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

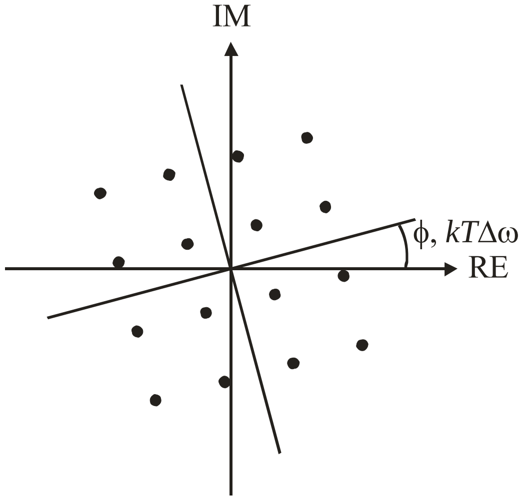
- **Constant carrier phase error** ϕ corresponds to a rotation of the constellation

$$\overline{A}_k = e^{j\phi} A_k$$

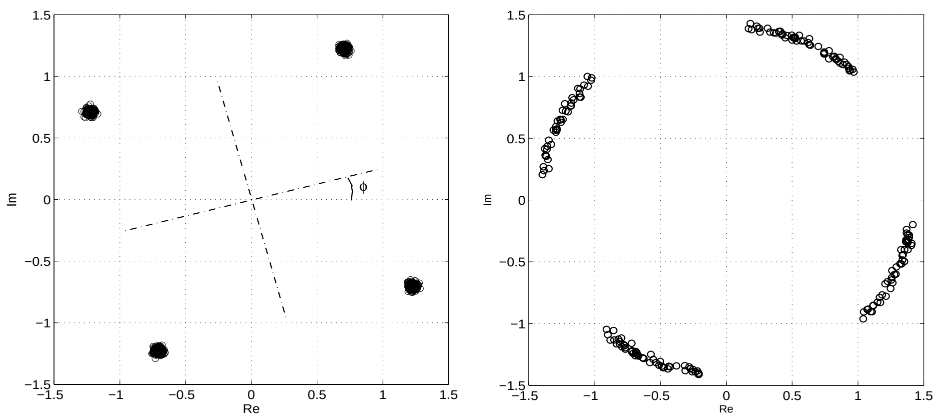
- **Constant carrier frequency error** $\Delta\omega$ corresponds to time-varying rotation of the constellation

$$\overline{A}_k = e^{jkT\Delta\omega} A_k$$

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.



Example: Effect of phase error (left) and frequency error (right) on QPSK constellation @ SNR=20 dB:

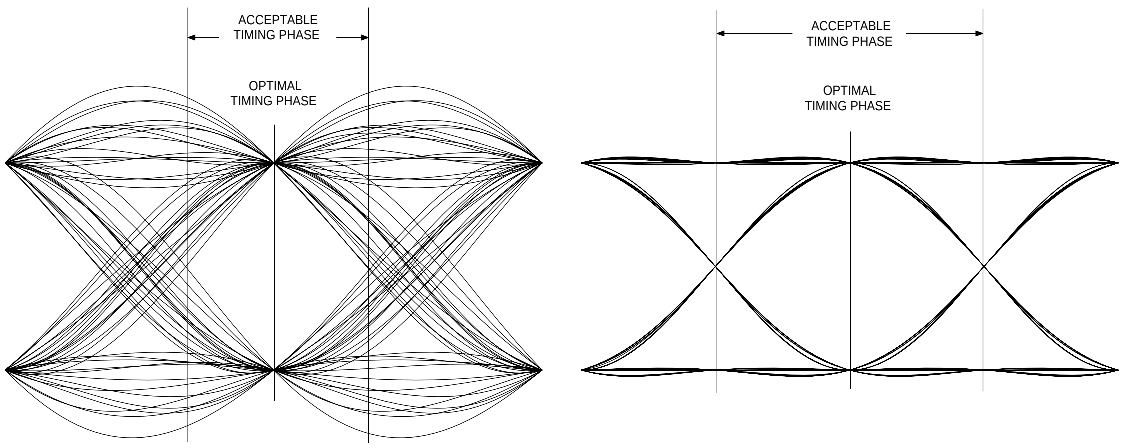


Symbol timing recovery

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

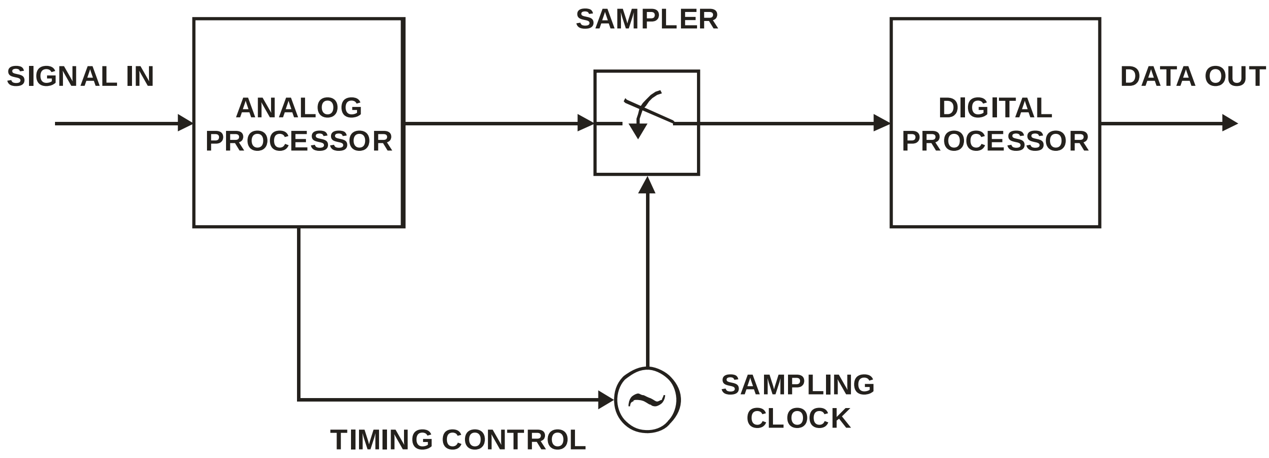
The effects of symbol timing errors can be seen with the aid of the eye pattern:

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

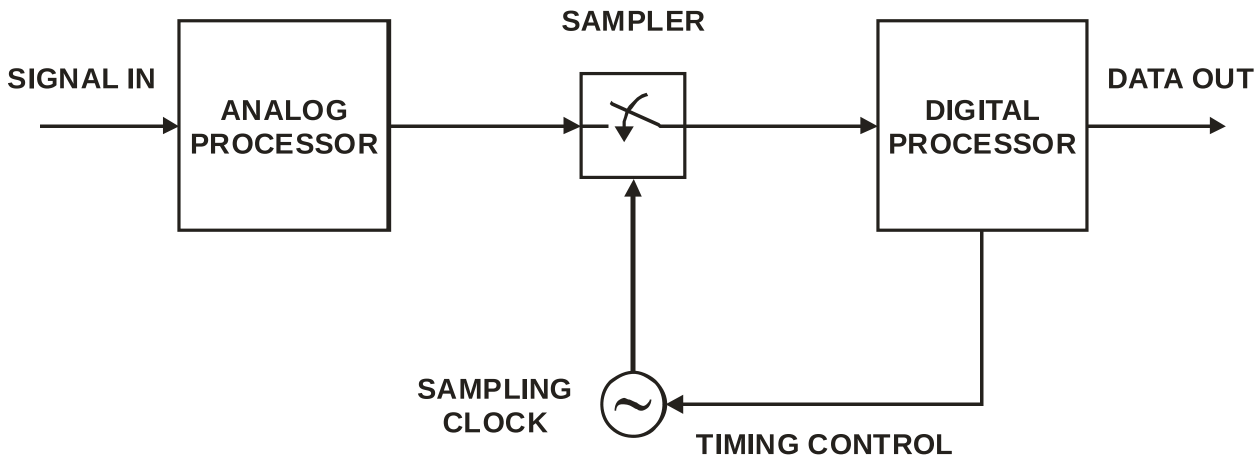


Synchroniser classes

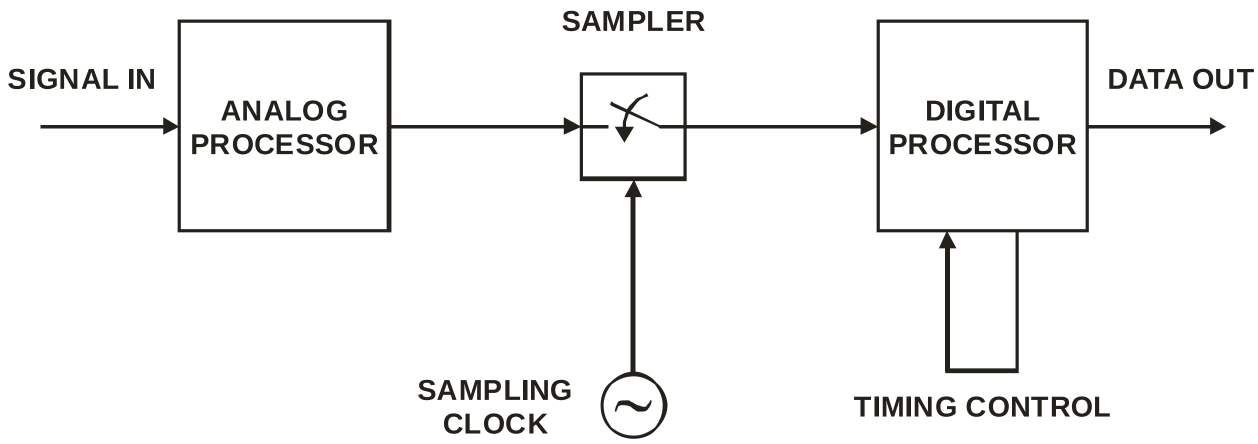
Analog timing recovery:



Hybrid timing recovery:



Digital timing 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.

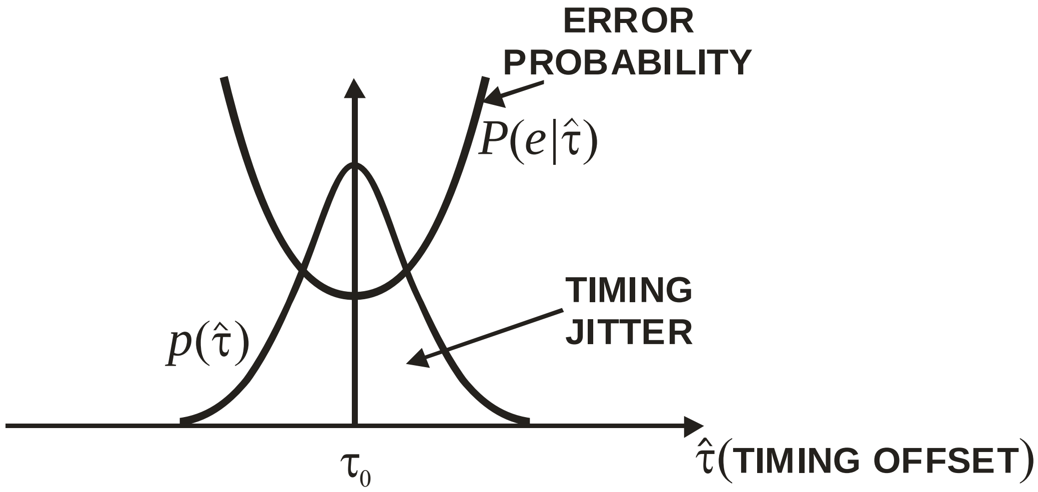
Hang-up, cycle slip

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.

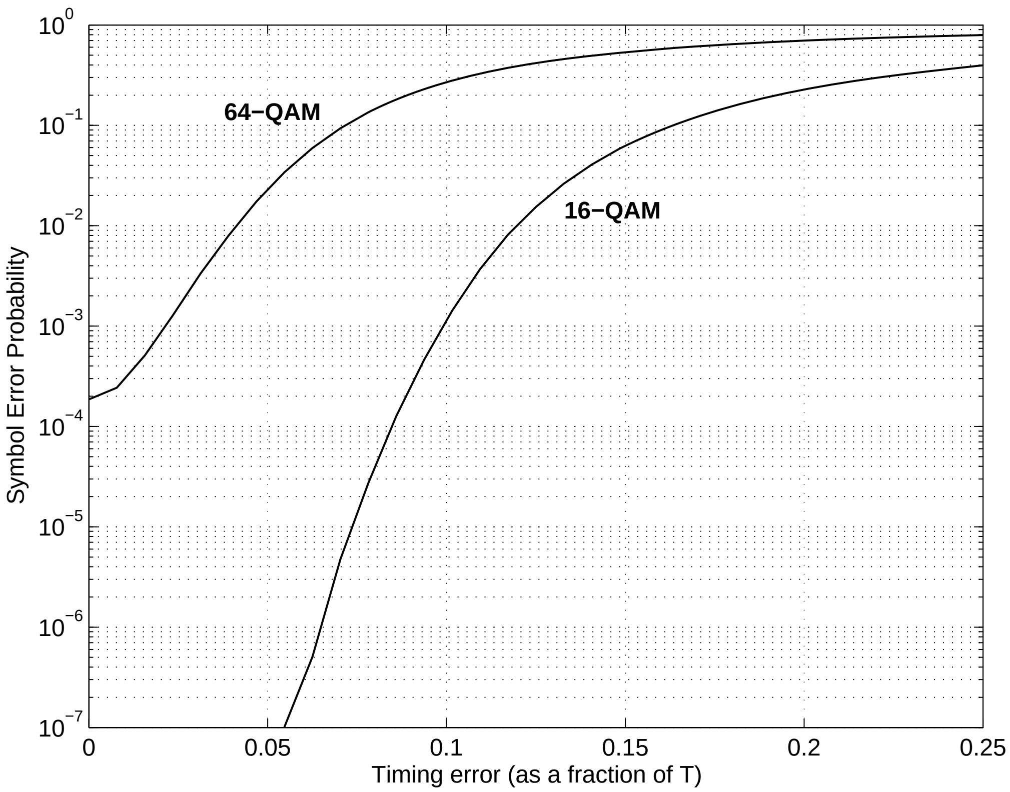
Symbol-error rate / Bit-error rate

Best measure of the overall performance.

Illustrating the Effect of Timing Jitter on Symbol Error Rate:

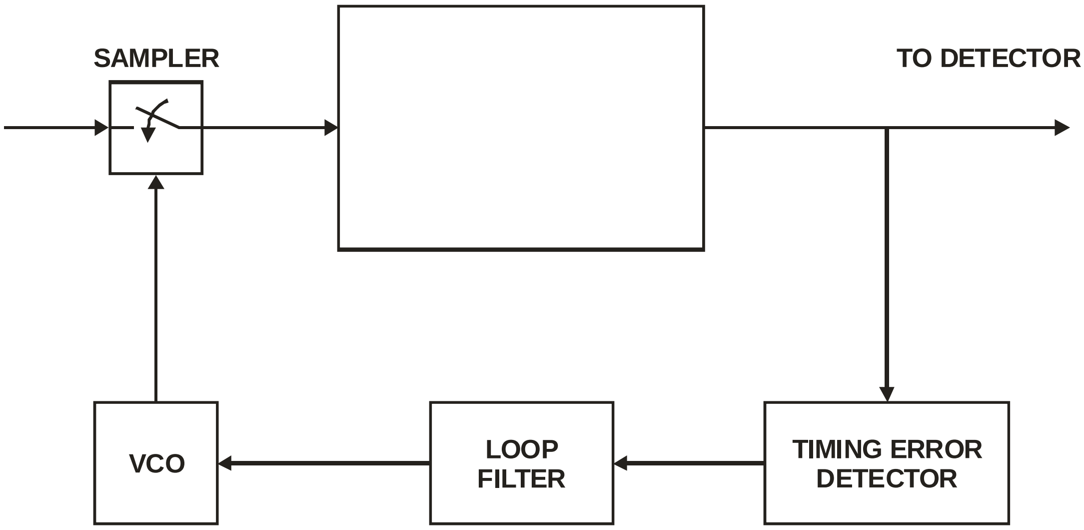


Examples of the Effects of Symbol Timing Errors on Symbol Error Rates, 2-level and 4-level PAM:

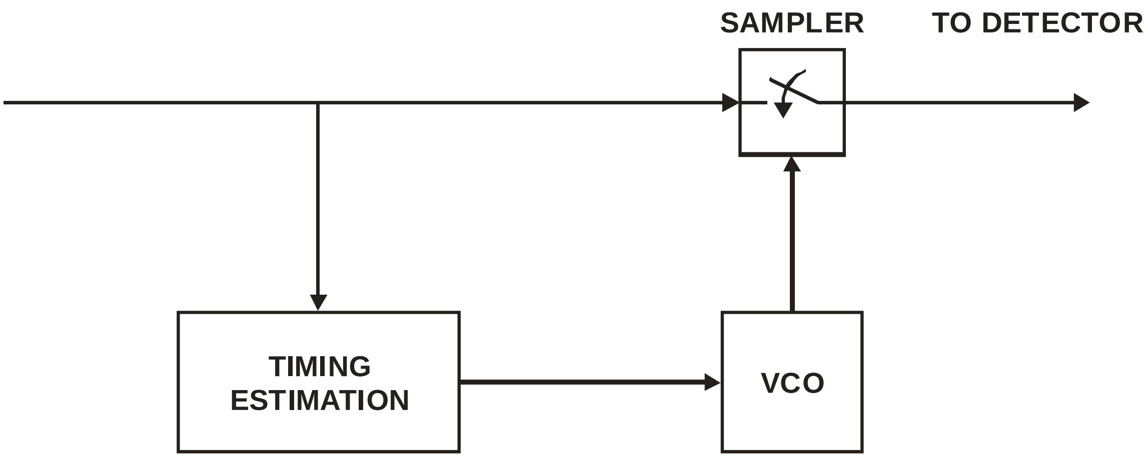


Feedback and feed-forward configurations

Symbol timing recovery can be done either in feedback configuration



or in feedforward configuration



The same is true also for carrier phase recovery.

Coarse carrier frequency recovery is usually based on feedback configuration.

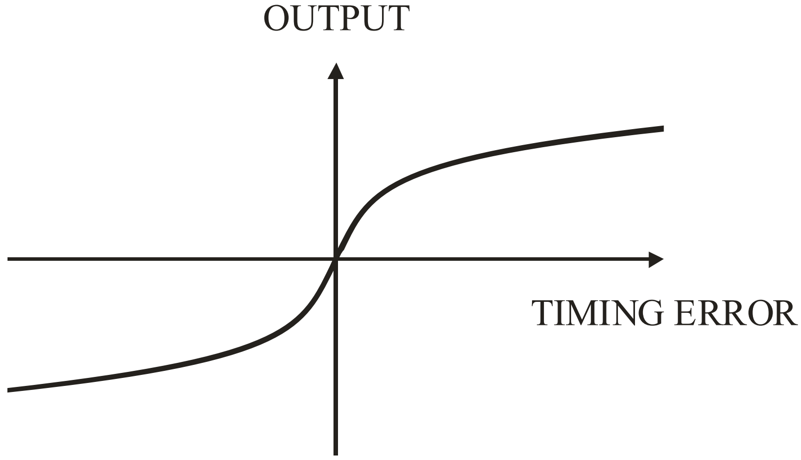
Comparing feedback and feed-forward configurations

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 continuosly.

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.

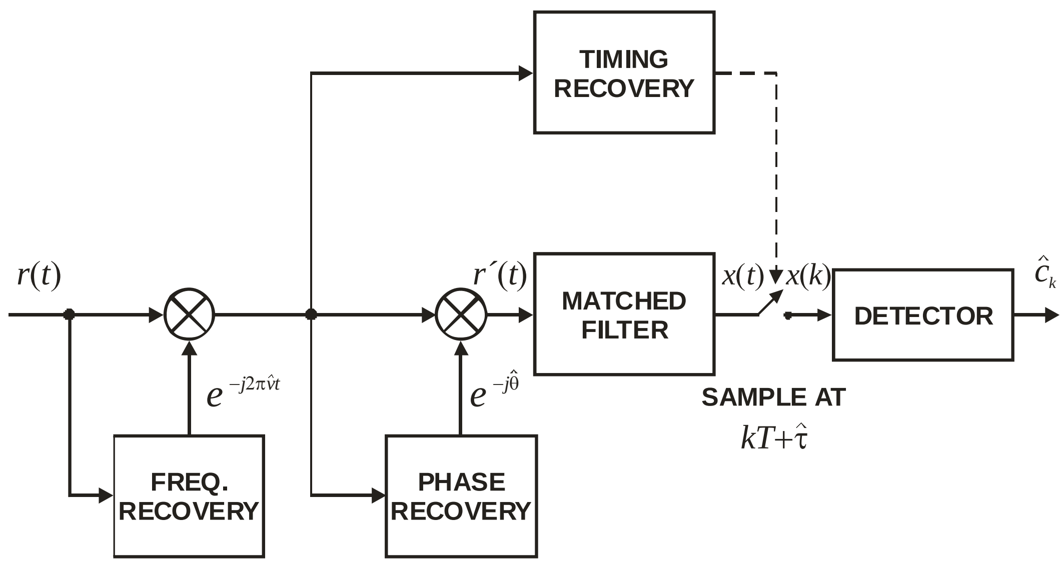
However, there are some hybrid timing recovery algorithms that operate on symbol-rate sampes.

Relation to channel equalisation

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

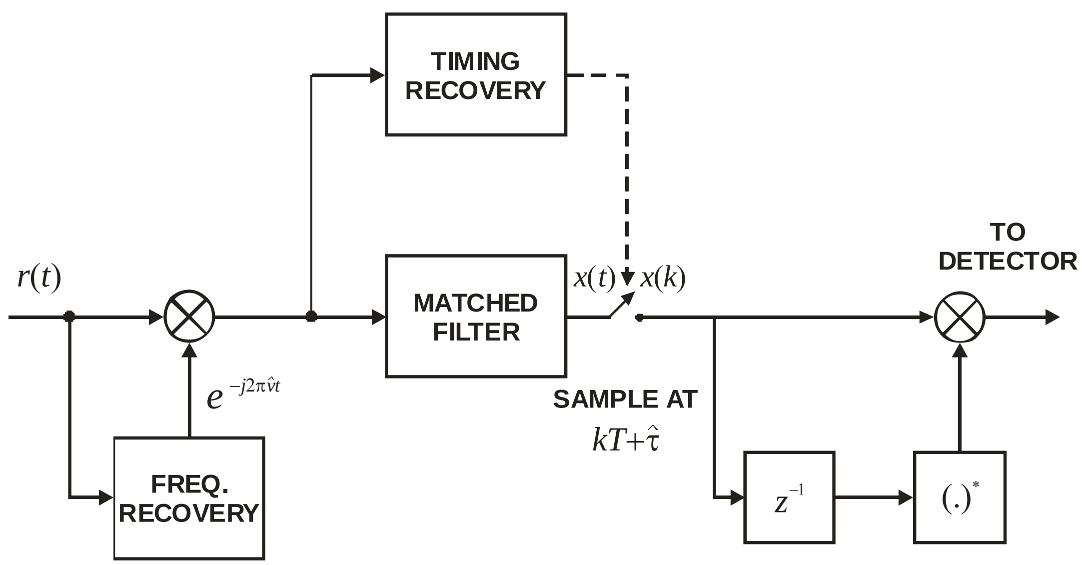
Examples of Receiver Structures

Coherent receiver:

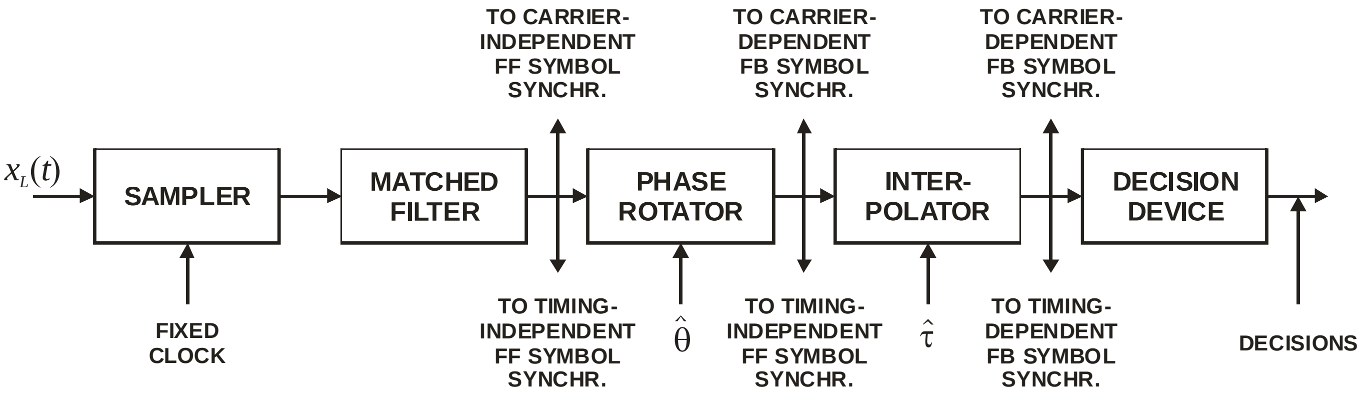


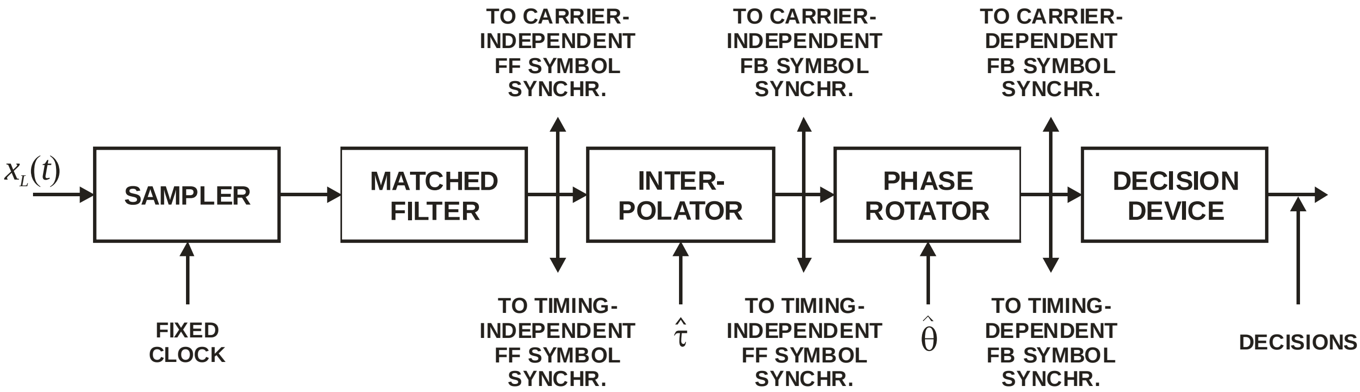
A more practical, modern receiver structure would have sampling at 2x symbol rate after the coarse frequency recovery and implement the phase and timing recovery in the DSP domain.

Differential receiver:



Order of the blocks in all-digital solutions





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 woud 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.

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.

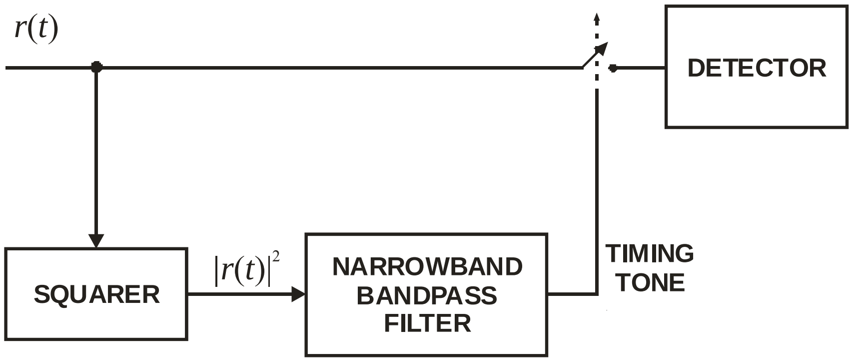
A general theoretical framework for developing and optimising algorithms for carrier and symbol timing recovery is the *Maximum Likelihood estimation theory*.

Some ad-hoc timing recovery algorithms

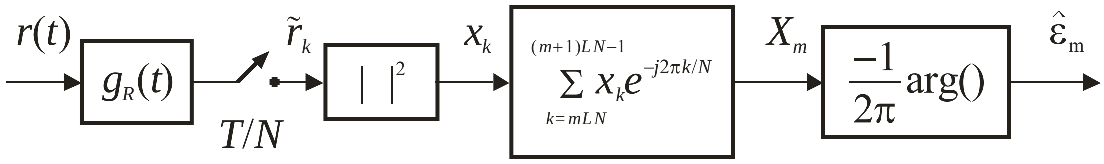
A set of algorithms utilizes a *timing tone,* a symbol-rate spectral component, which is obtained, e.g., by squaring the baseband data signal.

The timing tone can be extracted by

- a narrowband lowpass filter



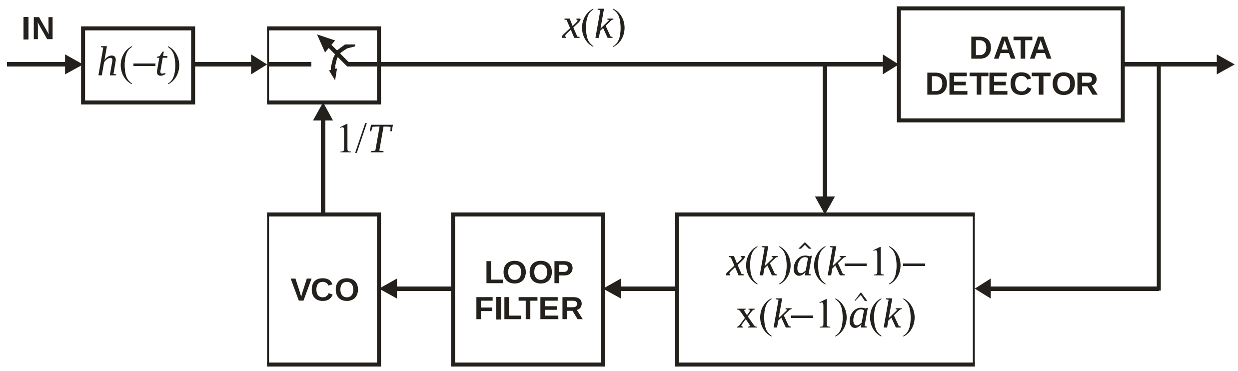
- phase-locked loop
- computing DFT at the symbol frequency (i.e., a single point of the DFT output is needed for each data block)



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.

Mueller&Müller algorithm

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, Lauri Anttila, Vesa Tuukkanen*
TUT/DCE
* 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 $h(t)$.

The unknown parameter vector Φ_0 may include carrier phase θ_0, timing delay τ_0, frequency error ν_0, amplitude A_0, and in the most general formulation, also the complex data sequence $\{a_I(n)+ja_Q(n)\}$. The synchronization parameters are 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 Φ_0.

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 $\tilde{\Phi}$,

$$L(\tilde{\Phi}) = p\left[r(T_0)\middle|\tilde{\Phi}\right],$$

where $r(T_0)$ is the observed $r(t)$ in the observation interval.

The aim is to find the specific trial value, $\hat{\Phi}$, of the parameter vector which maximizes the likelihood function,

$$L(\hat{\Phi}) = \max_{\tilde{\Phi}} L(\tilde{\Phi})$$

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

$$L(\tilde{\Phi}) = C_1 \exp\left[-\frac{C_2}{N_0}\int_{T_0} \left|r(t) - s(t, \tilde{\Phi})\right|^2 dt\right]$$

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

Expanding the integrand yields

$$\begin{aligned}\left|r(t) - s(t, \tilde{\Phi})\right|^2 &= \left|r(t)^2\right| + \left|s(t, \tilde{\Phi})\right|^2 - r(t)s^*(t, \tilde{\Phi}) - r(t)^* s(t, \tilde{\Phi}) \\ &= \left|r(t)^2\right| + \left|s(t, \tilde{\Phi})\right|^2 - 2\,\mathrm{Re}\left[r(t)s^*(t, \tilde{\Phi})\right]\end{aligned}$$

The integral of the <u>first term</u> represents the energy of the received noisy signal and is independent of $\tilde{\Phi}$, so it can be ignored.

The <u>second term</u> is more complicated because it is a function of $\tilde{\Phi}$. The integral of $\left|s(t, \tilde{\Phi})\right|^2$ 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 or m-PSK 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

$$L\left(\tilde{\Phi}\right) = C_3 \exp\left[\frac{2C_2}{N_0}\int_{T_0} \mathrm{Re}\left[r(t)s^*\left(t,\tilde{\Phi}\right)\right]dt\right]$$

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

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

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 estimator can be considered as a correlator where correlation between $r(t)$ and $s\left(t,\tilde{\Phi}\right)$ is computed at arbitrary values of $\tilde{\Phi}$. MLE is the value of $\tilde{\Phi}$ 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 multi-level signals is reduced because the term $\left|s\left(t,\tilde{\Phi}\right)\right|^2$, which was ignored, is not a constant anymore.

Joint DA/DD Symbol Timing and Carrier Phase Estimation for Linear Modulation

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

$$s(t,\Phi_0) = A_0 \exp(j\theta_0)\sum_n \left[a_I(n)h(t-nT-\tau_0) + ja_Q(n)h(t-nT-\varepsilon T-\tau_0)\right]$$

where ε is the stagger coefficient (e.g., for offset-QAM, $\varepsilon = 0.5$), and $h(t)$ is the pulse shape of the transmitted signal. Frequency offset is here assumed to be zero.

In this case the log likelihood function is

$$\begin{aligned}
&\Lambda(\tilde{\Phi}) \\
&= \mathrm{Re}\left[\exp(-j\tilde{\theta})\sum_{n=1}^{N} \int_{-\infty}^{\infty} r(t)\left[\hat{a}_I(n)h(t-nT-\tilde{\tau}) - j\hat{a}_Q(n)h(t-nT-\varepsilon T-\tilde{\tau})\right]dt \right] \\
&= \mathrm{Re}\left[\exp(-j\tilde{\theta})\sum_{n=1}^{N} \hat{a}_I(n) \int_{-\infty}^{\infty} r(t)\left[h(t-nT-\tilde{\tau})\right]dt \right] \\
&\quad + \mathrm{Im}\left[\exp(-j\tilde{\theta})\sum_{n=1}^{N} \hat{a}_Q(n) \int_{-\infty}^{\infty} r(t)\left[h(t-nT-\varepsilon T-\tilde{\tau})\right]dt \right]
\end{aligned}$$

where N is the number of symbols in the observation interval. The 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: the ML estimate $\hat{\Phi}$ can be extracted from the matched filter output samples.

If the received signal is non-staggered, that is $\varepsilon = 0$, we obtain

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

where $\hat{a}(n) = \hat{a}_I(n) + j\hat{a}_Q(n)$ and

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

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

$$\Lambda\left(\tilde{\Phi}\right) = \left|\sum_{n=1}^{N} \hat{a}(n)^* m\left(n,\tilde{\tau}\right)\right| \cdot \mathrm{Re}\left[e^{j\left(\arg\left[\sum_{n=1}^{N} \hat{a}(n)^* m(n,\tilde{\tau})\right] - \tilde{\theta}\right)} \right]$$

The latter term can always be made equal to 1 by proper choice of the phase offset. The ML timing estimate can now be written as

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

which is independent of the carrier phase.

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

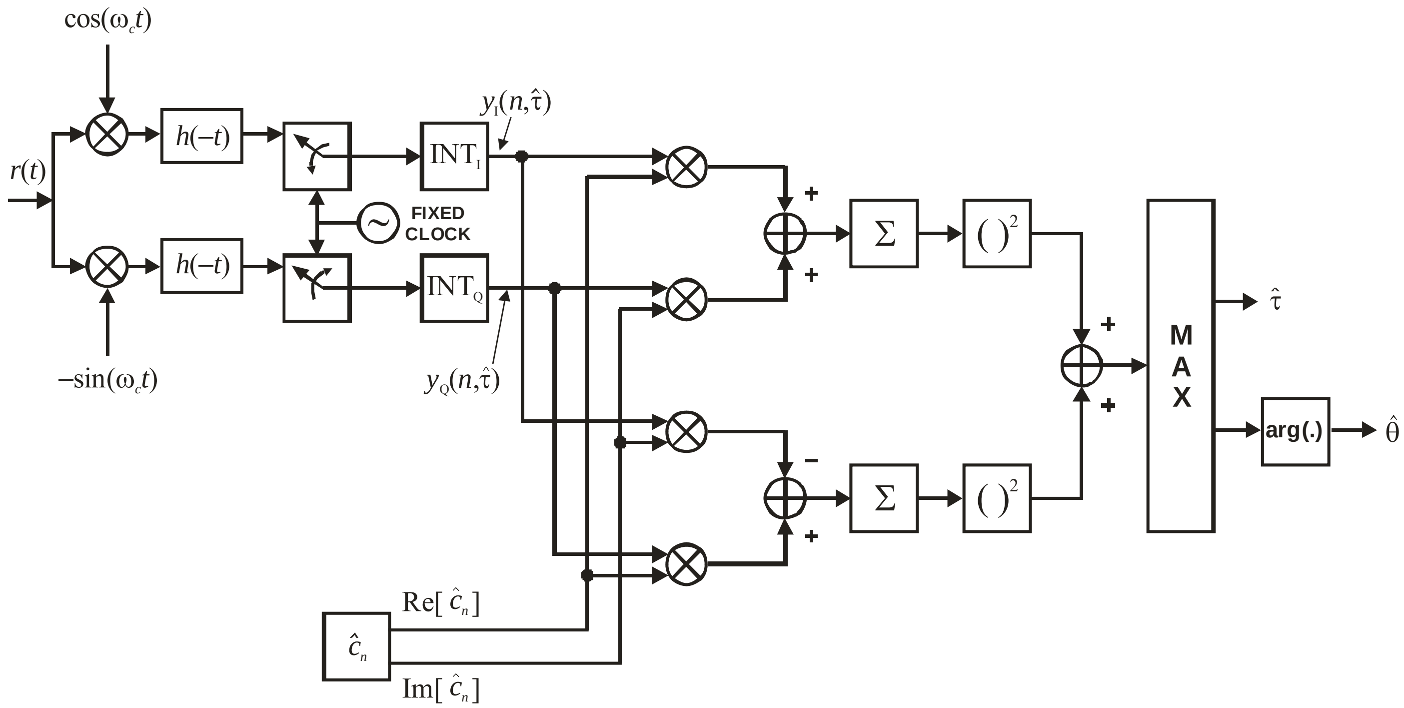
$$\hat{\theta} = \arg\left(\sum_{n=1}^{N} a(n)^* m(n,\hat{\tau}) \right)$$

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:

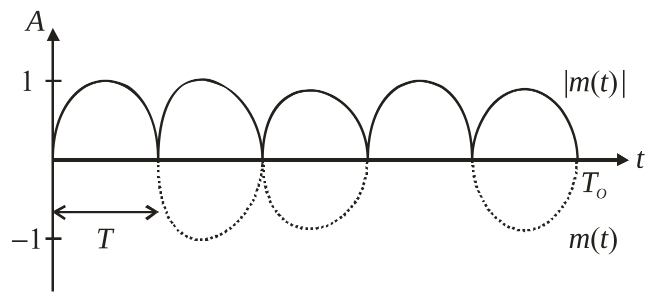
$$\begin{aligned} \left|\Lambda\left(\tilde{\tau}\right)\right|^2 = &\left(\sum_{n=1}^{N} \left(\hat{a}_I(n) m_I\left(n,\tilde{\tau}\right) + \hat{a}_Q(n) m_Q\left(n,\tilde{\tau}\right) \right) \right)^2 \\ &+ \left(\sum_{n=1}^{N} \left(\hat{a}_I(n) m_Q\left(n,\tilde{\tau}\right) - \hat{a}_Q(n) m_I\left(n,\tilde{\tau}\right) \right) \right)^2 \end{aligned}$$

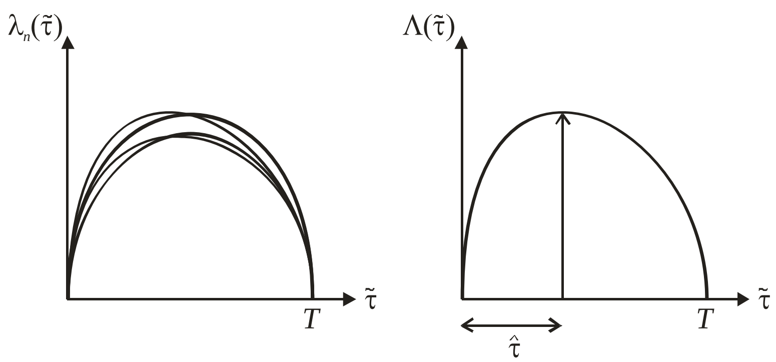
The structure of the data-aided ML timing estimator for I/Q modulations:



I

Illustration of the log-likelihood function for binary PAM (notice that $a(n) * m(t) = |m(t)|$, at least approximately when t is close to a symbol strobe):





Derivation of NDA Timing Error Estimators

Non-Data-Aided (NDA) 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. We consider mostly timing estimation. The joint timing and carrier phase estimation is possible in the case of m-PSK [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

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

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

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

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

and the corresponding log-likelihood function is

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

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

$$\ln \cosh\left(m(n,\tilde{\tau})\right) \approx \begin{cases} \left(m(n,\tilde{\tau})/2\right)^2, & \left|m(n,\tilde{\tau})\right| << 1 \\ \left|m(n,\tilde{\tau})\right| \quad , & \left|m(n,\tilde{\tau})\right| >> 1 \end{cases}$$

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 non-essential constants have been ignored.

Multilevel PAM Signal

When symbols have multilevel values, distribution $p(a)$ of the random continuos 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 the 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 the 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 commonly used ML-based symbol timing estimate is:

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

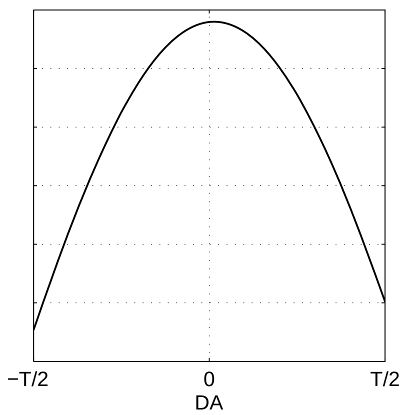
For M-PSK, the carrier phase estimate can be found as:

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

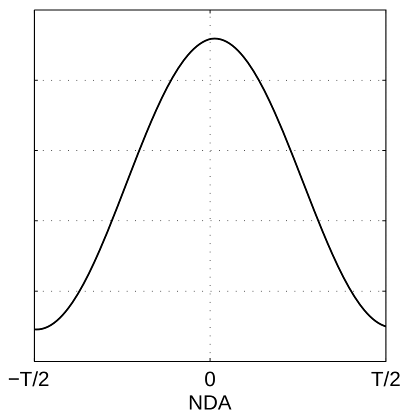
Examples of DA and NDA log-likelihood functions for binary PAM

Raised cosine pulse shape with 35 % roll-off.

DA case



NDA case



Extracting the Timing Estimates

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

Direct Computation

At the global maximum, the derivative of the ML function must be zero. It is sometimes possible to solve the value of $\tilde{\tau}$ that satisfies the condition $d\Lambda\left(\tilde{\tau}\right)/d\tilde{\tau}=0$. 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 exists in the analog implementations.

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.

Exact differentiator tracking algorithm

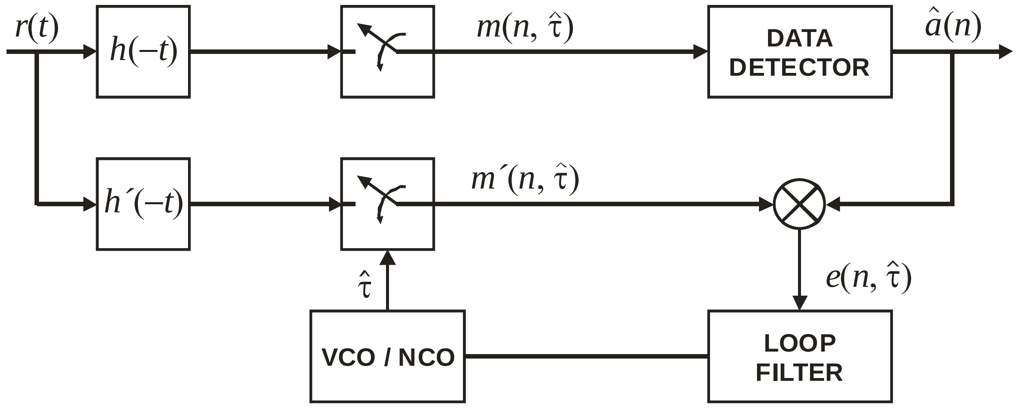
There are two filters in this model, namely the matched filter *h*(-*t*) and derivative matched filter *h´*(-*t*). The outputs of the matched filters are sampled at the symbol rate and the timing error signal for the tracking loop is defined as

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

where $m'(n,\hat{\tau})$ is the output of the derivative matched filter.

The following figure shows the algorithm in the DD form. In the DA case, the know training symbols are used instead of the detected ones. The algorithm can also be modified for the NDA case by defining the timing error signal as:

$$e_{NDA}(n,\hat{\tau})=m(n,\hat{\tau})m'(n,\hat{\tau})$$



The timing error is averaged in the feedback loop. The loop filter in the simplest case is just a small coefficient, and the VCO integrates the timing error values. The feedback drives the mean of the timing error, $E\left[e(n,\hat{\tau})\right]$, to zero. From the implementation point of view, the filters are the most complex part of the system.

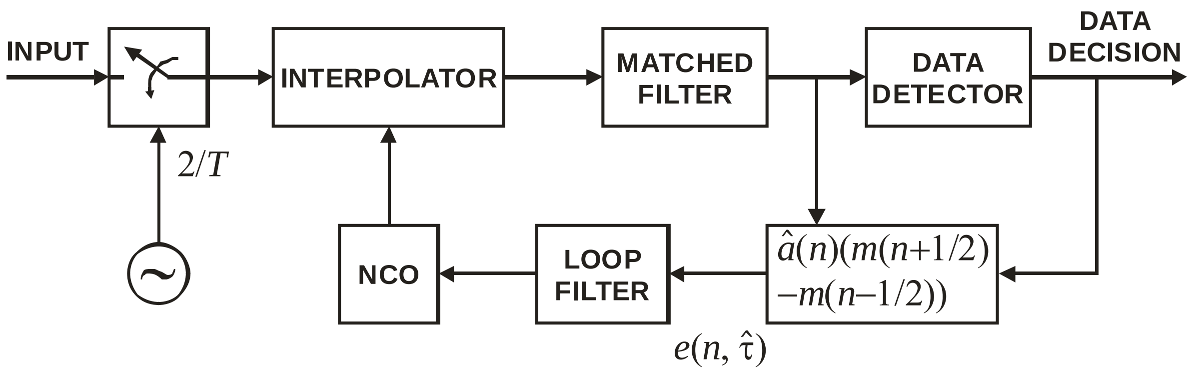
Approximate differentiator timing error tracker

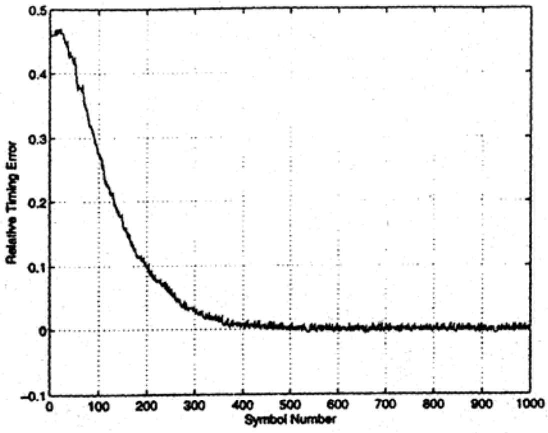
Here the derivative filter is replaced by a differentiator, implemented digitally. The simplest approximate differentiator tracking algorithm is

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

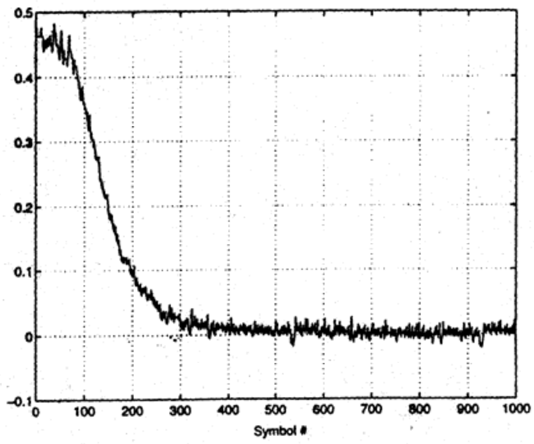
Now a single subtraction operation is replacing the derivative matched filter. However, the matched filter output has to be at twice the symbol rate.

This algorithm is also called as the *Gardner method* or *zero-crossing detector*.





(b) Convergence performance (α=0.35)



(c) Convergence performance (α=0.1).

Fig. 4.2. Performance of a DD timing error tracker.

Early-late tracking algorithms

Another, philosophically different, approach is based on the observation that the log-likelihood function is symmetric with respect to the peak location. Assume that the matched filter output is sampled at $nT-\Delta\tau$ and $nT+\Delta\tau$, in addition to *nT*. Then the tracking loop can be controlled in such a way that the early and late samples are driven to the same level. This is, of course, equivalent to the approximate differentiator idea. The above Gardner method uses $\Delta\tau = T/2$, which is very practical from the implementation point of view.

Parameter Search

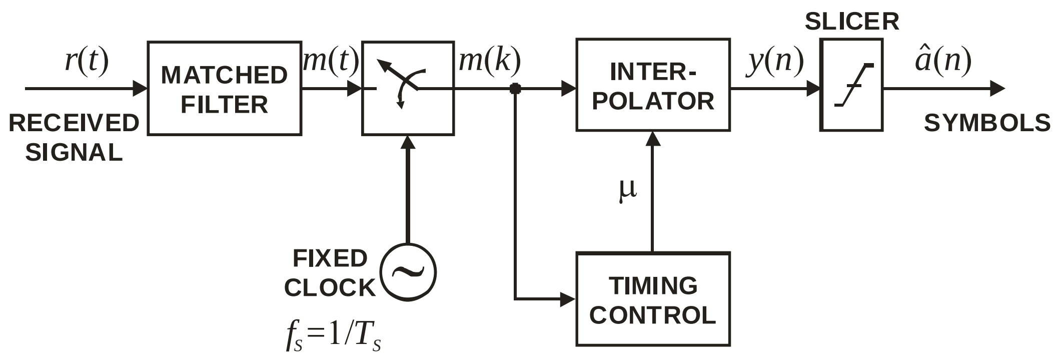
The ML function could be computed at a grid of evenly-spaced timing error estimate values $\tilde{\tau}$, after which the global maximum can be found. However, this would be computationally very heavy approach when high accuracy is required in timing estimation.

In contrast to the tracking or feedback timing recovery configurations discussed above, this class leads to feed-forward timing estimation algorithms. Feed-forward algorithms are characterized by rapid acquisition characteristics and avoidance of hang-up problems (i.e., locking to wrong sampling phase). The accuracy of the timing recovery is determined directly by the accuracy of the timing estimation algorithm. On the other hand, feed-forward algorithms tend to be more complicated than simple tracking loops. However, with today's technologies and utilizing efficient algorithms, the complexity is realistic.

Interpolation based approaches

The basic idea here is that the log-likelihood function is modelled by a simple mathematical function, which is fitted to a number of samples of the log-likelihood function calculated from the matched filter output, using the equations given above. The most common approach is to use low-order piecewise polynomial model, i.e., a polynomial model for each of the sub-intervals. The number of samples used for this fitting determines directly the needed over-sampling factor in the digital matched filter implementation. For example, 3 and 5 points can be used with over-sampling factors 2 and 4, respectively. It has turned out that over-sampling by a factor of 2 is sufficient without compromising the synchronization quality, if high-performance polynomial interpolation techniques are utilized.

After timing estimation, the signal values corresponding to the symbol strobes are interpolated, typically using polynomial interpolation techniques.



The approached developed by Sable and Cowley utilizes 4 samples per symbol and quadratic (2nd-order) polynomial model fitted to selected 3 points within the symbol interval, having the highest values for the log-likelihood function. This method can be directly applied to DA, DD, and NDA synchronizers.

Example: The quadratic approximation of the log-likelihood function is shown below. We see that indexes 0, 1, and 2 correspond to the highest values, so second-order polynomial fitted to the three points. Then the timing offset estimate can be derived to be

$$\hat{\tau} = \tilde{\tau}_0 + \left[\frac{3\Lambda(\tau_0) - 4\Lambda(\tau_1) + \Lambda(\tau_2)}{2(\Lambda(\tau_0) - 2\Lambda(\tau_1) + \Lambda(\tau_2))} \right] \frac{T}{4}$$

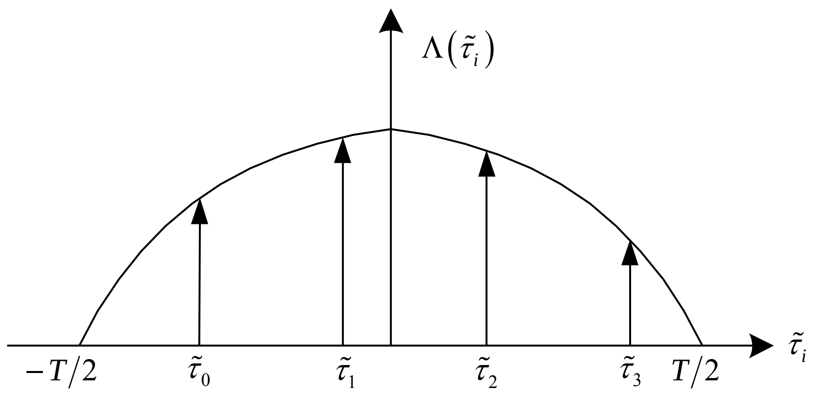
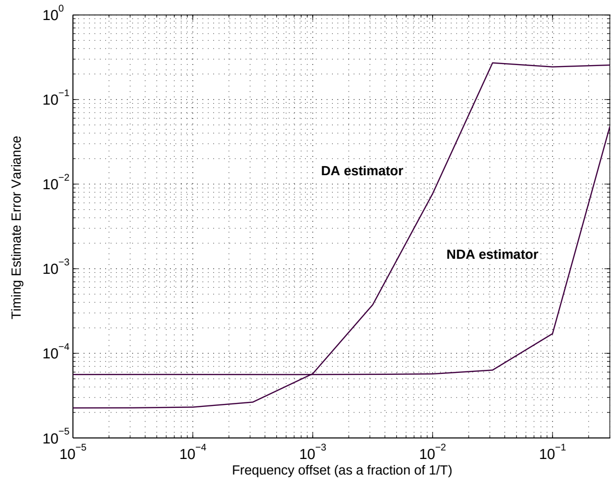


Figure 1.

Error variances of the DA and NDA timing estimators for SNR=20 dB. Block length is 64symbols and modulation is 16-QAM. Also the effect of frequency offset is included.

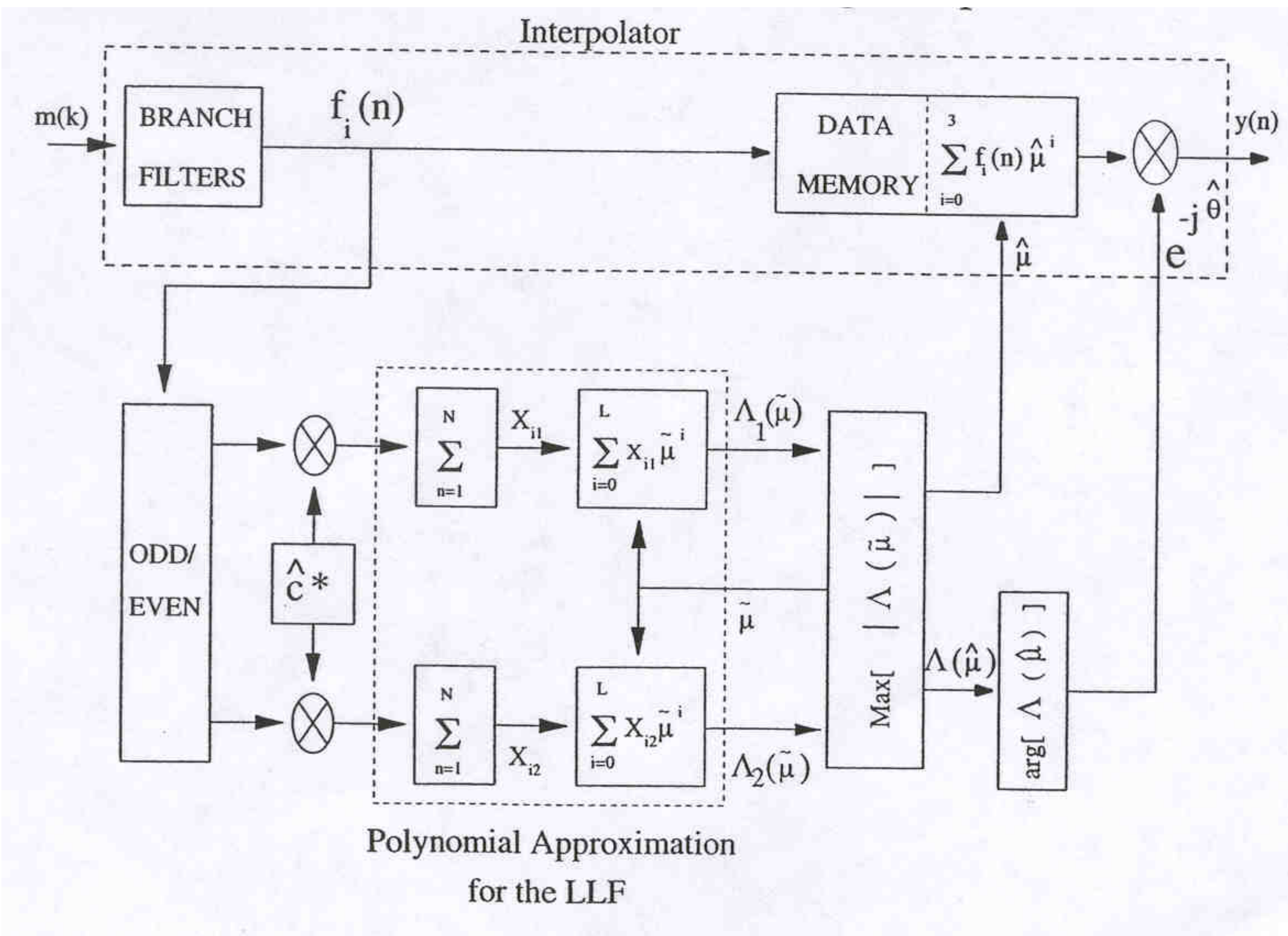


Interpolation based approach using the Farrow structure

The method developed at DCE (Ridha, Vesma, Tuukkanen, Renfors) starts from low-order (typically 3rd-order) piecewise polynomial model of the matched filter output signal. Oversampling by a factor of 2 is assumed in the studies. A piecewise polynomial model for the log-likelihood function can be calculated from the piecewise polynomial model of the signal. Especially, in the DA/DD case, the calculation of the log-likelihood function has lower computational complexity than using the model fitting approach.

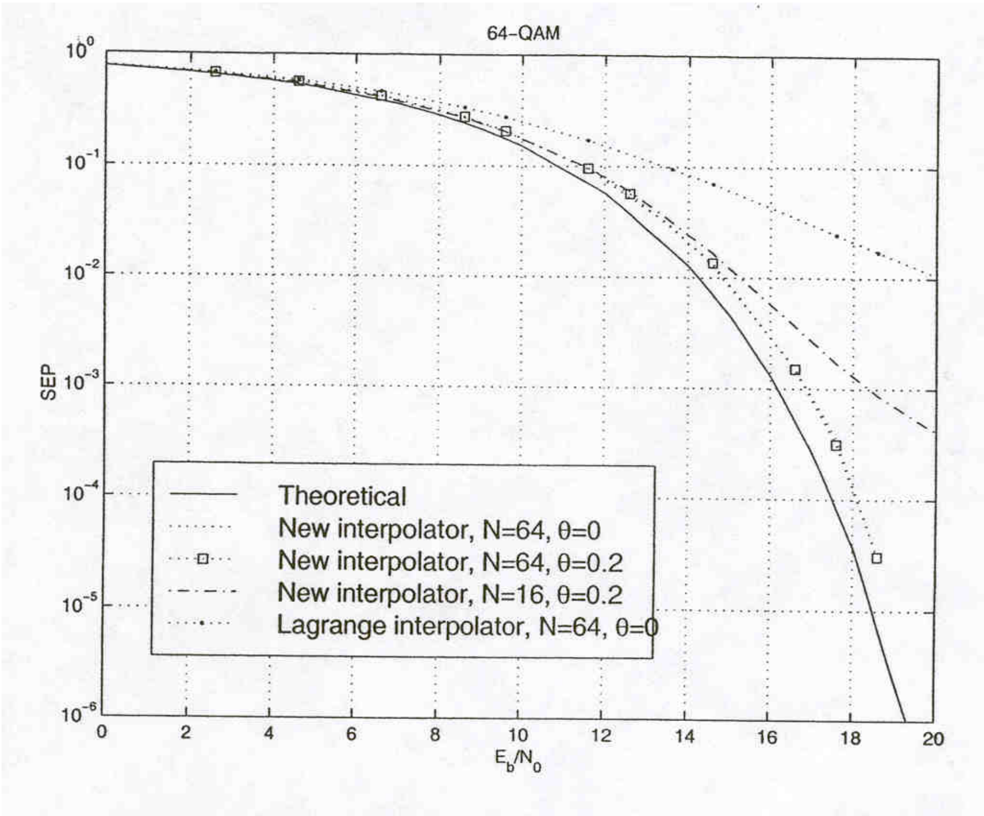
It should be noted that the piecewise polynomial model is quite useful also for sampling rate conversion by non-integer factors in multi-standard receivers. The symbol timing recovery function and the sampling rate conversion function can be efficiently combined by utilizing polynomial-domain signal processing.

Farrow-structure –based data-aided joint symbol timing and carrier phase recovery:



When low oversampling factor (e.g., 2x symbol-rate sampling) is used, the interpolator design has significant effect on the synchronization quality. Frequency-domain optimized interpolators give clearly better performance than the traditional Lagrange interpolators.

The following figure shows performance simulation results for Farrow-based recovery in the DA case. Here *N* is the estimation block-length in symbol intervals.



Performance Limits of ML-Based Synchronizers [2]

The ultimate accuracy that can be reached with MLE-based estimators in terms of estimate variance is the Cramer-Rao bound (CRB). The CRB is defined as the minimum attainable variance for an estimate recovered from a particular signal with a specified observation time. The CRB of parameter λ is written as

$$\operatorname{var}\left\{\hat{\lambda}(\mathbf{r})-\lambda\right\} \geq \mathrm{CRB}(\lambda).$$

where

$$\mathrm{CRB}(\lambda) \hat{=} -\frac{1}{E_{\mathbf{r}}\left\{\dfrac{\partial^2 \ln L(\mathbf{r}|\lambda)}{\partial \lambda^2}\right\}} = \frac{1}{\mathrm{E}_{\mathbf{r}}\left\{\left[\dfrac{\partial \ln L(\mathbf{r}|\lambda)}{\partial \lambda}\right]^2\right\}}$$

and $\mathrm{E}_{\mathbf{r}}\{.\}$ is the expectation of the enclosed quantity with respect to the observation **r**. and **r** is a vector of the synchronization parameters, and, in the NDA case, also the data symbol vector.

The CRB is often too difficult to express analytically. The difficulty arises from the necessity of computing $L\left(\mathbf{r}\middle|\tilde{\lambda}\right)$, which calls for averaging $L\left(\mathbf{r}\middle|\tilde{\lambda},\tilde{\mathbf{u}}\right)$ over the unwanted parameters (other synchronization parameters than λ, and, in the NDA case, also the data symbol vector) as follows:

$$L\left(\mathbf{r}\middle|\tilde{\boldsymbol{\lambda}}\right) = \int_{-\infty}^{\infty} L\left(\mathbf{r}\middle|\tilde{\boldsymbol{\lambda}},\tilde{\mathbf{u}}\right) p(\tilde{\mathbf{u}})\, d\tilde{\mathbf{u}}$$

This is usually not feasible because the integration can only be performed analytically in a few isolated cases.

The route to overcome this drawback is to use the *modified* CRB (MCRB), which still applies to any unbiased estimator. It is defined as

$$\operatorname{var}\left\{\hat{\lambda}(\mathbf{r})-\lambda\right\} \geq \mathrm{MCRB}(\lambda)$$

where

$$\mathrm{MCRB}(\lambda) \hat{=} \frac{N_0}{E_{\mathbf{u}}\left\{\int_0^{T_0}\left|\frac{\partial s(t,\lambda,\mathbf{u})}{\partial\lambda}\right|^2 dt\right\}}$$

for passband signals ($N_0/2$ in the numerator for baseband signals).

Example: The normalized CRB for timing, assuming known data and frequency offset, can be derived as [1], [2] (this is one of the cases where actual CRB can be found):

$$CRB(\hat{\varepsilon}) = \frac{1}{8\pi^2\xi N_0}\frac{1}{E_s/\eta}.$$

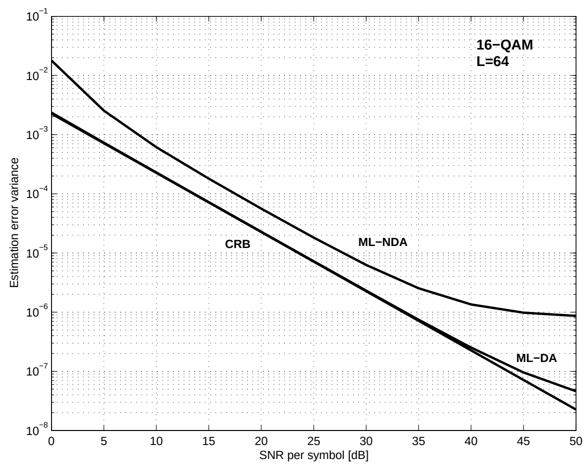
Here, E_s/η is the signal-to-noise ratio per symbol. The parameter ξ is defined as

$$\xi = T^2\frac{\int_{-\infty}^{\infty} f^2\left|G(f)\right|^2 df}{\int_{-\infty}^{\infty}\left|G(f)\right|^2 df}$$

and can be viewed as the normalized mean square bandwidth of $G(f)$. If $G(f)$ has a root-raised cosine roll-off, the above equation reduces to

$$\xi = \frac{1}{12} + \alpha^2\left(\frac{1}{4} - \frac{2}{\pi^2}\right).$$

Here the Sabel & Cowley DA and NDA estimator variances are compared to the Cramer-Rao bound:



Literature

[1] H. Meyr, M. Moeneclaey, S.A. Fechtel, *Digital Communications Receivers - Synchronization, Channel Estimation, and Signal Processing*. Wiley 1998.

[2] U. Mengali, A.N. D'Adrea, *Synchronization Techniques for Digital Receivers*. Plenum Press 1997.

[3] L.P. Sabel, W.G. Cowley, "Block processing feedforward symbol timing estimator for digital modems," *Electronic Letters*, vol. 30, no. 16, pp. 1273-1274, Aug. 1994.

[4] R. Hamila, J. Vesma, and M. Renfors, ”Polynomial-Based Maximum Likelihood Technique for Synchronization in Digital Receivers,” *IEEE Trans. Circuits Syst. II*, vol. 49, pp. 567-576, Aug. 2002.

[5] L. Anttila, *Baseband Signal Processing for a Narrowband Radio Modem.* M.Sc. Thesis, TUT, Dept. of Electrical Engineering, Jan. 2004.

[6] F. M. Gardner, *Frequency Detectors for Digital Demodulators via Maximum Likelihood Derivation*, Final Report: Part II: ESTEC Contract No. 8022/88/NL/DG, Gardner Research Company, 1990.

[7] P. Z. Fan, M. Darnell, *Sequence Design for Communications Applications*, Research Studies Press, John Wiley & Sons Ltd, London, 1996.

[8] S. Kay, *A Fast and Accurate Single Frequency Estimator*, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 37, No. 12, pp. 1987-1990, December 1989.

[9] M. P. Fitz, *Planar Filtered Techniques for Burst Mode Carrier Synchronization*, paper 12.1.1, GLOBECOM ’91, Phoenix, Arizona, December 1991.

[10] U. Mengali, M. Morelli, *Data-Aided Frequency Estimation for Burst Digital Transmission*, IEEE Transactions on Communications, Vol. 45, No. 1, pp. 23-25, January 1997.

[11] S. Tallak, R. Kubichek, J. Schroeder, *Time Delay Estimation for Objective Speech Quality*, IEEE Pac. Rim Conference on Communications, Computers and Signal Processing, Victoria, Canada, May 1993.

[12] E. Sourour, G. E. Bottomley, *Effect of Frequency Offset on DS-SS Acquisition in Slowly Fading Channels*, In Proc. Wireless Communications and Networking Conference, New Orleans, Louisiana, Vol. 2, pp. 569-573, September 1999.

APPEDIX: Additional Issues in Synchronization [5]

A1. DA Delay Estimation with Envelope Correlation

The ML principle can also be applied to delay estimation (or frame synchronization), i.e., to detect the beginning of a data burst and provide an initial timing estimate. However, two *ad hoc* schemes, whose operation principles closely resemble ML estimation, are reviewed here. The estimators are data-aided and use *envelope* or *squared-envelope correlation* between the received signal and the (local replica of the) transmitted training signal. The estimators can operate either before the matched filter, in which case they can handle large frequency offsets (frequency operation range restricted only by the prefilter band-width and/or the sample rate), or after the matched filter, when the usual approximate restriction $|\nu_0 T| < 0.15$ holds true.

The received signal is once again

$$r(t) = e^{j(2\pi\nu_0 t+\phi_0)} \sum_n a(n) g_T(t - nT - \varepsilon_0 T) + w(t). \quad \textbf{(1)}$$

The problem of delay estimation is to determine ε_0.

To eliminate the unwanted parameters we introduce

$$\begin{aligned} r_{env}(t) = |r(t)| &= \left| e^{j(2\pi\nu_0 t+\phi_0)} \sum_n a(n) g_T(t - nT - \varepsilon_0 T) + w(t) \right| \\ &\approx \left| \sum_n a(n) g_T(t - nT - \varepsilon_0 T) \right| \end{aligned} \quad \textbf{(2)}$$

for the envelope correlator and

$$\begin{aligned} r_{sq}(t) = |r(t)|^2 &= \left| e^{j(2\pi\nu_0 t+\phi_0)} \sum_n a(n) g_T(t - nT - \varepsilon_0 T) + w(t) \right|^2 \\ &\approx \left| \sum_n a(n) g_T(t - nT - \varepsilon_0 T) \right|^2 \end{aligned} \quad \textbf{(3)}$$

for the squared-envelope correlator. Here, a moderately high SNR is assumed and the noise terms are neglected. As can be seen, the dependency on ν_0 and ϕ_0 has been removed, but the dependency on *n* and ε_0 is retained.

The trial signals for the two estimators are now written as

$$\tilde{s}_{env}\left(t,\tilde{\varepsilon}\right)=\left|\tilde{s}\left(t,\tilde{\varepsilon}\right)\right|=\left|\sum_{n}a\left(n\right)g_{T}\left(t-nT-\tilde{\varepsilon}T\right)\right| \tag{4}$$

and

$$\tilde{s}_{sq}\left(t,\tilde{\varepsilon}\right)=\left|\tilde{s}\left(t,\tilde{\varepsilon}\right)\right|^{2}=\left|\sum_{n}a\left(n\right)g_{T}\left(t-nT-\tilde{\varepsilon}T\right)\right|^{2} \tag{5}$$

The trial signal has only one unknown parameter, ε_0. The correlation operations can now be written as

$$\begin{aligned}X_{env}\left(\tilde{\varepsilon}\right)&=\int_{0}^{T_0}r_{env}\left(t\right)\tilde{s}_{env}\left(t,\tilde{\varepsilon}\right)dt\\&=\int_{0}^{T_0}\left|r\left(t\right)\right|\left|\tilde{s}\left(t,\tilde{\varepsilon}\right)\right|dt\end{aligned} \tag{6}$$

and

$$\begin{aligned}X_{sq}\left(\tilde{\varepsilon}\right)&=\int_{0}^{T_0}r_{sq}\left(t\right)\tilde{s}_{sq}\left(t,\tilde{\varepsilon}\right)dt\\&=\int_{0}^{T_0}\left|r\left(t\right)\right|^{2}\left|\tilde{s}\left(t,\tilde{\varepsilon}\right)\right|^{2}dt\end{aligned} \tag{7}$$

The delay estimate is then found as the maximum point of the correlation,

$$\hat{\varepsilon}=\arg\max_{\tilde{\varepsilon}}X\left(\tilde{\varepsilon}\right). \tag{8}$$

It is worth noting, that the correlation operations in (3) and (4) clearly resemble the likelihood function, even though they were derived ad hoc. The envelope correlation method has also been used for time alignment of speech signals in [11]. In direct sequence spread spectrum (DS-SS) receivers, the use of a similar method for code acquisition has been studied for example in [12]. However, in the DS-SS scheme, the squared magnitudes of a number of short *coherent* correlations are combined.

It should also be noted, that taking the absolute value of, or squaring a signal doubles its bandwidth. When implementing a discrete-time version of either of the estimators, the sampling rate should be high enough so that no aliasing occurs.

Using sequences with good auto-correlation and cross-correlation properties as the training sequence could enable a multiuser system, in which the modems would be distinguished by different codes. One well-known code family meeting these criteria is Gold codes, and they are the code set chosen for the system. For a thorough explanation of Gold sequences, refer to [7].

Another merit of using this kind of timing estimation is that timing-directed (and possibly data-aided) frequency estimators can now be employed. However, even if timing is recovered accurately, it does not guarantee the correct operation of data-aided frequency estimation algorithms. This is due to the fact that for a frequency offset in the order of $1/T$ the signal will be severely distorted when passed through the matched filter. The approximate ML algorithm that was tested for the system under development does, however, perform surprisingly well with frequency offsets of up to 30 % of the symbol rate, and can thus be used for the coarse frequency acquisition stage if the frequency uncertainty is within this limit. Another option would be some timing-directed but non-data-aided algorithm that operates on $1/T_s$ -rate samples.

An obvious downside of correlation is its relatively high computational load, and hence, high power consumption. If this is not a limiting factor in modem design, correlation is a good option to providing an accurate delay estimate with large frequency offsets.

A2. ML Estimation of Carrier Frequency Offset

Carrier recovery is usually performed in two stages, i.e., non-timing directed *coarse recovery* in the analog domain or in the digital domain at rate $1/T_S$, and timing directed *fine recovery* with greater accuracy at rate $1/T$. This approach also leads to different ML algorithms for both cases.

Non-Timing Directed Estimation

For a frequency offset of $|\nu_0 T| \approx 1$, timing can generally not be recovered reliably (unless, e.g., envelope correlation is employed). In other words, the frequency offset must be corrected first before the other parameters and the data can be estimated. Hence, operation is non-timing directed and non-data aided. Two estimators for coarse frequency acquisition are reviewed here shortly, namely the maximum-searching frequency estimator and the dual filter detector.

Maximum-Searching Frequency Estimator

The low SNR approximation of the likelihood function

$$L(\tilde{\varepsilon},\tilde{\nu}) = \sum_{n=-L}^{L} \left| m\left(nT + \tilde{\varepsilon}T, \tilde{\nu}\right) \right|^2 . \tag{9}$$

for the joint estimation of symbol timing and frequency offset can be used as the starting point when deriving the estimator. It can be shown that there exists an unbiased estimator that requires no timing information.

Assuming that

1. the sampling rate fulfills $1/T_s > 2(1+\alpha)/T$
2. the ratio $T/T_s = M_s$ is an integer
3. data $\{a(n)\}$ is i.i.d.

the estimator

$$\hat{\nu} = \arg\max_{\tilde{\nu}} \sum_{n=-LM_s}^{LM_s-1} \left| m(nT_s, \tilde{\nu}) \right|^2 . \tag{10}$$

is an unbiased estimator [1].

Figure 1 shows the block diagram for the NDA maximum searching frequency estimator. In short, the algorithm chooses *the* frequency offset trial value that leads to maximum energy transferred through the matched filter.

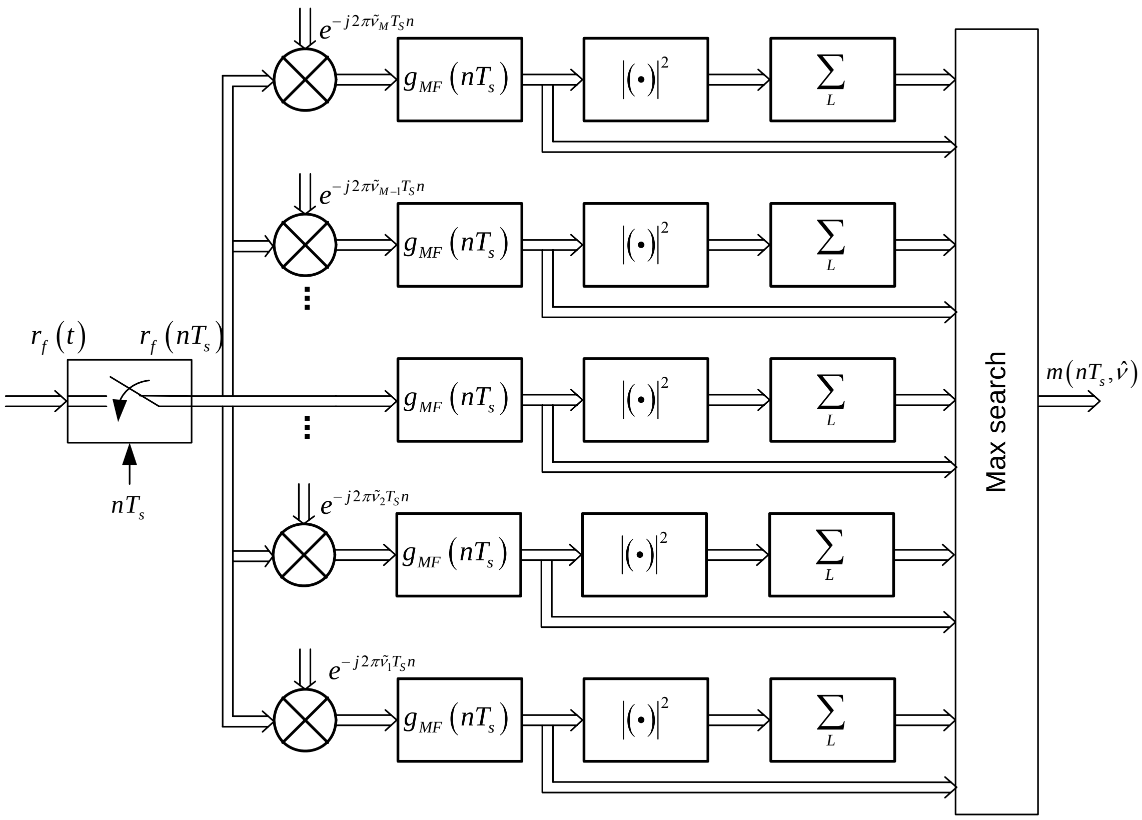


Figure 1. NDA maximum-searching frequency estimator

Dual-Filter Detector

The dual-filter detector (DFD) is an error feedback system, which basically measures the power difference of the outputs of two bandpass filters centered at $\pm 1/(2T)$. The output of the detector is an error signal, which is then ran through a loop filter and used to adjust the frequency offset estimate. The detector was originally developed ad hoc, but it can be derived systematically from the ML principle by making suitable approximations [1]. Skipping the details, the estimator structure is presented in Figure 2. Figure 3 illustrates the operation of the DFD by showing the power spectrum $R(f)$ of the received signal along with the amplitude responses of the two bandpass filters. The power spectrum of the received signal must be symmetric, and the lowpass prototype filter $h(t)$ must be real and symmetric for the detector to produce an unbiased estimate. [1], [2], [6].

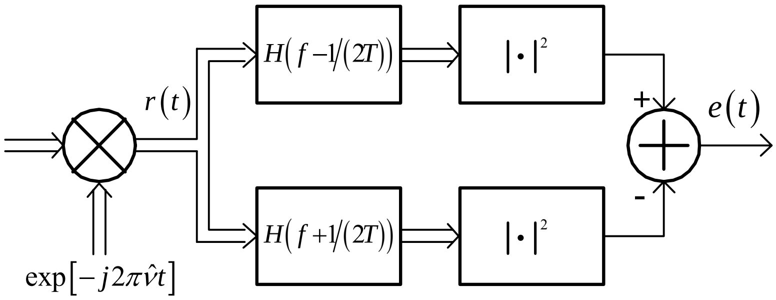


Figure 2. Dual filter detector block diagram

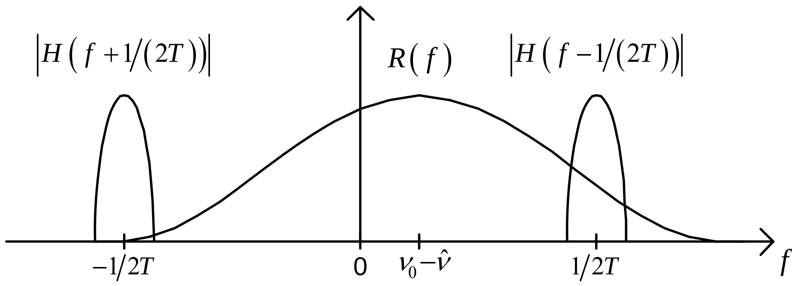


Figure 3. DFD operation

Timing-Directed Data-Aided Estimation

If the frequency offset is roughly $|\nu_0 T| < 0.15$ or less, timing can be recovered prior to frequency compensation. Operating in a timing-directed manner has two obvious merits: (1) clock-aided algorithms are generally more precise than the ones that operate without timing information, and (2) computational load is lighter, because the algorithm now runs at rate $1/T$ instead of $1/T_s$. The following assumptions are made: (1) timing is ideal, (2) data is known, (3) the frequency offset is a small fraction of the symbol rate, (4) pulse waveform is Nyquist, and (5) data symbols belong to a PSK constellation. Non-data-aided algorithms are also possible, but we concentrate on data-aided estimation.

Since timing and data are known, the likelihood function takes the form

$$L\left(\tilde{\nu},\tilde{\theta}\right)=\exp\left[\frac{2}{\sigma^{2}}\int_{0}^{T_0}\mathrm{Re}\left[r\left(t\right)\tilde{s}^{*}\left(t\right)\right]dt\right] \quad \textbf{(11)}$$

where

$$r(t)=e^{j(2\pi\nu_0 t+\phi_0)}\sum_{n}a(n)g_T(t-nT-\varepsilon_0 T)+w(t) \quad \textbf{(12)}$$

$$\tilde{s}\left(t\right)=e^{j\left(2\pi\tilde{\nu}t+\tilde{\phi}\right)}\sum_{n}a\left(n\right)g_T\left(t-nT-\varepsilon_0 T\right) \quad \textbf{(13)}$$

Again, ν_0 is the frequency offset, θ_0 is the carrier phase, ε_0 is the channel delay, $g_T(t)$ denotes the signal waveform, T the symbol interval, and $w(t)$ is white Gaussian noise with (two-sided) power spectral density $N_0/2$.

Writing out the integral, and making suitable approximations, the following log-likelihood function can be obtained [2].

$$\Lambda\left(\tilde{\nu}\right)=\left|\sum_{n=0}^{N_0-1}a^{*}\left(n\right)m\left(n,\tilde{\nu}\right)\right| \quad \textbf{(14)}$$

where

$$m(n,\tilde{\nu})=\int_{-\infty}^{\infty}r(t)e^{-j2\pi\tilde{\nu}t}g(t-nT-\varepsilon_0 T)dt\,. \quad \textbf{(15)}$$

The computation of $\Lambda\left(\tilde{\nu}\right)$ is illustrated in Figure 4. However, more practical estimators are needed, because the above likelihood function calls for computing $m\left(n,\tilde{\nu}\right)$ separately for each trial value $\tilde{\nu}$.

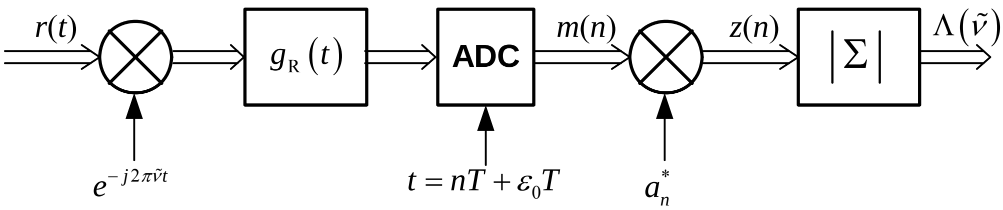


Figure 4. Illustrating the computation of the log-likelihood function

The computational burden could be significantly reduced if the positions of the phase rotator and the matched filter were interchanged, because the matched filtering operation would only have to be done once. The matched filter would of course distort the frequency-shifted received signal, but with small offset values the effect is insignificant. The derivation of the following practical estimators can be found in detail in [2]. Only the basic results are recapped in the following.

Feeding $r(t)$ into the matched filter $g_R(t)$ and sampling the output at instants $t=nT+\varepsilon_0 T$ produces, after some approximations [2]

$$m(n) = a(n)e^{j[2\pi\nu_0(nT+\varepsilon_0 T)+\theta_0]} + w'(n). \tag{16}$$

As can be seen, the signal $m(n)$ depends on ν_0, ε_0, θ_0 and the modulation a_n. The effect of modulation should be removed, which is an easy task when assuming M-PSK modulation with the property $a(n)a^*(n) = 1$. Removing the modulation produces

$$z(n) = e^{j[2\pi\nu_0(nT+\varepsilon_0 T)+\theta_0]} + w''(n). \tag{17}$$

which is the basis for most DA frequency offset estimation methods, including the two methods studied here. Figure 5 shows the computation of $z(n)$.

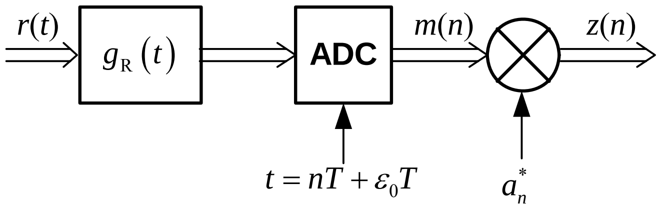


Figure 5. Illustrating the computation of $z(n)$.

Approximate ML Estimator

The Approximate Maximum Likelihood (AML) method is based on an approximation of the log-likelihood function of (14). Without going into details, the desired approximation for the log-likelihood function is [2]:

$$\Lambda(\tilde{\nu}) \approx \left| \sum_{n=1}^{N_0} z(n) e^{-j2\pi\tilde{\nu}nT} \right| \tag{18}$$

Compared to (14), the complexity is clearly reduced, since $m(n)$ need only be computed once. This naturally comes with the expense of a narrower frequency capture range, since the signal to be processed is now distorted by the matched filter.

The generation of $\Lambda(\tilde{\nu})$ can be made rapidly with the aid of FFT. The number of FFT points (i.e., the number of trial values $\tilde{\nu}$), the Doppler spread and the actual frequency offset determine the minimum estimation error. FFT length should be at least two times the length of the data block to attain a variance close to the CRB [2]. According to simulations in [2], the estimator is unbiased at least for frequency offsets $|\nu_0 T| < 0.15$.

Method by Fitz

The algorithm by Fitz [9] makes use of the autocorrelation of $z(n)$,

$$R(m) = \frac{1}{N_0 - m} \sum_{n=m}^{N_0-1} z(n) z^*(n-m), \quad 0 \le m \le N_0 - 1 \qquad \textbf{(19)}$$

where N_0 is the length of the observation interval. Skipping the details, the Fitz estimation formula is

$$\hat{\nu} = \frac{1}{\pi K(K+1)T} \sum_{m=1}^{K} \arg\{R(m)\} \qquad \textbf{(20)}$$

where K is a design parameter upper bounded by $K < 1/(2|\nu_{\max}|T)$. Intuitively, the algorithm calculates the average phase increment of successive samples, adds them up, and finally divides the sum with the observation time to obtain the frequency offset estimate. The original Fitz estimator introduced in [9] has quite a limited frequency capture range, even though the estimation range can be increased at the expense of estimation accuracy [2]. With large offsets, large estimation errors are also possible, because jumps of 2π are experienced when the value of $\arg\{R(m)\}$ crosses odd multiples of π. The estimation range could be extended by a post-processing unit that unwraps the phase, or by relating $\hat{\nu}$ to the phase increments $\phi(m) = \left[\arg\{R(m)\} - \arg\{R(m-1)\}\right]_{2\pi}$ rather than to $\arg\{R(m)\}$ itself, as was done in [8] and [10] for other similar estimators.