Maximum Likelihood Sequence Detection

Klaus Dums
9655278

1 The Channel

If electromagnetic energy carrying a modulated signal, propagates along more than one “path” connecting the transmitter to the receiver, this is called “multipath” propagation.
Multipath propagation occurs when for example radio waves are reflected, scattered or refracted by e.g. buildings, cars or hills.

1.1 Delay Spread

Signal components traveling along different paths suffer from different attenuations and delays, which combine at the receiver to produce a distorted version of the transmitted signal. To characterize the various delays incurred by the signal traveling through the channel the *delay spread* is introduced as the largest of the delays.
The delay spread causes the two effects of *time dispersion* and *frequency selective fading*.
In a time dispersion channel a signal bandwidth much larger than the inverse of the delay spread the various paths are distinguishable as they produce copies of the transmitted signal with different delays and attenuations. For narrower bandwidths however, the received signal copies tend to overlap. This generates a form of linear distortion known as *intersymbol interference (ISI).*
If the bandwidth is large, different frequencies of the signal typically suffer from different attenuations along different paths. This is called *frequency selective fading* which is not a tpic in this paper.

1.2 Channel Model

Any further only passband PAM modulation and its degenerate case of baseband PAM are considered, since FSK is generally not used in situations where ISI is an important phenomenon.
PAM is preferred in this situation because FSK is a nonlinear modulation technique and is therefore difficult to equalize properly.
In general a channel can be modeled as a linear time-invariant filter with impulse response b(t) and additive noise N(t), as shown in Figure 1-1.

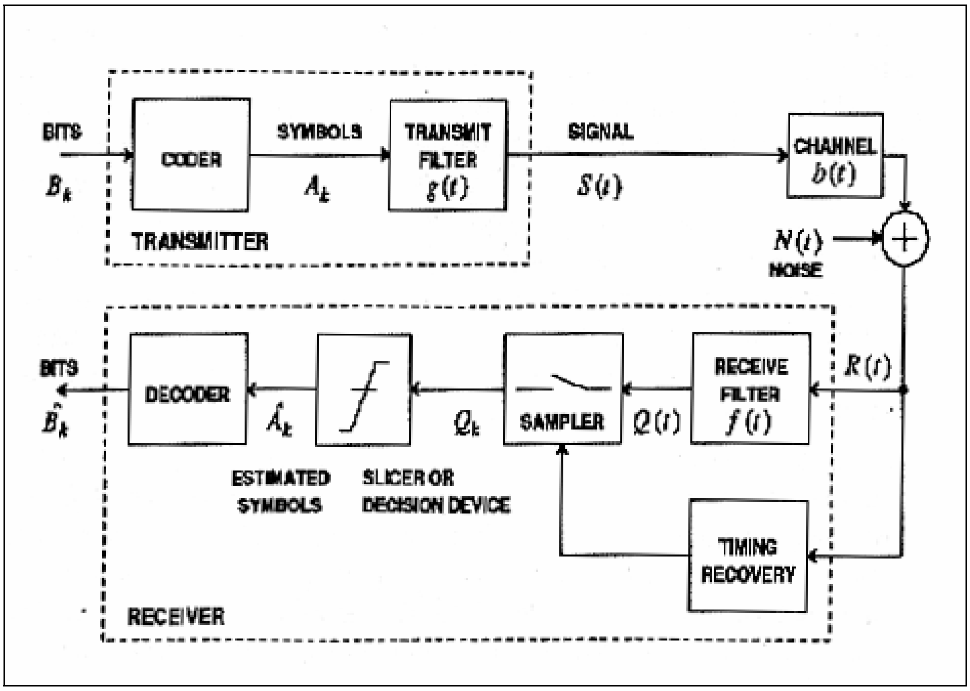

Figure 1-1

Since the input to this transmit filter is discrete-time, and the output is sampled by the sampler, we can replace the transmit filter, channel and receive filter plus the sampler with an equivalent discrete-time filter.
With

$$U_k = U(kT) = [N(t) * f(t)]_{t = kT} \tag{1}$$

and

$$p_k = p(kT) = [g(t) * b(t) * f(t)]_{t = kt} . \tag{2}$$

The discrete-time equivalent channel model for PAM (as in Figure 1-1) is then shown in Figure 1-2.

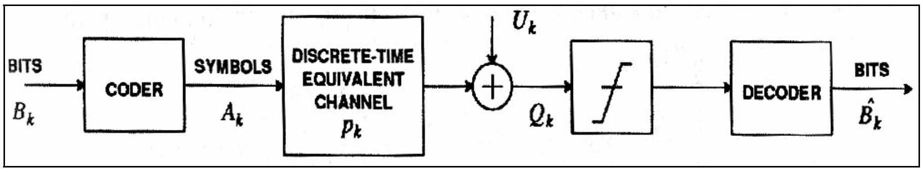

Figure 1-2

A passband PAM system with transmit filter g(t) and receive filter f(t)can be modeled as a completely discrete-time system if the continuous-time subsystems are considered as part of the channel. The equivalent discrete-time channel has impulse response p_k and transfer function P(z).

$$P(e^{j\omega T}) = \sum_{m=-\infty}^{\infty} G\left[j\left(\omega - \frac{2\pi}{T} m \right) \right] B_E\left[j\left(\omega - \frac{2\pi}{T} m \right) \right] F\left[j\left(\omega - \frac{2\pi}{T} m \right) \right] \tag{3}$$

The equivalent noise $Z_k = Z(kT)$ has independent real and imaginary parts each with variance given by σ^2. The power spectrum of the noise is given by

$$S_Z(e^{j\omega T}) = \frac{2N_0}{T} \sum_{m=-\infty}^{\infty} \left| F\left(j\left(\omega + m\tfrac{2\pi}{T} \right) \right) \right|^2 \tag{4}$$

where T equals the symbol rate.

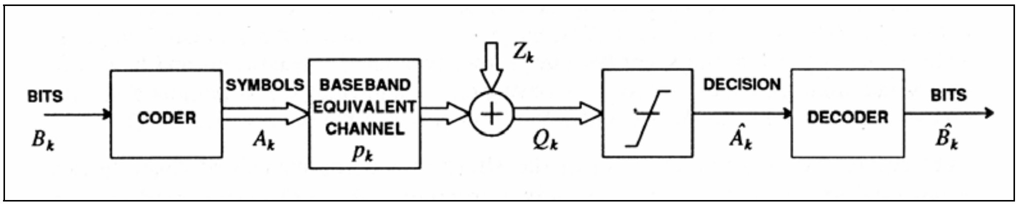

Figure 1-3

1.3 Matched Filter as Receiver Front End

The first part of the receiver – the receive filter and the sampler – are typically built of a matched filter that is sampled at symbol rate.
The discrete-time equivalent channel model utilizing a sampled matched filter may be depicted as in Figure 1-4.

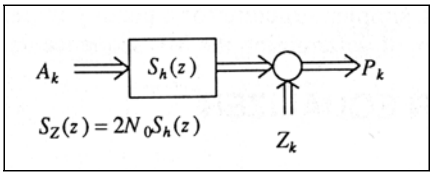

Figure 1-4

Of fundamental importance to the channel model is the autocorrelation function for a given baseband receive pulse shape h(t)

$$\rho_h(k) = \int_{-\infty}^{\infty} h(t)h^*(t-kT)dt\,. \tag{5}$$

The Fourier transform of (5) autocorrelation $S_h(e^{j\omega T})$ is called *folded spectrum*:

$$S_h(e^{j\omega T}) = \sum_{k=-\infty}^{\infty} \rho_h(k)e^{j\omega kT} = \frac{1}{T}\sum_{m=-\infty}^{\infty}\left|H\left(j\left(\omega + m\tfrac{2\pi}{T}\right)\right)\right|^2 \tag{6}$$

The folded spectrum is real valued, nonnegative and in general not symmetric.

Now the discrete time channel that includes the matched filter is easily obtained using the discrete-time model for PAM systems.
The receive filter from the discrete PAM model is replaced by the sampled matched filter $F(j\omega) = H^*(j\omega)$. The impulse response of the baseband channel $\rho_h(k)$, the pulse autocorrelation function and the transfer function are equal to the folded spectrum $S_h(e^{j\omega T})$.
Further, the equivalent discrete-time noise process Z_k has power spectrum

$$S_Z(e^{j\omega T}) = 2N_0 S_h(e^{j\omega T})\,. \tag{7}$$

At this point the calculation of probability density functions of the received estimates P_k is laborious because the noise in this model is not white and therefore the random variables Pk are not independent.
Therefore the colorization from the matched filter has to be compensated. This can be achieved by spectral factorization of

$$S_h(z) = \gamma^2 M(z)M^*(1/z^*)\,. \tag{8}$$

Then $S_h(z) = H_{\min}(z)H_{ap}(z)$ can be decomposed to a *minimal phase* filter and an *allpass filter* where $H_{\min}(z) = \gamma^2 M(z)$. So adding the inverse minimal phase filter to the matched filter compensates the transfer function to constant attenuation of all frequencies as in Figure 1-5.

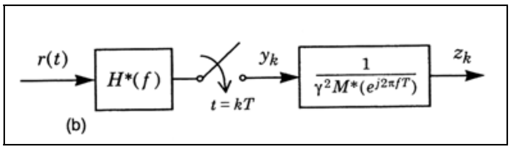

Figure 1-5

Therefore a new channel model is obtained where $S_N(z) = \frac{2N_0}{K}$ is the variance of the now Gaussian noise process N_k and U_k is the received symbol distorted by ISI.

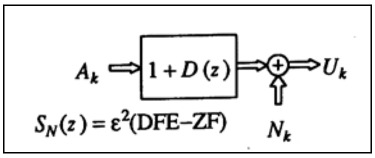

Figure 1-6

2 Detection

The general approach to deriving optimal receivers is to model the relationship between the transmitted and received signals by joint probability distributions. Based on a noisy observation (the received signal plus noise), we wish to estimate or detect the input signal. The term *estimation* is used when the transmitted signal is a continuous-valued random variable and the term *detection* when the transmitted signal is discrete-valued (even if the received signal is continuous-valued). So the first case refers to analog and the second to digital communication systems.
In this section the following model will be used.

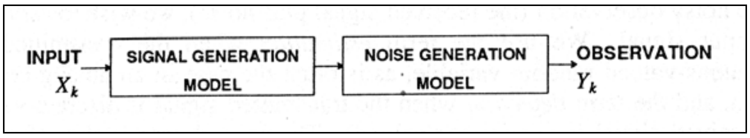

Figure 2-1

The deterministic portion of the channel is called *signal generation* and the random component the *noise generation*.
The input X_k is a discrete-time and discrete-valued random process and has only a finite number of possible values.

2.1 Terms

- Alphabet A:
  the set of symbols those are available for transmission. A = {-1,+1,-j,+j}
- Symbol a_k, A_k :
  k-th symbol from the alphabet. a_1=-1
- Size of the alphabet M:
  number of symbols in the alphabet. M = 4
- Sample space of the Alphabet Ω_A
- Symbol of the sample space â

2.2 Maximum Likelihood Detection of a Single Symbol

In order to decide about the received data symbol, knowledge about the noise generator is needed. The noise generator is completely specified by the discrete distribution of Y conditioned in the knowledge of the data symbol $p_{Y|A}(y\,|\,\hat{a})$.

The maximum likelihood (ML) detector chooses $\hat{a}\,\varepsilon\,\Omega_A$ to maximize the likelihood $p_{Y|A}(y\,|\,\hat{a})$, where y is the observation of Y.

The ML detection is a special case of maximum a-posteriori probability detection for the simplifying assumption that all the possible inputs $p_A(\hat{a})$ are equally likely.

$$p_{A|Y}(\hat{a}\,|\,y) = \frac{p_{Y|A}(y\,|\,\hat{a})\,p_A(\hat{a})}{p_Y(y)} \tag{9}$$

Example:

Suppose that we have additive discrete noise N, so that Y = A+N. Assume A and N are independent and take on values zero and one according to the flip of two fair coins. There are three possible observations, y = 0, 1, or 2.
The likelihoods $p_{Y|A}(y\,|\,\hat{a})$ for the observation y=0 are:

$p_{Y|A}(0\,|\,0) = 0.5$

$p_{Y|A}(0\,|\,1) = 0$

So if the observation is y=0, the ML detector selects â = 0. If the observation is y=1, then the likelihoods are equal
$p_{Y|A}(1\,|\,0) = p_{Y|A}(1\,|\,1) = 0.5$
and therefore the ML detector selects either zero or one. If the observation is y = 2, the ML detector selects â = 1.

The probability of error $\Pr[error] = \Pr[\hat{a} \neq a]$ is a measure of the quality of the detector.

Example:

From the previous example Y=A+N takes on values in {0, 1, 2} with probabilities {0.25, 0.5, 0.25}, respectively. If the observation is y = 0 or y = 2 the ML detector does not make errors. If the observation is y = 1, however, the detector is unsure and chooses for example at random. Then the detector will be wrong half the time.
Since $p_Y(1) = 0.5$, the total probability of error is $\Pr[error] = 0.25$.

2.3 Maximum Likelihood Detection of a Signal Vector

Since the goal in this paper is the detection of a sequence of symbols it is convenient to extend the ML detection of a single symbol to a vector of symbols.
Here the signal generator accepts an input X and maps it into a vector signal S with dimension M. The observation is a vector Y with the same dimension as the signal. The noise generator is specified by the conditional distribution of the observation given the signal $f_{\mathbf{Y}|\mathbf{S}}(\mathbf{y}\,|\,\mathbf{s})$.

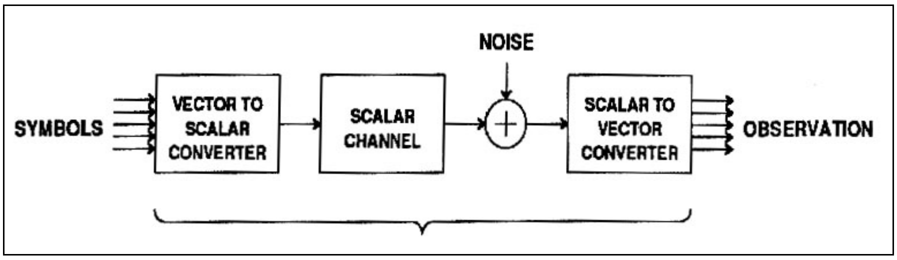

Figure 2-2

A common characteristic of the noise generator is independent noise components

$$f_{\mathbf{Y}|\mathbf{S}}(\mathbf{y}\,|\,\mathbf{s}) = \prod_{k=1}^{M} f_{Y_k|S_k}(y_k\,|\,s_k) \qquad (10).$$

The ML detector chooses the signal vector $\hat{\mathbf{s}}$ from all the possibilities in order to maximize the conditional probability $f_{\mathbf{Y}|\mathbf{S}}(\mathbf{y}\,|\,\hat{\mathbf{s}})$, which is given directly from the noise generator.

For additive Gaussian noise with variance σ^2 and independent noise components

$$f_{\mathbf{Y}|\mathbf{S}}(\mathbf{y}\,|\,\hat{\mathbf{s}}) = f_{\mathbf{N}|\mathbf{S}}(\mathbf{y}-\hat{\mathbf{s}}\,|\,\hat{\mathbf{s}}) = f_{\mathbf{N}}(\mathbf{y}-\hat{\mathbf{s}}) \qquad (11).$$

Hence the ML detector selects $\hat{\mathbf{s}}$ to maximize

$$f_{\mathbf{N}}(\mathbf{y}-\hat{\mathbf{s}}) = \frac{1}{(2\pi)^{M/2}\sigma^M}\exp\left(-\frac{1}{2\sigma^2}\left\|\mathbf{y}-\hat{\mathbf{s}}\right\|^2\right) \qquad (12)$$

where M is the dimension of the vectors and f_N is the PDF of the Noise. Maximizing this is equivalent to minimizing $\left\|\mathbf{y}-\hat{\mathbf{s}}\right\|$, the Euclidian distance between the received vector y and the signal vector $\hat{\mathbf{s}}$.

2.4 ML Detection with Intersymbol Interference

Consider a situation where the input to the signal generator is a single data symbol A, and the signal generator is a linear time-invariant filter with impulse response $h_k, 0 \le k \le M$, assumed to be finite in extent.

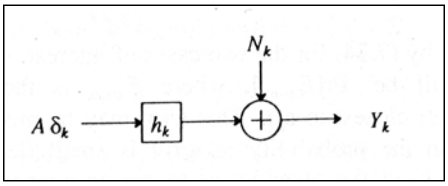

Figure 2-3

This filter represents the dispersion of the data symbol in time. A single isolated discrete-time impulse δ_k amplitude modulated by an isolated data symbol A is transmitted. The noise is independent additive and Gaussian generated. The observation Y_k is only considered within the range $0 \le k \le M$, since there are no signal components outside. Thus the observation may be written in vector notation as $\mathbf{Y} = \mathbf{h}A + \mathbf{N}$.

If the single data symbol A from alphabet Ω_A, the ML detector selects $\hat{a}\,\varepsilon\,\Omega_A$ to minimize the Euclidian distance between the vectors $\hat{a}\mathbf{h}$ and observation y. Thus the ML detector computes

$$\|\mathbf{y}-\mathbf{h}\hat{a}\|^2 = \|\mathbf{y}\|^2 - 2\langle \mathbf{y},\mathbf{h}\rangle \hat{a} + \|\mathbf{h}\|^2 \hat{a}^2 \tag{13}$$

and chooses â for which this is the minimum.

Since $\|\mathbf{y}\|^2$ does not depend on the decision, the ML detector equivalently selects â to maximize

$$2\langle \mathbf{y},\mathbf{h}\rangle \hat{a} - \|\mathbf{h}\|^2 \hat{a}^2. \tag{14}$$

The term

$$\langle \mathbf{y},\mathbf{h}\rangle = \sum_m y_m h_m = [y_k * h_{-k}]_{k=0} \tag{15}$$

may be computed using a correlator or a matched filter as depicted in Figure 2-4.

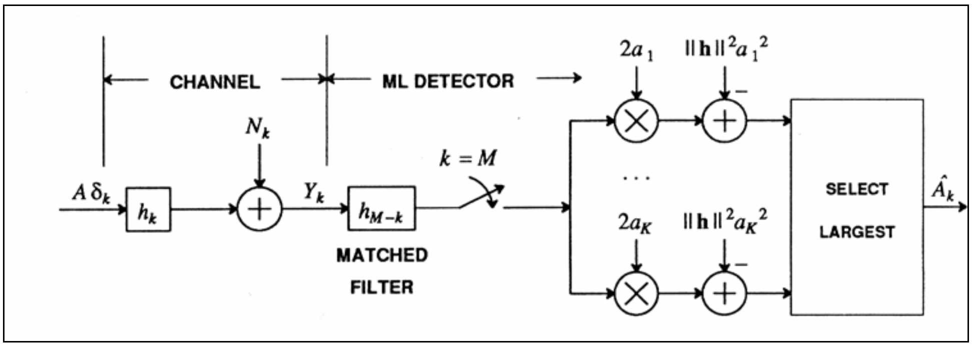

Figure 2-4

Example:

Suppose that a symbol A from the alphabet $\Omega_A = \{0,1\}$ is transmitted through the LTI system with impulse response $h_k = \delta_k + 0.5\delta_{k-1}$.

In this case M=1 and $\|\mathbf{h}\|^2 = 1.25$, so the ML detector can be implemented like in Figure 2-5.

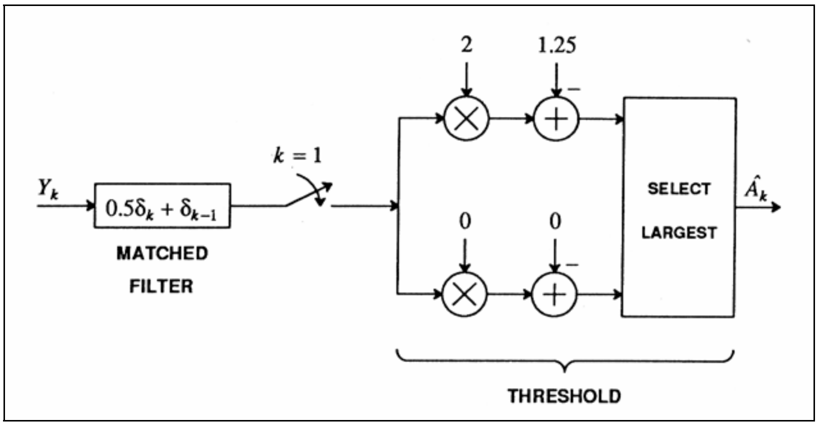

Figure 2-5

In practice, however, this solution would be unacceptable due to its excessive complexity. In fact, for a message of K M-ary symbols, M^K matched filters might be necessary, with about M^K comparisons to be performed to select their largest sampled output.

2.5 Sequence Detection

To overcome the problems of the last section A. Viterbi of UCLA in 1967 proposed an algorithm for digital communication problems that can be formulated as suitable Markov chains. This algorithm selects the most likely symbol sequence in a number of computations and memory size that grow only linearly with respect to the message length K.

2.5.1 Markov Chains

A discrete-time Markov process $\{\Psi_k\}$ is a random process that satisfies

$$p\left(\Psi_{k+1} \mid \Psi_k, \Psi_{k-1}, \ldots\right) = p\left(\Psi_{k+1} \mid \Psi_k\right). \tag{16}$$

In words, the future sample $\{\Psi_{k+1}\}$ is independent of past samples $\Psi_{k-1}, \Psi_{k-2}, \ldots$ if the present sample is known.

A Markov chain is called *homogenous* if the conditional probability $p\left(\Psi_k \mid \Psi_{k-1}\right)$ is not a function of k. A homogenous Markov chain is therefore a kind of stationary or time invariance.

A Markov chain can be visualized by a state transition diagram as shown in Figure 2-6.

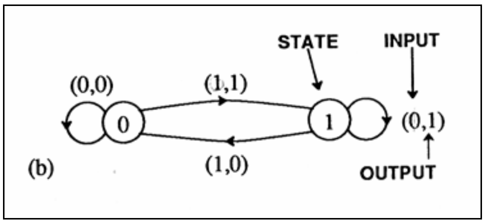

Figure 2-6

Another valuable visualization is called the Trellis diagram, first suggested by D. Forney.

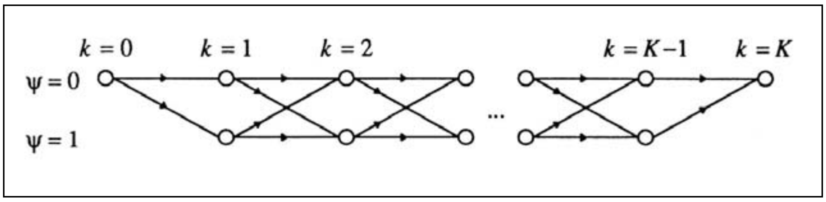

Figure 2-7

Figure 2-7 shows the possible progression of states over time. Each small circle is called a *node* of the trellis, and corresponds to the Markov chain being a particular state at a particular time. Each arc is called a *branch*, and corresponds to a particular state transition at a particular time. The branches may be labeled like the state transitions of the transition diagram.
A collection of branches from the beginning to the terminal node is called a *path*.

2.5.2 Markov Chain Signal Generator

Let Ψ_k be a sequence of a homogenous Markov chain. The sample space of each state Ψ_k is finite. For the signal generator of interest, the signal samples are a function of the Markov chain *state transitions* $S_k = g(\psi_k, \psi_{k+1})$ where $g(\cdot,\cdot)$ a memory less function as in Figure 2-8.

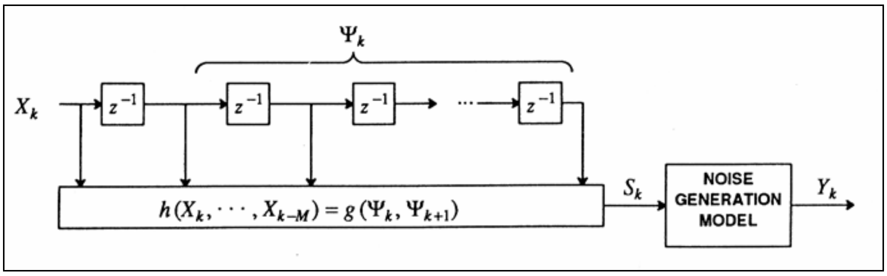

Figure 2-8

The ML detector has to detect the sequence of states given an observation sequence Y_k , which is S_k passed through a noise generator with independent noise components.

If h_k is a finite impulse response, than the inter symbol interference model that was previously developed in (see Figure 2-3) is an example of a shift-register process, where the observation function is

$$g(\psi_k, \psi_{k+1}) = \sum_{i=0}^{M} h_i A_{k-1} \,. \qquad (17)$$

The state of the Markov chain is

$$\Psi_k = \left[X_{k-1}, X_{k-2}, \ldots, X_{k-M} \right] \qquad (18)$$

where M is the length of the shift-register.

Example:

Consider a discrete-time channel with impulse response $h_k = \delta_k + 0.5\delta_{k-1}$.

Using the notation from the shift-register process this is depicted in Figure 2-9 (a). If the input symbols are independent identically distributed (i.i.d.), the observation Y_k is a noisy observation of the Markov chain with state diagram (b). Assuming binary inputs A_k, there are only two states in (b) corresponding the two possible previous values for A_{k-1}. The arcs are labeled with the input/output (A_k,S_k) of the signal generator.

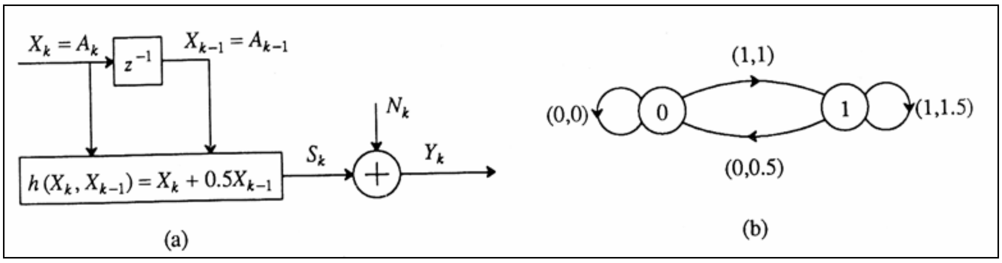

Figure 2-9

The Trellis diagram looks as follows:

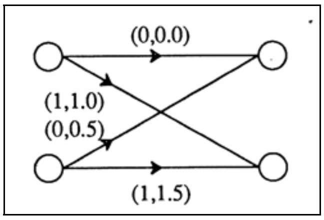

Figure 2-10

2.5.3 <u>The Viterbi Algorithm</u>

Every possible path in a trellis has a one-to-one correspondence with an input sequence $X_k, 0 \le k \le K$. The goal of a detector, based in the observation S_k corrupted by noise, is to decide the sequence of inputs. Deciding the sequence of inputs is equivalent to deciding the path through the trellis diagram.

Recall that there is a signal s_k associated with each branch and each stage k of the trellis. For each stage k there is also an observation y_k. After observing y_k, we can assign to each branch of the trellis a numerical value called the branch metric that is low if y_k is close to s_k and high otherwise. For the Gaussian case the correct *branch metric* is

$$\textit{branch metric} = |y_k - s_k|^2 \tag{19}$$

Then for each path through the trellis, the *path metric*, which is the sum of the branch metrics, is calculated. The preferred path is the one with the lowest path metric.

This is nothing else than minimizing the Euclidian distance $\|\mathbf{y} - \hat{\mathbf{s}}\|$ between the received vector y and the decision $\hat{\mathbf{s}}$, as in section 2.3.

Unfortunately the number of paths through the trellis grows exponentially with the message length K and therefore the computation time and memory usage, too.

The Viterbi algorithm is a clever way of reducing this exponential growth to linear.
Consider one node of the trellis an all paths through the trellis that pass through this node. There are a large number of paths through this node, but all these paths follow just *one* of four routs routes through this node AB, AD, BC or BD as in Figure 2-11.

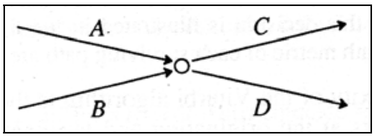

Figure 2-11

The path metric of a path through this node is the sum of the *partial path metrics* of the portion of the path on the right side of the node and on the left side of the node. Among all the partial paths to the left of the node, the detector picks the one with the smallest partial path metric, which is called the *survivor path* for that node. If any other partial path would be chosen the overall path metric would be grater than the one with the survivor path, and therefore would not be the smallest path. So every other partial path but the survivor path to the left may be thrown away.
The Viterbi algorithm finds the path with the minimum path metric by sequentially moving through the trellis and at each node retaining only the survivor paths. At each stage of the trellis it cannot be determined at which node the optimal path passes through, so one survivor

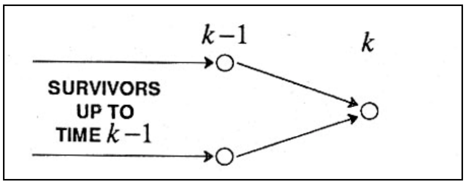

Figure 2-12

path for each and every node must be retained.
In a case where there are only two incoming branches to a given node at time k like in Figure 2-12, the two possibilities for the survivor path to the node at time k are consisting of the survivor paths at time k-1 plus the branches to time k. To decide the survivor the partial path metrics for each of those two paths are determined by summing the partial path metric of the survivor at time k-1 and the branch metric. The survivor path at that node is chosen as the path with the smaller partial path metric.

Example:

The trellis shown below is marked with branch metrics corresponding to the observation sequence {0.2, 0.6, 0.9, 0.1} of a discrete-time channel with impulse response $h_k = \delta_k + 0.5\delta_{k-1}$ and additive Gaussian noise as in the previous example.

The possible state transitions are the following.

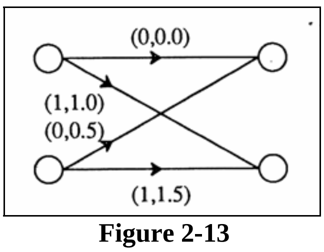

Figure 2-13

The branches are labeled with their corresponding branch metrics $\left|y_k - s_k\right|^2$.

Also the survivor paths for each node and the partial path metrics of each surviving path are shown in Figure 2-14.

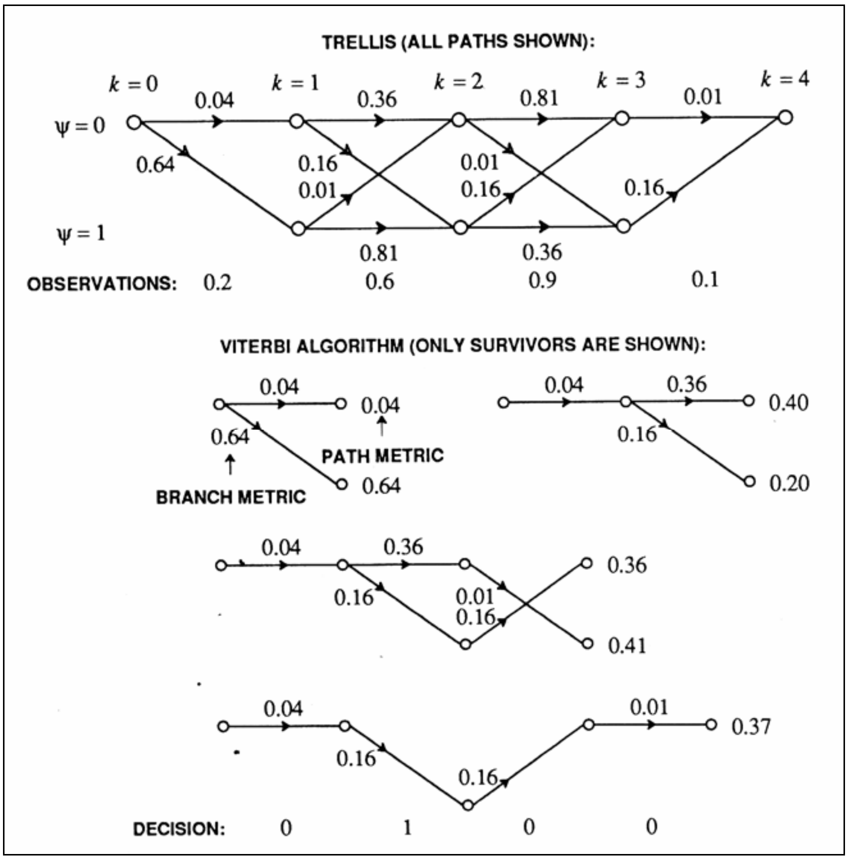

Figure 2-14

A symbol by symbol ML detector would have decided, that the transmitted bits were {0, 1, 1, 0} whereas the ML sequence detector decides for {0, 1, 0, 0} taking into account knowledge of the ISI.

The computational complexity of the Viterbi algorithm is the same at each time increment, and hence the total complexity is proportional to the length of K.
Yet there is a practical problem remaining. The algorithm does not determine the optimal path until the terminal code of the trellis.
Since in digital communication systems state sequences may be very long this may result in a very long delay and a very big memory consumption which is generally not affordable.
Occasionally all the survivor paths at some time k coincide up to some time k-d, then the partial paths have merged at depth d, and a decision can be made to all the inputs or states up to time k-d.
Unfortunately this depends on good fortune and therefore the algorithm is slightly modified to force a decision at time k for all transitions prior to time k-d, for some *truncation depth* d.
Usually all the partial path metrics of the N partial paths at time k are compared and the smallest path is picked. This path represents all the inputs or states until time k-d and the node corresponding to this path is chosen to be the input or state at time k-d for further computation.
If d is chosen to be large enough, this modification will have negligible impact on the probability of detecting the correct sequence.

3 Error Probability Calculation

The *sequence error probability* is the probability that the path chosen through the trellis does not correspond to the actual state sequence, or in other words the probability that one or more states are in error. As the length of the message K increases this probability usually approaches unity. Fortunately the concern is normally with the probability of a single symbol or bit error, or the probability of a sequence error for a relatively short sequence.
This implies that it would be better to optimize for single symbol error, but there are two reasons not to do this.
First, it is much more complicated to implement and in fact the best known algorithms for bit-by-bit decision have exponential complexity in K. Secondly, the performance of the ML sequence detector is almost identical at high SNR to the detector that minimizes the bit or symbol error probability.

3.1 Error Event

It turns out useful to introduce the concept of an *error event* for the purpose of determining the symbol error probability.
Let $\{\Psi_k\}$ be the actual state sequence and $\{\hat{\Psi}_k\}$ be selected by the Viterbi algorithm. Each error event starts when the actual path and the selected path start to diverge and ends when

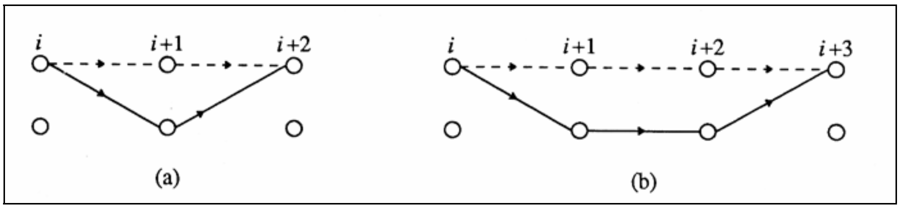

Figure 3-1

they remerge. The length of an error event is the number of incorrect nodes in the path.
So the error event in Figure 3-1 (a) has a length of one and in (b) a length of two.
An error event yields in one ore more *detection errors*, which are incorrect symbols or bits as a result from taking an incorrect path through the trellis.
The distance of the error event to the actual path is calculated in the following way.
Assume an actual state sequence, and label each branch in the trellis with its squared distance form the corresponding branch of the actual state sequence as in Figure 3-2. The actual state sequence will have branch metrics that are zero, and normally all the branches off the actual path sill have a non-zero branch metric.
The path metric of each path through the trellis is now the square of the Euclidian distance of that path from the correct all-zero path. Using the $branch\, metric = |y_k - s_k|^2$ the distance can easily be obtained to be $\sqrt{1.25}$ for (a) and $\sqrt{3.5}$ for (b) in Figure 3-2.

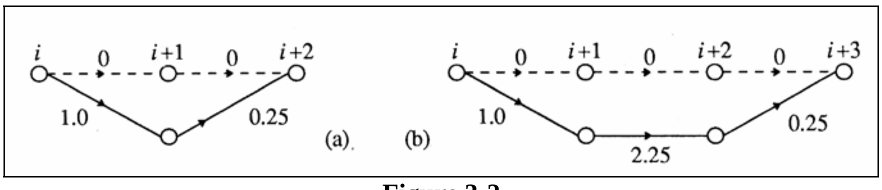

Figure 3-2

3.2 Symbol Error Probability

Now let E denote the set of all error events starting at time *i*. Each element *e* of *E* is characterized by both a correct path Ψ and an incorrect path $\hat{\Psi}$ that diverge and remerge sometime later.
Pr[e] is assumed stationary that is Pr [e] is independent of i, the starting time of the error event.
The approximation is accurate if the trellis is long relative to the length of the significant error events.
Each error event causes one or more detection errors, where a detection error at time k means that X_k at stage k of the trellis is incorrect. For the ISI example, each X_k is a symbol A_k, so a detection error is the same as a symbol error.
Function (20) characterizes the sample times corresponding to detection errors in error event *e*.

$$c_m = \begin{cases} 1; & \text{if e has a detection error in position m (from the start i)} \\ 0; & \text{otherwise} \end{cases} \tag{20}$$

The probability of a particular error event e starting at time i and causing a detection error at time k is $c_{k-i}(e)\Pr[e]$.

Of cause, if one error event occurs no other can occur. Therefore the error Events in E are disjoint so

$$\Pr[\text{detection error at time k}] = \sum_{i=-\infty}^{k} \sum_{e \varepsilon E} \Pr[e] c_{k-i}(e) = \sum_{e \varepsilon E} \Pr[e] \sum_{i=-\infty}^{k} c_{k-i}(e) \tag{21}$$

The definition of the total number of detection errors

$$w(e) = \sum_{i=-\infty}^{k} c_{k-i}(e) = \sum_{m=0}^{\infty} c_m(e) \tag{22}$$

yields

$$\Pr[\text{detection error}] = \sum_{e \varepsilon E} \Pr[e] w(e) . \tag{23}$$

Hence, the probability of a detection error is equal to the expected number of detection errors in all error events starting at any fixed time i.

The probability of the error event Pr[e] is depends on the probabilities of both the correct and incorrect paths Ψ and $\hat{\Psi}$ that make up *e*,

$$\Pr[e] = \Pr\left[\Psi\right] \Pr\left[\hat{\Psi} \middle| \Psi\right] . \tag{24}$$

Since it is usually difficult to extract expressions for $\Pr\left[\hat{\Psi} \middle| \Psi\right]$, only bounds are derived.

3.2.1 Upper Bound of Detection Error

Suppose that Ψ is the actual trajectory, corresponding to signals s in Figure 3-3 (a).

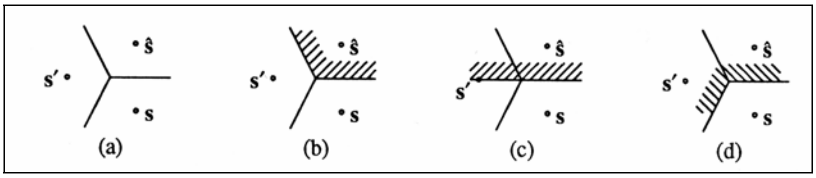

Figure 3-3

The region corresponding to the detection of $\hat{\Psi}$ is shown in (b). The probability of the noise carrying the observation in this region is very difficult to calculate, but it is easy to find an upper bound by using a larger decision region as in (c) which ignores the possibility of any trajectory other than Ψ and $\hat{\Psi}$.

For the additive white Gaussian noise model, the probability of region in (c) is

$$\Pr\left[\hat{\Psi} \mid \Psi\right] \le Q\left(d(\hat{\Psi}, \Psi)/2\sigma\right) \tag{25}$$

with Q(.) is defined as

$$Q(x) = \Pr[X > x] = 1 - F_X(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\alpha^2/2} d\alpha \tag{26}$$

If Q(x) is defined like that for a Gaussian random variable $X \sim N(\mu, \sigma^2)$ it becomes

$$\Pr[X > x] = Q\left(\frac{x-\mu}{\sigma}\right). \tag{27}$$

Here $d(\hat{\Psi}, \Psi)$ is the Euclidean distance between transmitted signal $\hat{\mathbf{s}}$ and s corresponding to the state trajectories $\hat{\Psi}$ and Ψ.

Hence the probability of a detection error can be written as

$$\Pr[\text{detection error}] \le \sum_{e \varepsilon E} w(e) \Pr[\Psi] Q\left(d(\hat{\Psi}, \Psi)/2\sigma\right) \tag{28}$$

If only the set B of error events with minimal distance d_{min} are considered

$$\Pr[\text{detection error}] \le \sum_{e \varepsilon E} w(e) \Pr[\Psi] Q\left(d_{\min}/2\sigma\right) + \text{other terms} \tag{29}$$

the other terms become insignificant at high SNR.

This probability approaches

$$RQ(d_{\min}/2\sigma) \tag{30}$$

where

$$R = \sum_{e \varepsilon B} w(e) \Pr[\Psi]. \tag{31}$$

Example:

Here the error event e1 with w(e1)=1 and the distance $\sqrt{1.25}$ is considered as depicted in Figure 3-4. This is the minimum distance.

For a trellis with two states there are eight error events where each of them has also w(e)=1.

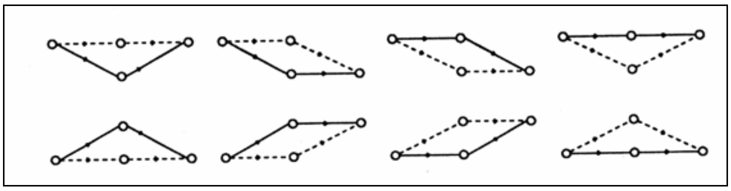

Figure 3-4

So $R = \sum_{e \varepsilon B} w(e)\Pr[\Psi]$ becomes $R = \sum_{e \varepsilon B} 1\Pr[\Psi]$. If all of these error events are equally likely then $R = \sum_{e \varepsilon B} 1*1/8 = 1$ and

$\Pr[\text{detection error}] \le \sum_{e \varepsilon E} Q\left(\sqrt{1.25}/2\sigma\right) + \text{other terms}$.

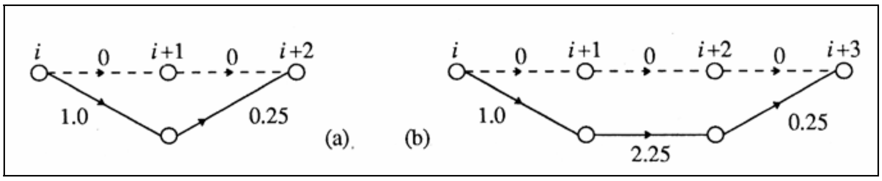

Figure 3-5

3.2.2 <u>Lower Bound of Detection Error</u>

To obtain the lower bound the decision region of (d) can be used. This is the decision region where any error event conditioned on the actual state sequence Ψ is shown.
The probability of any error event is clearly lower bounded by calculating the smaller decision region shown in (c).

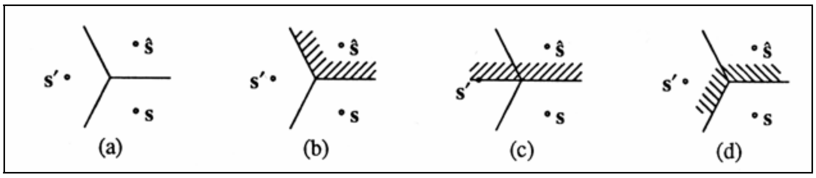

Figure 3-6

Since $w(e) \ge 1$ for all error events e the probability of a detection error can be written as

$$\Pr[\text{detection error}] \ge \sum_{e \varepsilon E} \Pr[e] = \Pr[\text{an error event}]. \quad (32)$$

For one particular path through the trellis let $d_{\min}(\Psi)$ denote the distance of the minimum distance error event. Let $d_{\min}$ be the minimum distance error event over all possible actual state sequences Ψ. Then $d_{\min}(\Psi) \ge d_{\min}$. To make the bound strongest, obviously the closest error event to Ψ has to be chosen, which is one of those at distance $d_{\min}(\Psi)$.
Than the probability of any error event becomes

$$\Pr[\text{an error event} \mid \Psi] \ge Q(d_{\min}(\Psi)/2\sigma). \quad (33)$$

Consequently the probability of a detection error can be written as

$$\Pr[\text{detection error} \mid \Psi] \geq Q(d_{\min}(\Psi)/2\sigma)\,. \quad (34)$$

And using the total probability this is

$$\Pr[\text{detection error}] \geq \sum_{\Psi} \Pr[\Psi] Q(d_{\min}(\Psi)/2\sigma)\,. \quad (35)$$

It is save to omit some terms here because all terms are positive. So consider only those state sequences Ψ for which $d_{\min}(\Psi) = d_{\min}$.

$$\Pr[\text{detection error}] \geq \sum_{\Psi \varepsilon A} \Pr[\Psi] Q(d_{\min}(\Psi)/2\sigma) \quad (36)$$

Here A is the set of actual paths Ψ that have the minimum distance error event, and $d_{\min}$ is that minimum distance.
Define the probability that a randomly chosen Ψ has an error event starting at any fixed time with distance $d_{\min}$ as

$$P = \sum_{\Psi \varepsilon A} \Pr[\Psi]\,. \quad (37)$$

Than the lower bound can be expressed as

$$\Pr[\text{detection error}] \geq PQ(d_{\min}(\Psi)/2\sigma)\,. \quad (38)$$

3.2.3 Upper and Lower Bounds of Detection Error

Combining upper an lower bounds,

$$PQ(d_{\min}/2\sigma) \leq \Pr[\text{detection error}] \leq RQ(d_{\min}(\Psi)/2\sigma) \quad (39),$$

where the upper bound is approximate since some terms were thrown away.

To conclude, at high SNR

$$\Pr[\text{detection error}] \approx CQ(d_{\min}/2\sigma) \quad (40)$$

for some constant C between P and R.

Example:

From the previous example note that P=R=1. Hence $\Pr[\text{detection error}] \approx Q(\sqrt{1.25}/2\sigma)$.
Here each detection error causes exactly one bit error, so $\Pr[\text{detection error}] = \Pr[\text{bit error}]$. Hence, with this sequence detector approximately the same probability of error is obtained as for an isolated pulse and a matched filter receiver.

In general a single detection error may cause more than one bit error. Suppose each input of the Markov chain X_k is determined by n source bits (hence the alphabet has size 2^n). Then each detection error causes at least one and at most n bit errors.

$$\frac{1}{n}\Pr[\text{detection error}] \leq \Pr[\text{bit error}] \leq \Pr[\text{detection error}] \quad (41)$$

The typical assumption is pessimistic:

$$\Pr[\text{detection error}] \approx \Pr[\text{bit error}] \quad (42)$$