Overview

- Additive white Gaussian noise assumption
  \[ r(t) = s(t) + n(t), \quad \text{where } n(t) \text{ is a Gaussian process.} \]

- Question: How to design a receiver to compensate such kind of noise?
  - 5.1 Simple analysis for memoryless modulated signals and modulated signals with memory
  - 5.2 Probability of error (receiver side)
5.1 Optimal receiver for signals corrupted by additive white Gaussian noise

**System view**

- **Transmitted signal** $s_{t,w}(t)$
- **Channel**
- **Received signal** $r_f(t) = s_{t,w}(t) + n(t)$
- **Noise** $n(t)$

**AWGN** : Additive White Gaussian Noise

$\Phi_\alpha(f) = \frac{1}{2} N_0 \text{ W/Hz}$

**Assumption**

For simplicity, assume that the baseband signal is real-valued.

$r_f(t) = s_{t,w}(t) + n(t)$

**Definition of optimality**

To estimate $m$ in order to minimize the error probability.
5.1 Optimal receiver for signals corrupted by additive white Gaussian noise

- **Model for analysis**
  - **Signal demodulator**
    - Vectorization $r_i(t) \rightarrow [r_1, r_2, \ldots, r_N]$.
  - **Detector**: minimize the probability of error in the above functional block
    estimator $[r_1, r_2, \ldots, r_N] = \hat{m}$

- **Realization of Signal demodulator**
  - **Signal correlators**
  - **Matched filters**
5.1.1 Correlation demodulator

- **Correlation demodulator**

\[ \int_{0}^{T} f(t) \, dt \]

- Vectorization of the received signal

Given the basis \( \{ f_{n} \}_{n=1}^{N} \),

\[ r_{j}(t) = \sum_{k} s_{nk} f_{k}(t) + \sum_{k} n_{k} f_{k}(t) + n'(t), \]

where \( s_{nk} = \int_{0}^{T} s_{k, m}(t) f_{k}(t) \, dt \) and \( n_{k} = \int_{0}^{T} n_{k}(t) f_{k}(t) \, dt \).

- \( \{ f_{1}(t), f_{2}(t), ..., f_{N}(t) \} \) span the channel symbol space.

- Note that \( \{ f_{n}(t) \} \) may not span the noise space, and hence, there may exist an extra term, \( n'(t) \). This term, however, is irrelevant to the decision error of the transmitted signal.

  - Alternative explanation is that \( n'(t) \) is irrelevant to decision error since decision is made only based on \( \{ r_{1}, r_{2}, ..., r_{N} \} \).
5.1.1 Correlation demodulator

Given the basis \( \{ f_n \}_{n=1}^N \),

\[
    r(t) = \sum_{k=1}^N s_{mk} f_k(t) + \sum_{k=1}^N n_k f_k(t) + n^*_k(t) = \sum_{k=1}^N f_k(t) + n_k(t),
\]

where \( r_k = s_{mk} + n_k \).
5.1.1 Correlation demodulator

\[ \text{Notes} \]
- If \( \{ f_n \} \) are not orthonormal, \( \{ n_1, n_2, \ldots, n_N \} \) become zero-mean dependent Gaussian noise.

**Statistical property of received signals** \( \{ r_1, r_2, \ldots, r_N \} \).

\[ r_k = s_{mk} + n_k \Rightarrow \{ r_k \}_{k=1}^N \text{ is uncorrelated Gaussian vector with mean } \{ s_{mk} \}_{k=1}^N. \]

\[
\begin{align*}
\hat{s}_1 &= [s_{11}, s_{12}, \ldots, s_{1N}] \
\hat{s}_2 &= [s_{21}, s_{22}, \ldots, s_{2N}] \
&\vdots \\
\hat{s}_M &= [s_{M1}, s_{M2}, \ldots, s_{MN}]
\end{align*}
\]

uncorrelated Gaussian with mean vector \( \bar{s}_1 \)
uncorrelated Gaussian with mean vector \( \bar{s}_2 \)
uncorrelated Gaussian with mean vector \( \bar{s}_M \)
5.1.1 Correlation demodulator

- **Maximum-likelihood decision maker based on Correlation Demodulator**

  \[ \text{MAP decision} = \arg \max_{i \in j, M} \Pr \left( \bar{s}_j \mid \bar{r}_1, \bar{r}_2, \ldots, \bar{r}_N \right) \]

  \[ = \arg \max_{i \in j, M} \frac{\Pr \left( \bar{s}_j \mid \bar{r}_1, \bar{r}_2, \ldots, \bar{r}_N \right)}{\Pr \left( \bar{r}_1, \bar{r}_2, \ldots, \bar{r}_N \right)} \]

  If equal prior, then \( \Pr \left( \bar{s}_j \right) = 1/M \) for every \( j \).

  and MAP decision maker becomes ML decision maker.

  \[ \text{ML decision} = \arg \max_{i \in j, M} \Pr \left( \bar{r}_1, \bar{r}_2, \ldots, \bar{r}_N \mid \bar{s}_j \right) \]

---

### Example 5.1-1. Baseband PAM signal with rectangular pulse shape.

channel symbol: \( s_m(t) = \text{Re}[A_m g(t) e^{j2\pi f_c t}] = A_m g(t) \cos(2\pi f_c t) \)

where \( A_m = (2m - 1 - M)d \), and \( m = 1, 2, \ldots, M \).

\[ \Rightarrow s_{i,m}(t) = A_m g(t). \]

\[ \Rightarrow \text{basis } f(t) = \begin{cases} \frac{1}{\sqrt{T}}, & 0 \leq t \leq T \\ 0, & \text{otherwise} \end{cases} \]

\[ \Rightarrow r_i = s_{i,m} + n_i, \text{ where } n \text{ is zero-mean Gaussian with variance } N_0/2, \]

and \( r_i = \frac{1}{\sqrt{T}} \int_0^T r_i(t) dt \) and \( s_{i,m} = \frac{1}{\sqrt{T}} \int_0^T S_{i,m}(t) dt \).

Correlation demodulator becomes a pure integrator.
5.1.1 Correlation demodulator

\[ s_{r,m} = \int_0^T \frac{1}{\sqrt{T}} A_0 g(t) \, dt = a A_n \sqrt{T}. \quad (g(t) = a \text{ for } 0 \leq t < T.) \]

\[ \Rightarrow \Pr(r_i \mid s_{r,m}) = \frac{1}{\sqrt{\pi N_0}} \exp \left[ - \frac{(r_i - s_{r,m})^2}{N_0} \right] \]

\[ \Rightarrow \text{ML decision} = \arg \max_{1 \leq j \leq M} \Pr \left\{ r_1, \ldots, r_N \mid s_j \right\} \]

\[ = \arg \min_{1 \leq j \leq M} (r_i - s_{r,m})^2. \]

--- Po-Ning Chen ---

5.1.1 Correlation demodulator

- In the following slides, we will assume the lowpass signals are always employed except otherwise stated. Hence, the “lengthy” subscript “l” will be omitted as the text did.

- To simplify the analysis, we will also assume the lowpass equivalent noise \( n_l(t) \) (now denoted by \( n(t) \) following the previous remark) is a white Gaussian noise, instead of a more practical bandlimited white Gaussian noise.
5.1.2 Matched-filter demodulator

- Linear filter demodulation (replacing the correlators)

\[ r_{\text{f}}(t) = \int_{-\infty}^{\tau} r(\tau) h_{\text{f}}(t-\tau) d\tau \]

Difference between correlators and filters

- Correlator:
  \[ r_{\text{c}}(t) = \int_{0}^{T} r(\tau) f_{\text{c}}(T-\tau) d\tau \]
  \( r(t) = 0 \) for \( t < 0 \)

- Filter:
  \[ r_{\text{f}}(t) = \int_{-\infty}^{\infty} r(\tau) h_{\text{f}}(t-\tau) d\tau = \int_{0}^{T} r(\tau) h_{\text{f}}(t-\tau) d\tau \]
  Then sampling at proper time, i.e., \( r_{\text{f}}(T) \).

Correlation-type demodulation is a special case of filter-type demodulation.

Let \( h_{\text{f}}(t) = f_{\text{c}}(T-t) \).

Then filter:
\[ r_{\text{f}}(t) = \int_{0}^{T} r(\tau) h_{\text{f}}(t-\tau) d\tau \]
\[ r_{\text{f}}(T) = \int_{0}^{T} r(\tau) f_{\text{c}}(T-T+\tau) d\tau \]
sampling at \( t = T \).

\[ \Rightarrow r_{\text{c}} = r_{\text{f}}(T) = \int_{0}^{T} r(\tau) f_{\text{c}}(\tau) d\tau \]
5.1.2 Matched-filter demodulator

- **Difference between correlators and filters**
  \[
  \begin{align*}
  \text{correlator:} & \quad r_s = \int_0^T r(\tau) f_s(\tau) d\tau \\
  \text{filter:} & \quad r_s(t) = \int_0^T r(\tau) h_s(t-\tau) d\tau.
  \end{align*}
  \]

  Then sampling at proper time, i.e., \( r_s(T) \).

  Yet, correlation demodulator requires that \{\( f_s(t) \)\} be pairwise orthogonal.

- **A correlation-type demodulator is a filter-type demodulator with each filter defined by**
  \[
  h_k(t) = f_k(T-t).
  \]

- **A filter-type demodulator is a correlation-type demodulator defined over the basis \{\( h_k(T-t) \)\}.**
  - Note that \{\( h_k(T-t) \)\} may not form a basis.
  - A filter-type demodulator with filter
  \[
  h_k(t) = s_k(T-t).
  \]
  is called the **matched-filter demodulator** to the signal \( s(t) \).
5.1.2 Matched-filter demodulator

Example of matched filters

\[ s(t) \]
\[ h(t) = s(T - t) \]

(a) Signal \( s(t) \)

(b) Impulse response of filter matched to \( s(t) \)

In the above example, the autocorrelation function of filter output peaks at \( t = T \).

\[
y(t) = \int_{-\infty}^{\infty} s(\tau)h(t-\tau)d\tau
= \int_{-\infty}^{\infty} s(\tau)s(T-(t-\tau))d\tau
= \begin{cases} 
  s(\tau)s(\tau+(T-t))d\tau, & 0 \leq t < T \\
  0, & T \leq t < 2T \text{ otherwise}
\end{cases}
\]

\[
= \begin{cases} 
  \frac{A^2}{2T} - \frac{A^2}{6T^3}t^2, & 0 \leq t < T \\
  \frac{A^2}{6T^2}t^3 - \frac{A^2}{2T^3}t^4 + \frac{2A^2T}{3}, & T \leq t < 2T \text{ otherwise}
\end{cases}
\]
5.1.2 Matched-filter demodulator

Properties of matched filters

★ Matched filter maximizes SNR₀ (output signal-to-noise ratio) under AWGN.

Proof. (i) Define $\text{SNR}_0 = \frac{y_2^2(T)}{E[y_2^2(T)]}$, where $y_2(T) = \int_0^T s(\tau)h(T-\tau)d\tau$ and $y_1(T) = \int_0^T n(\tau)h(T-\tau)d\tau$.

(Note that $r(\tau) = s(\tau) + n(\tau)$.)

(ii) $\text{SNR}_0 = \frac{y_2^2(T)}{E[y_2^2(T)]} = \frac{\left[\int_0^T s(\tau)h(T-\tau)d\tau\right]^2}{\int_0^T \int_0^T E[p(\tau)n(\tau)]h(T-\tau)h(T-\tau)d\tau d\tau}
= \frac{\left[\int_0^T s(\tau)h(T-\tau)d\tau\right]^2}{(N_0/2) \int_0^T h^2(T-\tau)d\tau}
\tag{5.23}

(iii) By Cauchy - Schwarz inequality,
\[
\left[\int_0^T s(\tau)h(T-\tau)d\tau\right]^2 \leq \left[\int_0^T s^2(\tau)d\tau\right] \left[\int_0^T h^2(T-\tau)d\tau\right]
\]
with equality holds if $h(T-\tau) = C \cdot s(\tau)$.

(iv) Hence, $\text{SNR}_0 = \frac{\left[\int_0^T s(\tau)h(T-\tau)d\tau\right]^2}{(N_0/2) \int_0^T h^2(T-\tau)d\tau} \leq \frac{2}{N_0} \left[\int_0^T s^2(\tau)d\tau\right]
\tag{5.24}
$}

where the upper bound is achieved by setting $h(T-\tau) = C \cdot s(\tau)$. 

\section*{5.1.2 Matched-filter demodulator}

(iii) By Cauchy - Schwarz inequality,
\[
\left[\int_0^T s(\tau)h(T-\tau)d\tau\right]^2 \leq \left[\int_0^T s^2(\tau)d\tau\right] \left[\int_0^T h^2(T-\tau)d\tau\right]
\]
with equality holds if $h(T-\tau) = C \cdot s(\tau)$.

(iv) Hence, $\text{SNR}_0 = \frac{\left[\int_0^T s(\tau)h(T-\tau)d\tau\right]^2}{(N_0/2) \int_0^T h^2(T-\tau)d\tau} \leq \frac{2}{N_0} \left[\int_0^T s^2(\tau)d\tau\right]
\tag{5.24}
$}

where the upper bound is achieved by setting $h(T-\tau) = C \cdot s(\tau)$.
5.1.2 Matched-filter demodulator

- Frequency-domain interpretation of the matched filter

\[ h(t) = C \cdot s(t) \Rightarrow H(f) = \int_{0}^{T} s(T - t)e^{-j2\pi f t} dt \]

\[ = e^{j2\pi f t} \int_{0}^{T} s(t)e^{-j2\pi ft} dt \]

\[ = S(-f)e^{j2\pi ft} = S^*(f)e^{-j2\pi ft}, \text{ if } s(t) \text{ is real.} \]

\[ Y_s(f) = S(f)H(f) = S(f)S(-f)e^{j2\pi ft} = |S(f)|^2 e^{j2\pi ft} \]

\[ \Phi_s(f) = \frac{1}{2} N_0 |H(f)|^2 \]

\[ SNR = \frac{\frac{1}{2} N_0 \int_{-\infty}^{\infty} Y_s(f) e^{j2\pi ft} df}{\int_{-\infty}^{\infty} \Phi_s(f) df} = \frac{2}{N_0} \int_{-\infty}^{\infty} |S(f)|^2 df = \frac{2}{N_0} \int_{-\infty}^{\infty} |s(t)|^2 dt \]

Parseval's relation

5.1.2 Matched-filter demodulator

- Example. Biorthogonal signals with basis

\[ f_1(t) = \begin{cases} \frac{1}{T} & \text{for } 0 \leq t < \frac{1}{T} \\ 0 & \text{otherwise} \end{cases}, \quad f_2(t) = \begin{cases} \frac{1}{T} & \text{for } 0 \leq t < \frac{2}{T} \\ 0 & \text{otherwise} \end{cases} \]

The matched filter to the above signals are:

\[ h_1(t) = f_1(T-t), \quad h_2(t) = f_2(T-t) \]
5.1.2 Matched-filter demodulator

\[ s_1(t) = +A\sqrt{T/2} f_1(t) = [+A\sqrt{T/2}, 0] \]
\[ s_2(t) = +A\sqrt{T/2} f_2(t) = [0, +A\sqrt{T/2}] \]
\[ s_3(t) = -A\sqrt{T/2} f_1(t) = [-A\sqrt{T/2}, 0] \]
\[ s_4(t) = -A\sqrt{T/2} f_2(t) = [0, -A\sqrt{T/2}] \]

\[ SNR_0 \text{ under AWGN} \]
\[ SNR_0 = \frac{\|s_n(t)\|^2}{N_0/2} = \frac{A^2T}{N_0} \]

5.1.2 Matched-filter demodulator

\[ y_{s_1} = \frac{H_1(s_1)}{1 + A^2T/2} \text{ if } s_1 \text{ is transmitted; } \]
\[ y_{s_2} = \frac{H_1(s_2)}{1 + A^2T/2} \text{ if } s_2 \text{ is transmitted; } \]
\[ y_{s_3} = \frac{H_1(s_3)}{1 + A^2T/2} \text{ if } s_3 \text{ is transmitted; } \]
\[ y_{s_4} = \frac{H_1(s_4)}{1 + A^2T/2} \text{ if } s_4 \text{ is transmitted. } \]
5.1.3 The optimum detector

- The optimum detector for memoryless modulated signals
  - Optimal = minimization of probability of error.
  - MAP (maximum a posterior probability) criterion
    \[ d_{MAP}(\tilde{r}) = \arg \max_{1 \leq m \leq M} \Pr\{\tilde{x}_m | \tilde{r}\} \]
  - Maximum-likelihood criterion
    \[ d_{ML}(\tilde{r}) = \arg \max_{1 \leq m \leq M} \frac{\Pr(\tilde{r}) \Pr(\tilde{x}_m | \tilde{r})}{\Pr(\tilde{x}_m)} \]

- Maximum-likelihood criterion = MAP criterion, if equal prior

---

5.1.3 The optimum detector

- MAP is optimal in probability of error.

**Proof.** Let \( R_n = \left\{ \tilde{r} : d_{ML}(\tilde{r}) = \arg \max_{i \leq M} \Pr(\tilde{x}_i | \tilde{r}) = m \right\} \)
  - \( [R_n]_m \) are pair-wise disjoint (with uniform tie breaker).
  - \( \Pr(\text{correct}) = \sum_{n=1}^{M} P(\tilde{x}_n) P(\text{correct} | \tilde{x}_n) \)
  - \( = \sum_{n=1}^{M} \int_{R_n} P(\tilde{r}) P(\tilde{x}_n | \tilde{r}) d\tilde{r} \)
  - \( = \sum_{n=1}^{M} \int_{R_n} P(\tilde{r}) d\tilde{r} \)

The proof is completed by observing that any re-assignment of \( \tilde{r} \)
(which formerly belongs to \( R_n \)) to other sets \( R_n \) will not increase \( P(\text{correct}) \).
5.1.3 The optimum detector

□ Maximum-likelihood criterion under AWGN

\[ s_i(t) = \tilde{\xi}_i = [s_{i1}, s_{i2}, \ldots, s_{iK}] \rightarrow [\tilde{r}_1, \tilde{r}_2, \ldots, \tilde{r}_K] \]
uncorrelated Gaussian with mean vector \( \tilde{\xi}_i \)

\[ s_j(t) = \tilde{\xi}_j = [s_{j1}, s_{j2}, \ldots, s_{jK}] \rightarrow [\tilde{r}_1, \tilde{r}_2, \ldots, \tilde{r}_K] \]
uncorrelated Gaussian with mean vector \( \tilde{\xi}_j \)

\[ \vdots \]

\[ s_m(t) = \tilde{\xi}_m = [s_{m1}, s_{m2}, \ldots, s_{mK}] \rightarrow [\tilde{r}_1, \tilde{r}_2, \ldots, \tilde{r}_K] \]
uncorrelated Gaussian with mean vector \( \tilde{\xi}_m \)

\[ \Rightarrow P(\tilde{r} | \tilde{\xi}_m) = \frac{1}{(\pi N_0)^{K/2}} \exp \left\{ - \frac{1}{N_0} \sum_{k=1}^{K} (\tilde{r}_k - s_{mk})^2 \right\} \]

5.31

\[ 5.32 \]

5.1.3 The optimum detector

\[ d_{ML}(\tilde{r}) = \max_{1 \leq m \leq M} \Pr \{ \tilde{r} | \tilde{\xi}_m \} \]

\[ = \max_{1 \leq m \leq M} \log \Pr \{ \tilde{r} | \tilde{\xi}_m \} \]

\[ = \max_{1 \leq m \leq M} \left[ -\frac{K}{2} \log(\pi N_0) - \frac{1}{N_0} \sum_{k=1}^{K} (\tilde{r}_k - s_{mk})^2 \right] \]

\[ = \min_{1 \leq m \leq M} \sum_{k=1}^{K} (\tilde{r}_k - s_{mk})^2 \]

\[ = \min_{1 \leq m \leq M} ||\tilde{r} - \tilde{\xi}_m||_{\text{Euclidean}} \]
5.1.3 The optimum detector

Example of signal space diagram for ML decision maker for one kind of signal assignment

Erroneous decision region for the 5th signal
5.1.3 The optimum detector

Example of alternative signal space assignment

5.1.3 The optimum detector

Erroneous decision region for the 5th signal
5.1.3 The optimum detector

- **Observations**
  - There are two factors that determine the error probability.
    - 1. The Euclidean distances among signal vectors.
      - Generally speaking, the larger the Euclidean distances among signal vectors, the smaller the error probability.
    - 2. The positions of the signal vectors.
      - The former two exemplified signal space diagrams have the same pair-wise Euclidean distance among signal vectors.
  - Minimizing the Euclidean distance = Maximum-likelihood (ML) criterion, (only) if the noise is AWGN.
  - If the noise is not AWGN, could we define a proper “distance” function so that the minimization of the “distance” = ML criterion?

- **Minimum Distance (MD) criterion**

Define a metric \( \text{metric}(\mathbf{r}, \mathbf{s}_m) \)

\[
\begin{align*}
d_{MD}(\mathbf{r}) &= \arg \min_{1 \leq m \leq M} \text{metric}(\mathbf{r}, \mathbf{s}_m).
\end{align*}
\]

- Minimum distance criterion = maximum likelihood criterion under AWGN, if Euclidean metric is considered.

\[
\begin{align*}
d_{MD}(\mathbf{r}) &= \arg \min_{1 \leq m \leq M} \left\| \mathbf{r} - \mathbf{s}_m \right\|^2 = d_{ML}(\mathbf{r}) \\
\text{Also, } d_{MD}(\mathbf{r}) &= \arg \min_{1 \leq m \leq M} \left( \left\| \mathbf{r} \right\|^2 - 2 \mathbf{r}^H \mathbf{s}_m + \left\| \mathbf{s}_m \right\|^2 \right) \\
&= \arg \max_{1 \leq m \leq M} \left( \mathbf{r}^H \mathbf{s}_m - \frac{\left\| \mathbf{s}_m \right\|^2}{2} \right)
\end{align*}
\]
### 5.1.3 The optimum detector

The optimum detector is realized as follows:

\[
d(\tilde{r}) = \arg \max_{1 \leq m \leq M} \left( < \tilde{r}, \tilde{s}_m > - \frac{\| \tilde{s}_m \|^2}{2} \right)
\]

- The 1st term = Projection of received signal onto each channel symbols.
- The 2nd term = Compensation for channel symbols with unequal powers, such as PAM.

\[
d(\tilde{r}) = \arg \max_{1 \leq m \leq M} \left( < \tilde{r}, \tilde{s}_m > - \frac{\| \tilde{s}_m \|^2}{2} \right)
\]

\[
= \arg \max_{1 \leq m \leq M} \left( \int_0^T \tilde{r}(t) s_m(t) dt - \frac{E_m}{2} \right)
\]

- Note that the front-end part is not necessarily a correlation-type demodulation, since channel symbols may not form a basis.
- The metric is named the **correlation** metric.
5.1.3 The optimum detector

- **Terminology: Correlation metric** (metric = distance function)
  - It is named the correlation metric because it is a measure of the correlation between the received vector and the m-th signal.
  
  \[ C(\tilde{r}, \tilde{s}_m) = \langle \tilde{r}, \tilde{s}_m \rangle = \frac{\| \tilde{s}_m \|^2}{2}. \]

  - Under AWGN, maximum distance criterion with correlation metric = ML criterion.

- **Terminology: PM metric**
  
  \[ PM(\tilde{r}, \tilde{s}_m) = p(\tilde{r} | \tilde{s}_m) p(\tilde{s}_m) = p(\tilde{s}_m) p(\tilde{r}). \]

  - Maximum distance criterion with PM metric = MAP criterion

---

**Example. Binary PAM with**

\[ \begin{align*}
  s_1 &= -s_2 = \sqrt{E}, \\
  P(s_1) &= 1 - P(s_2) = p.
\end{align*} \]

**Problem:** Determine the optimum MAP detector under AWGN with two-sided power spectrum density \(N_0/2\).

**Solution.**

\( r = \pm \sqrt{E} + n \), where \( n \) is zero - mean Gaussian distributed with variance \( N_0/2 \).

\( \Rightarrow r \) is Gaussian distributed with mean \( \pm \sqrt{E} \) and variance \( N_0/2 \).
5.1.3 The optimum detector

$$d_{\text{MAP}}(r) = \arg \max_{s_m \in \mathcal{S}_m} P(s_m | r)$$
$$= \arg \max_{s_m \in \mathcal{S}_m} P(s_m | r) P(r)$$
$$= \arg \max_{s_m \in \mathcal{S}_m} P(r | s_m) P(s_m)$$
$$= \arg \max_{s_m \in \mathcal{S}_m} PM(r, s_m)$$

$$= \arg \max \left\{ pe^{-t} e^{t/2} \frac{1}{N_r}, (1-p)e^{-t} e^{t/2} \frac{1}{N_r} \right\}$$

$$= \begin{cases} 
    s_1, & \text{if } r \geq \frac{N_r}{4\sqrt{E}} \ln \frac{1-p}{p} \\
    s_2, & \text{otherwise}
\end{cases}$$

Observations.

- The threshold depends not only on the prior but also on the noise power.
- With equal prior, the noise power becomes irrelevant.

$$d_{\text{MAP}}(r) = \begin{cases} 
    s_1, & \text{if } r \geq \frac{N_r}{4\sqrt{E}} \ln \frac{1-p}{p} \\
    s_2, & \text{otherwise}
\end{cases}$$
5.1.4 The maximum-likelihood sequence detector

- Optimal detector for signals with memory (not channel-with-memory or noise-with-memory, i.e., still, the noise is AWGN)
  - It is implicitly assumed that the order of the signal memory is known.
- Maximum-likelihood sequence detector
- Maximum a posteriori probability based on a sequence of received signal vectors.

Example study: NRZI

\[
\begin{align*}
\{s(t) = A \} \\
\{s(t) = -A \}
\end{align*}
\]

From the previous discussion,
\[r_k = \pm A + n_k,\]
where \(n_k\) is zero-mean Gaussian distributed with variance \(N_0/2\), and \(k\) is the index for time.
5.1.4 The maximum-likelihood sequence detector

PDF of a sequence of demodulation outputs

\[ P(r_1, \ldots, r_K \mid s_1, \ldots, s_K) = \frac{1}{(\pi N_0)^{K/2}} \exp \left[ -\sum_{k=1}^{K} \frac{(r_k - s_k)^2}{N_0} \right] \]

\( s_1, \ldots, s_K \) have memory, so it is advantageous to detect the original signals based on a sequence of outputs.

If ML rule is employed, the resultant detector is called the maximum-likelihood sequence detector.

5.1.4 The maximum-likelihood sequence detector

\[ d_{ML}(r_1, \ldots, r_K) = \arg \max_{(s_1, \ldots, s_K) \in \{0, 1\}^K} P(r_1, \ldots, r_K \mid s_1, \ldots, s_K) \]

\[ = \arg \max_{(s_1, \ldots, s_K) \in \{0, 1\}^K} \frac{1}{(\pi N_0)^{K/2}} \exp \left[ -\sum_{k=1}^{K} \frac{(r_k - s_k)^2}{N_0} \right] \]

\[ = \arg \min_{(s_1, \ldots, s_K) \in \{-1, 1\}^K} \sum_{k=1}^{K} (r_k - s_k)^2, \text{ Euclidean distance} \]

We therefore need to search for all possible combinations of \((s_1, \ldots, s_K)\), which consist of \(2^K\) possibilities.
5.1.4 The maximum-likelihood sequence detector

- **ML sequence detector for multi-dimensional signals with memory**

\[
d_{ML}(\tilde{r}_1, ..., \tilde{r}_K) = \arg \max_{(\tilde{s}_1, ..., \tilde{s}_K) \in S} P(\tilde{r}_1, ..., \tilde{r}_K | \tilde{s}_1, ..., \tilde{s}_K)
\]

\[
= \arg \max_{(\tilde{s}_1, ..., \tilde{s}_K) \in S} \frac{1}{(2\pi N_0)^{K/2}} \exp \left[ -\sum_{k=1}^{K} \sum_{j=1}^{N} (r_j - s_{kj})^2 \right]
\]

\[
= \arg \min_{(\tilde{s}_1, ..., \tilde{s}_K) \in S} \sum_{k=1}^{K} \sum_{j=1}^{N} (r_j - s_{kj})^2, \text{ Euclidean distance},
\]

where $|S| = N$. We therefore need to search for all possible combinations of $(\tilde{s}_1, ..., \tilde{s}_K)$, which consist of $2^{KN}$ possibilities.

- The complexity of “searching” the optimal solution becomes a burden.

---

**Viterbi (demodulation) Algorithm**

- A sequential trellis search algorithm that performs ML sequence detection
  - Transforming a search over $2^K$ vector points into a sequential search over a (vector) trellis
  - “sequential” = break the vectors into components and perform the search based on each component (in sequence) of the vectors

- (Also, it is a decoding algorithm for convolutional codes.)
5.1.4 The maximum-likelihood sequence detector

The number of sequences in the trellis search may be reduced by using the Viterbi algorithm.

The signal memory order of NRZI signals is 1 \((L=1)\).

- The current channel symbol only depends on the previous channel symbol.
- Assume the initial state is \(S_0\). Then the trellis will reach its regular form after the reception of the first two signals.
5.1.4 The maximum-likelihood sequence detector

- Explaining the Viterbi Algorithm (from \( S_0 \) at \( t = 0 \)).
  - There are two paths entering each node at \( t = 2T \).

  path \((I_1, I_2) = (0,0)\) or \((1,1)\)
  \(\rightarrow\) node \( S_0 \) at \( t = 2T \),
  denoted by \( S_0(2T) \).

  path \((I_1, I_2) = (0,1)\) or \((1,0)\)
  \(\rightarrow\) node \( S_1 \) at \( t = 2T \),
  denoted by \( S_1(2T) \).

- \[\begin{array}{c}
  S_0 \\
  0/\sim s(t) \\
  1/\sim s(t) \\
  0/\sim s(t) \\
  1/\sim s(t)
  \\
  S_1 \\
  0/\sim s(t) \\
  1/\sim s(t) \\
  0/\sim s(t)
  \end{array}\]

- \( t = 0 \quad t = T \quad t = 2T \)

- \( 5-53 \)

- Euclidean distance for each path
  - Euclidean distance for path \((0,0)\) entering node \( S_0(2T) = D_0(0,0) = (\tau_1 - (-A))^2 + (\tau_2 - (-A))^2 \)

  - Euclidean distance for path \((1,1)\) entering node \( S_0(2T) = D_0(1,1) = (\tau_1 - A)^2 + (\tau_2 - (-A))^2 \)

- Viterbi algorithm.
  - Discard, among the above two paths, the one with larger Euclidean distance.
  - The remaining path is called survivor at \( t = 2T \).

- Now, you should sense (at least, roughly) the key of the Viterbi algorithm.

- \[\begin{array}{c}
  S_0 \\
  0/\sim s(t) \\
  1/\sim s(t) \\
  0/\sim s(t) \\
  1/\sim s(t)
  \\
  S_1 \\
  0/\sim s(t)
  \end{array}\]

- \( t = 0 \quad t = T \quad t = 2T \)

- \( 5-54 \)
5.1.4 The maximum-likelihood sequence detector

- Euclidean distance for each path
  \[ D_1(0,1) = (r_1 - (-A))^2 + (r_2 - A)^2 \]
  \[ D_1(1,0) = (r_1 - A)^2 + (r_2 - A)^2 \]

- Viterbi algorithm.
  - Discard, among the above two paths, the one with larger Euclidean distance.
  - The remaining path is called survivor at \( t = 2T \).
- We therefore have two survivor paths after observing \( r_2 \).

![Diagram showing survivor paths](image)

5.1.4 The maximum-likelihood sequence detector

- Suppose the two survivor paths are \((0,0)\) and \((0,1)\)

![Diagram showing survivor paths](image)

- Then, there are two possible paths entering \( S_0 \) at \( t = 3T \), i.e., \((0,0,0)\) and \((0,1,1)\).

![Diagram showing survivor paths](image)
5.1.4 The maximum-likelihood sequence detector

- Euclidean distance for each path
  \[ D_0(0,0,0) = D_j(0,0) + (r_j - (-A))^2 \]
  \[ D_0(0,1,1) = D_j(0,1) + (r_j - (-A))^2 \]

- Viterbi algorithm.
  - Discard, among the above two paths, the one with larger Euclidean distance.
  - The remaining path is called survivor at \( t = 3T \).

![Diagram showing the Viterbi algorithm and survivor paths.](image-url)
5.1.4 The maximum-likelihood sequence detector

- **Viterbi algorithm**
  - Compute two metrics for the two signal paths entering a node at each stage of the trellis search
  - Remove the one with larger Euclidean distance
  - The survivor path for each node is then extended to the next state.

- The elimination of one of the two paths is done without compromising the optimality of the trellis search, because any extension of the path with the larger distance will always have a larger metric than the survivor that is extended along the same path.

- The number of paths searched reduced by a factor of two at each stage (cf. the next slide).

- The Viterbi algorithm does not reduce the computational complexity (still, $2^K$ metric computations are required).

- What the Viterbi algorithm minimizes is the number of trellis paths searched in performing ML sequence detection.
5.1.4 The maximum-likelihood sequence detector

Apply the Viterbi algorithm to Delay modulation
- 2 entering paths for each node
- \( L = \) memory order = 1
- 4 survivor paths at each stage

survivor paths = (0,0) and (0,1)

These dotted Paths are removed.
5.1.4 The maximum-likelihood sequence detector

- In the previous discussion, we only discussed “how to remove the paths?”, but did not touch the issue of “how to make decision?”.

Definition of decision delay for Viterbi decoding

- \( \text{delay} = k \) means that the transmitted bits corresponding to channel symbol at time instant \( i \) should be estimated after the reception of \( r_{k+i} \).

\[
\sum_{i=1}^{\infty} (r_i - s_i)^2
\]

The optimal decision is

\[
(\hat{\mathbf{I}}_1, \cdots, \hat{\mathbf{I}}_n)_{ML} = d_{ML}(r_1, \cdots, r_n) = \arg \min_{(r_1, \cdots, r_n) \in [-A, A]^n} \sum_{i=1}^{n} (r_i - s_i)^2
\]
5.1.4 The maximum-likelihood sequence detector

- Let's borrow an example from Example 14.3-1 of "Digital and Analog Communications" by J. D. Gibson.

- A code with $L = 2$

- Assume the received codeword is $(10,10,00,00,00,...)$

At time instant 2, one still does not know what the first two transmitted bits are. (There are two possibilities for time period 1; hence, decision delay $> 1$.) (If decision were made now, the decision delay $= T$.)
Hence, we get $r_3$ and compute the accumulated metrics for each path.

At time instant 3, one still does not know what the first two transmitted bits are. (Still, there are two possibilities for time period 1; hence, decision delay > 2.) (If decision were made now, the decision delay = $2T$.)
Hence, we get \( r_4 \) and compute the accumulated metrics for each path.

At time instant 4, one still does not know what the first two transmitted bits are. (There are two possibilities for time period 1; hence, decision delay > 3.) (If decision were made now, the decision delay = 3T.)
Hence, we get \( r_5 \) and compute the accumulated metrics for each path.

At time instant 5, one still does not know what the first two transmitted bits are. (There are two possibilities for time period 1; hence, decision delay > 4.) (If decision were made now, the decision delay = 4T.)
Hence, we get $r_6$ and compute the accumulated metrics for each path.

At time instant 6, one still does not know what the first two transmitted bits are. (There are two possibilities for time period 1; hence, decision delay $> 5$.) (If decision were made now, the decision delay $= 5T$.)
Hence, we get $r_7$ and compute the accumulated metrics for each path.

At time instant 7, one still does not know what the first two transmitted bits are. (There are two possibilities for time period 1; hence, decision delay > 6.) (If decision were made now, the decision delay = 6T.)
Hence, we get $r_8$ and compute the accumulated metrics for each path.

At time instant 8, one finally knows what the first two transmitted bits are, which is 00. Hence, the decision delay = 7T.
5.1.4 The maximum-likelihood sequence detector

- **Optimal Viterbi ML-decision maker**: To wait until there is only one possibility for the transmitted symbol. (The decision delay could be as large as the transmitted codeword length.)
- **Suboptimal Viterbi ML-decision maker**: Set a limit to the decision delay.

\[
Pr\left\{ \text{The previous } i - 5L \text{ transmitted symbols of all survivor paths are identical.} \right\} \approx 1
\]

\[
\begin{array}{c}
{r_1, \ldots, r_i} \\
\xrightarrow{\text{Viterbi path remover (after } 5L \text{ delays)}} \\
\hat{s}_{i-5L} = \text{ML estimate of } s_{i-5L}
\end{array}
\]

5.1.4 The maximum-likelihood sequence detector

- **Suboptimal Viterbi algorithm**: If there are more than one survivor paths (which results in more than one possible decoding results) remain for time period \(i - 5L\), just select the one with smaller metric, and forcefully remove the others.
- For example, NRZI codes with \(L=1\).

Suppose that two survivor paths are \((b_1, b_2, b_3, b_4, b_5)\) for state \(S_0\) and \((\bar{b}_1, \bar{b}_2, \bar{b}_3, \bar{b}_4, \bar{b}_5)\) for state \(S_1\).

If \(b_1 \neq \bar{b}_1\), select one from \((b_1, \bar{b}_1)\), whose metric (between \(D_b(b_1, b_2, b_3, b_4, b_5, b_6)\) and \(D_b(\bar{b}_1, \bar{b}_2, \bar{b}_3, \bar{b}_4, \bar{b}_5, \bar{b}_6)\)) is smaller, say the former.

Then, the two new survivor paths become \((b_1, b_2, b_3, b_4, b_5)\) for state \(S_0\) and \((\bar{b}_1, \bar{b}_2, \bar{b}_3, \bar{b}_4, \bar{b}_5)\) for state \(S_1\).

Since the contribution to accumulated metric before bit 1 is the same for both paths, we can drop it without changing the following decision results. I.e., the next decision will base on \((b_2, b_3, b_4, b_5, b_6)\) for state \(S_0\) and \((\bar{b}_2, \bar{b}_3, \bar{b}_4, \bar{b}_5, \bar{b}_6)\) for state \(S_1\).
5.1.4 The maximum-likelihood sequence detector

- The suboptimal Viterbi algorithm may still yield a bad decision (but with a very small probability);
- As far as the decision delay is concerned (with practical constraint), how about “to minimize the probability of decision error based on a fixed delay $D$.”
- Apparently, this will result in a smaller error probability than the suboptimal Viterbi algorithm.

The example in the previous slide is actually a symbol-by-symbol detector with symbol length = 6 information bits.

5.1.5 A symbol-by-symbol detector for signals with memory

- Abend and Fritchman (1970) algorithm

  Optimal in the sense of minimizing the symbol error for a given delay $D$

$$\hat{s}_k = \arg \max_{s_k \in \{s_{k-1}, s_{k-2}, \ldots, s_{k-L}, I_k\}} P(s_k \mid r_{k+D}, r_{k+D-1}, \ldots, r_1),$$

where $D$ is the decision delay with $D \geq L$.

$$\hat{s}_k = \arg \max_{s_k \in \{s_{k-1}, s_{k-2}, \ldots, s_{k-L}, I_k\}} P(s_k \mid r_{k+D}, r_{k+D-1}, \ldots, r_1)$$

$$= \arg \max_{s_k \in \{s_{k-1}, s_{k-2}, \ldots, s_{k-L}, I_k\}} \frac{P(s_k)P(r_{k+D}, r_{k+D-1}, \ldots, r_1 \mid s_k)}{P(r_{k+D}, r_{k+D-1}, \ldots, r_1)},$$

$$= \arg \max_{s_k \in \{s_{k-1}, s_{k-2}, \ldots, s_{k-L}, I_k\}} P(s_k)P(r_{k+D}, r_{k+D-1}, \ldots, r_1 \mid s_k).$$

$s_k$ is a function of $(s_{k-1}, s_{k-2}, \ldots, s_{k-L}, I_k)$, is the channel symbol, which has memory

where $I_k \in \{0,1, \ldots, M-1\}$ is the block digital input, which do not have memory

$r_k = s_k + n_k$ is a function of $(s_{k-1}, \ldots, s_{k-L}, I_k, n_k)$
5.1.5 A symbol-by-symbol detector for signals with memory

Estimation formula

\[ s_1 = \arg \max_{s_1 \in \{-1, 1\}^{M-1}} P(s_1) P(r_{1:D}, r_{1+1:D-1}, \ldots, r_1 | s_1) \]
\[ = \arg \max_{s_1 \in \{-1, 1\}^{M-1}} \sum_{s_2, s_3, \ldots} \cdots \sum_{s_1} \sum_{s_2} P(s_1, s_2, \ldots, s_{M-1}) \]
\[ = \arg \max_{s_1 \in \{-1, 1\}^{M-1}} \sum_{s_2, s_3, \ldots} \cdots \sum_{s_1} \sum_{s_2} q_1(s_1, s_2, \ldots, s_1) \]

where \( q_1(s_1, s_2, \ldots, s_1) = P(s_1, s_2, \ldots, s_1) P(r_{1:D}, r_{1+1:D-1}, \ldots, r_1 | s_1, s_2, \ldots, s_1) \)
\[ = P(r_{1:D}, r_{1+1:D-1}, \ldots, r_1, s_1, s_2, \ldots, s_1). \]

\[ \hat{s}_2 = \arg \max_{s_2 \in \{-1, 1\}^{M-1}} P(s_2) P(r_{2:D}, r_{1+1:D-1}, \ldots, r_1 | s_2) \]
\[ = \arg \max_{s_2 \in \{-1, 1\}^{M-1}} \sum_{s_1, s_3, \ldots} \cdots \sum_{s_2} \sum_{s_1} P(s_2, s_3, \ldots, s_1) \]
\[ = \arg \max_{s_2 \in \{-1, 1\}^{M-1}} \sum_{s_1, s_3, \ldots} \cdots \sum_{s_2} \sum_{s_1} q_2(s_2, s_3, \ldots, s_2) \]