# Design Course: Design of Digital Satellite Radio Systems - From Algorithm to DSP Implementation

Meik Dörpinghaus, Andreas Senst, and Gerd Ascheid

Institute for Integrated Signal Processing Systems
RWTH Aachen University
Templergraben 55, 52056 Aachen, Germany
Telephone: +49 (241) 80 27878. Fax: +49 (241) 80 22195.
E-mail: meik.doerpinghaus@iss.rwth-aachen.de

*Abstract*— **In this educational paper we present the Design Course: "Design of Digital Satellite Radio Systems - From Algorithm to DSP Implementation" we are offering for electrical and communications engineering students at the RWTH-Aachen university. This design course consists of two parts, first an algorithm design part, treating the signal processing algorithm design of a digital satellite receiver. This algorithm development is based on the software simulation tool *CoCentric® System Studio (CCSS)*[1] from Synopsys. In the second part the receiver algorithms should be implemented on a digital signal processor (DSP) of Texas Instruments (TI) *TMS320C67x™ DSP generation*. To test the receiver implementation on the DSP a sound file is "transmitted". The simulation tool *CCSS* still simulates the transmitter and the channel and thus generates the "received" signal, which serves as an input signal to the DSP by using the Real-Time Data Exchange (*RTDX™*[2]) module. One remarkable aspect of this lab configuration is the fact that the simulation tool *CCSS* is running on a Linux system, whereas the host of the DSP-card is a Microsoft *Windows®*[3] based PC.**

## I. The Communication System Model

I general the model for a digital transmission line consists of three main blocks, the transmitter, the channel and the receiver (see Fig. 1).

### A. The Transmitter

The transmitter consists of a source, a symbol generator and a transmit filter. In the system we are considering, a QPSK modulation scheme is used. To relax the requirements for the phase synchronization, a differential pre-coding is performed. The root raised cosine transmit filter with the impulse response $g(t)$ is used for pulse shaping and limits the bandwidth of the transmit signal. Hence, the transmitted signal is given by

$$s(kT_s) = \sum_m a(m) g(kT_s - mT), \tag{1}$$

where $a(m)$ are the complex symbols. The quantity $T$ denotes the symbol period. The quantity $T_s$ denotes the sampling period. Here, we assume that the signal is oversampled by a factor of 4 and thus, for the whole simulation setup it is $T/T_s = 4$.

[1]CoCentric System Studio is a registered trademark of Synopsys Inc.

[2]RTDX is a trademark of Texas Instruments

[3]Windows is a registered trademark of Microsoft Corporation in the United States and other countries

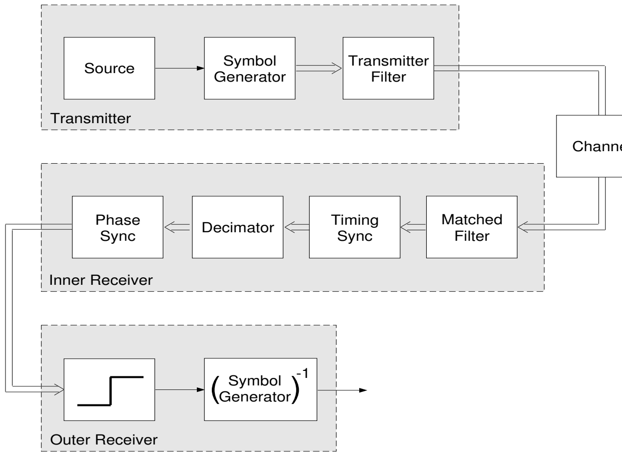


Fig. 1. Model of a communication system

### B. The Channel Model

Figure 2 depicts a model of the propagation channel like it holds for a satellite channel. Here, we make the simplifying, but realistic assumption, that the analog mixer units of the transmitter and the receiver do not have to be simulated. In the Figure, $s(t)$ denotes the transmitted signal and $r(t)$ denotes the received signal. It can be observed, that due to different disturbing mechanisms of a satellite channel, the received signal $r(t)$ is not identical to the transmitted one $s(t)$. In our lab we consider the following disturbing mechanisms:

1) Unknown channel fading $c(t)$ .
2) Unknown phase shift $e^{j\theta(t)}$.
3) Additive noise $n''(t)$.
4) Unknown channel delay $\delta(t - \varepsilon(t)T)$.

Hence, it holds

$$r(t) = e^{j\theta(t)} c(t) s(t - \varepsilon(t)T) + n''(t). \tag{2}$$

Even if in general $c(t)$, $\varepsilon(t)$ and $\theta(t)$ are time varying, it is assumed here, that the changes are negligibly small compared to the duration of transmission.

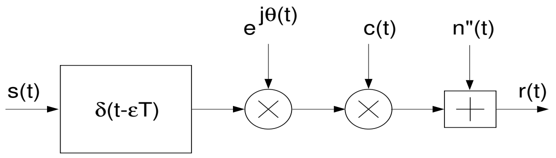


Fig. 2. Model of a propagation channel

### C. The Receiver

In our laboratory course, the focus is mainly on the inner receiver and its synchronization units with respect to satellite communication, where the channel changes only slowly with time. The task of these blocks is to reverse the effects of the channel, so that at the output of the inner receiver one obtains the original transmitted signal, disturbed by additive noise but no longer distorted. Finally, the decoding units, which recover the transmitted bit stream from the noisy signal, are located within the outer receiver.

#### *1) The Matched Filter*

After AD-conversion, the received signal passes the so-called matched filter $g^*(-kT_s)$. This filter serves to obtain the maximal ratio between signal and noise power $SNR = E_s/N_0$. Since a root-raised-cosine filter is used in the transmitter, the same filter is used in the receiver. After the matched filter it is

$$z(kT_s) = e^{j\theta_0} \sum_m a(m)h(kT_s - mT - \varepsilon_0 T) + n'(kT_s). \quad (3)$$

The resulting effective filter $h(kT_s)$ with $h(kT_s) = g(kT_s) \star g^*(-kT_s)$ is a raised-cosine filter and hence, fulfills the Nyquist-criterion $h(nT) = 0 \ \forall \ n \neq 0$. This implies that in principle a reception without inter-symbol-interference (ISI) is possible. If e.g. the time shift $\varepsilon_0$ and the phase error $\theta_0$ are zero, it holds

$$z(nT) = \sum_m a(m)h(nT - mT) + n'(nT) \quad (4)$$

$$= a(n)h(0T) + n'(nT). \quad (5)$$

The digital Matched filter (MF) – mostly implemented as an FIR filter – is only an approximation of the theoretically optimal filter, because, due to the finite bandwidth, it would have an impulse response of infinite length, but in reality it always has a finite length impulse response. In our lab it is the task of the students to find a compromise between acceptable complexity and tolerable performance loss. ISI-free reception means, that at the sampling instant $nT$ only the transmitted symbol $a(n)$ affects the signal $z(nT)$. However, in general it is $\varepsilon_0 \neq 0$ and $\theta_0 \neq 0$. Therefore, these quantities have to be estimated and their effects have to be compensated for, so that the decoding can be performed.

#### *2) Timing Synchronization*

The task of the timing synchronization is to reverse the effect of the unknown time shift $\varepsilon_0 T$. This can be reached if the time-scales of the transmitter and the receiver (see Fig. 3) are properly adjusted to each other. Therefore, having the signal sampled at time instants $kT_s$ (see equation (3)

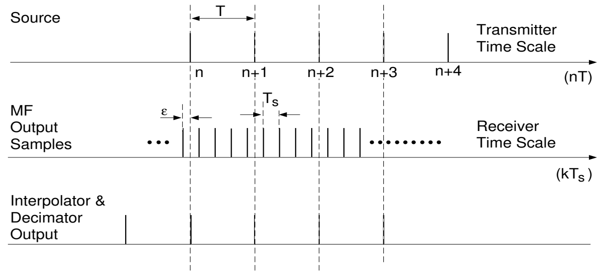


Fig. 3. To illustrate the time instant for interpolation

and illustrated by the solid vertical lines on the receiver time-scale in Figure 3) one has to generate the signal, that would have been obtained, if one had sampled the signal directly at the time instants $nT + \varepsilon_0 T$ (the vertical dashed lines in Figure 3). In fact, for the time instants $nT + \hat{\varepsilon}T$ with $\hat{\varepsilon} = \varepsilon_0$ the condition for ISI-free reception is fulfilled. It is

$$z(nT + \hat{\varepsilon}T) = e^{j\theta_0} \sum_m a(m)h(nT - mT - \varepsilon_0 T + \hat{\varepsilon}T) + n_n(6)$$

$$= e^{j\theta_0} a(n) + n_n, \quad (7)$$

where it has been used, that the transmit and receive filters are normalized in the way, that $h(0T) = 1$ (see equation (5)). In our lab the principle of synchronization via interpolation and decimation is used for timing synchronization. If a bandlimited signal is sampled with a sufficiently high sampling rate, arbitrary intermediate values can be calculated from the sequence of sampling values $z(kT_s)$ via the following equation

$$z(kT_s + \varepsilon T) = \sum_{l=-\infty}^{\infty} z(lT_s)p((kT_s + \varepsilon T) - lT_s). \quad (8)$$

Here, the function $p(lT_s, \varepsilon T)$ denotes an interpolator, operating on rate $T/T_s$. The quantity $z(nT + \varepsilon T)$ can be obtained from $z(kT_s + \varepsilon T)$ by performing a sampling rate reduction by a rate of $T/T_s : 1$ (see Figure 4).

In our lab we use a linear interpolator, which is a function of the estimated value $\hat{\varepsilon}$. The generation of this estimation value is dealt with in the following section.

*The Timing Estimation Algorithm*

The basic idea of the timing estimation algorithm is to generate an estimation value of the unknown quantity $\varepsilon_0$ directly from the sampling values $z(kT_s)$ of the received signal [1]. For the derivation of the estimator one uses the fact, that the quantity

$$x_k = |z(kT_s)|^2 \quad (9)$$

contains a spectral component $X(1/T)$ at $1/T$ and the phase of this component

$$\hat{\varepsilon} = -\frac{1}{2\pi} \arg\{X(1/T)\} \quad (10)$$

builds an unbiased estimator of $\varepsilon_0$. The spectral component $X(1/T)$ can be calculated by performing a Discrete Fourier Transform (DFT) on the sampling values. In general, the equation for a DFT is as follows:

$$X(k) = \sum_{n=0}^{NDFT-1} x_n e^{-j2\pi nk/NDFT}. \quad (11)$$

Here, NDFT denotes the number of sampling values, that are used for calculating the DFT. Here, let $NDFT = LT/T_s$. Then, it follows

$$X_m(k) = \sum_{n=mT/T_s}^{(m+L)T/T_s-1} x_n e^{-j2\pi nk/(LT/T_s)}. \quad (12)$$

For the index $k$ of the DFT it is

$$k = f \cdot NDFT \cdot T_s = f \cdot L \cdot T = L, \quad (13)$$

where $f = 1/T$ has been set, since the spectral component $X(f = 1/T)$ has to be computed. Therefore one finally obtains

$$X_m(f=1/T) = X_m(k=L) = \sum_{n=mT/T_s}^{(m+L)T/T_s-1} x_n e^{-j2\pi nT_s/T}. \quad (14)$$

The multiplications with $e^{-j2\pi nT_s/T}$ can be implemented very easily for an oversampling of $T/T_s = 4$, since the exponent can only take four different values $\{1, -1, j, -j\}$. For example, if $L = 1$ one gets

$$X_m(1/T) = (x_{4m} - x_{4m+2} - j(x_{4m+1} - x_{4m+3}). \quad (15)$$

*Performance*

If $\hat{\varepsilon} \neq \varepsilon_0$, a complete correction of the timing error $\varepsilon_0 T$ is not possible. The larger the error is, the larger is the bit error rate of the whole system. One measure for the performance of the estimator is the variance of the estimation error $Var\{\hat{\varepsilon}\} = E\left[|\varepsilon_0 - \hat{\varepsilon}|^2\right]$. It can be shown that the excepted performance is

$$Var\{\hat{\varepsilon}\} = E\left[|\varepsilon_0 - \hat{\varepsilon}|^2\right] \approx \frac{1}{L}\frac{1}{E_s/N_0}. \quad (16)$$

On the average, the smaller the variance of the estimator, the smaller is the deviation of the estimated value $\hat{\varepsilon}$ from the true value $\varepsilon_0$ and the smaller is also the degradation of the bit error rate (BER). Hence, the desired performance in the sense of a tolerable estimation error which is linked to the respective effect on the BER, can be reached by a proper choice of $L$. The optimization of this parameter L concerning system performance and implementation complexity is the task of the students.

Note, that the estimator is independent of the unknown phase $\theta_0$ and does not assume any training symbols. It has just to be ensured, that the timing error $\varepsilon_0$ is constant for the duration of the estimation period. Depending on the dynamic of the timing error it may even be sufficient to calculate the estimation value only every $L_T$-th symbol period.

### 3) Phase Synchronization

The task of the phase synchronization is to reverse the effect of the unknown phase shift $\theta_0$. Hence, based on the signal

$$z(nT + \hat{\varepsilon}T) = e^{j\theta_0} a(n) + n_n, \quad (17)$$

which is obtained after timing synchronization and decimation, it is desired to generate a signal with a useful signal part, which solely depends on the transmitted signal:

$$z(nT, \hat{\varepsilon}, \hat{\theta}) = a(n) + \tilde{n}_n. \quad (18)$$

The easiest method to generate the desired signal is to derotate the signal via

$$z(nT, \hat{\varepsilon}, \hat{\theta}) = e^{-j\hat{\theta}} z(nT + \hat{\varepsilon}T), \quad (19)$$

where for a totally coherent transmission $\theta_0 = \hat{\theta}$ must hold (see Fig. 4). In case of using a differential pre-coding it is sufficient that $(\theta_0 - \hat{\theta}) mod \frac{\pi}{2} = 0$.

*The Phase Estimation Algorithm*

The basic idea of the phase estimation algorithm is to construct an estimation value of the requested quantity $\theta_0$ directly from the samples $z(nT + \hat{\varepsilon}T)$. The following estimator is considered [2]:

$$\hat{\theta}_m = \frac{1}{4} \arg \sum_{n=m-\frac{L_V-1}{2}}^{m+\frac{L_V-1}{2}} e^{j4\arg\{z(nT+\hat{\varepsilon}T)\}}. \quad (20)$$

Here, the fact is used that the argument multiplied by 4 does no longer depend on the transmitted symbols, because it is $a^4 \equiv 1$ for $a$ being a QPSK symbol. In general $\hat{\theta} = \theta_0$ does not hold due to the constrained codomain of the arg-function. However, it is easy to show that in the noiseless case $(\theta_0 - \hat{\theta}) mod \frac{\pi}{2} = 0$ always holds, which suffices due to the use of differential pre-coding.

*Performance*

A measure for the performance of the estimator is the variance of the estimation error $Var\{\hat{\theta}\} = E\left[\left|(\theta_0 - \hat{\theta}) mod \frac{\pi}{2}\right|^2\right]$. It can be shown that the excepted performance is

$$Var\{\hat{\theta}\} \approx \frac{1}{L_V}\left(\frac{1}{E_s/N_0}\right). \quad (21)$$

On the average, the smaller the variance of the estimator, the smaller is the deviation of the estimated value $\hat{\theta}$ from the true value $\theta_0$ and the smaller is also the BER of the whole system. Hence, the desired performance in the sense of a tolerable estimation error which is linked to the respective effect on the BER, can be reached by a proper choice of $L_V$.

Note, that like the timing estimator this estimator does not assume any known training symbols. However, it has to be ensured, that the phase error $\theta_0$ is constant for the duration of an estimation period. In contrast to the timing parameter $\varepsilon_0$, this assumption does not always hold. Therefore, in general it is of great importance for the overall performance, that the estimation interval $L_V$ is set symmetric in time with respect to the estimation value $\hat{\theta}_m$.

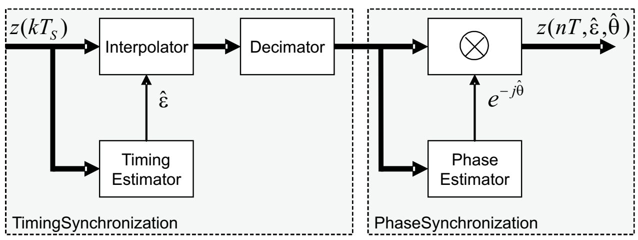


Fig. 4. Block diagram of the timing and phase synchronization units

*4) Outer Receiver*

After the synchronization tasks within the inner receiver have been completed, the decoding of the signal – within the outer receiver – is performed. In this case study no special coding scheme is intended within the transmitter and hence the following optimum decoding is performed within the receiver. The decision device chooses the respective symbol out of the set of the possible symbols, which has the smallest euclidian distance to the received noisy symbol. The block (symbol generator)$^{-1}$ maps the complex symbols back to a bit stream.

## II. Receiver DSP Implementation - Lab Configuration

The receiver algorithms that have been designed and optimized in the first part of the laboratory should be implemented on the DSP afterwards. Therefore we use a TI *TMS320C6701™ evaluation module (EVM)*. TI's *Code Composer Studio™ Integrated Development Environment (IDE)*[4] is used to programm the DSP.
The receiver implementation is tested by processing a "received" signal containing a sound file. The "received" signal is generated by the simulation tool *CCSS* which still simulates the transmitter and the channel. It is then used as input signal to the DSP using the *RTDX™* module. All the synchronization tasks and the decoding is performed by the DSP. Then the audio codec on the DSP card is used to give out the sound via a speaker (see Fig. 5).
One remarkable aspect of our laboratory configuration is that the simulation tool *CCSS* runs on a Linux system whereas the host of the DSP-card is a *Windows®* based machine. To transfer the data from the Linux system to the *Windows®* based PC we use the open source tool *Mono™*[5].

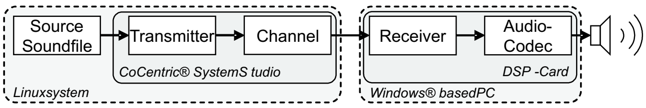


Fig. 5. Laboratory Configuration

### A. System Performance - Design specifications

The performance of a transmission system can be described by the BER as a function of the signal to noise ratio (SNR). In Figure 6 the BER for perfect synchronization and optimal decoding is plotted as a function of the SNR for a differential pre-coded QPSK-transmission. If we assume, that $h(0T) = 1$ (and therefore $z(nT) = a(nT) + n'(nT)$) holds, it follows

$$\frac{E_s}{N_0} = SNR = \frac{h_0^2}{E[|n'|^2]} = \frac{1}{\sigma^2}. \quad (22)$$

Here, $E[x]$ denotes the expectation of x and $E[|n'|^2] = \sigma^2$ the power of the complex noise quantity $n'$.
The results from Figure 6 can only be achieved if all

[4]Code Composer Studio is a trademark of Texas Instruments
[5]Mono is a trademark of Novell Inc.

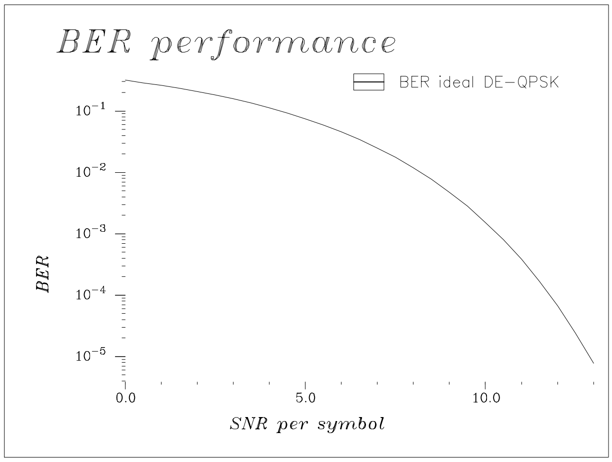


Fig. 6. Bit-error rate vs. SNR

components of the transmission line – especially the receiver components – are assumed to be ideal. A realistic receiver (with finite complexity) can reach these values only approximately. The reasons for this are the performance losses, which are mainly due to the matched filter and the imperfect synchronization.
The design specifications shown in Table I should be fulfilled during receiver design and implementation in our lab. The receiver should work for an SNR larger than 8 dB (per symbol). "Work" means that the theoretical optimum BER for a differential pre-coded QPSK-transmission is reached using an additional SNR of at most 0.2 dB. The reasons for the performance losses compared to the theoretical optimum are mainly due to the matched filter and the imperfect synchronization with finite complexity. The implementation complexity to reach the given design specifications is limited. The tasks of the students is to implement the developed receiver algorithms on the DSP. For the signal processing tasks the implementation is allowed to require at maximum 330 million cycles, corresponding to 3.3s on the 100 MHz DSP, to decode a soundfile of 8.128s sampled at 8 ksamples/s. Furthermore the codesize has to be reduced as much as possible.

| Variable | Value |
|---|---|
| SNR working range | $\geq$ 8 dB |
| SNR implementation loss | $\leq$ 0.2 dB |
| min. data rate | 25 kbit/s |
| coherence time of channel parameters $\theta$, $\varepsilon$ | 8 ms |

TABLE I
Design specification

## References


[1] M.Oerder and H. Meyr, "Digital filter and square timing recovery", IEEE Trans. on Communications, vol. 36(5), pp. 605-612, May 1988.
[2] A.J. Viterbi and A.M.Viterbi, "Nonlinear estimation of PSK-modulated carrier phase with application to burst digital transmission", IEEE Trans. on Information Theory, vol. 29(4), pp. 543-551, July 1983.