SIGNAL SPACE CONCEPTS

In this section we familiarize ourselves with the representation of discrete-time and continuous-time communication signals using the concepts of vector spaces.

These concepts provide a powerful and intuitive framework for attacking various problems in communications signal processing.

The results are used later for developing optimum signal processing methods to convert a received continuous-time signal to a symbol-rate discrete-time signal, without loosing any information that could be useful in detecting a digitally modulated signal, both in ISI free channels and in the presence of ISI.

Source: Lee & Messerschmitt, Section 2.6.
Vector Spaces

A linear space, or vector space consists of
• a set of vectors
• operations:
  o vector addition
  o multiplication by a scalar

The space must include a zero-vector and each vector \( \mathbf{X} \) must have an additive inverse \(-\mathbf{X}\). The operations must satisfy the following properties:

\[
\begin{align*}
\mathbf{X} + \mathbf{Y} &= \mathbf{Y} + \mathbf{X} \\
0 + \mathbf{X} &= \mathbf{X} \\
\alpha \cdot \mathbf{X} \text{ belongs to the space} \\
\alpha \cdot (\beta \cdot \mathbf{X}) &= (\alpha \beta) \cdot \mathbf{X} \\
\alpha \cdot (\mathbf{X} + \mathbf{Y}) &= \alpha \cdot \mathbf{X} + \alpha \cdot \mathbf{Y} \\
(\alpha + \beta) \cdot \mathbf{X} &= \alpha \cdot \mathbf{X} + \beta \cdot \mathbf{X} \\
0 \cdot \mathbf{X} &= 0
\end{align*}
\]

Here \( \alpha \) and \( \beta \) are arbitrary real or complex numbers (scalars).

The usual \( n \)-dimensional Euclidean space is a familiar example. In this case, a vector \( \mathbf{X} \) is represented by its \( n \) elements \( x_1, x_2, \ldots, x_n \):

\[
\mathbf{X} \leftrightarrow (x_1, x_2, \ldots, x_n)^T
\]

The elements may be real- or complex-valued.
Signal Spaces

**Discrete-Time Signal Space** consists of discrete-time complex-valued signals:

\[ Y \leftrightarrow (\ldots, y_{-1}, y_0, y_1, \ldots)^T \]

Here the vectors are considered to be infinite-dimensional. This space has similar properties as finite-dimensional Euclidean space. In addition, the signals are required to have finite energy:

\[ \sum_{k} |y_k|^2 < \infty \]

**Continuous-Time Signal Space** consists of continuous-time complex-valued signals:

\[ Y \leftrightarrow y(t) \]

Also here, the signals are required to have finite energy:

\[ \int_{-\infty}^{\infty} |y(t)|^2 dt < \infty \]

This is a generalization of Euclidean spaces, where the enumerable coordinate system is replaced by a continuum. This is intuitively a bit difficult step, but most of the useful properties of signal spaces are still valid.
Signal Spaces (continued)

Example: In a PAM/PSK/QAM system, with transmitted symbol $a$ and pulse shape $h(t)$, the signal
\[ x(t) = ah(t) \]
can be represented using the signal space concepts as
\[ X = a \cdot H \]
where $x(t)$ and $h(t)$ are considered as vectors:
\[ X \leftrightarrow x(t) \]
\[ H \leftrightarrow h(t) \]
About the Geometric Structure of Signal Spaces

The concept of inner product of two vectors can be used for defining such concepts as length (or norm) of a vector, and the angle between two vectors. Hilbert space is an inner product space (i.e., a space where an inner product is defined) with certain additional completeness property.

The inner product, and the corresponding (squared) norms are denoted and defined as follows:

**Discrete-Time Signal Space:**

\[
\langle X, Y \rangle = \sum_{k=-\infty}^{\infty} x_k y_k^* \\
\|X\|^2 = \sum_{k=-\infty}^{\infty} |x_k|^2
\]

**Continuous-Time Signal Space:**

\[
\langle X, Y \rangle = \int_{-\infty}^{\infty} x(t) y^*(t) dt \\
\|X\|^2 = \int_{-\infty}^{\infty} |x(t)|^2 dt
\]

In general, two vectors are orthogonal iff \( \langle X, Y \rangle = 0 \). The angle between two vectors can be defined as:

\[
\langle X, Y \rangle = \|X\| \cdot \|Y\| \cos \theta
\]
Example: Matched Filter

In a PAM/PSK/QAM system, with pulse shape $h(t)$, the matched filter impulse response is $h^*(-t)$. If $x(t)$ is the received signal, then the matched filter output at time zero is

$$y(0) = \int_{-\infty}^{\infty} x(t)h^*(t)dt = \langle X, H \rangle$$
Subspace

Subspace is a subset of the vectors satisfying the properties of a vector space.

A subspace can be obtained from an arbitrary set of vector by including all the linear combinations of those vectors.

The following result is important in the continuation:

*Projection Theorem:* Assume that $H$ is a Hilbert space, $M$ is its subspace, and $X$ is a vector of $H$. Then there is a unique vector of $M$, $P_M(X)$, that has the property

$$\langle X - P_M(X), Y \rangle = 0$$

for every vector $Y$ of $M$. $P_M(X)$ is the projection of $X$ in $M$. 
Schwarz Inequality

Two arbitrary vectors of an inner product space satisfy:

$$\langle X, Y \rangle \leq \|X\| \cdot \|Y\|$$

Equality applies only if $X = K \cdot Y$.

Note

In the following, we show both vector notation and waveform notation, in parallel. These expressions are equivalent, just using different notations to highlight the signal vector concept. In the literature, it is quite common to use just the waveform notation but still talk about signal vectors.
Signal Space Models of Digitally Modulated signals

**Binary FSK:**

Isolated symbol waveforms are orthogonal and, with suitable scaling, orthonormal with the following basis:

$$\Phi_j \leftrightarrow \phi_j(t) = \sqrt{\frac{2}{T}} \cos(\omega_j t) \quad 1 \leq j \leq 2$$

The possible transmitted signal vectors are

$$X_1 = \sqrt{\frac{T}{2}} \Phi_1 \quad X_2 = \sqrt{\frac{T}{2}} \Phi_2$$

**Binary PAM-signal:**

Alphabet \{-1, 1\}, square pulse shape.

Let us consider $K$ bit binary PAM sequences. There are $M = 2^K$ of those and they have an orthonormal basis

$$\Phi_j \leftrightarrow \phi_j(t) = \sqrt{\frac{1}{T}} g(t - jT) \quad 1 \leq j \leq K$$

$K$ bit PAM signals can be represented as:

$$X_m = \sqrt{T} \sum_{k=1}^{K} a_k \Phi_k \leftrightarrow x_m(t) = \sqrt{T} \sum_{k=1}^{K} a_k \phi_k(t)$$
MAXIMUM LIKELIHOOD DETECTION FOR CONTINUOUS-TIME CHANNELS

So far we have considered ML detection in case of discrete-time channel, based either on discrete-valued or continuous-valued observation sequences.

In this part we consider the case with
- continuous-time, continuous-valued channel
- additive Gaussian noise
- arbitrary channel impulse response -> ISI

The analysis is based on signal space concepts.

**Main idea:** We show first that certain discrete-time observation vectors have the same information as the continuous-time received signal. Then we can use the ML-sequence detection principles as described earlier.

Source: Lee & Messerschmitt, Sections 8 - 8.2.
Signals

We assume that the noise-free received signal is one out of \( M \) possible ones:

\[
S_m \leftrightarrow s_m(t), \quad -\infty < t < \infty, \quad 1 \leq m \leq M, \quad E_m = \int_{-\infty}^{\infty} s_m^2 dt < \infty
\]

We use mainly the signal space model.

The \( M \) signal vectors span a subspace \( M_S \), which has a dimension \( K \leq M \). This subspace has an orthonormal basis \( \Phi_1, \ldots, \Phi_K \)

\[
\Phi_j \leftrightarrow \phi_j(t).
\]

All the \( M \) signal vectors can be represented in this basis as:

\[
S_m = \sum_{k=1}^{K} s_{m,k} \Phi_k \quad \leftrightarrow \quad s_m(t) = \sum_{k=1}^{K} s_{m,k} \phi_k(t)
\]

Note

In practical applications, \( s_m(t) \) will be the signals after the deterministic signal generation model, i.e., the effects of the multipath channel are included, but not channel noise. This will be the case on page 231 and onwards. However, for the time being, they may be considered as the undisturbed baseband or bandpass waveforms of the transmitter.
Signals with Additive Gaussian Noise

We assume that the received signal is one out of $M$ possible ones and it includes also additive Gaussian noise:

$$y(t) = s_m(t) + N(t)$$

Such a noisy signal cannot be represented as such in the signal space, e.g., because the noise energy is not limited. We define, instead, the following signal:

$$\bar{y}(t) = \sum_{k=1}^{K} \Xi_k \phi_k(t) \leftrightarrow \bar{Y} = \sum_{k=1}^{K} \Xi_k \Phi_k$$

where

$$\Xi_k = \int_{-\infty}^{\infty} y(t) \phi_k^*(t) dt = s_{m,k} + n_k$$

$$n_k = \int_{-\infty}^{\infty} N(t) \phi_k^*(t) dt$$

In a way, $\bar{Y}$ is the projection of the observed signal to the subspace of possible signals $M_s$. (However, this is mathematically not quite correct statement.)

If $y(t)$ was a vector of the signal space, $\bar{Y}$ would be its projection to $M_s$. In other words, out of the vectors of $M_s$, it would be the one that is closest to the observation.
About the Effects of Noise in Detection

Starting with a generic formulation, let $\alpha(t)$ and $\beta(t)$ be complex-valued signal (vectors), belonging to the signal space. Let $N(t)$ be white Gaussian noise with power spectral density $N_0$.

We define the random variables

$$n_\alpha = \int_{-\infty}^{\infty} N(t)\alpha^*(t)dt$$

$$n_\beta = \int_{-\infty}^{\infty} N(t)\beta^*(t)dt$$

Then

$$E\left[n_\alpha n_\beta^*\right] = N_0 \langle \alpha(t), \beta(t) \rangle$$

$$E\left[|n_\alpha|^2\right] = N_0 \|\alpha(t)\|^2$$

$$\langle \alpha(t), \beta(t) \rangle = 0 \quad \Rightarrow \quad E\left[n_\alpha n_\beta^*\right] = 0$$

Thus, since the basis functions are orthonormal, the noise components in $\Xi_k$ are uncorrelated and they all have the same variance, $N_0$. 
Sufficient Statistics

It can be shown that $\overline{Y}$, i.e., the random variables $\overline{X}_k, 1 \leq k \leq K$ form a sufficient statistics for the received signal $y(t)$.

This means that $\overline{y}(t)$ can be used in detection instead of $y(t)$ without losing any useful information that could help to improve the detection performance. What is lost, is insignificant in the detection, it is independent of both of the signal and the noise components of $\overline{y}(t)$.

An optimum receiver may make the decision about which of the $M$ signals was transmitted equally well from $\overline{y}(t)$ as from $y(t)$. 
The random variables $\Xi_k$, $1 \leq k \leq K$ can be considered as a vector that is composed of the signal vector and additive noise:

$$
\begin{bmatrix}
\Xi_1 \\
\vdots \\
\Xi_K
\end{bmatrix} = 
\begin{bmatrix}
S_{m,1} \\
\vdots \\
S_{m,K}
\end{bmatrix} + 
\begin{bmatrix}
n_1 \\
\vdots \\
n_K
\end{bmatrix}
$$

The noise components are uncorrelated, so the discrete-time ML detection principles discussed earlier can be used in detection.

If $[\xi_1, \ldots, \xi_K]_T$ is the observation vector, the ML detector minimizes the Euclidean distance

$$
d_m = \sum_{k=1}^{K} |\xi_k - s_{m,k}|^2 = \| \bar{Y} - S_m \|^2.
$$

This can be illustrated as follows:

The detection is based on the differences of $S_m$ and $\bar{Y}$, that are determined by the noise components included in the sufficient statistics. The noise is Gaussian distributed and its variance is $N_0$ in the direction of all unit vectors. Thus, the decision regions are “in the middle” of the signal vectors, as illustrated in the figure.
In general, an ML receiver works as follows:

1. Calculate the correlations

\[ \xi_k = \int_{-\infty}^{\infty} y(t) \phi_k^*(t) dt, \quad 1 \leq k \leq K \]

2. Calculate the distances and choose \( m \) giving minimum:

\[
\| \bar{Y} - S_m \|^2 = \sum_{k=1}^{K} |\xi_k - s_{m,k}|^2 = \sum_{k=1}^{K} (\xi_k - s_{m,k})(\xi_k^* - s_{m,k}^*)
\]

\[
= \sum_{k=1}^{K} \left( |\xi_k|^2 - \text{Re}[\xi_k s_{m,k}^*] + |s_{m,k}|^2 \right)
\]

Since the term \( \sum |\xi_k|^2 \) doesn’t have any effect on the choice, the decision variables can be calculated as:

\[
K_m = \sum_{k=1}^{K} \text{Re}[\xi_k s_{m,k}^*] - E_m / 2 \quad \quad E_m = \sum_{k=1}^{K} |s_{m,k}|^2 = \|S_m\|^2
\]

Now the decision variable \( K_m \) is to be maximised.

Generally, the following structure is needed for obtaining each of the \( M \) decision variables. Of course, the correlations with basis vectors need to be computed only once.
Correlation Receiver: Binary FSK Example

According to the model (branches with 0-weight not included):

\[ \int \phi_1(t) t \int \phi_2(t) \]

\[ \text{SELECT LARGEST} \]

\[ y(t) \]

\[ \phi_1(t) \]

\[ \phi_2(t) \]

Simplified form:

\[ \int \phi_1(t) \]

\[ \int \phi_2(t) \]

\[ \text{SELECT LARGEST} \]
Matched Filter Receiver

Correlation and convolution are closely related operations. We can write

\[
\xi_k = \int_{-\infty}^{\infty} y(t) \phi_k(t) * (t) \, dt = \int_{-\infty}^{\infty} y(\tau) \phi_k(t-\tau) \, d\tau \bigg|_{t=0} \\
= [y(t) * \phi_k(-t)] \bigg|_{t=0}
\]

So the correlation can be obtained by filtering the input signal with the matched filter and sampling at \( t=0 \).

The matched filters are here matched to the basis signals \( \phi_k(t), k = 1, \ldots, K \) and they have the impulse responses \( \phi_k * (-t) \), i.e., they are complex conjugated mirror images of the basis waveforms.
About Correlation and Matched Filter Receivers

Above, the correlation receiver was based on correlating the received signal with the basis vectors, and the equivalent matched filter receiver was based on filters matched to pulse shapes corresponding to the basis vectors.

Equivalently, instead of correlating with the basis functions, the input signal can be correlated with all the possible signal vectors to get the decision variables:

\[ K_m = \int_{-\infty}^{\infty} y(t)s_m^* (t)dt - \frac{1}{2} \int_{-\infty}^{\infty} s_m^2 (t)dt, \quad m = 1, \ldots, M \]

or the matched filters can be matched to the corresponding ‘clean’ received waveforms.

The choice between these alternatives depends on the dimensionality and structure of the signal space, and application in mind.

It should be kept in mind that the noise properties may be different in these cases. The important property of uncorrelated and equal variance noise components in a correlation receiver (or, equivalently, in matched filter receiver) is achieved only when the decision variables correspond to orthonormal basis functions!
Error Probability

The evaluation of error probabilities is based on the general results for discrete-time ML detection.

Now we have also the tools to understand the significance of the signal geometry from the system performance point of view. We have obtained a generic signal vector presentation, such that the received signal can be considered as a sum of ‘clean’ signal vector and additive Gaussian noise. Also, the noise is symmetrically distributed, the noise components being uncorrelated and the noise variance being $N_0$ in the direction of all basis vectors.

Then the Q-function can be used for calculating the symbol error probability from the signal power/energy and noise standard deviation, in the same way as we did in the case of PAM/PSK/QAM modulations.
Error Probability: Binary Signalling Example

In the generic binary signalling case, we have the following signal geometry:

The noise component in the direction of vector $S_2 - S_1$ is Gaussian distributed and its variance is $N_0$. A decision error takes place if this noise component is larger than $\frac{\|S_2 - S_1\|}{2}$ in magnitude, with suitable choice of the sign. Thus, the error probability is

$$P_e = Q\left[\frac{\|S_2 - S_1\|}{2\sqrt{N_0}}\right].$$

In case of equal-length signal vectors, $\|S_1\| = \|S_2\| = \sqrt{E}$, where $E$ is the signal energy for both signals.

In the special cases of antipodal (PSK) and orthogonal (FSK, MSK) modulations, we have the following signal geometries, and there is a 3 dB difference in performance:

$$P_e = Q\left[\sqrt{E}/\sqrt{2N_0}\right].$$
Baseband PAM Example

In case of detecting an isolated symbol $a_0$ using pulse shape $h(t)$, we can write:

$$y(t) = a_0 h(t) + N(t)$$

$$H \leftrightarrow h(t)$$

$$\sigma_h = \|H\|$$

$$\Phi_1 = \frac{1}{\sigma_h} H$$

One basis vector, $\Phi_1$, is needed (i.e., the dimensionality of the signal space is 1) and the sufficient statistic is just a single random variable $\Xi_1$.

In the implementation, one correlator and slicer are needed:

![Diagram](https://via.placeholder.com/150)
Passband PSK/QAM Example

In case of detecting an isolated symbol $a_0$ using pulse shape $h(t)$, we can write:

$$y(t) = \sqrt{2} \Re \left[ a_0 h(t) e^{j\omega_c t} \right] + N(t)$$

$$H \leftrightarrow h(t)$$

$$\sigma_h = \|H\|$$

$$\Phi_1 \leftrightarrow \phi_1(t) = \frac{1}{\sigma_h} \sqrt{2} \Re \left[ h(t) e^{j\omega_c t} \right]$$

$$\Phi_2 \leftrightarrow \phi_2(t) = -\frac{1}{\sigma_h} \sqrt{2} \Im \left[ h(t) e^{j\omega_c t} \right]$$

$$y(t) = \sigma_h \Re [a_0] \phi_1(t) + \sigma_h \Im [a_0] \phi_2(t) + N(t)$$

In the implementation, two correlators and a slicer are needed:

In an alternative model, the dimensionality of the signal space is 1, but complex scalar coefficients are used. This results in use of a complex correlator, which can be realized using two real correlators, just as in the above figure.
Matched Filter for Passband PAM/PSK/QAM

We have earlier seen that the needed correlations can be calculated using a matched filter:

\[
\int_{-\infty}^{\infty} y(t)h^*(t)e^{-j\omega_c t} \, dt = (y(t)e^{-j\omega_c t})*h^*(-t)|_{t=0}.
\]

The matched filter structure for passband PAM/PSK/QAM:

The matched filter has the properties:

- For the received pulse shape \( h(t) \), the matched filter impulse response is \( h^*(-t) \).
- The transfer function is \( H^*(f) \), where \( H(f) \) is the Fourier transform of \( h(t) \). The magnitude response of the matched filter is the same as that of \( H(f) \).
- The matched filter attenuates those frequencies that are attenuated by the channel, by the same amount. In doing so, it maximises the S/N ratio at the matched filter output.
- The cascade of received pulse shape and matched filter has symmetric zero-phase impulse response. (In a way, the matched filter acts as a phase equalizer.) The corresponding frequency response is \( |H(f)|^2 \), i.e., real.
Matched Filter in Case of Coloured Noise

If the channel noise is not white, but has a non-flat power spectrum \( S_N(f) \), then the previous ideas are not optimal as such. But if a noise whitening filter \( S_N^{1/2}(f) \) is included at the receiver input, then the noise at its output is white. The matched filter to be added as the next stage should now be matched to the cascade of received pulse shape and noise whitening filter:

\[
y(t) \xrightarrow{f_w(t)} S_N^{-1/2}(f) \xrightarrow{f_N^{-1/2}(f) - 1/2} H^*(f) \frac{S_N^{1/2}(f + f_c)}{S_N^{1/2}(f + f_c)}
\]

The filter stages can be combined as follows:
Splitting the Filtering Functions

The received pulse shape, $h(t)$, and the corresponding transfer function, $H(f)$, includes the cascade of the transmit filter, $G(f)$, and the channel transfer function, $B(f)$.

The channel cannot usually be affected, but the transmit filter can be optimized for system performance.

In the simple case of ideal channel ($B(f) = 1$), the usual way is to use root-raised cosine filters, resulting in raised cosine overall response, since the receiver filter is also a root-raised cosine filter according to the matched filter principle. Also the Nyquist criterion for zero ISI is satisfied in this case.

In case of non-ideal channel, the typical approach is to use root-raised cosine transmit filter also in this case. Applying the matched filter receiver principle doesn’t provide a zero ISI solution, since the overall transfer function is the cascade of raised cosine response and squared magnitude of the channel frequency response. In this case, some kind of channel equalizer is needed.

The system performance doesn’t usually depend critically on satisfying exactly the matched filter condition. Cancelling the ISI induced by the channel is far more important in practice. Anyway, the correlation receiver, or matched filter receiver, is the theoretical basis for the development of channel equalizers in the continuation.
Generic Model for Passband Signal Detection

We consider signals of the form

\[ s_m(t) = \sqrt{2} \Re \left[ h_m(t)e^{j\omega_c t} \right], \quad 1 \leq m \leq M \]

where \( h_m(t) \) is the equivalent baseband signal.

When the received signal is \( y(t) \) and the carrier phase, \( \theta \), is known in the receiver, the sufficient statistics is obtained from the random variables:

\[ Y_m = \int_{-\infty}^{\infty} \sqrt{2} \Re \left[ h_m(t)e^{j(\omega_c t + \theta)} \right] dt = \sqrt{2} \Re \left[ e^{-j\theta} \int_{-\infty}^{\infty} y(t)e^{-j\omega_c t} h_m^*(t) dt \right] \]

The correspond structure is as follows:

This formulation is
- Generally applicable, also for non-linear modulation
- Shows that carrier phase offset can be compensated at baseband after sampling.

If non-coherent detection is possible for the used modulation, the following optimal non-coherent ML detection structure can be derived:
Other Optimization Criteria

We have seen that the optimal structures based on ML estimation theory for detecting isolated symbols is the correlation receiver or matched filter receiver. The same principle can also be applied when detecting a sequence of symbols if there is no intersymbol interference.

Instead of ML-criterion, also the other optimization criteria to be described briefly in the following could be considered for use. In fact, in case of isolated symbols or 0 ISI, they lead to same result as ML criterion. However, as will be seen soon, the results are rather different when these results are applied in the case of non-ideal channel, and the different criteria lead to different sub-optimal equalizer solutions.

S/N-ratio as a criterion:

$$SNR = \frac{\sigma_A^2 |\langle H, F \rangle|^2}{2N_0 \|F\|^2} \leq \frac{\sigma_A^2 \|H\|^2 \|F\|^2}{2N_0 \|F\|^2}$$

This is maximized by the choice $F = c \cdot H$, which is exactly the matched filter solution. With $E_b = \sigma_A^2 \|H\|^2$, the maximal SNR can be written as:

$$SNR_{MF} = \frac{E_b}{2N_0}$$
Other Optimization Criteria (continued)

Zero forcing criterion:

This criterion forces the ISI to zero, and assuming root-raised cosine type of transmit filtering, it gives naturally the same result as the matched filter receiver in case of an ISI-free channel.

Mean Squared Error (MSE) criterion:

Here the mean squared error between the correct symbol values and samples taken at the receiver filter output are minimized. The result can again be shown to be essentially equivalent to the previous criteria in the ideal channel case. (You can check this later easily as a special case of the MSE solution for linear equalizer.)
**Sampled Matched Filter**

Now we start to consider PAM/PSK/QAM signal detection in case of ISI channel.

We consider detecting $2^B$-level sequences of $K$ symbols so the are $M = 2^{B \cdot K}$ different possible transmitted signals in the signal space, which has the dimensionality $K$.

The received signal is:

$$y(t) = \sqrt{2} \text{Re} \left[ \sum_{k=1}^{K} a_k h(t - kT) e^{j\omega_t} \right] + N(t)$$

In this case, the random variables

$$p_k = \int_{-\infty}^{\infty} y(t) \sqrt{2} h^*(t - kT) e^{-j\omega_t} dt, \quad k = 1, \ldots, K$$

form the sufficient statistics, corresponding to set of basis vectors obtained from $h(t - kT), \ k = 1, \ldots, K$.

The first stages of the receiver are as follows:

This structure is referred to as sampled matched filter.
Sampled Matched Filter (continued)

In this model, sampling is done at symbol rate. This is usually smaller than twice the highest frequency of the desired signal, so aliasing takes place. In spite of this, no useful information is lost in the sampling, provided that the preceding receiver filter is truly a matched filter.

It should be noted that there is one very strong implicit assumption in this model, that has to be taken into account in practical implementation.

The matched filter is the non-causal mirror image of the received pulse shape, and it is affected also by the channel delay, in addition to the shape of the waveform. Symbol synchronization functionality is needed in practical receivers to measure the channel delay. In the non-causal sampled matched filter model, correct symbol synchronization is assumed implicitly. Failure to synchronize the sampling clock properly to the received signal would cause a major performance degradation.
Equivalent Discrete-Time Channel Model

The sampled matched filter is a safe 'pre-processing' solution, independently of the used equalizer optimization criterion. In a way, it transforms the continuous-time receiver input to a symbol-rate discrete-time sequence.

When analysing the equalizer and other discrete-time processing functions, a discrete-time channel model is quite sufficient.

The impulse response of the chain is:

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

This is actually the symbol-rate-sampled autocorrelation of the received pulse shape. Its Fourier-transform is:

$$S_h(e^{j2\pi fT}) = \sum_{k=-\infty}^{\infty} \rho_h(k)e^{-j2\pi fkT} = \frac{1}{T} \sum_{k=-\infty}^{\infty} \left| H(f + \frac{k}{T}) \right|^2$$

This is referred to as the aliased spectrum of the received pulse shape. The general discrete-time system model, including transmit filter, channel, and sampled matched filter is now as follows:

receiver filter \( F(f) = H^*(f) \)

impulse response of the baseband channel model: \( \rho_h(k) \)

system frequency response and transfer function: \( S_h(e^{j2\pi fT}) \), \( S_h(z) \)

equivalent noise spectrum: \( S_Z(e^{j2\pi fT}) = 2N_0 S_h(e^{j2\pi fT}) \)
Equivalent Discrete-Time Channel Model (cont.)

It should be noted that the model developed above is a special case, with matched filter as the receiver filter, of the generic model introduced on page 101.

In the special case when the transfer function of the equivalent discrete-time model satisfies the Nyquist criterion, the aliased spectrum of the received pulse shape is flat:

$$S_h(e^{j2\pi f T}) = \rho_h(0)$$

More generally, it can be seen that ISI=0 if and only if the aliased spectrum of the received pulse shape is flat.

Then the sample sequence at the output of the sampled matched filter is

$$p_k = \rho_h(0)a_k + Z_k.$$ 

Furthermore, the noise sequence, $Z_k$, is white and, consequently, the noise samples are independent of each other.

The development so far provides a formal proof for the optimality of the ad hoc decision principles presented earlier for linear digital modulation methods in case of zero ISI channel.

It is important to remember that in case of a dispersive (non-ideal) channel, the noise in the sampled matched filter model is not white. This is consistent with the observation that the set of basis vectors used on slide 231, $h(t - kT)$, $k = 1, \cdots, K$, is orthogonal only in case of ideal channel.
In the following, we will first introduce the basic channel equalizer structures:

- Linear Equalizer (LE)
- Decision Feedback Equalizer (DFE)
- Maximum Likelihood Sequence Detector (MLSD)

The last one of the mentioned equalizer structures is the most optimal one in the ML sense. LE and DFE are suboptimal solutions based on certain other optimization criteria. They are interesting because they are much simpler from the implementation point of view, and provide reasonably good performance in modestly frequency selective cases.

The performance of these general equalizer structures is discussed in general terms.

Also the idea of Fractionally Spaced Equalizer (FSE) is introduced, and the importance of multirate signal processing is emphasized in this context.

Then adaptive methods for adjusting the equalizer coefficients are introduced, notably the LMS algorithm.
Linear Equalizer (LE)

Linear equalizer is simply a linear filter, with transfer function $C(z)$, placed after the sampled matched filter.

Linear equalizers have the so call noise enhancement problem:

In those frequencies where the channel attenuates the signal, a notch is seen in the aliased spectrum. The equalizer tries to recover these frequency components and in doing so, it greatly amplifies the noise in those frequencies.

In the following, we consider two different criteria for designing an LE, which treat the noise enhancement problem in a somewhat different manner:

- LE-ZF  zero-forcing criterion
- LE-MSE  mean-square error criterion

In this part, we are not (yet) worried about the equalizer complexity, so there is no limitation to the length of the equalizer impulse response.
Zero-Forcing LE

This criterion forces the ISI to zero in the ideal case, with possibly infinite length equalizer. This is achieved by choosing the equalizer transfer function as:

\[ C(z) = S_h^{-1}(z) \]

in which case the overall transfer function becomes

\[ S_h(z)S_h^{-1}(z) = 1. \]

In case of a carrier modulated system, there is no symmetry of the channel transfer function with respect to 0-frequency. This means that the lowpass equivalent transfer function \( S_h(e^{j\omega}) \) has complex coefficients, and so does also the equalizer.

The noise spectrum in the above model can be written as:

\[ S_Z(e^{j\omega T}) = 2N_0S_h(e^{j\omega T})\left|C(e^{j\omega T})\right|^2 = \frac{2N_0S_h(e^{j\omega T})}{\left|S_h(e^{j\omega T})\right|^2} \]

\[ = \frac{2N_0}{S_h(e^{j\omega T})} \]
Zero-Forcing LE (continued)

If the channel has a transmission zero on the signal frequency band, then the LE-ZF would have infinite gain at that frequency, which would mean an unstable (and unusable) equalizer. This clearly indicates that the linear equalizer with zero forcing criterion cannot be implemented in all cases.

However, if \( C(e^{j\omega T}) \), as given above, is integrable, the noise power to be calculated next is finite, and the LE-ZF solution exists.

The noise power at the LE-ZF equalizer output is given by:

\[
\varepsilon^2(LE - ZF) = 2N_0 \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} S_h^{-1}(e^{j\omega T}) d\omega = 2N_0 c_0
\]

The latter form assumes a transversal (FIR) filter model for the equalizer, \( c_0 \) being the center tap value. This form is obtained, because the integral is actually the inverse Fourier Transform of \( C(z) \) at \( k = 0 \).

The noise power spectrum at the equalizer output is not flat, which is one indication of the fact that LE, with a slicer following it, is not an optimal solution for the channel equalization problem in the ML sense.
Example of LE-ZF

In this example, we assume an exponentially decaying real impulse response for the equivalent channel:

\[ h(t) = \sqrt{2}ae^{-at}u(t) \]

The impulse and frequency responses of the discrete-time model are (see Exercise 9):

\[ \rho_h(k) = A^{|k|}, \quad A = e^{-aT}, \quad |A| < 1 \]

\[ S_h(z) = \frac{1 - A^2}{(1 - Az)(1 - Az^{-1})} \]

The equalizer transfer function becomes:

\[ C(z) = S_h^{-1}(z) = -\frac{A}{1 - A^2}z^{-1} + \frac{1 + A^2}{1 - A^2} - \frac{A}{1 - A^2}z \]

and the mean-squared error of the equalizer can be written as:

\[ \varepsilon^2(LE - ZF) = 2N_0 \frac{1 + A^2}{1 - A^2} \]

In this case, the reduction in performance, in comparison to isolated symbol detection (the so-called matched filter bound) is:

\[ \frac{\varepsilon^2(LE - ZF)}{\varepsilon^2(MF - ZF)} = \frac{1 + A^2}{1 - A^2} \]

When \( A \rightarrow \pm 1 \), the performance gets worse.
Mean-Square Error (MSE) Criterion

An equalizer with zero forcing criterion tries to cancel ISI completely, if possible. However, since the target is to minimize the error probabilities, it is better to allow some ISI if the noise enhancement can be simultaneously reduced.

An MSE-equalizer minimizes the mean-squared error, taking into account both the residual ISI and noise effects.

The error is defined as the difference between the equalizer output value and the correct symbol value.

Assumptions:

- Symbols and noise are uncorrelated
- Consecutive symbols are uncorrelated with each other:

\[
E\left[ a_k a_j^* \right] = \begin{cases} \sigma_A^2 & j = k \\ 0 & j \neq k \end{cases}
\]

In this case, the symbol sequence is wide-sense stationary and its power spectrum is white, \( \sigma_A^2 \).
An optimal LE-MSE equalizer has the transfer function:

\[ C(z) = \frac{1}{S_h(z) + 2N_0 / \sigma^2_A} \]

This can be easily proven as follows (from Lee & Messerschmitt):

**Linear Equalizer (LE-MSE)**

We can minimize the MSE in figure 8-12b with the help of figure 8-32. The linear equalizer is assumed to have transfer function \( C(e^{j\omega T}) \), \( E_k \) is the error between the slicer input and the transmitted data symbol, which includes the residual ISI and the enhanced noise. The transfer functions for ISI and noise are shown separately in figure 8-32b. It follows that the power spectrum of the error is given by

\[ S_e(e^{j\omega T}) = \sigma_A^2 \left| 1 - S_h(e^{j\omega T})C(e^{j\omega T}) \right|^2 + 2N_0 S_h(e^{j\omega T}) C(e^{j\omega T}) + C(e^{j\omega T})^2 \]  \((8.209)\)

The total MSE will be the integral of \((8.209)\). Since there is no constraint on equalizer complexity, the equalizer transfer function can be chosen independently at each frequency, and we can minimize that MSE by minimizing \((8.209)\) at each frequency with the choice of the linear equalizer \( C(e^{j\omega T}) \). There are several equivalent ways of doing this, but our own favorite method is to complete the square. Write \((8.209)\) in the form (suppressing the \( e^{j\omega T} \) in every term)

\[ S_e(e^{j\omega T}) = S_h(\sigma_A^2 S_h + 2N_0) \left| C - \frac{\sigma_A^2}{\sigma_A^2 S_h + 2N_0} \right|^2 + \frac{\sigma_A^2 2N_0}{\sigma_A^2 S_h + 2N_0} \]  \((8.210)\)

![Figure 8-32. Intermediate steps in the calculation of the MSE for a linear equalizer \( C(z) \).](a) Generation of the error signal \( E_k \). (b) Equivalent system showing explicitly the transfer function from the data symbols and noise to the error.)
In this solution, it is guaranteed that the denominator of the transfer function always takes a nonzero, positive value, since $S_h(e^{j\omega})$ is a positive real function.

MSE at the equalizer output is:

$$\varepsilon^2(LE - MSE) = 2N_0 \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \frac{1}{S_h(e^{j\omega T}) + 2N_0 / \sigma_A^2} d\omega = 2N_0c_0$$

In general: $\varepsilon^2(MF - ZF) \leq \varepsilon^2(LE - MSE) \leq \varepsilon^2(LE - ZF)$
Transversal Filter as an Equalizer

The linear equalizer solutions developed above have rational transfer functions, if the aliased spectrum of the received pulse shape is rational. If the resulting equalizer transfer function is stable, it could be realized as an IIR filter.

However, such an equalizer with IIR transfer function would be difficult to realize in adaptive way. Therefore, channel equalizers are usually implemented as tapped delay lines, i.e., transversal filters, with transfer function is of the form:

\[ C(z) = \sum_{k=-N}^{N} c_k z^{-k} \]

And since digital signal processing is the practical way of implementing discrete-time signal processing, linear equalizers are in practice implemented as digital FIR filters.

In practical implementation, the length of the FIR filter has to be finite, which may involve truncated approximation of the ideal solution. Furthermore, a delay has to be added to make the filter causal:

\[ z^{-N} C(z) = \sum_{k=0}^{2N} c_k z^{-k} \cdot \]

However, as usual in this course, non-causal forms may be used in the developments for notational simplicity.

In practice, equalizers are optimised/adapted using the MSE criterion. Sometimes, also the peak distortion criterion is considered, which basically maximises the vertical opening of the eye diagram.
Problems with Symbol Rate Sampling

As discussed before, the sampled matched filter, with symbol rate sampling, is optimal pre-processing for channel equalizers. However, it has some problems from practical implementation point of view:

1. The channel matched filter part should be implemented as an adaptive analog filter, which is difficult in practice.

2. It assumes that sampling can be done in correct phase.

   However, if there is a timing offset in the sampling clock, so that the samples are \( h(kT - t_0) \), then the aliased spectrum is:

\[
S_{h,t_0}(e^{j2\pi fT}) = \frac{1}{T} e^{j2\pi f t_0} \sum_{m=-\infty}^{\infty} \left| H\left(f + \frac{m}{T}\right) \right|^2 e^{j2\pi m t_0/T}
\]

Assuming a roll-off of no more than 100 %, there are only two overlapping terms at each transition band, and we can write for the range \( 0 \leq f \leq 1/(2T) \):

\[
\left| S_{h,t_0}(e^{j2\pi fT}) \right| = \frac{1}{T} \left| H(f) \right|^2 \left( 1 + \alpha^2 + 2\alpha \cos(2\pi t_0 / T) \right)
\]

\[
\alpha = \frac{|H(j(f - 1/T))|^2}{|H(f)|^2}
\]

In case of correct sampling phase, the baseband spectrum and its first image add up constructively, in phase. Sampling phase error destroys this property, and may cause deep notches in the spectrum around the center of the transition band. This would result in loss of some useful information in the received signal.

In case of incomplete matched filter, same kind of effects may appear, independently of the sampling phase.

One possible way to solve these problems is to use higher than symbol-rate sampling.
Fractionally Spaced Equalizer (FSE)

Assumption: Excess bandwidth < 100%. (In practice, higher values are not interesting, so this is not an essential limitation.)

⇒ The baseband signal bandwidth < \(1/T\).

We consider using the following structure:

Here the sampling is done at twice the symbol rate and (an adaptive) digital filter is used for implementing the matched filter section.

The analog filter preceding the sampler has the task of attenuating the out-of-band signal and noise components before they are sampled on top of the baseband.

At the output of the discrete-time matched filter, the sampling rate is reduced to the symbol rate.

It is easy to see that the input of the linear equalizer has sufficient statistics (since, in principle it is the same as the sampled matched filter output).

In the implementation, the discrete-time matched filter can be combined with the linear equalizer, to form the so-called fractionally-spaced equalizer (FSE):
Fractionally Spaced Equalizer (continued)

FSE is an application of multirate signal processing in communication systems.

From communication theoretic point of view, there is usually no reason for going beyond twice the symbol rate sampling. However, this may be justified by implementation reasons:

- The analog filter section is simplified if the sampling rate is increased.
- There is also increased tendency to use DSP for channel selection filtering in order to increase the flexibility of the receiver, e.g., in case of multistandard systems (“software defined radio”).
- The widely used delta-sigma AD conversion technology is based on heavy oversampling and low number of bits in the initial ADC.

In general, due to the fast development of digital VLSI technologies, the role of multirate signal processing is increasing heavily in the receiver signal processing.
This is a nonlinear equalizer principle.

It includes a feed-forward section $C(z)$, which is similar to LE or FSE, and a feedback from the decided symbols.

The basic idea:

- The feed-forward section removes the pre-cursor ISI, i.e., the ISI due to a later symbol to the preceding symbols.

The pulse shape after this is as follows:

- The feedback section removes the post-cursor ISI, i.e., the ISI from a symbol to later symbols.
The advantage of DFE is that noise enhancement is reduced when compared to LE. This is because the feedback section doesn’t enhance noise and the noise enhancement of the feed-forward section can be reduced.

However, error accumulation reduces the performance: A detection error causes imperfect cancellation of post-cursor ISI for some of the following symbols, which may cause additional detection errors.

Once the feed-forward section of a DFE is defined, the feedback section is obtained in a straightforward way from the impulse response:

This solution removes the post-cursor ISI completely. It is an optimal solution for the feedback section also in the MSE sense.

The feed-forward section can be design using either the MSE criterion or the zero forcing criterion.
Zero-Forcing DFE

When developing the optimal solution, we assume the feed-forward part to be composed of two sections (in the actual implementation, the sections are combined):

- Zero-forcing LE (or FSE)
- Transfer function \((1+D(z))\)

Assuming that the decisions are correct, we can develop the block diagram in the following way:

The final form shows the equivalent transfer functions for the signal and noise.

The symbols are transmitted as they are, independently of \(D(z)\). The ISI is removed completely.

The noise transfer function depends on \(D(z)\).
The noise at the input of $(1 + D(z))$ is coloured. The problem is now to optimize $(1 + D(z))$ such that the noise variance at the slicer is minimized. It can be shown that the optimal solution whitens the noise:

$$\frac{2N_0}{S_h(e^{j\omega})} \left| (1 + D(e^{j\omega})) \right|^2 = \varepsilon^2 (DFE - ZF) \quad \text{(constant)}$$

(This is actually a rather generic problem in statistical signal processing, and the solution for $-D(z)$ is the so-called *linear predictor.*)

In this case

$$\left| (1 + D(e^{j\omega T})) \right|^2 = (1 + D(e^{j\omega T}))(1 + D^*(e^{-j\omega T}))$$

$$= \frac{\varepsilon^2 (DFE - ZF)}{2N_0} S_h(e^{j\omega T})$$
Zero-Forcing DFE (continued)

When the aliased spectrum $S_h(e^{j\omega T})$ is known, the problem is to find the transfer function $D(z)$ and the corresponding complex-valued impulse response $d_k$, that satisfies the following conditions:

\[
\begin{align*}
    d_k & = 0 \quad \text{for} \quad k < 0 \quad \text{(causal filter)} \\
    d_0 & = 0 \quad \text{(no delay free loops)}
\end{align*}
\]

If $S_h(z)$ is a rational function,

\[
S_h(z) = K \frac{\prod_{i=1}^{m}(1-a_i z^{-1})(1-a_i^* z)}{\prod_{i=1}^{n}(1-b_i z^{-1})(1-b_i^* z)}
\]

then

\[
1 + D(z) = \frac{\prod_{i=1}^{m}(1-a_i z^{-1})}{\prod_{i=1}^{n}(1-b_i z^{-1})}
\]

To obtain a stable, causal transfer function, poles and zeros inside the unit circle are chosen from each pair, according to the stability theory of $z$ transfer functions.

The noise variance becomes:

\[
\epsilon^2 (DFE - ZF) = \frac{2N_0}{K}
\]
Zero-Forcing DFE  \((continued)\)

The transfer function of the feed-forward filter becomes:

\[
C(z) = \frac{1 + D(z)}{S_h(z)} = \frac{1/K}{1 + D^*(1/z^*)}
\]

This is an anti-causal (maximum-phase) filter. In practical implementations, it has to be approximated with a finite-length filter and delayed to make it causal.
Zero-Forcing DFE - Example

In the previous example case,

\[ S_h(z) = \frac{1 - A^2}{(1 - Az)(1 - Az^{-1})} \quad |A| < 1 \]

\[ 1 + D(z) = \frac{1}{1 - Az^{-1}} \quad D(z) = \frac{Az^{-1}}{1 - Az^{-1}} \quad C(z) = \frac{1 - Az}{1 - A^2} \]

\[ \epsilon^2 (DFE - ZF) = 2N_0 \frac{1}{1 - A^2} \]

The noise variance is (slightly, up to 3 dB) smaller than in the LE-ZF case:

\[
\frac{\epsilon^2 (LE - ZF)}{\epsilon^2 (DFE - ZF)} = 1 + A^2
\]

This was not a particularly good example for the DFE. There are many practical examples where the DFE provides significantly better performance than the LE, e.g., in heavily frequency selective cases.
**DFE-MSE**

The DFE can also be designed according to the MSE criterion. The resulting solution is quite similar to the DFE-ZF case. The only difference is in the coefficient values.

Of course, in the MSE-sense, the DFE-MSE must be at least as good (and usually clearly better) than DFE-ZF.

\[
\varepsilon^2(DFE - MSE) \leq \varepsilon^2(DFE - ZF)
\]

\[
\varepsilon^2(MF - MSE) \leq \varepsilon^2(DFE - MSE) \leq \varepsilon^2(LE - MSE)
\]

**Concluding Remarks for DFE**

The central problem in DFE is the error propagation.

On the other hand, the noise enhancement effects are much smaller than in LEs, and therefore this structure is commonly used, also in case of heavily frequency selective cases.
ML Sequence Detection (MLSD)

Earlier we have seen that ML sequence detection based on the Viterbi algorithm is optimal when the noise generation process is additive Gaussian noise with independent samples, i.e., in the AWGN case.

With an ISI channel, the samples at the output of the sampled matched filter do not satisfy the independence/whiteness condition, so placing a Viterbi detector (the form which we have been using) after the sampled matched filter is not an optimal solution.

However, in the DFE case we found out a structure that removes the correlation of the samples, and makes the noise white again (at the slicer in the DFE case). We can call this structure as whitened matched filter. It is easy to see that the output of the whitened matched filter has the sufficient statistics.

Now we can apply the ML sequence detection to the output of the symbol-rate sampled sequence at the whitened matched filter output. This provides a solution to the channel equalization problem that is optimal in the ML sequence detection sense.
ML Sequence Detection (continued)

The receiver structure and equivalent discrete-time model:

\[
y(t) \xrightarrow{e^{-j\omega t}} H'(f) \xrightarrow{h(-t)} t = kT \xrightarrow{1/K(1 + D'(1/z'))} u_k \xrightarrow{\text{ML SEQUENCE DETECTION}} \hat{a}_k
\]

\[
a_k \xrightarrow{1 + D(z)} u_k
\]

\[
S_n(z) = \varepsilon^2(DFE-ZF)
\]

Here

\[
E\left[ u_k \right] = a_k + \sum_{j \geq 1} d_j a_{k-j}; \quad \sigma_u^2 = \varepsilon^2(DFE-ZF).
\]

An ML sequence detector chooses the symbol sequence \( \bar{a}_k \) that minimizes the distance:

\[
\sum_k \left| u_k - \bar{a}_k - \sum_{j \geq 1} d_j \bar{a}_{k-j} \right|^2
\]

In practical applications, with unknown and time-varying channel, the receiver must include also a channel estimator that provides the coefficients, \( \{d_j\} \), to the ML sequence detector.
Using the Viterbi Algorithm

An MLSD receiver is usually based on the Viterbi algorithm. (However, there are also other algorithms for the same purpose!)

In practical implementation, the used symbol-spaced channel model (which includes the effects of transmit filter, channel, receiver matched filter and noise whitening filter) should be truncated to obtain a finite-length FIR filter model, with length $L$. The distance function can be written as:

$$\left( \sum_k \left| u_k - \bar{a}_k - \sum_{j=1}^{L} d_j \bar{a}_k - j \right|^2 \right)$$

The Lee&Messerschmitt book includes the error probability analysis of MLSD-based receivers, which is based on the ideas described earlier. The general conclusion is that the error probability of MLSD-based receiver is bounded from below by the single symbol matched filter bound and from above by the DFE error probability.

In the Proakis book, some additional information can be found for performance analysis: It gives a table of worst case channels with given length, with $L=2, \ldots, 6$, as well as the performance loss in dBs with respect to the ISI free channel.
Comparison of DFE and MLSD Based Receivers

There is a great similarity between the DFE and MLSD based receiver structures presented above. They have exactly the same pre-processing, including the noise whitening filter, i.e., feed-forward part of the DFE.

The main differences are:

- **In MLSD, the symbol sequence is detected as whole.** ISI is removed by (in principle) trying all possible symbol sequences and choosing the one that matches best with the received and pre-processed sequence.

- **DFE can be thought as a simplification/approximation of the MLSD receiver, where the symbol is detected right away, and post-cursor ISI is removed based on the earlier decisions.**

Let us look again the equivalent impulse response, which includes the whitened matched filter and where the pre-cursor ISI is removed (or in practice, minimized in the MSE sense):

- **DFE uses only the energy of the first sample (which may have only a small part of the post-cursor energy).**

- **MLSD uses (in principle) the whole post-cursor energy.**
The Channel Equalization Problem in Broadband Wireless Communications

The complexity of an MLSD-based receiver depends heavily on the alphabet size, \(2^B\), and the length of the channel model, \(L\). It is easy to see that the number of parallel states (nodes) in the trellis diagram is

\[ N = 2^{B(L-1)}. \]

For example, in the GSM case (where Viterbi detection is commonly used), the value of \(L=7\) corresponds to about 25 \(\mu\)s delay spread, that is quite sufficient in most macrocell environments. In this case, the number of states is 64, which is a small number for today’s implementation technologies.

But let us consider a case with higher data rate, say 8 Mbps, and assuming 3 \(\mu\)s delay spread (this a realistic value for an urban macrocell). Consider two cases:

- Binary modulation (e.g., PSK), with 8 MHz symbol rate. Then the delay spread is 24 symbol intervals, \(L=25\), and the Viterbi detector would have \(2^{24}\) states, which a quite an unrealistic number from practical implementation point of view.

- 16 QAM modulation, with 2 MHz symbol rate. In this case, we have \(L=7\), and the Viterbi detector would have \(16^6=2^{24}\) states.

It is easy to see that the number of states is actually \(2^b\) where \(b\) is the number of bits that are ‘in the air’ during a time interval of length equal to the delays spread.
From the previous examples, we can see that it is, in practice, impossible to use the ideal MLSD principle in case of wideband mobile/wireless communication systems. For such systems, what could be used are DFE, and certain combinations of DFE and MLSD. One clear general idea in this direction is the following combination:

- Use the MLSD principle (Viterbi) for the few first, most significant samples of the equivalent impulse response with minimized precursor ISI.
- Use the DFE principle to cancel the post-cursor ISI due to the last samples of the impulse response.

In broadband wireless channels, the signal bandwidth may be much higher than the coherence bandwidth of the channel and there is a lot of frequency diversity available (i.e., while some frequencies are severely faded, others have very good signal level). In such cases, even linear equalization may provide sufficient performance. However, a time-domain linear equalizer would have tens or even hundreds of taps, which is quite impractical. But, using frequency-domain linear equalization, based on FFTs and IFFTs, gives a practical equalization approach for single-carrier system in broadband transmission.

Completely different approaches to manage the channel equalization problem are:

- Spread-spectrum techniques, especially DS-CDMA.
- Multicarrier techniques.
In this section we study practical equalization techniques, which take into account the following aspects:

- Limited implementation complexity by using finite-length FIR filters.
- Since the channel response is usually not known in advance, adaptive algorithms are used to compute the filter coefficients.

The basic equalizer structures (LE, FSE, DFE, MLSD) presented above are used anyway.

Depending on the channel characteristics, the coefficient updating can be done in one of the following ways:

- During the modem/system initialization (possibly with re-initialization when the system performance is detected to be unsatisfactory).
- At regular intervals, e.g., once for each transmission burst in burst-mode communications.
- Continuously during normal operation.

In most practical cases, the filtering (and other processing) in the transmitter is fixed and the receiver filter(s) are adapted. However, in point-to-point transmission links, adaptation of the transmitter signal processing is possible.

In certain wireline modem technologies, this feature is commonly utilized in the form of Tomlinson-Harashima precoding.

This kind of precoding ideas enable to utilize the channel capacity more efficiently than the more traditional approaches and, therefore, they are becoming interesting also in the development of future wireless communication technologies.
Adaptive Channel Equalization (continued)

Generally, in addition to the blocks of the signal path (filter stages, Viterbi algorithm, etc.), adaptive equalization includes one or both of the following tasks:

- **Channel estimation**: Determining the channel characteristics (either in time or frequency domain) from the received signal.
- **Determining the coefficients of the equalizer filters**, either based on the channel estimate or directly by minimizing certain error criterion.

From the basic equalizer configurations considered here, MLSD is based on the estimation of the channel impulse response (CIR) and Viterbi algorithm, whereas LE, FSE, DFE are based on the minimization of the MSE at the slicer input.

An adaptive equalizer can be designed to operate in one of the following modes:

- **Training symbol based (data-aided, DA)**: Short known training symbol sequences are included in the transmitted signal and these are used as the reference in the equalizer adaptation.
- **Decision-directed (DD)**: The detected symbols are used as the reference in the equalizer adaptation.
- **Blind equalization**: No training sequences are used. Basically, DD methods could be considered blind, but usually the term refers to certain advanced statistical signal processing based techniques for equalization (that are outside the scope of this course).
In the following, we focus on adaptive implementation of the linear equalizer (LE), but the DFE case is also considered briefly. The overall structure is as follows:

The receiver filtering part is normally not adapted, so in case of symbol rate equalization, the structure is not optimal. However, FSE-based DA algorithms with sufficiently long equalizers approach the optimum MSE performance.

The above structure includes both training symbol based and DD algorithms. One viable alternative is to use the training symbol based adaptation during the initialization phase, and DD adaptation later during normal operation. (DD algorithms may not converge, when the initial state is far from the optimum equalizer coefficients.)

The adaptation algorithms used in training symbol based and DD adaptation are the same, the choice of the operation mode doesn’t have much significance in the continuation.

Anyway, it should be understood that DD based systems work reliably only in reasonably good SNR/BER operation conditions, and would have great difficulties in case of fading channels.
Adaptive Equalizer Structure

The following generic structure represents both the LE and DFE cases (in the DD mode). Naturally, the feedback section is omitted in the LE case:

The error is

\[ E_k = A_k - Q_k \]

In the analysis of the DD mode, it is commonly used that there are no detection errors.

The following notation and indexing is used for the FIR filter sections:

LE:

\[ C(z) = \sum_{m=-L}^{L} c_m z^{-m} \]

DFE:

\[ C(z) = \sum_{m=-N}^{N-1} c_m z^{-m} \]

\[ D(z) = \sum_{m=1}^{M} d_m z^{-m} \]

In the implementation, \( C(z) \) is made causal by including a suitable delay.
Matrix Representation for a Linear Equalizer

\[ c' = [c_{-L}, \ldots, c_L] \quad r'_k = [R_{k+L}, \ldots, R_k, \ldots, R_{k-L}] \]

\[ E_k = A_k - Q_k \quad Q_k = c'r_k \]

\[ E\left[|E_k|^2\right] = E\left[|A_k|^2\right] - 2 \text{Re}\left\{ c'^* \alpha \right\} + c'^* \Phi c \]

\[ \alpha = E\left[A_k r_k^*\right] \]

\[ \Phi = E\left[r_k^* r'_k\right] = \begin{bmatrix} \phi_0 & \phi_1 & \cdots & \phi_{N-1} \\ \phi_{-1} & \phi_0 & \phi_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{-(N-1)} & \phi_{-(N-2)} & \cdots & \phi_0 \end{bmatrix} \quad \phi_j = E\left[R_{k+j} R_k^*\right] \]
Matrix Representation for a Linear Equalizer (cont.)

Above

- $c$ is the equalizer coefficient vector to be optimized; length $N = 2L + 1$
- $r_k$ is the vector of equalizer input samples appearing in the delay line at time $k$.
- $\Phi$ is the autocorrelation matrix of the input signal (Hermitian and Toeplitz matrix, normally positive definite and non-singular).
- $\alpha$ is the cross-correlation vector between the transmitted symbol and the equalizer input samples.
- $E\left[|E_k|^2\right]$ is the MSE to be minimized.

A basic assumption is that all the random processes are wide-sense stationary and their statistical properties are known.
The MSE can be expressed as:

\[
E\left[|E_k|^2\right] = E\left[|A_k|^2\right] - \alpha^* \Phi^{-1} \alpha + \left(\Phi^{-1} \alpha - c\right)^* \Phi \left(\Phi^{-1} \alpha - c\right)
\]

Here only the last term depends on the vector to be optimised. The last term is also non-negative, as a quadratic form of a positive definite matrix. The optimum solution is thus:

\[
c_{opt} = \Phi^{-1} \alpha
\]

So the optimal equalizer tap coefficients can be found by solving a set of linear equations. The solution exists if the autocorrelation matrix is non-singular.

The MSE can now be expressed as:

\[
E\left[|E_k|^2\right] = \xi_{\text{min}} + \left(c - c_{\text{opt}}\right)^* \Phi \left(c - c_{\text{opt}}\right)
\]

The optimum equalizer coefficient vector is also characterized by the following alternative conditions (which are of some use in the continuation):

Zero-gradient condition: \[
\nabla_c E\left[|E_k|^2\right] = 2\Phi c_{\text{opt}} - 2\alpha = 0
\]

Orthogonality condition: \[
E\left[E_k r_k^*\right] = \alpha - \Phi c_{\text{opt}} = 0
\]
MSE Gradient Algorithm

We have seen that the optimal equalizer coefficient vector of length $N$ can be found by solving a set of linear equations with $N$ unknowns.

This is, however, far too complicated computationally as adaptive receiver algorithm. Next we look for lighter methods based on iterative algorithms.

In the MSE gradient algorithm (MSEG), the principle is to move "down-hill", in the direction of the negative gradient of the MSE function.

During each iteration cycle, the movement is proportional to the step-size parameter $\beta$, so the new coefficient vector is:

$$c_{j+1} = c_j - \frac{\beta}{2} \nabla c_j E\left[|E_k|^2\right]$$

$$= c_j + \beta(\alpha - \Phi c_j) = (I - \beta \Phi)c_j + \beta \alpha$$

During iteration cycle $j+1$, a new coefficient vector $c_{j+1}$ is found, and it is closer to the optimum as the previous one, assuming that the step size is not too large.
The convergence of the algorithm depends greatly on the eigenvalues $\lambda_1, \ldots, \lambda_N$ of the autocorrelation matrix $\Phi$. It can be shown (see Lee&Messerschmitt), that the algorithm converges iff

$$0 < \beta < \frac{2}{\lambda_{\text{max}}}$$

and the optimal step size is

$$\beta_{\text{opt}} = \frac{2}{\lambda_{\text{min}} + \lambda_{\text{max}}}.$$ 

(If there is a big difference in the eigenvalues, the MSE function is very elliptic, and the convergence is slow.)
LMS Algorithm

Usually, it cannot be assumed that the autocorrelation matrix is known. And even if it were known, it's inversion would be computationally demanding. Next we consider the so-called stochastic gradient algorithm, or LMS algorithm, which doesn’t need this information (nor matrix inversion).

This method is obtained from the previous one by taking the squared magnitude values instead of its expectation so the function to be minimized is:

\[ |E_k|^2 = |A_k|^2 - 2 \text{Re} \left[ A_k c^* r_k^* \right] + c^* r_k r_k' c \]

Its gradient is

\[ \nabla_c |E_k|^2 = -2r_k^* (A_k - r_k' c) = -2E_k r_k^* \]

and the iterative coefficient adaptation algorithm becomes:

\[ c_{k+1} = c_k + \beta E_k r_k^* = \left( I - \beta r_k r_k' \right) c_k + \beta A_k r_k^* . \]

Notice that here the coefficients are updated once per symbol interval, while in the MSEG algorithm, the idea was to find the optimum point iteratively for every symbol interval.

Even though a crude approximation was made to arrive at the LMS algorithm, it uses an unbiased estimate for the gradient, which in the stationary situation leads towards the optimum solution.
The adaptation algorithm for coefficient $c_j, -L \leq j \leq L$, is:

$$\left[c_{k+1}\right]_j = \left[c_k\right]_j + \beta E_k R_k^* - j$$

This can be realized according to the following block diagram:

The calculation is simple and straightforward.
About the Convergence of the LMS Algorithm

In the LMS algorithm, the trajectory of the coefficient vector is random. However, if an average trajectory is defined by taking expectations, it converges in the stationary case towards the optimum given that the step-size fulfills the previous stability condition. Of course, in any particular algorithm realization, the trajectory doesn’t converge all the way to the optimum, but after the convergence remains fluctuating randomly around the optimum point.

It is clear that the amount of random variations is proportional to the step-size parameter $\beta$. To make the random variation reasonably small, the value of $\beta$ should be chosen much smaller than in the MSEG algorithm.

On the other hand, with larger value of $\beta$, the algorithm is able to follow faster changes in the channel characteristics.

When choosing the value of $\beta$ a tradeoff must be made between these two aspects.

The MSE can be expressed as:

$$E\left[|E_k|^2\right] = \xi_{\text{min}} + E\left[\mathbf{q}_k^*\Phi\mathbf{q}_k\right]$$

The 'excess MSE' due to the latter term depends on the error (wrt. $\mathbf{c}_{\text{opt}}$) in the coefficient vector:

$$\mathbf{q}_k = \mathbf{c}_k - \mathbf{c}_{\text{opt}}$$
About the Convergence of the LMS Algorithm (cont)

Let us consider the case where the input samples are uncorrelated and zero-mean. Then the autocorrelation matrix is diagonal:

$$\Phi = \phi_0 I, \phi_0 = E\left[|R_k|^2\right]$$

The MSE can then be written as:

$$E\left[|E_k|^2\right] = \xi_{\text{min}} + \phi_0 E\left[\|q_k\|^2\right]$$

The error in the coefficient vector can be written as:

$$E\left[\|q_{k+1}\|^2\right] = \gamma E\left[\|q_k\|^2\right] + \beta^2 N \phi_0 \xi_{\text{min}}$$

and the stability condition becomes:

$$|\gamma| < 1.$$ 

The smaller $\gamma$ is the faster convergence can be expected. Since $\gamma$ is a second-order function of the step-size $\beta$, it has an optimum point with respect to the convergence speed:

$$\beta_{\text{opt}} = \frac{1}{N \phi_0}.$$ 

The condition for convergence is:

$$0 < \beta < \frac{2}{N \phi_0} = 2 \beta_{\text{opt}}.$$

This is much more restrictive than the corresponding condition for MSEG.
The algorithm converges exponentially and it has the time constant:

\[ \tau = \frac{1}{2 \beta \phi_0} \]

The minimum value of the time constant is:

\[ \tau_{\text{opt}} = \frac{1}{2 \beta_{\text{opt}} \phi_0} = \frac{N}{2} \]

So we see that the convergence speed depends essentially on the length of the equalizer!

Another essential quantity is the mean-squared error of the coefficient vector, which satisfies:

\[ E \left[ \| q_k \|^2 \right] \rightarrow \frac{N \beta}{2 - N \beta \phi_0} \xi_{\min} \quad \text{for } k \rightarrow \infty \]

For \( \beta = \beta_{\text{opt}} \)

\[ E \left[ \| q_k \|^2 \right] \rightarrow \frac{1}{\phi_0} \xi_{\min} \quad \text{for } k \rightarrow \infty \]

\[ E \left[ |E_k|^2 \right] \rightarrow 2 \xi_{\min} \]
Conclusions:

• If the target is to minimize MSE, the step-size should be chosen as small as possible. The wordlength used in the implementation may set a lower bound on this.

• If maximum convergence speed is required, one should choose $\beta = \beta_{\text{opt}}$. The MSE takes then twice the minimum value.

The analysis and results summarized above can be done easily only in a highly simplified case. However, it has been verified by numerous computer simulations and practical implementations, that the above results hold also in case of arbitrary (yet well-behaved) input signals.
Modifications of the LMS Algorithm

**Normalizing the step size**

In the solution presented above, the signal power has a big effect on the convergence. It is easy to see that $\beta_{opt}\phi_0$ is constant, so the step size should be inversely proportional to the signal power. In practice, it is better to choose:

$$\beta_k = \frac{a}{\sigma_k^2 + b}$$

where $a$ and $b$ are suitable constants and $\sigma_k^2$ is the signal power estimate at time $k$. The signal power can be estimated, e.g., as follows:

$$\sigma_k^2 = (1-\alpha) \sum_{j=0}^{\infty} \alpha^j |R_{k-j}|^2$$

This can be calculated recursively:

$$\sigma_k^2 = \alpha \sigma_{k-1}^2 + (1-\alpha)|R_k|^2$$

**Gear-Shift algorithms**:

The idea is to use a reasonably high step size in the beginning, and reduce it as the convergence progresses. Various methods for the adjustment can be considered (e.g., based on constant timing, estimation of the MSE, or monitoring the changes in the coefficient values.) This idea works well, if the changes in the channel conditions are slow.
Adaptive DFE

When using finite-length transversal filters, the DFE block diagram can be drawn as:

The structure resembles very much the linear equalizer. The difference is that the detected symbol sequence is used whenever it is possible, i.e., in the causal part.

Now the signal at the slicer can be written as:

\[
Q_k = 0 \sum_{i=-(N-1)}^{0} c_i R_{k-i} - \sum_{i=1}^{M} d_i \hat{A}_{k-i}
\]
Adaptive MSE Solution for DFE

Assuming that there are no symbol errors ($\hat{A}_k = A_k$) and that the signal and noise are uncorrelated, the system model can be developed as follows:

$$R_k Z_k Q_k A_k P_z C_z D_z()()() - C_z() Z_k A_k P(z)C(z)-D(z) Q_k$$

The MSE includes two additive parts, one of which is due to the noise and the other due to ISI. The noise term doesn’t depend on $D(z)$.

With this information, the choice of the feedback coefficients is clear: They are chosen to force the $M$ first ISI values to zero:

$$d_m = \sum_{i=-N+1}^{0} c_i p_{m-i}, \quad 1 \leq m \leq M$$

The feed-forward coefficients are optimised using the MSE solution of LE, with the additional equation as an additional constraint.
The LMS algorithm is essentially the same as in the LE case:

\[ v' = \left[ c_{-(N-1)} \cdots c_0 d_1 \cdots d_M \right] \]

\[ w'_k = \left[ R_k + (N-1) \cdots R_k A_{k-1} \cdots A_{k-M} \right] \]

\[ E_k = \hat{a}_k - v' w_k \]

\[ v_{k+1} = v_k + \beta E_k w_k^* \]
Adaptive Fractionally-Spaced Equalizer (FSE)

As discussed earlier, oversampled equalizers have various benefits:

- In case of unknown channel, it is not practical to implement the sampled matched filter (including the analog matched filter as well as the critical symbol timing functionality) in the optimal way. Adaptive FSE implements this part implicitly in an adaptive way, and the problems are avoided.

- The matched filter doubles the channel attenuation in attenuated parts of the spectrum. This increases the dynamic range (and wordlength) requirements in symbol-rate equalizers. Again, the problem is avoided in FSE because it implements the cascade of matched filter and equalizer as a combined block.

In principle, an adaptive fractionally spaced LE or DFE can be implemented in the same way as the previously presented symbol-rate equalizers. The input samples are coming (usually) at twice the symbol rate, so the length of the equalizer needs to be doubled to get the same performance. But the output is computed only at symbol rate. The adaptation is based on the output samples.

However, there are some special concerns in FSE solution, as discussed below.
The Stability of Adaptive FSE

FSEs may have some stability problems:

- Some parts of the extended frequency band (especially the parts outside the desired signal band which have been attenuated by earlier receiver filter stages) may have practically zero spectrum.

⇒ This causes the autocorrelation matrix to become singular, and there is no unique solution to the MSE minimization task.

⇒ The algorithm may drift to a state where some of the coefficient values are larger than the used number system allows. This can be illustrated as follows:

The solution tends to drift in the direction where the sensitivity of the MSE to the coefficient values is smallest.
The following alternative ideas have been used for solving the stability problem of adaptive FSEs:

- The coefficients are forced (saturated) to stay in the used number range.
- By adding some noise to the input signal (dithering), the empty parts of the spectrum are avoided. When minimizing the MSE, the equalizer attenuates the out-of-band part of the added noise. However, some part of the dither noise ends up at the slicer and degrades the performance.
- Minimizing, instead of MSE, the function

\[ E\left|E_k\right|^2 + \mu\|c\|^2 \]

where \(\mu\) is a suitable constant. In addition to minimizing the MSE, the algorithm tries then to minimize also the length of the coefficient vector. Also this approach degrades the performance somewhat from the optimal.
Bandpass Equalizers

The basic equalizer structures presented above have some limitations when considering the overall signal processing architecture of a receiver. One example is that the equalizer may be located inside the synchronization loop(s) of the receiver, in which case the synchronization and equalization functionalities would be effecting each others, possibly in a way that could be difficult to control.

There are certain variations, referred to as bandpass equalizers that provide some additional flexibility in designing the overall structure.

We consider a model where the receiver LO frequency is $\omega_1$ instead of $\omega_c$, so the frequency offset is $\Delta \omega = \omega_c - \omega_1$.

Equivalent system models:
In the (c) case,

\[ A_k = A_k e^{j\Delta \omega kT} \]

is called the rotated symbol sequence and

\[ h(t) = h(t) e^{j\Delta \omega kT} \]

is the impulse response of a passband channel.

Example of rotated constellation in the 16QAM case:
Bandpass Equalizers (continued)

Bandpass equalizer structure, two equivalent forms:

![Diagram of Bandpass Equalizer Structure]

The structure uses the rotated symbols $\bar{A}_k$ and the error sequence for equalizer adaptation is:

$$\bar{E}_k = E_k e^{j\Delta \omega kT}.$$  

It tries to equalize the passband channel $\bar{h}(t)$ and the carrier phase offset is corrected after the equalizer.

Otherwise the realization principles are the same as discussed before.

This general model can be used in different ways:

- $\omega_1 = \omega_c$ ($\Delta \omega = 0$) : This is the special case of baseband equalizer.

- $\omega_1 = 0$ ($\Delta \omega = \omega_c$) : This is the true passband equalizer case (that has limited use in practice).

- $\omega_1 \approx \omega_c$ ($\Delta \omega \approx 0$) : This can be considered as a modification of the baseband equalizer, where the carrier fine synchronization is done after the equalizer.
Concluding Remarks on Equalizers

In this course, only the basic equalizer structures and algorithms have been presented.

There is huge amount of literature about channel equalizers, presenting numerous variations of the basic structures.

Especially, there are a variety of algorithms providing essentially faster convergence speed than the LMS algorithm, but also clearly higher implementation complexity.

The Recursive Least Squares (RLS) is an important class of algorithms with such features.