This document contains the specification of the transmission techniques for the C-band Satellite Radio system.

Authors:

dr hab. inż. Rafał Krenz (ed.)

dr inż. Marcin Rodziewicz

dr inż. Michał Sybis

prof. dr hab. inż. Krzysztof Wesołowski

Contents

List of acronyms

BB	Base-band
CBSR	C-band Satellite Radio
CRI	Code Rate Indicator
TC	Turbo Code

1. Link Budget

Radio link budget is on the most important parameter effecting the selection of transmission techniques which enable fulfilling the requirements of the system under development. It can be calculated given the parameters of the transmitter and receiver hardware, propagation conditions and other factors.

1.1. Link Model

The International Amateur Radio Union together with Radio Amateur Satellite Corporation have developed a family of tools for spacecraft and communication design. One of the tool, the AMSAT-IARU_Link_Model [1] can be used to calculate the link budget for any type of satellite communication systems, including Cubesats. It is an Excell spreadsheet, where user after entering several parameters can find the predicted system performance for the uplink and the downlink.

1.2. System Parameters

The link budget for CBSR system has been calculated based on the following assumptions:

- height of apogee: *626km*
- height of perigee: *606km*
- elevation angle: 10 – 90 deg.
- operating frequency: 5840MHz
- transmitter power: 2W (+33dBm)
- total transmitter line losses: 1dB
- total losses from antenna to LNA: 1,5dB
- antenna noise temperature: *150-290K* (function of elevation angle)
- ground station feedline noise temperature: 290K
- LNA noise temperature: 120K (F=1.5dB at 290K)
- LNA gain: 28dB
- feeder and jumper losses: 6dB
- communications receiver front end noise temperature: 2600K (F=10dB at 290K)
- transmit antenna gain: *6dB*
- receive antenna gain: 42dB
- transmit antenna pointing losses: *0.6dB* (20 deg.)
- receive antenna pointing losses: *10dB* (1 deg.)
- loss due to atmospheric gases: 1dB (at 10 deg. elevation angle)
- link losses resulting from signals passing through the ionosphere: 0.1dB
- ground station receiver bandwidth (channel bandwidth): 0.5 – 20MHz

Some of the parameters have been selected arbitrary (e.g orbit height) and are subject to change, those are printed in italic.

IMPORTANT: the transmit antenna gain of 0dB should be considered for the design process; therefore the curves in Fig. 1.1. should be shifted by 6dB to the lower values

1.3. Downlink Budget

The predicted link budget is expressed as signal-to-noise ratio at the demodulator input. Therefore, it is independent of the actual transmission technique, i.e. modulation, channel coding, etc. (Fig. 1.1).

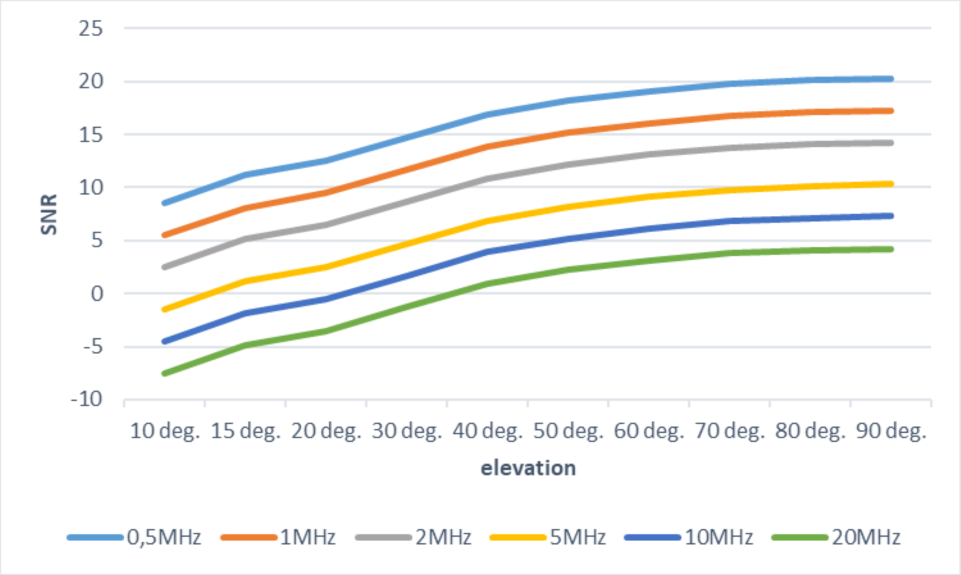

Fig. 1.1 Received SNR vs. elevation angle and channel bandwidth

2. Baseband Processing

2.1. Modulation

Modulation is one of the crucial procedures in transmission of digital signals over satellite links. Its selection has serious consequences for satellite link performance. When choosing a particular modulation scheme we have to take into account several limitations appearing for a satellite link. The main one is power efficiency of the on-board satellite transmitter, in particular due to the fact that in a nanosatellite system that is a subject of our design, source of energy is very limited. Typically, a source of energy are small solar cells. For such a system, an on-board power amplifier cannot be too advanced so the applied modulation should be possibly robust against amplifier nonlinearity. Desired power efficiency implies low value of peak-to-average power ratio (PAPR) of selected modulations. From that point of view a constant envelope modulation featuring PAPR=0 dB would be the optimum solution. Application of particular modulations in satellite communications is discussed in the Blue Book of the Consultative Committee for Space Data Systems (Recommendations for Space Data System Standards CCSDS 401.0.-B-30, February 2020) [2] and the informational report CCSDS 413.0.-G-3 [3] concerning space data system standards entitled ”Bandwidth-Efficient Modulations – Summary of Definition, Implementation and Performance” published in February 2018. The conclusions which can be drawn from both reports are straightforward: the modulations worth considerations are Gaussian Minimum Shift Keying (GMSK) and Offset Quadrature Phase Shift Keying. Both modulations were the subject of consideration and research in our project.

2.1.1. Modulation type

2.1.1.1. GMSK Modulation

GMSK modulation was presented by Moruta and Hirade in [4]. It is a frequency shift keying (FSK) modulation with the modulation index *h*=1/2 and carefully selected frequency shaping pulse *g*(*t*) given by the formula:

$$g(t) = \frac{1}{\sqrt{2\pi}\sigma T} \exp\left(\frac{-t^2}{2\sigma^2 T^2}\right) * \mathrm{rect}\left(\frac{t}{T}\right) \tag{2.1}$$

where *T* is the modulation interval, and rect(*x*) is the gate function. As we see, the frequency pulse *g*(*t*) is in fact a convolution of the Gaussian pulse with the gate function of length equal to the modulation interval. The Gaussian shape is determined by the product *σT*. In turn, the variable σ is determined by the expression:

$$\sigma = \sqrt{\ln 2}/(2\pi BT) \tag{2.2}$$

where *B* is a 3 dB bandwidth of the Gaussian pulse. Typically, *BT*=0.3. The phase pulse *q*(*t*) for GMSK modulation is an integral of the frequency pulse therefore it is given by the formula:

$$q(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}\sigma T} \exp\left[-\frac{\tau^2}{2\sigma^2 T^2}\right] * \mathrm{rect}\left(\frac{\tau}{T}\right) d\tau \tag{2.3}$$

The shape of this pulse is shown in Fig.2.1. It is clearly seen that due to the fact that the current phase *φ*(*t*) of the continuous phase modulated signal *x*(*t*)=cos(2*πf*$_c$*t*+*φ*(*t*)) is described by the formula (a_i is a bipolar data symbol):

$$\varphi(t) = 2\pi h \sum_{i=-\infty}^{n} a_i \int_{-\infty}^{t} g(\tau - iT)\mathrm{d}\tau \qquad \text{dla} \qquad nT \leq t < (n+1)T \tag{2.4}$$

several data symbols participate in the calculation of the current phase $\varphi(t)$. In Fig. 2.1. we can see that a few (mostly four or five, depending on *BT*) data symbols participate in generation of the current phase carrying data symbols.

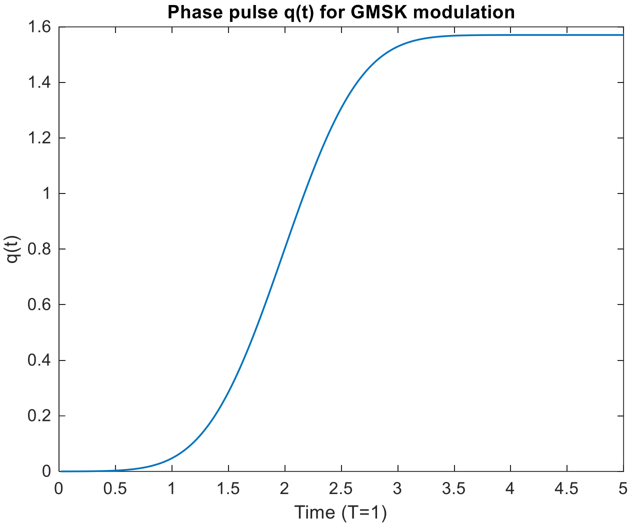

Fig. 2.1. Phase pulse shape for GMSK modulation with BT=0.3

To simplify implementation of GMSK modulator and to make it more universal for other potentially applicable modulations such as OQPSK and 8-PSK we decided to apply a linearized version of GMSK. It is based on the widely known paper by Laurent [5]. For *BT*=0.3 GMSK modulation can be approximated by a linear modulation with the carefully selected baseband pulse shaping filter $C_0(t)$. We have calculated the impulse response of this filter using the procedure proposed in [5]. For the details of the algorithm see [5]. The impulse response is shown in Fig. 2.2.

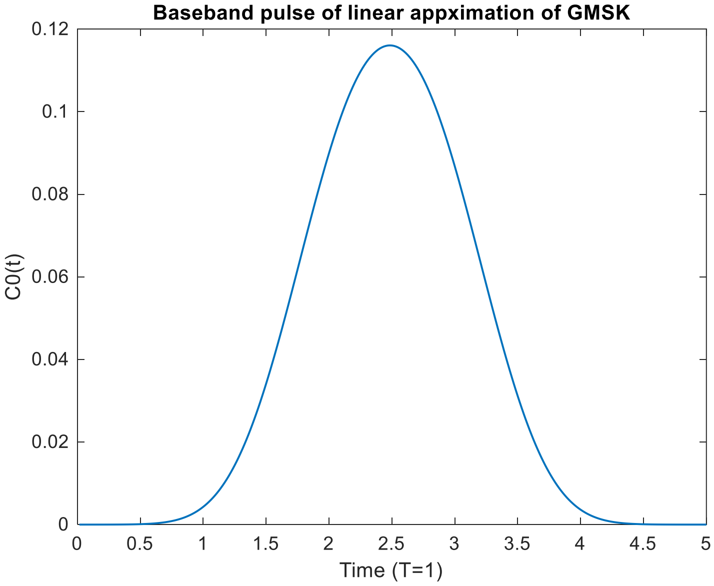

Fig.2.2. Calculated baseband pulse for linear approximation of GMSK for BT=0.3

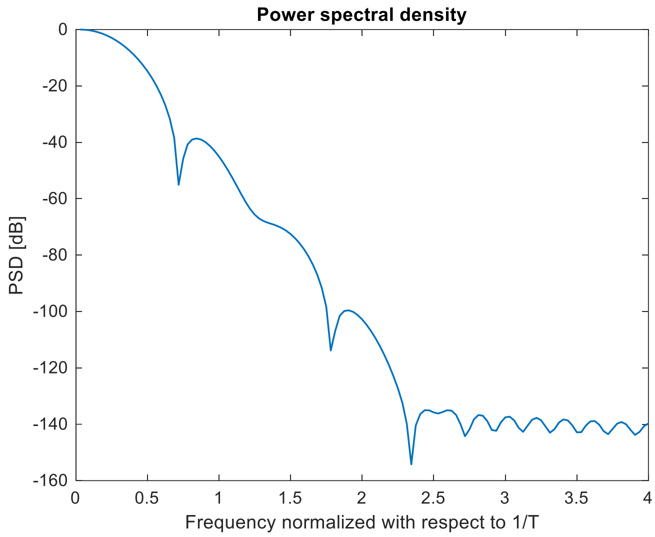

Fig. 2.3. Power spectral density of the baseband pulse of the linearized GMSK

Due to linear approximation the modulation loses its constant envelope, however PAPR value remains small. The envelope is visible in the form of the signal trajectories shown as the plot of points (*I*(*t*),*Q*(*t*)) of the modulated signal.

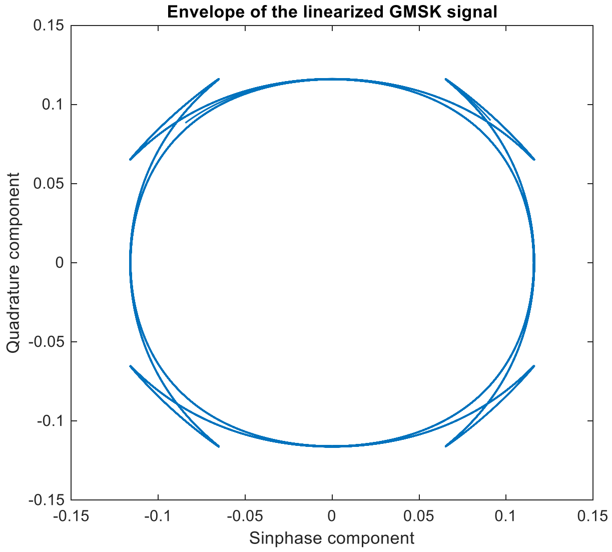

Fig. 2.4. Trajectory of the linearized GMSK signal

The scheme of the linearized GMSK transmitter is presented in Fig. 2.5.

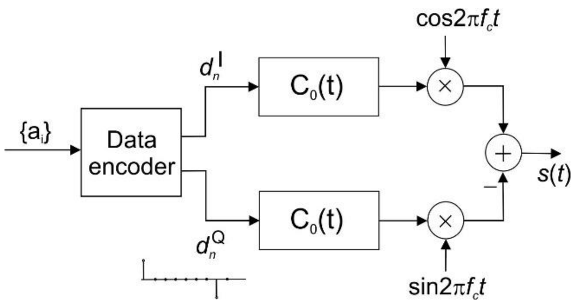

Fig. 2.5. Scheme of the GMSK linearized modulator

The data encoder generates in-phase and quadrature data symbols using the following rule:

$$d_n = d_n^I + jd_n^Q = d_{n-1}\exp\left[j\frac{\pi}{2}\left(2a_n - 1\right)\right], \quad d_0 = 1 \tag{2.5}$$

GMSK modulation and its linearized version are inherently differential. This means that, as in MSK, the phase features memory, i.e. the new phase after transmission of a new data symbol depends on the final phase achieved in the previous modulation period. One solution to this problem is to use differential decoder on the receiver side, however, it results in the SNR loss. Another solution is to apply a differential encoder before feeding data symbols to the modulator. Its scheme is shown in Fig. 2.6.

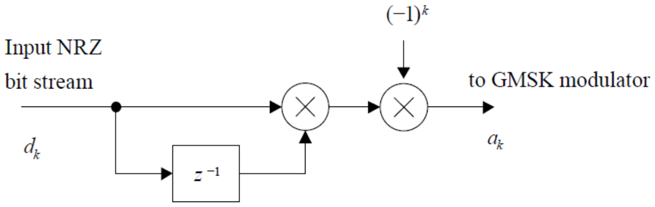

Fig. 2.6. Differential encoder in front of the GMSK modulator (dk and ak are bipolar data symbols)

As a result, the decisions in the receiver are made alternately on the in-phase and quadrature components on the receiver side.

The characteristic feature of GMSK modulation is intersymbol interference which has to be taken into account in the receiver. A typical solution for this problem is the application of sequential detection. Additional requirement is generation of soft output values on the output of the detector, as the next functional block is soft-input channel code decoder. A turbo code decoder is planned to be applied. The most popular soft-output decoder was proposed by Hoeher and Hagenauer [6]. They designed the soft output algorithm for Maximum Likelihood Forney type detector [7]. This type of the detector requires whitened matched filter. In order to avoid its synthesis, we have applied the Ungerboeck detector [8] requiring the application of the matched filter only and we supplemented it with the soft output. Using the algorithm proposed by Hoeher and Hagenauer in [6] we showed that the hard outputs astonishingly resulted in higher performance as compared with the soft outputs. The explanation of the author of the algorithm, namely Prof. Hoeher, indicated that this algorithm is suboptimal and works well for higher SNR, whereas in the considered satellite applications SNR values are low. We used convolutional codes for our experiments, although other simulation experiments proved that it is also true for turbo codes. In this situation the Hoeher-Hagenauer algorithm was abandoned and another one was tested. Among several proposals the one based on the US patent [9] turned to be easily implementable and simple.

Our receiver calculates hard symbols on the basis of Ungerboeck detector, however, it supplements it with the soft values helping to perform channel code decoding. The algorithm calculates theoretical values of LLR for subsequent data symbols which are soft inputs for the channel decoder. In the simple experiments convolutional code [133,171] was applied for that purpose. The LLR values are given by the formula:

$$\Lambda(d_k) = \ln\frac{\Pr\left(d_k = A \mid \mathbf{y}\right)}{\Pr\left(d_k = -A \mid \mathbf{y}\right)}, \quad \text{with } A = 1 \text{ or } A = j = \sqrt{-1},\ k - \text{even or odd} \tag{2.6}$$

If we assume that probabilities of d_k symbols are equal then

$$\Lambda(d_k)=\ln\frac{\Pr(d_k=A\,|\,\mathbf{y})}{\Pr(d_k=-A\,|\,\mathbf{y})}=\ln\frac{p(\mathbf{y}\,|\,d_k=A)\Pr(d_k=1)/p(\mathbf{y})}{p(\mathbf{y}\,|\,d_k=-A)\Pr(d_k=-1)/p(\mathbf{y})}$$
$$=\ln\frac{p(\mathbf{y}\,|\,d_k=A)}{p(\mathbf{y}\,|\,d_k=-A)} \tag{2.7}$$

Samples of the received signal used by the Ungerboeck detector are those which are seen on the output of the receive filter matched to the pulse shaping transmit filter. The joint impulse response of the cascade of the transmit and receive filter is

$$\mathbf{h}=C_0(t)*C_0(t)\big|_{t=iT}\quad i=-L,\ldots,0,\ldots,L \tag{2.8}$$

Therefore the samples on the input to the Ungerboeck detector are characterized by the formula

$$y_k=\sum_{i=-L}^{L}h_i d_{k-i}+\nu_k \tag{2.9}$$

where h_i are the elements of vector **h**. To determine soft outputs of the algorithm our aim is to find conditional probabilities of the sequence **y**, conditioned on the data sequence **d**. In calculations of soft decisions we assume that data symbols preceding and following the analyzed one are correct. This assumption is justified if hard output error probability is, say, in the order of 0.1 or less. For higher error probabilities such decisions would not be useful, anyway. The analyzed data symbol d_k contributes to the sequence of the following received samples

$$y_{k-2L},\ldots,y_{k-1,}y_k,y_{k+1},\ldots,y_{k+2L} \tag{2.10}$$

We conclude that assuming white Gaussian additive noise we can write the conditional probability density as:

$$p(y_{k+L},\ldots,y_{k+1},y_k,y_{k-1},\ldots,y_{k-L}\,|\,d_{k-2L},\ldots,d_{k-1},d_k,d_{k+1},\ldots,d_{k+2L})$$
$$=\left(\frac{1}{2\pi\sigma^2}\right)^{L}\prod_{i=-L}^{L}\exp\left\{-\frac{\left|y_{k+i}-\sum_{j=-L}^{L}h_j d_{k+i-j}\right|^2}{2\sigma^2}\right\} \tag{2.11}$$

Therefore, after simple calculations we receive the following expression for the LLR values:

$$\Lambda(d_k)=\ln\frac{p(\mathbf{y}\,|\,d_k=A)}{p(\mathbf{y}\,|\,d_k=-A)}=$$
$$=\frac{1}{2\sigma^2}\sum_{i=-L}^{L}\left\{-\left|y_{k+i}-\sum_{j=-L,\,j\neq i}^{L}h_j d_{k+i-j}-h_iA\right|^2+\left|y_{k+i}-\sum_{j=-L,\,j\neq i}^{L}h_j d_{k+i-j}+h_iA\right|^2\right\} \tag{2.12}$$

where A=1 or j depending on that if the time index k is even or odd. The same virtue is related to the subsequent data symbols d_{k+i-j}.

In order to evaluate quality of the soft decision calculations, some simulation experiments have been performed using GMSK linearized modulation, the Ungerboeck detector and the soft output calculator for several quantization levels presented above. The reference results for SOVA Viterbi detector [6] were also given. The applied channel code was convolutional one with [133,171] generators. The results presented in Fig. 2.7 show substantial superiority of the newly investigated soft output calculator as compared with the SOVA detector. The number of quantization levels very close to the optimum one is *N*=16. When *N*=8 the detection performance is only slightly worse than for *N*=16. SOVA is worse by several dB of SNR.

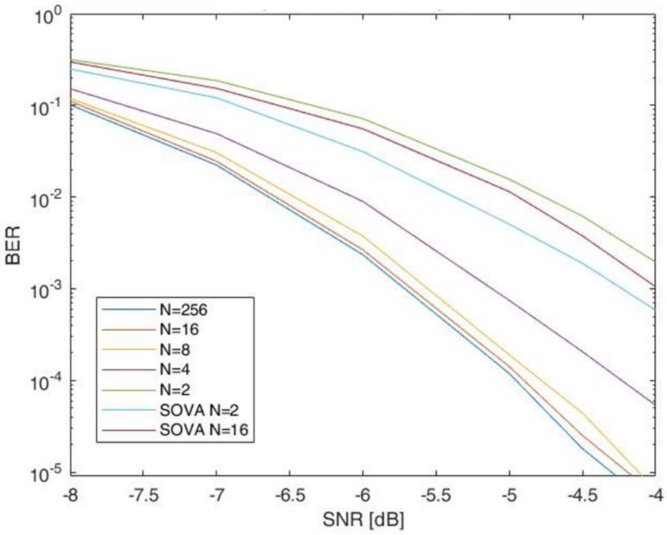

Fig. 2.7. BER versus SNR on the output of [133,171] convolutional code decoder fed with the LLR values calculated by the considered LLR calculator and SOVA algorithm

Our choice of modulation scheme should take into account not only the spectral properties and energy efficiency but also some other factors like implementation of synchronization algorithms. Let us recall that our system starts at low SNR values when the visible satellite is in the longest distance from the ground station. Application of sequential detection does not help to achieve robust and fast initial synchronization due to the delay of generation of reliable decisions which have to be used by the synchronization subsystem. Therefore this was one of the reasons to consider the second widely used modulation, namely OQPSK as an alternative to GMSK.

2.1.1.2. OQPSK Modulation

Offset Quadrature Phase Shift Keying (OQPSK) was designed for satellite communications as a modulation featuring low PAPR value. It is a modification of Quadrature Phase Shift Keying in which a quadrature branch in the transmitter is delayed by a half of a modulation period. The basic scheme of the OQPSK modulator is shown in Fig. 2.8.

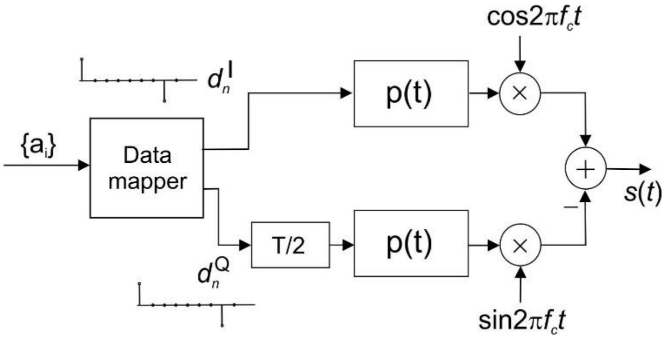

Fig. 2.8. Basic scheme of the OQPSK modulator

The envelope of the OQPSK signal applied in the simulation experiments is shown in Fig. 2.9. Out of several possible pulse shaping filters $p(t)$, the square-root raised cosine filter with the roll-off factor equal to α=0.35 was applied. It is a compromise between spectral compactness and PAPR value. In the experiments it was assumed that we have 4 samples of the channel impulse response per modulation interval and the impulse response of the pulse shaping filter spans 8 modulation intervals. Other oversampling values can be selected, too.

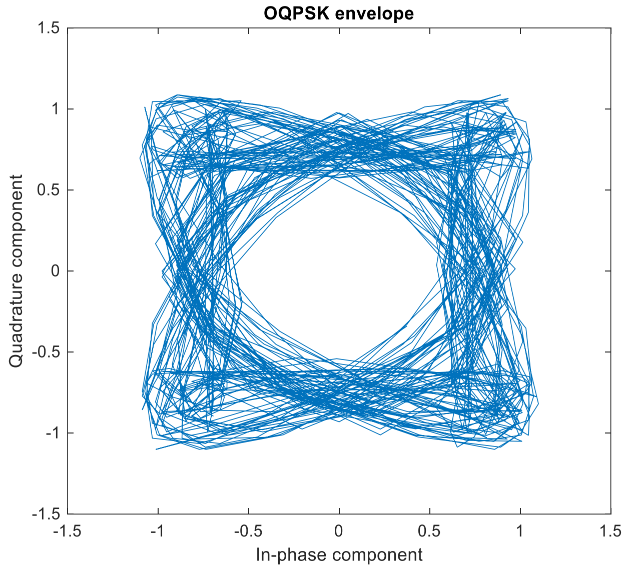

Fig. 2.9. Envelope of the OQPSK signal with the square-root raised cosine filter with α=0.35

Selection of α=0.35 implies the symbols rates R for the channel bandwidths B given in Table 2.1. Certainly the symbol rate and the channel bandwidth are associated with each other by the following inequality:

$$R(1+\alpha) \leq B \tag{2.13}$$

Table 2.1. The list of channel bandwidths and symbol rates

	Channel bandwidth *B*	QPSK symbol rate *R*
1	1.25 MHz	0.92 Msymb/s
2	5 MHz	3.7 Msymb/s
3	10 MHz	7.4 Msymb/s
4	20 MHz	14.8 Msymb/s

The OQPSK demodulator is based on the QPSK demodulator. The received signal, after down-converting, is filtered by the matched filter. In case of AWGN channel model the receive filters are matched to the transmit ones, so they have the same square-root raised cosine characteristics. The output signals are sampled at the moments $t=nT/K$, where K is oversampling value. In case of 4 samples per data symbol K=4. In the case of two samples per data symbol period (as shown in Fig. 2.10), K=2. The decision device, or LLR calculator in the case of soft demodulator output takes the pair of samples for analysis, where the quadrature one is delayed by $T/2$ with respect to the in-phase one. This way we get the sample pair which can be interpreted in QPSK signal coordinate system. For hard outputs the decision upon received constellation points are made. In the case of soft output, approximate LLR values for each transmitted bit are calculated according to the formula:

$$LLR(b_i) \approx -\frac{1}{\sigma^2}\left(\underset{\lambda_0 \in \Lambda_{0,i}}{\arg\min} \left| s-\lambda_0 \right|^2 - \underset{\lambda_1 \in \Lambda_{1,i}}{\arg\min} \left| s-\lambda_1 \right|^2 \right), \quad i=1,2 \tag{2.14}$$

where b_i denotes the *i*-th bit in a block mapped to a constellation point, $\Lambda_{0,i}$ and $\Lambda_{1,i}$ are the sets of constellation points which are mapped to zero bit or one bit on the i-th position and λ_0 and λ_1 are the constellation points mapped with zero bit and one bit, respectively, on the *i*-th position in the bit block mapped to a constellation point.

Selection of the square-root raised cosine filter characteristics of the pulse shaping and matched filters allows to assume that if the appropriate sampling phase is found, the samples are selected without intersymbol interference.

2.1.1.3. 8-PSK Modulation

After longer discussions it was decided not to consider 8-PSK modulation due to probable necessity of application of sequential detection and other reasons.

2.1.2. Simulation results

In our simulation experiments we assumed ideal sampling moments. We investigated the sensitivity of the OQPSK system to the frequency and phase offset for several lengths of the data blocks. Midambles that are basically needed for phase offset estimation were used in the form of oversampled and filtered Zadoff-Chu sequences. In the reported simulation results the midambles were based on ZC sequence of length 63 supplemented at the beginning and the end of it by 10 symbols from the end and beginning of the ZC sequence. The Zadoff-Chu sequence is given by the formula:

$$x_k(n) = \exp\left[-j\frac{\pi k n(n+1)}{N_{ZC}} \right] \tag{2.15}$$

where N_{ZC}=63 and *k*=1. The whole midamble consists of the blocks [A (ZC sequence) B], where A = [ZC sequence(N_{zc}-9:N_{zc})], and B=[ZC sequence(1:10)]. In order to maintain the spectral properties, the midamble is filtered by the filter with the square-root raised cosine characteristic and oversampling factor was equal to 4. In Fig. 2.10 the organization of the data blocks is shown.

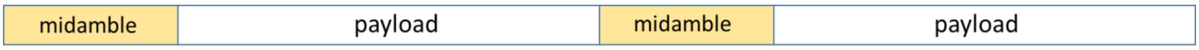

Fig. 2.10. OQPSK Frame organization

The transmitted sequence starts with a midamble that is followed by OQPSK data symbols of length N=330. Thus, 660 data bits are transmitted in each block. Then the next midambles and data blocks follow. In Fig. 2.11 magnitude of the data block and a single midamble is shown.

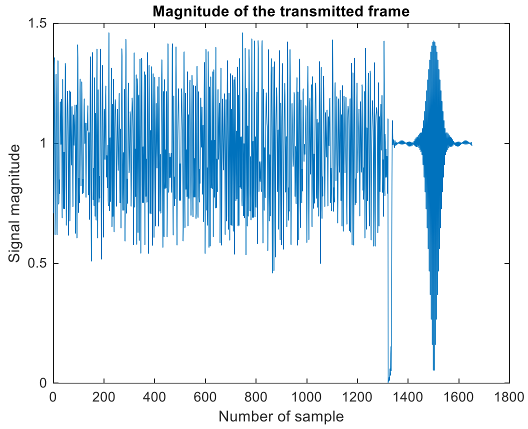

Fig. 2.11 Magnitude of the data block followed by the midamble

In Fig. 2.12 the efficiency of the correlative method by means of the filter matched to the ZC sequence is shown for SNR = 5 dB and 0 dB. It can be noticed that peaks on the output of the ZC matched filter are very strong. The phase shift caused by the channel can be estimated on the basis of the imaginary and real parts of the complex peak value after finding its place and determining its value. The clear peak indicates the beginning of the block as well.

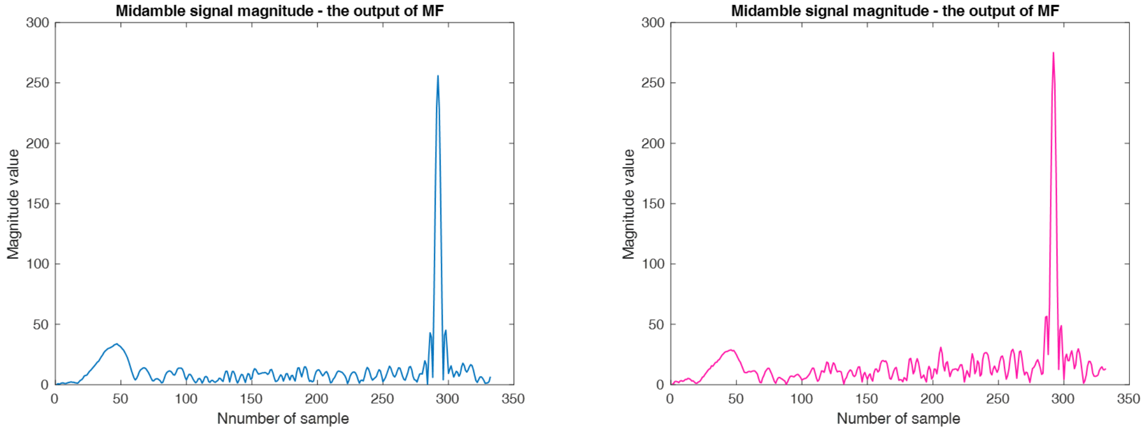

Fig. 2.12. Midamble signal magnitude on the output of the filter matched to the filtered ZC sequence for SNR=5 dB (left) and SNR=0 dB (right).

The estimated phase is given by the formula:

$$\varphi = \text{atan}\frac{Im(y_i)}{Real(y_i)} \tag{2.16}$$

where *i* is the time index for the maximum magnitude on the matched filter output. It is well known that atan function delivers results in the range $-\pi \leq \varphi < \pi$ therefore unwrapping process is necessary in real implementation of the algorithm. It means that if the change of phase exceeds a certain value, e.g. in the subsequent midambles the detected phases jump from a big positive to a negative value, the new phase should be corrected by adding 2π to it. Similar procedure should be performed if the phase values decrease in subsequent midamble moments and suddenly its value jumps to a positive one. Performed simulation experiments prove that the quality of determining the phase shift using this method is quite effective even if the SNR is low. In Fig. 2.13 the histogram of the phase shift estimate is shown for a real phase shift equal to 30 degrees for 100000 samples.

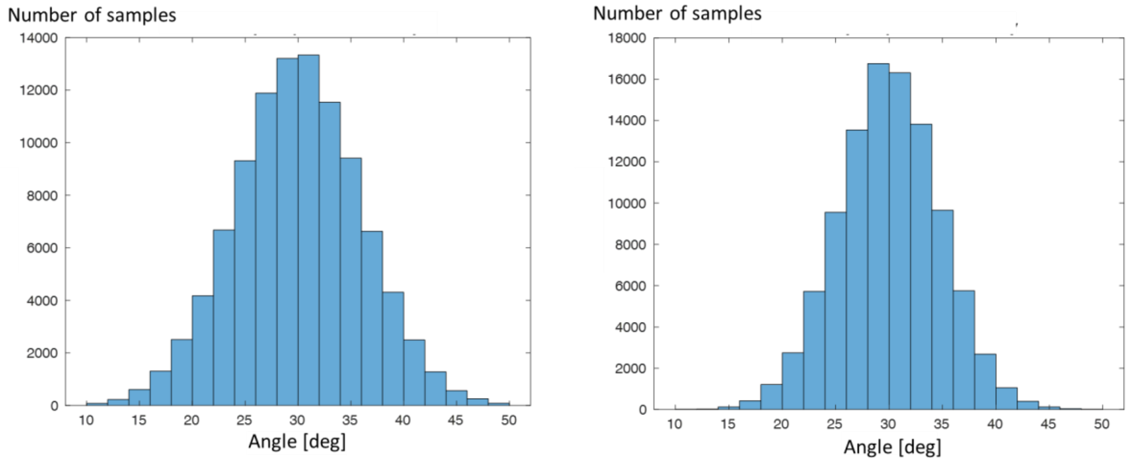

Fig. 2.13. Histogram of the phase shift estimates for SNR = 0 dB (left) and SNR = 2 dB (right one)

The phase shift estimated in two subsequent midambles can be theoretically applied to estimate the frequency offset as well. In case of a non-zero frequency offset, e.g. *Δf*, the phase change grows with time instant according to the expression:

$$\varphi(n\Delta t) = 2\pi\Delta f n\Delta t + \Delta\varphi + \varphi_0 \tag{2.17}$$

where $\Delta\varphi$ is the phase offset, φ_0 is the initial phase. If the distance between the subsequent peaks of the midambles is equal to *L* samples, then the difference between the estimated phases is:

$$\varphi(n_2) - \varphi(n_1) = \varphi(L + n_1) - \varphi(n_1) = 2\pi\Delta f\Delta tL \tag{2.18}$$

Therefore the estimated phase increment per sampling period Δt caused by the frequency offset Δf is

$$2\pi\Delta f\Delta t = \frac{\varphi(n_2) - \varphi(n_1)}{L} \tag{2.19}$$

This expression can be applied to compensate for the influence of the frequency shift in the received signal.

Several simulation experiments have been performed to check the quality of the transmission in the presence of phase and frequency offsets. The parameters of the simulated systems were the following:

- 330 OQPSK symbols in the transmitted data block (payload)
- 83 Zadoff-Chu symbols in the midamble
- Selected data rates contained in Table 2.1 (3.7 and 14.8 Mbit/s)
- Several values of frequency offsets (10 Hz, 1000 and 2000 Hz)
- Possibility of frequency offset compensation and leaving it without compensation

The simulation results are shown in Figs. 2.14 and 2.15.

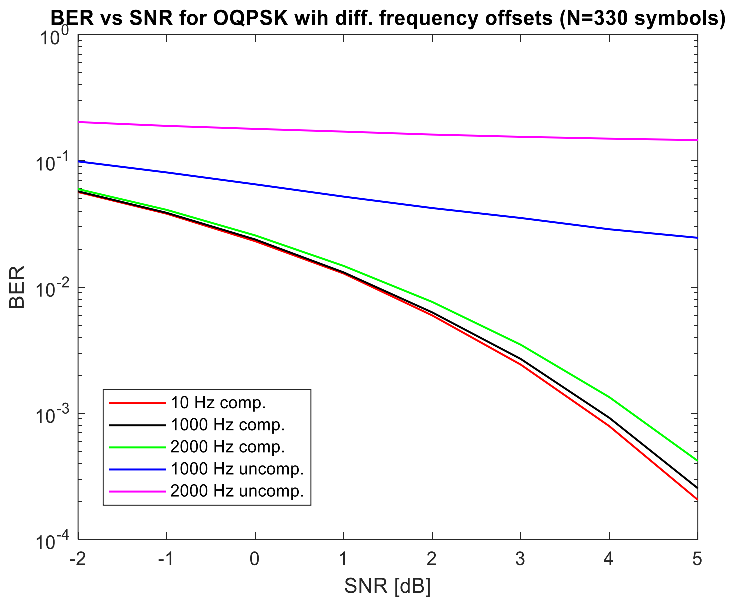

Fig. 2.14. BER versus SNR for OQPSK data symbols transmitted at the rate of 3.7 Mbit/s corrupted by frequency offset and phase offset of 30 degrees

It is clearly seen that the algorithm of frequency offset compensation is quite effective. The difference in SNR value required to achieve the BER value of about 10^{-3} is around 0.5 dB between the plots for the frequency offsets of 10 and 2000 Hz. The performance of the system with uncompensated frequency offset is unacceptable for the channel of 5 MHz bandwidth. Time span of 330 OQPSK symbols in the data block is so large that the phase shift caused by uncompensated frequency offset has destructive influence on the system performance.

In Fig. 2.15 similar results are shown for 20 MHz channel for which the data rate is assumed to be 14.8 Mbit/s.

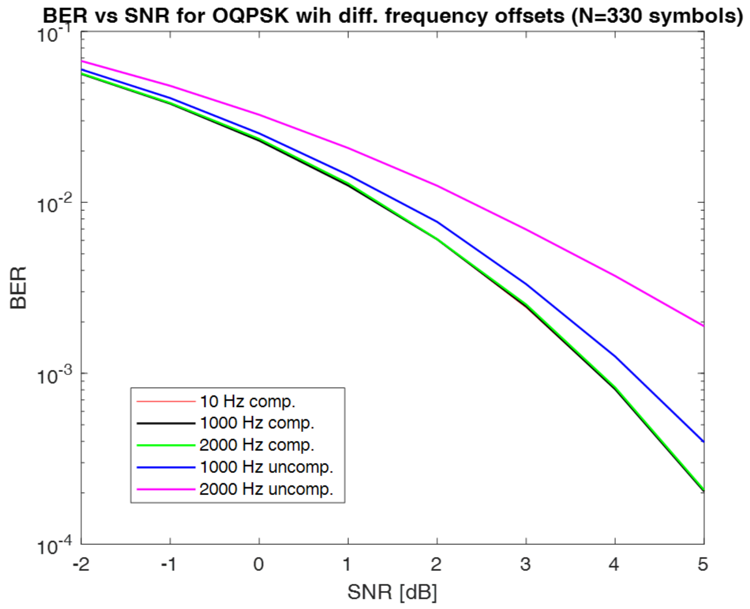

Fig. 2.15. BER versus SNR for OQPSK data symbols transmitted at the rate of 14.8 Mbit/s corrupted by frequency offset and phase offset of 30 degrees

This time the influence of compensation of frequency offset on the system performance is even better. There is practically no difference in BER plots for frequency offsets equal to 10, 1000 or 2000 Hz. Due to the fact that the same 330 symbols blocks are transmitted four times faster than for 5 MHz channel, influence of uncompensated frequency offset is much smaller, in particular for Δf=1000 Hz. However, leaving 2000 Hz of frequency offset uncompensated results in the loss of several dB in SNR for BER=10^{-3}.

The open question remains what the required value of uncoded system BER is to enable effective operation of turbo channel code resulting in low BER on the decoder output.

2.2. Synchronization

Synchronization is a key element of the receiver processing in a communication system. Its performance is crucial especially in a burst-mode transmission which is considered also in this work, where the acquisition time needs to be kept as low as possible. The synchronization task for the considered CBSR system is particularly challenging due to the expected operation with high sampling rates in very low SNR conditions and the expected very high frequency shift due to Doppler effect (over 100 kHz offset possible). Due to the required short acquisition time for synchronization mechanism, a data-aided (DA) feedforward structure is considered, where the synchronization procedure is divided into three tasks: frame (time) synchronization, frequency synchronization and phase synchronization with frequency tracking.

2.2.1. Functional blocks of synchronization system

The developed synchronization system relies on sequential processing of the received signal using specific functional blocks, depicted in Fig. 2.16:

1. The frequency of received signal is initially corrected using the coarse carrier frequency offset (CFO) estimates derived based on information of the orbital parameters of the satellite. Such parameters can be derived e.g. from the so-called Two Line Element (TLE), describing satellite position in a specific time, provided by the North American Aerospace Defense Command (NORAD) in public for investigating and tracking satellites. An exemplary procedure of processing of the TLE data in order to derive the estimated Doppler frequency shift is provided in [10]. Application of such initial correction should result in significant reduction in CFO of the further processed signal.
2. The frame detection and synchronization block is responsible for locating the start of a received radio frame, and, thus, identification of any relevant fields of the frame (preambles, data blocks, etc.).
3. The frequency offset estimation and correction block is responsible for fine estimation and removal of the remaining CFO, that, apart from the errors in Doppler shift compensation in the initial step, can be caused also by the local oscillators mismatch between the transmitter and the receiver.
4. The phase offset estimation and correction + frequency tracking block is responsible for finding and correcting the phase offset of the received signal, as well as monitoring of any changes in the CFO within the frame duration due to the changing Doppler shift.

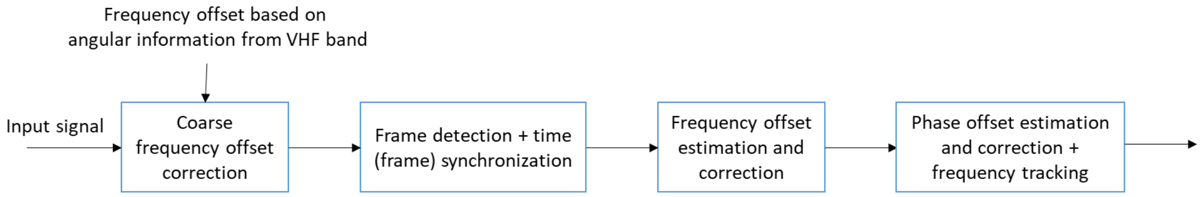
Frequency offset based on angular information from VHF band
Input signal
Coarse frequency offset correction
Frame detection + time (frame) synchronization
Frequency offset estimation and correction
Phase offset estimation and correction + frequency tracking

Fig. 2.16 Functional blocks of synchronization system.

Except for the initial coarse CFO correction, every functional block of synchronization requires transmission of a dedicated sequence – preamble – in order to perform the estimation. Therefore, the resulting radio frame, apart from the transmitted data, will contain two preambles: one for frame synchronization and one for frequency synchronization, as shown in Fig. 2.17. Moreover, the data part will be interleaved with short midambles that will be used for phase offset correction and frequency tracking.

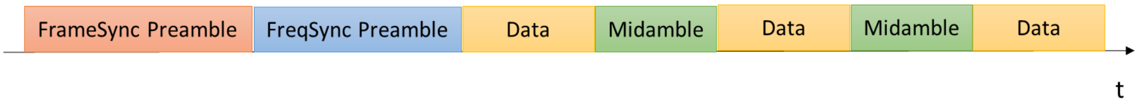

Fig. 2.17 Time-ordering of synchronization signals.

The following subsections describe the investigation and proposals of the functional blocks introduced above.

2.2.2. Frame (time) synchronization

The main purpose of the frame (time) synchronization block is to detect the start of a radio frame with an accuracy at a single sample level. As the considered system is supposed to operate at a very low SNR conditions, a robust method is required here. Due to the outstanding properties of the Zadoff-Chu sequences when performing their correlation in noisy environment, an extended version of such sequence has been chosen as the base for the frame synchronization preamble. The Zadoff-Chu sequence can be described with the following formula:

$$x_k(n) = exp\left[-j\frac{\pi k n(n+1)}{N_{ZC}}\right], \qquad (2.20)$$

where k is the sequence index and N_{ZC} is the sequence length. For the purpose of frame synchronization in the considered system a sequence with k=1 and N_{ZC}=107 has been chosen (although other variants can be also considered). This sequence has been extended to 128 symbols to account for the transmit and receive filtering impact as follows:

$$p(n) = \begin{cases} x(N_{ZC} - 12 + n) & for\; n \leq 10 \\ x(n-11) & for\; 10 < n \leq N_{ZC} + 10 \\ x(n - N_{ZC} - 10) & for\; n > N_{ZC} + 10 \end{cases} \qquad (2.20a)$$

In order to increase the detection capability at the receiver, such extended sequence have been repeated twice in the frame synchronization preamble.

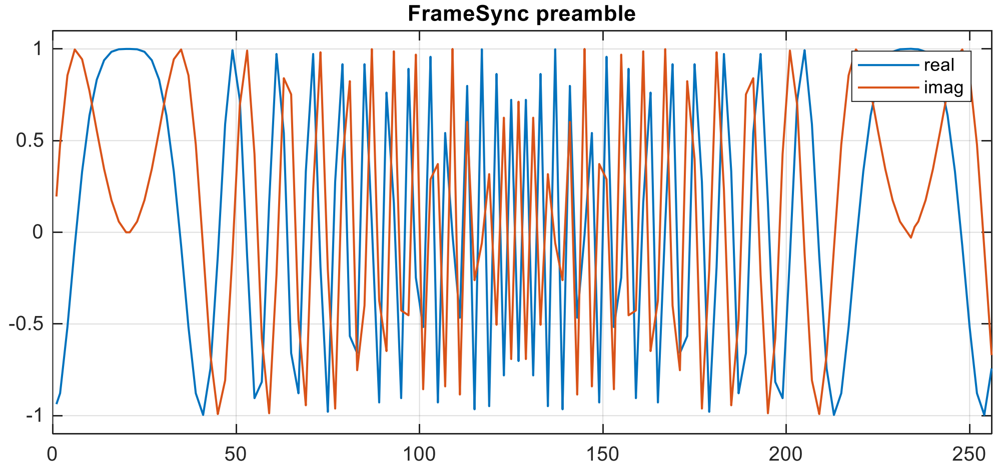

Fig. 2.18 Example of a half of frame synchronization preamble for Zadoff-Chu sequence with length N=107 (oversampling factor of 2).

The frame detection and synchronization block in the receiver consist of an input filter matched to the Zadoff-Chu sequence used to construct the preamble and the maximum detection block for the resulting output signal (synchronization metric), as shown in Fig. 2.19. In order to increase the frame detection capability, the preamble is constituted with two extended ZC sequences, each of length 128 symbols (thus, assuming oversampling with factor of 2, each sequence comprises 256 samples). Hence, at the receiver side the results of filtering of the two sequences are combined to reduce the impact of noise.

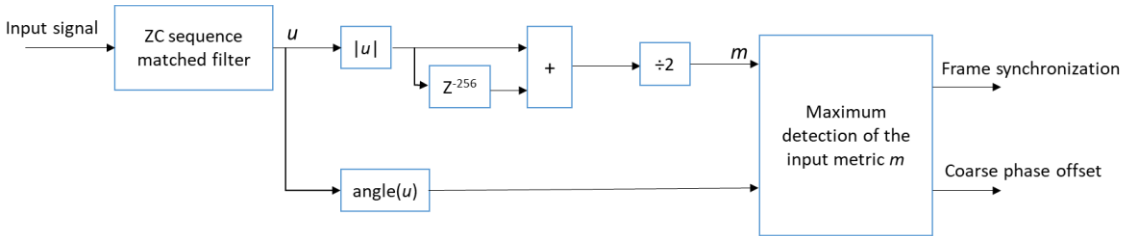

Fig. 2.19 Functional structure of the frame (time) synchronization block.

The operation of the block can be described as follows:

$$u(t) = r(t) \circledast h(t) \tag{2.21}$$

where ⊛ denotes the convolution, $r(t)$ is the received signal and $h(t)$ is the impulse response of the filter matched to the Zadoff-Chu sequence used to construct the preambule, i.e.:

$$h_k(n) = (x_k(N_{ZC} - n))^* \tag{2.22}$$

where $(.)^*$ denotes the complex conjugate. The metric $m(n)$ used for detection of the start of frame is calculated as the average of the modulus of the output signal from the matched filtering block with its version delayed by 256 samples:

$$m(n) = \frac{|u(n)|+|u(n-256)|}{2} \tag{2.23}$$

Examples of the time evolution of the metric $m(n)$ are shown in Fig. 2.20 and Fig. 2.21.

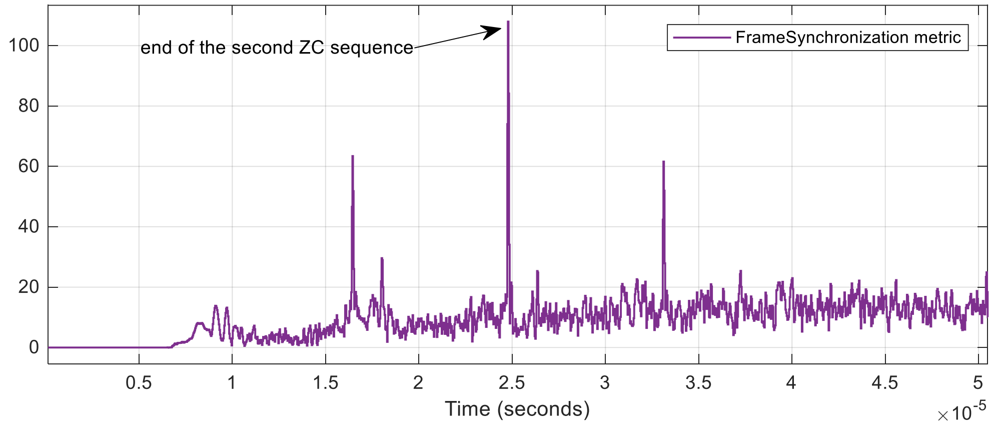

Fig. 2.20 Example of an observed frame synchronization metric evolution for SNR=0 dB

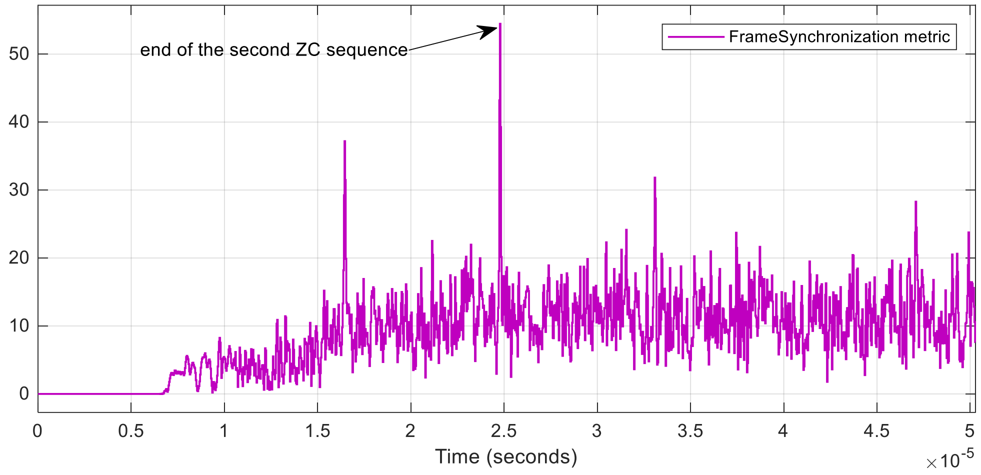

Fig. 2.21 Example of an observed frame (time) synchronization metric evolution for SNR=-10 dB

Additionally to frame (time) synchronization, the initial phase offset can be estimated as the angle of the sample of signal *u*(*n*) for which the maximum of metric *m*(*n*) has been detected. This allows for the initial compensation of the phase offset prior to frequency synchronization.

2.2.3. Frequency offset estimation

In a mobile or satellite communications system, there is usually a mismatch at the receiver between the reference frequency and the received signal carrier, that must be removed and the actual carrier frequency recovered in order to demodulate the signal. This mismatch can be caused by the Doppler shift, but also due to the mismatch between local oscillators used in the transmitter and the receiver. As the significant part of the CFO caused by Doppler shift can be removed with the use of coarse CFO correction block (based on orbital parameters of the satellite), the purpose of the fine CFO estimation algorithm is to detect the mismatch caused by differences in oscillators and the eventual remaining residual CFO due to Doppler shift. Such estimation can be realized using a variety of methods described in the literature, however, due to the expected very difficult reception conditions (low SNR) a robust and accurate method needs to be found.

2.2.3.1. The Cramer-Rao lower bound

The Cramer-Rao lower Bound (CRB) is the most important and most widely used representation of the theoretical limit for the estimation of the frequency of a sinusoid, represented as the bound on the variance of the estimates. According to the derivation presented in [11], the CRB for estimation of the frequency of a sinusoid with an unknown phase can be formulated as:

$$CRB = \frac{6F_S^2}{(2\pi)^2 \rho N(N^2-1)}, \qquad (2.24)$$

where F_s is the sampling frequency, ρ is the SNR (in linear scale) and N is the number of samples used for frequency estimation (in our case the FreqSync preamble length).

Based on the formula (2.24) for CRB, the limitations on the performance (expected minimum variance of estimates) of CFO estimation for different preamble lengths have been calculated. These are illustrated in Fig. 2.22 vs. SNR.

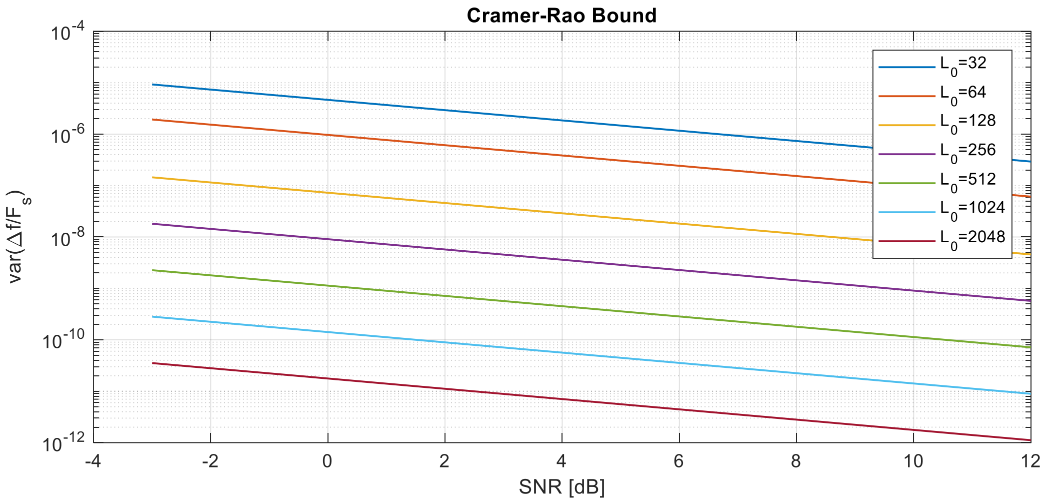

Fig. 2.22 Normalized variance (related to the sampling frequency) of the CFO estimation vs. SNR for different synchronization preamble lengths – Cramer-Rao lower bound

The variance values obtained with CRB indicate the best performance that can be achieved given the preamble length. E.g. with preamble of 256 samples and SNR=0 dB, the normalized (with respect to the sampling frequency) variance of 10^{-8} is expected. Hence, assuming e.g. F_s = 30 MSps, the expected estimation error will be around 3 kHz. Such outcome indicates, that sufficient preamble length is needed to achieve high performance of CFO estimation. Moreover, the length of used preamble will depend on the sampling frequency used, thus, will be dependent on the configured system bandwidth.

2.2.3.2. Frequency synchronization preamble

With the considered data-aided feedforward synchronization method, a properly designed and sufficiently long frequency synchronization preamble is required. For the purpose of fine CFO estimation two different preamble structures were considered, depending on the selected synchronization method:

- Complex sinusoidal signal that can be described with the following formula:

$$x(n) = exp\left[-j\frac{\pi n}{4}\right], n = 0,1, \dots, L_0 - 1, \qquad (2.25)$$

The complex sinusoid preamble was used with most of the considered CFO estimation methods due to its robustness to any changes caused e.g., by transmit or receive filtering. An example of the real and imaginary parts of the complex sinusoid preamble of length L0=128 is given in Fig. 2.23 (there are more than 250 samples as the signal is upsampled with factor of 2).

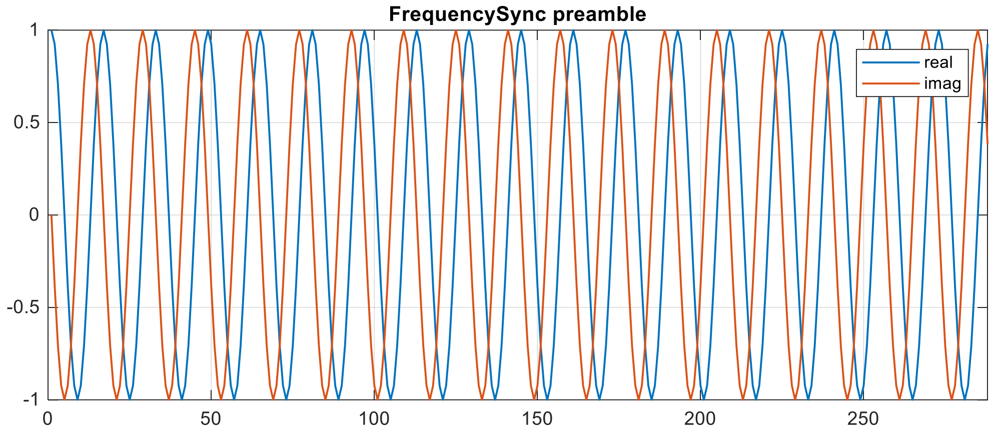

Fig. 2.23 Example of a complex sinusoid used as the frequency synchronization preamble of length L0=128 (oversampling factor of 2).

- Preamble constituted of multiple Zadoff-Chu sequences, that can be used for tracking of phase offset within the duration of preamble. The base Zadoff-Chu sequence used for creating the preamble was constructed using (2.20), with *k*=1 and N_{ZC}=61. Such base sequence was extended to get the base sequence of length L_s=64 (and upsampled with factor of 2). The full preamble of length L_0 comprised *k* such base sequences, with *k*= L_0/L_s. An example of the Zadoff-Chu-based preamble of length L_0 = 128 is shown in Fig. 2.24.

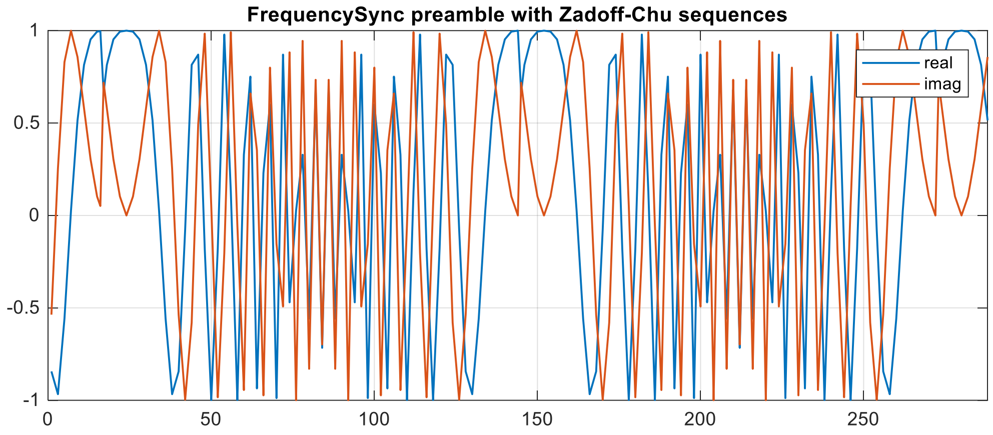

Fig. 2.24 Example of a frequency synchronization preamble of length L0=128 constructed with repeating Zadoff-Chu sequences (oversampling factor of 2).

For both the complex sinusoid preamble and the Zadoff-Chu-based preamble, they were extended with L_F samples on both the beginning and the end of the preamble, where L_F is the length of the transmit and receive filter impulse response, to mitigate the impact of inter-symbol interference introduced by filtering. Hence, the total length of the used FreqSync preamble was $L_0 + 2*L_F$ (for the considered system configuration L_F = 16 was assumed).

2.2.3.3. Considered frequency offset estimation methods

Many different CFO estimation methods have been proposed in the literature for wireless systems, that can be categorized into the following groups: phase-based frequency estimation, correlation-based frequency estimation, filtering-based estimation (using e.g. ARMA modeling or Kalman filters) or the frequency-domain frequency estimation [11]. Among the considered solutions only filtering-based methods were excluded from investigation due to their computational requirements and the considered algorithms are presented below.

- Phase-based frequency estimation

With phase-based frequency estimation the main focus is on estimating the change in phase of signal between consecutive symbols of the preamble. The phase of the received signal with CFO of Δf is given as:

$$\angle x(n) = 2\pi n \frac{\Delta f}{F_s} + \theta + u(n), \qquad (2.26)$$

where F_s is the sampling frequency, θ is the constant phase offset and $u(n)$ is the noise component. Assuming that the current phase $\angle x(n)$ can be calculated reliably (low impact of variation due to $u(n)$), one can estimate the CFO based on phase offset obtained for two samples offset by d as follows:

$$\angle x(n) - \angle x(n-d) = 2\pi d \frac{\Delta f}{F_s} + u(n) - u(n-d), \qquad (2.27)$$

Hence, assuming the noise components cancel out, the CFO can be estimated as:

$$\widehat{\Delta f} = \frac{\angle x(n) - \angle x(n-d)}{2\pi d} F_s, \qquad (2.28)$$

Certainly the CFO estimate will be erroneous, as the noise components will usually impact the estimate. However, when the phase estimates can be reliably obtained (with low error due to noise), such estimator may be sufficient. This is the case when e.g. Zadoff-Chu sequence is used for phase estimation. With the use of a filter matched to the ZC sequence, one can obtain reliable phase estimates for samples spaced by N_{ZC} when at least two such sequences are repeated. Hence for the considered system the following methods exploiting such properties of ZC sequence were employed:

- Estimation based on the two ZC sequences in the frame synchronization preamble – denoted in the results section as *Phase Diff* (here the length of preamble is constant and equal 512 samples)
- Estimation based on the FreqSync preamble using repeating ZC sequences of length L_s=64 (128 samples with upsampling of 2). At least two such sequences were required, thus the minimum length of such preamble was 256 samples (128 symbols). Such approach is denoted in the results section as *Zadoff-Chu sequence*.

- Correlation-based frequency estimation:

Another approach to CFO estimation is to use the autocorrelation-based estimation to detect the frequency of a sinusoid resulting from the existing CFO. Such a sinusoid can be obtained by demodulating the known received signal (hence, can be used as data-aided estimation) when multiplying the received sequence with conjugated reference sequence. Then the frequency of the resulting sinusoid due to CFO can be e.g. obtained with the Mengali-Morelli (MM) estimator [12] as follows:

$$\widehat{\Delta f} = \frac{1}{2\pi T} \sum_{m=1}^{N} w(m) \times [\angle R(m) - \angle R(m-1)]_{2\pi}, \qquad (2.29)$$

where *T* is the sampling interval, $w(m)$ is a smoothing function (given in [12]) and $R(m)$ is the autocorrelation of the sampled sinusoid *z*(*n*) due to CFO, calculated as:

$$R(m) = \frac{1}{L_0 - m} \sum_{n=m}^{L_0 - 1} z(n) z^*(n-m)\,, 1 \leq m \leq D, \tag{2.30}$$

D is the design parameter not greater than $L_0/2$, where L_0 is the preamble length.

Unfortunately, such autocorrelation-based methods exploiting the demodulated received sequence turned out to be unreliable in low SNR conditions, resulting in significant CFO estimation errors. Therefore, these were not considered for the final evaluation of considered solutions.

Another interesting approach based on autocorrelation of the received signal is the Schmidl-Cox algorithm proposed originally for multicarrier systems [13]. It makes use of a preamble comprising two identical halves (hence, can be applied both with the complex sinusoid preamble or the ZC-based preamble) by correlating them. Assuming the correlation of the two received identical sequences is calculated as:

$$P = \sum_{m=0}^{L_s - 1} x^*(n) x(n + L_s), \tag{2.31}$$

where L_s is the length of a single sequence, the CFO estimate can be calculated as:

$$\widehat{\Delta f} = \frac{1}{\pi T} \angle P, \tag{2.32}$$

The advantage of the Schmidl-Cox method is its simplicity of implementation, as it can be realized using an iterative formula.

- Frequency-domain frequency estimation:

The Maximum Likelihood estimation of the CFO can be realized as a periodogram maximizer [11]. As the calculation of the full periodogram (including averaging) might require too many computations and can result in a significant delay, it can be simplified with the use of a single Discrete Fourier Transform (DFT) or its efficient implementation in the form of Fast Fourier Transform (FFT). Unlike the phase and correlator-based estimators, the FFT based algorithms exhibit excellent performance at low SNR. The CFO estimate can be found as the peak of the DFT of the sinusoid resulting from the presence of CFO. Therefore, the procedure here is similar as in case of the correlation-based methods – first the received preamble sequence needs to be demodulated by multiplying the received symbols with conjugated reference sequence.

The drawback of the FFT-based methods is that they are accurate with the resolution depending on the FFT size (N_{FFT}) [11]:

$$\left|\Delta f - \widehat{\Delta f}\right| = \left|\Delta f - m \frac{F_s}{N_{FFT}}\right| \leq \frac{F_s}{2N_{FFT}}, \tag{2.33}$$

where F_s is the sampling frequency and *m* is the selected FFT bin index. Clearly, with high F_s a very large N_{FFT} would be required to obtain accurate CFO estimates. Hence, as an alternative, interpolation methods can be applied. An example of a simple interpolation method based on three samples around the FFT maximum is given in [14]. Such a solution, employing FFT of size 2048 and interpolation based on the approach proposed in [14] is evaluated in the results section with the name *FFT*.

Alternatively, one can consider a more sophisticates interpolation approach that is based on iterative processing of the so-called Fractional Fourier Coefficients (FFC), as proposed in [11]. Having found the index *m* of the maximum bin of DFT, one can use it as an initial estimate of the CFO to calculate the fine estimate using two FFCs. The fine estimate of CFO can be calculated as:

$$\widehat{\Delta f} = \left(m + \hat{\delta}\right) \frac{F_s}{N_{DFT}}, \tag{2.34}$$

where:

$$\hat{\delta} = \frac{1}{2} Re \left\{ \frac{X_p + X_{-p}}{X_p - X_{-p}} \right\}, \tag{2.35}$$

and the FFC is calculated as:

$$X_p = \sum_{k=1}^{N_{DFT}-1} x(k) e^{-j2\pi \frac{k(m+p)}{N_{DFT}}} \tag{2.36}$$

Such approach can be applied iteratively, with $(m + \hat{\delta}_{i-1})$ applied as the initial CFO estimate (instead of only *m* in (2.35)) for calculation of the FFCs in *i*-th iteration.

The approach based on iterative interpolation using Fractional Fourier coefficients have been considered with two offset values *p*: 0.5 and 0.125. Moreover, as for most cases the initial CFO estimate from FFT block is expected to correspond to bin index of 0, the following configurations are considered in the final evaluation:

- *Fractional Interpolation* 0.5, assuming *m*=0 (no FFT used) and *p*=0.5.
- *Fractional Interpolation* 0.125, assuming *m*=0 (no FFT used) and *p*=0.125.
- *FFT+Fractional Interpolation*, with the use of FFT to obtain *m* and *p*=0.5.

2.2.3.4. Simulation results

In this section the simulation results of comparison of selected CFO estimation methods are presented. The considered performance measure of interest was the root mean squared error (RMSE) of the CFO estimate calculated based on 10 independent simulation runs. The results were obtained assuming different bandwidth configuration of the considered system, different SNR values (-5, 0 and 5 dB) and various preamble length. In all cases except for the *Phase Diff* method and *Zadoff-Chu sequence* method the complex sinusoid preamble was used. For *Phase Diff* method the FrameSync preamble was used (two extended ZC sequences – 512 samples in total). With *Zadoff-Chu sequence* method a repeating ZC-based preamble was used.

Figure 2.25 presents the RMSE observed for system with bandwidth of 1.25 MHz, CFO of 100 Hz, and preambles of length 128 and 256 symbols (256 and 512 samples). Clearly one can notice the significant estimation error at low SNR (-5 dB) with the shorter preamble. The remaining CFO of over 500 Hz in the case of the *Schmidl-Cox* algorithm and the *Zadoff-Chu sequence* method disqualifies these solutions. On the other hand, one can notice a very good performance with the *Phase Diff* method, as it uses a relatively long preamble. Similar performance is observed for the frequency-domain frequency estimation methods (relying of Fourier coefficients), however, only for the longer preamble of 256 symbols.

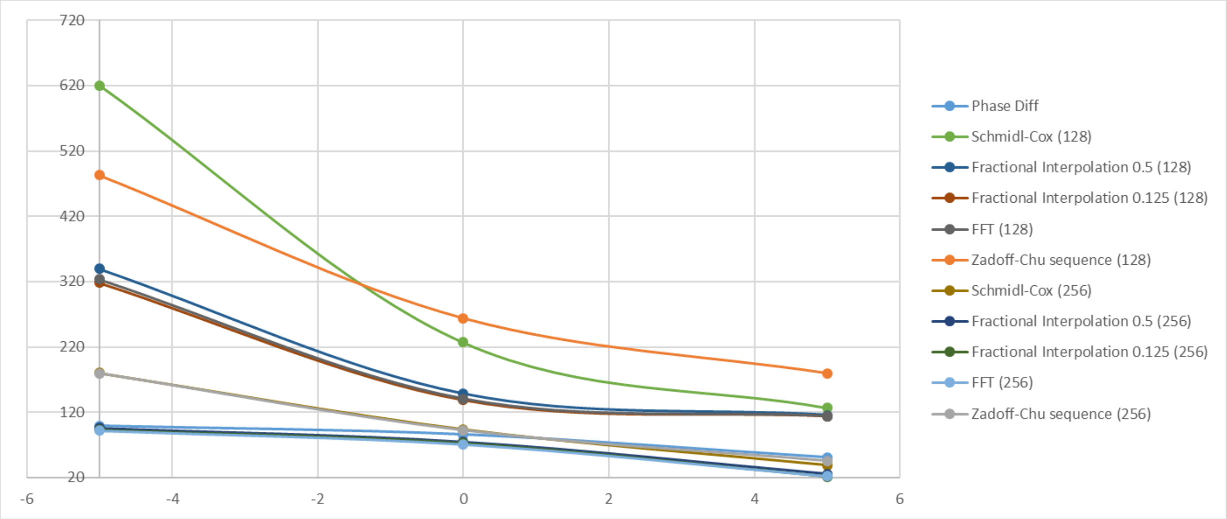

Fig. 2.25 Comparison of the RMSE of different CFO estimation methods for Δf=100 Hz, bandwidth of 1.25 MHz and preamble lengths of 128 and 256 symbols.

Similar observations can be made with the system bandwidth of 5 MHz, CFO of 100 Hz, with the results shown for preamble lengths of 256 and 512 in Fig. 2.26 and Fig. 2.27, respectively. One can notice, that the *Phase Diff* method performs worse than the methods relying on Fourier coefficients, as it has no more the advantage of a longer preamble sequence. Hence, it was omitted in Fig. 2.27. One can also notice, that in case of 5 MHz bandwidth the preamble using 256 symbols might be not sufficient to achieve good CFO estimation performance (CFO error of over 400 Hz remaining in low SNR region). Much better results are observed with preamble constituted from 512 symbols.

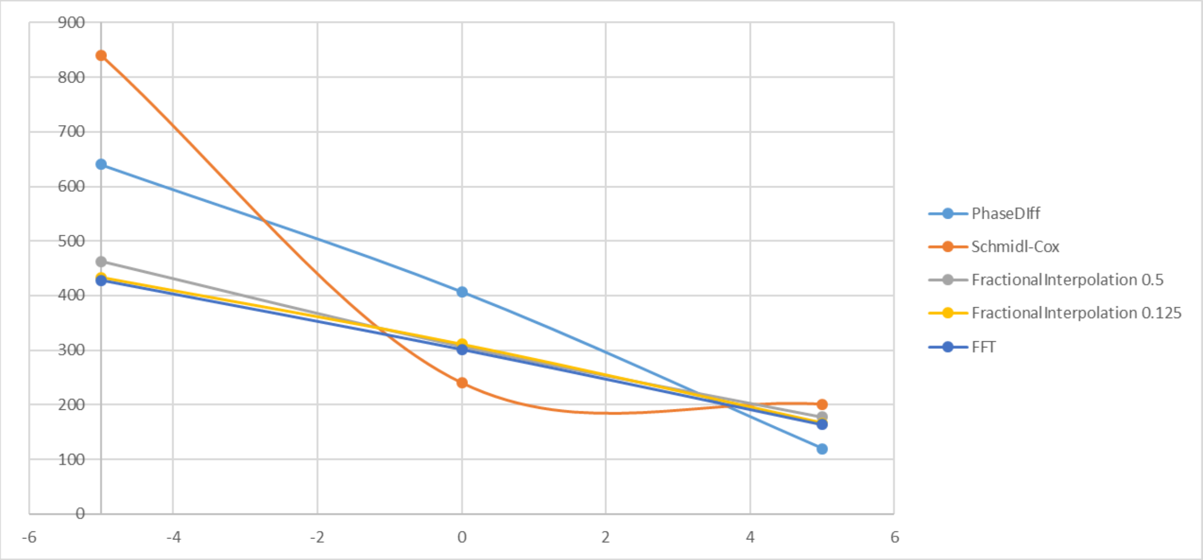

Fig. 2.26 Comparison of the RMSE of different CFO estimation methods for Δf=100 Hz, bandwidth of 5 MHz and preamble length of 256 symbols.

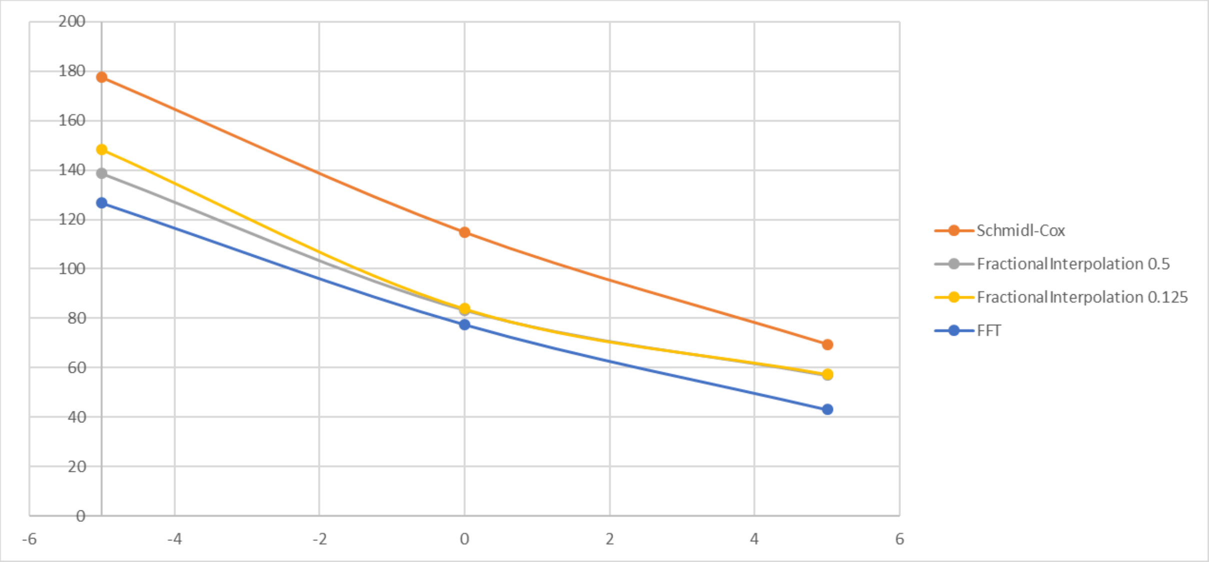

Fig. 2.27 Comparison of the RMSE of different CFO estimation methods for Δf=100 Hz, bandwidth of 5 MHz and preamble length of 512 symbols.

Analyzing the results obtained with the system bandwidth of 10 MHz and CFO of 100 Hz, with results shown in Fig. 2.28 and Fig. 2.29 for sequence length of 512 and 1024 symbols, respectively, one can notice that further increase in preamble length is required. Again, clearly the best performance is achieved with methods relying on Fourier coefficients, however, it can be noted that with preamble length of 1024, the *FFT* method employing interpolation from [14] performs much worse. This is caused by reaching the imposed limit on FFT size, which is equal to the number of preamble samples, thus resulting in low interpolation accuracy. Such phenomenon is not observed with the iterative methods relying on FFCs, that are the best performers here.

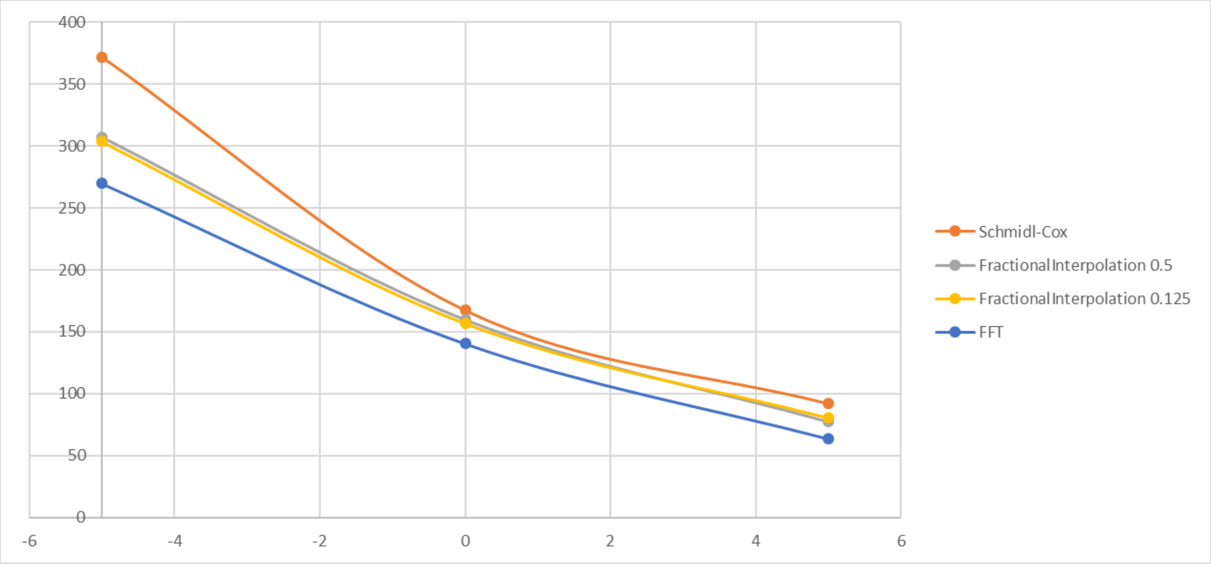

Fig. 2.28 Comparison of the RMSE of different CFO estimation methods for Δf=100 Hz, bandwidth of 10 MHz and preamble length of 512 symbols.

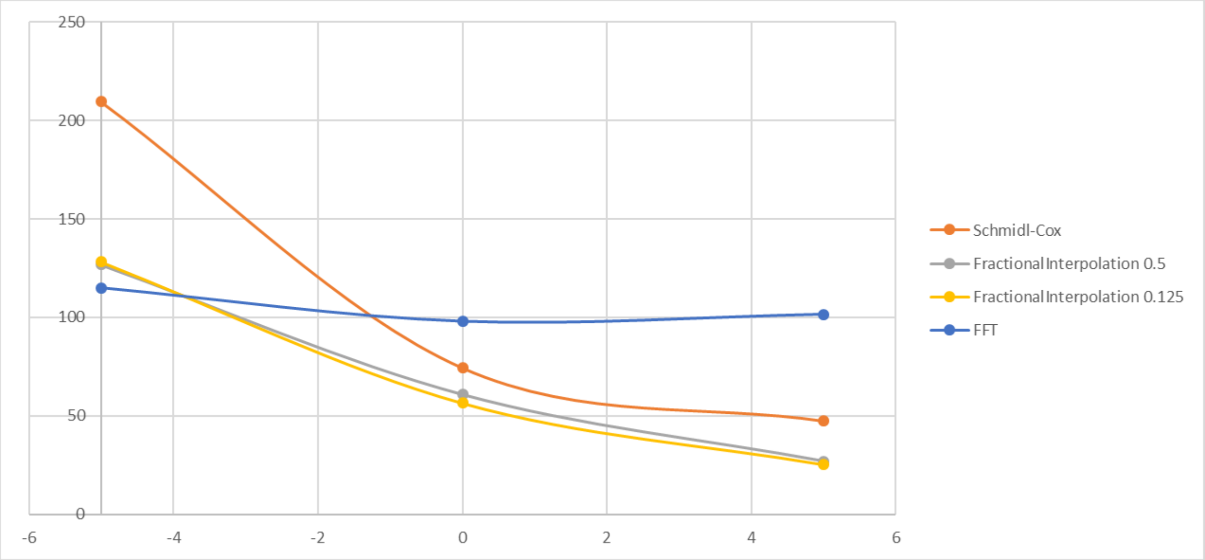

Fig. 2.29 Comparison of the RMSE of different CFO estimation methods for Δf=100 Hz, bandwidth of 10 MHz and preamble length of 1024 symbols.

Figure 2.20 and Fig. 2.31 present the results obtained with bandwidth of 20 MHz, CFO of 100 Hz and two preamble lengths: 512 and 1024 symbols. The conclusions are similar as in case of 10 MHz configuration, however, the error observed in this case is slightly higher. Such a situation, however, is acceptable, as the impact of higher CFO in this case is lower due to faster sampling.

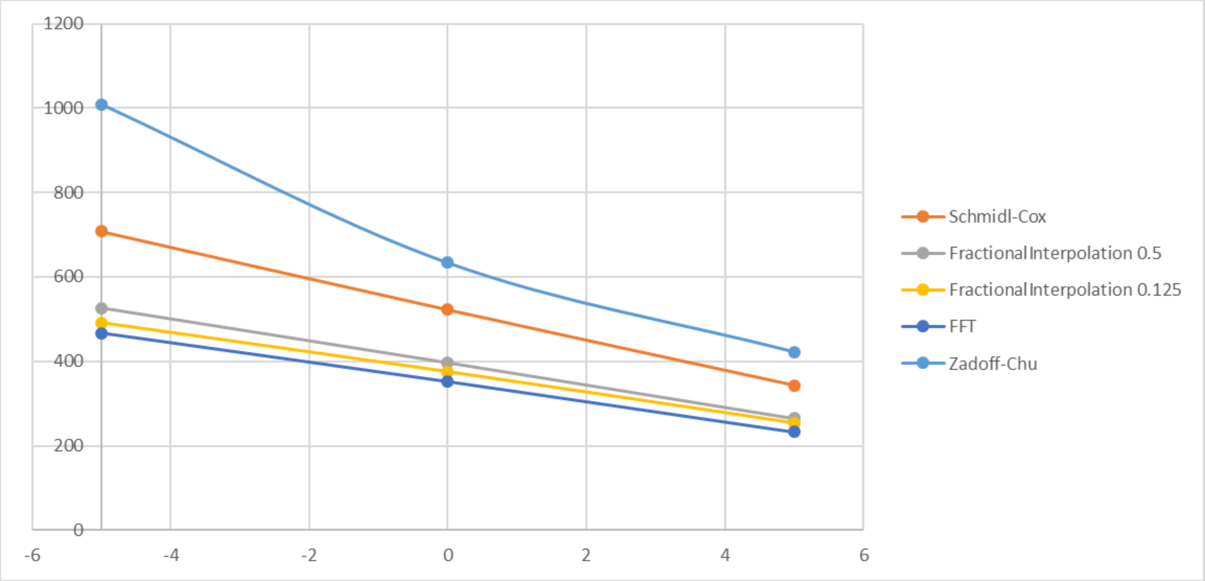

Fig. 2.30 Comparison of the RMSE of different CFO estimation methods for Δf=100 Hz, bandwidth of 20 MHz and preamble length of 512 symbols.

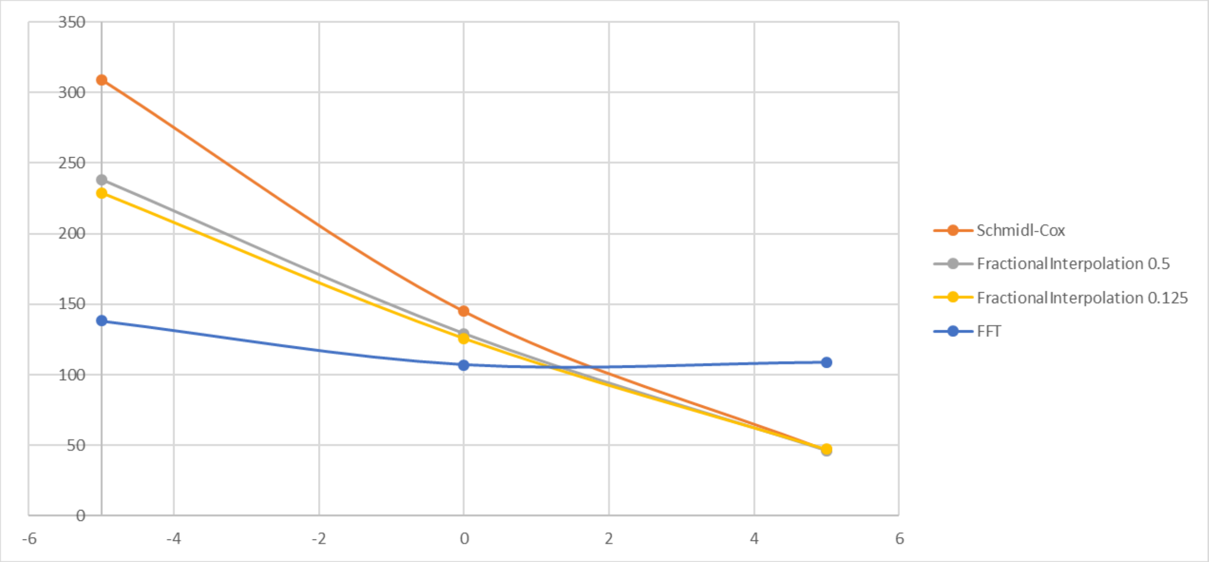

Fig. 2.31 Comparison of the RMSE of different CFO estimation methods for Δf=100 Hz, bandwidth of 20 MHz and preamble length of 1024 symbols.

In order to verify the robustness of the considered methods to high CFO, investigations on their performance with CFO=12100 Hz were conducted for bandwidths of 1.25 MHz and 20 MHz, shown in Fig. 2.32 and Fig. 2.33, respectively. One can clearly notice, that several methods, including the Phase Diff, Schmidl-Cox and also Fractional Interpolation without additional FFT input (initial CFO estimate corresponding to the maximum bin) experience very bad performance, no matter what is the preamble length. This is the result of a phase rotation of the received signal due to CFO by at least 2π, so the correct CFO cannot be estimated in this case. The only methods capable of finding the CFO estimate close to the correct one are the FFT-based. Hence, clearly there is a need to include also the initial FFT-based estimation for the considered scheme in case a large CFO occurs. One can clearly note, that the best approach overall is the iterative Fractional Interpolation aided with FFT.

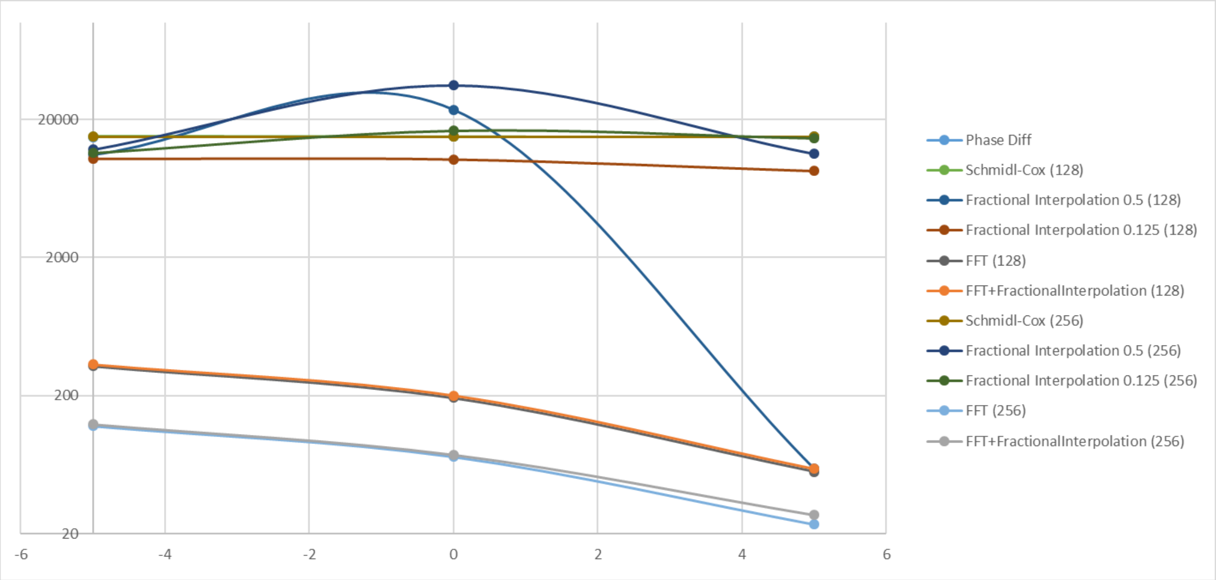

Fig. 2.32 Comparison of the RMSE of different CFO estimation methods for Δf=12100 Hz, bandwidth of 1.25 MHz and preamble lengths of 128 and 256 symbols.

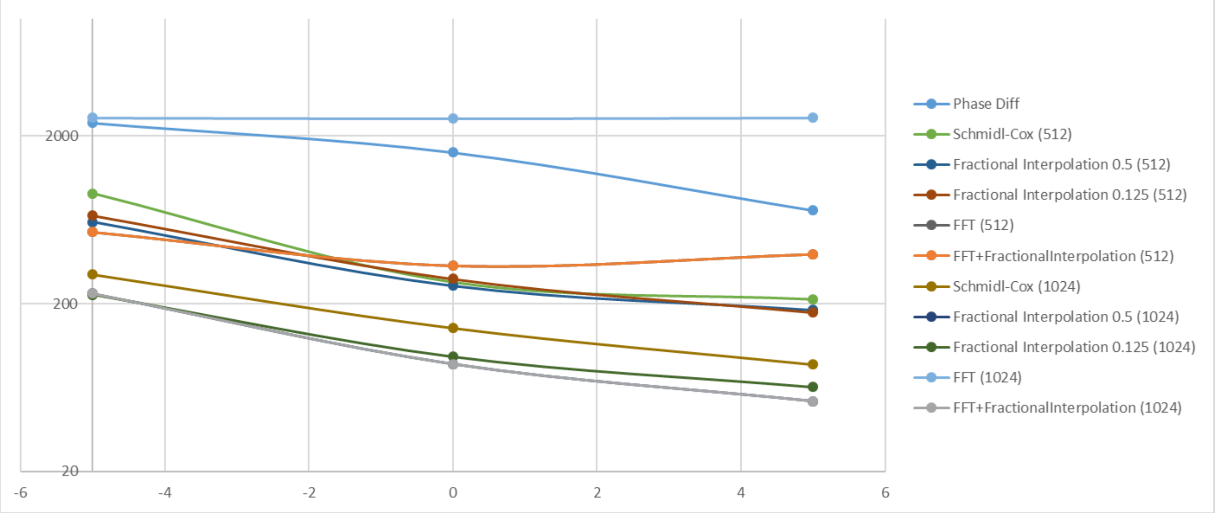

Fig. 2.33 Comparison of the RMSE of different CFO estimation methods for Δf=12100 Hz, bandwidth of 20 MHz and preamble lengths of 512 and 1024 symbols.

2.2.3.5. Proposed solution for carrier frequency offset estimation

Based on the analysis carried out in the simulation section, a proposal of the frequency synchronization method can be formulated. Clearly the best performance can be acheived with the iterative interpolation based on Fractional Fourier coefficients, that, however, requires provisioning of the inital coarse CFO estimate from an FFT block. The schematic of the proposed CFO estimation block is given in Fig. 2.34.

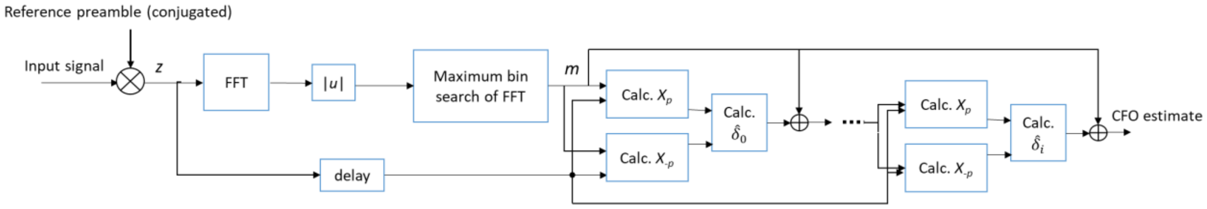

Fig. 2.34 Functional block schematic of the proposed CFO estimator.

Moreover, one can conclude, that a preamble based on a complex sinusoid should be used due to its simplicity of processing and robustness. The exact length of FreqSync preamble used is dependent on the considered system bandwidth, and should range from 256 symbols for 1.25 MHz to 1024 symbols for 10MHz or 20 MHz configurations.

2.2.4. Phase synchronization and frequency tracking

The last element of the synchronization system at the receiver is the phase synchronization and frequency tracking block. For the purpose of the phase offset estimation, the FrameSync preamble can be used initially, as described above. However, due to the remaining small CFO error and the changes in Doppler shift as the satellite changes its position, the phase offset might change within a single long frame. Thus, dedicated short midambles are included in the radio frame, interleaved with data blocks, in order to continuously monitor phase offset changes, and, thus, applying corrections to the CFO estimates. Such a mechanism can be realized in a similar way to processing of the FrameSync preamble, with Zadoff-Chu sequence, described by (2.20), used as the midambles. As the midamble is expected to be short, comparing to the data block (the midamble of length of 64 symbols – 128 samples – was used in the evaluation), an extended ZC sequence, with *k*=1 and N_{ZC}=47 was used, with 8 and 9 samples appended to the front and to the tail, respectively, similarly to the approach taken in (2.20a).

At the receiver side, the phase offset tracking is performed using a filter matched to the ZC sequence used in the midamble. The difference in phase offset estimates for subsequent midambles is then calculated to find the estimate of remaining CFO according to (2.28). In order to mitigate the impact of noise components, the consecutive estimates are averaged using simple exponential smoothing formula as follows:

$$\widehat{CFO}_i = \alpha\widehat{\Delta f}_i + (1-\alpha)\widehat{CFO}_{i-1}, \tag{2.37}$$

where α is an arbitrarily selected constant. Figure 2.35 shows an example of tracking of CFO with the proposed method.

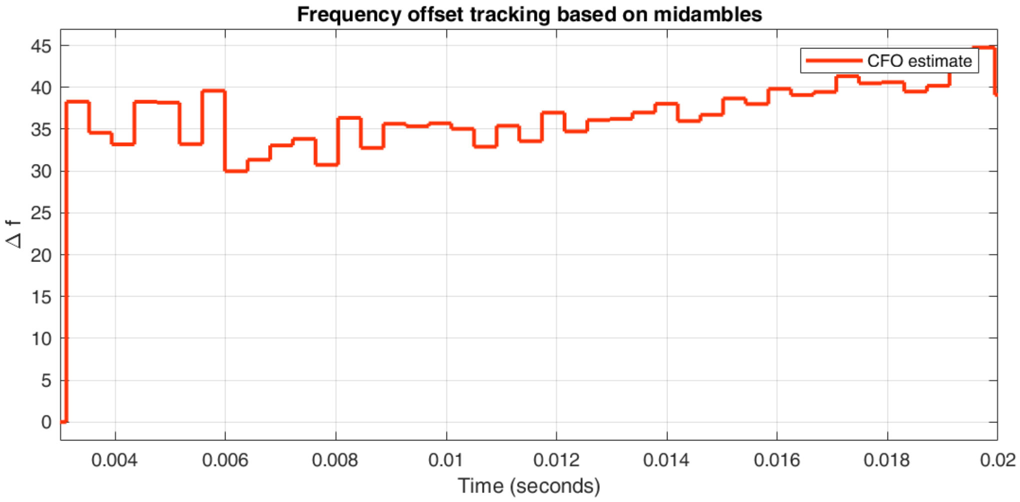

Fig. 2.35 Example of tracking of CFO for system bandwidth of 1.25 MHz, CFO change of 1 kHz/s, SNR=5 dB and $\alpha = 0.1$.

2.3. Channel coding

Selection of the proper channel coding scheme is of the highest importance in the case of any transmission system. Channel coding needs to fulfill specific requirements like allowable complexity (both encoding and decoding processes), achievable BER/BLER performance, decoding latency, and flexibility (w. r. t. variable code lengths and code rates, etc.).

In case of the satellite communication considered in this report, the following codes have been found as potential candidates for channel coding since they allow to obtain best performance (decoding quality) for long code words:

- Turbo codes - Turbo codes (TC) are usually praised for achieving small error probabilities using SNRs just slightly above the capacity limit when the code block is large (1024 bits or more). However, it has been shown that turbo codes can approach theoretical bounds almost uniformly over a wide range of code rates and block sizes [15]. The most common approach to TC is the use of the parameters determined in 3GPP LTE standards. They include the constituent recursive systematic convolutional code polynomials, the interleaver algorithm, and Rate-Matching. The max-log-MAP algorithm with eight iterations is usually applied in the decoder. In most cases, this scheme is considered a reference for selecting the parameters of all the remaining coding schemes to have similar algorithmic complexity.
- LDPC codes – LDPC codes are block codes that, similar to turbo codes, can achieve performance close to the Shannon limit for long codewords. In the synthesis of an LDPC code, different approaches can be used, e.g., finding the shortest girth on the Tanner graph describing the code properties determines the code performance. The so-called Progressive Edge Growth (PEG) algorithm described in [16] is an example of the parity check matrix synthesis algorithm, which results in possibly high girth value. The most considered decoding algorithms are suboptimal min-sum algorithm (MSA), its modification called Offset-MSA (OMSA) and optimal sum-product algorithm (SPA) [17].
- Polar codes - Polar codes are one of the newest channel coding schemes introduced in [18]. The idea of polar codes is to create from N independent copies of a given channel, N different channels $W(i)$, $1 \leq i \leq N$ through a linear transformation. As N grows large, the channels seen by individual bits start polarizing to either noiseless or pure-noise channels. The fraction of channels becoming noiseless is close to the mutual information $I(W)$, while the fraction of channels becoming pure-noise is close to $1 - I(W)$. Polar codes use the noiseless channels for transmitting information while the pure-noise channel transmits fixed (known by the transmitter and the receiver) symbols. The performance of classic polar codes with successive cancellation decoding proposed in [18] is somewhat disappointing. However, successive cancelation list (SCL) decoding, introduced in [19], has been shown to substantially improve the performance of polar codes, which is why they are strong candidate codes for satellite communication.

To obtain a fair performance comparison of the considered candidate codes, an algorithmic complexity associated with the decoding process, as presented in [20], can be used. As shown in the aforementioned paper, all the proposed coding schemes offer similar decoding performance (with a slight indication for turbo codes). Thus the selection of the channel coding scheme is actually motivated by the encoding process complexity, which is the smallest for turbo codes.

2.3.1. Modulation and Code scheme selection

A set of different modulation and coding schemes (MCSs) need to be chosen to improve system flexibility. These are selected depending on the satellite-to-Earth link quality. Since at this level of the project there was no agreement on modulation and codeword length used, the initial set of MCSs has been obtained for BPSK, QPSK, and 8-PSK modulation and the message length of 4096 bits (this length is a good approximation of the longest codes that can be handled by LTE TC code).

In Figs 2.36-2.38, a BER performance for the BPSK, QPSK, and 8-PSK have been shown, respectively. These curves were further used to assess the achievable link quality.

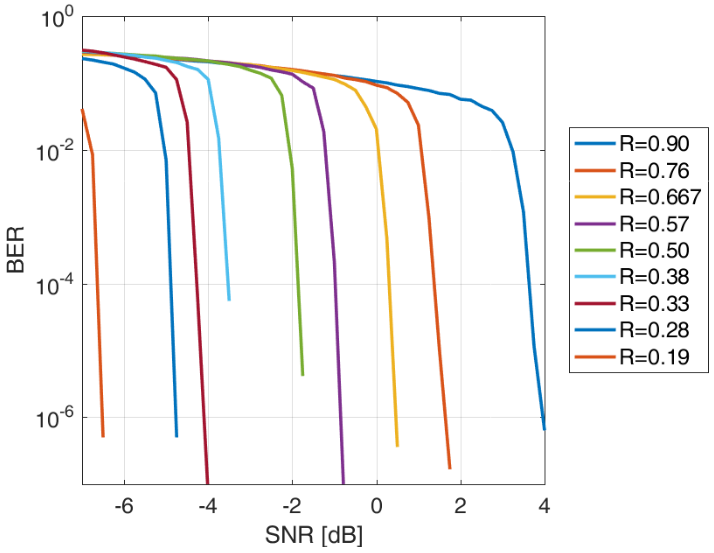

Fig. 2.36 BER performance for different coding rates for BPSK modulation

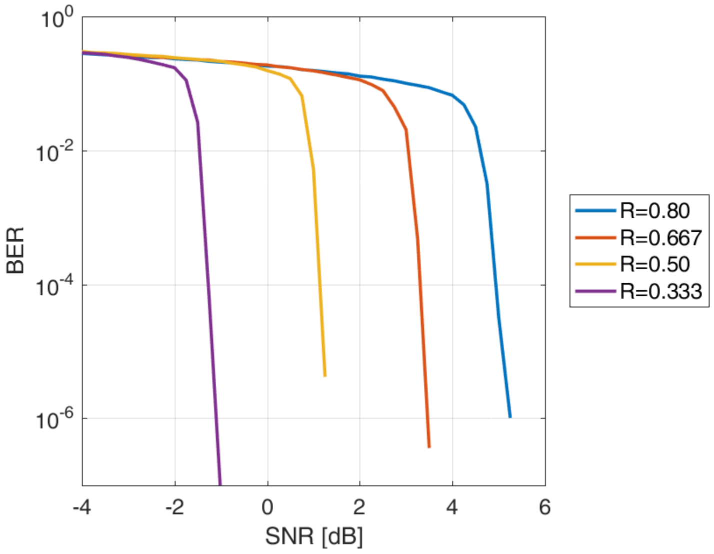

Fig. 2.37 BER performance for different coding rates for QPSK modulation

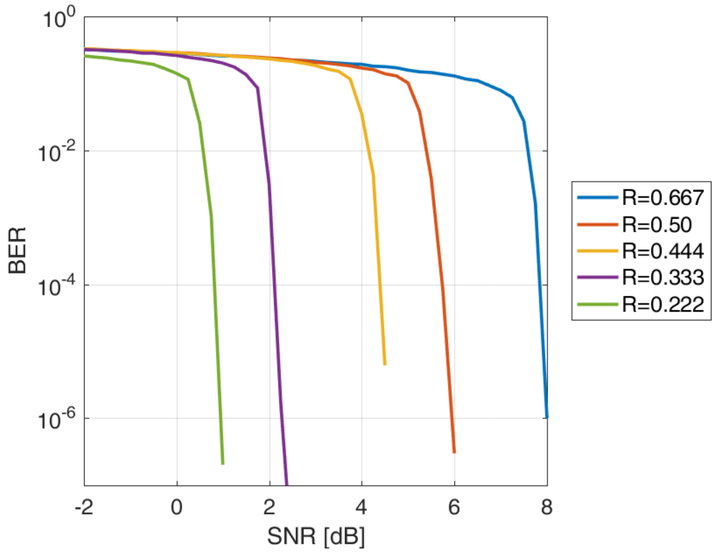

Fig. 2.38 BER performance for different coding rates for 8-PSK modulation

2.3.2. Throughput estimation

Since the system is designed to operate in four different bandwidths, i.e., 20 MHz, 10 MHz, 5 MHz, and 1.25 MHz, the throughput estimation needs to be performed separately for each bandwidth. Moreover, data presented in Fig. 1.1 are used as a benchmark for link SNR values for different bandwidths. Throughput estimation will be performed with the use of coding schemes developed in Section 2.3.1.

In Tab. 2.2, an estimation of throughput for a 20 MHz channel has been presented.

Tab. 2.2. Throughput estimates for 20 MHz channel

Modulation	**R**	**Throughput [Mbps]**	**Required SNR [dB]**
BPSK (GMSK)	R = 0.76	20.27	1.75
O-QPSK	R = 0.50	15.38	1.3
BPSK(GMSK)	R = 0.57	15.20	-0.75
8-PSK	R = 0.225	10.38	1
O-QPSK	R = 0.333	10.24	-1
BPSK(GMSK)	R = 0.38	10.13	-3.25
BPSK(GMSK)	R = 0.285	7.60	-4.75
BPSK(GMSK)	R = 0.19	5.07	-6.5

As presented in the above table, a set of different throughput values can be obtained in the system. This set of throughputs need to be also verified against the required SNR value for the 20 MHz channel. As seen in Fig. 1.1, the SNR range for this channel is from -7 to 4 dB. However, the introduction of an additional margin of 3 dB (e.g., for lower antenna gains, etc.) drives the maximum allowable SNR to 1 dB only. Based on this, all the combinations that require at least 1 dB SNR for proper operation need to be neglected. Based on this, a set of throughputs for 20 MHz presented in Tab. 2.3 have been chosen (additionally, in the project at this point, GMSK modulation has been assumed).

Tab. 2.3. Final throughput values for 20 MHz channel

Modulation	**R**	**Throughput [Mbps]**	**Required SNR [dB]**
BPSK(GMSK)	R = 0.57	15.20	-0.75
BPSK(GMSK)	R =0.38	10.13	-3.25
BPSK(GMSK)	R = 0.285	7.60	-4.75
BPSK(GMSK)	R = 0.19	5.07	-6.25

A similar investigation has been performed for the remaining channels' width. Results are presented in Tabs 2.4 and 2.5 for 10 MHz channel, in Tabs 2.6 and 2.7 for 5 MHz channel and in Tabs 2.8 for 1.25 MHz channel.

Tab. 2.4. Throughput estimates for 10 MHz channel

Modulation	**R**	**Throughput [Mbps]**	**Required SNR [dB]**
BPSK (GMSK)	R = 0.90	12.00	4
8-PSK	R = 0.45	10.38	4.6
O-QPSK	R = 0.667	10.26	3.5
BPSK(GMSK)	R = 0.76	10.13	1.75
8-PSK	R = 0.333	7.68	2.3
O-QPSK	R = 0.50	7.69	1.3
BPSK(GMSK)	R = 0.57	7.60	-0.75
8-PSK	R = 0.225	5.19	1
O-QPSK	R = 0.333	5.12	-1
BPSK(GMSK)	R = 0.38	5.07	-3.25
BPSK(GMSK)	R = 0.285	3.80	-4.75
BPSK(GMSK)	R = 0.19	2.53	-6.5

Tab. 2.5. Final throughput values for 10 MHz channel

Modulation	**R**	**Throughput [Mbps]**	**Required SNR [dB]**
BPSK(GMSK)	R = 0.76	10.13	1.75
BPSK(GMSK)	R = 0.57	7.60	-0.75
BPSK(GMSK)	R = 0.38	5.07	-3.25
BPSK(GMSK)	R = 0.285	3.80	-4.75
BPSK(GMSK)	R = 0.19	2.53	-6.5

Tab. 2.6. Throughput estimates for 5 MHz channel

Modulation	**R**	**Throughput [Mbps]**	**Required SNR [dB]**
8-PSK	R = 0.667	7.68	8.25
8-PSK	R = 0.53	6.11	5.9
O-QPSK	R = 0.80	6.14	5.25
BPSK (GMSK)	R = 0.90	6.00	4
8-PSK	R = 0.45	5.18	4.6
O-QPSK	R = 0.667	5.12	3.5
BPSK(GMSK)	R = 0.76	5.07	1.75
8-PSK	R = 0.333	3.84	2.3
O-QPSK	R = 0.50	3.84	1.3
BPSK(GMSK)	R =0.57	3.80	-0.75
8-PSK	R = 0.225	2.59	1
O-QPSK	R = 0.333	2.56	-1
BPSK(GMSK)	R = 0.38	2.53	-3.25
BPSK(GMSK)	R = 0.285	1.90	-4.75
BPSK(GMSK)	R = 0.19	1.27	-6.5

Tab. 2.7. Final throughput values for 5 MHz channel

Modulation	**R**	**Throughput [Mbps]**	**Required SNR [dB]**
BPSK (GMSK)	R = 0.90	6.00	4
BPSK(GMSK)	R = 0.76	5.07	1.75
BPSK(GMSK)	R = 0.57	3.80	-0.75
BPSK(GMSK)	R = 0.38	2.53	-3.25
BPSK(GMSK)	R = 0.285	1.90	-4.75
BPSK(GMSK)	R = 0.19	1.27	-6.5

Tab. 2.8. Final throughput values for 1.25 MHz channel

Modulation	**R**	**Throughput [Mbps]**	**Required SNR [dB]**
BPSK (GMSK)	R = 0.95	1.59	5.25
BPSK (GMSK)	R = 0.855	1.43	3
BPSK(GMSK)	R = 0.76	1.26	1.75
BPSK(GMSK)	R = 0.57	0.95	-0.75
BPSK(GMSK)	R = 0.38	0.64	-3.25
BPSK(GMSK)	R = 0.285	0.48	-4.75
BPSK(GMSK)	R = 0.19	0.31	-6.5

Tab. 2.9. Summary of all coding rates and achievable throughputs for four considered bandwidths

	20 MHz	**10 MHz**	**5 MHz**	**1.25 MHz**
R = 0.95	------	------	------	1.59 Mbps
R = 0.855	------	------	6.00 Mbps	1.43 Mbps
R = 0.76	------	10.13 Mbps	5.07 Mbps	1.26 Mbps
R = 0.57	15.20 Mbps	7.60 Mbps	3.80 Mbps	0.95 Mbps
R = 0.38	10.13 Mbps	5.07 Mbps	2.53 Mbps	0.64 Mbps
R = 0.285	7.60 Mbps	3.80 Mbps	1.90 Mbps	0.48 Mbps
R = 0.19	5.07 Mbps	2.53 Mbps	1.27 Mbps	0.31 Mbps

2.3.3. Results for GMSK with Turbo Code

Since all the performance curves have been obtained for the BPSK modulation scheme, it is necessary to compare them to curves obtained using GMSK modulation. It is expected that the performance of GMSK should be slightly worse than BPSK because of the inter-symbol interference introduced by the GMSK itself.

In Fig. 2.39, a comparison of BPSK and GMSK performance with R = 1/3 Turbo Code for different number of quantization levels in the GMSK detector. As can be seen, in the case of optimal GMSK decoding (with an infinite number of quantization levels), the performance deterioration is roughly 0.4 dB. In the case of 16 and 64/256 quantization levels, further degradation of tenths of dB is observed.

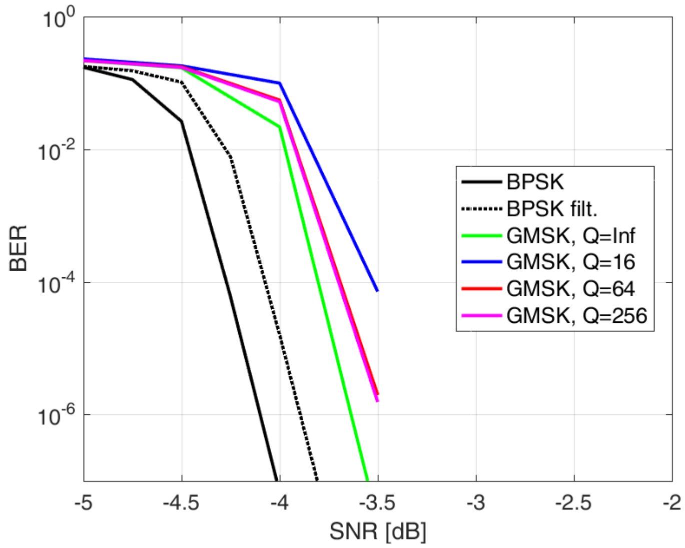

Fig. 2.39 BER performance for GMSK with Turbo Code

2.3.4. Code Rate Indicator selection

Since it has been assumed that more than one coding rate is used in the system it is necessary do develop an reliable way to inform the receiver which coding rate has been chosen for the transmission. This information is critical from the decoder point of view (assuming wrong coding rate makes the decoding process erroneous). Thus, to transmit this information an additional field in the physical frame have been considered. However, to obtain optimal performance a proper sequence need to be used for this purpose.

To find the best solution we investigated the following sequences/codes as the Code Rate Indicators (CRI):

- BCH code (59, 3)
- Repetition code (60, 3), denoted as REP in Fig. 2.40
- Cascade code of repetition code (60, 6) and BCH code (6, 3)
- M-sequence of length 63
- Walsh-Hadamard sequence of lengths 64, 115, and 128
- Gold sequence of length 63
- Kasami sequence of length 63
- Zadof-Chu sequence of lengths 63 and 127

Obtained results are presented in Fig. 2.40. The performance of all the considered CRI candidates are compared against the performance of the Turbo-codes with data rate R = 0.19.

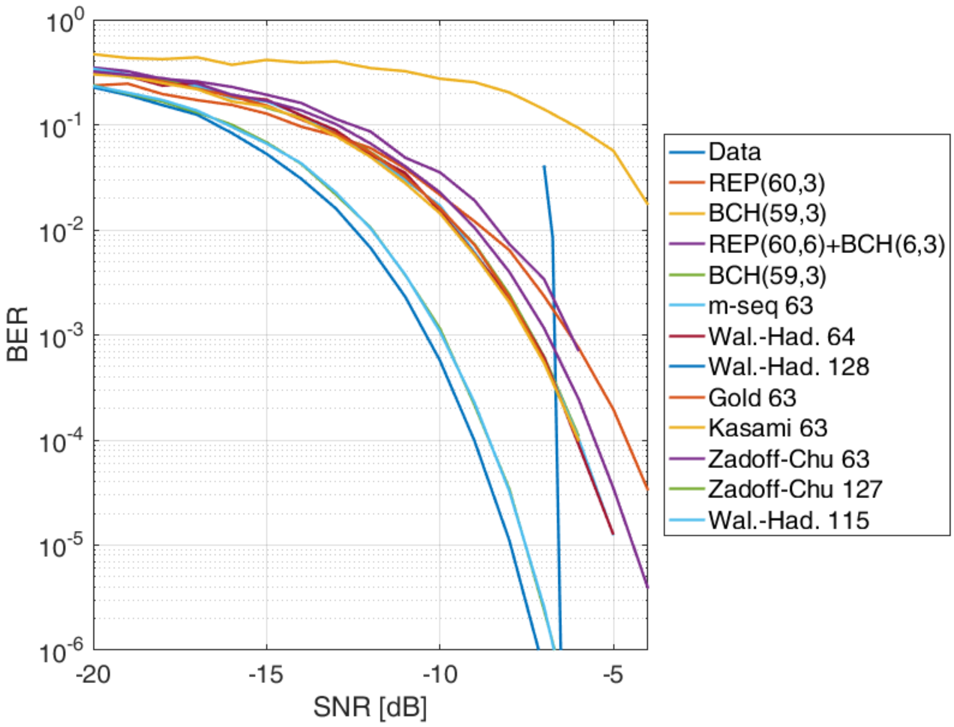

Fig. 2.40 BER performance for different CRI candidates

We can conclude that Walsh-Hadamard sequence of length 128 (or its shortened to 115 symbols version) or Zadoff-Chu sequence of length127 perform good enough to be applied as CRI and they will not limit the performance of channel coding used for protecting transmitted data.

2.4. Final remarks

At this stage of the project (Phase 1) most of the baseband processing algorithms and techniques proposed for application in the system have been evaluated separately. However, this evaluation has helped us to define which of them could be implemented to satisfy the system performance requirements. In Phase 2 the selected algorithms will be implemented in Matlab/Simulink environment as Simulink blocks and the complete model of the system, including transmitter, channel and receiver will be created. This will allow for evaluating the system as a whole and verify its performance for different values of parameters such as signal bandwidth, coding rate, physical frame structure, length of reference sequences, etc.

One should note that the system will be highly configurable when in use (on-orbit) as well. The settings will be selected by means of the forward control channel operating in UHF (uplink). It is possible to implement a decision system, using machine learning, which makes the transmission more efficient by selecting optimum values of several parameters automatically, based on satellite position, weather conditions, available channel bandwidth, etc.

Bibliography

[1] http://www.amsatuk.me.uk/iaru/spreadsheet.htm

[2] 2.1. Recommendations for Space Data System Standards “Radio Frequency and Modulation Systems – Part 1 Earth Stations and Spacecraft”, The Consultative Committee for Space Data Systems, Blue Book, February 2020

[3] 2.2. Report Concerning Space Data System Standards “Bandwidth-Efficient Modulations – Summary of Definition, Implementation, and Performance, The Consultative Committee for Space Data Systems, Blue Book, February 2018

[4] 2.3. K. Moruta, K. Hirade, "GMSK Modulation for Digital Mobile Radio Telephony," IEEE Transaction on Communications, Vol. 29, No. 7, July 1981

[5] 2.4. P. A. Laurent, „Exact and Approximate Construction of Digital Phase Modulations by Superposition of Amplitude Modulated Pulses (AMP)”, IEEE Transactions on Communications, Febr. 1986

[6] 2.5. J. Hagenauer, P. Hoeher, “A Viterbi algorithm with soft decision outputs and its applications,” IEEE Global Telecommunications Conference, GLOBECOM 1989.

[7] 2.6. D. Forney, “The Viterbi Algorithm”, Proceedings of IEEE, 61 (3), 1973, pp. 268–278

[8] 2.7. G. Ungerboeck, “Adaptive Maximum-Likelihood Receiver for Carrier-Modulated Data-Transmission Systems”, IEEE Transactions on Communications, Vol. 22, No. 5, May 1974

[9] 2.8. US Patent No. 6,731,700 B1 pt. „Soft Decision Output Generator”, Comsys Communication & Signal Processing Ltd., May 4, 2004

[10] H. Rouzegar, M. Ghanbarisabagh, “Estimation of Doppler Curve for LEO Satellites”, *Wireless Personal Communications* 108, 2195–2212 (2019). https://doi.org/10.1007/s11277-019-06517-5

[11] Elias Aboutanios, "Frequency Estimation for Low Earth Orbit Satellites", *PhD thesis*, 2002 [online, available at: https://opus.lib.uts.edu.au/bitstream/10453/43211/2/02whole.pdf, 12.04.2021]

[12] U. Mengali and M. Morelli, "Data-aided frequency estimation for burst digital transmission," in IEEE Transactions on Communications, vol. 45, no. 1, pp. 23-25, Jan. 1997, doi: 10.1109/26.554282.

[13] T. M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for OFDM," in IEEE Transactions on Communications, vol. 45, no. 12, pp. 1613-1621, Dec. 1997, doi: 10.1109/26.650240.

[14] E. Hosseini and E. Perrins, "Timing, Carrier, and Frame Synchronization of Burst-Mode CPM," in IEEE Transactions on Communications, vol. 61, no. 12, pp. 5125-5138, December 2013, doi: 10.1109/TCOMM.2013.111613.130667.

[15] S. Dolinar, D. Divsalar, and F. Pollara, “Code performance as a function of block size,” Jet Propulsion Laboratory (JPL), TMO Progress Report 42– 133, pp. 1–23, 1998. [Online]. Available: http://ipnpr.jpl.nasa.gov/progress_report/42-133/133K.pdf. [Acessed: June 6, 2016].

[16] X.-Y. Hu, E. Eleftheriou, D. M. Arnold, "Regular and Irregular Progressive Edge-Growth Tanner Graphs", IEEE Transactions on Information Theory, Vol. 51, No. 1, 2005, pp. 386 – 398

[17] S. Lin, D. J. Costello, “Error Control Coding”, Pearson Prentice-Hall, Upper Saddle River, NJ, 2004

[18] E. Arıkan, “Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels,” IEEE Trans. Inf. Theory, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[19] I. Tal and A. Vardy, “List Decoding of Polar Codes,” IEEE Int’l. Symp. Info. Theory (ISIT), 2011, pp. 1–5.

[20] M. Sybis, et al., “Channel Coding for Ultra-Reliable Low-Latency Communication in 5G Systems”, 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall), DOI: 10.1109/VTCFall.2016.7880930,