Synchronisation in Digital Receivers

In a digital transmission system, the transmission chain includes several oscillators for

• modulation and demodulation
• up- and down-conversion
• clocking symbol and bit-streams
• sampling

The synchronisation functions of the receiver have to be locked to the received signal. It is not possible to recover correct frequencies/phases without any control mechanisms. The receiver has to ‘dig out’ the synchronisation information from the received signal.

Synchronisation has to be done at least in the following levels:

1. Carrier recovery
2. Symbol timing recovery
3. Frame synchronisation

We concentrate here on the two first ones.
Carrier recovery

Reasons for carrier phase/frequency errors:

- Transmitter local oscillators used for up-conversion and modulation
- Receiver local oscillators used for down-conversion and demodulation
- Up/down-conversion in repeaters
- Doppler shift in mobile channels and non-geostationary satellite systems

In the traditional approach, the receiver local oscillator(s) are adjusted adaptively to match the frequency and phase with the received signal.

- In differentially coherent (e.g., DPSK) or non-coherent (e.g., FSK) systems, sufficiently accurate frequency adjustment is enough
- Accurate phase recovery is needed in coherent detection
- For complex alphabets, accurate phases for the quadrature carriers are needed (90 degree phase shift for I & Q branches)

Carrier recovery is often done in two parts:
1. Coarse frequency adjustment
2. Fine frequency adjustment and phase recovery
Effects of carrier synchronisation errors

- **Constant carrier phase error** $\phi$ corresponds to a rotation of the constellation

$$A_k = e^{j\phi}A_k$$

- **Constant carrier frequency error** $\Delta \omega$ corresponds to time-varying rotation of the constellation

$$A_k = e^{jkT\Delta \omega}A_k$$

If the parameters can be estimated with sufficient accuracy, these effects can easily be compensated by a complex multiplier in baseband processing, prior to detection.
Symbol timing recovery

In the traditional approach, the sampling clock is synchronised to the symbol frequency and phase of the received signal.

The effects of symbol timing errors can be seen with the aid of the eye pattern:

- the samples are not taken at the maximum eye opening
- ISI is introduced
- noise margin is reduced
Synchroniser classes

Analog recovery

Hybrid recovery

Digital recovery
Analog and Digital Solutions

Traditionally, analog or hybrid analog/digital solutions have been used for carrier and timing recovery.

- Adjusting directly the demodulator local oscillator or sampling clock
- Phase-locked loops (PLLs) and related circuits have been used as key elements.

There is interest in digital solutions:

- Possibly better in terms of size or power consumption or economy
- Better solutions can be found in terms of jitter performance and/or tracking speed
  - Convergence speed is very important in mobile systems
  - Long feedback loops can be avoided (e.g., in hybrid solutions, loops from the digital part back to the analog part are needed)
    => speed

In many cases, the same algorithms can be adapted to analog, hybrid, or digital implementations.

But there are also cases of digital algorithms which cannot be implemented in analog form reasonably.
All-Digital Synchronisation Concept

- Free-running local oscillators for demodulation and frequency conversion
- Free-running sampling clock
- Errors are compensated in digital part

=> All synchronisation functions can be implemented using digital techniques
Sampling Rate Options

Theoretically in an optimal receiver for a linear transmission system, symbol-rate sampling can be used if and only if

- matched filter is included in the receiver prior to sampling or sampling rate conversion to symbol rate
- sampling or sampling rate conversion to symbol rate is done at the correct phase

Otherwise there will be destructive aliasing effects.

Then it is clear that A/D conversion can be done at the symbol rate only in the analog or hybrid symbol timing recovery schemes. (For the carrier recovery scheme, there are no such hard constraints.)

All-digital symbol timing recovery scheme requires higher-than-symbol-rate sampling.

A natural choice is to sample at twice the symbol rate (since the excess bandwidth is normally less than 100%).

It is also possible to use a sampling rate which is not in simple relation to the symbol rate. In principle

$$f_s \geq (1 + \alpha) / T$$

is sufficient to avoid destructive aliasing effects.

On the other hand, higher oversampling factors may be used in order to get the benefits of multirate filtering to reduce the complexity of the needed analog filtering.
Synchronisation principles

**Data-aided (DA)**

Based on reference symbol sequences known to the receiver (training signals, preambles/midambles, pilot frequencies, etc.).

**Decision-directed (DD)**

Utilises detected symbol values as reference.

**Non-data-aided (NDA)**

Does not depend on known or detected symbol values.

It is clear that DA techniques result in the best synchronisation performance, but some part of bandwidth or data transmission capacity is lost for the pilot signals or training sequences.

Decision-directed methods are sensitive to detection errors (but not very sensitive).
Synchroniser operation modes

Acquisition mode

Start-up phase, during which the system achieves stable operation.

Tracking mode

Normal operation mode, during which the system adapts to slow variations of the synchronisation parameters.

Different synchronisation principles can be used during the two phases.
Quality factors

Convergence speed
Especially in mobile communication systems, the carrier phase recovery and symbol timing recovery functions have to be able to follow very rapid changes.

In TDMA systems, phase recovery and timing recovery are carried out for each burst independently of the others.

Timing error, carrier phase error, carrier frequency error
For example, in case of timing recovery, the estimation errors appear as timing jitter, random fluctuation of the timing strobes due to noise and non-idealities of the algorithm.

The estimation error may include a dc-combonent, a bias, and it has a certain (in practice non-zero) variance.

The errors are partly due to the noise in the received signal and partly due to nonidealties of the algorithm.

Self noise, pattern noise
Additional noise contribution at the detector due to the synchronisation technique used.

This is small, if the system finds after the acquisition phase, a stable state (with zero phase error or symbol timing error) which is independent of the actual received symbol pattern.

Phase noise
Phase noise is partly due to the instability of the oscillators, but it could also be effected by the carrier recovery method.

Hang-up, cycle slip
Different types of misbehaviour of the synchroniser, e.g., locking to a wrong phase or frequency or loosing a symbol or a complete cycle of the carrier.

Symbol-error rate / Bit-error rate
Best measure of the overall performance.
Illustrating the Effect of Timing Jitter on Symbol Error Rate:

![Figure 2.3. Illustrating the shape of $P(e|\hat{t})$ and $p(\hat{t})$.](image)

Examples of the Effects of Symbol Timing Errors on Symbol Error Rates, 2-level and 4-level PAM:
Examples of the Effects of Symbol Timing Errors on Symbol Error Rates

Figure 2.12. SEP degradations due to uncompensated frequency offsets.

Figure 2.14. Effect of timing errors with MSK modulation (with internal detection)

2.9. SEP degradations due to phase errors with QPSK modulation.
Feedback and feed-forward configurations

Symbol timing recovery can be done either in feedback configuration

\[ \text{sampler} \rightarrow \text{VCO} \rightarrow \text{loop filter} \rightarrow \text{timing error detector} \rightarrow \text{to detector} \]

or in feedforward configuration

\[ \text{sampler} \rightarrow \text{timing estimation} \rightarrow \text{VCO} \rightarrow \text{to detector} \]

The same is true also for carrier phase recovery.

Coarse carrier frequency recovery is usually based on feedback configuration.
Comparing feedback and feed-forward configurations

In feedforward configuration, accuracy of the timing estimator determines directly the accuracy of timing recovery.

Feedback algorithms are usually based on simple stochastic approximation approach: timing error detector gives a signal which is a monotonic function of the error.

Based on this signal, the timing is adjusted in such a way that the error is reduced.

The characteristics of the timing error detector can be described with the aid of the so-called S-curve:

Linearity is not a necessary requirement.

The loop filter is usually a single multiplier. A small value is need for it in order to make the self noise or timing jitter small. This leads to slow convergence speed.

In more advanced algorithms, the loop gain is adjusted in such a way that the convergence is fast in the beginning or when the error is large. In stationary operation the gain is small to make the self noise small.
Some properties of synchronisation algorithms

A key trade-off
In many types of carrier and symbol timing recovery algorithms, there is a trade-off between the convergence speed and estimation error variance.

When the parameters are such that the convergence is fast, the estimation error variance is high, and vice versa.

Block-based vs. continuous algorithms
Forward type of timing estimation and phase estimation algorithms are typically block-based, i.e., the estimates are computed for fixed-length blocks samples. The blocks may be partly overlapping or non-overlapping.

Feedback algorithms usually operate continuously.

Symbol rate and higher than symbol rate algorithms
Most timing recovery algorithms assume a sampling rate that is higher than the symbol rate. Recall that in all-digital timing recovery this is a necessity.

However, there are some hybrid timing recovery algorithms that operate on symbol-rate samples.

Relation to channel equalisation
In many cases, the channel equalisation methods are able to correct carrier and timing recovery errors in some extent.

Order of the blocks
The order of the symbol timing and carrier recovery blocks can be selected rather freely.

- Normally the coarse frequency synchronization is done first. Frequency errors which are considerable in comparison to the signal bandwidth would destroy the band-selection and matched filtering.

- Phase rotation and sampling/symbol timing interpolation blocks can be interchanged.

- Different timing estimation / timing error detection and phase estimation / phase error detection methods may be needed depending on the order of the blocks.

For example, if the system includes timing estimation before phase rotation, the timing estimation method should tolerate carrier phase errors.
Examples of Receiver Structures

Figure 2.8. Block diagram of a coherent receiver.

Figure 2.11. Block diagram of a differential receiver.
Order of the blocks in all-digital solutions

(a)

(b)
There are so many different algorithms

Both for timing recovery and for carrier recovery, a very large number of different methods can be found from the literature. There are a lot of ad-hoc methods, derived from different approaches.

A general theoretical framework for developing and optimising algorithms for carrier and symbol timing recovery is the *Maximum Likelihood estimation theory*. 
Some ad-hoc timing recovery algorithms

A set of algorithms utilizes a timing tone, a symbol-rate spectral component, which is obtained, e.g., by squaring the baseband data signal.

The timing tone can be extracted by

- a narrowband lowpass filter
- phase-locked loop
- computing DFT at the symbol frequency (i.e., a single point of the DFT output is needed for each data block)

The timing jitter of these algorithms (and many others) can be improved by a special prefilter, which produces a symmetric, bandlimited spectrum at half of the symbol rate. In fact, in some cases it is possible to achieve jitter-free timing recovery.
The original paper describes a family of hybrid algorithms working at symbol-rate samples. The most common (and most simple one) is the following.

This is a decision-directed (or data-aided) algorithm.

It can be shown to be free of timing jitter in ideal conditions (after convergence to zero timing error, no ISI).

This algorithm was developed originally for 2-PSK and QPSK, but has been shown to work also in the case of high-order QAM-constellations.
Maximum Likelihood Estimation Techniques for Symbol Timing and Carrier Phase Recovery

Markku Renfors, Vesa Tuukkanen*
TUT/Telecommunications

* Presently at Nokia Research Center, Tampere

Maximum Likelihood estimation (MLE) theory provides a unified framework for developing optimal synchronization algorithms for receivers. It can be applied to both open loop and closed loop synchronizers, in data-aided, decision-directed, and non data aided configurations.

**Basics of Maximum Likelihood Estimation**

In Maximum Likelihood estimation, it is required that the transmitted signal $s(t, \Phi_0)$ is known at the receiver in most of its characteristics: nominal carrier frequency $f_0$, nominal symbol rate $1/T$, modulation format, symbol alphabet, and pulse waveshape $h(t)$.

Unknown parameter vector $\Phi_0$ may include carrier phase $\theta_0$, timing delay $\tau_0$, frequency error $\nu_0$, amplitude $A_0$, and complex data sequence $\{a(n), b(n)\}$. The parameter vector $\Phi_0$ is supposed to be unchanging during the observation time $T_0$, and the purpose of MLE is to compute estimates for some or all parameters of the vector.

We consider here symbol timing recovery and carrier phase estimation. When both parameters are to be estimated, joint ML estimation is the optimum approach.

The received noisy signal is

$$r(t) = s(t, \Phi_0) + w(t)$$

where $w(t)$ is white, stationary, Gaussian noise of two-sided spectral density $N_0/2$.

The likelihood function is defined as the pdf of the observations, conditioned on a trial vector $\tilde{\Phi}$

$$L(\tilde{\Phi}) = p\left| \rho(T_0) \right| \tilde{\Phi}$$

where $\rho = \rho(T_0)$ is the measurement result of $r(t)$ in the observation interval.
The aim is to find trial value $\hat{\Phi}$ of the parameter vector which maximizes $L(\hat{\Phi})$,

$$L(\hat{\Phi}) = \max L(\hat{\Phi})$$

When the noise is Gaussian, the likelihood function for the signal is

$$L(\hat{\Phi}) = C_1 \exp \left[ -\frac{C_2}{N_0} \int |r(t) - s(t, \hat{\Phi})|^2 dt \right]$$

where $C_1$ and $C_2$ are positive constants that can be ignored. MLE is trying to minimize the integral.

Expanding the integral yields

$$|r(t) - s(t, \hat{\Phi})|^2 = |r(t)|^2 + |s(t, \hat{\Phi})|^2 - r(t)s^*(t, \hat{\Phi}) - r(t)^*s(t, \hat{\Phi})$$

$$= |r(t)|^2 + |s(t, \hat{\Phi})|^2 - 2\text{Re}[r(t)s^*(t, \hat{\Phi})]$$

The first term represents the energy of the received noisy signal and is independent of $\hat{\Phi}$, so it can be ignored.

The second term is more complicated because it is a function of $\hat{\Phi}$. $|s(t, \hat{\Phi})|^2$ is the energy of the local replica and it can be ignored on condition that the signal $s(t)$ has constant amplitude or the parameter to be estimated is either carrier phase or frequency.

If timing error is the parameter to be estimated, and the signal has constant amplitude, e.g., 2-PAM without pulse shaping, the second term can be regarded as a constant and can be ignored.

After these simplifications the likelihood function is given by

$$L(\hat{\Phi}) = C_3 \exp \left[ \frac{2C_2}{N_0} \int \text{Re}[r(t)s^*(t, \hat{\Phi})] dt \right]$$
Since the logarithm is a monotonic function, a value of $x$ that maximizes $f(x)$ also maximizes $\ln[f(x)]$. In many cases, it is more convenient to maximize the log-likelihood function

$$\Lambda(\Phi) = \ln[L(\Phi)] = \int_{T_0} \text{Re}\left[r(t)\tilde{s}(t,\Phi)^*\right]dt$$

where the constant terms $\ln(C_3)$ and $2C_2/N_0$ have been dropped since they do not influence the maximization.

As can be seen, in this simplified form, the maximum likelihood estimation can be considered as a correlator where correlation between $r(t)$ and $\tilde{s}(t,\Phi)$ is computed at arbitrary values of $\Phi$. MLE is the value of $\Phi$ which maximizes the correlation.

Even though the log likelihood function was derived for constant amplitude signals, it can be used for other systems, like QAM-signals, as an approximation to the log-likelihood function. However, the performance for the multilevel signals is decreased because the term $|\tilde{s}(t,\Phi)|^2$, which was ignored, is not a constant anymore.
Joint DA/DD Symbol Timing and Carrier Phase Estimation for Linear Modulations

In most general form, the transmitted complex signal using linear modulation is given by

$$s(t, \Phi_0) = A_0 \exp(j\theta_0) \sum_n [a_n h(t - nT - \tau_0) + jb_n h(t - nT - \epsilon T - \tau_0)]$$

where $\epsilon$ is the stagger coefficient, and $h(t)$ is the pulse shape of the transmitted signal. Frequency offset is assumed to be zero.

In this case the log likelihood function is

$$\Lambda(\Phi) = \text{Re} \left[ \exp(-j\tilde{\theta}) \sum_1^N \int_{-\infty}^{\infty} r(t) h(t - nT - \tilde{\tau}) \right] dt$$

where $N$ is the number of symbols in the observation interval.

Integrals in the last two lines are output samples of a matched filter in the I and Q branches. Samples are taken once per symbol interval. This leads to the following important result: ML estimate $\hat{\Phi}$ can be extracted from the matched filter samples.

If the received signal is nonstaggered, that is $\epsilon=0$, we find that samples from the output of the matched filter are equal. For $\hat{c}_n = \hat{a}_n + j\hat{b}_n$ we obtain

$$\Lambda(\Phi) = \text{Re} \left[ \exp(-j\tilde{\theta}) \sum_{n=1}^{N} \hat{c}_n^* m(n, \tilde{\tau}) \right]$$

where

$$m(n, \tilde{\tau}) = \int_{-\infty}^{\infty} r(t) h(t - nT - \tilde{\tau}) dt$$

is the matched filter output signal with timing offset $\tilde{\tau}$. 
We can write

\[ \Lambda(\Phi) = \left| \sum_{n=1}^{N} \hat{c}_n^* m(n, \bar{\tau}) \right| \cdot \text{Re} \left[ j \left( \arg \left[ \sum_{n=1}^{N} \hat{c}_n^* m(n, \bar{\tau}) \right] - \bar{\theta} \right) \right] \]

The ML timing estimate is

\[ \hat{\tau} = \arg \max_{\tau} \left| \sum_{n=1}^{N} \hat{c}_n^* m(n, \bar{\tau}) \right| \]

which is independent of the carrier phase.

After finding the timing estimate, the ML carrier phase estimate is

\[ \hat{\theta} = \arg \left( \sum_{n=1}^{N} c_n^* m(n, \hat{\tau}) \right) \]

In this way, the problem of jointly estimating the symbol timing and carrier phase errors can be converted to two one-dimensional optimization problems.

From the implementation point of view, it is more convenient to maximize the squared ML function

\[ |\Lambda(\bar{\tau})|^2 = \left( \sum_{n=1}^{N} \left( \hat{a}_n y_I(n, \bar{\tau}) + \hat{b}_n y_Q(n, \bar{\tau}) \right) \right)^2 + \left( \sum_{n=1}^{N} \left( \hat{a}_n y_Q(n, \bar{\tau}) - \hat{b}_n y_I(n, \bar{\tau}) \right) \right)^2 \]

The structure of the data-aided ML timing estimator for I/Q modulations is:
The principle of the ML estimation procedure for binary PAM:
Derivation of NDA Timing Error Estimators

Non-Data-Aided timing recovery is used when actual symbol values or decisions are not available or they are not reliable. In NDA, symbol values are treated as random values and $\Lambda(\tilde{\tau})$ is simply averaged over these random values prior to maximization. Here we consider only timing estimation, but also the joint timing and carrier phase estimation is possible in the case of m-PSK modulations [2].

In order to derive the log-likelihood function, either the actual probability density function (pdf) or reasonable approximation of the pdf for the symbols has to be known. It has been found that Gaussian distribution of the symbol values yields a good approximation to the log-likelihood function.

**Binary PAM Signal**

The log-likelihood approximation is in this case

$$L(\tilde{\tau}) = \prod_{n=1}^{N} \exp(\hat{a}_n m(n, \tilde{\tau}))$$

Data averaged likelihood function for binary symbols $a = \pm 1$ with equal probability can be written as

$$L_a(\tilde{\tau}) = \int_{-\infty}^{\infty} L(\tilde{\tau}) p(a) da$$

$$= \prod_{n=1}^{N} \left[ \frac{1}{2} \exp(1 \cdot m(n, \tilde{\tau})) + \frac{1}{2} \exp(-1 \cdot m(n, \tilde{\tau})) \right] = \prod_{n=1}^{N} \text{cosh}(m(n, \tilde{\tau}))$$

and the corresponding log-likelihood function is

$$\Lambda_a(\tilde{\tau}) = \sum_{n=1}^{N} \text{ln}\text{cosh}(m(n, \tilde{\tau}))$$

The nonlinear function $\text{ln}\text{cosh}(m(n, \tilde{\tau}))$ is not easily computed in digital environment. Hence, $\text{ln}\text{cosh}(m(n, \tilde{\tau}))$ is approximated by

$$\text{ln}\text{cosh}(m(n, \tilde{\tau})) \approx \begin{cases} (m(n, \tilde{\tau})/2)^2, & |m(n, \tilde{\tau})| << 1 \\ |m(n, \tilde{\tau})|, & |m(n, \tilde{\tau})| >> 1 \end{cases}$$
Square-law approximation will be pursued here and then the approximated log-likelihood function is given by

\[ \Lambda_a(\tilde{\tau}) \approx \sum_{n=1}^{N} m^2(n, \tilde{\tau}) \]

where all immaterial constants have been ignored.

**Multilevel PAM Signal**

When symbols have multilevel values, distribution \( p(a) \) of the random continuous symbols can be approximated using zero mean Gaussian with unit variance. Thus,

\[ p(a) = \frac{1}{\sqrt{2}} \exp \left( -\frac{a^2}{2} \right) \]

If we average \( \Lambda(\tilde{\tau}) \) over the assumed pdf of \( a \), it can be shown that obtained log-likelihood function corresponds to the previous equation.

**QAM Signal**

For complex-valued symbols, two-dimensional Gaussian approximation is used in averaging over the symbol values. We assume that symbol values are statistically independent with zero mean and unit variance. It can be shown that two-dimensional Gaussian distribution \( p(a,b) \) corresponds to

\[ p(a,b) = \frac{1}{2\pi} \exp \left( -\frac{a^2 - b^2}{2} \right) \]

and log-likelihood function can be given by

\[ \Lambda_a(\tilde{\tau}) \approx \sum_{n=1}^{N} \left[ m_i^2(n, \tilde{\tau}) + m_Q^2(n, \tilde{\tau}) \right] \]

where \( m_i(n, \tilde{\tau}) \) and \( m_Q(n, \tilde{\tau}) \) are outputs of the matched filters in the in-phase and quadrature branches, respectively. As can be seen, after demodulation one can treat the real and imaginary parts of the signal as two independent PAM signals.
In general, for linear digital modulations, the symbol timing estimate is:

$$\hat{\tau} = \arg \max_{\tilde{\tau}} \sum_{n=1}^{N} |m(n, \tilde{\tau})|^2$$

For M-PSK, the carrier phase estimate can be found as:

$$\hat{\theta} = \frac{1}{M} \arg \left( \sum_{n=1}^{N} (m(n, \hat{\tau}))^M \right)$$

Examples of DA and NDA log-likelihood functions for binary PAM

(raised cosine pulse shape with 35 % roll-off)
Extracting the Timing Estimates

Here three different methods for extracting the timing estimate are represented. This corresponds to finding the maximum of the ML function.

Parameter Search

The ML function could be computed at a grid of evenly-spaced timing error estimate values $\tilde{\tau}$ until the global maximum is found.

Trackers

At the global maximum, the derivative of the ML function must be zero. The derivative defines an error-detector algorithm, the output of which is applied to the closed loop tracker. The function of the tracker is to drive the error, i.e., derivative to zero.

Direct Computation

It is sometimes possible to solve the value of $\tilde{\tau}$ that satisfies the condition $d\Lambda(\tilde{\tau})/d\tilde{\tau} = 0$. Taking care that the solution corresponds to the global maximum, this provides an algorithm for the direct computing of timing estimate. Once the timing estimate $\hat{\tau}$ has been computed, a feedforward correction can be applied. Direct computing possibility does not exist in the analog implementations.

Literature


Examples of Data-Aided Feedback and Feedforward Timing Estimation Techniques

Exact Differentiator Tracking Algorithm

There are two filters in this model, namely the matched filter $h(-t)$ and derivative matched filter $h'(-t)$. The outputs of the two filters are sampled once per symbol interval and obtained samples from the matched filters are denoted by $m(n,\hat{\tau})$ and $m'(n,\hat{\tau})$. Symbol decisions $\hat{a}(n)$ are obtained from the $m(n,\hat{\tau})$, and the timing error tracking algorithm is given by

$$e(n,\hat{\tau}) = \hat{a}(n)m'(n,\hat{\tau})$$

Timing error trials are averaged in the loop filter and feedback loop drives the $E[e(n,\hat{\tau})]$ to zero in order to obtain $\hat{\tau}$. From the implementation point of view, the filters are the most complex part of the system.

Implementation can be simplified by replacing the derivative filter by differentiator. The differentiator can be implemented
digitally and corresponding simplest two-point approximate differentiator tracking algorithm is

\[ e(n, \hat{\tau}) = \hat{a}(n) \left[ m \left( n + \frac{1}{2}, \hat{\tau} \right) - m \left( n - \frac{1}{2}, \hat{\tau} \right) \right] \]

Now, a single subtract operation has replaced the filter \( h'(-t) \). However, the sampling rate has to be doubled in the digital differentiation.

Two-point approximate differentiator timing error tracker.

- Zero-crossing detector
- Gardner method
(b) Convergence performance ($\alpha=0.35$)

(c) Convergence performance ($\alpha=0.1$).

Fig. 4.2. Performance of a DD timing error tracker.
Feedforward Timing Estimation Techniques

Several ML-based feedback timing estimation techniques have been introduced in the literature but there are not many feedforward techniques. In fact, we know from the literature only one ML-based timing estimation technique introduced by Sabel and Cowley. One reason for this is that feedforward estimation is unknown in analog implementations and thus unfamiliar to most designers.

Direct timing error estimate symbol synchronizer.

Benefits:

- rapid acquisition characteristics
- no hang-up problems as in cases of feedback and PLL based schemes
- accuracy of the timing estimator determines directly the accuracy of timing recovery
PARTIAL RESPONSE (PR) SIGNALLING

One of the primary goals of line-coding, such as AMI-codes, is to control the spectrum of the transmitted signal. PR signalling can also be used to control the spectrum of transmitted signal.

The basic idea here is to change the spectrum by filtering the signal before transmission with a filter that inserts spectral zeros at DC and/or half the symbol rate.

The important thing to remember is the fact the transmitted signal has ISI, but this is controlled ISI which can be removed in the receiver.

PR signalling is used together with Nyquist pulse-shaping. Using this principle it is possible to transmit signal with no excess bandwidth, i.e., excess bandwidth is 0%.

For binary antipodal data at the input of PR-signalling link, the output of the filter has three levels, thus it becomes a pseudo ternary channel. This result in larger constellation which reduces the noise immunity.

PR-signalling is used with binary PAM and also with Continuous-phase modulation, CPM.

Source: Lee& Messerschmitt, Sections 10.3-10.4.
**Principle**

We study in details the PR principle for the PAM system. The transmitted symbol sequence is passed through a discrete-time spectral shaping filter:

\[ F(z) = \sum_{i=0}^{N} f_{i}z^{-1}, \quad f_{0} = 1 \]

The interesting cases have the following simple general form:

\[ F(z) = (1 + z^{-1})^{m}(1 - z^{-1})^{n} \]

This filter can be used to place an arbitrary number of zeros in the spectrum at DC and/or at half the symbol rate.

There are three important examples:

(a) **Dicode** (twinned binary linecode):
\[ F(z) = 1 - z^{-1}, \quad F(e^{-j\omega T}) = 1 - e^{-j\omega T} \]
places spectral zero at DC.

(b) **Duobinary**:
\[ F(z) = 1 + z^{-1}, \quad F(e^{-j\omega T}) = 1 + e^{-j\omega T} \]
places spectral zero at half the symbol rate

(c) **Modified duobinary**:
\[ F(z) = (1 + z^{-1})(1 - z^{-1}) = 1 - z^{-2}, \quad F(e^{-j\omega T}) = 1 - e^{-j2\omega T} \]
places zeros at both DC and half the symbol rate
**Constellations**

For the three cases above, if we put a binary antipodal data symbol \(\{\pm 1\}\) into the filter, the output is three-level symbol \(\{\pm 2, 0\}\). Since there is only one bit of information conveyed per output symbol, the resulting code is **pseudo-ternary**. Choosing other \(F(z)\) can result in a much larger constellation. This is the price to pay for ability to control the spectrum of the signal. In general, the larger constellation reduces the noise immunity.

The filter \(F(D)\) introduces ISI (there is a correlation of successive symbols). This is **controlled ISI** which can be removed in the receiver; it is done intentionally and it is not a side effect of the channel.
Receiver

In the receiver, the controlled ISI is removed. In the following, four possible ways to remove this controlled ISI are described in the example case of a dicode pseudo ternary signal.

(a) The received sequence is filtered by using the transfer function \(1/(1-z^{-1})\). After this filter, the signal is again binary antipodal. However, the noise enhancement in this case is infinite because of the algebraic zero in the denominator of the transfer function. Thus this principle cannot be used in practise. This is actually a linear equalizer with zero forcing criterion, LE-ZF.

(b) DFE-type of equalizer works better, it does not have the noise enhancement problem. However, it introduces the problem of error propagation.

(c) This structure applies directly ternary slicer (according to nature of the received signal), after which the linear equalizer is used. This method does not have the noise enhancement problem but it has the problem of error propagation.

(d) The best performance is achieved by using the ML sequence detection using, e.g., the Viterbi algorithm.

The rest of receiver is the same as in any PAM receiver.
**Precoding**

There is another way to deal with ISI, which is a generalization of the approach used to go from twinned binary to AMI.

The input bit stream is put through a precoder prior to the spectral shaping filter $F(z)$. The receiver is very similar to case (c) from the previous slide. The slicer has more than two levels, and it is followed by a decoder to recover the bit stream. Because of precoding, the decoder is memoryless, and there is no error propagation.

This method gives a simple technique for gaining the benefits of spectral shaping, with an implementation simpler than the ML sequence detector, and without the noise enhancement of LE or error propagation of DFE.

The combination of a precoder with spectral shaping filter $F(z)$ is called **partial response** (PR). PR is also sometimes called correlative coding.

Here are two simple examples of PR:

<table>
<thead>
<tr>
<th><strong>Dicode</strong></th>
<th><strong>Duobinary</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_{k-1}$</td>
<td>$b_k$</td>
</tr>
<tr>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>+1</td>
<td>0</td>
</tr>
<tr>
<td>+1</td>
<td>1</td>
</tr>
</tbody>
</table>

The example of dicode partial response here shows, that in this case the derived simple precoder is identical to that for AMI:

\[
\hat{a}_k \quad \hat{b}_k \\
-2 \quad 1 \\
0 \quad 0 \\
+2 \quad 1
\]
PR & Nyquist Pulse-Shaping

In the case when the band of the transmitted signal is minimized, PR signalling is used together with Nyquist pulse-shaping. In that case, the equalized pulse shape and its Fourier transform are

\[ h(t) = \sum_{i=0}^{N} f_i g(t - iT) \]
\[ H(f) = F(e^{-j2\pi T})G(f) \]

\( F(e^{-j2\pi T}) \) corresponds to one of three main cases. \( G(f) \) is spectrum of the basic pulse shape.

For duobinary and modified duobinary cases, \( F(e^{-j2\pi T}) \) has a zero at half the symbol rate. In these cases it is practical to use 0 % excess bandwidth. For 0 % excess bandwidth \( G(f) \) uniquely an ideal lowpass filter, and the PR responses are:

(a) Duobinary: \( h(t) = \text{sinc} \left( \frac{\pi}{T} t \right) + \text{sinc} \left( \frac{\pi}{T} t - \pi \right) \)

(b) Modified duobinary: \( h(t) = \text{sinc} \left( \frac{\pi}{T} t \right) - \text{sinc} \left( \frac{\pi}{T} t - 2\pi \right) \)
Other PR Purposes

The main purpose of PR signalling is the additional spectral shaping, as it was explained above.

There are possible motivations for using PR signalling as part of channel or receiver.

This is the case when the channel naturally makes a zero in the spectrum (e.g., and AC-coupled channel has zero at DC) in which case part of $F(z)$ arises naturally.

A zero placed in the receiver may help to reduce the noise at the slicer input, since it reduces the gain in the receive equalizer for some frequencies.

We can conclude that PR is not only a method of spectrum control but also as an alternative to DFE for improving the noise immunity, but without the error propagation.

The following examples can be found in the text book (Examples 10-20, -21, -22). Here we look at the main conclusions.
Examples

**Case with ISI=0, AWGN-channel:**

(a) **Duobinary & transmit filtering vs. 2-PSK**

2-PSK 3 dB better with respect to average signal power
" 6 dB better for peak signal power

(b) **Duobinary & receive filtering vs. 2-PSK**

2-PSK 3 dB better with respect to both the peak and average powers

(c) **Duobinary, transmit filter & MLSD vs. 2-PSK**

the performance is 3dB better with respect to peak signal power, and equal with respect to average signal power!

**Channel with duobinmary ISI:** \( Y_k = X_k + X_{k-1} + N_k \)

Possibilities:
- duobinary signaling
- 2-PSK & DFE (the performance is the same.)

ML-sequence detector has 3 dB better performance than the simple slicer.

In pair-cable or coaxial cable systems with losses of 60 - 80 dB at half the symbol rate, the duobinary system has an essential noise advantage over binary antipodal signaling.
In the book and in lecture notes, the Continuous phase modulation (CPM), for example MSK, has already been treated.

These modulation methods are advantageous particularly for channels with nonlinearities by keeping constant envelope and by avoiding abrupt phase changes.

Also in this case, PR principle helps in improving the spectral characteristics of the transmitted signal.

The spectrum becomes narrower than a constant envelope PSK.

CPM signal detection can be performed with the Viterbi-algorithm.

In the MSK case, the obtained increase in performance is 3 dB better than in the simple slicer case. In this way, the MSK power efficiency becomes the same as the power efficiency of 2-PSK.
SCRAMBLING

We often make assumptions that the transmitted data symbols are independent of each other. However, in practice there will appear particular bit sequences which can cause some difficulties. (For example, long sequences of zeros or ones will cause difficulty in timing recovery and adaptive equalization.)

Scrambling is a method of achieving DC balance and eliminating long sequences of same symbols, to ensure accurate timing recovery without redundant line coding.

This method is not based on redundancy; therefore it does not add anything in signal. Scrambling performs one-to-one mapping between input data bits and coded data bits. The objective is to map bit sequences that are problematic and likely to occur into a coded sequence that looks more random and less problematic.

All CCITT-standardized voice-band data modems incorporate scrambling.

Scrambling also helps to cope with the problem of removing the DC component and baseline wander ISI. A combination of scrambling with AMI coding would be effective in eliminating low frequency components and providing adequate timing energy.

Source: Lee & Messerschmitt, Section 10.5.
Scrambling is based on using maximal-length shift-register sequences, which are periodic bit sequences with properties that make them appear random. They are also called pseudorandom sequences, which can be generated using a feedback shift register:

Here the coefficients are binary, and summation is modulo 2. The shift-register cannot have state 0 0 ... 0, because nothing would happen in the structure after that state.

The coefficients $h_1, ..., h_n$ are defined by the generator polynomial. There is one-to-one mapping between generator polynomials and feedback structures.

The maximum length of the generated pseudorandom sequence is $2^n - 1$. This is easy to see, since (i) the circuit is deterministic in the sense that starting from a given initial state, the generated sequence is always the same, (ii) there are $2^n$ different states, out of which one (the zero-state) cannot be used sequence.

Generator polynomials that provide such maximum length sequences (or m-sequences) are tabulated in the literature.
Generator Polynomials for m-Sequences

Here the generator polynomials are given in octal form. When they are converted to binary form, the taps of the feedback shift-register structure can be directly identified. The MSB corresponds to $h_1$ and LSB $h_n$.

Notice that the first and last taps are always 1’s.

Example

The generator polynomial $h(D) = 1 \oplus D^{14} \oplus D^{17}$ (to be mentioned a few pages later) corresponds to row 17 of the Table:

400011 => 100 000 000 000 001 001

<table>
<thead>
<tr>
<th>2</th>
<th>7</th>
<th>19</th>
<th>2000047</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>13</td>
<td>20</td>
<td>4000011</td>
</tr>
<tr>
<td>4</td>
<td>23</td>
<td>21</td>
<td>10000005</td>
</tr>
<tr>
<td>5</td>
<td>45</td>
<td>22</td>
<td>20000003</td>
</tr>
<tr>
<td>6</td>
<td>103</td>
<td>23</td>
<td>40000041</td>
</tr>
<tr>
<td>7</td>
<td>211</td>
<td>24</td>
<td>100000207</td>
</tr>
<tr>
<td>8</td>
<td>435</td>
<td>25</td>
<td>200000011</td>
</tr>
<tr>
<td>9</td>
<td>1021</td>
<td>26</td>
<td>4000000107</td>
</tr>
<tr>
<td>10</td>
<td>2011</td>
<td>27</td>
<td>1000000047</td>
</tr>
<tr>
<td>11</td>
<td>4005</td>
<td>28</td>
<td>2000000011</td>
</tr>
<tr>
<td>12</td>
<td>10123</td>
<td>29</td>
<td>4000000005</td>
</tr>
<tr>
<td>13</td>
<td>20033</td>
<td>30</td>
<td>10040000007</td>
</tr>
<tr>
<td>14</td>
<td>42103</td>
<td>31</td>
<td>20000000011</td>
</tr>
<tr>
<td>15</td>
<td>100003</td>
<td>32</td>
<td>40020000007</td>
</tr>
<tr>
<td>16</td>
<td>210013</td>
<td>33</td>
<td>100000020001</td>
</tr>
<tr>
<td>17</td>
<td>400011</td>
<td>34</td>
<td>201000000007</td>
</tr>
<tr>
<td>18</td>
<td>1000201</td>
<td>35</td>
<td></td>
</tr>
</tbody>
</table>
Frame-Synchronized Scrambler

The user bit sequence is summed, in modulo, 2 with the output of the maximal-length shift-register in the transmitter to produce the scrambled bit stream. The stream is descrambled in receiver using another modulo 2 summation with the output of identical maximal-length shift-register.

The correct operation of the frame-synchronized scrambler depends on the time alignment of the two periodic maximal length sequences in transmitter and receiver. This is called frame-synchronization, which is an additional functionality in receivers for such systems.
Self-Synchronized Scrambler

This method does not need frame-synchronization.

Identical shift registers are used both in transmitter and receiver. Here the input stream is added directly to the input of the shift register in the transmitter. The scrambled stream, which is also used as the input to the transmitter shift register, is fed directly to the input to shift register in receiver (in the ideal model).

This method has the following disadvantages:

- error propagation
- problems with periodic input bit streams

Examples

- V.22bis voiceband data modem uses a self-synchronizing scrambler with the generating polynomial
  \[ h(D) = 1 \oplus D^{14} \oplus D^{17} \]
- V.26ter modem uses two generating polynomials in for the two directions of transmission:
  \[ h(D) = 1 \oplus D^{18} \oplus D^{23} \]
  \[ h(D) = 1 \oplus D^{5} \oplus D^{23} \]