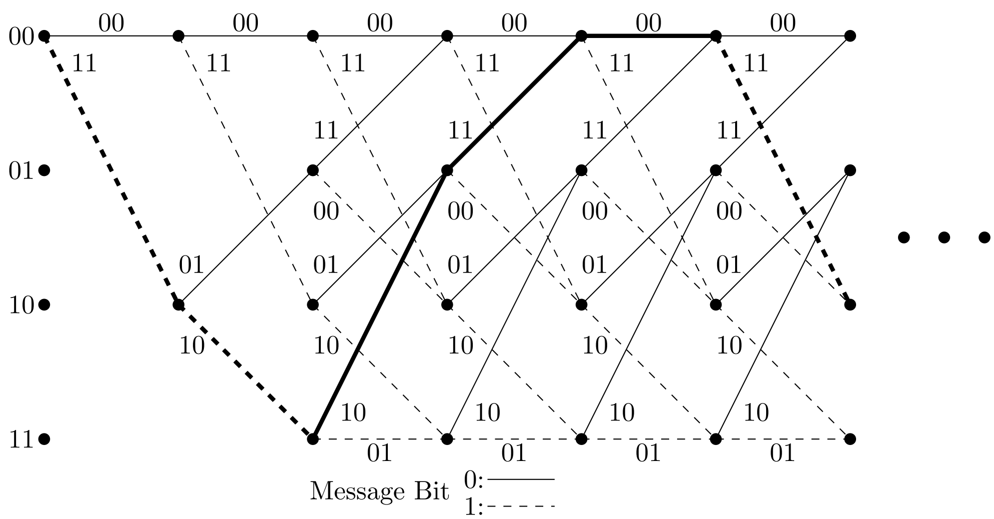
The thick lines represent the encoding of the message sequence $\underline{a} = 1100\ldots$, which, by reading off the bits along the path, gives the code sequence $\underline{c} = 11101011\ldots$

Close observation of the tree diagram reveals that the structure repeats itself after the third stage. That is, the output in the fourth and following steps does not depend on the first input bit. This is because the encoder output at time i depends only on the input at time i, and the 2 previous inputs (i.e., the constraint length is 3).

State Diagram: The code structure can also be represented as a state diagram, showing the output produced by transitions between encoder states.



Trellis Diagram: Because of the repetitive nature of the tree diagram, the code structure can be represented with a trellis diagram by merging states.



Block encoding with convolutional codes

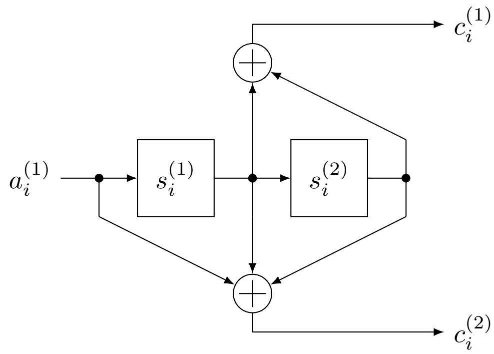
Although convolutional codes are designed to work with a continuous stream of message bits, they may also be used for finite-length blocks. When encoding a block of N_a message symbols with a rate k/n convolutional encoder, performance can be improved by feeding sufficient zeros into the encoder after the block has been encoded to drive the encoder state to 0. A total of $K-1$ extra zero symbols must be encoded to force all the memory bits to 0. The code symbols produced during this operation are transmitted along with the regular code sequence. Therefore the total code sequence length is

$$N_c = N_a + K - 1$$

symbols. The actual rate of the code is therefore

$$R = \frac{kN_a}{nN_c} = \frac{k}{n}\,\frac{N_a}{N_a + K - 1}$$

Tutorial: Constructing Trellis Diagrams



Parameters:

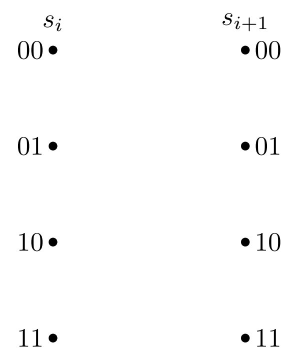
$k = 1$ – # of bits per message symbol
$n = 2$ – # of bits per code symbol
$m = 2$ – # of memory bits
$N_S = 2^m = 4$ – # of states

Notation:

Message symbol	$a_i = a_i^{(1)}$
Code symbol	$c_i = c_i^{(1)} c_i^{(2)}$
Encoder state	$s_i = s_i^{(1)} s_i^{(2)}$

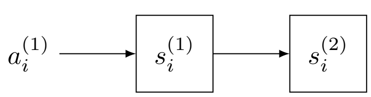
Possible states: $s_i \in \{00, 01, 10, 11\}$

One section of the trellis will look something like this: $\Longrightarrow$



State Transitions

We need to find the mapping from the current state (s_i) and input (a_i) to the next state (s_{i+1}). That is, construct a look-up table for $s_{i+1} = \mathrm{ST}[s_i, a_i]$. It may be easier to find this mapping by redrawing the encoder without any of the code symbol output connections.



$\Downarrow$

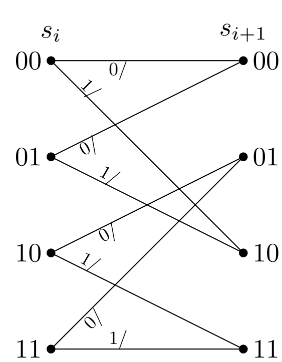
$$s_i = s_i^{(1)} s_i^{(2)} \qquad s_{i+1} = a_i^{(1)} s_i^{(1)} \quad \Longrightarrow \quad s_{i+1}^{(1)} = a_i^{(1)} \qquad s_{i+1}^{(2)} = s_i^{(1)}$$

$\Longrightarrow$

State Transition Table

$\mathrm{ST}[s,a]$		$a=0$	$a=1$
s	00	00	10
	01	00	10
	10	01	11
	11	01	11

These state transitions can then be placed on the section of the trellis:
(The number before the slash (/) on each branch is the value of the message symbol that will cause the transition.) $\Longrightarrow$



Code Symbols

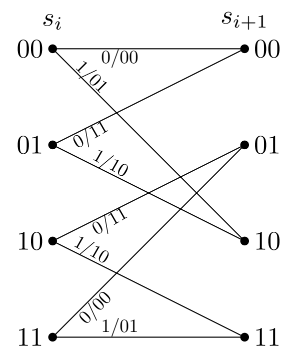
We can construct a look-up table that gives the code symbol that is generated with each message symbol and state. That is, construct the mapping $c_i = \mathrm{SG}[s_i, a_i]$. The code symbols are calculated with:

$$
\begin{aligned}
c_i^{(1)} &= s_i^{(1)} \oplus s_i^{(2)} \\
c_i^{(2)} &= a_i^{(1)} \oplus s_i^{(1)} \oplus s_i^{(2)}
\end{aligned}
\quad \Longrightarrow
$$

Symbol Generation Table

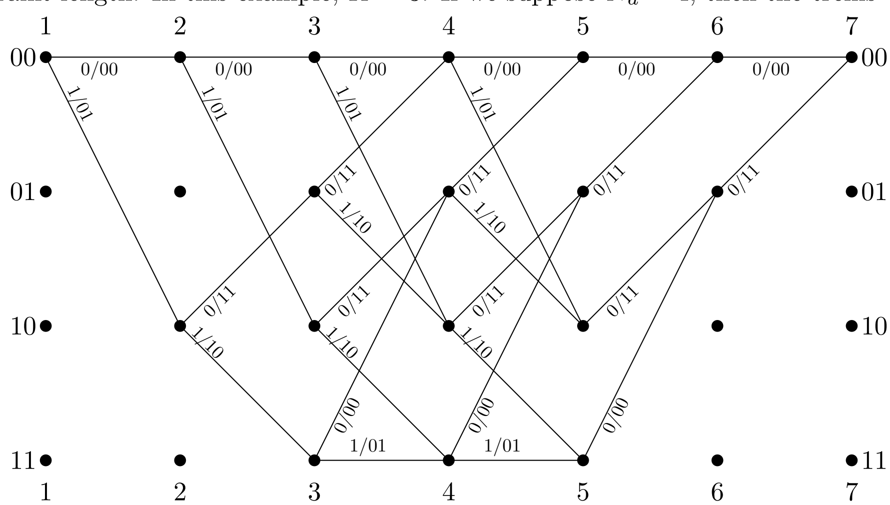
SG$[s, a]$		a = 0	a = 1
s	00	00	01
	01	11	10
	10	11	10
	11	00	01

These can be used to label the branches of the trellis section:
(The code symbols are indicated after the slash (/) on each branch.) $\Longrightarrow$



Trellis Diagram

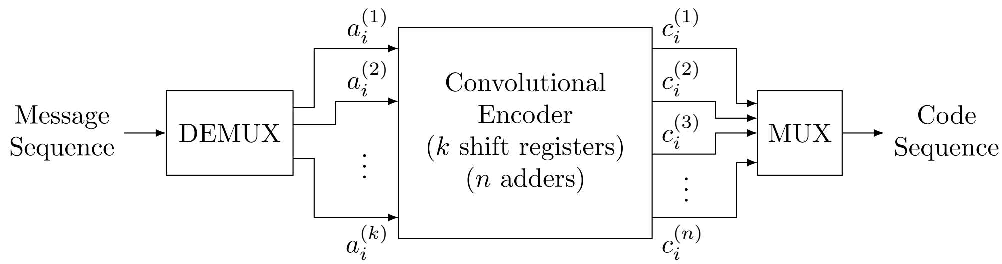
A complete trellis diagram is produced by repeating the section. The total number of sections should be $N_a + (K - 1)$, where K is the constraint length. In this example, $K = 3$. If we suppose $N_a = 4$, then the trellis would be:



Note that inaccessible states and branches have been removed at the beginning and end of the trellis, since the encoder always starts and ends each block in the all-zero state.

Decoding of Convolutional Codes

Notation



<u>Message Sequence:</u>

- consists of N_a symbols, with each symbol composed of k bits.
- denoted by

$$\underline{a} = \underbrace{a_1^{(1)} a_1^{(2)} \cdots a_1^{(k)}}_{a_1} \; \underbrace{a_2^{(1)} a_2^{(2)} \cdots a_2^{(k)}}_{a_2} \; \cdots \; \underbrace{a_i^{(1)} a_i^{(2)} \cdots a_i^{(k)}}_{a_i} \; \cdots \; \underbrace{a_{N_a}^{(1)} a_{N_a}^{(2)} \cdots a_{N_a}^{(k)}}_{a_{N_a}}$$

- i^{th} message symbol is $a_i = a_i^{(1)} a_i^{(2)} \ldots a_i^{(k)}$, where $a_i^{(j)} \in \{0, 1\}$ is a single bit.
- for each encoder clock cycle, one message symbol (of k bits) enters the encoder.

<u>Code Sequence:</u>

- consists of N_c symbols, with each symbol composed of n bits.
- denoted by

$$\underline{c} = \underbrace{c_1^{(1)} c_1^{(2)} \cdots c_1^{(n)}}_{c_1} \; \underbrace{c_2^{(1)} c_2^{(2)} \cdots c_2^{(n)}}_{c_2} \; \cdots \; \underbrace{c_i^{(1)} c_i^{(2)} \cdots c_i^{(n)}}_{c_i} \; \cdots \; \underbrace{c_{N_c}^{(1)} c_{N_c}^{(2)} \cdots c_{N_c}^{(n)}}_{c_{N_c}}$$

- i^{th} code symbol is $c_i = c_i^{(1)} c_i^{(2)} \ldots c_i^{(n)}$, where $c_i^{(j)} \in \{0, 1\}$ is a single bit.
- for each encoder clock cycle, one code symbol (of n bits) is produced by the encoder.
- $N_c = N_a + (K - 1)$ where K is the constraint length of the code.
- The extra $(K - 1)$ symbols arise as the encoder is forced back to the all-zero state.

<u>Received Sequence:</u>

- also consists of N_c symbols, with each symbol composed of n bits.
- denoted by

$$\underline{r} = \underbrace{r_1^{(1)} r_1^{(2)} \cdots r_1^{(n)}}_{r_1} \; \underbrace{r_2^{(1)} r_2^{(2)} \cdots r_2^{(n)}}_{r_2} \; \cdots \; \underbrace{r_i^{(1)} r_i^{(2)} \cdots r_i^{(n)}}_{r_i} \; \cdots \; \underbrace{r_{N_c}^{(1)} r_{N_c}^{(2)} \cdots r_{N_c}^{(n)}}_{r_{N_c}}$$

- i^{th} received symbol is $r_i = r_i^{(1)} r_i^{(2)} \ldots r_i^{(n)}$, where $r_i^{(j)} \in \{0, 1\}$ is a single bit.
- j^{th} bit of the i^{th} symbol is given by $r_i^{(j)} = c_i^{(j)} \oplus e_i^{(j)}$, where $e_i^{(j)}$ is the bit error indicator which is equal to 1 if the bit is in error, and 0 otherwise.
- the *channel transition probability* is

$$\Pr\left\{ r_i^{(j)} \mid c_i^{(j)} \right\} = \begin{cases} 1 - p\,, & \text{if } r_i^{(j)} = c_i^{(j)} \\ p\,, & \text{if } r_i^{(j)} \neq c_i^{(j)} \end{cases}$$

As an example, consider the rate 1/3, constraint length $K = 3$ encoder with generators $\underline{g}^{(1)} = 110$, $\underline{g}^{(2)} = 111$, and $\underline{g}^{(3)} = 101$ shown below:

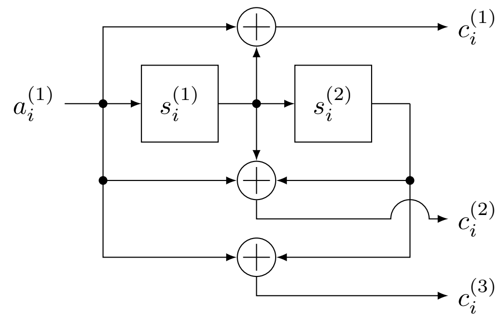
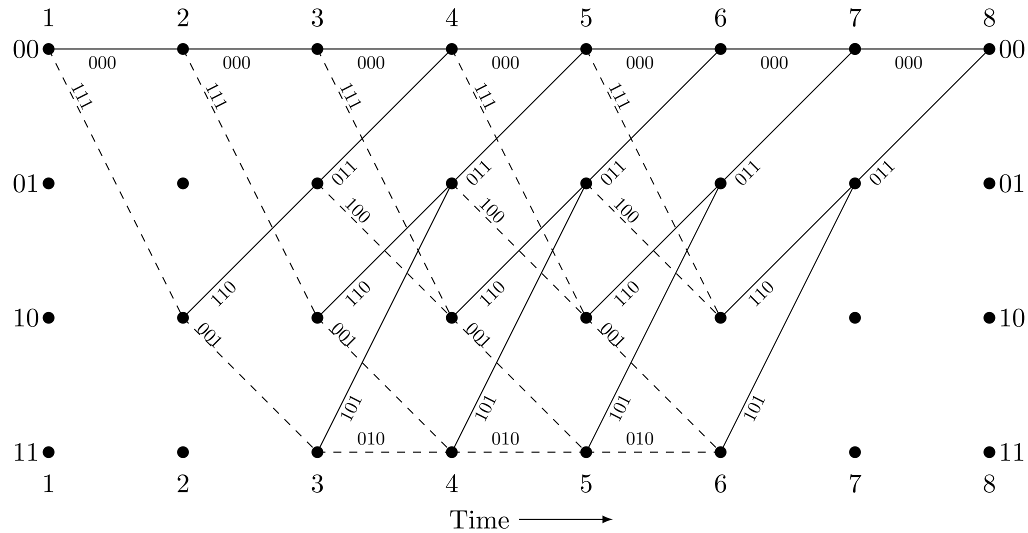


Figure 2.

For a message sequence length of $N_a = 5$ symbols, the corresponding trellis structure is:



- A solid line is used to represent a message symbol of $a_i = 0$, while a dashed line represents $a_i = 1$.
- There are 2^k branches entering each state, and 2^k branches leaving each state.
- Each branch is labelled with the corresponding output code symbol, c_i.
- The trellis contains $N_c + 1$ time units or levels.
- Since the encoder always starts and ends in the all-zero state, the first $K-1$ time units correspond to the encoder's departure from the all-zero state, and the last $K-1$ time units correspond to the encoder's return to the all-zero state.
- Therefore, not all states can be reached in the first $K-1$ or last $K-1$ time units.
- However, in the centre portion, all states can be reached, and the trellis structure repeats with each time unit.
- For example, for the message $\underline{a} = 11001$ the encoder follows the highlighted path through the trellis, and the corresponding code sequence is $\underline{c} = 111\ 001\ 101\ 011\ 111\ 110\ 011$.
- Each of the 2^{kN_a} possible message sequences (and code sequences) is represented by a unique path through the trellis.

Maximum Likelihood Sequence Estimation (MLSE)

As far as the receiver is concerned, one out of a total of 2^{kN_a} possible message sequences was transmitted. Based on the received word, $\underline{r}$, the receiver must determine which message sequence was most likely to have been transmitted. If all possible message sequences have equal *a priori* probability, the maximimum likelihood decision rule is to choose $\widehat{\underline{a}} = \underline{a}$ if

$$\Pr\{\underline{r} \mid \underline{a}\} \geq \Pr\{\underline{r} \mid \underline{a}'\}$$

for all other possible message words, $\underline{a}'$. That is, the receiver must find $\underline{a}$ which maximizes $\Pr\{\underline{r} \mid \underline{a}\}$.
Because of the one-to-one relationship between message sequences and code sequences, this is equivalent to finding $\underline{a}$ which maximizes $\Pr\{\underline{r} \mid \underline{c}\}$, where $\underline{c}$ is the code sequence corresponding to $\underline{a}$.

If $\underline{c}$ is transmitted and $\underline{r}$ is received, the number of code bit errors is the number of bit positions in which $\underline{c}$ and $\underline{r}$ differ. This is given by $d_H(\underline{r}, \underline{c})$. Since a total of nN_c code bits are transmitted, the probability of receiving $\underline{r}$ given that $\underline{c}$ was transmitted is, for a BSC with crossover probability p,

$$\begin{aligned}\Pr\{\underline{r} \mid \underline{c}\} &= p^{d_H(\underline{r},\underline{c})}(1-p)^{nN_c - d_H(\underline{r},\underline{c})} \\ &= \left(\frac{p}{1-p}\right)^{d_H(\underline{r},\underline{c})} (1-p)^{nN_c}\end{aligned}$$

where $d_H(\underline{r}, \underline{c})$ is the Hamming distance between the received word $\underline{r}$ and code sequence $\underline{c}$ (i.e., the number of bits in which $\underline{r} \neq \underline{c}$).

Because the log function is monotonically increasing, maximizing the likelihood function is equivalent to maximizing the log-likelihood function

$$\begin{aligned}\log \Pr\{\underline{r} \mid \underline{c}\} &= \log\left\{\left(\frac{p}{1-p}\right)^{d_H(\underline{r},\underline{c})} (1-p)^{nN_c}\right\} \\ &= d_H(\underline{r},\underline{c}) \log\left(\frac{p}{1-p}\right) + nN_c \log(1-p) \ .\end{aligned}$$

Assuming $0 < p < 0.5$ (so that $\log \frac{p}{1-p} < 0$), maximizing the log-likelihood function is equivalent to finding the code sequence which minimizes the *path metric*

$$\mathrm{M}(\underline{r} \mid \underline{c}) = d_H(\underline{r}, \underline{c}) \ .$$

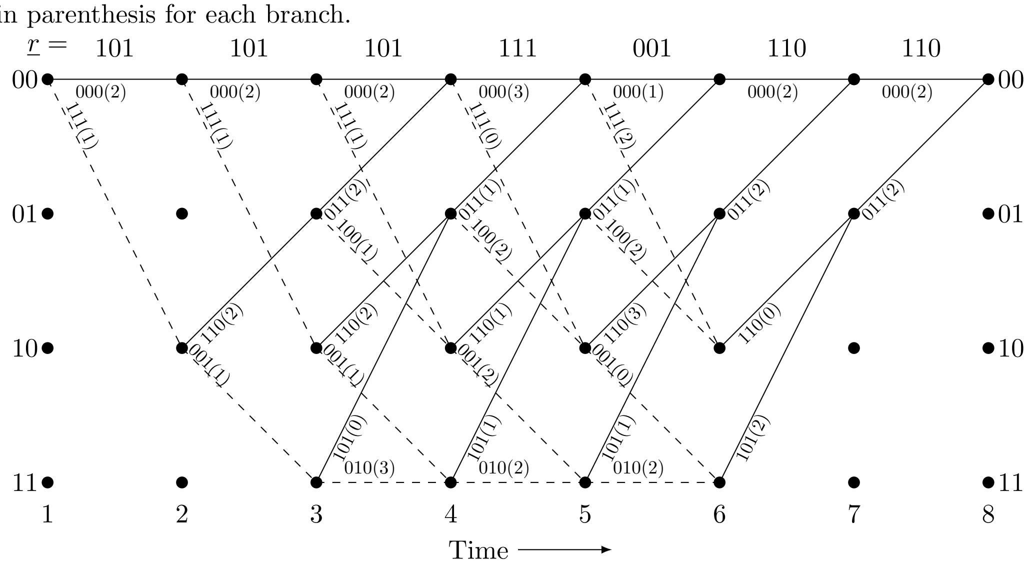
The trellis diagram is a useful tool for finding the path metric for each of the 2^{kN_a} possible message sequences. Note that the path metric can be written as

$$\mathrm{M}(\underline{r} \mid \underline{c}) = \sum_{i=1}^{N_c} d_H(r_i, c_i) = \sum_{i=1}^{N_c} \mu(r_i \mid c_i) \ ,$$

which is the sum of the *branch metrics*

$$\mu(r_i \mid c_i) = d_H(r_i, c_i) \ .$$

As an example of using the trellis to calculate the branch metric, suppose that the encoder shown in Fig. 2 is used and the received sequence is: $\underline{r} = 101\ 101\ 101\ 111\ 001\ 110\ 110$. The trellis diagram can be redrawn as below, with the branch metrics shown in parenthesis for each branch.



To find the path metric for a given message sequence, the decoder only needs to trace the message through the trellis, summing the branch metrics. For example, the path for the message $\underline{a} = 10110$ is highlighted in the trellis. The path metric is

$$\mathrm{M}(\underline{r} \mid \underline{c}) = 1 + 2 + 1 + 2 + 1 + 2 + 2 = 11$$

By tracing all 2^{kN_a} possible paths through the trellis, the decoder can find the path metrics of all 2^{kN_a} possible message sequences, and select that sequence which has the smallest path metric as its estimate of the transmitted message.

The Viterbi Algorithm

The Viterbi algorithm provides a simple method for decoding convolutional codes which is optimal in that it always finds the path with the smallest path metric.

Algorithm:

1) Beginning at time unit $i = K$, compute the partial path metric

$$\mathrm{M}([\underline{r} \mid \underline{c}]_{i-1}) = \sum_{l=1}^{i-1} \mu(r_l \mid c_l)$$

for the single path entering each state. Store the path (called the *survivor*) and its metric for each state.

2) Increase i by 1. Compute the partial path metric for all paths entering each state by adding the branch metric entering that state to the partial path metric of the corresponding survivor at the previous time unit. For each state, store the path with the smallest partial path metric (the survivor) together with its metric, and eliminate all other paths.

3) If $i \leq N_c$, repeat step 2. Otherwise, stop.

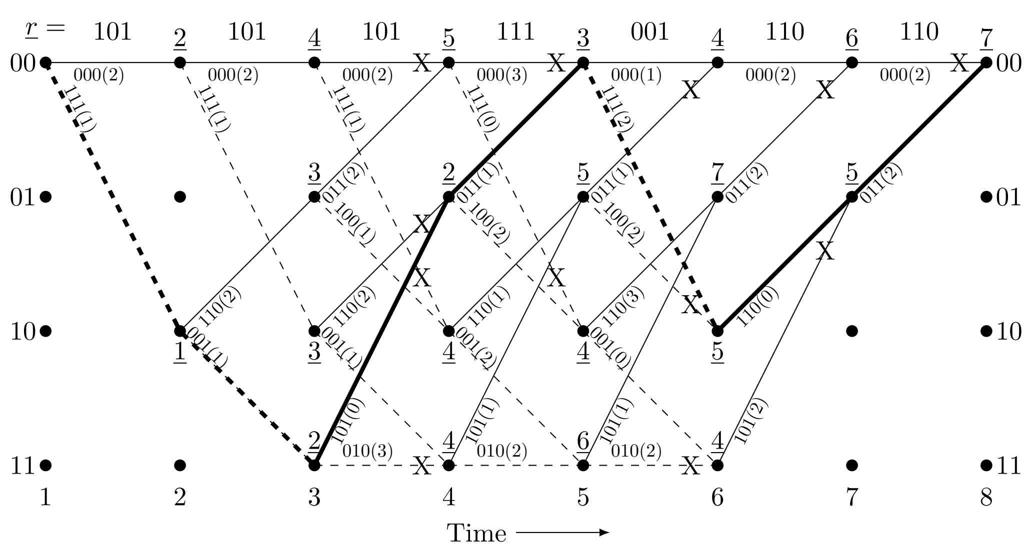
Notes:

– For time unit K to $N_a + 1$ there are N_S survivors, one for each state.

– After time unit $N_a + 1$ there are fewer survivors since there are fewer valid states while the encoder is returning to the all-zero state.

– Finally, at time unit $N_c + 1$ there is only one valid state, the all-zero state, and hence there is only one survivor.

Example: For the encoder of Fig. 2 with $\underline{r}$ = 101 101 101 111 001 110 110, the trellis diagram below shows the result of the Viterbi algorithm. The underlined number at each state shows the partial path metric of the survivor path at that state. Branches marked with an X represent eliminated paths.



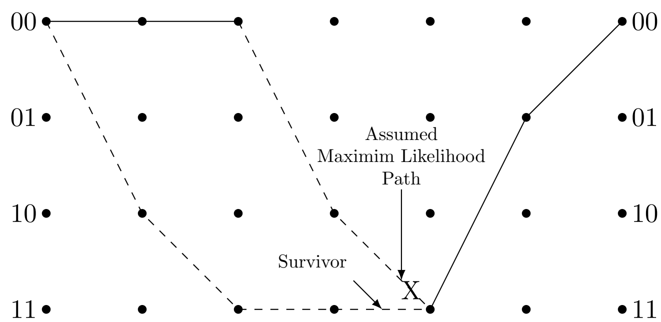
The final survivor path, highlighted in the trellis, has a path metric of 7, and corresponds to the message sequence $\underline{a} = 11001$.

Note that at some states neither path is crossed out, indicating a tie. If the final survivor goes through any of these states, there is more than one maximum likelihood path. Either path can be selected as the best path, without affecting the average error probability of the decoder.

Theorem: The final survivor $\underline{\widehat{a}}$ in the Viterbi algorithm is the path with maximum likelihood. That is, it has the smallest path metric, so

$$\mathrm{M}(\underline{r} \mid \underline{\widehat{c}}) \leq \mathrm{M}(\underline{r} \mid \underline{c}) \qquad \forall\, \underline{c} \neq \underline{\widehat{c}}\,.$$

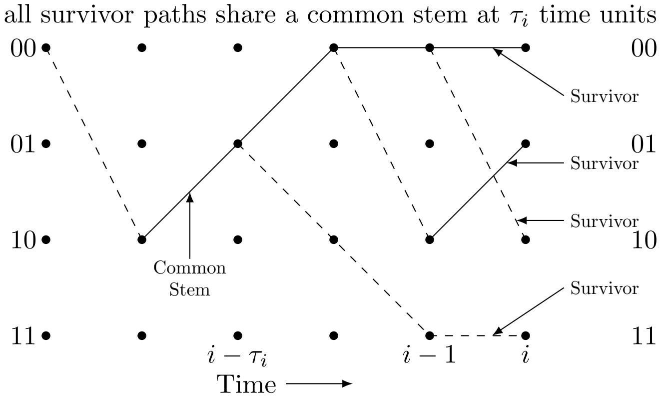
Proof: Suppose the maximum likelihood path is eliminated by the algorithm at time unit i, as illustrated below. This implies that the partial path metric of the survivor is less than that of the maximum likelihood path at this point. Now, if the remaining portion of the maximum likelihood path is appended to the survivor at time unit i, the total metric of this path will be less than the total metric of the maximum likelihood path. This contradicts the definition of the maximum likelihood path as the path with the smallest metric. Therefore, the maximum likelihood path cannot be eliminated by the algorithm, and must be the final survivor.



Continuous Decoding

Since a final decision on the maximum likelihood path is not made until the entire received sequence has arrived, this may cause an unacceptably long decoding delay if the message length is long. Furthermore, the length of the survivor paths grows over time, placing a burden on memory requirements. To address these issues, some modifications to the Viterbi algorithm are needed.

In practice, at any time unit i, all survivor paths share a common stem at τ_i time units back, as illustrated below.

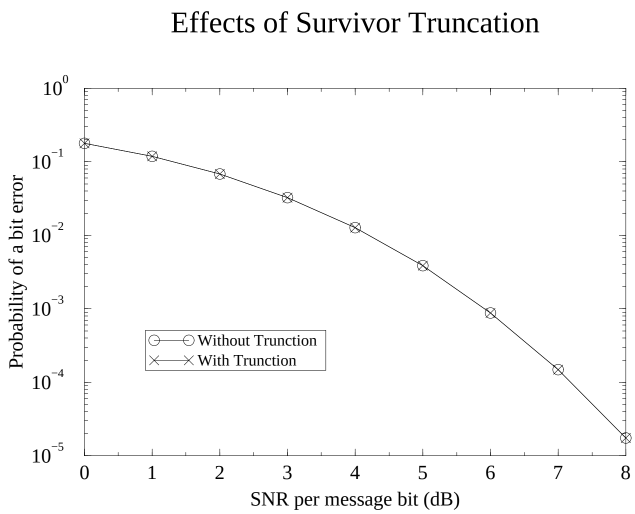


Regardless of which of the N_S survivors is the final survivor, the maximum likelihood path will contain the common stem. The Viterbi algorithm can then modified so that at time unit i the decoder compares all survivors to find a common stem. If one is found, the message symbols for the stem are emitted by the decoder, and the survivors are truncated to the last τ_i message symbols.

This approach is optimal in that the message sequence produced by the decoder is the true maximum likelihood path. However, the output of the decoder is produced at a variable rate. A long time can pass while the survivors do not share a common stem, and then suddenly several message symbols are produced all at once when a common stem finally occurs.

Survivor Truncation:

A more practical alternative is to pick a fixed delay, τ, and at each time unit, i, make a decision about the message symbol $a_{i-\tau}$. This can be done simply by assuming that the path with the smallest partial path metric at time i will in fact turn out to be the maximum likelihood path, and tracing back along that path by τ time units to find $a_{i-\tau}$. Note that the resulting decoder is sub-optimal since this path is not necessarily the final survivor, but if τ is large enough this does not have much impact on performance. Experimental and theoretical research has shown that taking $\tau \geq 5K$ is sufficient.



Soft-decision (Soft-input) Decoding

The decoder described above is referred to as a hard-decision (or hard-input) decoder because decoding is based on the received data at the output of the receiver's decision device (i.e., the BSC output). An advantageous alternative is soft-decision (or soft-input) decoding, which is based on the received data at the output of the receiver's matched filter for an AWGN channel.

Example: Comparison of hard- and soft- decision decoding.

Consider the use of a double-repetition block code to transmit a single bit over an AWGN channel with BPSK.

For a message word of $\underline{a} = 0$, the code word is $\underline{c} = 00$. The two code bits are sent sequentially over the channel, and the output of the receiver's matched filter is

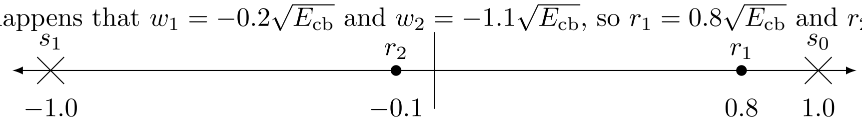
$$r_1 = \sqrt{E_{\text{cb}}} + w_1$$

after the first received bit, and

$$r_2 = \sqrt{E_{\text{cb}}} + w_2$$

after the second bit, where w_1 and w_2 are independent Gaussian noise samples.

Suppose it just so happens that $w_1 = -0.2\sqrt{E_{\text{cb}}}$ and $w_2 = -1.1\sqrt{E_{\text{cb}}}$, so $r_1 = 0.8\sqrt{E_{\text{cb}}}$ and $r_2 = -0.1\sqrt{E_{\text{cb}}}$.



The (hard) output of the decision device would be $\underline{r} = 01$, which is the (hard) input to the decoder. Although an error is detected, the decoder is unable to decide whether the message was a 0 or a 1.

However, if (soft) output of the matched filter is passed directly to the decoder (as soft-input), it can decide that a 00 was transmitted since the noise would have to be much larger if 11 was transmitted.

ie. (r_1, r_2) is closer to $(\sqrt{E_{\text{cb}}}, \sqrt{E_{\text{cb}}})$ than to $(-\sqrt{E_{\text{cb}}}, -\sqrt{E_{\text{cb}}})$.

$$[(0.8) - (1)]^2 + [(-0.1) - (1)]^2 = 1.25 \qquad < \qquad 4.05 = [(0.8) - (-1)]^2 + [(-0.1) - (-1)]^2$$

The same idea can be used for decoding convolutional codes with the Viterbi algorithm. Let

$$r_i^{(j)} = \sqrt{E_{\text{cb}}}\left(1 - 2c_i^{(j)}\right) + w_i^{(j)}$$

be the sampled output of the receiver's matched filter for the j^{th} bit of the i^{th} code symbol, where $w_i^{(j)}$ is a Gaussian noise sample with zero mean and variance $\mathcal{N}_0/2$. Therefore

$$f\left(r_i^{(j)} \mid c_i^{(j)}\right) = \frac{1}{\sqrt{\pi \mathcal{N}_0}} \exp\left\{-\frac{1}{\mathcal{N}_0}\left|r_i^{(j)} - \sqrt{E_{\text{cb}}}\left(1-2c_i^{(j)}\right)\right|^2\right\} .$$

The complete received sample sequence is

$$\underline{r} = r_1^{(1)} r_1^{(2)} \cdots r_1^{(n)} \;\; r_2^{(1)} r_2^{(2)} \cdots r_2^{(n)} \;\; \ldots \;\; r_{N_c}^{(1)} r_{N_c}^{(2)} \cdots r_{N_c}^{(n)} \;\; ,$$

and

$$f(\underline{r} \mid \underline{c}) = \prod_{i=1}^{N_c}\prod_{j=1}^{n} f\left(r_i^{(j)} \mid c_i^{(j)}\right) = \frac{1}{\left(\sqrt{\pi \mathcal{N}_0}\right)^{nN_c}} \exp\left\{-\frac{1}{\mathcal{N}_0}\sum_{i=1}^{N_c}\sum_{j=1}^{n}\left|r_i^{(j)} - \sqrt{E_{\text{cb}}}\left(1-2c_i^{(j)}\right)\right|^2\right\} .$$

The MLSE decoder must find the path through the trellis which maximizes the likelihood function $f(\underline{r} \mid \underline{c})$, or equivalently, maximizes the log-likelihood function $\log f(\underline{r} \mid \underline{c})$. However,

$$\begin{aligned}\log f(\underline{r} \mid \underline{c}) &= \sum_{i=1}^{N_c}\sum_{j=1}^{n} \log f\left(r_i^{(j)} \mid c_i^{(j)}\right) \\ &= \sum_{i=1}^{N_c}\sum_{j=1}^{n}\left[-\frac{1}{\mathcal{N}_0}\left|r_i^{(j)} - \sqrt{E_{\text{cb}}}\left(1-2c_i^{(j)}\right)\right|^2\right] - \left(\sqrt{\pi \mathcal{N}_0}\right)^{nN_c} \;\; ,\end{aligned}$$

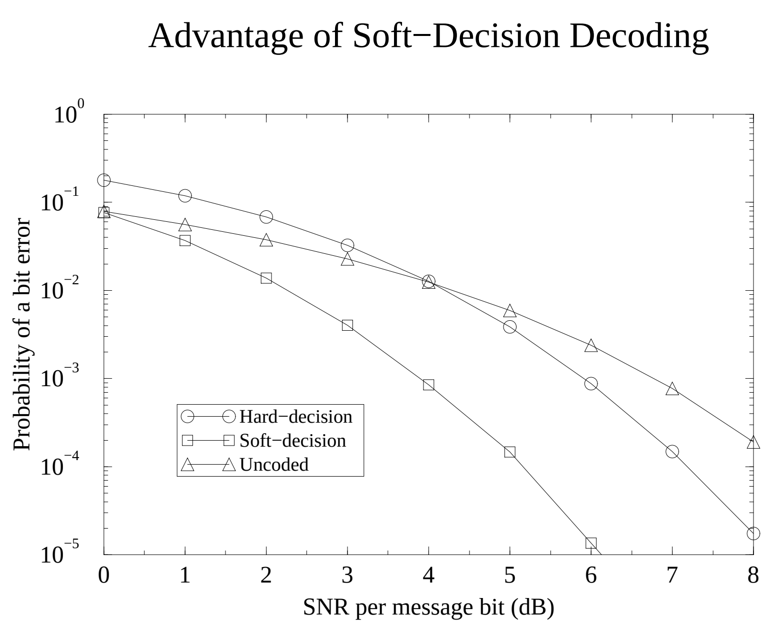
so maximizing the log-likelihood function is equivalent to minimizing the path metric

$$\mathrm{M}(\underline{r} \mid \underline{c}) = \sum_{i=1}^{N_c}\sum_{j=1}^{n}\left|r_i^{(j)} - \sqrt{E_{\text{cb}}}\left(1-2c_i^{(j)}\right)\right|^2 = \sum_{i=1}^{N_c}\sum_{j=1}^{n}\mu(r_i^{(j)} \mid c_i^{(j)}) = \sum_{i=1}^{N_c}\mu(r_i \mid c_i) \; ,$$

where $\mu(r_i^{(j)} \mid c_i^{(j)}) = \left|r_i^{(j)} - \sqrt{E_{\text{cb}}}\left(1-2c_i^{(j)}\right)\right|^2$ is referred to as the *bit metric*, and $\mu(r_i \mid c_i) = \sum_{j=1}^{n}\mu(r_i^{(j)} \mid c_i^{(j)})$ is the branch metric.

Thus, for soft-decision decoding, the decoder must find the path through the trellis with the minimum Euclidean distance between the soft-input $\underline{r}$ and $\sqrt{E_{\text{cb}}}\,(1-2\underline{c})$. This differs from hard-decision decoding which must find the path with the minimum Hamming distance between the hard-input $\underline{r}$ and the code word $\underline{c}$.

For soft-decision decoding, the Viterbi algorithm is implemented as described above for hard-decision decoding, but the branch metrics described here are used instead. Again, the decoder is optimal in that it finds the path with the smallest path metric, but in this case the maximum likelihood path has the smallest Euclidean distance from the received samples.



Performance Analysis of Convolutional Codes

The error correcting capability of a convolutional code is a function of the Hamming distances between codewords. If the codewords of code $\mathcal{C}_1$ are further apart than the codewords of code $\mathcal{C}_2$, then code $\mathcal{C}_1$ will be able to correct more errors.

Fact: Convolutional codes are linear.

Example: For the rate 1/2 convolutional code generated by $\underline{g}_1 = 110$ and $\underline{g}_2 = 101$, the code bits are related to the message bits by

$$\begin{aligned} c_i^{(1)} &= a_i \oplus s_i^{(1)} = a_i \oplus a_{i-1} \\ c_i^{(2)} &= a_i \oplus s_i^{(2)} = a_i \oplus a_{i-2} \end{aligned}$$

This code could also be generated by multiplying the message sequence $[a_1\ a_2\ a_3\ a_4\ \cdots]$ by the generator martrix

$$\underline{\underline{G}} = \begin{bmatrix} 1\,1\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0 & \\ 0\,0\,1\,1\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0 & \\ 0\,0\,0\,0\,1\,1\,1\,0\,0\,1\,0\,0\,0\,0 & \\ 0\,0\,0\,0\,0\,0\,1\,1\,1\,0\,0\,1\,0\,0 & \cdots \\ 0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,0\,1\,1 & \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,0 & \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1 & \\ \vdots & \ddots \end{bmatrix}$$

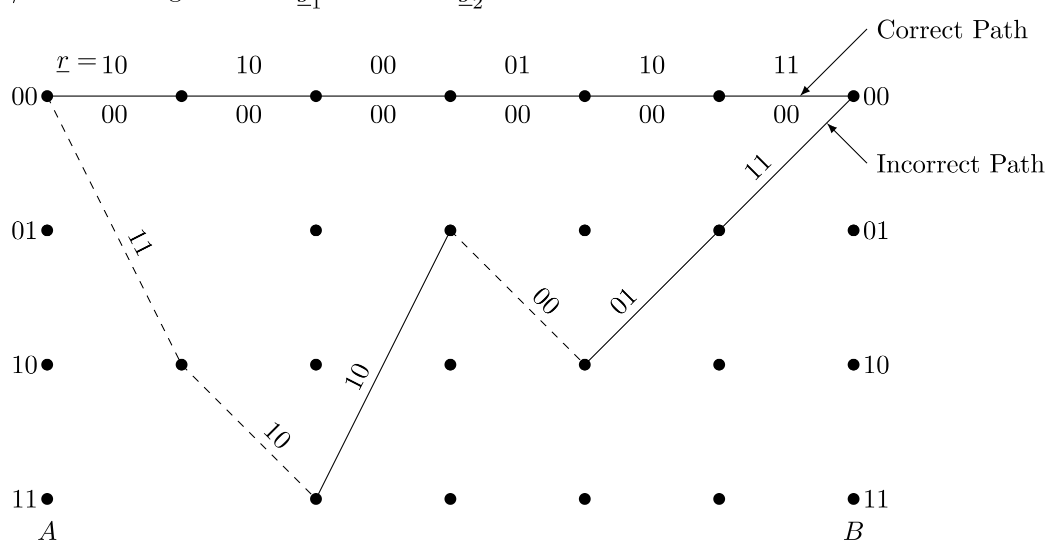
Note: Since the code is linear, the distances between codeword $\underline{c}$ and all other codewords in $\mathcal{C}$ are the same for all $\underline{c} \in \mathcal{C}$.

i.e. The probability of a bit error is not a function of the transmitted codeword.

Therefore, for the purpose of error analysis, we can assume that the all-zero codeword was transmitted.

Errors occur when the correct path is pruned in favour of an incorrect path. If the partial path metric of the correct path is greater than the partial path metric of another path, the correct path is discarded and errors will occur. Consider a path which first separates from the correct path at time A, and merges with the correct path at time B.

Example: Rate 1/2 code with generators $\underline{g}_1 = 101$ and $\underline{g}_2 = 111$.



The partial path metric from A to B for the correct path is 6 and the partial path metric for the incorrect path is 5, so in this case the incorrect path would be chosen.

An expression for the probability that this incorrect path with be selected by the decoder instead of the correct path can be found.

- Suppose the two paths differ in d bit positions.
- Let $N_{1,d}$ be the number of errors that occur in those positions where the two paths have the same code bits, and let $N_{2,d}$ be the number of errors that occur in those positions where the two paths have different code bits.
- The partial path metric for the correct path is $N_{1,d} + N_{2,d}$.
- The partial path metric for the incorrect path is $N_{1,d} + d - N_{2,d}$.
- The correct path will be selected if $N_{1,d} + N_{2,d} < N_{1,d} + d - N_{2,d}$, which simplifies to $N_{2,d} < d/2$.

Note: The value of $N_{1,d}$ is not relevant to the decision between these two paths.

In greater detail, there are three cases:
- If $N_{2,d} < d/2$ then the correct path will be chosen.
- If $N_{2,d} = d/2$ then a tie occurs and one of the two paths is selected randomly.
- if $N_{2,d} > d/2$ then the incorrect path will be chosen.

Note: Since $N_{2,d}$ is a binomial random variable with parameters (d, p), where p is the channel crossover probability, the probability of getting $N_{2,d} = l$ errors out of d bits is

$$\Pr\{N_{2,d} = l\} = \binom{d}{l} p^l (1-p)^{d-l} .$$

If d is even, then the probability of selecting the incorrect path is

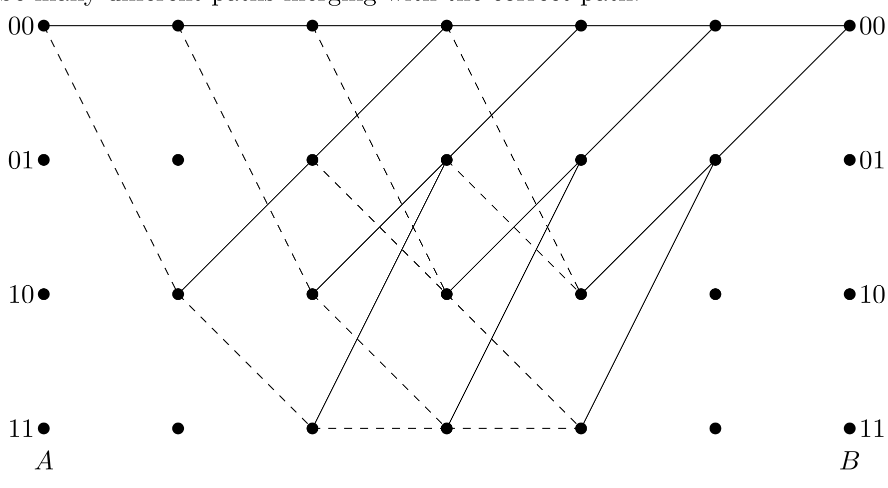
$$P(d) = \sum_{l=d/2+1}^{d} \Pr\{N_{2,d} = l\} + \frac{1}{2}\Pr\{N_{2,d} = d/2\}$$

where the second term is due to the possibility of a tie.
If d is odd, then a tie is impossible, so the probability of selecting the incorrect path is

$$P(d) = \sum_{l=(d+1)/2}^{d} \Pr\{N_{2,d} = l\}$$

First-event error probability

At time B there will be many different paths merging with the correct path.



The first-event error probability, P_e, is the probability that that correct path is pruned at time B. This occurs if the metric for the correct path is greater than the metric for any of the other paths. Let M_0 be the partial path metric of the correct path and M_i be the partial path metric of the i^{th} incorrect path. The first-event error probability is

$$\begin{aligned} P_e &= \Pr\left\{\bigcup_i (M_0 > M_i)\right\} \\ &= \Pr\left\{\bigcup_i (N_{2,d_i} \geq d_i/2)\right\} \end{aligned}$$

Using the union bound gives

$$P_e < \sum_i \Pr\{N_{2,d_i} \geq d_i/2\} = \sum_i P(d_i)$$

By grouping paths with the same distance, d, this can be written as

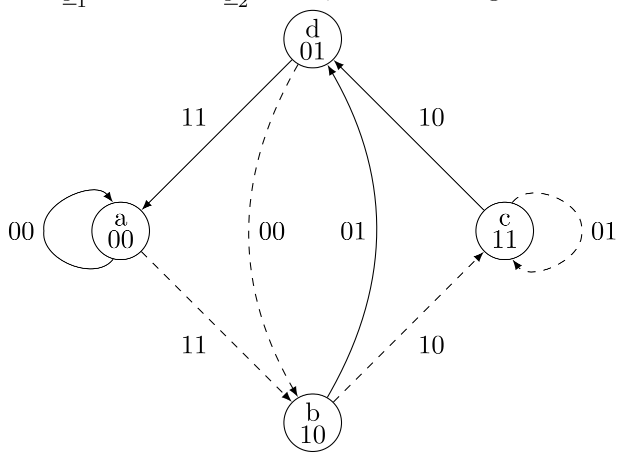
$$P_e < \sum_{d=1}^{\infty} a_d P(d)$$

where a_d is the number of paths which differ from the correct path by d bits and merge with the correct path at time B. The set $\{a_d\}$ is the *codeword weight distribution* of the code.

Transfer function

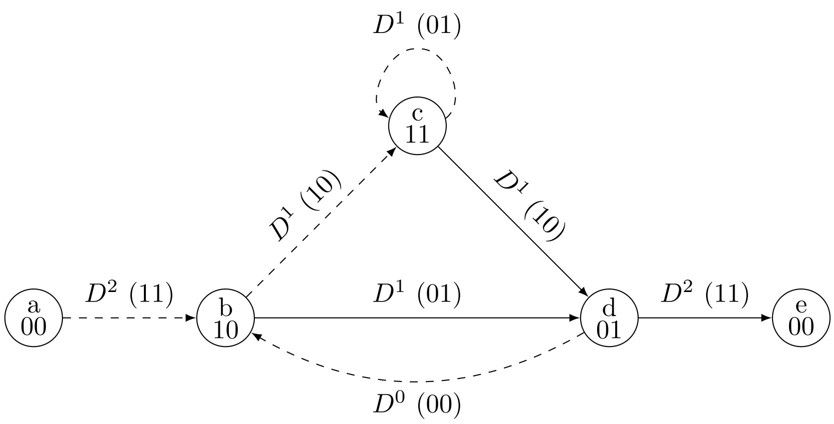
The weight distribution of the code can be found from the state diagram, by examining all paths that start and end in the zero state.

Example: For the code with generators $\underline{g}_1 = 101$ and $\underline{g}_2 = 111$, the state diagram is:



The weight distribution can easily be determined from the *transfer function* of the code. To determine the transfer function, the space diagram is modified slightly:

- The self-loop at the zero state is removed because it does not add anything to the weight of the codeword.
- The zero-state is split into two states.
- A value of D^m is assigned to each branch, where m is the Hamming weight of the code symbol generated by that branch.



The state equations for this diagram are:

$$\begin{aligned} X_b &= D^2 X_a + X_d \\ X_c &= D X_b + D X_c \\ X_d &= D X_b + D X_c \\ X_e &= D^2 X_d \end{aligned}$$

Solving these equations for X_e in terms of X_a gives

$$X_e = \frac{D^5}{1-2D} X_a$$

The transfer function is defined as

$$T(D) = \frac{X_e}{X_a} = \frac{D^5}{1-2D}$$

Making use of the identity

$$\frac{1}{1-\alpha} = \sum_{i=0}^{\infty} \alpha^i$$

yields

$$\begin{aligned} T(D) &= D^5 \left[1 + (2D) + (2D)^2 + (2D)^3 + \cdots\right] \\ &= D^5 + 2D^6 + 4D^7 + 8D^8 + \cdots \\ &= \sum_{d=5}^{\infty} 2^{d-5} D^d \\ &= \sum_{d=5}^{\infty} a_d D^d \end{aligned}$$

where $a_d = 2^{d-5}$ is the number of codewords of weight d. Thus, there is one path segment of weigth 5, two path segments of weight 6, four path segments of weight 7, and so on.

Def^n: The weight minumum weight path segment that merges with the all-zero path is called the *minimum free distance*, d_{free}, of the code. In the above example, $d_{\text{free}} = 5$.

The first-event error probability is then bounded by

$$P_e < \sum_{d=d_{\text{free}}}^{\infty} a_d P(d)$$

A crude approximation is

$$P_e \cong a_{d_{\text{free}}} P(d_{\text{free}})$$

In general, a code with a larger minimum free distance than another code will give better performance.

Probability of a bit error

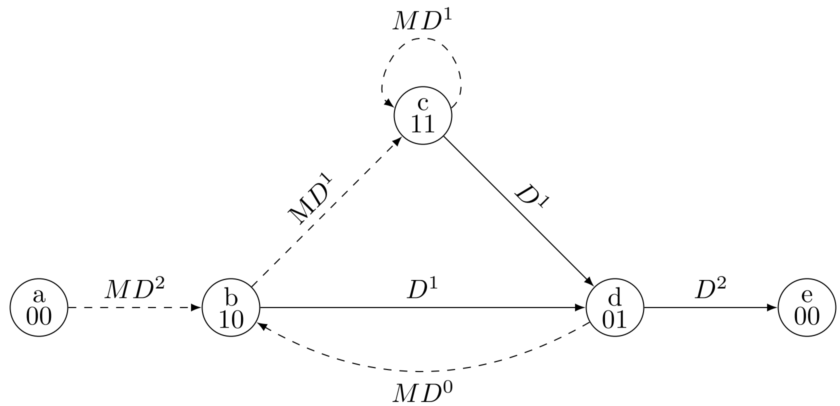
A bound for the probability of a bit error can also be found. Suppose an incorrect path that differs from the correct path in m message bit positions is selected instead of the correct path. This event case will cause m message bits to be in error. Using the bounding technique described previously, the probability of a bit error is bounded by

$$P_b < \sum_{d_{\text{free}}}^{\infty} \sum_{m=1}^{\infty} a_{d,m} m P(d)$$

where $a_{d,m}$ is the number of codewords of weight d corresponding to message words of weight m.

Input/ouput weight enumeration function

The input/output weight distribution can be calculated from the state diagram for the code. In addition to labelling each branch by D^i, where i is the weight of the code symbol, each branch is also labelled with M^m, where m is the weight of the corresponding message symbol.



The transfer function is the solution to the following equations:

$$X_b = MD^2X_a + MX_d$$

$$X_c = MDX_b + MDX_c$$

$$X_d = DX_b + DX_c$$

$$X_e = D^2X_d$$

Solving these equations for X_e in terms of X_a gives

$$X_e = \frac{MD^5}{1-2MD}X_a$$

The transfer function is

$$T(D,M) = \frac{X_e}{X_a} = \frac{MD^5}{1-2MD}$$

Making use of the identity

$$\frac{1}{1-\alpha} = \sum_{i=0}^{\infty}\alpha^i$$

yields

$$\begin{aligned} T(D,M) &= MD^5\left[1+(2MD)+(2MD)^2+(2MD)^3+\cdots\right] \\ &= MD^5 + 2M^2D^6 + 4M^3D^7 + 8M^4D^8 + \cdots \\ &= \sum_{d=d_{\text{free}}}^{\infty} 2^{d-5}M^{d-4}D^d \\ &= \sum_{d=d_{\text{free}}}^{\infty}\sum_{m=1}^{\infty} 2^{d-5}\delta_{m-(d-4)}M^mD^d \\ &= \sum_{d=d_{\text{free}}}^{\infty}\sum_{m=1}^{\infty} a_{d,m}M^mD^d \end{aligned}$$

where $a_{d,m} = 2^{d-5}\delta_{m-(d-4)}$ is the number of codewords of weight d corresponding to message words of weight m. That is, there is one codeword of weight 5 corresponding to a message word of weight 1, but no other codewords of weight 5 corresponding to message words of other weights. There are two code words of weight 6, and they both correspond to message words of weight 2.

Trellis Coded Modulation (TCM)

Trellis coded modulation involves the combination of convolutional coding with spectrally efficient M-ary modulation.

Objectives: To improve system performance without

- increasing the transmitted energy per bit
- reducing the message transmission rate
- increasing the bandwidth

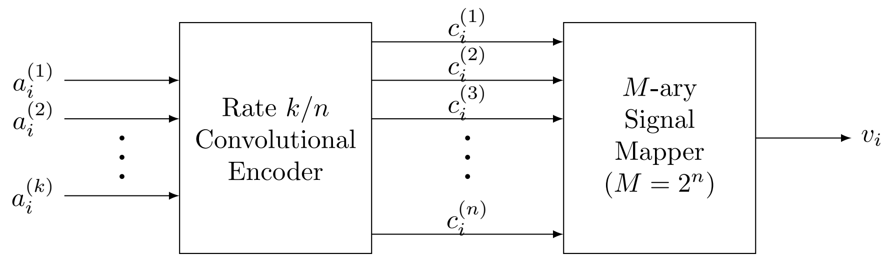
Example: rate 1/3 convolutional code, sending one code bit at a time with BPSK

- can improve performance over uncoded BPSK without increasing $\mathcal{E}_{\mathrm{mb}}$.
- but, we must either slow down the message transmission rate or increase the code bit transmission rate (and therefore, increase bandwidth) to accommodate the additional parity bits

Example: rate 1/3 convolutional code, sending each code symbol using 8-PSK

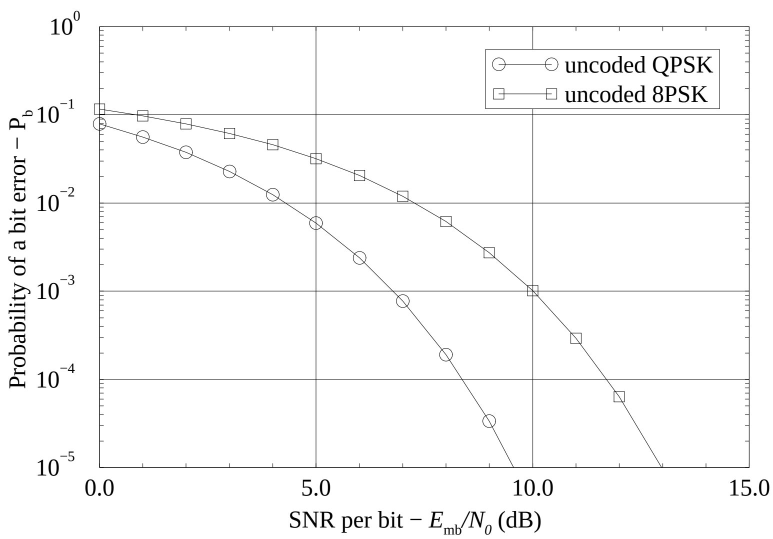
- three code bits are sent with each channel use
- one message bit is transmitted with each channel use
- there is no need to increase the bandwidth, and the message bit transmission rate is the same as for uncoded BPSK.

In general, trellis coded modulation involves the use of a rate k/n convolutional code and an M-ary signalling scheme, where $M = 2^n$. Each code symbol is transmitted in a single use of the channel.



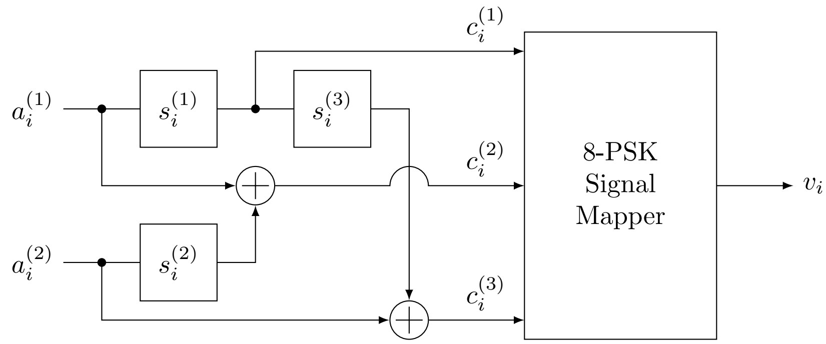
Trellis coded modulation with a rate k/n convolutional code must yield better performance than uncoded modulation with $M = 2^k$. However, increasing M is known to reduce performance substantially, so the coding gain of the convolutional code must be large.

Example: Uncoded QPSK vs. uncoded 8-PSK

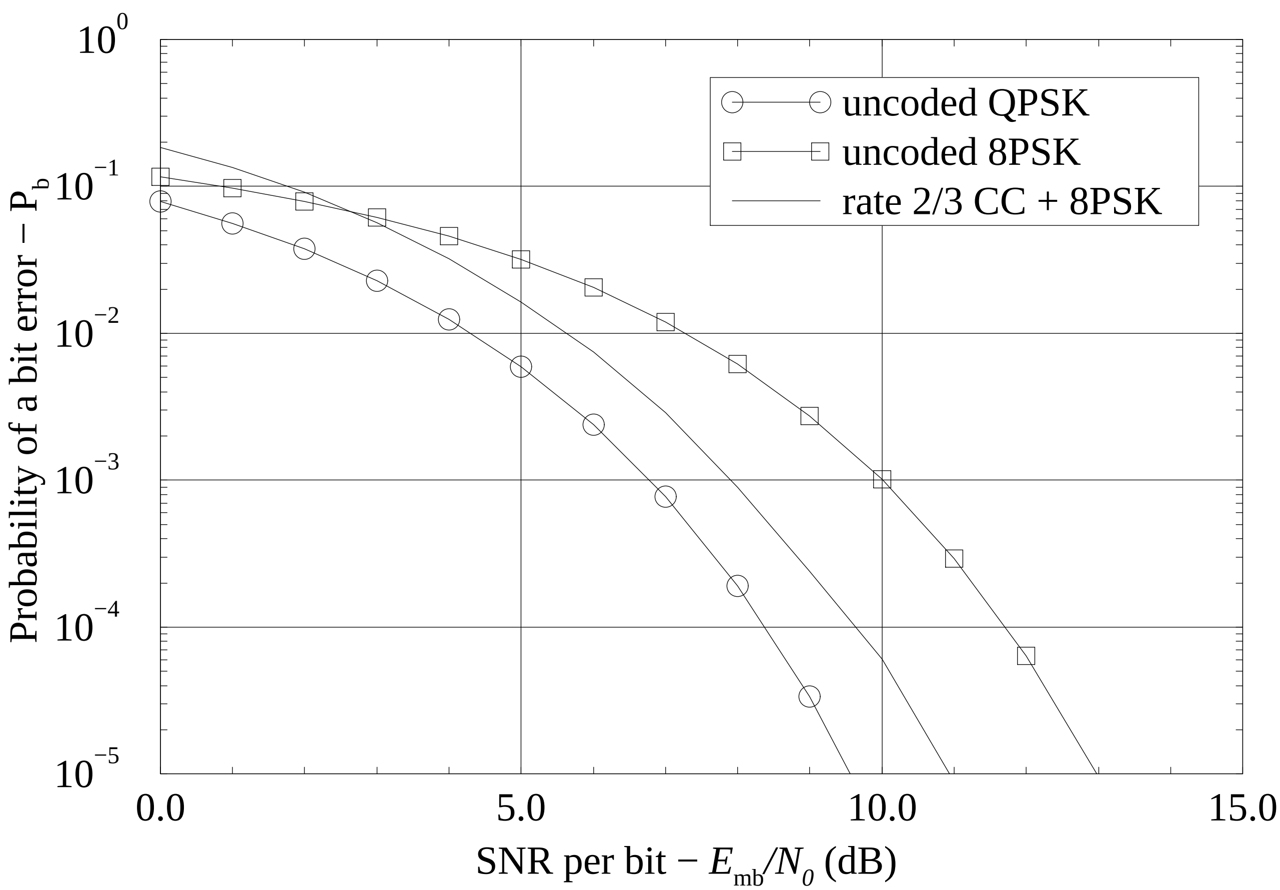


If a rate 2/3 convolutional code is used with 8-PSK, the code must be able to improve the performance of uncoded 8-PSK by at least 3.5 dB.

Example: Consider the following rate 2/3 convolutional code used with Gray-mapped 8-PSK.



The performance of this code/modulation scheme is



Note: In this case, the performance of the coded scheme is worse than uncoded QPSK. There is a clear disadvantage to using coding in this case. The coding gain does not offset the degradation due to the higher-order modulation scheme.

The degradation could be overcome be using a more powerful code (longer constraint length), but this solution involves increasing the complexity of the decoder (because the number of states has increased).

To effectively use M-ary modulation with a convolutional code, care must be taken when assigning code symbols to points in the signal constellation. The goal is to maximize the Euclidean distance between symbol sequences. This can be accomplished through *set partioning.*

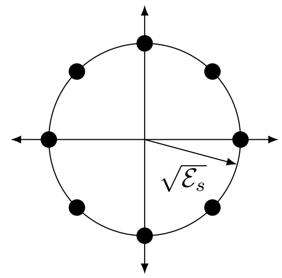
Set Partitioning:

Set partitioning involves dividing the signal constellation into subsets with larger minimum distance between points in the subset than the minimum distance between points in the original constellation.

Example: 8-PSK

The signal constellation for 8-PSK is

A

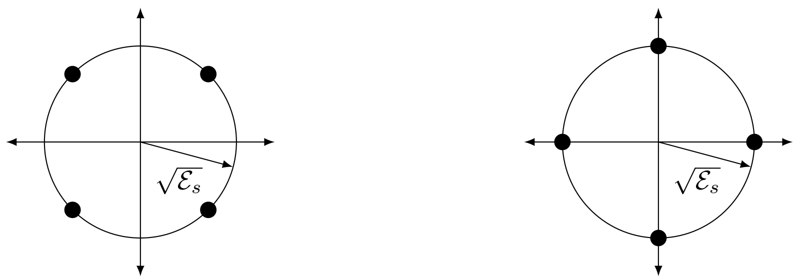


The minimum distance between two points in set A (the original 8-PSK constellation) is

$$d_0 = \sqrt{(2-\sqrt{2})\mathcal{E}_s}\ .$$

Divide the constellation into two subsets, B_0 and B_1 as shown below

B_0 B_1



The minimum distance between two points in subset B_0 is

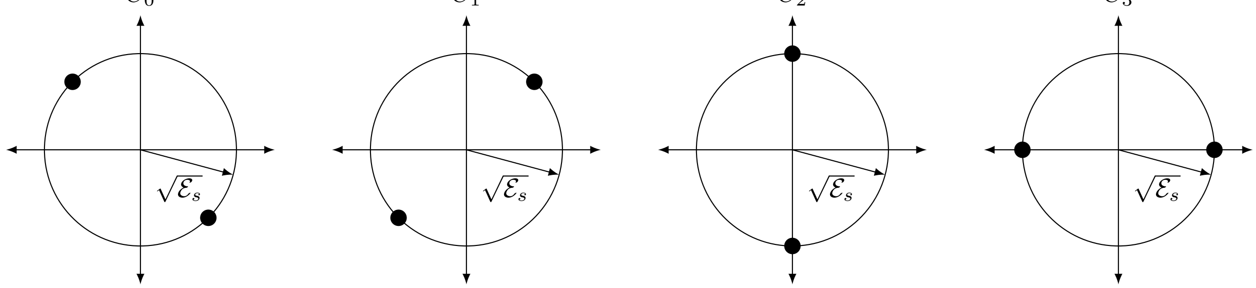
$$d_1 = \sqrt{2\mathcal{E}_s}$$

The minimum distance between two points in subset B_1 is also

$$d_1 = \sqrt{2\mathcal{E}_s}$$

Divide subset B_0 into two subsets, C_0 and C_1, and divide subset B_1 into two subsets, C_2 and C_3, as shown below

C_0 C_1 C_2 C_3



The minimum distance between two points in subset C_i is

$$d_2 = 2\sqrt{\mathcal{E}_s}$$

for all $i \in \{0, 1, 2, 3\}$.

At each level of subdivision, the minimum distance is greater than the minimum distance at the previous level.

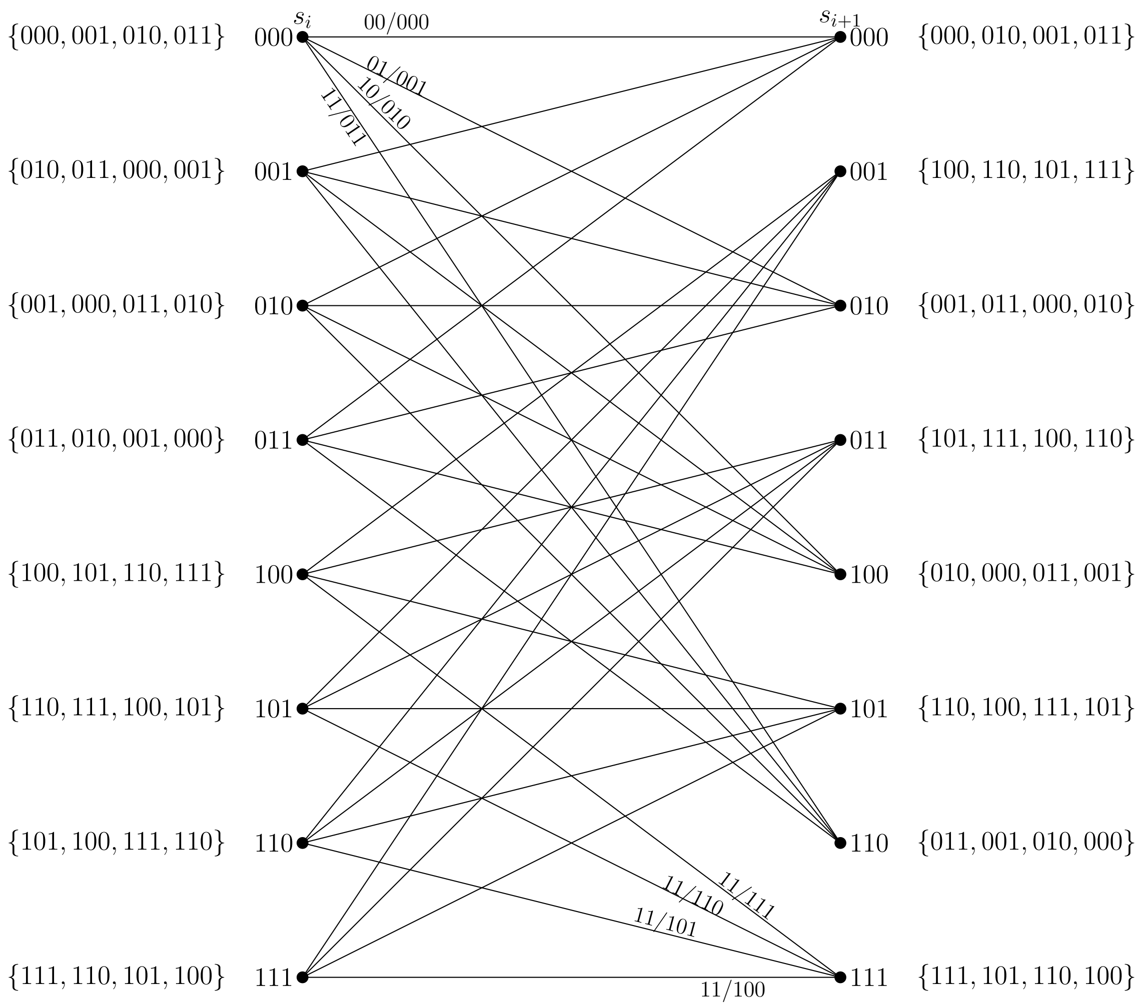
This same procedure can also be used with higher-order modulation schemes. See, for example, Figure 8.12-2 in Proakis.

For trellis-coded modulation to be effective, the following rule must be obeyed:

The code symbols for transitions originating from and merging into any state must be assigned to points in the same subset.

Example: Rate 2/3 convolutional code with 8-PSK

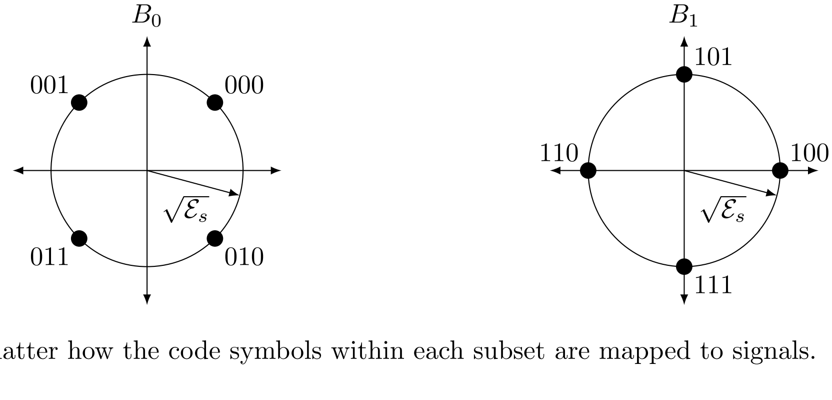
For the rate 2/3 convolutional code described by the block diagram given above, one section of the trellis is



As can be seen, in this example the code symbols fall into two groups, {000, 001, 010, 011} and {100, 101, 110, 111}. Code symbols in the first group should be mapped into signals in subset B_0, and code symbols in the second group should be mapped into signals in subset B_1.

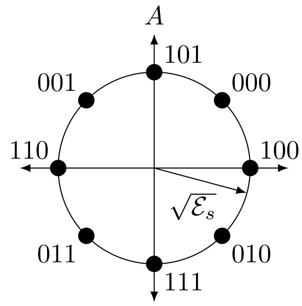
Note: It does not matter which group of code symbols is mapped to which subset. You could also map the first group to subset B_1 and the second group to subset B_0.

One possible mapping of code symbols within each subset is

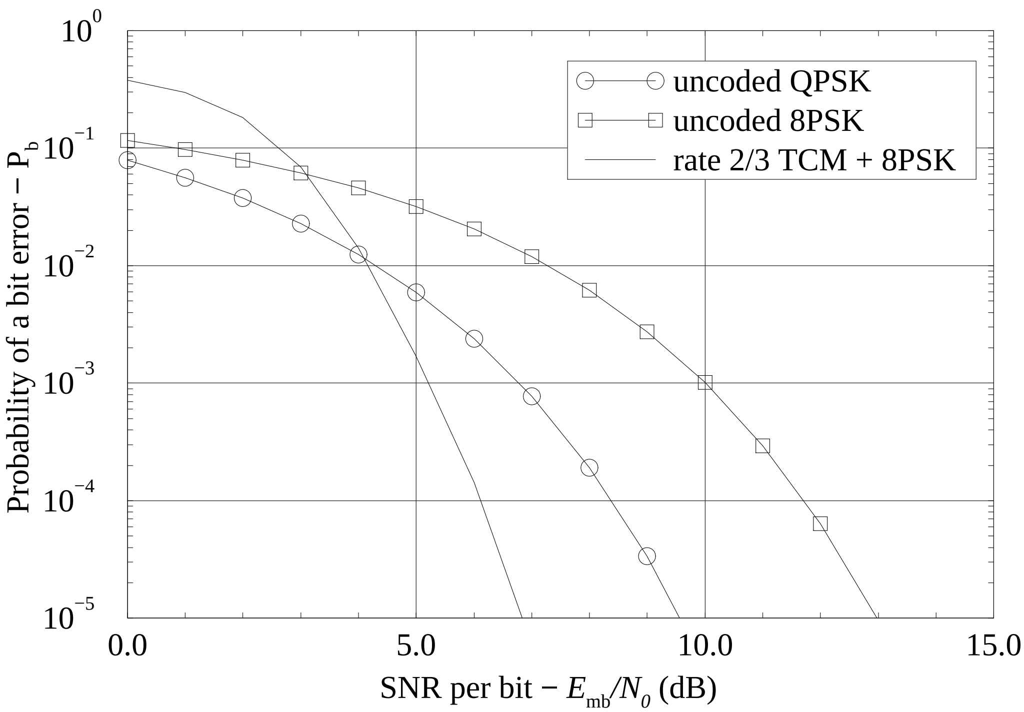


Note: It does not matter how the code symbols within each subset are mapped to signals.

The combined signal consellation is



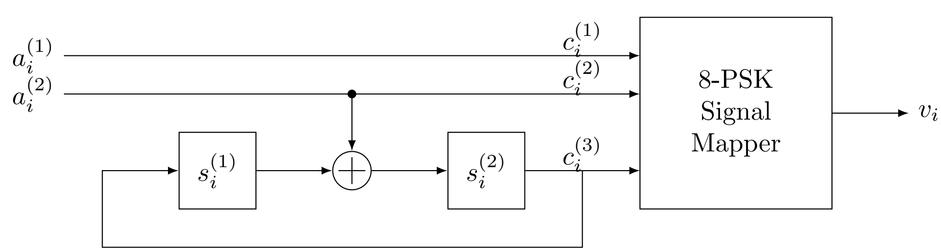
When this mapping of code symbols to signal points is used, the performance of the coding/modulation scheme is much better.

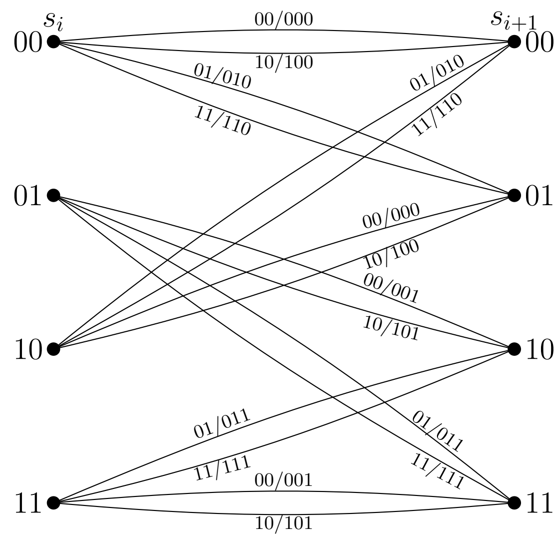


Parallel Transitions:

For some codes, particularly systematic codes, it is possible for parallel transitions to occur. Parallel transistions occur when one (or more) message bits are not encoded by the convolutional code

Example: rate 2/3 convolutional code with parallel transitions





If parallel transitions occur in the trellis diagram, the following rule must also be obeyed when mapping code symbols to signal points

> The code symbols for parallel transitions must be assigned to signal points with maximum Euclidean separation.

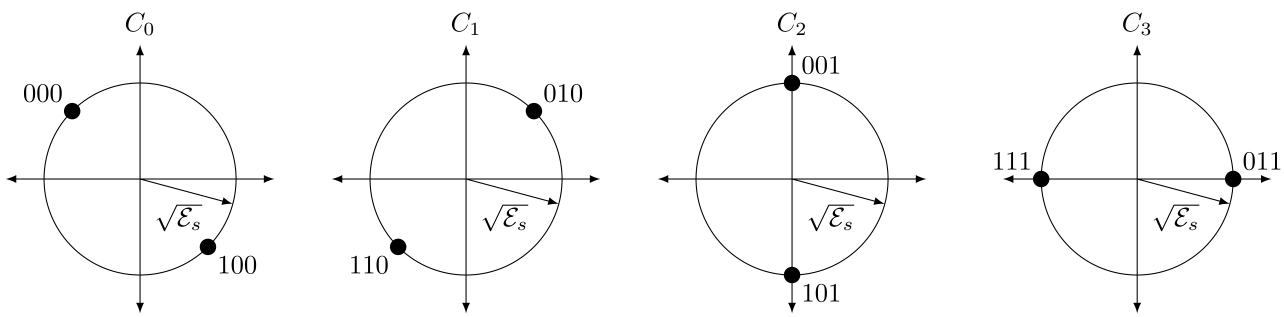
The code symbols for the parallel transitions are $\{000, 100\}$, $\{010, 110\}$, $\{001, 101\}$, and $\{011, 111\}$.

Also, the code symbols orginating from or merging into any state fall in two groups, $\{000, 100, 010, 110\}$, which originate from states 00 and 10 and merge in states 00 and 01, and $\{001, 101, 011, 111\}$, which originate from states 01 and 11 and merge in states 10 and 11.

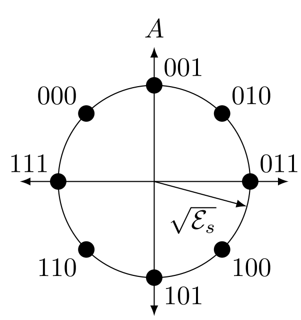
We should assign the code symbols in $\{000, 100, 010, 110\}$ to signals in subset B_0, and the code symbols in $\{001, 101, 011, 111\}$ should be assigned to signals in subset B_1.

Also, to make sure code symbols in parallel transitions are assigned to signals as far apart as possible, we should assign the code symbols in $\{000, 100\}$ to subset C_0, the code symbols in $\{010, 110\}$ to subset C_1, the code symbols in $\{001, 101\}$ to subset C_2, and the code symbols in $\{011, 111\}$ to subset C_3.

Since is does not matter how we assign the symbols within each subset, we can assign the symbols as follows:



The signal constellation showing all the mappings is



Selected Mathematical Tables

Trigonometric Identities

$$\sin(A \pm B) = \sin A \cos B \pm \cos A \sin B$$
$$\cos(A \pm B) = \cos A \cos B \mp \sin A \sin B$$
$$\sin A \sin B = \tfrac{1}{2}[\cos(A-B) - \cos(A+B)]$$
$$\cos A \cos B = \tfrac{1}{2}[\cos(A-B) + \cos(A+B)]$$
$$\sin A \cos B = \tfrac{1}{2}[\sin(A+B) + \sin(A-B)]$$
$$\cos A \sin B = \tfrac{1}{2}[\sin(A+B) - \sin(A-B)]$$
$$\sin 2A = 2 \sin A \cos A$$
$$\cos 2A = 2\cos^2 A - 1 = 1 - 2\sin^2 A = \cos^2 A - \sin^2 A$$
$$\sin^2 A = \tfrac{1}{2}(1 - \cos 2A)$$
$$\cos^2 A = \tfrac{1}{2}(1 + \cos 2A)$$
$$\sin A = \frac{1}{j2}\left(e^{jA} - e^{-jA}\right)$$
$$\cos A = \frac{1}{2}\left(e^{jA} + e^{-jA}\right)$$
$$e^{\pm jA} = \cos A \pm j \sin A$$

Miscellaneous Identities

$$\sum_{m=-\infty}^{\infty} e^{-j2\pi f m T} = \frac{1}{T}\sum_{n=-\infty}^{\infty} \delta(f - \frac{n}{T})$$

Definite Integrals

$$\int_0^\infty \frac{\sin ax}{x}\,dx = \begin{cases} \pi/2 & a > 0 \\ 0 & a = 0 \\ -\pi/2 & a < 0 \end{cases}$$
$$\int_0^x \frac{\sin au}{u}\,du = \mathrm{Si}(x)$$
$$\int_0^\infty \frac{\sin^2 ax}{x^2}\,dx = |a|\pi/2$$
$$\int_0^\infty e^{-ax^2}\,dx = \tfrac{1}{2}\sqrt{\pi/a}$$
$$\int_0^\infty x e^{-ax^2}\,dx = \frac{1}{2a}$$
$$\int_0^\infty x^2 e^{-ax^2}\,dx = \frac{1}{4a}\sqrt{\pi/a}$$
$$\int_0^x \frac{2}{\sqrt{\pi}} e^{-u^2}\,du = \mathrm{erf}\,(x)$$
$$\int_{-\infty}^\infty e^{j2\pi f t}\,dt = \delta(f)$$

Indefinite Integrals

$$\int \sin(ax+b)\,dx = -\frac{1}{a}\cos(ax+b)$$
$$\int \cos(ax+b)\,dx = \frac{1}{a}\sin(ax+b)$$
$$\int \sin^2 ax\,dx = \frac{x}{2} - \frac{\sin 2ax}{4a}$$
$$\int \cos^2 ax\,dx = \frac{x}{2} + \frac{\sin 2ax}{4a}$$
$$\int \sin ax \cos ax\,dx = \frac{1}{2a}\sin^2 ax$$
$$\int \sin ax \sin bx\,dx = \frac{\sin(a-b)x}{2(a-b)} - \frac{\sin(a+b)x}{2(a+b)}$$
$$\int \cos ax \cos bx\,dx = \frac{\sin(a-b)x}{2(a-b)} + \frac{\sin(a+b)x}{2(a+b)}$$
$$\int \sin ax \cos bx\,dx = -\frac{\cos(a-b)x}{2(a-b)} - \frac{\cos(a+b)x}{2(a+b)}$$
$$\int \cos ax \sin bx\,dx = \frac{\cos(a-b)x}{2(a-b)} - \frac{\cos(a+b)x}{2(a+b)}$$
$$\int x \sin ax\,dx = \frac{1}{a^2}(\sin ax - ax\cos ax)$$
$$\int x \cos ax\,dx = \frac{1}{a^2}(\cos ax + ax\sin ax)$$
$$\int x^2 \sin ax\,dx = \frac{1}{a^3}(2ax\sin ax + 2\cos ax - a^2x^2\cos ax)$$
$$\int x^2 \cos ax\,dx = \frac{1}{a^3}(2ax\cos ax - 2\sin ax + a^2x^2\sin ax)$$
$$\int e^{ax}\,dx = \frac{1}{a}e^{ax}$$
$$\int x e^{ax}\,dx = \frac{1}{a^2}e^{ax}(ax-1)$$
$$\int x^2 e^{ax}\,dx = \frac{1}{a^3}e^{ax}(a^2x^2 - 2ax + 2)$$
$$\int e^{ax}\sin bx\,dx = \frac{1}{a^2+b^2}e^{ax}(a\sin bx - b\cos bx)$$
$$\int e^{ax}\cos bx\,dx = \frac{1}{a^2+b^2}e^{ax}(a\cos bx + b\sin bx)$$
$$\int \left[\frac{\sin ax}{x}\right]^2 dx = a\int \frac{\sin 2ax}{x}\,dx - \frac{\sin^2 ax}{x}$$
$$\int \ln x\,dx = x\ln x - x$$

Properties of the Fourier Transform

Operation	$h(t)$	$H(f)$
Linearity	$a_1 h_1(t) + a_2 h_2(t)$	$a_1 H_1(f) + a_2 H_2(f)$
Complex conjugate	$h^*(t)$	$H^*(-f)$
Scaling	$h(\alpha t)$	$\frac{1}{\lvert\alpha\rvert} H(\frac{f}{\lvert\alpha\rvert})$
Delay	$h(t - t_0)$	$H(f) e^{-j2\pi f t_0}$
Frequency translation	$h(t) e^{j2\pi f_0 t}$	$H(f - f_0)$
Amplitude modulation	$h(t)\cos(2\pi f_0 t)$	$\frac{1}{2} H(f - f_0) + \frac{1}{2} H(f + f_0)$
Time convolution	$\int_{-\infty}^{\infty} h_1(\tau) h_2(t - \tau)\, d\tau$	$H_1(f) H_2(f)$
Frequency convolution	$h_1(t) h_2(t)$	$\int_{-\infty}^{\infty} H_1(u) H_2(f - u)\, du$
Duality	$H(t)$	$h(-f)$
Time differentiation	$\frac{d}{dt} h(t)$	$j2\pi f H(f)$
Time integration	$\int_{-\infty}^{t} h(\tau)\, d\tau$	$\frac{1}{j2\pi f} H(f) + \frac{H(0)}{2} \delta(f)$

Some Fourier Transform Pairs

$$h(t) \to H(f)$$

$$e^{-at} u(t) \to \frac{1}{a + j2\pi f}$$

$$t e^{-at} u(t) \to \frac{1}{(a + j2\pi f)^2}$$

$$e^{-a|t|} \to \frac{2a}{a^2 + (2\pi f)^2}$$

$$e^{-t^2/(2\sigma^2)} \to \sqrt{2\pi\sigma^2} e^{-2\pi^2 f^2 \sigma^2}$$

$$u(t) \to \frac{1}{2}\delta(f) + \frac{1}{j2\pi f}$$

$$\delta(t - t_0) \to e^{-j2\pi f t_0}$$

$$\frac{\sin 2\pi W t}{2\pi W t} \to \frac{1}{2W} \mathrm{rect}(\frac{f}{2W})$$

$$\mathrm{rect}(\frac{t}{T}) \to T \frac{\sin \pi f T}{\pi f T}$$

$$\sum_{m=-\infty}^{\infty} \delta(t - mT) \to \frac{1}{T} \sum_{n=-\infty}^{\infty} \delta(f - \frac{n}{T})$$