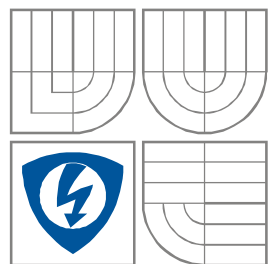




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ  
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF RADIO ELECTRONICS

## ADVANCED ALGORITHMS FOR SATELLITE COMMUNICATION SIGNAL PROCESSING

POKROČILÉ ALGORITMY ZPRACOVÁNÍ SIGNÁLŮ PRO DRUŽICOVOU KOMUNIKACI

DIZERTAČNÍ PRÁCE  
DOCTORAL THESIS

AUTOR PRÁCE  
AUTHOR

Ing. FILIP ZÁPLATA

VEDOUCÍ PRÁCE  
SUPERVISOR

prof. Ing. MIROSLAV KASAL, CSc.

BRNO, 2015

## **Abstract**

The dissertation is focused on software defined receivers intended for narrowband satellite communication. The satellite communication channel including deep space communication suffers from a high level of noise, typically modeled by AWGN, and from a strong Doppler shift of a signal caused by the unprecedented speed of an object in motion. The dissertation shows possible approaches to the issues of computationally efficient digital downconversion of narrowband signals and the carrier frequency estimation of narrowband signals distorted by the Doppler shift in the order of multiples of the signal bandwidth. The description of the proposed algorithms includes an analytical approach of its development and, if possible, the analytical performance assessment. The algorithms are modeled in MATLAB Simulink and the models are used for validating the performance by the simulation. The models were also used for experimental tests on the real signal received from the PSAT satellite at the laboratory of experimental satellites at the department of radio electronics.

## **Keywords**

Software defined radio, digital signal processing, satellite communication, deep-space communication, digital downconversion, multirate filtration, undersampling, Goertzel algorithm, frequency estimation, Doppler shift, MATLAB Simulink.

## **Abstrakt**

Dizertační práce je zaměřena na softwarově definované přijímače určené k úzkopásmové družicové komunikaci. Komunikační kanály družicových spojů zahrnujících komunikaci s hlubokým vesmírem jsou zatíženy vysokými úrovněmi šumu, typicky modelovaného AWGN, a silným Dopplerovým posuvem signálu způsobeným mimořádnou rychlostí pohybu objektu. Dizertační práce představuje možné postupy řešení výpočetně efektivní digitální downkonverze úzkopásmových signálů a systému odhadu kmitočtu nosné úzkopásmových signálů zatížených Dopplerovým posuvem v řádu násobků šířky pásma signálu. Popis navrhovaných algoritmů zahrnuje analytický postup jejich vývoje a tam, kde je to možné, i analytické hodnocení jejich chování. Algoritmy jsou modelovány v prostředí MATLAB Simulink a tyto modely jsou využity pro ověření vlastností simulacemi. Modely byly také využity k experimentálním testům na reálném signálu přijatém z družice PSAT v laboratoři experimentálních družic na ústavu radioelektroniky.

## **Klíčová slova**

Softwarově definované rádio, číslicové zpracování signálu, družicová komunikace, komunikace s hlubokým vesmírem, číslicová downkonverze, filtrace s více vzorkovacími kmitočty, podvzorkování, Goertzelův algoritmus, synchronizace nosné, Dopplerův posuv, MATLAB Simulink.

## Declaration

I declare that I have written my dissertation “Advanced algorithms for satellite communication signal processing” independently, under the guidance of the dissertation supervisor and using technical literature and other sources of information which are all quoted in the dissertation and listed in the references at the end of the dissertation.

As the author of the dissertation, I, furthermore, declare that, as regards to the creation of this dissertation, I have not infringed any copyright. In particular, I have not unlawfully encroached on anyone’s personal and/or ownership rights and I am fully aware of the consequences in the case of breaking Regulation §11 and the following Copyright Act No. 121/2000 Coll., and of the rights related to intellectual property right and changes in some Acts (Intellectual Property Act) and formulated in later regulations, inclusive of the possible consequences resulting from the provisions of Criminal Act No. 40/2009 Coll., Section 2, Head VI, Part 4.

Brno \_\_\_\_\_

\_\_\_\_\_  
(author’s signature)

## Bibliographic citation

ZÁPLATA, F. *Advanced algorithms for satellite communication signal processing*. Dissertation. Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, 2015. 116 pages.

## Acknowledgement

I would like to express my gratitude to my supervisor prof. Ing. Miroslav Kasal, CSc. for his advice, invaluable guidance and help throughout my research. I would also like to thank my girlfriend and family for their endless encouragement and patience throughout my studies.

Brno \_\_\_\_\_

\_\_\_\_\_  
*(author's signature)*

## **List of Abbreviations**

ADC	Analog to Digital Converter
AGC	Automatic Gain Control, Amplitude Gain Control
AM	Amplitude Modulation
ARM	Advanced RISC Machine
ASIC	Application Specific Integrated Circuit
AU	Astronomical Unit
AWGN	Additive White Gaussian Noise
BEP	Bit Error Probability
BER	Bit Error Ratio
BPM	Baseband Processing Module
BPSK	Binary Phase Shift Keying
CCSDS	Consultative Committee for Space Data Systems
CIC	Cascaded Integrator-Comb
CL	Closed Loop
CORDIC	Coordinate Rotation Digital Computer
CPFM	Continuous Phase Frequency Modulation
CPLD	Complex Programmable Logic Device
CRB	Cramer-Rao Bound
DA	Data Aided
DAC	Digital to Analog Converter
DBPSK	Differential BPSK
DD	Decision Directed
DDC	Digital Downconverter
DDFS	Direct Digital Frequency Synthesizer
DDS	Direct Digital Synthesizer
DFD	Dual Filter Detector
DFE	Direct From Earth
DFT	Discrete Fourier Transform
DMF	Derivative Matched Filter
DQPSK	Differential QuadriPhase Shift Keying
DSN	Deep-Space Network

DSP	Digital Signal Processing (Processor)
DTE	Direct To Earth
EDAC	Error Detection And Correction
EDL	Entry, Descent, Landing
EEPROM	Electrically Erasable Programmable Read-Only Memory
ESA	European Space Agency
FB	Feed-Back
FCC	Federal Communications Commission
FEC	Forward Error Correction
FF	Feed-Forward
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
FW	Firmware
GEO	Geostationary Earth Orbit
GMSK	Gaussian Minimum Shift Keying
GPM	General Purpose processing Module
GPP	General Purpose Processor
GPS	Global Positioning System
HB	Half-Band
HGA	High Gain Antenna
HW	Hardware
IC	Integrated Circuit
IF	Intermediate Frequency
IIR	Infinite Impulse Filter
IQ	Inphase-Quadrature
ISI	Intersymbol Interference
ISS	International Space Station
ITU	International Telecommunications Union
LEO	Low Earth Orbit
LGA	Low Gain Antenna
LMS	Least Mean Square
LNA	Low Noise Amplifier
LUT	Look-Up Table
MAC	Multiply and Accumulate

MAP	Maximum A Posteriori
MCRB	Modified CRB
MER	Mars exploration Rover
MEX	Mars Express
MGS	Mars Global Surveyor
ML	Maximum Likelihood
MRO	Mars Reconnaissance Orbiter
MSK	Minimum Shift Keying
MSL	Mars Science Laboratory
NASA	National Aeronautics and Space Administration
NCO	Numerically Controlled Oscillator
NDA	Non-Data Aided
NRZ-L	Non-Return to Zero Level
NRZ-M	Non Return to Zero Mark
ODY	Odyssey
OL	Open Loop
OQPSK	Offset Quadriphase Shift Keying
OS	Operational System
OSR	Oversampling Ratio
PA	Power Amplifier
PLL	Phase Locked Loop
PSD	Power Spectral Density
PSK	Phase Shift Keying
QPSK	Quadriphase Shift Keying
RAM	Random Access Memory
RC	Raised Cosine, Residual Carrier
RC BPSK	Residual Carrier BPSK
RF	Radio Frequency
RFM	Radio Frequency Module
RISC	Reduced Instruction Set Computing
RMS	Root Mean Square
ROM	Read-Only Memory
RSP	Reconfigurable Space Processor
RTOS	Real Time Operational System
RTTY	Radio-Teletype

SA	Spectrum Averaging
SC BPSK	Suppressed Carrier BPSK
SDR	Software Defined Radio
SDRAM	Synchronous Dynamic
SDST	Small Deep-Space Transponder
SEL	Single Event Latch-up
SET	Single Event Transient
SEU	Single Event Upset
SFDR	Spurious Free Dynamic Range
SNR	Signal to Noise Ratio
SP-L	Split Phase Level
SPM	Signal Processing Module
SRRC	Square Root Raised Cosine
SSPA	Solid State Power Amplifier
STRS	Space Telecommunication Radio System
SW	Software
TCM	Trellis Code Modulation
TDMA	Time Division Multiple Access
TDRSS	Tracking and Data Relay Satellite System
TMR	Triple Mode Redundancy
TT&C	Tracking, Telemetry and Command
TWTA	Travelling Wave Tube Amplifier
UHF	Ultra High Frequency
USO	Ultra Stable Oscillator
UW	Unique Word
VCO	Voltage Controlled Oscillator
VLIW	Very Long Instruction Word



## List of Symbols

$a$	Modulation symbol
$a_I$	Inphase modulation symbol
$a_Q$	Quadrature modulation symbol
$B$	Bandwidth
$C$	Tuning coefficient of the Goertzel filter
$\tilde{C}$	Complex tuning coefficient of the Goertzel filter
$C(k)$	Covariance function
$c_k$	$k^{\text{th}}$ symbol
$d$	Data symbol, Goertzel filter tuning offset
$E_b$	Energy per bit
$E\{x\}$	Expectation of $x$ , Energy of $x$
$e$	Error function
$f$	Frequency
$f_c$	Frequency of the carrier signal, Cutoff frequency of the filter
$f_N$	Cutoff frequency of the RC filter
$f_S$	Sampling frequency
$f_u$	Upper boundary frequency of the band
$f(t)$	Continuous function (signal)
$f_{BB}(t)$	Baseband continuous function (signal)
$f_{BP}(t)$	Bandpass continuous function (signal)
$F(\omega)$	Spectrum of the continuous function (signal)
$g(t)$	Sampled function (signal)
$G(\omega)$	Spectrum of the sampled function (signal), Power spectral density of the matched filter
$H(\omega)$	Heaviside step function in frequency domain, Frequency response of the filter
$h(t)$	Impulse response of the filter
$Im\{x\}$	Imaginary value of the complex variable $x$
$K$	Sampling order
$k$	Discrete Fourier transform bin index, Goertzel filter center frequency
$M$	Decimation coefficient of the CIC-filter, M-PSK modulation order

$m(t)$	Amplitude modulation function
$m_I(t)$	Amplitude modulation function of the inphase component
$m_Q(t)$	Amplitude modulation function of the quadrature component
$N$	Discrete Fourier transform length, Goertzel filter length
$N_0$	Noise power spectral density
$N_{FFT}$	DFFT width
$N(\omega)$	Power spectral density of noise
$OSR$	Oversampling ratio
$P$	Signal power
$p(t)$	Pulse function
$r(t)$	Analyzed input signal
$R(\omega)$	Autocorrelation function in frequency domain
$Re\{x\}$	Real value of the complex variable $x$
$r$	Index of the Nyquist zone
$s(t)$	Reconstruction function
$S(\omega)$	Spectrum of the reconstruction function, S-curve function, Power spectral density of the signal
$SNR$	Signal to noise ratio
$t$	Time
$T_R$	Symbol period
$T_S$	Sampling period
$u$	Vector of known parameters
$u(n)$	Discrete rectangular pulse function
$W$	Angular bandwidth
$w(t)$	AWGN noise
$x(n)$	Input samples of the discrete filter
$Y_k(n)$	Goertzel filter output samples
$y(n)$	Output samples of the discrete filter
$\alpha$	Roll-off factor; sampling time offset
$\beta$	Modulation index
$\delta(t)$	Dirac delta function
$\varepsilon$	Quantization error
$\zeta$	Damping ratio
$\Lambda(x)$	Log-likelihood function
$\lambda$	Vector of estimated parameters

$\mu$	Mean value of the probability function
$\sigma$	Standard deviation of the probability function
$\tau$	Symbol timing offset
$\varphi$	Signal phase
$\tilde{\varphi}$	Estimate of the signal phase
$\varphi_c$	Phase of the carrier signal
$\varphi(t)$	Phase modulation function
$\omega$	Angular frequency
$\tilde{\omega}$	Estimate of the angular frequency
$\omega_c$	Angular carrier frequency
$\mathcal{F}\{x\}$	Fourier transform of $x$

# Contents

<b>1.</b>	<b>Introduction.....</b>	<b>1</b>
<b>2.</b>	<b>State of the art.....</b>	<b>3</b>
2.1.	SDR for deep space communication .....	3
2.2.	Sampling theory .....	7
2.2.1.	Nyquist and bandpass sampling theory .....	7
2.2.2.	Non-uniform sampling theory .....	10
2.2.3.	Other effective ways of sampling .....	11
2.3.	Modulations for deep space communication.....	11
2.3.1.	BPSK .....	12
2.3.2.	QPSK .....	13
2.3.3.	OQPSK .....	13
2.3.4.	$\pi/4$ -QPSK.....	14
2.3.5.	Filtered OQPSK.....	14
2.3.6.	GMSK.....	15
2.4.	Downconversion .....	15
2.5.	Carrier synchronization .....	20
<b>3.</b>	<b>Objectives of the thesis .....</b>	<b>27</b>
<b>4.</b>	<b>Downconversion of narrowband signals.....</b>	<b>29</b>
4.1.	Digital downconversion of PM signals .....	29
4.2.	Novel approach to digital downconversion.....	35
4.2.1.	Low speed digital AM signal detection .....	37
4.2.2.	Goertzel filter composition .....	40
4.2.3.	Goertzel filter modification for real frequencies .....	42
4.2.4.	Improved structure of the decimating filter .....	49
4.3.	Digital downconverter performance .....	54
4.3.1.	BER influence .....	54
4.3.2.	Computational demands .....	57
4.3.3.	Quantization errors of the algorithms .....	58
4.4.	Simulink tests.....	60
4.4.1.	Modulator model.....	60
4.4.2.	RF channel .....	62

4.4.3.	Downconverter model based on CIC-filter.....	63
4.4.4.	Downconverter model based on the Goertzel algorithm .....	63
4.4.5.	BERtool models.....	66
4.4.6.	BER simulation results .....	69
4.5.	Summary .....	70
<b>5.</b>	<b>Carrier synchronization .....</b>	<b>72</b>
5.1.	Cramer-Rao bounds .....	72
5.2.	Maximum likelihood estimation .....	74
5.3.	Frequency estimation .....	75
5.3.1.	Maximum likelihood frequency estimator.....	75
5.3.2.	Spectrum averaging frequency estimator .....	77
5.3.3.	Closed loop spectrum averaging frequency estimator.....	81
5.3.4.	Modified closed loop spectrum averaging frequency estimator.....	83
5.4.	Maximum likelihood phase estimation .....	84
5.5.	Simulink tests.....	86
5.5.1.	DDS model .....	86
5.5.2.	Loop filter model .....	87
5.5.3.	DBPSK modulator and demodulator .....	87
5.5.4.	Maximum likelihood frequency estimator model.....	88
5.5.5.	Spectrum averaging frequency estimator model .....	91
5.5.6.	Maximum likelihood phase estimator model.....	95
5.5.7.	Estimation simulation results.....	95
5.6.	Summary .....	97
<b>6.</b>	<b>Experimental tests.....</b>	<b>99</b>
6.1.	Frequency synchronization .....	99
6.2.	Digital downconversion .....	106
6.3.	Summary .....	108
<b>7.</b>	<b>Conclusion .....</b>	<b>110</b>
<b>8.</b>	<b>References.....</b>	<b>112</b>

## List of figures

<b>Figure 2.1:</b> Diagram of allowed sampling frequency (shaded – aliasing region) [22]. ...	9
<b>Figure 2.2:</b> Signal diagram of the generic oscillator [40]. .....	18
<b>Figure 2.3:</b> Carrier synchronization of the coherent receiver signal diagram [41]. .....	21
<b>Figure 4.1:</b> Digital downconverter in Electra SDR [1]. .....	29
<b>Figure 4.2:</b> Half-band filter characteristics. ....	30
<b>Figure 4.3:</b> Integration intervals. ....	30
<b>Figure 4.4:</b> Integrator frequency characteristics ( $M = 4$ ). .....	31
<b>Figure 4.5:</b> Comb-filter frequency characteristics ( $M = 4$ ). .....	32
<b>Figure 4.6:</b> CIC-filter frequency characteristics. ....	33
<b>Figure 4.7:</b> Goertzel filter as a downconverter. ....	36
<b>Figure 4.8:</b> Goertzel filter frequency characteristics ( $N = 12, k = 2.3$ ). .....	37
<b>Figure 4.9:</b> Spectra of sampled AM signal. ....	38
<b>Figure 4.10:</b> The influence of different lengths of the Goertzel filter. ....	39
<b>Figure 4.11:</b> The example of signal windowing. ....	40
<b>Figure 4.12:</b> Spectrum position in the base band and filter selectivity. ....	40
<b>Figure 4.13:</b> Goertzel filter composition. ....	40
<b>Figure 4.14:</b> Realization structure of the composed Goertzel downconverter. ....	42
<b>Figure 4.15:</b> Plot of parameter $A$ for $N = 6$ . ....	43
<b>Figure 4.16:</b> Compensation coefficients for the algebraic form for $N = 6$ . ....	44
<b>Figure 4.17:</b> Compensation coefficients for magnitude for $N = 6$ . ....	45
<b>Figure 4.18:</b> Compensation coefficients for phase for $N = 6$ . ....	46
<b>Figure 4.19:</b> Phase compensation influence $N = 6; k = 0.82$ . ....	47
<b>Figure 4.20:</b> Spectrograms of the Goertzel filter for $N = 12$ – <b>a)</b> Uncompensated filter for $k = 2$ , <b>b)</b> Compensated filter for $k = 2$ , <b>c)</b> Uncompensated filter for $k = 2.3$ , <b>d)</b> Compensated filter for $k = 2.3$ . ....	48
<b>Figure 4.21:</b> Impulse responses of the modified Goertzel filter for $N = 128$ and $k = 0$ (left), $k = 4$ (right). ....	50
<b>Figure 4.22:</b> Magnitude frequency characteristics of the Goertzel filter and its modified version for $N = 32$ and $k = 0$ . ....	52
<b>Figure 4.23:</b> Phase frequency characteristics of the Goertzel filter and its modified version for $N = 32$ and $k = 0$ . ....	52
<b>Figure 4.24:</b> Signal diagram of the modified Goertzel filter. ....	53
<b>Figure 4.25:</b> Downconverter filter aliasing with decimation factor $M$ (left) and $M2$ (right). ....	54

<b>Figure 4.26:</b> Aliasing noise performance of the Goertzel and modified Goertzel downconverter.....	55
<b>Figure 4.27:</b> Aliasing noise performance of the Goertzel and modified Goertzel downconverter for the half decimation factor.....	56
<b>Figure 4.28:</b> CIC decimator in complex representation.....	57
<b>Figure 4.29:</b> Goertzel filter poles (left) and zero (right) misalignment due to quantization error. ....	58
<b>Figure 4.30:</b> Absolute quantization error for coefficient $C$ . ....	60
<b>Figure 4.31:</b> General schematic of the tested model.....	60
<b>Figure 4.32:</b> BPSK modulator model. ....	60
<b>Figure 4.33:</b> Modulator settings.....	62
<b>Figure 4.34:</b> AWGN channel settings.....	62
<b>Figure 4.35:</b> CIC-filter based downconverter.....	63
<b>Figure 4.36:</b> CIC-filter based downconverter settings.....	63
<b>Figure 4.37:</b> Goertzel filter based downconverter (left) and its setting window (right). ....	64
<b>Figure 4.38:</b> Goertzel downconverter of half decimation factor. ....	64
<b>Figure 4.39:</b> BERtool model of baseband BPSK modulator. ....	66
<b>Figure 4.40:</b> BERtool model of baseband BPSK modulator with Square-Root Raised Cosine filter.....	67
<b>Figure 4.41:</b> BERtool model of CIC-filter based downconverter.....	67
<b>Figure 4.42:</b> BERtool model of Goertzel downconverter.....	67
<b>Figure 4.43:</b> Modified Goertzel downconverter BER model. ....	68
<b>Figure 4.44:</b> Modified Goertzel downconverter BER model of half decimation factor. ....	68
<b>Figure 4.45:</b> BER simulation results of the Goertzel and modified Goertzel downconverters with no matched filter on the transmission side. The full-band version on the left hand side and the half-band version on the right hand side. ....	69
<b>Figure 4.46:</b> BER simulation results of the Goertzel and modified Goertzel downconverters and both full-band and half-band versions with the Raised-Cosine matched filter on the transmission side. The simulation for the filter with the roll-off factor 0.5 on the left hand side and for the roll-off factor 1.0 on the right hand side. ....	70
<b>Figure 5.1:</b> Closed loop ML frequency estimator.....	77
<b>Figure 5.2:</b> Spectrum averaging frequency estimator.....	78
<b>Figure 5.3:</b> Covariance estimation cumulation algorithm. ....	80
<b>Figure 5.4</b> Averaged covariance function for $E_bN_0 = 10dB$ (left) and $E_bN_0 = 0dB$ (right). ....	80
<b>Figure 5.5:</b> General closed loop frequency estimator. ....	81
<b>Figure 5.6</b> Direct digital synthesizer [61]. ....	82

<b>Figure 5.7</b> Second order proportional-plus-integrator loop filter [61].....	82
<b>Figure 5.8</b> Modified spectrum averaging frequency estimator.....	83
<b>Figure 5.9</b> Covariance peak of the modified spectrum averaging algorithm (left) and its derivative (right).....	84
<b>Figure 5.10</b> Closed loop ML phase estimator.....	86
<b>Figure 5.11</b> General schematic of the frequency estimator simulation model.....	86
<b>Figure 5.12</b> DDS Simulink model (left) and its configuration mask (right).....	87
<b>Figure 5.13</b> Loop filter model (left) and its configuration mask (right).....	87
<b>Figure 5.14</b> DBPSK modulator model.....	88
<b>Figure 5.15</b> DBPSK demodulator model.....	88
<b>Figure 5.16</b> DBPSK modulator settings (left) and demodulator settings (right).....	88
<b>Figure 5.17</b> ML closed loop frequency estimator.....	89
<b>Figure 5.18</b> ML estimator settings (left) and spectrum averaging frequency estimator settings (right).....	89
<b>Figure 5.19</b> ML closed loop estimator model.....	90
<b>Figure 5.20</b> Open loop spectrum averaging frequency estimator.....	91
<b>Figure 5.21</b> Open loop spectrum averaging frequency estimator settings.....	93
<b>Figure 5.22</b> Open loop spectrum averaging frequency estimator model.....	93
<b>Figure 5.23</b> Closed loop spectrum averaging frequency estimator.....	93
<b>Figure 5.24</b> Closed loop spectrum averaging frequency estimator model.....	95
<b>Figure 5.25</b> Closed loop ML phase estimator - Costas loop.....	95
<b>Figure 5.26</b> Frequency estimators S-curves.....	96
<b>Figure 5.27</b> Frequency estimators BER results (left) and carrier variance (right).....	97
<b>Figure 6.1:</b> Simulink model of the Costas loop with frequency aid.....	99
<b>Figure 6.2:</b> Simulink model of the open loop frequency and Costas loop phase synchronizers.....	100
<b>Figure 6.3:</b> Simulink model of the closed loop frequency and Costas loop phase synchronizers.....	100
<b>Figure 6.4:</b> Waterfall spectrogram with the frequency estimate from the open loop spectrum averaging estimator – coarse estimate (left) and from the Costas loop – fine estimate (right).....	101
<b>Figure 6.5:</b> Waterfall spectrogram frequency compensated by the coarse estimate of the open loop spectrum averaging frequency estimator.....	101
<b>Figure 6.6:</b> Waterfall spectrogram with the frequency estimate from the closed loop spectrum averaging estimator – coarse estimate (left) and from the Costas loop – fine estimate (right) with 100 Hz initial frequency.....	102
<b>Figure 6.7:</b> Waterfall spectrogram with the frequency estimate from the closed loop maximum likelihood estimator – coarse estimate (left) and from the Costas loop – fine estimate (right) with 0 Hz initial frequency.....	102



<b>Figure 6.8:</b> Waterfall spectrograms frequency compensated by the coarse estimate of the closed loop frequency estimators – spectrum averaging with 100 Hz initial frequency (left) and maximum likelihood with 0 Hz initial frequency (right). .....	103
<b>Figure 6.9:</b> Waterfall spectrogram with the frequency estimate from the closed loop spectrum averaging estimator – coarse estimate (left) and from the Costas loop – fine estimate (right) with 450 Hz initial frequency.....	103
<b>Figure 6.10:</b> Waterfall spectrogram with the frequency estimate from the closed loop maximum likelihood estimator – coarse estimate (left) and from the Costas loop – fine estimate (right) with 450 Hz initial frequency.....	104
<b>Figure 6.11:</b> Waterfall spectrograms frequency compensated by the coarse estimate of the closed loop frequency estimators with 450 Hz initial frequency – spectrum averaging (left) and maximum likelihood (right).....	104
<b>Figure 6.12:</b> Carrier synchronized complex (real part – green, imaginary part – red) signal of the open loop spectrum averaging estimator and Costas loop. ....	105
<b>Figure 6.13:</b> Carrier synchronized complex (real part – green, imaginary part – red) signal of the closed loop spectrum averaging estimator and Costas loop.....	105
<b>Figure 6.14:</b> Carrier synchronized complex (real part – green, imaginary part – red) signal of the closed loop maximum likelihood estimator and Costas loop.....	105
<b>Figure 6.15:</b> Simulink model of the experimental demodulator with Goertzel downconverter.....	106
<b>Figure 6.16:</b> Waterfall signal spectrogram after downconversion by the Goertzel full-band downconverter (left) and carrier synchronized output of the Costas loop (right). ....	106
<b>Figure 6.17:</b> Waterfall signal spectrogram after downconversion by the Goertzel modified full-band downconverter (left) and carrier synchronized output of the Costas loop (right).....	107
<b>Figure 6.18:</b> Waterfall signal spectrogram after downconversion by the Goertzel half-band downconverter (left) and carrier synchronized output of the Costas loop (right). ....	107
<b>Figure 6.19:</b> Waterfall signal spectrogram after downconversion by the Goertzel modified half-band downconverter (left) and carrier synchronized output of the Costas loop (right).....	108
<b>Figure 6.20:</b> Output synchronized signal from the Goertzel modified half-band downconverter and Costas loop phase synchronizer. ....	108

## List of tables

<b>Table 2.1:</b> Symbols of BPSK modulation.....	13
<b>Table 2.2:</b> Symbols of QPSK modulation.....	13
<b>Table 2.3:</b> Symbols of $\pi/4$ -QPSK modulation. ....	14
<b>Table 2.4:</b> Characteristic parameters of the recursive oscillators [40].....	19
<b>Table 4.1:</b> Half-band filter integration results.....	31
<b>Table 4.2:</b> CIC-filter integration results.....	33
<b>Table 4.3:</b> CIC-filter integration results for half bandwidth. ....	33
<b>Table 4.4:</b> Downconverter integration results.....	34
<b>Table 4.5:</b> Downconverter computational demands. ....	58

## 1. Introduction

Modern technology allows digitalization of signals of high bandwidths at high frequencies by powerful integrated analog to digital converters. Due to many years of work on digital signal processing, software defined radio (SDR) is becoming common nowadays. Even though its power consumption, space and technology demands are very high, compared with traditional analog technology, its benefits of reconfigurability, adaptability and algorithm abilities definitely make SDR a very perspective technology.

The SDR application is actually a program flashed into memory and run by a processing core like an ARM or some DSP, or it is a configuration file also stored in memory, but used for configuring a programmable gate array – CPLD or bigger FPGA. The processors work, sequentially, instruction by instruction and it slows their processing down, but their processing variability is vast, depending just on the flash memory space, which is mostly vastly extendable. There is a variety of processors available from general purpose microcontrollers with many integrated peripherals over specialized DSPs controlled by a very long instruction set (VLIW), including parallel processing, to GHz order speed multi-core signal processors of the highest performance able to process and put through tenths of Gbps. Programmable gate arrays, on the other hand, work with several times lower speed, but many processes are made in parallel and the processes seem to be processed in one instant. Except for its enormous processing speed and throughput, the number of gates is highly limited and the cost is still very high.

The conversion between the analog and digital domain is performed by an analog to digital converter (ADC) and a digital to analog converter (DAC), respectively. Technology in this area is also highly developed; there are many approaches to the conversion designed for specialized applications like instrumentation, audio and video applications or radio frequency processing. The sigma-delta converters provide, due to their oversampling property, the best noise performance, but the oversampling, on the other hand, does not allow high speed sampling. The integration converters are able to suppress periodic disturbances like power-line harmonic noise, and therefore are used in measurement equipment. The RF signals are converted mostly by the pipelined ADCs, because the pipelining used in this technology allows to reach the highest sampling frequencies and that of several hundreds even thousands Msps. The bandwidth of a usual signal is often many times lower, but the signal is positioned at higher frequencies. Using bandpass sampling is wise, as the need for sampling frequency depends on the signal's bandwidth instead of its frequency position. The sampler still has to be fast enough to catch the sample correctly.

Deep-space exploration is provided by probes, but it is not a rare situation that the mission is composed of more than one single machine. A distant planet may be explored by several landers, which have to communicate with the Earth to send images or results of an analysis or even with each other. It is impossible to provide direct connection to the Earth all the time and for all positions on the planet. An orbiter is used to overcome the signal drop-out and the long distances between the landers. It is unlikely that all landers will use the same signal and coding configurations, therefore the orbiter has to

be adaptable for all possible communication schemes. The adaptation can be controlled simply by the Earth station, but the signal propagation time for such long distances could cause many problems. There are also unpredictable or anomalous events during entry, descent and landing, which can cause a temporary loss of communication. The prediction of, e.g., the object speed and the precise influence of the atmosphere, is a very difficult task and even tiny uncertainties may cause Doppler swings and failure of the mission. These are the examples where slight tuning will save much money and time, but hard wired circuits do not allow it, thus there is a place for SDRs. [1]

Additionally, an SDR saves space on a probe and weight because of its reconfigurability for many communication standards, where the processing unit replaces blocks of analog circuits for each standard. It is even possible to make an upgrade of the whole transceiver and add or modify the communication standards. Considering extreme conditions in deep-space, there comes up questions about power consumption, heat sinking, radiation resistance, etc.

In the following text, the novel algorithms for the selected operations within the software defined receivers for satellite communication are discussed. The first chapter briefly describes the existing solutions of the SDRs and spectral efficient modulations used for space communication, revises the common sampling theory needed for the proper understanding of digital signal processing and reviews the two main particular topics of the dissertation; digital downconversion and carrier frequency estimation. The definition of the dissertation objectives follows in chapter 3. Chapter 4, 5 and 6 make the core of the dissertation and thoroughly describe the development of the novel approaches to digital downconversion and frequency estimation in SDRs including simulations and real satellite signal tests.

## 2. State of the art

### 2.1. SDR for deep space communication

The communication of spacecrafts and probes exploring deep space (distances greater than 2 million km [2]) and the Earth is provided by the deep-space network (DSN). The on-earth system consists of three stations: Goldstone in California, Madrid and Canberra in Australia; all with 70 m diameter and smaller antennas. The location is not random, but rather makes a constellation allowing as much coverage of the open space as possible. The probes are made to generate a large amount of uncompressed data (MRO expects a return link transferring up to 5 Gbits a day [3]) and the transfer direct to the Earth (DTE) for distances of millions of km and more requires high energy per bit and the operational complexity of pointing the high-gain antenna. The complexity of the ground stations is performed by the huge antennas and associated control stations, but it is difficult to provide such a performance on the extra-terrestrial surface by mobile rovers. Such inconvenience is usually overcome by a relay link, which is typical, e.g., for Mars exploration, where the communication with the spacecraft Phoenix, the rovers Spirit (Mars exploration rover – A, MER-A), Opportunity (Mars exploration rover – B, MER-B), Curiosity (Mars science laboratory, MSL) and more is passed primarily through NASA's Mars orbiters Odyssey (ODY) and Mars reconnaissance orbiter (MRO) and secondarily by NASA's Mars global surveyor (MGS) and ESA's Mars express (MEX). The orbiters carry the low gain nadir looking antennas (LGA) to provide communication with the landers and a large (MRO – 3 m diameter) gimbaled high gain antenna (HGA) to provide communication with the Earth DSN station at a distance of up to 2.6 AU (400 mil. km) from Mars's orbit. There is also a need for robust capture of telemetry during critical mission events (entry, descent and landing – EDL), which is provided more easily by the orbiters. [4] NASA even had established the policy of requiring the communication during EDL after the loss of a Mars polar lander in 1999. [5] Moreover, it is sometimes naturally impossible to provide the communication links DTE and DFE (direct from the Earth) due to inexpedient planetary constellation. For communication with DSN, frequency channels in X-band are reserved and for Doppler suffering in situ communication (hundreds to thousands of km), the channels around 400 MHz in the UHF band are used. The X-band is used for high volume data from the orbiters by its HGA (MRO - 500 kbps at 400 mil km distance and 4 Mbps at 100 mil km) [3], for DTE and DFE commanding at low data rates (160 bps at 34 m DSN antenna, 800 bps at 70 m DSN antenna uplink and 1 – 2 kbps downlink) the UHF band is used [6]. The DTE and DFE communication is useful during descent, when the UHF band frequencies are shielded by the heating plasma envelope, and in safe mode, when the spacecraft (e.g. MRO) is automatically aimed at the Sun by its solar panels independently on the HGA pointing. [3] Although the UHF band may be used for commanding, it is primarily intended for high volume data transfer at rates in the order of 4 Mbps. For even higher volume transfer the Ka-band communication has already been tested. To provide such challenging communication links, highly reconfigurable radios are essential. NASA started using software defined radios onboard spacecrafts with the MRO mission in 2005 and SDR Electra. The design

has been so successful that ESA uses it in its ExoMars missions planned in 2016 and 2018 demonstrating relay communication. [7]

In past and older running missions, conventional non-programmed radios were assembled. The typical representative of the conventional radio, which has been used in Deep Space, ODY, MER and other spacecrafts, is a Small deep-space transponder (SDST) [3], [8]. It is a reconfigurable radio, in which support of different transmission schemes lies in different hardware switched on depending on specific requirements. [9] Such a radio must be heavy, large-sized and power hungry to be highly flexible, however, the flexibility is still very limited. The Cassini-Huygens story is a nice example of conventional radio limitations. The long term mission to Saturn and its moon Titan would fail during the Huygens probe descent to Titan because the radio on the Cassini spacecraft did not properly account for the Doppler profile. Increasing the bandwidth of the bit synchronization loop by an in-flight software upgrade would easily save the mission, but it is hard-wired. [10] Actually, the SDR architecture has unprecedented operational flexibility allowing in-flight functionality upgrades or failure corrections by data patches. The same HW can be used across mission phases and also SW is reusable. [11] There are more general types of SDR from the flexibility and hardware configuration point of view [9], [1]:

- Legacy radio  
It is a conventional non-programmable transceiver with limited processing and HW functionality under SW control.
- Software defined radio  
The physical layer of the transceiver is in SW or FW, the functionality is defined by the SW, but it is not reconfigurable. The radio still needs some RF HW.
- Software radio  
Communication functions are performed primarily through SW and minimal RF HW. Digitization of the RF signal occurs at the antenna.
- Autonomous radio  
The radio is able to recognize the features of an incoming signal and respond intelligently without explicit pre-configuration or reprogramming.

The autonomous radio is a highly cognitive radio used in deep-space exploration. Automatic communication without pre-configuration is valuable in situations where the orbiter may come within view of various landed assets several times a day. Reconfiguration would be this frequent and manual reconfiguration would be a daunting task. This radio can almost optimally handle the large Doppler swings caused by rocket firings, a parachute opening, a back shell ejecting and a bouncing landing on the surface or unpredictable properties of the atmosphere. It decreases needed link margins to a minimum and saves the spacecraft's resources. [1], [6]

The need for SDRs has become obvious, indeed, and many manufacturers have shown an interest in the development of HW and SW of a space compliant SDR. Space agencies allow outsourcing of SDR development, but it needs some standardization. NASA prepared a Space telecommunication radio system (STRS), which defines the design rules intended for 3<sup>rd</sup> party SW, FW and HW developers of SDR. [9], [11] According to STRS, the SDR should be dividable into 3 main functional components with the following tasks [9]:

- General purpose processing module (GPM)  
It usually contains a general purpose processor (GPP) and/or a digital signal processor (DSP), memory, tracking, telemetry and command (TT&C) interface and ground support and a testing interface all connected by a system bus. The GPP elements accommodate an embedded operational system (OS) supporting configuration, control and status check of the radio and the SRTS compliant operational environment. The memory is volatile for OS functionality and non-volatile as storage for FPGA configuration. EEPROM is typically used, flash memories are allowed, but these are usually not space qualified.
- Signal processing module (SPM)  
The module typically consists of application specific ICs (ASICs), FPGAs and DSPs, memory and an interconnection bus. It serves as a more or less autonomous signal processor transforming the received signal to data packets and the data packets to the signal to be transmitted. The main tasks of the digital signal processing are digital up/down conversion, digital filtering, carrier recovery and tracking, data and symbol synchronization, forward error correction (FEC), digital automatic gain control (AGC), symbol mapping, data detection, spreading/despreading, scrambling/descrambling, encryption/decryption and input/output interfacing.
- Radio frequency module (RFM)  
The RFM performs the interconnection between the analog and digital domain and the power conversion by LNA at the RF input and PA at the RF output. The main constituents of the module are ADCs and DACs with analog filters, diplexers, LNAs, PAs, RF switches and analog up/down converters. The radiometric tracking needs full or partial support from the analog module.

The electronics sent out of the Earth's magnetosphere is exposed to several times stronger radiation causing temporary or permanent changes in its behavior. The electronics are shielded from the flow of high energetic particles from the Sun, but shielding alone is often not practical and components with improved radiation hardness have to be used. The minimal absorbed radiation dose of the space qualified components should be 100 krad. It is quite easily feasible for low Earth orbits (LEO), but for geostationary orbits (GEO) additional shielding is needed. The demands on the reliability of the system in such a violent environment are so high that software measures have to be implemented. The errors are divided into three groups [12]:

- Single event latch-up (SEL)
- Single event upset (SEU)
- Single event transient (SET)

The used methods dealing with these errors are watchdog, scrubbing, error detection and correction (EDAC), or triple mode redundancy (TMR). The watchdog is a simple timer which, when it is not reset by SW, causes a hard reset of the sequence processing system (GPP, DSP) and thus prevents unrecoverable states. The EDAC and TMR serve mainly for FPGA configuration and the non-volatile memory setting correction; the actual state is compared with the redundant storage and the errors are corrected. It is usually combined with error detection and correction codes. The volatile memory uses

“hot-swap” of 2 spare SDRAM columns for fast error correction. Harris corp. tested its radio with scrubbing and TMR, which resulted in 150:1 decrease of error occurrence with only scrubbing used and 2500:1 with the combination of scrubbing and TMR. It is interesting that TMR alone increased the error occurrence by 10%; it was possibly caused by the increased area of the chip. [13]

Many of the previously mentioned spacecrafts exploring Mars have used conventional transceivers with limited capability, particularly Odyssey orbiter and Mars exploration rovers (MER-A, MER-B) are assembled with a CE505 radio manufactured by Cincinnati electronics [11], [14]. The ESA’s MEX carries the UHF transponder MELACOM, which is a legacy type radio with 1-bit ADC and memory for recording. The recorded data is processed offline on Earth [5]. The radios are used for relays in running missions, but without adaptive data rate capability, which an Electra SDR assembled on MRO offers [6]. Modern radios using SDR technology are developed mainly by Jet propulsion laboratory, Harris Corp. and General Dynamics Corp [11].

Electra is NASA’s first highly capable SDR for space missions. It works in the UHF frequency range 390 – 450 MHz (Rx 390 – 405 MHz, Tx 435 – 450 MHz) with suppressed carrier BPSK (SC BPSK) modulation and residual carrier BPSK (RC BPSK) modulation with a modulation index typically  $\pi/3$ . The data rate is automatically controlled according to the link quality in the range from 1 kbps to 4 Mbps. Electra’s RFM includes single channel analog up/down conversion from/to IF at 71.795 MHz, ADC, DAC and 15 W solid state power amplifier (SSPA) in the UHF band. The 12-bit ADC performs the first-order bandpass sampling wherein the IF is mapped directly down to the digital baseband. The sampling rate of 19.18 MHz and conversion frequencies are derived from an external ultra-stable oscillator (USO) with a frequency of 76.72 MHz. The signal in the digital domain in the baseband is processed by FPGA within the SPM especially for Electra called a baseband processing module (BPM). Processing within the receiver includes a digital downconversion from the digital low IF controlled by the system of carrier recovery and tracking, matched filtering, timing recovery and demodulation. The system is able to track up to 20 kHz of Doppler frequency uncertainties, which is typical for the UHF band, and the bandwidth of the tracking loop is programmable from 10 Hz to 10 kHz. The RC BPSK is tracked by the PLL, and the SC BPSK is tracked by the Costas loop. The arm filters of the loops are 1<sup>st</sup> order Butterworth low pass filters with a programmable cut-off frequency to efficiently reduce noise and set the loop bandwidth. The values are pre-computed as a function of data rate and two sets are needed; one for acquisition and one for tracking. The acquisition is performed as a simple saw-tooth sweeping starting at a defined frequency with a typical step of 0.375 loop bandwidth; when the difference of the accumulated real and imaginary magnitude reaches the threshold, the acquisition is switched to tracking. The general purpose processing is a part of the BPM and is provided by a 32-bit microprocessor with memory and accommodates the Proximity-1 link protocol. The Electra radio with a mass of 4.9 kg and dimensions  $17.2 \times 21.9 \times 14$  cm is used onboard the MRO for high data rate relay within the return link from Mars’s surface. [4], [3], [1]

The resources available on landed assets are even more limited than on orbiters and this is the reason why the Electra lite SDR was developed. The aim was to leverage the Electra radio rather than designing a new radio. The Electra lite is less capable, but less massive and less power hungry; the mass was reduced to 3 kg. The MSL carries 2 Electra lite radios (one redundant) to transfer high volume data to the Earth through the orbiters. Thanks to the autonomous architecture, the radio is compatible with MRO’s Electra radio as with ODY’s CE505, MEX’s MELACOM and the older version of the



UHF radio onboard MGS. Compatibility is limited to the lower fixed data rates and the residual carrier modulation excluding MRO, where the adaptive data rates and suppressed carrier modulation are fully supported. Additionally, MRO communication supports two more radio channels compatible with CCSDS, 404.4 MHz and 397.5 MHz, besides the standard channel 0 at 401.585625 MHz. [5], [6], [4], [15]

In late 2012, NASA tested its first Ka-band SDR developed by Harris Corp. It was placed on the truss of the International Space Station (ISS) together with the 2 S-band and 1 L-band radios creating a STRS compliant testbed. The testbed is intended for tracking and data relay satellite system (TDRSS) with data rates up to 100 Mbps in Ka-band. The RFM employs direct sampling at the IF, the radio has the output/input at the S-band, and the Ka-band conversion is performed externally, close to the antenna due to coaxial cable losses. The heart of the GPM is a PowerPC 750 class processor with a triple redundant 128 MB SDRAM and a 64 MB flash taking care of control and monitoring of the radio and running commercially available VxWorks RTOS with the STRS operating environment. The SPM is created by a V4 reconfigurable space processor (V4 RSP), which contains a DSP module SMJ320C6701 from Texas Instruments and four Xilinx Virtex 4 FPGAs. Two FPGAs are intended for the transmission direction and two for the reception direction, but the neighboring two are interconnected. The radiation error prevention system includes the EDAC system and scrubbing ASIC. The V4 RSP can be completely reprogrammed and begin operating in a new configuration in less than 15 seconds. Harris Corp. continued with the development and significantly increased the performance with a new SiP100 SPM. The SiP100 has increased levels of integration compared to V4 RSP and consists of 5 FPGAs, one DaVinci DSP with DSP plus ARM9 cores, two 125 Msps 14-bit ADCs, two 500 Msps DACs, 1 GB RAM and 1 GB flash memory. The design is suited for small systems. [12], [13], [11]

## 2.2. Sampling theory

### 2.2.1. Nyquist and bandpass sampling theory

Sampling is the first of two key actions of digitizing the signal, i.e. the discretization in time. The mathematical expression of such a process is depicted in

$$g(t) = \sum_{n=-\infty}^{\infty} f(t)\delta(t - nT_s). \quad (2.1)$$

The real signal  $f(t)$  is multiplied by a set of Dirac delta functions  $\delta(t)$  equidistantly spaced by a sampling period  $T_s$ .

For errorless reconstruction, sampling has to meet the condition of double sampling frequency – Nyquist condition [16]

$$T_s < \frac{1}{2B} \quad (2.2)$$

where  $B$  is the bandwidth of the sampled signal. It is obvious from the condition that the sampled signal has to be bandlimited; such a signal can be expressed by the Fourier transform

$$f_{BB}(t) = \frac{1}{W} \int_{-\frac{W}{2}}^{\frac{W}{2}} F(\omega) e^{j\omega t} d\omega, \quad W = 2\pi B, \int_{-\frac{W}{2}}^{\frac{W}{2}} |F(\omega)|^2 d\omega < \infty \quad (2.3)$$

as has been published, e.g., in [17].  $F(\omega)$  is the signal representation in the frequency domain and  $W$  is the angular bandwidth defined in (2.3). The Fourier transform is defined only for the signal of finite energy, i.e. the signal has to meet the condition (2.3).

The reconstruction itself is a convolution of the samples of  $g(t)$  with a reconstruction function  $s(t)$  as is mathematically denoted by the sum [18]

$$f(t) = \sum_{n=-\infty}^{\infty} g(nT_s) s(t - nT_s) = \sum_{n=-\infty}^{\infty} f(nT_s) s(t - nT_s). \quad (2.4)$$

There is also denoted the equality of the sampled and continuous signal at the time position of the sample acquisition.

It is possible to find the reconstruction function by comparing the spectrum of the sampled signal  $G(\omega)$  with the spectrum of its original signal  $F(\omega)$  [19]. Assuming that the Fourier transform of the Dirac delta function is

$$\sum_{k=-\infty}^{\infty} \delta(n - kN) = \frac{1}{N} \sum_{k=-\infty}^{\infty} e^{j2\pi n \frac{k}{N}} \quad (2.5)$$

where the sampling period is denoted in discrete form by  $N$ , the spectrum of the signal (2.4) is

$$\mathcal{F} \left\{ \frac{1}{T_s} \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} f(\tau) s(t - \tau) e^{j2\pi n \frac{\tau}{T_s}} d\tau \right\} = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} S(\omega) F \left( \omega + 2\pi \frac{n}{T_s} \right). \quad (2.6)$$

From (2.6) it is also noticeable the periodicity of the spectrum of the sampled signal as is depicted in

$$G(\omega) = F \left( \omega + 2\pi \frac{n}{T_s} \right). \quad (2.7)$$

It is clearly seen that the reconstruction function has to cut off the additional periodic spectrum of the signal  $g(t)$ , and if the sampling meets the condition (2.2), the signal will be ideally reconstructed. The cutting spectra is expressed by the subtraction of two shifted Heaviside step functions  $H(\omega)$ , and its inverse Fourier transform finally gives the reconstruction function for the Nyquist sampling

$$s(t) = \mathcal{F}^{-1} \left\{ H \left( \omega + \frac{W}{2} \right) - H \left( \omega - \frac{W}{2} \right) \right\} = \frac{1}{W} \int_{-\frac{W}{2}}^{\frac{W}{2}} e^{j\omega t} d\omega = \frac{2 \sin \left( \frac{W}{2} t \right)}{Wt}. \quad (2.8)$$

In the following text, the maximal signal bandwidth is assumed as says the condition

$$\frac{T_s}{2\pi} \rightarrow \frac{1}{4\pi B} = \frac{1}{2W}. \quad (2.9)$$

If the Nyquist condition is not fulfilled, the baseband signal will be aliased with the signal above the half of the sampling frequency. Nevertheless, if the baseband signal is suppressed to zero and the sampled signal is bandlimited, the signal will not be aliased. This technique is known as bandpass sampling or undersampling [20].

Minding the finite energy condition (2.3), a real bandpass signal can be defined by

$$f_{BP}(t) = \frac{1}{2W} \int_{\omega_c - \frac{W}{2}}^{\omega_c + \frac{W}{2}} F(\omega) e^{j\omega t} d\omega + \frac{1}{2W} \int_{-\omega_c - \frac{W}{2}}^{-\omega_c + \frac{W}{2}} F^*(-\omega) e^{j\omega t} d\omega. \quad (2.10)$$

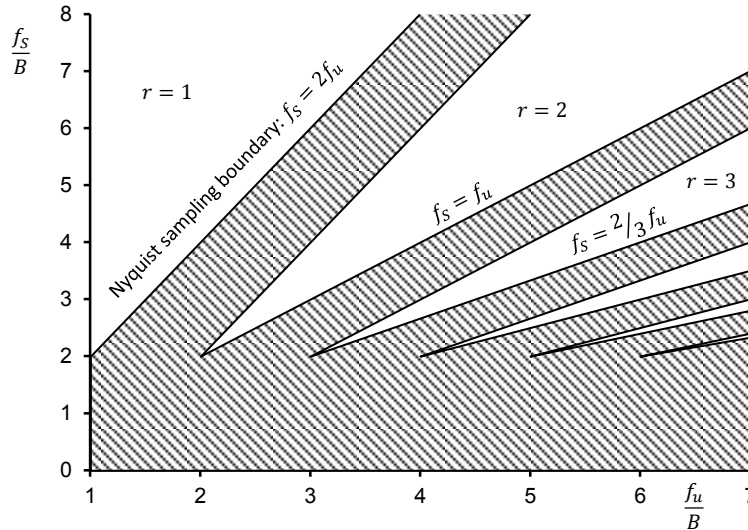
The real bandpass signal arises from the baseband signal by splitting its spectrum into two identical parts of halved amplitudes and shifting them into center frequency  $\pm\omega_c$ . [21].

Assuming uniform bandpass sampling, the condition (2.2) extends into

$$\frac{2\omega_c + W}{r} < \frac{2\pi}{T_s} < \frac{2\omega_c - W}{r-1}, \quad 1 \leq r \leq \left\lfloor \frac{\omega_c}{W} + \frac{1}{2} \right\rfloor. \quad (2.11)$$

The condition defines the regions of the aliasing-free sampling; it can be visualized in the well-known Figure 2.1 [22], where, for brevity, the sampling frequency  $f_s = 1/T_s$  and the upper frequency boundary of the band  $f_u = \omega_c/2\pi + W/4\pi$ . The coefficient  $r$  determines the so-called Nyquist zones, where the first Nyquist zone corresponds to the Nyquist sampling. The brackets  $\lfloor \cdot \rfloor$  denote the floor function.

The region of allowed sampling frequencies widens when using oversampling, but when the sampling frequency goes down to the boundary level of twice the sampling frequency, the permitted region collapses into discrete points. These represent the most efficient sampling and demand precise frequency planning by using so-called integer-band-positioning or half-integer-band-positioning [22].



**Figure 2.1:** Diagram of allowed sampling frequency (shaded – aliasing region) [22].

### 2.2.2. Non-uniform sampling theory

The previous chapter dealt with the sampling theorem using uniform sampling, i.e. each sample is equally spaced from the adjacent sample. However, real sampling suffers from non-ideal timing and this leads to sample value errors. One method how to deal with this problem is to minimize timing errors. Another way is to use the reconstruction function to find the real value depending on the knowledge of the precise time of acquisition. Even though the theory of reconstruction of non-uniform sampling has been known for over fifty years, in radio-communications it is realizable since the digital signal processing boom offers higher computational power.

There are many distributions of non-uniform sampling [23], but the simplest one seems to be high-order sampling ( $K^{th}$ -order sampling), which can be realized in two ways as has been shown in [24]. The first approach uses  $K$  separate samplers with the same input signal, but for each, the timing is differently phase-shifted. Another way is to use the same clock and delay the input signal differently for each sampler by the sampling time offset  $\alpha$ . The second approach is mathematically expressed in

$$g(t) = \sum_{k=0}^{K-1} f(t + \alpha_k) \sum_{n=-\infty}^{\infty} \delta(t - nKT_s). \quad (2.12)$$

Letting the order  $K$  to be infinite, sampling will become random. Literature [23] divides the random sampling into some special cases, but all approaches have to meet the Nyquist condition (2.2), i.e. the average sampling frequency has to be at least twice the bandwidth of the sampled signal. In the following, the discussion is aimed at second order sampling.

One can imagine the second order sampling as two independent chains of samples, the spectra will be correspondingly independent as well and will possibly be aliased, because of the halved sampling frequency. It is obvious that the reconstruction function has to suppress aliasing and keep the signal within the desired spectrum.

The second order sampled signal is defined by

$$g(t) = \sum_{n=-\infty}^{\infty} f(t)\delta(t - 2nT_s) + f(t)\delta(t - 2nT_s + \alpha). \quad (2.13)$$

Using the shift property yields the spectrum as shows the sum with spectra of reconstruction functions  $S(\omega)$ .

$$F(\omega) = \frac{1}{2T_s} \sum_{n=-\infty}^{\infty} S_A(\omega)F\left(\omega + \pi \frac{n}{T_s}\right) + S_B(\omega)F\left(\omega + \pi \frac{n}{T_s}\right)e^{-j\pi\alpha\frac{n}{T_s}}. \quad (2.14)$$

The parameters of both spectra are the same except for the phase, which logically depends on the known sample dislocation.

It is obvious from the analysis of the sum of the spectra (2.14) according to [19] that the spectra affecting the baseband frequency region are for  $n = \{-1, 0, 1\}$  yielding the system of equations

$$\frac{1}{2T_s} S_A(\omega) + \frac{1}{2T_s} S_B(\omega) = 1 \frac{1}{2T_s} S_A(\omega) + \frac{1}{2T_s} e^{-j\pi\alpha\frac{1}{T_s}} S_B(\omega) = 0 \quad (2.15)$$

with the solution

$$S_A(\omega) = S_B^*(\omega) = \frac{2T_s}{1 - e^{-j\pi\alpha\frac{1}{T_s}}}. \quad (2.16)$$

Assuming the reconstruction functions  $s(t)_{A,B}$  are real, the inverse Fourier transform of (2.16) gives the reconstruction function for the second order sampling [19]

$$s_A(t) = s_B(-t) = \frac{\cos\left(Wt - \pi\alpha\frac{1}{2T_s}\right) - \cos\left(\pi\alpha\frac{1}{2T_s}\right)}{Wt\sin\left(\pi\alpha\frac{1}{2T_s}\right)}. \quad (2.17)$$

The reconstruction process itself is provided by the convolution of the split samples  $g(t)$  and the corresponding reconstruction function  $s(t)$  as depicted in equation

$$f(t) = s_A(t) * g_A(t) + s_B(t) * g_B(t). \quad (2.18)$$

The splitting of the samples simply means the separation of the addends within the sum (2.14) into  $g_A(t)$  and  $g_B(t)$ , respectively. In real applications, the reconstruction function may act as an impulse response of an FIR filter.

Finally, some words about the extreme case of high-order sampling, random sampling, should be mentioned. There have been concepts published about free-aliasing sampling [25] or [26], when by using random sampling it is possible to reconstruct the spectrum of the sampled signal even without using any anti-aliasing filtration. However, for full signal reconstruction, the Nyquist condition still has to be met.

### 2.2.3. Other effective ways of sampling

In [27] and [28] there was proposed the concept of the so-called level-crossing sampling. This idea does not use the common way of acquiring the value of the input signal at a precisely defined time, but it is based on measuring the time of the event when the signal crosses a precisely defined value of voltage. The result of such sampling is apparently a set of samples non-uniform in time and the average sampling frequency is much higher than the Nyquist rate. This is true for properly set acquisition levels and the signal of an expected range of amplitude. Thus such a technique could be useful for constant envelope signals.

For highly sparse signals, a new approach has been published [29]. Before sampling by the analog to digital converter, an input signal is smeared across a wide spectrum, then the signal is filtered by an anti-aliasing filter. The sampling is at a much lower rate than is necessary in common ways of sampling. Afterwards it is no longer possible to reconstruct the original signal by linear operations. The reconstruction computations are highly non-linear and consume a lot of digital processing power. The proposed technique has an advantage in lowering the demands on the analog to digital converter and analog hardware.

## 2.3. Modulations for deep space communication

Compared to the influence of Earth's atmosphere, the disturbing effects that may occur in space are benign. The major influence to the received signal is an AWGN and the amplitude fluctuations caused by multipath interferences and solar scintillations.

The Doppler effect is, unlike Earth's services, more crucial because of much higher relative velocities of the objects. [1]

The sources of power on board a spacecraft or a probe are highly limited, and therefore there are stringent requirements for low-power devices. Efficient power amplifiers (SSPA, TWTA) of RF signals are highly non-linear, and therefore suffer from parasitic amplitude modulation of the signal. The non-linearity of the amplifier additionally spreads the spectrum and it causes problems with meeting the requirements of regulating agencies (FCC, ITU) defined by the frequency masks [30].

The above limitations suggest using constant-envelope and bandwidth-efficient modulations. The following overview mathematically describes the selected modulations which will be used for the sampling analysis. In [31] there are defined two grades of baud-rates. The boundary line between both groups is 2 Msps for the bands below 8500 MHz and 10 Msps for the 31.8 – 32.3 GHz band. For the group of low symbol rates, modulations BPSK, QPSK or OQPSK shall be used, otherwise in new designs TCM 8PSK and GMSK shall be used.

Data for transmission is represented as a set of bits. Before modulation the bits are converted into state levels by pre-coding. The pre-coders perform several special codes such as NRZ-L, NRZ-M or SP-L. The description of the codes is beyond the scope of this work, but for more details see [31]. In the following, coding is assumed to be performed by a map-table as shown in (2.19) for BPSK modulation. Possible mapping for each modulation is summarized in the table. The propounded maps are not strictly set, but may vary among applications.

In addition, sometimes a differential coding is used for pre-coded data according to the data symbol

$$d_n = \text{bit\_map}\{0 \rightarrow -1; 1 \rightarrow 1\} \quad (2.19)$$

which is represented by the modulation symbol

$$a_n = d_n d_{n-1} \quad (2.20)$$

unlike the non-differential modulation symbol

$$a_n = d_n. \quad (2.21)$$

### 2.3.1. BPSK

A BPSK is a 2-state modulation where one symbol corresponds to one data bit as is clear from Table 2.1. The modulation can be expressed as [30]

$$f_{BPSK}(t) = m(t)\sqrt{P} \cos(2\pi f_c t + \varphi_c) \quad (2.22)$$

where  $P$  is the power of the carrier and  $f_c$  and  $\varphi_c$  are its frequency and phase, respectively.

The modulation function

$$m(t) = \sum_{n=-\infty}^{\infty} a_n p(t - nT_s) \quad (2.23)$$

consists of a pre-coded symbol (2.21) and a pulse function  $p(t)$ , which for a rectangular pulse has unity amplitude and duration  $T_s$ . The differential representation of BPSK modulation uses pre-coding from equation (2.20).

**Table 2.1:** Symbols of BPSK modulation.

Symbol	Bit code	Real (I)	Imaginary (Q)	Phase [rad]
1	0	-1	0	$\pi$
2	1	1	0	0

### 2.3.2. QPSK

QPSK is short for quadri-phase shift keying modulation. Mostly, the modulation is provided by two independent signal multiplications of the inphase and the quadrature modulation signals

$$m_I(t) = \sum_{n=-\infty}^{\infty} a_{In}p(t - nT_s), \quad m_Q(t) = \sum_{n=-\infty}^{\infty} a_{Qn}p(t - nT_s) \quad (2.24)$$

with two orthogonally shifted carriers from a local oscillator [30]

$$f_{QPSK}(t) = m_I(t)\sqrt{P} \cos(2\pi f_c t + \varphi_c) - m_Q(t)\sqrt{P} \sin(2\pi f_c t + \varphi_c). \quad (2.25)$$

According to Table 2.2, the serial stream of pre-coded symbols is parallelized into two (inphase and quadrature) streams. The differential representation DQPSK also uses the function (2.20).

**Table 2.2:** Symbols of QPSK modulation.

Symbol	Bit code	Real (I)	Imaginary (Q)	Phase [rad]
1	00	$-\sqrt{2}$	$-\sqrt{2}$	$-3\pi/4$
2	01	$\sqrt{2}$	$-\sqrt{2}$	$-\pi/4$
3	10	$-\sqrt{2}$	$\sqrt{2}$	$3\pi/4$
4	11	$\sqrt{2}$	$\sqrt{2}$	$\pi/4$

### 2.3.3. OQPSK

QPSK modulation changes both symbols at the same time, and then the change of phase by  $\pi$  rad may occur. Such a change is accompanied by a large fluctuation of the instantaneous amplitude. The offset (staggered) QPSK modulation solves the simultaneous occurrence of the changes by shifting the quadrature modulation function in time by a half of  $T_s$  [30]

$$m_I(t) = \sum_{n=-\infty}^{\infty} a_{In}p(t - nT_s), \quad (2.26)$$

$$m_Q(t) = \sum_{n=-\infty}^{\infty} a_{Qn}p\left(t - \left(n + \frac{1}{2}\right)T_s\right).$$

### 2.3.4. $\pi/4$ -QPSK

Another way how to deal with the parasitic amplitude modulation caused by the phase change of  $\pi$  rad is by using a  $\pi/4$ -QPSK modulation. It uses two alternating IQ diagrams, where the second one is phase rotated by  $\pi/4$  rad. The maximal change of phase is, unlike QPSK's  $\pi$  rad, only  $3\pi/4$  rad [30]. The possible expression of IQ modulation functions is shown by equations

$$\begin{aligned} m_I(t) &= \sum_{n=-\infty}^{\infty} \left( a_{I2n}p(t - nT_S) + \frac{a_{I(2n-1)} - a_{Q(2n-1)}}{2} p(t - (2n-1)T_S) \right), \\ m_Q(t) &= \sum_{n=-\infty}^{\infty} \left( a_{Q2n}p(t - nT_S) + \frac{a_{I(2n-1)} + a_{Q(2n-1)}}{2} p(t - (2n-1)T_S) \right). \end{aligned} \quad (2.27)$$

From Table 2.3 it is clearer that symbols have to be mapped alternately to the different set of symbols. The alternative values are enclosed in brackets.

**Table 2.3:** Symbols of  $\pi/4$ -QPSK modulation.

Symbol	Bit code	Real (I)	Imaginary (Q)	Phase [rad]
1	00	$-\sqrt{2}$ (-1)	$-\sqrt{2}$ (-1)	$-3\pi/4$ ( $-\pi/2$ )
2	01	$\sqrt{2}$ (1)	$-\sqrt{2}$ (-1)	$-\pi/4$ (0)
3	10	$-\sqrt{2}$ (-1)	$\sqrt{2}$ (1)	$3\pi/4$ ( $\pi$ )
4	11	$\sqrt{2}$ (1)	$\sqrt{2}$ (1)	$\pi/4$ ( $\pi/2$ )

### 2.3.5. Filtered OQPSK

The filtered OQPSK modulation may be used for high-speed communication. The modulation scheme is the same as for the OQPSK modulation above, but the modulated signal is filtered after the modulation. The most common filtration type is the square root raised cosine filter (SRRC) of transfer function [31]

$$H(f) = \begin{cases} 1, & |f| < f_N(1 - \alpha) \\ \sqrt{\frac{1}{2} + \frac{1}{2} \sin\left(\frac{\pi}{2f_N} \left(\frac{f_N - |f|}{\alpha}\right)\right)}, & f_N(1 - \alpha) \leq |f| \leq f_N(1 + \alpha) \\ 0, & |f| > f_N(1 + \alpha). \end{cases} \quad (2.28)$$

The characteristic is defined by a cut-off frequency  $f_N$  and a roll-off factor  $\alpha$ , which defines the sharpness of the filter. When the roll-off factor goes to zero, the characteristic collapses to a rectangular function.

The filtration of such characteristic has to be applied on both transmission and reception side and then the multiplication eliminates the square root. In cases of very high data rates, where the SRRC filtration is difficult to use, a post amplifier shaping of different frequency characteristics may be used [31].



### 2.3.6. GMSK

The Gaussian frequency shift keying modulation is of the class of continuous phase frequency modulations (CPFM), particularly of the sub-class of a partial-response CPFM. The modulation scheme is obtained by filtering of rectangular frequency pulses of the MSK by the Gaussian impulse response filter [30].

In [31] the GMSK modulation is defined by the equations

$$f_{GMSK}(t) = \sqrt{P} \cos(2\pi f_c t + \varphi(t, a) + \varphi_c), \quad (2.29)$$

$$\varphi(t, a) = \frac{\pi}{2T_s} \sum_n a_n \int_{-\infty}^{t-nT_s} g(\tau) d\tau, \quad (2.30)$$

$$g(t) = Q\left(\frac{2\pi BT_s}{\sqrt{\ln(2)}}\left(\frac{t}{T_s} - 1\right)\right) - Q\left(\frac{2\pi BT_s}{\sqrt{\ln(2)}}\frac{t}{T_s}\right), \quad (2.31)$$

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{y^2}{2}} dy. \quad (2.32)$$

The carrier phase of the signal (2.29) is modulated by the function  $\varphi(t, a)$  (2.30), which is the sum of MSK pulses (2.31). Each MSK pulse is obtained as a combination of the data symbol and the integral of Gaussian probability function (2.32). The data symbol is differentially pre-coded by (2.20).

The  $BT_s$  parameter in (2.31) is selected as a compromise between spectral efficiency and bit error ratio performance, and as mentioned in [31], a value of 0.5 shall be used. The  $BT_s$  parameter also defines the truncation length of the shaping function (2.31) [30].

## 2.4. Downconversion

The role of the downconverter in radio frequency (RF) systems, generally, is to translate the frequency band of interest down to the baseband, to filter that relatively narrowband signal and to reduce its sampling rate in digital systems [32], [33]. As a part of the software defined radio the digital downconverter (DDC) may be implemented as the firmware in DSP and more often in FPGA [34]. The implementation of the DDC is performed by several algorithms with different properties dependent on the target application. Generally, the approaches are of two types: undersampling and mixer based downconversion.

The undersampling based downconversion, also called the harmonic subsampling or the integer-band decimation [35], is in [36] described as sampling of a signal at a sampling rate much lower than one quarter of the Nyquist rate. The signal is aliased to the baseband or the first Nyquist zone, thus the technique effectively performs the frequency downconversion and the required sampling frequency reduction. The process of aliasing causes spectral reversal of alternate spectral zones and this needs to be taken into account when designing the subsequent baseband signal processing. Since the downsampling is joined with a frequency shift, avoidance of the destructive aliasing of adjacent channels to the baseband has to be performed first by complex bandpass

filtering. The intermediate frequency (IF), which is the object of the DDC, is usually fixed, but if not, the filter needs to be tunable [35]. The IF is usually chosen to be an integer multiple of a certain fraction of the sampling frequency. One special case is when the  $f_{IF}$  is selected as an integer multiple of  $f_s/4$ , then the IF band is downconverted right to the baseband. Undersampling is purely a sampling process, so any mathematical operation is needed; thus, the technique provides very short latency times, which is useful in feedback applications. The drawback of discrete frequency shifts can be neglected in systems where the sampling frequency may be locked to the IF [34].

The second approach uses a mixer to generate the sum and difference spectral components by multiplication of the input signal and the internally generated carrier replica. The input signal is generally real, but the mixer is more often quadrature producing the complex output shifted only one way by a complex exponential. The quadrature mixer is also common in analog downconverters consisting of separate mixing with sine and cosine signals [35]. In digital systems the carrier replica is generated in a numerically controlled oscillator (NCO) which has the advantage of producing sine and cosine values perfectly synchronized, because they are derived from a common clock, and with a precise  $90^\circ$  phase shift [34]. The typical NCO requirements are ultra-low phase noise and high precision tunability, which depend on the reference clock precision, the large frequency range and the suppressing harmonics [37]. The level of spurious harmonics produced by the NCO are crucial to the spectral purity of the output signal because they are mixed and translated to the band of interest and reduce the spurious free dynamic range (SFDR) [32]. The typical requirements for SFDR vary from around 60 dB to over 100 dB [33]. Compared to undersampling, the mixer based downconverters provide better signal to noise performance and are suitable for large varying IF, but have reasonable latency times. The DDC allows extremely fast phase continuous hopping without over or under-shot and without loop setting anomalies [34]; this allows using the DDC in systems for very fast frequency switching and are also effective for GMSK modulations [38].

Carrier signal generation is an individual task. The straightforward approach is to pre-compute values of sine and cosine for every possible phase of one period and save it into the memory RAM or ROM. Then the counter generates the samples of continuously increasing phase similarly to the time flow; it is known as an accumulator. The accumulator points to the memory and the corresponding value of the signal is read out, therefore the memory in which the signal values are stored is usually called the look up table (LUT) [38]. Frequency tuning is provided by changing the phase increment of the accumulator and some values of the LUT can be periodically skipped. It is obvious that the LUT has to contain the signal samples of one period of the lowest frequency in the generator tuning range. If we assume storing only the sine wave, because the cosine is only  $90^\circ$  shifted, the size of the LUT in bits is

$$2^n \cdot m \quad (2.33)$$

where  $n$  is the bit resolution of the accumulator and  $m$  is the bit depth of the LUT [35]. For instance, for a 32-bit phase accumulator at 8-bit signal resolution, the LUT size is 4 GB. Such a large memory implementation is undesired due to its chip area hungriness and consequent power consumption and cost [35]. We can take advantage of the sine symmetry and modify the phase accumulator in such a way that we store the values only up to the phase  $\pi/2$  which quarters the needed memory [38], [33]. More savings are achieved by phase truncation, where the phase accumulator resolution stays the same,

but only upper bits are used for LUT translation. It naturally leads to errors in the phase-to-amplitude conversion, which are periodic and appears as line spectra known as phase truncation spurs, consequently the SFDR is worsened [34]. The degree of phase quantization leads to phase errors and quantization of the LUT causes amplitude errors. The spurs caused by periodical phase errors are effectively managed by phase dithering, which can improve the SFDR by 12 dB without increasing the LUT bit depth [33]. Such a dithered architecture may be multiplier free, but still consumes many memory blocks in the FPGA for high SFDR designs. The Taylor corrected method can significantly improve the SFDR up to 120 dB [33]. The LUT based oscillators are typically used in the direct digital synthesizers (DDS), where the NCO values are converted by the DAC to the analog domain. A typical DAC performs a zero hold function and produces a staircase output with sinc spectrum, that is why the additional sinc pre-compensation is needed [34].

Another often utilized technique employs the coordinate rotation digital computer (CORDIC) as a LUT substitution. CORDIC has originally been used for the conversion between Cartesian and polar coordinates and further for calculation of hyperbolic functions, multiplication, division, exponentials and logarithms, and it makes CORDIC suitable for the implementation of the direct digital frequency synthesizer (DDFS) [35], [38]. The algorithm performs rotation by means of elementary rotations of decreasing size; the value of the signal function is found iteratively. The basic implementation uses few values of tangent  $\Delta\varphi$  and scaling factor  $1/\cos(\Delta\varphi)$  stored in the LUT and 4 multipliers while there are special multiplier free cases employing only a logical shift and add and subtract operations. Setting  $\tan(\Delta\varphi) = \left(\frac{1}{2}\right)^i$  the multiplication in all iterations may be performed by the logical shift; successively decreasing elementary rotations are then defined by  $\Delta\varphi_i = \pm \text{atan}(2^i)$  with  $\pm \frac{\pi}{2}$  uncertainty which must be managed before the first iteration [35]. CORDIC is suitable for HW mapping, a CORDIC element – 1 iteration – is placed in parallel to the next in a cascaded structure, even the pipelining is regular [38]. The least mean square (LMS) error from the true value of the sine wave fast converges to zero and it is very low for more than 16 iterations, which is a good choice [38]. For 16-bit precision, only 14 iterations are sufficient [39]. The overall hardware effort of the optimized CORDIC is approximately the same as implementation of 3 multipliers, so it can be stated that CORDIC saves 1 multiplier compared to the non-optimized version [35]. The downconversion by CORDIC can be provided directly by continuously rotating the complex phase of the input signal instead of generating the carrier by the NCO (CORDIC) and mixing with the input signal. More savings of the HW demands may be achieved by approximations of the  $\text{atan}(2^i)$  function, which for  $i > 4$  is sometimes linearized, and thus it is necessary to have the LUT of size 4 for any precision [38]. Generally, for high speed ADCs, the sine and cosine function is needed to be calculated in one cycle so the cascaded and pipelined versions are suitable. To provide chip area savings, folded and time shared iterative CORDIC algorithms are more convenient. The folded CORDIC folds the number of cascades and the rest stays in parallel; the degree of folding is defined by the so-called folding factor. Additionally the folded CORDIC, especially the C slowed folded CORDIC, may also be pipelined. The CORDIC NCO and downconverters are typical for its spectral purity and high SFDR [38].

The last no less important group of NCOs is the IIR or recursive oscillators. A large LUT is avoided by a filter with infinite impulse response (IIR) with a transfer function that has a complex or conjugate complex pole on the unit circle [35]. More

generally, the recursive oscillators have been designed via old trigonometric formulas in a recursive form. A sum-and-difference-of-angles formula in recursive form

$$\cos(\varphi + \theta) = 2 \cos(\theta) \cos(\varphi) - \cos(\varphi - \theta) \quad (2.34)$$

is usually referred as the biquad or the direct form implementation. The formula is used for the iterative calculation of two sinusoids with an angle step  $\theta$  and a phase difference  $\varphi$ . If the formula is rewritten as a pair of trigonometric formulas

$$\begin{aligned} \cos(\varphi + \theta) &= \cos(\varphi) \cos(\theta) - \sin(\varphi) \sin(\theta), \sin(\varphi + \theta) \\ &= \cos(\varphi) \sin(\theta) + \sin(\varphi) \cos(\theta) \end{aligned} \quad (2.35)$$

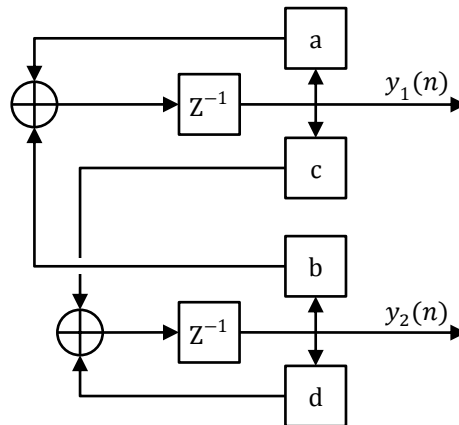
it is referred as a coupled form, since each equation uses its past value and a value produced by other equation. The general form that expresses both oscillators is a 2-by-2 rotation matrix and two state variables

$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (2.36)$$

where the left vector is the new iteration sample and the right vector is the past one. To keep the oscillations there are two Barkhausen criteria

$$\begin{aligned} ad - bc &= 1, \\ |a + d| &< 2. \end{aligned} \quad (2.37)$$

From the analysis of eigenvalues the oscillator behavior can be evaluated directly from the rotation matrix. If  $a = d$  the difference angle  $\varphi$  must be  $\pm 90^\circ$  and the oscillator will produce quadrature output. Similarly, if  $b = -c$  the outputs will have equal amplitudes. The generic oscillator iteration signal diagram is shown in Figure 2.2.



**Figure 2.2:** Signal diagram of the generic oscillator [40].

There are many versions of the recursive oscillators, few of them are presented with its characteristic parameters in Table 2.4.

The recursive oscillators are loaded with some values and allowed to free run. If the run time is short, the amplitude errors may be tolerable, but the errors accumulate and after longer time they may reach an unacceptable value. A standard approach, dealing with the amplitude error, uses amplitude gain control (AGC), where the power of the sinusoids are estimated rather than the amplitude to avoid square roots. The gain correction factor is calculated every iteration and is used to scale the state variables of

the oscillator. This reasonably brings complexity to the algorithm and increases processing effort. The frequency control of the recursive oscillators often constraints the constant amplitude range and it has to be corrected by additional dependencies of the AGC. [40]

**Table 2.4:** Characteristic parameters of the recursive oscillators [40].

Oscillator	Multiplies per iteration	Equi-amplitude	Quadrature output	k	Rotation matrix
Biquad	1	Yes	No	$2 \cos(\theta)$	$\begin{bmatrix} k & -1 \\ 1 & 0 \end{bmatrix}$
Digital waveguide	1	No	Yes	$\cos(\theta)$	$\begin{bmatrix} k & k-1 \\ k+1 & k \end{bmatrix}$
Equi-amplitude staggered update	2	Yes	No	$2 \sin\left(\frac{\theta}{2}\right)$	$\begin{bmatrix} 1-k^2 & k \\ -k & 1 \end{bmatrix}$
Quadrature staggered update	2	No	Yes	$\cos(\theta)$	$\begin{bmatrix} k & 1-k^2 \\ -1 & k \end{bmatrix}$
Coupled standard quadrature	4	Yes	Yes	$\sin(\theta)$	$\begin{bmatrix} \sqrt{1-k^2} & k \\ -k & \sqrt{1-k^2} \end{bmatrix}$

The mixer or other downconverter is followed by the receiving filter. Its role is to reduce the combined effect of noise and inter-symbol interferences (ISI) [41], it serves as an anti-aliasing filter and a channel selector [33]. Generally it is a low-pass or band-pass filter sufficiently suppressing undesired signals and with an impulse response meeting the condition  $g(mT_s) = 0$  for  $m \in \mathbb{Z} - 0$  to reduce ISI. The bandwidth of the received signal is reduced by filtration, thus there is no need to keep a high sample rate. The signal is therefore usually downsampled to decrease the processing effort. The downsampler must not distort the signal of interest, but there is no reason to not distort other channels, e.g. by aliasing. The degree of downsampling is defined by the oversampling ratio

$$OSR = \frac{f_s}{B} \quad (2.38)$$

which is the ratio of the sampling frequency and the bandwidth of interest. Filtration combined with downsampling can be decomposed in cascade to perform successive filtration and downsampling; so-called multirate filtering. High OSR allows a wide transition band and therefore leads to a smaller number of filter coefficients. For multirate filtering, the OSR of the first stages is high, the transition band is wide and the stop-band may be narrow, thus the CIC filters are suitable for these stages. In the last stages OSR is low and the FIR filters with desired transfer function are used here. Such FIR filters are of high order, but are carried on a low sample rate, thus processing effort is kept low. [35] The narrowband signals are those, where the bandwidth of interest is significantly lower than the sampled bandwidth, typically it is when the sample rate change  $\geq 32$  [33], [34].

Two main groups of digital filters are used for receiver filtering, the finite impulse response (FIR) filters and the infinite impulse response (IIR) filters. The FIR filters are the most popular because of exactly linear phase of their transfer function and their unconditional stability [32]. A big drawback is the need for high order to reach a good filter comparable to the IIR [35]. The IIR's less order leads to a lower number of

computations and consequently lower processing load and lower group delay. The linear phase of the filters is important especially when using signals carrying information in phase. [32] The narrowband IIR filters, where it is easy to control the phase characteristic, are commonly unstable, and therefore using them is generally avoided. [35]

For multirate processing, half-band filters are typical. From its name it is obvious that the filter passes the lower half of the sampled spectrum and attenuates the upper band. The signal is then decimated by 2. The filters are mostly realized as FIR filters and are very popular because they need only half the number of coefficients. Every second sample is ignored and thus easily provides downsampling by 2. [32]

In 1981, Eugene Hogenauer presented a special multiplier free low pass filter with a linear phase composed of a cascaded integrator and a comb so-called CIC filter [41]. The multiplication is very complex and power hungry to implement in silicon [32]. CIC filters are mainly combined with a decimator included between the integrator and the comb providing downsampling and simultaneously reducing comb processing demands. The number of cascaded integrator and comb pairs defines the CIC filter order; with each additional stage the stopband attenuation is increased. The filter's passband lobe provides an undesired passband droop and aliasing after the decimation is worse towards the lobe's edges [34]. The rule of thumb is that the band of interest should never be wider than 25% of the main lobe span, consequently it generally performs sufficiently for decimating down to four the Nyquist rate, i.e. the OSR after decimation should never be less than 4 [33], [35]. CIC filters are suitable for the first stages of multirate filtering for high speed sample rate reduction and should be used with low sample rate droop compensation filters, half band filters and matched filters in cascade to provide a high quality baseband signal [35]. There are also application specific efficient versions of the CIC filters using polynomial factoring and polyphase decomposition for reduction of data word growth, where the recursive structure is split into more parallel non-recursive sections. [40]

We assumed low pass filtering of an already downconverted signal so far, but it is possible to interchange these operations while using the bandpass filters instead of the lowpass filters. The impact on the signal is the same, but the realization aspects are much different. Firstly, the filter must be complex and thus it has two impulse responses, and demands twice the processing effort. The downconversion of the filtered signal results in a baseband signal and also one with double the carrier frequency, which must be removed by filtering, so the low-pass filtering cannot be fully omitted on the baseband side. This drawback usually disqualifies band-pass filtering from being used as a receiving filter. [35]

Filter realization on FPGAs is easily possible by using multiply-and-accumulate (MAC) operations on the DSP slices, which often also includes several rounding modes. For instance, the XtremeDSP slices allow up to 18-bit signed coefficients and the Virtex 5 DSP48 slice can accommodate multiplication of 25-bit by 18-bit 2's complement word with full precision and a 48-bit result. [33]

## **2.5. Carrier synchronization**

Carrier synchronization is an essential task in many radio receiving systems. Due to the residual frequency error, the complex baseband signal is slowly rotating at an angular frequency equal to the frequency difference between the transmitter and

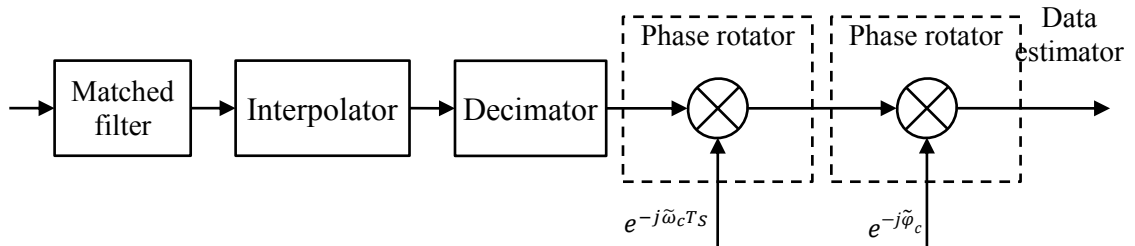
receiver oscillators. It is possible to build extremely accurate clocks, but there always exists a small frequency difference between two separate and free running oscillators. Even the slowest rotation of the baseband signal will cause detection errors. The same applies to timing synchronization, where the slightest difference will cause cycle slips in the long run. [41]

The need for a carrier recovery system may be overcome, usually by differential coding. It allows correct data detection if the rotation is relatively slow, i.e. the frequency error is slight, and obviously the frequency error contributes to BER degradation. Such an approach is known as non-coherent detection and, unlike the coherent systems, it has worse characteristics for low SNR signals. Coherent detection requires a carrier signal replica derived directly from the received signal and it significantly increases the system complexity, but it leads to the highest spectral efficiency due to no power penalty. The timing recovery requirement is common for both coherent and non-coherent systems. [43], [41]

The ultimate goal of the receiver is to detect the symbol sequence in the received signal disturbed by noise with a minimal probability of detection error. The received signal parameters that are generally unknown and thus are of concern in the estimation within the receiver system are:

- $\omega_c$  – carrier frequency
- $\varphi_c$  – carrier phase
- $\tau$  – symbol timing offset
- $c_k$  - symbol

An optimal receiver maximizes the a posteriori probability (MAP) for which it needs to know the a priori probability distribution and the distribution of the received signal conditioned on the knowledge of the data sequence. The exact distribution of the data symbols cannot be a priori known since it carries useful information, but the prerequisite can be simplified by the assumption of equally probable data sequences and converting MAP to the maximizing of the likelihood function (ML). In an optimal receiver there exists no separate estimation unit for each parameter, all must be estimated jointly. On the other hand, such a system is still far too complex for practical applications and it is often needed to accept a compromise and average the likelihood function over unwanted parameters. [41]



**Figure 2.3:** Carrier synchronization of the coherent receiver signal diagram [41].

A possible signal flow within the coherent receiver is depicted in Figure 2.3 and comprises of the digital matched filter, the time variant interpolator, the decimator, the phase rotation unit and the data sequence estimator. The frequency and the phase errors are corrected by counter-rotating the input signal in a mixer by the frequency and phase estimate. The phase estimation is usually performed after the matched filter, but the

optimum filtering for phase estimation may coincide with optimum filtering for data detection. In any bandwidth efficient transmission the power is concentrated within a constrained bandwidth while only a negligible amount of the power is located outside, thus the signal may be decimated to relax the processing effort. Unlike the phase estimation, which is usually performed at rates near to the symbol rate, the frequency estimation must be computed at a higher rate and with a wider matched filter since the frequency offset can move the useful signal out of the processing range. [41], [44]

The bandwidth reducing filters, typically the matched filters, reduce the noise coming in on the undesired frequencies, but consequently reduce the range of influence of the frequency estimator. This is even more typical for satellite communication, where the data rates are low and the signals are shifted by the Doppler effect especially at high carrier frequencies. In modern receivers, phase recovery is often performed after timing recovery due to its nearly optimal performance; this is the opposite order as found in classic analog receivers. The timing can be estimated even in the presence of a residual frequency offset, typically the used structures are useful for the range

$$\frac{|\Delta\omega_c T_s|}{2\pi} \leq 0.15 \quad (2.39)$$

where  $\Delta\omega_c$  is the frequency difference between the estimate and the actual carrier frequency and  $T_s$  is a symbol period. For larger frequency offsets, a two stage frequency estimator is typically used. The first step is coarse frequency error compensation to fit the signal within the range (2.39), and after timing recovery the residual frequency offset is adjusted by the phase synchronizer. The phase synchronizer can be easily designed to be able to compensate for varying phase, i.e. for the residual frequency offset. This strategy differs from the optimal MAP receiver, because the coarse frequency estimator must work independently on the values of other parameters. [41]

The categorization of the synchronizers according to the way, how the estimate is extracted from the received signal, is as follows:

- open-loop (OL) or feedforward (FF)
- closed-loop (CL) or feedback (FB)

The OL estimators estimate the unknown parameter directly. The systems usually employ parallel search processing or iterative search processing for likelihood function's maximum search. The systems are usually more complex and resource demanding, but are able to release the estimate after a relatively low number of processed symbols and thus are suitable for burst mode communication typical for TDMA systems. [41], [44]

CL estimators use the differentiated objective function to produce an error signal used to predict the new estimate. The error signal is filtered to smooth the noise and is used to correct the actual value of the estimation; this fact enables the estimator to automatically track slowly varying parameter changes. Due to its longer setting times, CL estimators are rather suitable for continuous mode transmission. [41], [44]

To be able to automatically provide the correct carrier recovery, the estimators have to perform two modes of operation as is also typical for phase locked loops (PLL):

- acquisition
- tracking



The acquisition mode is the state of bringing the system from its initial state to the tracking mode. The frequency estimators use several methods employing non-linear processes, widening the frequency range of operation and sweeping over the defined frequency range. The tracking mode is in operation when the estimation error is sufficiently small. The acquisition and tracking operation of the CL estimators are understandable when disconnecting the VCO from the mixer and driving the latter at a fixed frequency. Sweeping the offset frequency at the input gives an error generator response at its output; the response is dubbed as an S-curve due to its similarity to an “S” rotated by 90°. The S-curve is zero at the origin and extends firstly linearly and then after its limiting range vanishes to zero. Obviously, within the non-zero range, the estimator is able to make an effort to correct the frequency offset and eventually lock on to the incoming carrier; this is referred as the acquisition range. The S-curve takes significant values in the order of  $\pm 10$  to  $\pm 15\%$  of the symbol rate. The region around the origin can be linearized

$$S(\Delta\omega_c) \approx A\Delta\omega_c \quad (2.40)$$

and is referred as a tracking range. [41], [44]

The noise coming through the estimator causes the noise in the estimate. The measure of the noise amount is its variance. The variance can be made as small as desired by reducing the estimator pre-filter bandwidth, but it results in very long acquisition time, so a trade-off is needed between the acquisition length and the tracking performance. There exists a theoretical lower bound on the variance of the parameter estimate, a Cramer-Rao lower bound (CRB) or a Modified Cramer-Rao lower (MCRB) bound. Any estimator has variance lower than this bound. Details about the CRB and the MCRB are discussed in [44] and [45]. The practical applications of the estimators may exhibit the floor in the variance measure as the SNR increases, such a phenomena is called the self-noise and causes irreducible degradation of the estimate independently on the thermal noise component. The practical applications of the estimates are calculated over the finite number of the symbols which is in contrast to the ideal ML approach; such a simplification naturally must bring some degradation on the estimate, and this is the self-noise. The self-noise is also related to any averaging performed in the estimator design. [41], [44]

Besides the open and closed loop, the frequency estimators are categorized according to the information available or, in other words, according to the degree of averaging employed. There are three main groups:

- data aided (DA)
- decision directed (DD)
- non-data aided (NDA)

NDA estimators can be additionally divided depending on the timing information availability. DA and DD estimators inherently suppose timing synchronization available from the data source, and therefore are mostly useful for the correction of offsets much smaller than the symbol rate where receivers are operating in steady state conditions. The coarse frequency estimators have to deal with offsets on the order of the symbol rate and the timing is perhaps also unknown and so NDA estimators are employed there.

DA estimators suppose that the data sequence and its position within the received signal are completely known during the estimation process and the frequency offset is a small fraction of the symbol rate. TDMA systems transmit data in bursts; each burst

contains a preamble with the so-called unique word (UW) to identify the frame. The UW may be prepended with an unmodulated sequence or the sequence modulated with a signal which exhibits some kind of periodicity, this part is used for fast carrier synchronization, where data dependency is simple and easily resolved. Such a preamble obviously leads to channel transmission efficiency reduction. The demands can be relaxed by using a DA estimator during UW transmission, where the data dependency is a priori known. The preamble which is transmitted only a small fraction of the transmission time, but periodically, suffices to the correct carrier parameter estimation since the frequency varies slowly and remains constant over many bursts. DA frequency estimators require timing synchronization which is also performed during UW transmission. The differential encoding of the UW easily allows frequency estimation and frame and timing detection to be performed simultaneously.

The variations of the synchronization parameters are usually slow compared with its symbol period; this allows using an already decoded data sequence to update the estimate. Using the decoded data sequence for estimation is known as the DD estimation. Obviously all the DD require an initial estimate to make decoded data available, it is usually performed in NDA estimators. The DD is also simpler to implement with differential PSK modulations, where the modulation can be removed using differential decisions which is feasible even in the presence of moderate frequency offsets. [41], [44]

There are many algorithms besides the ML based ones found in [41] and [44] used for the DA and DD frequency estimation. Messerschmitt's is a rotational frequency class estimator [46], a cross product detector published by Natali [47] and others for the estimation of a single frequency in a noisy environment, Classen [48], Tretter [49] or Kay [50]. Also, Fitz [51] and Louise and Reggiannini [52] proposed their methods. The methods are more or less successful and each method provide different operating ranges and different self-noise. Excluding Kay, all these estimators are close to the MCRB for  $E_b/N_0$  values down to 0 dB. True DA ML estimators are not used due to high complexity, that is why the above mentioned simpler methods are resorted to. As is expected, the simplification worsens the performance compared to the optimal ML estimator, but in a selected region the performance of the algorithms is surprisingly close.

From the above mentioned, it is clear that satellite communication suffers from large frequency offsets that must be corrected in two stages. The first stage additionally must release the frequency estimate independently on the other unknown and therefore unwanted parameters. The scale of observation of the estimated parameters is much larger than the scale necessary to express symbol duration, so the only practical possibility is to employ NDA estimators for this. The basis of NDA frequency estimators is the ML function. The systematic derivation of the ML synchronizer is straightforward, the likelihood function must be averaged over the unwanted parameters. Averaging is unfortunately possible to be performed in a closed form in only a few isolated cases, otherwise the likelihood function must be simplified by approximation. This changes the understanding of derivation of the new synchronization algorithm as a task of finding suitable approximations. Common to all NDA frequency estimators is their poor performance for low and medium SNR values, because due to the non-linear operations required to remove modulation, noise is strongly increased.

The OL ML based estimator for M-PSK modulations is defined by

$$\tilde{\omega}_c = \frac{1}{2\pi MT} \arg \left( \sum_{k=1}^{L_0-1} (y(k)y^*(k-1))^M \right) \quad (2.41)$$

where  $y$  is the input signal and  $M$  is the order of the modulation [44]. The estimate is calculated over  $L_0$  symbols and the modulation is removed by the  $M$ -th power. Typical  $M$ -th power estimators are the squaring estimators for BPSK and the quadrupling estimator for QPSK [43], but removing data dependency by power operation is not recommended for higher order modulation schemes [41] and is difficult to implement at high frequencies [43]. Better performance is given by the well-known Costas loop, which is directly based on the ML phase estimator. It can be used with the timing aid or the integrators can be substituted by the low pass arm filters. How to do that and how it affects the Costas loop performance is discussed in [1]. The Costas loop was designed for BPSK modulation, but its modified version for higher order modulations is usually used, especially QPSK [43]. The appearance of the  $M$ -th power estimator and the Costas loop may seem different, but the performance is the same since the  $M$ -th power is also accommodated within the Costas loop [53]. It is worth mentioning that since modulation is removed by the non-linear operation with no data aid, it suffers from phase ambiguity, which must be eliminated differently, e.g. employing differential coding.

The estimator in (2.41) modified for no timing information as follows

$$\tilde{\omega}_c = \frac{1}{2\pi\Delta T} \arg \left\{ \int_0^{T_0} y(t)y^*(t - \Delta T)dt \right\} \quad (2.42)$$

where  $\Delta T$  is a design parameter, is known as a delay and multiply frequency estimator. The weakness of the delay and multiply estimator and all NDA estimators is the sensitivity to adjacent channel interference. [44]

The foregoing open loop estimators need to perform the maximum search function, which is highly non-linear. The phase increment method [41] similar to Messerschmitt's DA estimator gives the frequency estimate directly. The method can be extended as a DD estimator by measuring phase rotations between the consecutive samples [44]. Unfortunately, these methods exhibit high estimate variance for low SNR values. In [41] there was also presented a method for frequency estimation via spectrum analysis. The spectrum, usually computed by the FFT, is segmented and each segment spectral density is analyzed. The estimation is calculated via the equation

$$\tilde{\omega}_j = \arg \left( \max_j \sum_{m=-\left(\frac{N_{FFT}}{2}-1\right)}^{\frac{N_{FFT}}{2}} |R_{Ts}(m\Delta f)|^2 G(m\Delta f - (j\Delta f))^2 \right) \quad (2.43)$$

where  $R$  is the autocorrelation function of the input signal and  $G$  is the power spectral density of the matched filter. The resolution of the FFT obviously determines the estimation accuracy. To obtain a reliable estimate the spectrum must be averaged, which, however, decreases the self-noise, but prolongs the estimation time on the contrary.

The first very often used type of NDA closed loop frequency estimator independent of timing information is derived from the ML function employing

averaging approximation by the Taylor series. The error signal is produced by a simplified derivative of the approximated log-likelihood function

$$e(t) = \frac{d\Lambda(r|\tilde{\omega}_c)}{d\tilde{\omega}_c} = 2 \int_0^{T_0} \text{Im}\{x(t)y^*(t)\}dt \quad (2.44)$$

where  $r$  stands for the input signal and  $x$  and  $y$  are the responses of two matched filters. The responses of the filters are

$$H_1(f) = G(f) \rightarrow h_1(t) = g(t) \quad (2.45)$$

which is identical to the pulse matched filter with an impulse response  $g$  and

$$H_2(f) = j \frac{dG^*(f)}{df} \rightarrow h_2(t) = 2\pi t g(-t) \quad (2.46)$$

called a derivative matched filter (DMF) or a frequency matched filter. The frequency matched filter can be approximated by a set of simple filters which perform the differential power measurement. This results in algorithms known as a dual filter or a mirror image detector (DFD) and a quadricorrelator developed ad hoc. Interestingly, the ad hoc developed estimators are close or even equivalent to the ML based ones. However, the distance from the MCRB is still huge, and it can be eliminated only by given additional signal information, i.e. by applying DD and DA estimators.

### 3. Objectives of the thesis

It can be concluded that SDR is very useful in modern satellite communication including DSN, so the discussion, whether to focus on digital or analog signal processing for autonomous communication payloads, can be considered as closed. The discussion on downconversion and carrier synchronization techniques showed that there is a huge amount of possible solutions. Indeed, this has been a subject of deep theoretical and practical research for decades and many publications are available. Maybe this is the reason why one can find it as depleted, but this is definitely not true and will never be.

Our interests focus on the satellite communication especially at very long distances, where signal power is lost so much that noise power becomes dominant. The satellite communication must cope with large Doppler shifts of the signal frequency. The way to deal with low signal to noise ratios is slowing the communication and narrowing the occupied frequency band; the significant noise power is then usually removed by filtering. By contrast, narrowband filters makes carrier synchronization very difficult.

The objectives of the dissertation thesis are formulated as follows:

- Improve digital downconversion based on undersampling by using Fourier transform algorithms. The downconverter is supposed to be a part of a multi-rate downconversion.

The implementation of a simple mixer is straightforward and provides excellent results, but it requires to be driven by the NCO. It was presented that the implementing NCO can be a daunting task which finally makes the downconverter require many processing resources. On the contrary, the undersampling technique seems to be very effective especially for multirate downconversion. The problematic bandpass filtering would be provided by employing Fourier transform algorithms.

- Investigate the possibility of frequency synchronization based on the spectral analysis of the received signal. Focus on the narrowband signal with poor SNR and large frequency offsets.

Carrier synchronization of signals of such properties has been said to be difficult. The acquisition range of the known algorithms is too narrow to accommodate the Doppler shifts typical for satellite communication without systematically sweeping the frequency over the range. A special approach to spectrum averaging would give attractive results.

- Develop simulation models of the investigated algorithms and provide the performance analyses.

It is necessary to compare developed algorithms with existing algorithms from a performance point of view. Since the focus is aimed at digital processing, the models in MATLAB Simulink are nearly equivalent to real

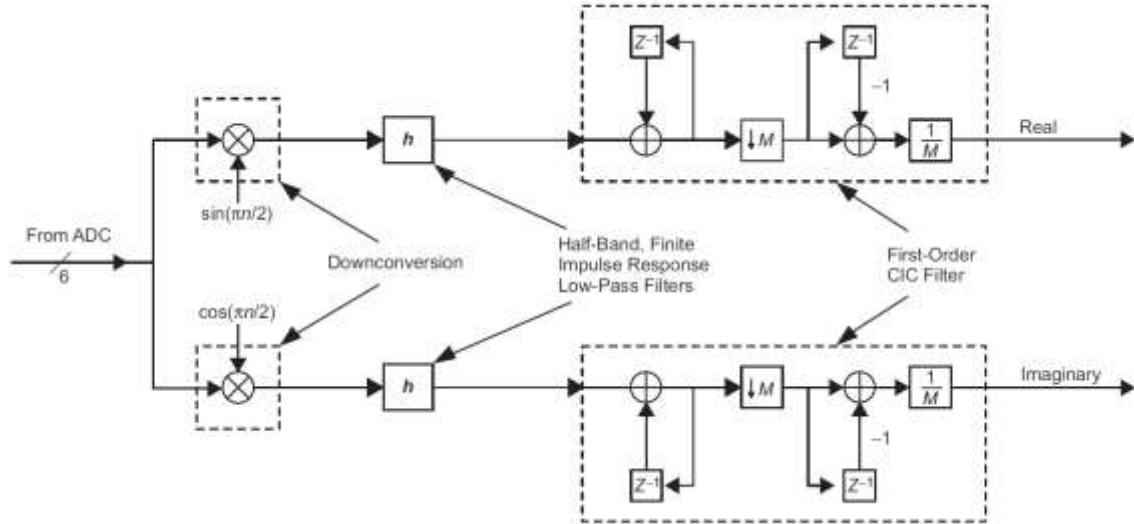
implementation as firmware on DSP or the configuration data for FPGA and the simulation with real input data is informative enough.

## 4. Downconversion of narrowband signals

In digital receivers the received signals are digitalized after analog processing (amplification, analog downconversion, filtering). Demodulation is then provided by a digital processing unit as mathematical computation. The desired signal is mostly several times oversampled and data is carried at a low intermediate frequency. Thus, the signal has to be digitally downconverted to the baseband.

### 4.1. Digital downconversion of PM signals

The downconversion of phase-modulated signals from a low intermediate frequency is mostly provided by the concept shown in Figure 4.1. Particularly, the concept is identical to that used in the Electra SDR [1].



**Figure 4.1:** Digital downconverter in Electra SDR [1].

The signal, as a set of samples from the ADC, is divided into two branches by multiplication with a complex harmonic signal. The complex harmonic signal consists of a sine wave for an inphase branch and a cosine wave for a quadrature branch. Both waves are generated according to the parameters derived from the received signal (carrier frequency and phase) by a carrier recovery and tracking block.

The resulted signal shifted in frequency contains the desired modulated signal newly located in the zero-IF carrier, i.e. the baseband, but is still excessively oversampled. In addition, the band above the frequencies of the desired signal is assumed to be noise, and therefore it is appropriate to be removed. The following blocks provide such filtration and decimation to remove the redundant samples. The concept uses a combination of a half-band filter (low-pass) and a CIC-decimator (Cascaded Integrator-Comb), which is an ideal structure for FPGA implementation, due to the absence of the multiplication operation.

Neglecting multiplicative noise caused by the wave generator of which amount depends only on the implementation parameters, the multiplication blocks do not change the noise performance of the signal. Assuming using ideal filters (rectangular shape – infinite *sinc* impulse response) and distortion by white Gaussian noise, the signal performance is improved with respect to equations

$$\frac{SNR_{out}}{SNR_{in}} = M \quad (4.1)$$

and

$$M = \frac{B_{full}}{B_{pass}} = \frac{f_s}{f_c} \quad (4.2)$$

Where  $M$  is a decimation factor, i.e. the absolute ratio between a pass-band of bandwidth  $B_{pass}$  and a full-band of bandwidth  $B_{full}$  of the filter power spectral density characteristic (PSD). The decimation factor of an ideal filter may be expressed by equation (4.2). The relation is equivalently expressed by a sampling frequency  $f_s$  and a filter cut-off frequency  $f_c$ .

An ideal filter is suitable only for the first approach to the system development. The calculation of noise performance of a non-ideal filter can be calculated from the Parseval's theorem by integrating PSD over a specified interval. In real tests, the 43-order FIR equiripple filter with frequency characteristics denoted in Figure 4.2 was used as the half-band filter.

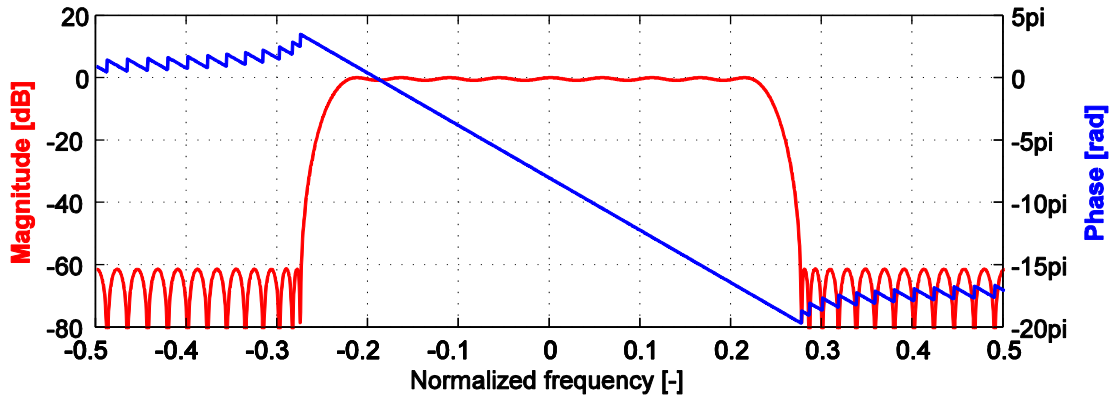


Figure 4.2: Half-band filter characteristics.

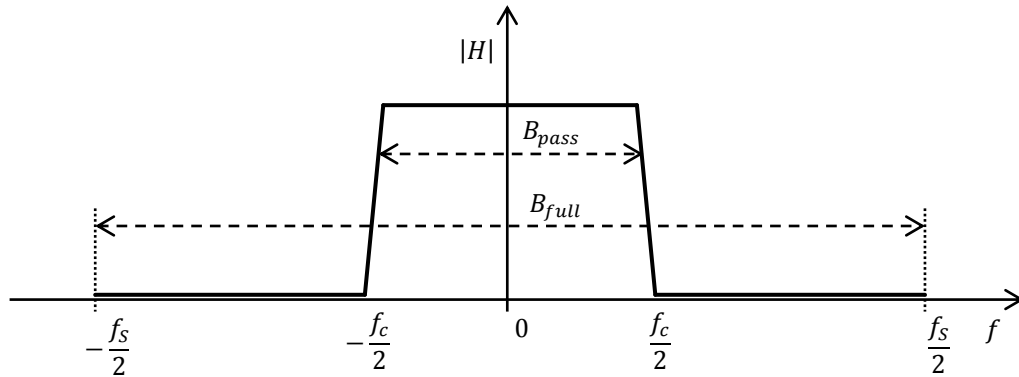


Figure 4.3: Integration intervals.



The power of the magnitude characteristic was integrated over two intervals; first, for the desired signal over its pass-band and second over the whole sampled interval to see noise performance. For clarity, the intervals are shown in Figure 4.3.

Results of the integration were obtained by numerical integration of the oversampled characteristic in MATLAB and are summarized in Table 4.1.

**Table 4.1:** Half-band filter integration results.

$N_{HB}(f_S) = \frac{1}{f_S} \int_{-\frac{f_S}{2}}^{\frac{f_S}{2}}  H_{HB}(f) ^2 df$	$N_{HB}(1) = 0.4732$
$S_{HB}(f_c, f_S) = \frac{1}{f_S} \int_{-\frac{f_c}{2}}^{\frac{f_c}{2}}  H_{HB}(f) ^2 df$	$S_{HB}(0.5, 1) = 0.4725$

As expected, the full-band power is nearly as low as the desired band power, i.e. the reduction of undesired signals/noise out of the pass band is high. The positive difference between powers,  $N_{HB} - S_{HB}$ , will, after decimation, be aliased and will, finally, contribute to BER degradation.

The frequency response of the CIC-decimator can be found analytically as described, e.g., in [54]. The first stage is an integrator described by the difference equation

$$y(n) = y(n-1) + x(n) \xrightarrow{\text{Z-transform}} H_I(z) = \frac{1}{1-z^{-1}} \quad (4.3)$$

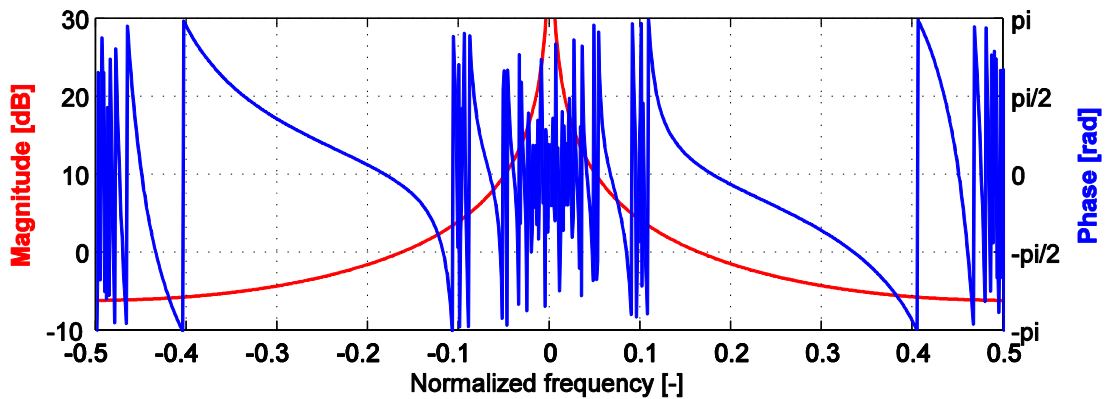
from which the transfer function can be obtained by the Z-transform [54]. Its magnitude

$$|H_I(e^{j\omega})|^2 = \frac{1}{2(1 - \cos(\omega))} \quad (4.4)$$

and phase

$$\arg(H_I(e^{j\omega})) = -\tan^{-1}\left(\frac{\sin(\omega)}{\cos(\omega) - 1}\right) \quad (4.5)$$

frequency responses [55] are drawn in Figure 4.4.



**Figure 4.4:** Integrator frequency characteristics ( $M = 4$ ).

The second stage is a comb-filter and its derived functions are

$$y(n) = x(n) - x(n - M) \xrightarrow{Z\text{-transform}} H_C(z) = 1 - z^{-M} \quad (4.6)$$

for the transfer function and

$$|H_C(e^{j\omega})|^2 = 2(1 - \cos(\omega M)) \quad (4.7)$$

and

$$\arg(H_C(e^{j\omega})) = \tan^{-1}\left(\cot\left(\frac{\omega M}{2}\right)\right) = -\frac{\omega M}{2} \quad (4.8)$$

for the frequency responses [55]. Figure 4.5 also shows the frequency responses.

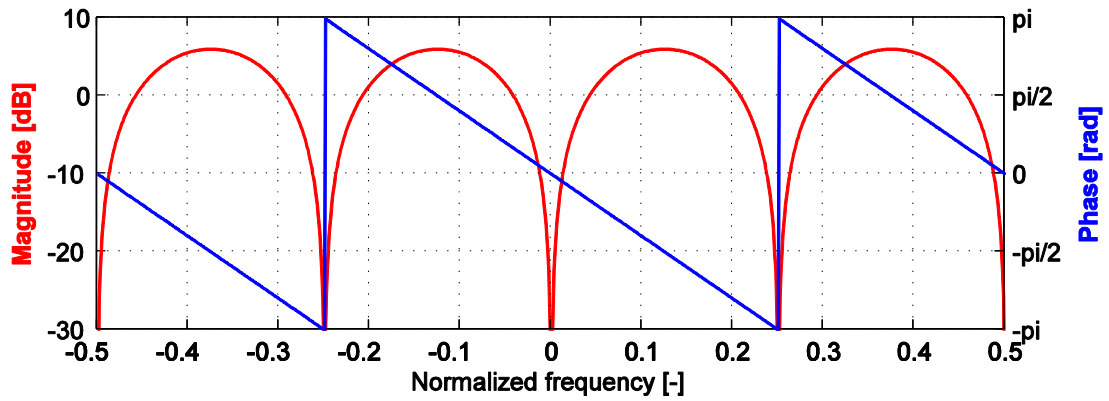


Figure 4.5: Comb-filter frequency characteristics ( $M = 4$ ).

By putting both transfer functions together a first order CIC-filter transfer function

$$H_{CIC}(z) = H_I(z) \cdot H_C(z) = \frac{1 - z^{-M}}{1 - z^{-1}} = \sum_{k=0}^{M-1} z^{-k} \quad (4.9)$$

is obtained. The equations

$$|H_{CIC}(e^{j\omega})|^2 = \frac{1 - \cos(\omega M)}{1 - \cos(\omega)} \quad (4.10)$$

and

$$\arg(H_{CIC}(e^{j\omega})) = \frac{\omega(1 - M)}{2} \quad (4.11)$$

show the magnitude and the phase responses [54]. The first-order CIC-filter is used in the Electra radio and its frequency response is shown in Figure 4.6.

Finally, the gain of the filter has to be normalized to unity by the reciprocal value of the decimation coefficient  $M$ ; such a necessity is obvious from the magnitude response in Figure 4.6 and the realization is also depicted in Figure 4.1.

Now it is possible to evaluate the signal and noise power performance of the CIC-filter. Before any calculations, it should be noted that decimation, meaning sample reduction, does not influence noise performance. The sampled band is reduced by the

decimation factor  $M$ , but it does not mean that the residual noise out of the newly sampled band is cleared. The noise is aliased into the new narrow band and the power of noise stays unchanged. For the following calculation we assume a decimation factor  $M = 4$ , meaning that the desired signal is originally four times oversampled. Table 4.2 shows the results of the integration for the example.

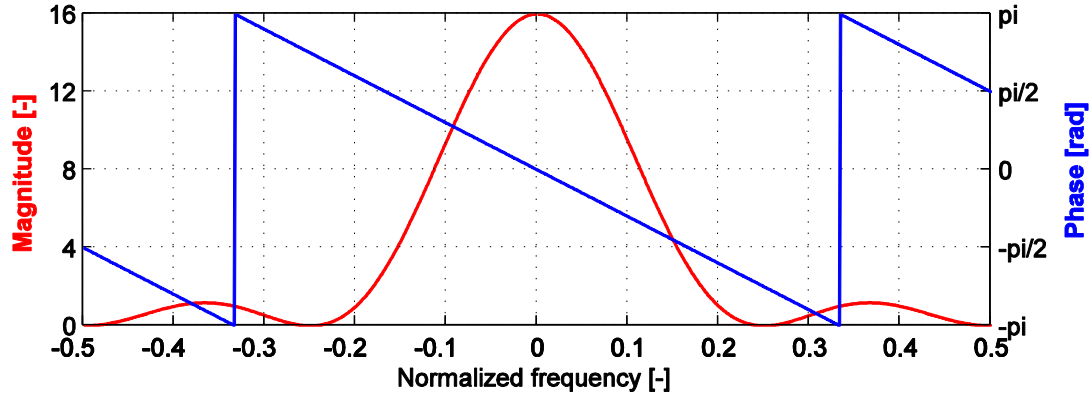


Figure 4.6: CIC-filter frequency characteristics.

Table 4.2: CIC-filter integration results.

$N_{CIC}(f_s, M) = \frac{1}{M^2 \cdot f_s} \int_{-\frac{f_s}{2}}^{\frac{f_s}{2}}  H_{CIC}(2\pi f) ^2 df$	$N_{CIC}(1, 4) = 0.250$
$S_{CIC}(f_s, M) = \frac{1}{M^2 \cdot f_s} \int_{-\frac{f_s}{2M}}^{\frac{f_s}{2M}}  H_{CIC}(2\pi f) ^2 df$	$S_{CIC}(1, 4) = 0.196$

The behavior of noise coming through the CIC-filter is in compliance with the expectations; three quarters of the noise power has been removed, but the performance of the signal power is poor. The magnitude characteristic is not sharp enough, and therefore almost a whole quarter of the desired signal power has been removed too, and therefore the aliasing impact is very high. Obviously, such a filter is not very useful for downsampling to the final data bandwidth. The filter can be used as a pre-downsampling filter which still produces an oversampled signal, but with a lower oversampling factor. Let us see the results for the same filter used for the half data rate and the half data bandwidth. The noise performance stays the same, but the signal performance is now calculated as shown in Table 4.3.

Table 4.3: CIC-filter integration results for half bandwidth.

$S_{CIC}(f_s, M, r) = \frac{1}{M^2 \cdot f_s} \int_{-\frac{r \cdot f_s}{2}}^{\frac{r \cdot f_s}{2}}  H_{CIC}(2\pi f) ^2 df$	$\frac{S_{CIC}(1, 4, 0.5)}{r} = 0.235$
---	--

The coefficient  $r$  was omitted in previous calculations because it equals one, but the coefficient is now lower than one depending on downsampled bandwidth utilization, i.e. for our example equaling 0.5. Now the degradation of the desired signal is much lower, but the price paid is the need of additional filtration and decimation. The aliasing

impact is not given by the simple difference as for  $r = 1$  because the aliasing components are not all added to the useful bandwidth and its distribution even is not uniform. The exact calculation is more complex and will be discussed later.

The overall performance of the downconverter in Figure 4.1 can now be expressed by

$$N_{DC}(f_S, M) = \frac{1}{M^2 \cdot f_S} \int_{-\frac{f_S}{2}}^{\frac{f_S}{2}} |H_{HB}(f) \cdot H_{CIC}(2\pi f)|^2 df \quad (4.12)$$

for noise and

$$S_{DC}(f_S, M, r) = \frac{1}{M^2 \cdot f_S} \int_{-\frac{r \cdot f_S}{2}}^{\frac{r \cdot f_S}{2}} |H_{HB}(f) \cdot H_{CIC}(2\pi f)|^2 df \quad (4.13)$$

for the signal. The analytical computation of these equations is highly impractical, more convenient is the numerical computation. Moreover, for our purposes it is sufficient to use only approximation of the half-band filter.

Since the magnitude characteristic of the CIC-filter is nearly zero around the half-band filter cut-off frequency and the pass band of the half-band filter has a flat magnitude characteristic and finally the stop-band gain of the half-band filter is relatively low, the real half-band filter can be replaced by an ideal filter with a rectangular shape magnitude characteristic. Thus, the computation simplifies to

$$N_{DC}(f_S, M) = \frac{1}{M^2 \cdot f_S} \int_{-\frac{f_S}{4}}^{\frac{f_S}{4}} |H_{CIC}(2\pi f)|^2 df \quad (4.14)$$

and

$$S_{DC}(f_S, M, r) = \frac{1}{M^2 \cdot f_S} \int_{-\frac{r \cdot f_S}{2}}^{\frac{r \cdot f_S}{2}} |H_{CIC}(2\pi f)|^2 df. \quad (4.15)$$

The results for the example are calculated in Table 4.4. Notice the differences from values in Table 4.2, from which is seen the impact of the half-band filter. Remember also that for the second case, where the useful band is halved, additional filtration and decimation will lower noise power.

**Table 4.4:** Downconverter integration results.

$N_{DC}(1, 4) = 0.231$
$S_{DC}(1, 4, 1) = 0.196$
$S_{DC}(1, 4, 0.5) = 0.117$

## 4.2. Novel approach to digital downconversion

Let us have a look at the downconverter in Figure 4.1 as a “black box”. The input is a continuous portion of signal samples from the ADC which contains a narrow-band signal somewhere within the sampled band among other signals, undesired ones, acting as noise. The output of the black-box is also a portion of samples, but less frequent and containing different information. Each sample is an average of the complex information contained in the selected narrow-band. That is, in fact, what the discrete Fourier transform does, particularly from a selection of effective algorithms the Goertzel algorithm fits the most to such a purpose.

The derivation of the Goertzel algorithm can be found in many books, e.g. [54], but the derivation of filtration characteristics is not so common. On the other hand, to be complete, let us briefly show the derivation from the beginning. The basic relation for the discrete Fourier transform

$$X(k) = \sum_{r=0}^{N-1} x(r) e^{-j2\pi r \frac{k}{N}} \quad (4.16)$$

can be put into a convolutional form

$$y_k(n) = \sum_{r=-\infty}^{\infty} x(r) \cdot e^{j2\pi(N-r)\frac{k}{N}} \cdot u(n-r) = x(n) * e^{j2\pi n \frac{k}{N}} \Big|_{n=N} \quad (4.17)$$

where  $x(r)$  is the  $r^{th}$  signal sample in the time domain and  $X(k)$  is the  $k^{th}$  bin of the Fourier spectrum. The function  $u(n)$  represents a rectangular shape function.

An impulse response of the derived filter is then a complex harmonic signal

$$h(n) = e^{j2\pi n \frac{k}{N}} \cdot u(n). \quad (4.18)$$

By applying the Z-transform to the impulse response (4.18) excluding the window, it is possible to find the transfer function of the Goertzel filter

$$H(z) = \sum_{n=0}^{\infty} h(n) z^{-1} = \frac{1}{1 - z^{-1} e^{j2\pi \frac{k}{N}}} \quad (4.19)$$

More convenient for implementing the Goertzel algorithm is the modified form

$$H(z) = \frac{1}{1 - z^{-1} e^{j2\pi \frac{k}{N}}} \cdot \frac{1 - z^{-1} e^{-j2\pi \frac{k}{N}}}{1 - z^{-1} e^{-j2\pi \frac{k}{N}}} = \frac{1 - z^{-1} e^{-j2\pi \frac{k}{N}}}{1 - 2z^{-1} \cos\left(2\pi \frac{k}{N}\right) + z^{-2}} \quad (4.20)$$

which can be split into real recursive and complex direct computational parts.

The realization of the transfer function (4.20) is shown in Figure 4.7. Notice that the filter has two complex poles located on the unit circle which is a condition of stability.



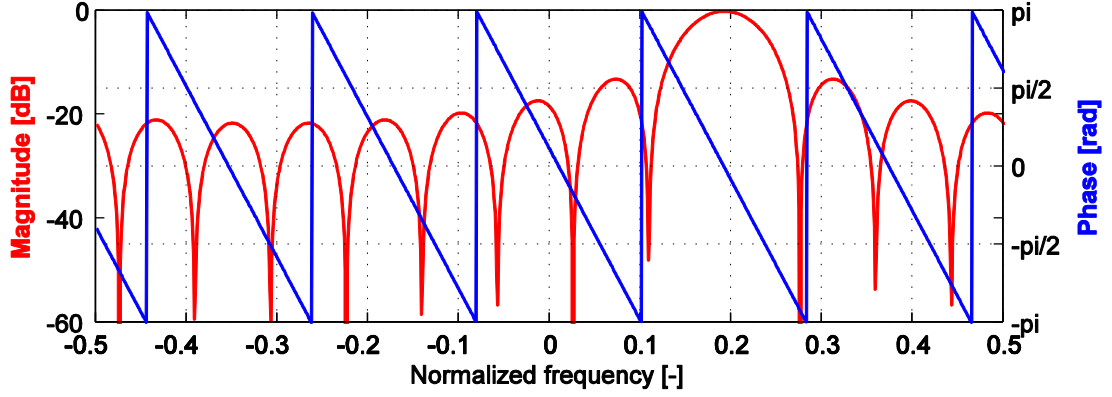


Figure 4.8: Goertzel filter frequency characteristics ( $N = 12, k = 2.3$ ).

Notice that PSD only has a one-side lobe due to the complex character of the transfer function, and the filtered signal is then, in general, also complex. The filter is tuned to a non-integer frequency,  $k \in \mathbb{R}$ , which is generally not possible for the discrete Fourier transform (DFT). If DFT is used, the frequency will be fixed for each bin and will depend on the length of DFT  $N$ . Such behavior would make DFT useless as a downconverter. Fortunately, the Goertzel algorithm may be tuned to whatever real frequency from the sampled range. When comparing the downconverter based on the CIC filter in Figure 4.1 and the Goertzel downconverter in Figure 4.7 one can see that the Goertzel filter does not perform any frequency conversion in the sense of a complex mixer. Since the algorithm is able to filter the desired signal in the passband, the downconversion can be provided by downsampling as aliasing to the baseband. The technique has the restriction of the integer decimation factor  $N$ , which makes the Goertzel algorithm useful only as a pre-downconverter, where the final downconversion has to be performed, e.g. by the Costas loop.

The characteristics in Figure 4.8 cannot be compared with the CIC-filter; first some conditions have to be applied to get a desired form for comparison. Substituting the normalized frequency  $f$  for the angle frequency  $\omega$  and assuming a desired frequency of zero yields the relation

$$|H_G(\omega)|^2\{k=0\} = \frac{1}{N^2} \frac{1 - \cos(\omega N)}{1 - \cos(\omega)}. \quad (4.25)$$

The Goertzel algorithm frequency characteristic is exactly the same as the CIC filter's one. Thus, the noise performance of the Goertzel downconverter will be the same.

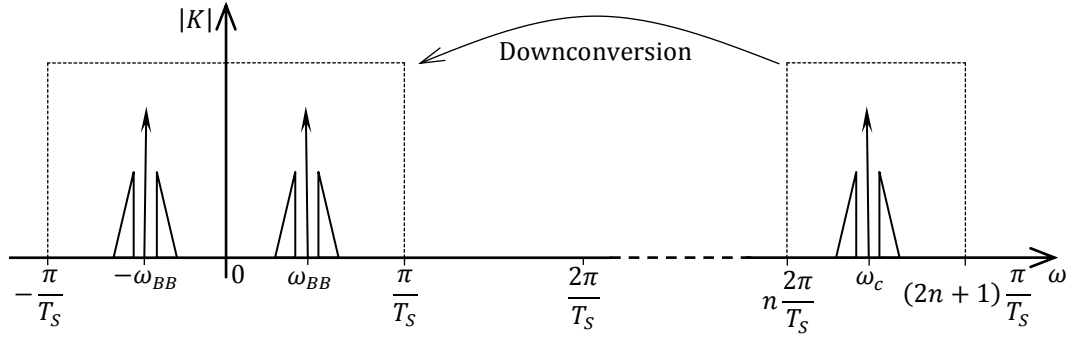
#### 4.2.1. Low speed digital AM signal detection

A good example of downconversion by undersampling is the DCF77 receiver published in [56]. The amplitude modulated signal expressed by

$$f_{AM}(t) = m(t) \cos(\omega_c t + \varphi_c) \quad (4.26)$$

is noticeably similar to the BPSK modulation (2.22), but the modulation function  $m(t)$  is generally the analog band-limited function.

The spectrum of the amplitude modulated signal is shown in Figure 4.9. It also shows the frequency downconversion.



**Figure 4.9:** Spectra of sampled AM signal.

The spectrum of the AM signal can be found by the Fourier transform of (4.26) as

$$\begin{aligned}
 \mathcal{F}\{f_{AM}(t)\} &= \int_{-\infty}^{\infty} f_{AM}(\tau) e^{-j\omega\tau} d\tau = \int_{-\infty}^{\infty} m(\tau) \cos(\omega_c t + \varphi_c) e^{-j\omega\tau} d\tau \\
 &= \frac{e^{j\varphi_c}}{2} \int_{-\infty}^{\infty} m(\tau) e^{j\omega_c\tau} e^{-j\omega\tau} d\tau + \frac{e^{-j\varphi_c}}{2} \int_{-\infty}^{\infty} m(\tau) e^{-j\omega_c\tau} e^{-j\omega\tau} d\tau \quad (4.27) \\
 &= \frac{e^{j\varphi_c}}{2} M(\omega - \omega_c) + \frac{e^{-j\varphi_c}}{2} M(\omega + \omega_c) = F_{AM}(\omega).
 \end{aligned}$$

The spectrum consists of the modulation signal shifted to the carrier frequency and its conjugate shifted to the negative carrier frequency. Applying sampling to the signal produces spectrum replicas according to (2.6) as depicted in the reconstruction formula

$$\begin{aligned}
 F_{AM}(\omega) &= \frac{1}{T_S} \sum_{n=-\infty}^{\infty} S(\omega) F_{AM}\left(\omega + 2\pi \frac{n}{T_S}\right) \\
 &= \frac{e^{j\varphi_c}}{2T_S} \sum_{n=-\infty}^{\infty} S(\omega) M\left(\omega - \omega_c + 2\pi \frac{n}{T_S}\right) \quad (4.28) \\
 &\quad + \frac{e^{-j\varphi_c}}{2T_S} \sum_{n=-\infty}^{\infty} S(\omega) M\left(\omega + \omega_c + 2\pi \frac{n}{T_S}\right).
 \end{aligned}$$

There is also the desired spectrum in the baseband that has to be reconstructed. Assuming the rectangular reconstruction function (2.8) expressed as  $S(\omega)$ , the baseband signal position  $\omega_{BB}$  can be found. It is clear that the conditions for the positive replica

$$\frac{2\pi}{2T_S} > \left| 2\pi \frac{n}{T_S} - \omega_c \right| \Rightarrow \left| n - \frac{\omega_c T_S}{2\pi} \right| < \frac{1}{2} \quad (4.29)$$

and for the negative replica

$$\left| n + \frac{\omega_c T_S}{2\pi} \right| < \frac{1}{2} \quad (4.30)$$

has to be met.

Substituting the values used in the DCF77 system to the conditions results in



$$\left| n - \frac{2\pi \cdot 77.5 \cdot 10^3}{2\pi \cdot 24 \cdot 10^3} \right| = |n - 3.23| < \frac{1}{2} \Rightarrow n = 3 \quad (4.31)$$

and

$$|n + 3.23| < \frac{1}{2} \Rightarrow n = -3. \quad (4.32)$$

From the values it is possible to evaluate the position of the signal's carrier after sampling

$$\begin{aligned} \omega_{BB} &= \omega_c - 2\pi \frac{n}{T_S} \\ &= 2\pi \cdot 77.5 \cdot 10^3 - 2\pi \cdot 3 \cdot 24 \cdot 10^3 = 2\pi \cdot 5.5 \cdot 10^3 \text{ Hz} \end{aligned} \quad (4.33)$$

and

$$\begin{aligned} -\omega_{BB} &= -\omega_c - 2\pi \frac{n}{T_S} \\ &= -2\pi \cdot 77.5 \cdot 10^3 + 2\pi \cdot 3 \cdot 24 \cdot 10^3 = -2\pi \cdot 5.5 \cdot 10^3 \text{ Hz}. \end{aligned} \quad (4.34)$$

The signal located at the low intermediate frequency has to be converted to the baseband to be decoded. For such a purpose, when the signal varies relatively slowly, the Goertzel algorithm can be used.

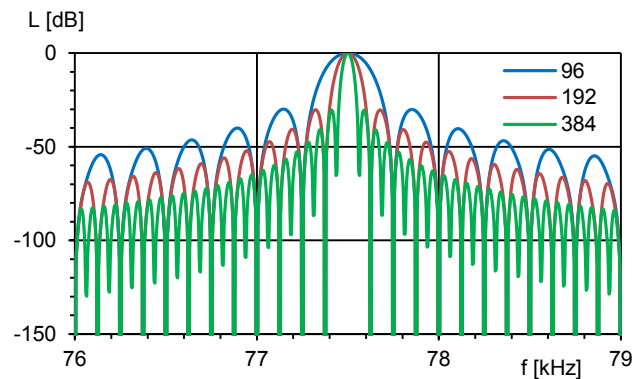
The demodulation can be provided just by finding the amplitude of the bin; such evaluation does not need any complex computation. The difference equation

$$|y(n+1)| = \sqrt{d^2(n+1) - 2 \cos\left(2\pi \frac{k}{N}\right) d(n+1)d(n) + d^2(n)} \quad (4.35)$$

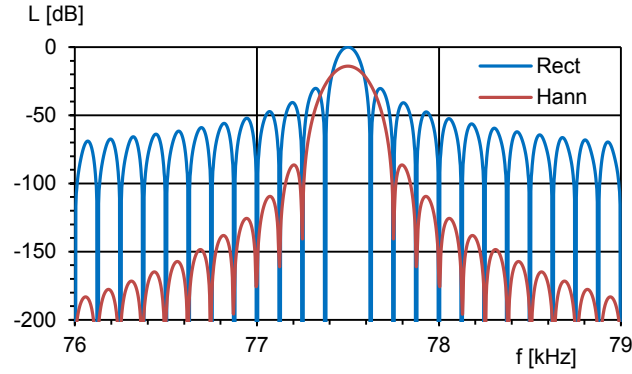
evaluates the absolute value from the state variables  $d(n)$ .

The influence of the different length is demonstrated on three examples in Figure 4.10 [56].

Due to the rectangular windowing the leakage of the side lobes is high. The example of using the Hanning window is shown in Figure 4.11 [56]. The side lobes are really suppressed as has been expected, but the main lobe widens twice.

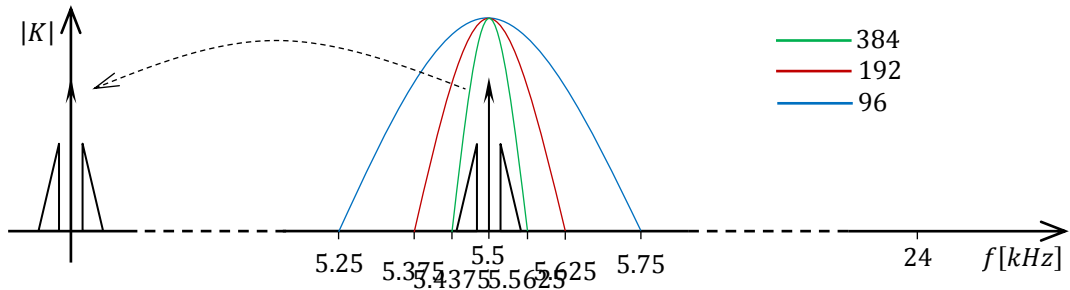


**Figure 4.10:** The influence of different lengths of the Goertzel filter.



**Figure 4.11:** The example of signal windowing.

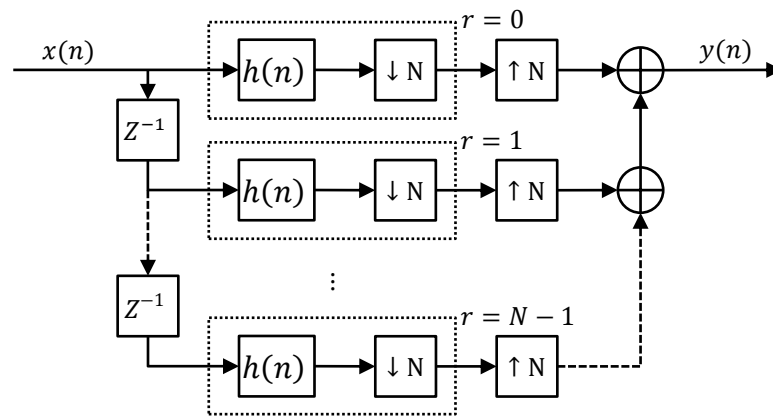
The figures above show the filter characteristics positioned to the frequency of the carrier to show how the signal is affected together with the undersampling effect. As has been shown, the DCF77 signal is actually, by undersampling, shifted to 5.5 kHz and that is shown in Figure 4.12 also with three examples of filter selectivity.



**Figure 4.12:** Spectrum position in the base band and filter selectivity.

A more detailed description of the DCF77 receiver and the prototype measurement results can be found in [56]. The description of the implementation of the DCF77 receiver and the discussion of the implementation problems is published in [57].

#### 4.2.2. Goertzel filter composition



**Figure 4.13:** Goertzel filter composition.

Consider the digital system in Figure 4.13. The system consists of  $N$  mutually delayed branches of the Goertzel filter. The signals are then successively multiplexed to the output; this operation is depicted by upsampling and summing the signals.

The dashed rectangle wraps the Goertzel filter which is represented by its impulse response  $h(n)$  and decimation ability with factor  $N$ . Each branch is indexed by the index  $r$ . Let us express each operation by equations, which can be found in [54]. It is more convenient to provide further calculations in the frequency domain. Applying the impulse response to the input signal  $x(n)$  yields multiplication in the frequency domain

$$X_h(\omega) = X(\omega) \cdot H(\omega). \quad (4.36)$$

The next stage, decimation or downsampling, adds  $N$  copies of the original spectrum shifted by the corresponding fraction of the original sampling frequency

$$X_d(\omega) = \frac{1}{N} \sum_{k=0}^{N-1} X_h\left(\frac{\omega}{N} - 2\pi \frac{k}{NT_s}\right). \quad (4.37)$$

Notice that since the input signal is discretized in time, the spectrum  $X(\omega)$  is already periodical with period  $T_s$ .

Upsampling does not add any spectra; the only change is expansion of the signal bandwidth, so that the  $N$ -times upsampled bandwidth is as wide as the bandwidth of the original signal, but still contains the spectra added by decimation. The bandwidth expansion is expressed by

$$X_u(\omega) = X_d(\omega N). \quad (4.38)$$

The delay in time is transformed to the frequency domain as a phase shift of the signal spectrum

$$x_z(n) = x(n - rT_s) \xrightarrow{\mathcal{F}\text{-transform}} X_z(\omega) = X(\omega) \cdot e^{-j\omega rT_s}. \quad (4.39)$$

Putting all together according to Figure 4.13 yields a relation for the output spectrum of the proposed system

$$Y(\omega) = \sum_{r=0}^{N-1} \left( \frac{1}{N} \sum_{k=0}^{N-1} H\left(\omega - 2\pi \frac{k}{MT_s}\right) \cdot X\left(\omega - 2\pi \frac{k}{MT_s}\right) \right) \cdot e^{-j\omega rT_s}. \quad (4.40)$$

The order of the sums can be exchanged and the copies of filtered spectra may be expressed as a convolution of the original spectrum with periodic impulse train  $\delta(\omega)$

$$Y(\omega) = (H(\omega) \cdot X(\omega)) * \frac{1}{N} \sum_{k=0}^{N-1} \left( \delta\left(\omega - 2\pi \frac{k}{NT_s}\right) \cdot \sum_{r=0}^{N-1} e^{-j\omega rT_s} \right). \quad (4.41)$$

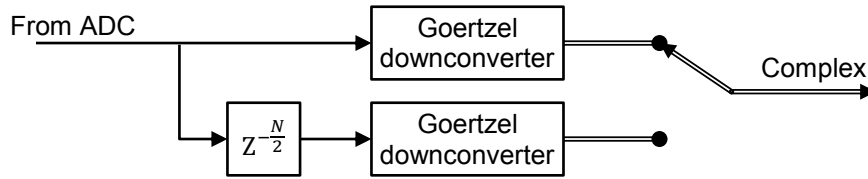
The impulse train has a non-zero value only when its input parameter is zero, so that condition may also be applied to the inner sum item as a substitution of angle frequency  $\omega$ . To pass the condition of convergence of summation of the geometric series the item for  $k = 0$  has to be excluded from the sum

$$Y(\omega) = (H(\omega) \cdot X(\omega)) * \left( \frac{1}{N} \delta(\omega) \sum_{r=0}^{N-1} 1 + \frac{1}{N} \sum_{k=1}^{N-1} \left( \delta\left(\omega - 2\pi \frac{k}{NT_s}\right) \frac{1 - e^{-j2\pi k}}{1 - e^{-j2\pi \frac{k}{N}}} \right) \right). \quad (4.42)$$

Finally, the first sum eliminates weighting of the Dirac impulse and the inner item of the second sum is found to be zero, so the output signal has been proved to be the input signal convoluted with the Goertzel impulse response

$$Y(\omega) = H(\omega) \cdot X(\omega) \xrightarrow{\mathcal{F}^{-1}\text{-transform}} y(n) = x(n) * h(n). \quad (4.43)$$

It has been found that the proposed system eliminates the decimation property of the Goertzel filter, thus, it may be used as a downconverter with a specified decimation factor independently on its filtration characteristics. In Figure 4.14 there is shown the realization structure of the downconverter based on the Goertzel algorithm with the decimation factor  $\frac{N}{2}$ .



**Figure 4.14:** Realization structure of the composed Goertzel downconverter.

#### 4.2.3. Goertzel filter modification for real frequencies

From its nature, the discrete Fourier transform evaluates the spectrum of the input signal that is sampled and each sample (bin) corresponds to a precisely defined frequency. It is not possible to simply change coefficients in the fast Fourier transform algorithms to change the frequency of the bin. The Goertzel algorithm is different; its coefficient can be very easily changed to tune the Goertzel filter to any arbitrary frequency from the range limited by the sampling frequency [58]. This is true, but let us see how it behaves when using a simple rectangular window.

The analytical response of the Goertzel algorithm to a real harmonic signal on which it is tuned can be expressed as limited convolution in

$$Y_k(n) = s_k(n) * h_k(n)|_{n=N} = \sum_{m=0}^{N-1} \cos\left(2\pi m \frac{k}{N}\right) e^{j2\pi \frac{k}{N}(n-m)}. \quad (4.44)$$

Where the convolution is limited to  $N$  samples and  $k$  is the same variable as in the Fourier transform expression (4.16), but from now it is a real parameter ( $k \in \mathbb{R}$ ). The relation between the tuned frequency and the variable  $k$  is expressed in

$$\frac{k}{N} = \frac{f_c}{f_s} \quad (4.45)$$

where  $f_s$  is the sampling frequency,  $f_c$  is the tuned frequency and  $N$  is the length of the Fourier transform. The exact analytical evaluation of the sum after some computation yields the relation

$$Y_k(n) = \frac{1}{2} \left( N + \frac{1 - e^{-j4\pi k}}{1 - e^{-j4\pi \frac{k}{N}}} \right). \quad (4.46)$$

The first addend corresponds to one side spectral amplification of the Fourier transform and also the Goertzel algorithm. The second parameter is obviously zero whenever  $k$  is an integer ( $k \in \mathbb{Z}$ ) and that is in agreement with the discrete Fourier transform. However, since  $k$  is real, it is nonzero and the resulting value is not equal to really applied signal parameters. The distortion comes from the opposite side of the Fourier spectrum as spectral leakage and can be eliminated by additional post-processing or pre-processing. The impact on the output value is more obvious from the graphical example of the second parameter in Figure 4.15.

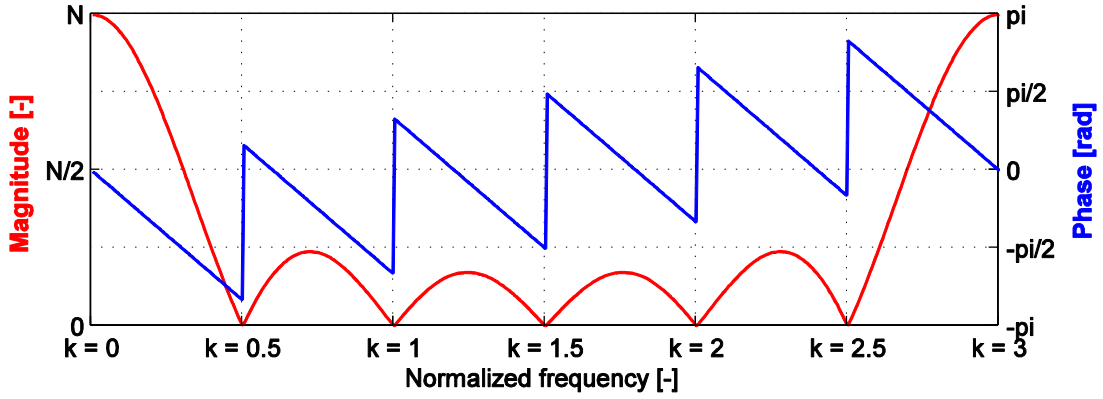


Figure 4.15: Plot of parameter  $A$  for  $N = 6$ .

For a more general input harmonic signal, equations (4.44) and (4.45) yield

$$\begin{aligned} Y_k(n) &= \sum_{m=0}^{N-1} M \cdot \cos\left(2\pi \frac{k}{N} m + \varphi\right) e^{j2\pi \frac{k}{N}(n-m)} \\ &= \frac{M}{2} \left( N e^{-j\varphi} + e^{j\varphi} \frac{1 - e^{-j4\pi k}}{1 - e^{-j4\pi \frac{k}{N}}} \right) \end{aligned} \quad (4.47)$$

The new parameters are the magnitude  $M$  and the phase  $\varphi$ .

Substitution according to

$$A = \frac{1 - e^{-j4\pi k}}{1 - e^{-j4\pi \frac{k}{N}}} \quad (4.48)$$

and

$$B = M e^{-j\varphi} \quad (4.49)$$

yields

$$Y_k(n) = \frac{1}{2} (BN + B^* A) \quad (4.50)$$

which is a more convenient relation for the following operations.

Equation (4.50) can be solved for unknown  $B$ , where  $Y_k$  is the Goertzel filter output, so the result

$$B = \frac{2}{AA^* - N^2} (AY_k^* - NY_k) \quad (4.51)$$

is actually the compensation for the real  $k$ .

From now, it is possible to find the compensation coefficients for an algebraic form of the result for a polar form. For clarity, the real and imaginary parts of the values are denoted only by the indexes  $R$  and  $I$ , respectively, the index  $k$  is omitted. Equations (4.52) through (4.56) show the compensation coefficients for the real and imaginary parts.

$$B_R = Y_I Q_1 + Y_R(Q_2 - Q_3) \quad (4.52)$$

$$B_I = Y_R Q_1 - Y_I(Q_2 + Q_3) \quad (4.53)$$

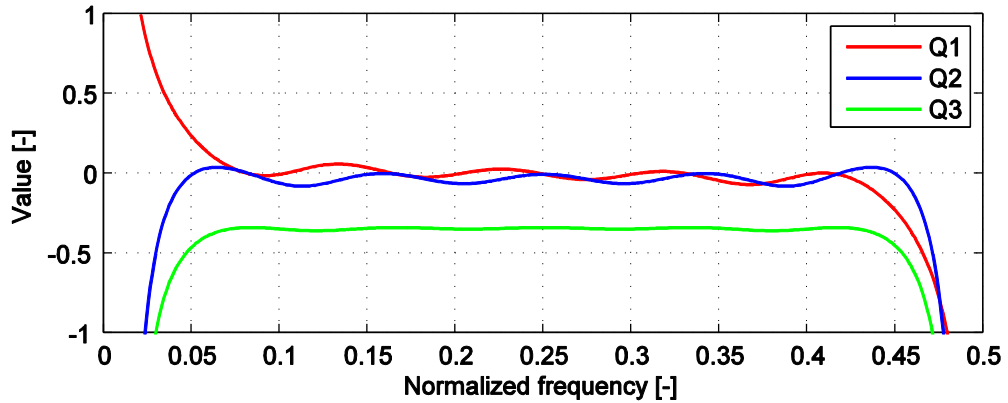
$$Q_1 = \frac{2A_I}{A_R^2 + A_I^2 - N^2} \quad (4.54)$$

$$Q_2 = \frac{2A_R}{A_R^2 + A_I^2 - N^2} \quad (4.55)$$

$$Q_3 = \frac{2N}{A_R^2 + A_I^2 - N^2} \quad (4.56)$$

Notice that when  $k$  is an integer, i.e.  $A$  is zero, the coefficients collapse to one weighting coefficient  $\frac{2}{N}$ . The character of the coefficients is clearly seen in Figure 4.16.

For the realization structure, a more convenient form of the Goertzel algorithm (4.20) should be considered, in which the complex output is obtained by weighting and summing two mutually delayed samples of the recursive part, let us entitle them  $d(n)$  for the direct one and  $d(n-1)$  for the delayed one. The original weighting and compensation can then be put together to result in a more complex realization, a little computation yields equations (4.57) through (4.61).



**Figure 4.16:** Compensation coefficients for the algebraic form for  $N = 6$ .

$$Y_k(n) = d(n) - d(n-1)\tilde{C} \rightarrow \begin{aligned} Y_R &= d_n - d_{n-1}\tilde{C}_R \\ Y_I &= -d_{n-1}\tilde{C}_I \end{aligned} \quad (4.57)$$

$$B_R = d_n Q_{d2} - d_{n-1}(\tilde{C}_R Q_{d2} + \tilde{C}_I Q_{d1}) \quad (4.58)$$

$$B_I = d_n Q_{d1} + d_{n-1}(\tilde{C}_R Q_{d1} - \tilde{C}_I Q_{d2}) \quad (4.59)$$

$$Q_{d1} = \frac{2A_I}{A_R^2 + A_I^2 - N^2} \quad (4.60)$$

$$Q_{d2} = \frac{2(A_R + N)}{A_R^2 + A_I^2 - N^2} \quad (4.61)$$

The coefficients  $C$  and  $\tilde{C}$  have already been defined in Figure 4.7 and its real and imaginary parts are again indexed by  $R$  and  $I$  respectively. The algebraic form of the output filtered signal can be used to obtain the magnitude and the phase of the signal, which are more convenient in some systems. For the original Goertzel algorithm, there is a known optimized method for computing signal power

$$|Y_k(n)|^2 = d(n)^2 + d(n-1)^2 - Cd(n)d(n-1). \quad (4.62)$$

It is easy to adjust equation (4.62) to compensate the distorted output samples. The modified equation

$$|Y_k(n)|^2 = d(n)^2 Q_{M1} + d(n-1)^2 Q_{M2} - Cd(n)d(n-1) Q_{M3} \quad (4.63)$$

newly needs more coefficients than the original one

$$Q_{M1} = \frac{4(A_R^2 + A_I^2 + N^2 - 2A_R N)}{(A_R^2 + A_I^2 - N^2)^2}, \quad (4.64)$$

$$Q_{M2} = \frac{4 \left( A_R^2 + A_I^2 + N^2 - 2N \left( A_R \cos \left( 4\pi \frac{k}{N} \right) + A_I \sin \left( 4\pi \frac{k}{N} \right) \right) \right)}{(A_R^2 + A_I^2 - N^2)^2}, \quad (4.65)$$

$$Q_{M3} = \frac{4 \left( A_R^2 + A_I^2 + N^2 - 2N \left( A_R + A_I \frac{\sin \left( 2\pi \frac{k}{N} \right)}{\cos \left( 2\pi \frac{k}{N} \right)} \right) \right)}{(A_R^2 + A_I^2 - N^2)^2}. \quad (4.66)$$

The evolution of the coefficients dependent on the normalized frequency  $k$  is depicted in the example in Figure 4.17. Note that the mean is  $\left(\frac{2}{N}\right)^2$  as expected for the integer  $k$ .

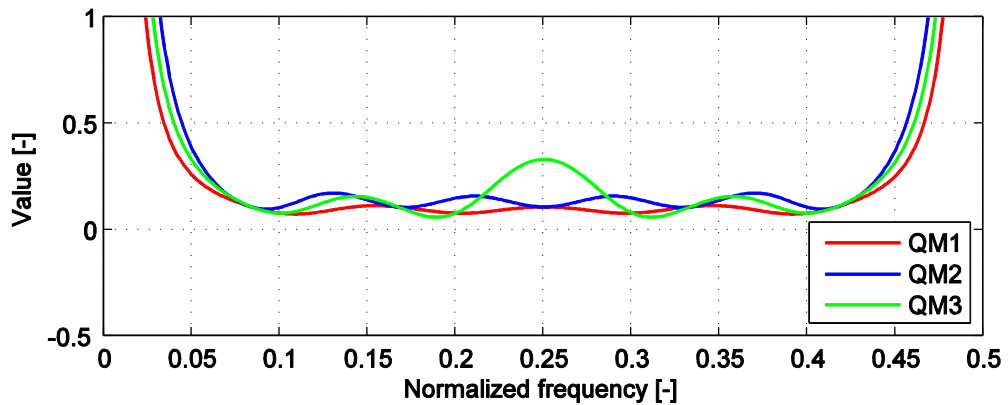


Figure 4.17: Compensation coefficients for magnitude for  $N = 6$ .

The phase computation can be similarly obtained as follows in equations (4.67) - (4.70).

$$\varphi_k(n) = \arctan\left(\frac{d(n)Q_{\varphi 1} + d(n-1)Q_{\varphi 2}}{d(n) - d(n-1)Q_{\varphi 3}}\right) \quad (4.67)$$

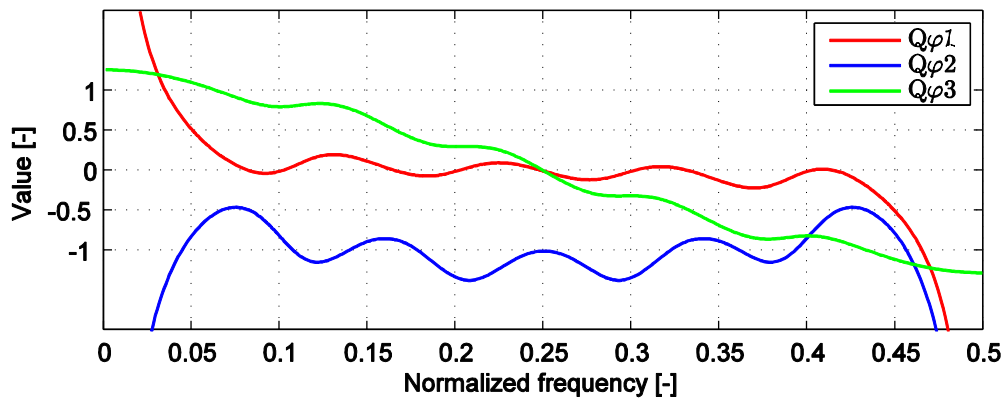
$$Q_{\varphi 1} = \frac{A_I}{A_R - N} \quad (4.68)$$

$$Q_{\varphi 2} = \frac{A_R + N}{A_R - N} \sin\left(2\pi \frac{k}{N}\right) - \frac{A_I}{A_R - N} \cos\left(2\pi \frac{k}{N}\right) \quad (4.69)$$

$$Q_{\varphi 3} = \frac{A_I}{A_R - N} \sin\left(2\pi \frac{k}{N}\right) + \cos\left(2\pi \frac{k}{N}\right) \quad (4.70)$$

The plot of the phase correction coefficients is shown in Figure 4.18. The  $\arctan(x)$  function can be approximated by known algorithms, e.g. the CORDIC algorithm.

Finally, the influence of the compensation should be presented on the real example. First, the magnitude computation is visualized as waterfall spectrograms in Figure 4.20. As expected, the spectrograms for the integer  $k$  are the same as for the original and the compensated Goertzel algorithm. Bad results are given by the uncompensated algorithm for the non-integer value of  $k$  as shown in spectrogram c). After compensation, the distorting pattern is suppressed around the central frequency and the evolution of the central frequency magnitude is constantly unity as is in accordance to the input signal. The distorting pattern on the spectrograms causing undesired fluctuations of the output magnitude value is in fact the result of spectral leakage from the negative to the positive frequency side and vice versa. The shorter the Goertzel filter and the closer the central frequency to the band edges is, the more apparent the leakage is. The proposed compensation technique recalculates the magnitude to be more or less matching the real value and is effective but does not solve the problem from the root of the cause. To get better results, it is necessary to use another technique to suppress spectral leakage to a minimum, e.g. windowing.



**Figure 4.18:** Compensation coefficients for phase for  $N = 6$ .

The Goertzel algorithm is intended to be used as a downconverter mainly for phase modulated signals, therefore let us now see the influence on phase. Originally, the Goertzel algorithm deals with the integer multiple of periods of the harmonic impulse



response, see 5.18. Since the frequency coefficient  $k$  is not an integer, also the multiple of periods is not an integer and the impulse response ends with a non-zero phase. That is the reason why each sample is phase-shifted against the previous one; the relative phase shift is

$$\Delta\varphi_r = 2\pi(1 - \text{mod}(k; 1)) \quad (4.71)$$

where the  $\text{mod}(x, d)$  function is the remainder after division of  $x$  by  $d$ .

Within the working frequency range, besides the edges, the leakage influence on phase is pretty small and the compensation may then be, in many cases, omitted. The range width is obvious from Figure 4.18, where the coefficients fluctuate around the constant, i.e. 0 and -1 respectively. Note that the coefficient  $Q\varphi3$  includes the original coefficient  $C$ , which causes the function to descend, otherwise the function is constant. The impact on the real example is shown in Figure 4.19.

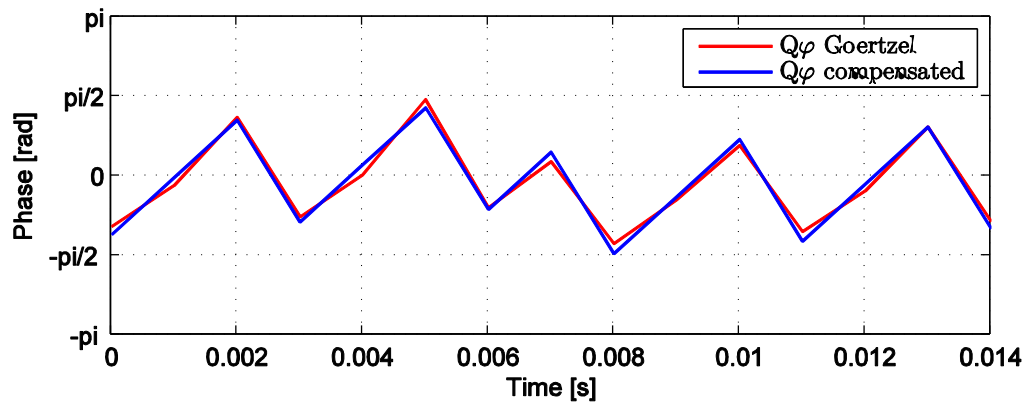
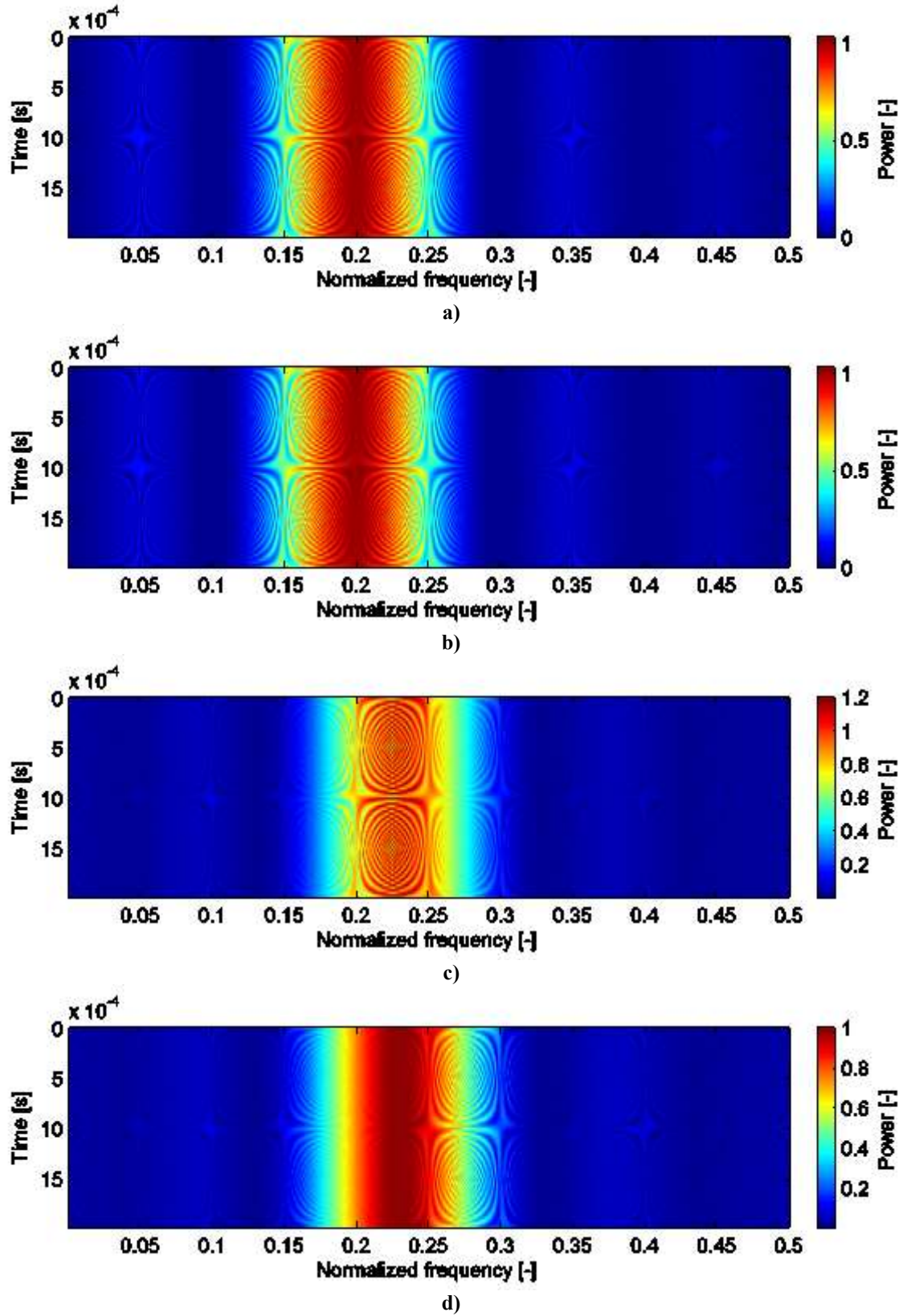


Figure 4.19: Phase compensation influence ( $N = 6; k = 0.82$ ).



**Figure 4.20:** Spectrograms of the Goertzel filter for  $N = 12$  – **a)** Uncompensated filter for  $k = 2$ , **b)** Compensated filter for  $k = 2$ , **c)** Uncompensated filter for  $k = 2.3$ , **d)** Compensated filter for  $k = 2.3$ .

#### 4.2.4. Improved structure of the decimating filter

The filtration characteristic of the Goertzel algorithm has, due to its simplicity, a quite low level of signal suppression outside the pass band and the subsequent decimation causes aliasing noise. The character of the power spectral density characteristic depends directly on the impulse response of the filter and especially the sharp edges are linked with the power level in the stop band of the frequency characteristic. Common knowledge suggests weighting of the impulse response by one of special weighting windows the same way as is common for an FFT analysis. Such a way is, of course, possible, but there is a question: Can we somehow modify the Goertzel algorithm to make its impulse response to be smoother? And, we can.

The impulse response of the original Goertzel algorithm is a complex harmonic signal as has been defined above in (4.18) and, as we know, the special combination of two harmonic signals of different but near frequencies results in beat tones. The idea is to use beat modulation to shape the impulse response. First, we take two impulse responses indexed as  $A$

$$h_A(n) = e^{j2\pi\frac{k+d}{N}n} \quad (4.72)$$

and  $B$

$$h_B(n) = e^{j2\pi\frac{k-d}{N}n} \quad (4.73)$$

respectively, which are moved from the center frequency by the offset  $d$ , and calculate the convolution as defined by

$$h_{M1}(n) = \sum_{s=0}^{\infty} h_A(s)h_B(n-s) = e^{j2\pi\frac{k-d}{N}n} \sum_{s=0}^{\infty} e^{j4\pi\frac{d}{N}s}. \quad (4.74)$$

The impulse response of the Goertzel filter is cut by the rectangular window and the convoluted impulse characteristics also have to be limited. The convolution (4.74) becomes limited where the impulse characteristics have length  $N$ . The result is nonzero only for the partial and complete overlaps and the convolution sum can be constrained by zero and the sample index  $n$ , as is in the first part of equation (4.75), which, finally, has a more convenient form.

$$h_{M1}(n) = \sum_{s=0}^n h_A(s)h_B(n-s) = e^{j2\pi\frac{k}{N}n} \frac{\sin\left(2\pi\frac{d}{N}(n+1)\right)}{\sin\left(2\pi\frac{d}{N}\right)} \quad (4.75)$$

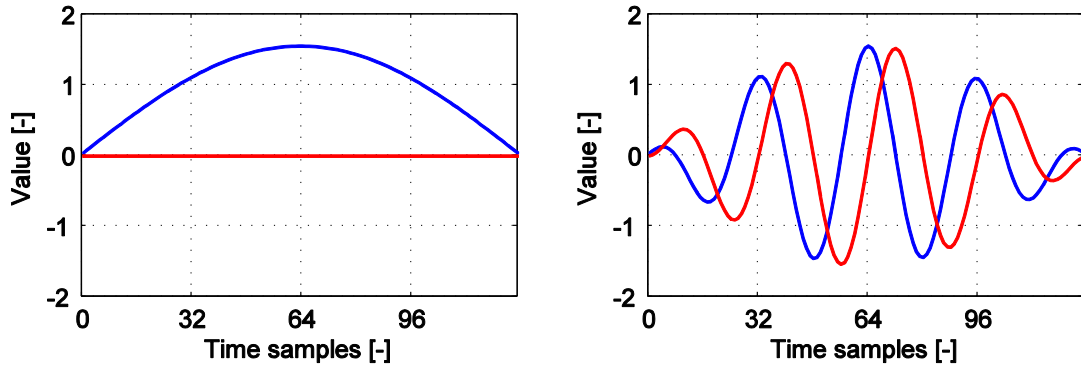
Now we have to find the value  $d$ , where one beat of the impulse response fits to the length of  $N$  samples, so it has minimal values on its edges. The first sample is defined exactly by the character of the recursive realization structure and always equals  $1 + 0j$ . The characteristic then crosses the zero point, which could be taken as the last sample, but when the sample of the impulse response equals zero, the filter output sample equals the combination of weighted delayed samples and it is not a valid output for this type of filter. Therefore, the sample one step back from zero is assumed as the last sample, which equals  $e^{-j2\pi\frac{k}{N}}$ . Assuming these conditions, we define

$$\begin{aligned}
 h_{M1}(N-1) &= e^{-j2\pi\frac{k}{N}} \\
 e^{-j2\pi\frac{k}{N}} &= e^{-j2\pi\frac{k}{N}} \frac{\sin(2\pi d)}{\sin\left(2\pi\frac{d}{N}\right)} \\
 \sin(2\pi d) &= \sin\left(2\pi\frac{d}{N}\right).
 \end{aligned} \tag{4.76}$$

The first suitable solution of (4.76) is

$$d = \frac{1}{2} \frac{N}{N+1} \tag{4.77}$$

In Figure 4.21 are shown the examples of derived impulse responses for the baseband ( $k = 0$ ) and the passband ( $k = 4$ ). Both characteristics are normalized to its length  $N$  and are visualized as real (red) and imaginary (blue) parts.



**Figure 4.21:** Impulse responses of the modified Goertzel filter for  $N = 128$  and  $k = 0$  (left),  $k = 4$  (right).

The derivation of the frequency characteristics is similar to the Goertzel filter frequency characteristics derivation. The discrete Fourier transform of the impulse response is defined and expanded in

$$\begin{aligned}
 H_{M1}(f) &= \sum_{n=0}^N h_{M1}(n) e^{-j2\pi f n} \\
 &= \frac{1}{\sin\left(2\pi\frac{d}{N}\right)} \sum_{n=0}^N \sin\left(2\pi\frac{d}{N}(n+1)\right) e^{j2\pi\frac{k}{N}n} e^{-j2\pi f n} \\
 &= \frac{1}{e^{j2\pi\frac{d}{N}} - e^{-j2\pi\frac{d}{N}}} \sum_{n=0}^N \left( e^{j2\pi\left(\frac{d}{N}(n+1) - n\left(f - \frac{k}{N}\right)\right)} - e^{-j2\pi\left(\frac{d}{N}(n+1) + n\left(f - \frac{k}{N}\right)\right)} \right).
 \end{aligned} \tag{4.78}$$

To simplify further operations let us assume the substitutions

$$e^{\alpha} = e^{j2\pi\left(f - \frac{k}{N}\right)} \tag{4.79}$$

and

$$e^{\beta} = e^{j2\pi d}. \tag{4.80}$$

Now we split the equation and evaluate the sums separately

$$H_{M1}(f) = K(A + B), \tag{4.81}$$

$$K = \frac{1}{e^{\frac{\beta}{N}} - e^{-\frac{\beta}{N}}}, \quad (4.82)$$

$$A = e^{\frac{\beta}{N}} \sum_{n=0}^N e^{n(\frac{\beta}{N}-\alpha)} = \frac{1 - e^{(\frac{\beta}{N}-\alpha)(N+1)}}{1 - e^{(\frac{\beta}{N}-\alpha)}}, \quad (4.83)$$

$$B = e^{-\frac{\beta}{N}} \sum_{n=0}^N e^{-n(\frac{\beta}{N}+\alpha)} = \frac{1 - e^{-(\frac{\beta}{N}+\alpha)(N+1)}}{1 - e^{-(\frac{\beta}{N}+\alpha)}}. \quad (4.84)$$

Equation (4.81) after some manipulation yields

$$H_{M1}(n) = \frac{\left(1 - e^{-\alpha(N+1)} e^{\frac{\beta}{N}+\beta}\right) \left(e^{\frac{\beta}{N}} - e^{-\alpha}\right) - \left(1 - e^{-\alpha(N+1)} e^{-\left(\frac{\beta}{N}+\beta\right)}\right) \left(e^{-\frac{\beta}{N}} - e^{-\alpha}\right)}{\left(e^{\frac{\beta}{N}} - e^{-\frac{\beta}{N}}\right) \left(e^{-\frac{\beta}{N}} - e^{-\alpha}\right) \left(e^{\frac{\beta}{N}} - e^{-\alpha}\right)}. \quad (4.85)$$

The result is too complex because it also includes non-optimal characteristics. If we assume the usage in the optimal configuration, we substitute the relation for the optimal value of  $d$  (4.77). The resulting simplification lies in

$$\beta = j\pi \frac{N}{N+1} \rightarrow e^{\frac{\beta}{N}+\beta} = -1. \quad (4.86)$$

Substituting (4.86) into (4.85) after some manipulation yields the final frequency characteristic for the optimal modified filter

$$\begin{aligned} H_{M1}(f) &= \frac{1 + e^{-\alpha(N+1)}}{\left(e^{-\frac{\beta}{N}} - e^{-\alpha}\right) \left(e^{\frac{\beta}{N}} - e^{-\alpha}\right)} \\ &= \frac{e^{j2\pi(f-\frac{k}{N})} + e^{-j2\pi N(f-\frac{k}{N})}}{\left(e^{j2\pi(f-\frac{k}{N})} + e^{j2\pi(f-\frac{k}{N})}\right) - \left(e^{j2\pi\frac{d}{N}} + e^{-j2\pi\frac{d}{N}}\right)} \\ &= \frac{\cos\left(\pi\left(f - \frac{k}{N}\right)(N+1)\right)}{\cos\left(2\pi\left(f - \frac{k}{N}\right)\right) - \cos\left(\frac{\pi}{N+1}\right)} e^{-j\pi\left(f - \frac{k}{N}\right)(N-1)}. \end{aligned} \quad (4.87)$$

The normalization coefficient for the transfer function can be found by setting the transfer at the interested frequency to one, and then the multiplicative coefficient is equal to the reciprocal value of

$$H_{M1}\left(\frac{k}{N}\right) = \frac{1}{1 - \cos\left(\frac{\pi}{N+1}\right)}. \quad (4.88)$$

Figure 4.22 and Figure 4.23 compare the frequency characteristics with the Goertzel filter characteristics. Obviously the suppression of a stop band is rapidly improved and even the bandwidth is not excessively widened compared to windowing by the Hann or Hamming windows [56].

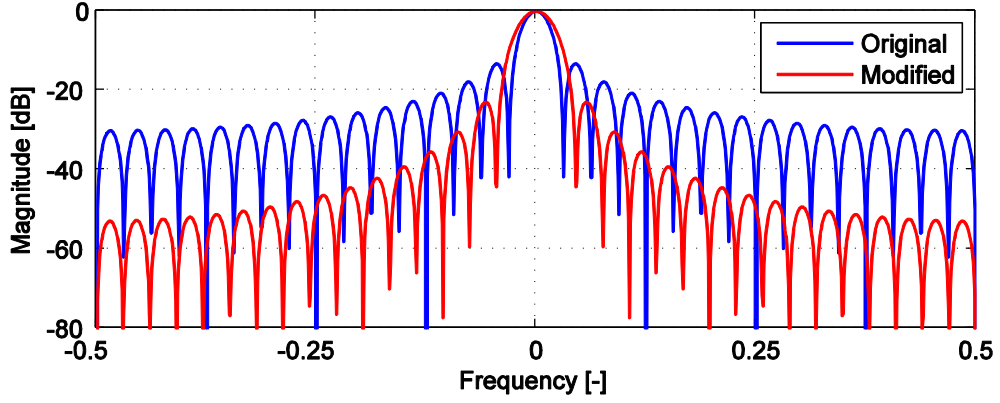


Figure 4.22: Magnitude frequency characteristics of the Goertzel filter and its modified version for  $N = 32$  and  $k = 0$ .

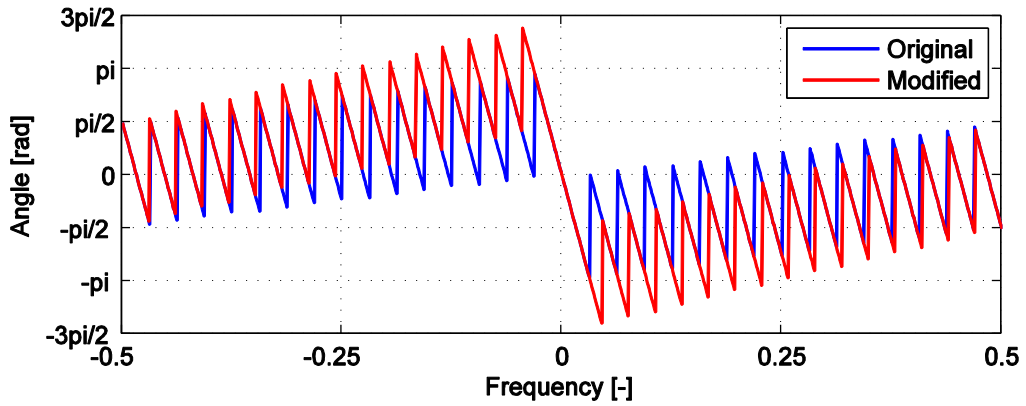


Figure 4.23: Phase frequency characteristics of the Goertzel filter and its modified version for  $N = 32$  and  $k = 0$ .

When deriving the frequency response we could not use duality between the frequency and time domain because of the limitation for  $N$  samples, which is excluded from the realization structure. The structure directly has to realize the impulse response, so the duality of multiplication vs. convolution can be used for deriving the realization structure. Thus, the transfer function in the  $Z$  domain consists of the multiplication of transfer functions defining the impulse responses (4.72) and (4.73) where the response with a positive offset is marked by the  $+$  sign and with a negative offset by the  $-$  sign

$$H_{M1}(z) = H_{G+}(z)H_{G-}(z) = \frac{1 - \tilde{C}^+ z^{-1}}{1 - C^+ z^{-1} + z^{-2}} \cdot \frac{1 - \tilde{C}^- z^{-1}}{1 - C^- z^{-1} + z^{-2}} \quad (4.89)$$

The coefficients

$$\tilde{C}^{\pm} = e^{-j2\pi \frac{k \pm d}{N}} \quad (4.90)$$

and

$$C^{\pm} = 2 \cos\left(2\pi \frac{k \pm d}{N}\right) \quad (4.91)$$

are defined in accordance to the Goertzel filter coefficients in Figure 4.7.

In final form

$$H_{M1}(z) = \frac{1 - \tilde{C}_1 z^{-1} + \tilde{C}_2 z^{-2}}{1 - C_2(z^{-1} + z^{-3}) + C_1 z^{-2} + z^{-4}} \quad (4.92)$$

the coefficients are reformulated to a more convenient form

$$\tilde{C}_1 = 2 \cos\left(2\pi \frac{d}{N}\right) e^{-j2\pi \frac{k}{N}}, \quad (4.93)$$

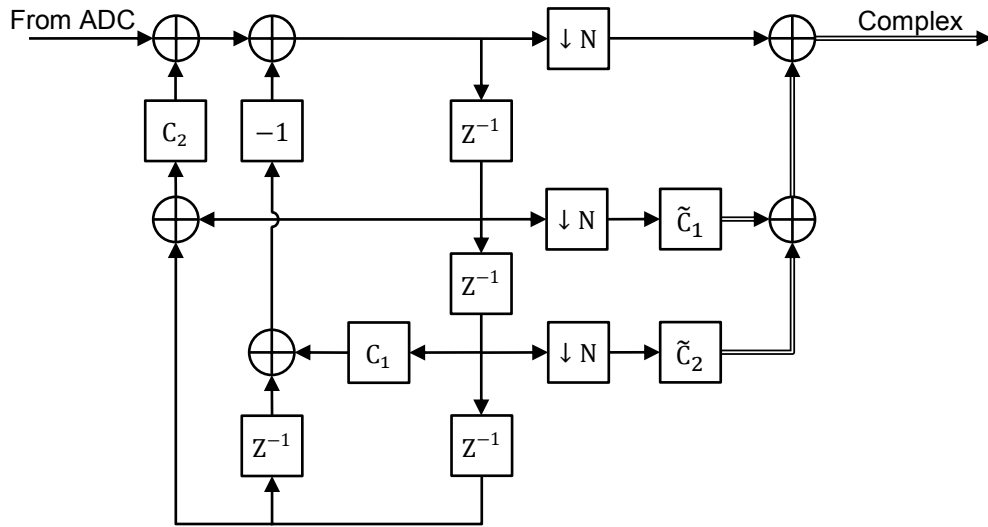
$$\tilde{C}_2 = e^{-j4\pi \frac{k}{N}}, \quad (4.94)$$

$$C_1 = 2 \left( \cos\left(4\pi \frac{k}{N}\right) + \cos\left(4\pi \frac{d}{N}\right) + 1 \right), \quad (4.95)$$

$$C_2 = 4 \cos\left(2\pi \frac{k}{N}\right) \cos\left(2\pi \frac{d}{N}\right). \quad (4.96)$$

The transfer function (4.92) is realized by the difference equation

$$\begin{aligned} y(n) = & x(n) - \tilde{C}_1 x(n-1) + \tilde{C}_2 x(n-2) \\ & + C_2(y(n-1) + y(n-3)) - C_1 y(n-2) \\ & - y(n-4). \end{aligned} \quad (4.97)$$



**Figure 4.24:** Signal diagram of the modified Goertzel filter.

The realization structure in Figure 4.24 is the application of the modified Goertzel filter as a downconverter according to the previous discussion. It was expected that the derived structure will be twice more complex than the original Goertzel downconverter, but the  $C_2$  coefficient can be applied after summation, which saves one multiplication in the recursive part. The resulting number of multiplication in the recursive part is then 2 and equals to the number of multiplication in the structure using the original Goertzel algorithm applied to the window weighted signal. Of course, the variability of the frequency characteristic of the modified filter cannot compete with the numerous set of

weighting windows, but has the undisputed advantage in memory savings and implementation simplicity.

### 4.3. Digital downconverter performance

The proposed algorithms are compared with the basic 1<sup>st</sup> order CIC-filter downconverter from Figure 4.1, which is used in the Electra SDR. One can be interested in two important characteristics, first, the influence on BER of the processed signal and second, the implementation complexity and hardware demands.

#### 4.3.1. BER influence

As has been discussed in chapter 4.1, the filtration property of the downconverter combined with decimation has the main important influence on signal degradation. Decimation always causes aliasing and there is no way to prevent it, except for filtration. Filters attenuate undesired components to some residual power. Since the frequency characteristics of filters are analytically known, it is possible to find the theoretical influence of aliasing on the desired signal and subsequently on the BER.

The frequency characteristics of the CIC-filter and the Goertzel filter are the same, so its comparison is not relevant; we perform a comparison between the Goertzel filter and its modified version in the first step. Secondly, the filters are compared with a half decimation factor, where the half sampling frequency reduction, but significantly lower aliasing, is expected. In Figure 4.25 there are depicted two aliasing situations for the Goertzel filter ( $N = 8, k = 0$ ). It is obvious, that the side lobes contribute to aliasing noise several fold less in the case of the half decimation factor (right plot). In chapter 4.1 we calculated the signal to noise performance also for the half signal bandwidth; the results were improved and we also mentioned that the aliasing performance is even better, now, from the figures, it is obvious why. The characteristic of the pass band is flatter and aliasing is lower near the center frequency. The Goertzel algorithm characteristic has zeros periodically located in such a way that they, after decimation, fit right to the center of the desired band and they have an essential impact on aliasing noise performance. The modified algorithm has different positions of zeros as is seen in Figure 4.22 and aliasing performance should be expected significantly worse even if it, from a global view, looks better.

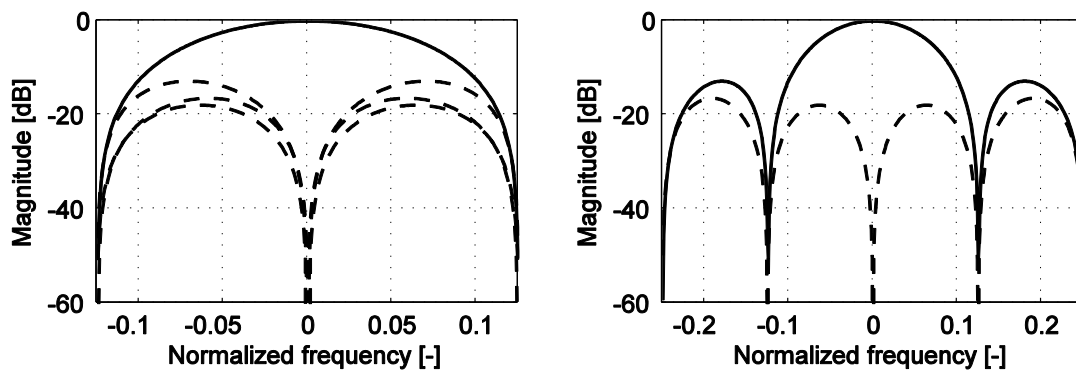


Figure 4.25: Downconverter filter aliasing with decimation factor  $M$  (left) and  $\frac{M}{2}$  (right).

According to the analysis, we based our calculations on Parseval's theorem as in chapter 4.1. The integration over the pass band stays the same

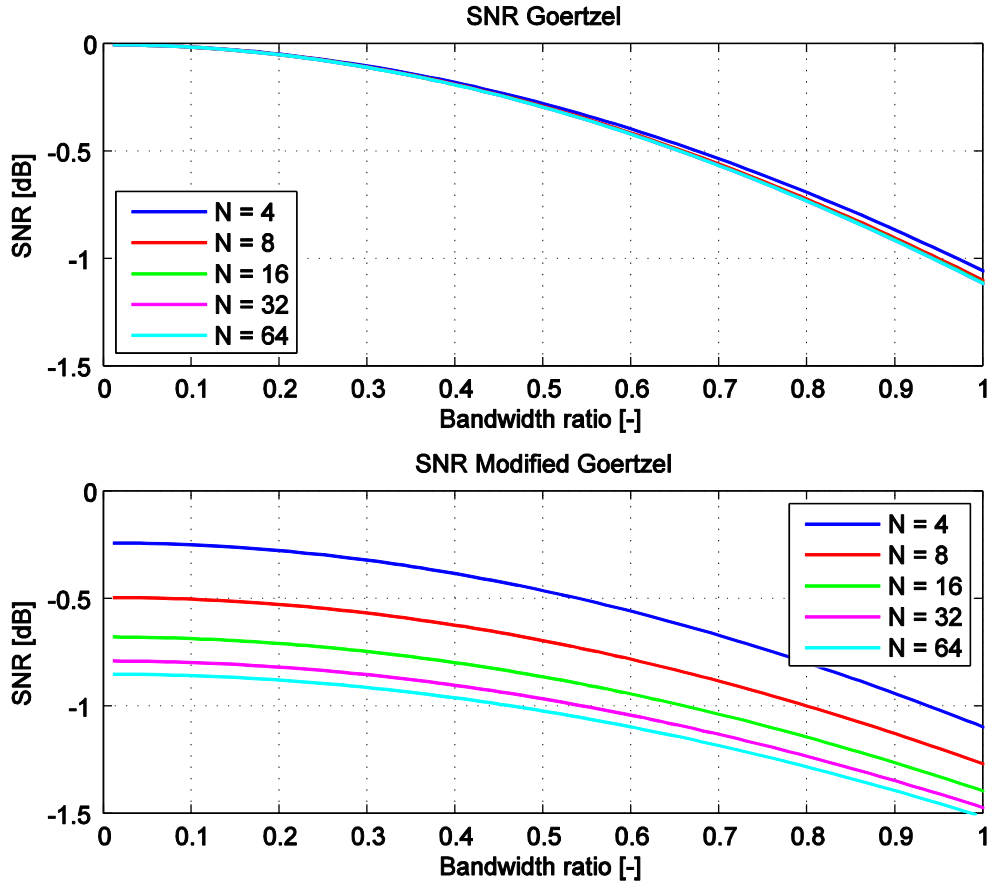


$$S(f_s, N, r) = \frac{1}{N^2 \cdot f_s} \int_{-\frac{r \cdot f_s}{2}}^{\frac{r \cdot f_s}{2}} |H(f)|^2 df \quad (4.98)$$

and analyzed parameters are the desired bandwidth (bandwidth ratio)  $r$  and the filter length  $N$ , which in this case equals the decimation factor. The aliasing noise has to be expressed differently now. When the bandwidth ratio is lower than 1, the aliased spectra does not contribute to aliasing noise of the desired signal by its whole bandwidth and the excess has to be excluded. For the Goertzel filter and its modified form, we can easily tune the filter during integration and calculation, finally, it has the form

$$N_G(f_s, N, r) = \frac{1}{N^2 \cdot f_s} \sum_{k=0}^{N-1} \int_{-\frac{r \cdot f_s}{2}}^{\frac{r \cdot f_s}{2}} |H_G(f, k, N)|^2 df. \quad (4.99)$$

The integration is calculated over the same bandwidth of all discrete frequencies ( $k \in \mathbb{Z}$ ).



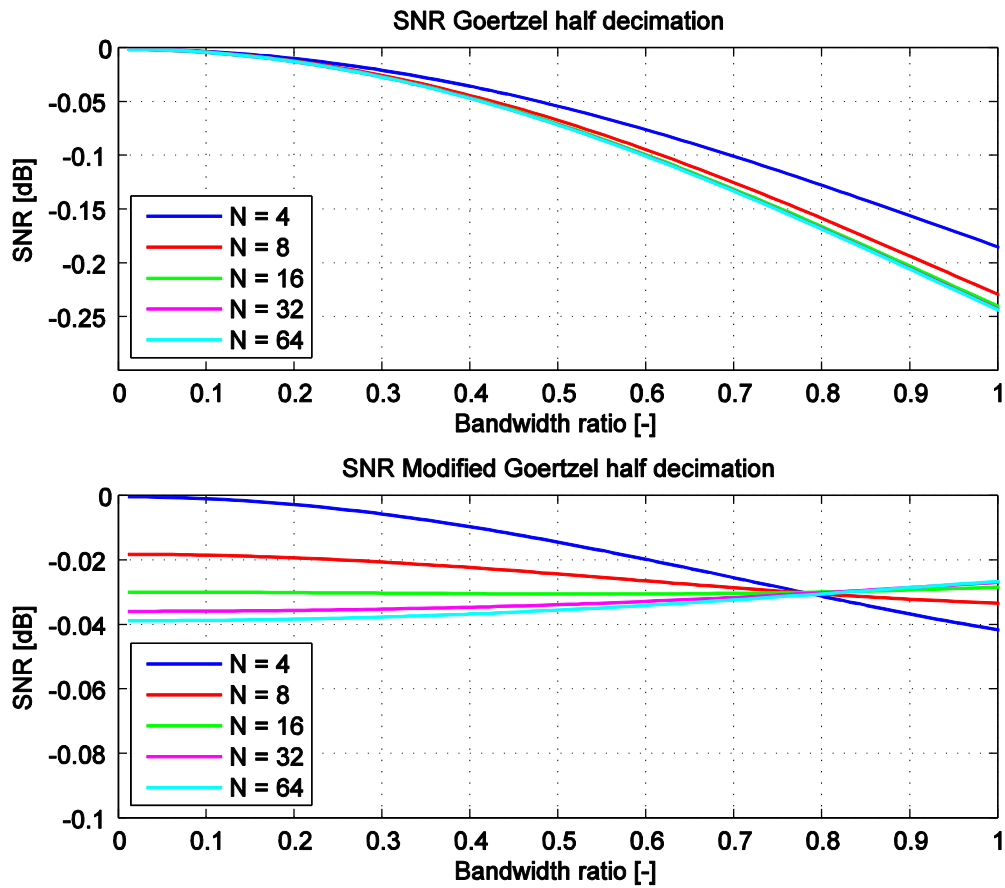
**Figure 4.26:** Aliasing noise performance of the Goertzel and modified Goertzel downconverter.

The results for the Goertzel downconverter and the modified Goertzel filter downconverter are depicted in Figure 4.26. The signal  $S$  and noise  $N_G$  powers are evaluated as a ratio

$$SNR = \frac{S}{N_G} \quad (4.100)$$

and plotted in decibels as a value of BER degradation.

The results say that the bandwidth ratio significantly influences BER degradation and is the lowest for narrowband signals. The Goertzel length does not influence the results significantly. If we look at the modified Goertzel downconverter, the results are completely different. The pass lobe of the filter is wider and worsens BER performance due to its main contribution to the aliasing noise power. The different location of zeros of the magnitude characteristic causes the aliasing noise to spread and the bandwidth ratio then does not have a very helpful impact. The differences for each length  $N$  are also caused by the different zero locations.



**Figure 4.27:** Aliasing noise performance of the Goertzel and modified Goertzel downconverter for the half decimation factor.

The strong influence of the wide side lobe of the modified Goertzel downconverter supports our assumption that using the half decimation factor may cause a significant improvement in performance. The calculation of the aliasing noise power has to be modified again. The noise contribution comes newly from every second integer frequency of the Goertzel filter and its modified version

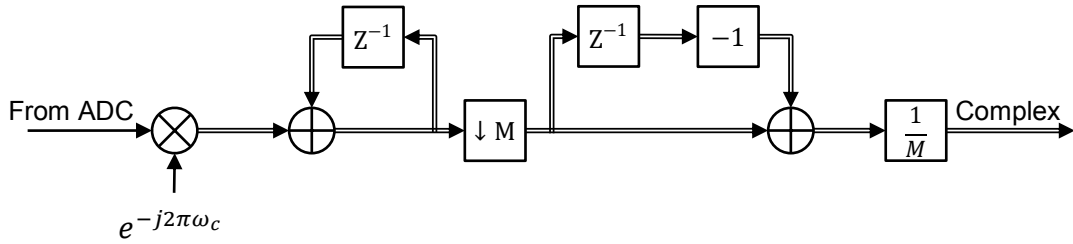
$$N_G(f_s, N, r) = \frac{1}{N^2 \cdot f_s} \sum_{k=0}^{\frac{N}{2}-1} \int_{-\frac{r \cdot f_s}{2}}^{\frac{r \cdot f_s}{2}} |H_G(f, 2k, N)|^2 df. \quad (4.101)$$

The results are shown in Figure 4.27. The performance of the Goertzel filter is significantly better, but its character did not change. The situation is different for the modified filter, the influence of the main lobe has been removed and better suppression of the side lobes is obvious. BER degradation stays very low over the whole bandwidth ratio interval and significantly surpasses the Goertzel downconverter for higher bandwidths.

#### 4.3.2. Computational demands

Once the target platform, where the downconverter is planned to be implemented, the computational demands of each algorithm have to be known. One of the assessment methods is the evaluation of multiplication and addition operation demands. The actual demands can then easily be judged over lots of platforms including microcontrollers and gate arrays.

The number of each operation was found from the signal diagram of each algorithm. This way is more illustrative and the difference equations do not include the decimation operation, which significantly affects results. First, to find the results for the CIC downconverter it is better to modify its diagram in Figure 4.1 to the complex form in Figure 4.28. The double lines depict complex paths.



**Figure 4.28:** CIC decimator in complex representation.

The multiplication of complex and real values commonly needs two purely real multiplications, the addition of two complex addends uses two purely real additions, and the addition of real and complex addends uses only one real addition. A purely complex multiplication is not performed by any tested algorithm.

The results are summarized in Table 4.5. The number of operations is evaluated for one output sample and the convention of decimation factor designation is kept unchanged, but  $M = N$ . The CIC decimator results do not include carrier replica generation; a look-up table or another type of discrete oscillator will cause additional demands. On the other hand, the proposed algorithms need additional downconversion on the lower sample rate as discussed in chapter 4.2.

The results show that the original Goertzel algorithm is more effective from the number of operations point of view. Portable devices usually have limited resources and the efficiency of the signal processing algorithms may be crucial. The Goertzel algorithm is able to reduce sampling rate, which is important especially in sequential

systems like microcontrollers. The downfall of the improved algorithms is their complexity, but the benefit is better characteristics.

**Table 4.5:** Downconverter computational demands.

Downconverter type	Multiplications	Additions	Memory space
CIC	$2M + 2$	$2M + 2$	$5^1$
Goertzel	$N + 2$	$2N + 1$	5
Modified Goertzel	$2N + 4$	$4N + 3$	10
Goertzel – half decimation	$2N + 4$	$4N + 2$	$N/2 + 10$
Modified Goertzel – half decimation	$4N + 8$	$8N + 6$	$N/2 + 20$

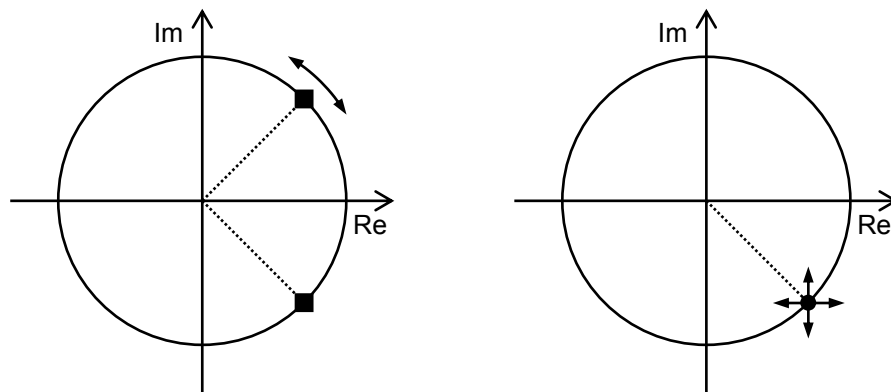
#### 4.3.3. Quantization errors of the algorithms

The implementation on real systems suffers from limited accuracy of number representation. The general issues of overflowing and rounding errors are strongly dependent on the final implementation, but we can find general influence of the coefficient quantization error.

To find the influence of Goertzel filter coefficient quantization, let us have a look at the transfer function (4.20), which directly corresponds to the realization structure in Figure 4.7. The function has two poles from which one is eliminated by a zero. The error of  $C$  coefficient  $\varepsilon$  causes misalignment of the poles

$$1 - 2(C + \varepsilon)z^{-1} + z^{-2} = (1 - e^{j \cos^{-1}(C+\varepsilon)})(1 - e^{j \cos^{-1}(C+\varepsilon)}), \quad (4.102)$$

but its movement is limited to the unity circuit for relatively small errors, which do not make the coefficient be out of the range  $\langle 1; -1 \rangle$  as depicted in Figure 4.29. It means that the quantization error causes frequency tuning error.



**Figure 4.29:** Goertzel filter poles (left) and zero (right) misalignment due to quantization error.

It is common to implemented the coefficient of poles  $C$  as a signed fractional number and the multiplication by 2 is implemented within the code according to

<sup>1</sup> The memory space does not include the carrier signal replica (LUT, harmonic generator etc.).

$$C = \cos\left(2\pi \frac{k}{N}\right) = \cos(\alpha). \quad (4.103)$$

This is different from the previous declaration of  $C$  in Figure 4.7. The real position of the pole, which is compensated by the zero, is expressed by

$$e^{-j \cos^{-1}(C+\varepsilon)} = \cos(\cos^{-1}(C + \varepsilon)) - j \sin(\cos^{-1}(C + \varepsilon)). \quad (4.104)$$

If the complex coefficient of the zero  $\tilde{C}$  is implemented separately for the real and imaginary parts, the complex quantization error will affect the zero position according to

$$\tilde{C} = e^{-j2\pi \frac{k}{N}} + \tilde{\varepsilon} = (\cos(\alpha) + \text{Re}\{\tilde{\varepsilon}\}) - j(\sin(\alpha) + \text{Im}\{\tilde{\varepsilon}\}). \quad (4.105)$$

The real part of the complex coefficient  $\tilde{C}$  is identical to the real coefficient  $C$ , and therefore we can assume that the implementation of these coefficients are also identical or even the real part of the  $\tilde{C}$  coefficient is used instead of the  $C$  coefficient with improved memory savings. This assumption implies that the error  $\varepsilon$  equals the error  $\text{Re}\{\tilde{\varepsilon}\}$ . Now we can evaluate the misalignment of the pole and its compensation zero as an error vector

$$\begin{aligned} \vec{e} &= \cos(\cos^{-1}(C + \varepsilon)) - j \sin(\cos^{-1}(C + \varepsilon)) - (\cos(\alpha) + \text{Re}\{\tilde{\varepsilon}\}) \\ &\quad - j(\sin(\alpha) + \text{Im}\{\tilde{\varepsilon}\}) \\ &= \cos(\alpha) + \text{Re}\{\tilde{\varepsilon}\} - j \sin(\cos^{-1}(\cos(\alpha) + \text{Re}\{\tilde{\varepsilon}\})) - \cos(\alpha) \\ &\quad - \text{Re}\{\tilde{\varepsilon}\} + j(\sin(\alpha) + \text{Im}\{\tilde{\varepsilon}\}) \\ &= j \left( \sin(\alpha) + \text{Im}\{\tilde{\varepsilon}\} - \sqrt{1 - (\text{Re}\{\tilde{\varepsilon}\} + \cos(\alpha))^2} \right). \end{aligned} \quad (4.106)$$

The error vector says that the mentioned way of implementation minimizes the zero from the pole misalignment only to the imaginary dimension. The function has its extremes at the edges of the sampled frequency band.

From the analysis of the frequency tuning error caused by the quantization error of the coefficient  $C$  we can find the relation

$$e_f = \frac{\cos^{-1}\left(\cos\left(2\pi \frac{k}{N}\right) + \varepsilon\right)}{2\pi}. \quad (4.107)$$

The relation simply finds the real tuning frequency, which is distorted by the quantization error, by reverse calculation.

The function is plotted together with the simulation of the real filter in Figure 4.30. It is obvious that the most critical region is also near the edges of the sampled frequency band. A possible way to deal with this issue is pre-filtering as published in [59].

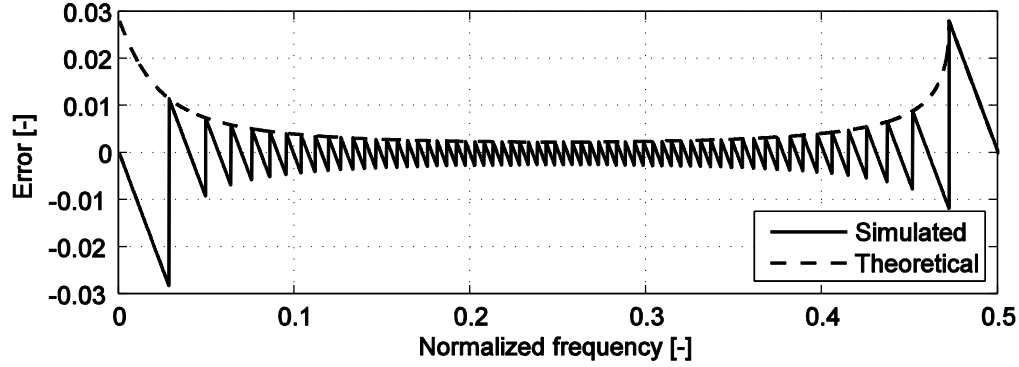


Figure 4.30: Absolute quantization error for coefficient  $C$ .

#### 4.4. Simulink tests

The tests of the proposed downconverters and of the derived performance analysis were achieved by simulations in MATLAB Simulink. Each testing model has more or less been composed from the available design blocks. The following text describes each model and at the end of this section the results are summarized and discussed.

An assessing criterion is the sensitivity of the RF system to SNR performance. The chain of the general tested model is depicted in Figure 4.31. A random signal modulated onto the RF carrier was put through a radio channel and then processed by the tested item, i.e. the demodulator block. The “received” data was compared to the original copy and the Bit Error Rate was estimated.



Figure 4.31: General schematic of the tested model.

##### 4.4.1. Modulator model

The tests were focused on testing BPSK modulation. A schematic of the Simulink model of the modulator is depicted in Figure 4.32. The model is based on common theoretical interpretation of BPSK modulation, i.e. symbol mapping and complex multiplication of the baseband signal with the carrier.

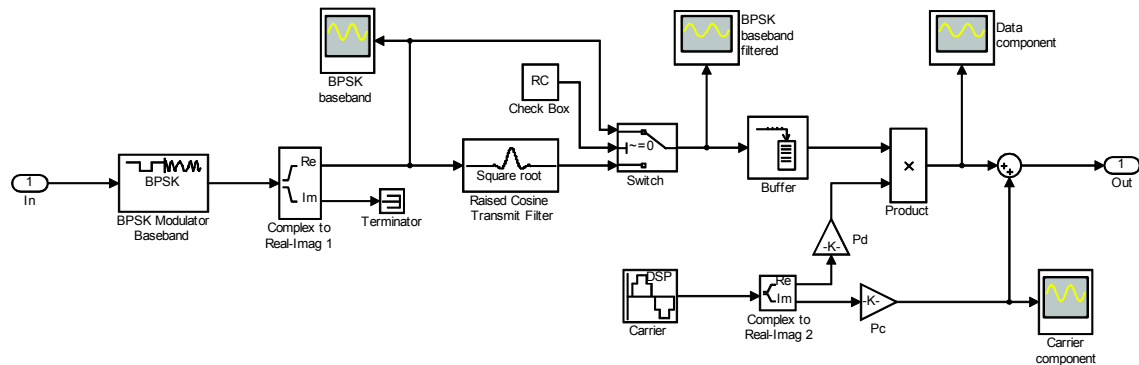


Figure 4.32: BPSK modulator model.

In Simulink there exists a model for BPSK symbol mapping under the name BPSK Baseband Modulator. Due to the real character of the BPSK baseband signal, the zero imaginary part of the generally complex output of the modulator block is omitted. Before upconversion the baseband signal is usually filtered to suppress undesired frequency components. The filter shape used for our tests is the Raised-cosine (RC) filter, where filtration is applied only on the transmitter side, and the Square-root Raised-cosine (SRRC) filter shape, where the same shape has to be applied at both the transmitter and the receiver sides. The filter may be bypassed by a switch. Finally, the baseband signal is upconverted with a specific modulation index which defines the residual carrier power.

The BPSK signal with a suppressed carrier was defined in (2.22) and we can rewrite it into the form

$$s_{BPSK}(t) = \sqrt{2P} \sin(\omega_c t + \beta m(t) + \varphi_c) \quad (4.108)$$

maintaining the symbol function  $m(t)$  be the same taking on digital values  $\pm 1$ .  $\beta$  is then called the modulation index and in this case obviously equals its maximal value  $\pi/2$  rad.

When the modulation index  $\beta$  is less than  $\pi/2$  rad, the signal becomes BPSK modulated with the residual carrier. The carrier and data signal components may be split by trigonometry from (4.108) resulting in

$$s_{BPSK}(t) = \sqrt{2P} \cos(\beta) \sin(\omega_c t + \varphi_c) + \sqrt{2P} \sin(\beta) m(t) \cos(\omega_c t + \varphi_c). \quad (4.109)$$

Obviously the carrier power is then defined by

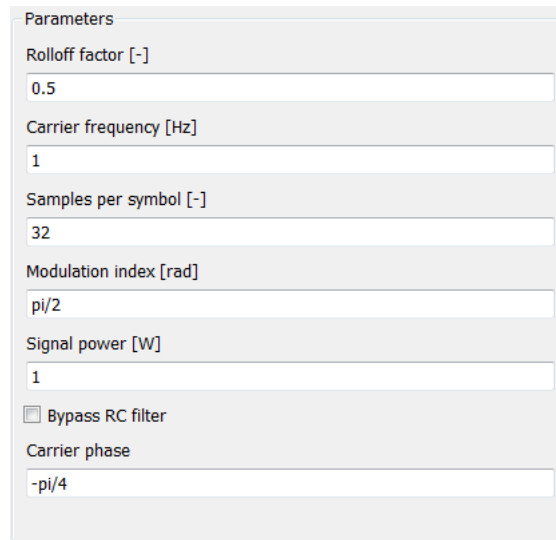
$$P_c = P \cos(\beta)^2 \quad (4.110)$$

and data power by

$$P_d = P \sin(\beta)^2. \quad (4.111)$$

This definition of the modulation index is in accordance with [1] and the Simulink model is able to provide BPSK signals with an adjustable modulation index by converting  $\beta$  to the data carrier gain  $P_d$  and the residual carrier gain  $P_c$ .

The modulator is packed into a subsystem and a mask is assigned to it. Figure 4.33 shows the mask by which the modulator settings are provided. The parameters need not be described closely, just note that each parameter may be set by an immediate value or a variable defined in the MATLAB console.



Parameters

Rolloff factor [-]

0.5

Carrier frequency [Hz]

1

Samples per symbol [-]

32

Modulation index [rad]

pi/2

Signal power [W]

1

☐ Bypass RC filter

Carrier phase

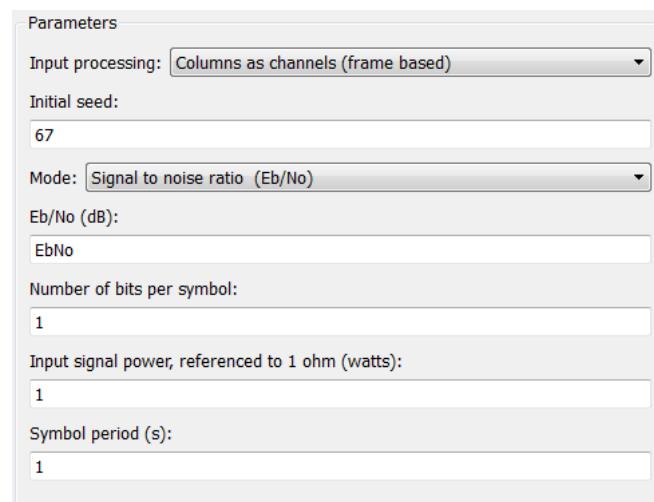
-pi/4

**Figure 4.33:** Modulator settings.

#### 4.4.2. RF channel

An AWGN channel (Added White Gaussian Noise) which is suitable for general simulations of noise performance is used as the simulation model of the RF channel. The model is available in the Simulink library and Figure 4.34 shows its settings window (mask).

The first parameter, an initial seed, sets the initial number of a random number generator. It is not an important parameter for our tests compared with the others. The signal to noise parameter  $E_b/N_0$  corresponds to the energy per bit to noise power spectral density ratio in decibels. For BPSK modulation the number of bits per symbol is naturally one. The modulator model is set to generate a signal of 1 W of effective power, so the signal power should then be set to 1 W. The symbol period is given by the symbol period in the data generator block and is set to 1 s.



Parameters

Input processing: Columns as channels (frame based)

Initial seed:

67

Mode: Signal to noise ratio (Eb/No)

Eb/No (dB):

EbNo

Number of bits per symbol:

1

Input signal power, referenced to 1 ohm (watts):

1

Symbol period (s):

1

**Figure 4.34:** AWGN channel settings.



#### 4.4.3. Downconverter model based on CIC-filter

The model in Figure 4.35 is in compliance with the concept in Figure 4.1. The input signal is firstly multiplied by the carrier replica. The following half-band filter is an equiripple 43-order FIR filter as has been mentioned in section 4.1. The CIC-decimation block is followed by a baseband BPSK demodulator, which is the inversion to the BPSK modulator. The last two blocks just help to convert the data to the desired format. Between the half-band filter and the CIC-decimator is inserted a delay block, which compensates the filters group delay to maintain symbol synchronism.

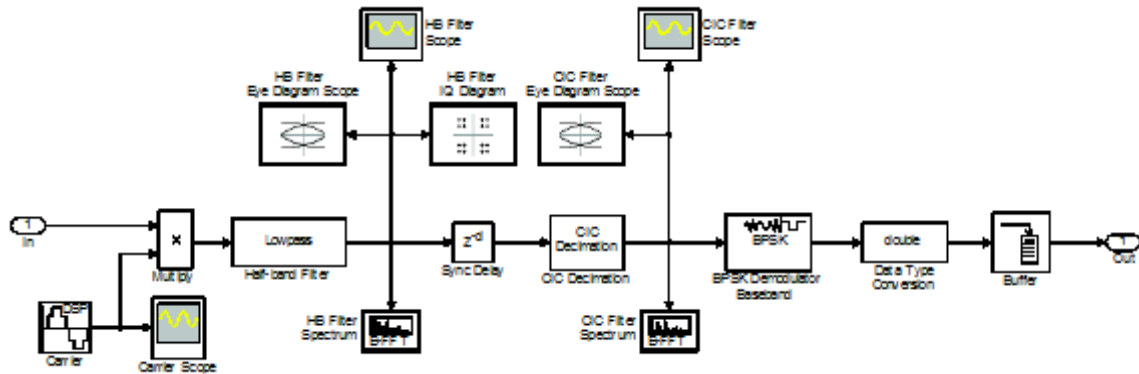


Figure 4.35: CIC-filter based downconverter.

All necessary parameters are settable through the mask shown in Figure 4.36.

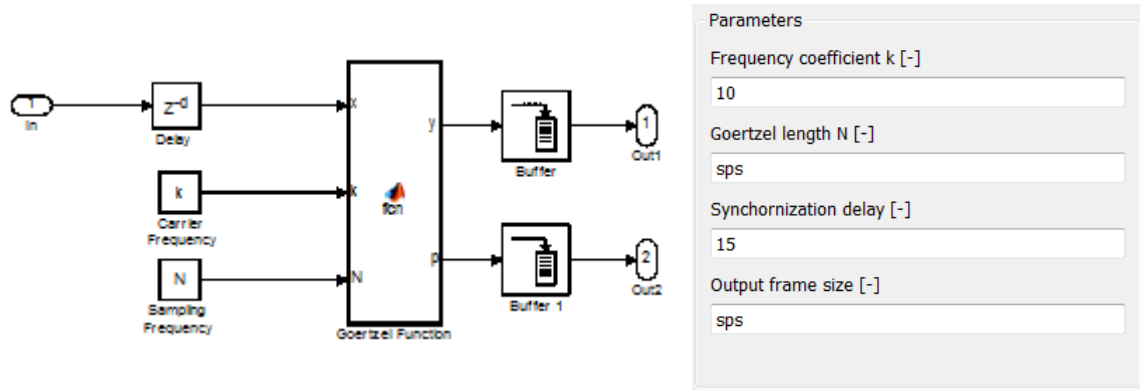
Parameters	
Carrier frequency [Hz]	10
Samples per symbol [-]	sps
Synchronization delay [-]	26
Output frame size [-]	sps

Figure 4.36: CIC-filter based downconverter settings.

#### 4.4.4. Downconverter model based on the Goertzel algorithm

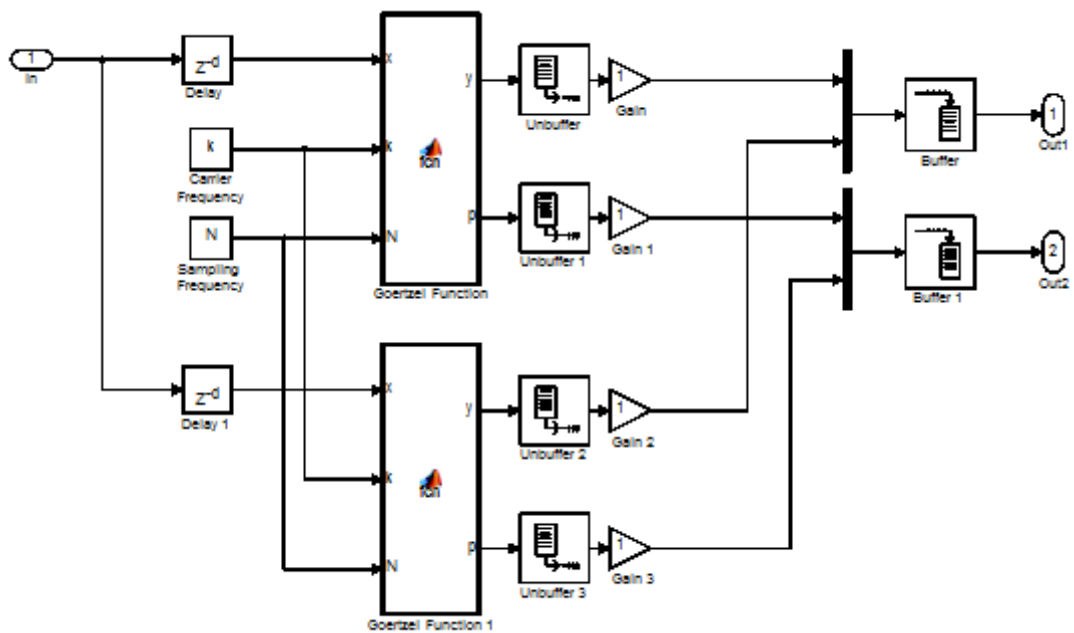
There are more possibilities of the Goertzel downconverter realization. The choice made for our purpose is a MATLAB function block. The core of the code had already existed, only small changes were performed. The Goertzel downconverter model does not include the BPSK demodulator, which has to be postponed together with the data type conversion and the buffer as in Figure 4.35. The parameters of the Goertzel algorithm are set by the constant settable in the mask. The downconverter outputs the complex downconverted signal and its compensated power.

The mask of the Goertzel downconverter is shown in Figure 4.37. Tuning is very similar to the CIC-filter downconverter. The Goertzel length  $N$  sets the bandwidth of the filter the same way as the Samples per symbol parameter in Figure 4.36 does. The frequency coefficient  $k$  is relative to the Goertzel length just as the Carrier frequency of the previous model is relative to the sampling frequency.



**Figure 4.37:** Goertzel filter based downconverter (left) and its setting window (right).

The code of the Goertzel downconverter is shown in Listing 4.1. After setting the working arrays, the complex and real coefficients are computed. A computing loop follows, where all iterations are done. After  $N$  iterations, each output sample is processed to the complex output signal according to the Fourier transform output and additionally to the power of the signal. Both output values are also normalized to the unity gain.



**Figure 4.38:** Goertzel downconverter of half decimation factor.

The implementation of the half decimation factor version is depicted in Figure 4.38. The two identical Goertzel downconverters are parallelized with  $N$  and  $N/2$  delay and then sample-by-sample put into the output.

```

function [y,p] = fcn(x, k, N)

size = 32;
persistent win;
if isempty(win)
    win = rectwin(size);
end
size = length(x)/size;
y = complex(zeros(size, 1), zeros(size, 1));
p = zeros(size, 1);

C = 2*cos(2*pi*k/N);
W = cos(2*pi*k/N) - 1i*sin(2*pi*k/N);

A = (1 - exp(-1i*4*pi*k))./(1 - exp(-1i*4*pi*k/N));
AR = real(A);
AI = imag(A);

QM1 = 4*(AR.^2 + AI.^2 + N^2 - 2*AR*N)./((AR.^2 + AI.^2 - N^2).^2);
QM2 = 4*(AR.^2 + AI.^2 + N^2 - 2*N*(AR.*cos(4*pi*k/N) + ...
    AI.*sin(4*pi*k/N)))./((AR.^2 + AI.^2 - N^2).^2);
QM3 = 4*(AR.^2 + AI.^2 + N^2 - 2*N*(AR + ...
    AI.*sin(2*pi*k/N)./cos(2*pi*k/N)))./((AR.^2 + AI.^2 - N^2).^2);

for a = 0:N:length(x) - N
    z1 = 0;
    z2 = 0;
    for n = 1:N
        s = (x(a + n)*win(n)) + C*z1 - z2;
        z2 = z1;
        z1 = s;
    end
    y(a/N + 1) = (z1 - W*z2)/(N/2);
    p(a/N + 1) = QM1*z1*z1 + QM2*z2*z2 - QM3*C*z1*z2;
end

```

**Listing 4.1:** Goertzel function code.

```

d = .5*N/(N + 1);
W1 = exp(-1i*2*pi*k/N)*2*cos(2*pi*d/N);
W2 = exp(-1i*4*pi*k/N);
C1 = 2*(cos(4*pi*k/N) + cos(4*pi*d/N) + 1);
C2 = 4*cos(2*pi*k/N)*cos(2*pi*d/N);

for a = 0:N:x_len - N
    z1 = 0;
    z2 = 0;
    z3 = 0;
    z4 = 0;
    for n = 1:N
        s = x(a + n) + C2*(z1 + z3) - C1*z2 - z4;
        z4 = z3;
        z3 = z2;
        z2 = z1;
        z1 = s;
    end
    y(a/N + 1) = (z1 - W1*z2 + W2*z3)*4/((sin(pi*d)/sin(pi*d/N))^2);
end

```

**Listing 4.2:** Modified Goertzel function code.

Listing 4.2 shows the code part of the modified Goertzel downconverter version. The definition of the coefficients is different including the frequency difference  $d$ . The processing loop then contains two more poles and one more zero.

#### 4.4.5. BERtool models

BERtool is a simple MATLAB tool for bit error rate simulation of communication systems. It is possible to plot and export simulation results as the bit error ratio dependent on the signal energy to signal power ratio. The theoretical functions for basic modulations are implemented, which for our purpose serve as a reference. The BER function for BPSK modulation was found to be

$$BER = Q\left(\sqrt{2 \cdot 10^{\frac{E_b}{N_0}}}\right) = \frac{1}{2} \operatorname{erfc}\left(\sqrt{10^{\frac{E_b}{N_0}}}\right) \quad (4.112)$$

where the  $Q(x)$  function is a probability function of the Gaussian probability distribution and  $\operatorname{erfc}(x)$  is a complementary Gaussian error function.

BER performance of the user system may be tested by a Monte Carlo analysis which uses Simulink models of the system. The model has its own data for testing, in our case the data is generated by a random integer generator block. These are put through the system under test, and then the input and the output data are compared to evaluate BER. The output of the Error Rate calculation block contains the value of the desired BER, the number of tested bits and the number of errors. These values are displayed as well as saved to the workspace variable, which is a response for BERtool. BERtool controls the model through workspace variables.

- $EbNo$  – sets the desired power of noise in the AWGN block
- $maxNumBits$  – sets the simulation stop condition of maximum tested bits, important when low values of BER are reached
- $maxNumErrs$  – sets the simulation stop condition of maximum error occurrence, important when high values of BER are reached

The model in Figure 4.39 is a basic BPSK baseband system, which should follow the theoretical BER characteristic.

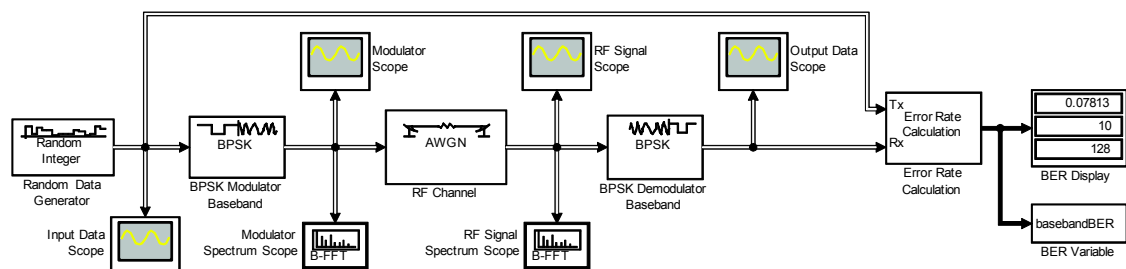
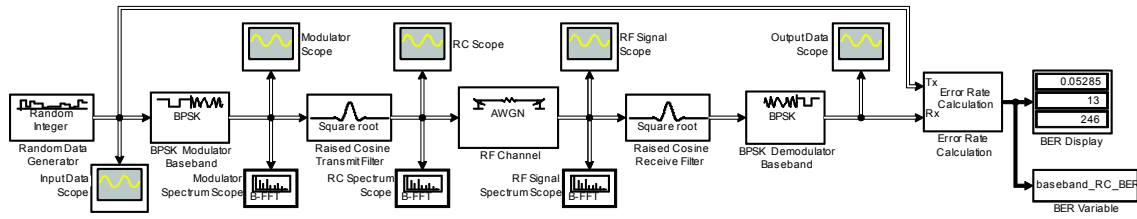


Figure 4.39: BERtool model of baseband BPSK modulator.

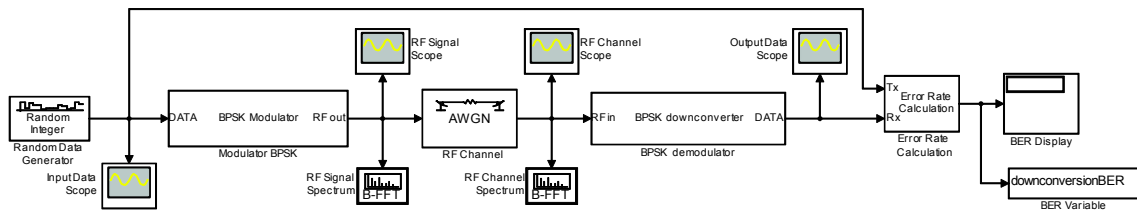
The influence of the Raised-cosine filter was also tested; the model is depicted in Figure 4.40. The implemented filter is the square-root type, therefore two filters are used, one on the transmitter side and one on the receiver side. The filter transfer function is designed to suppress side lobes of the BPSK spectral density characteristic,

while maintaining the eye-diagram as wide as possible; the BER characteristic should then be close to the ideal one.

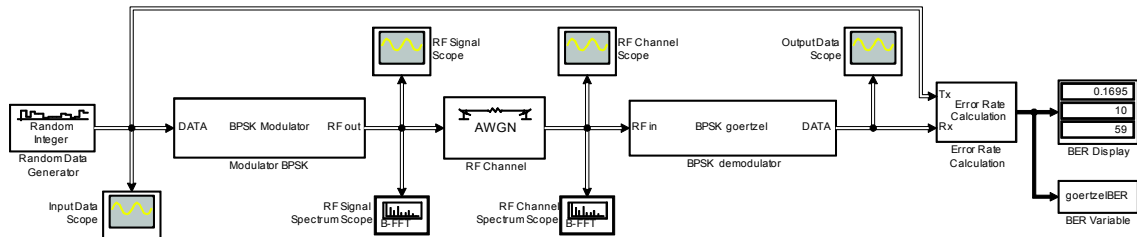


**Figure 4.40:** BERtool model of baseband BPSK modulator with Square-Root Raised Cosine filter.

In Figure 4.41 we already see the modulator with the upconverter on the transmission side and the CIC-filter based downconverter previously described on the receiver side. In Figure 4.42 there is depicted the model for the Goertzel algorithm based downconverter simulations.



**Figure 4.41:** BERtool model of CIC-filter based downconverter.



**Figure 4.42:** BERtool model of Goertzel downconverter.

The models for the modified Goertzel downconverter and for the version of half decimation factor are depicted in Figure 4.43 and Figure 4.44 respectively. The models were not tested by BERtool, but a special script to see the difference from the original Goertzel downconverter immediately. Since the half decimation factor version gives twice more samples than the symbol rate, the downconverter must be followed by an additional decimator consisting of a half-band filter and a downsampling module.

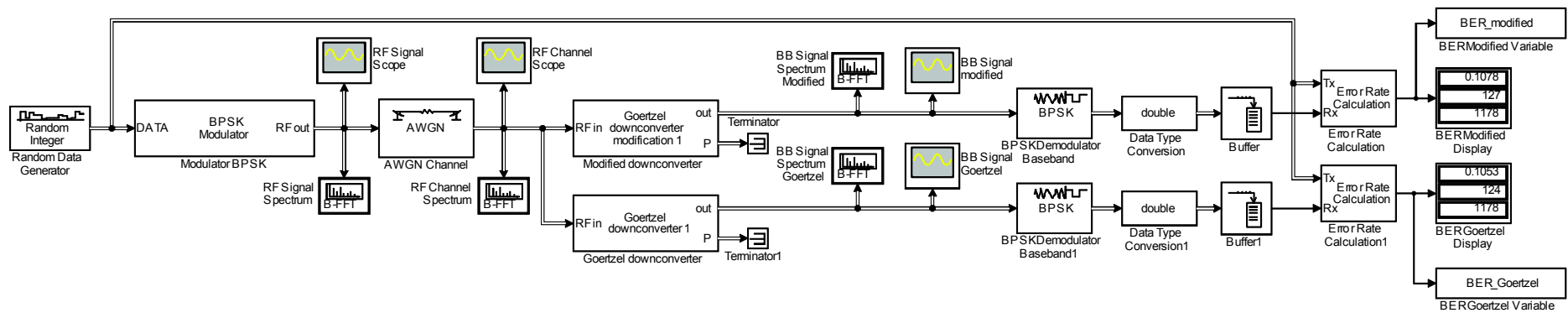


Figure 4.43: Modified Goertzel downconverter BER model.

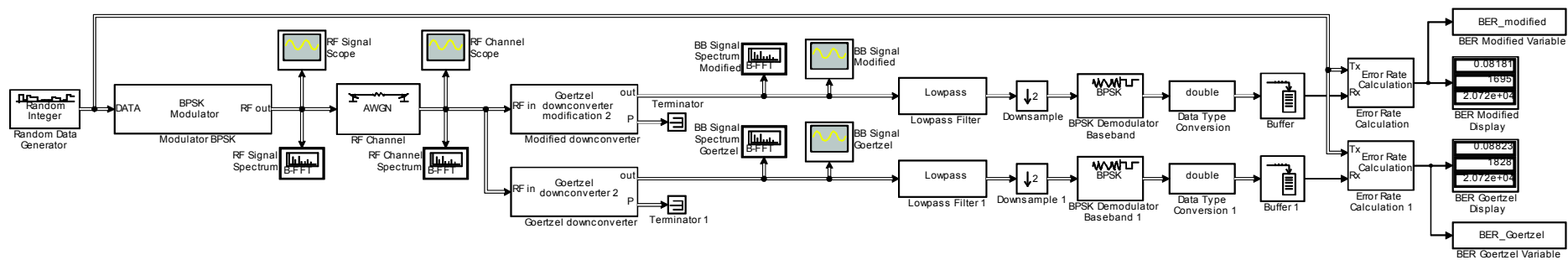
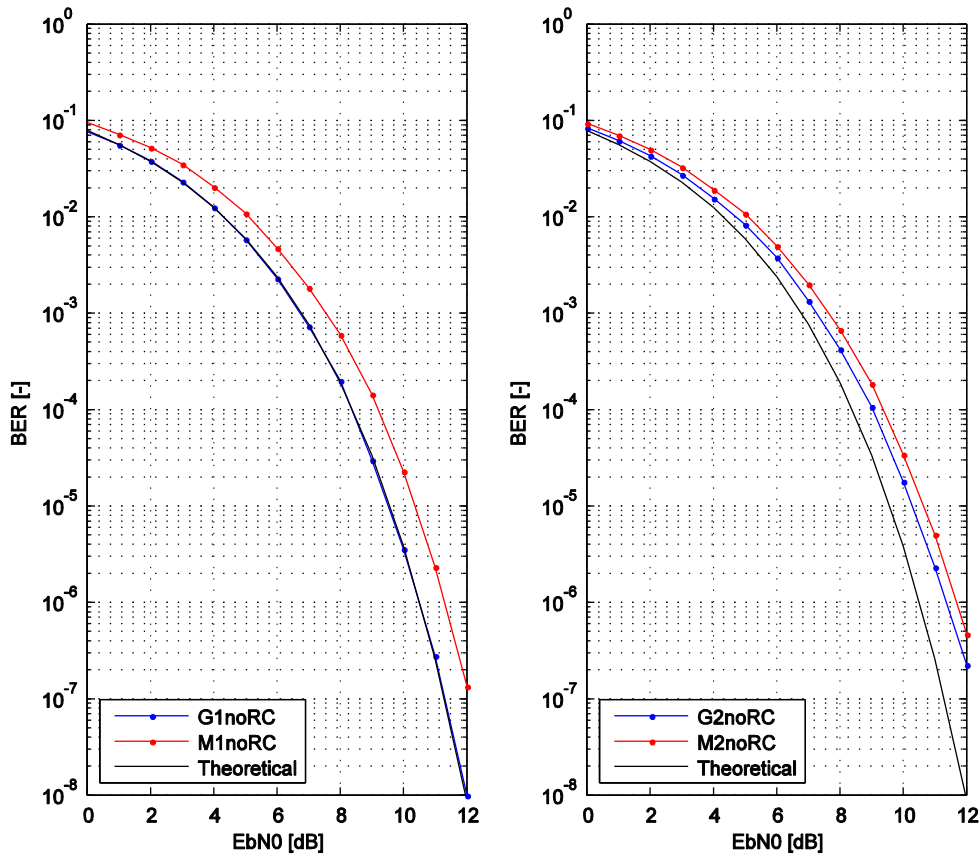


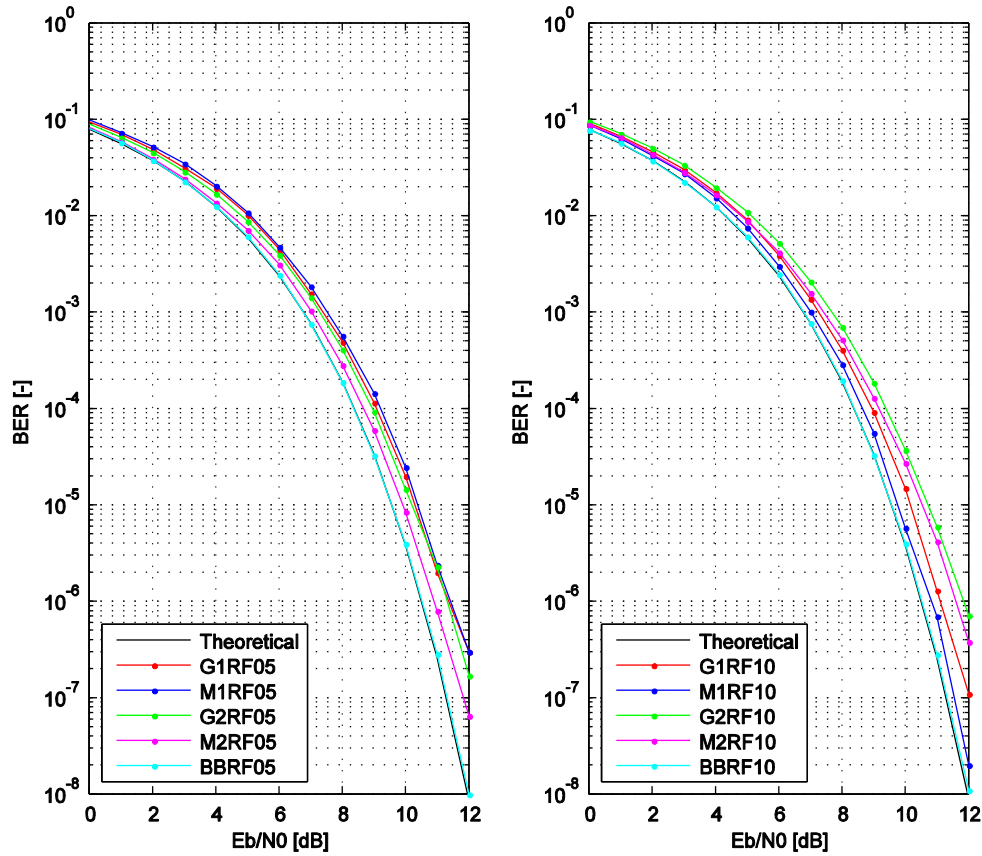
Figure 4.44: Modified Goertzel downconverter BER model of half decimation factor.

#### 4.4.6. BER simulation results

The BER characteristics of the proposed downconversion algorithms obtained by simulation are shown in the following figures. In Figure 4.45 there are shown characteristics for the proposed Goertzel and modified Goertzel downconverters in two versions dependent on the used main lobe width. The results may be surprising because they are in strong disagreement with the theoretical assumption in chapter 4.3.1. However, the explanation is clear. In the theoretical analysis we supposed the out-of-band signals to be noise with a destructive effect on the in-band signal, but the unfiltered BPSK signal has a significant part of its power out of the downconverted band. The full-band Goertzel downconverter has the filtration characteristic somehow “matched” to this power spectral density, because it comprises the rectangular window of the same width as the input signal symbol period, and therefore aliasing is partly constructive. The modified version of the Goertzel downconverter has better attenuation in the stop-band and has zeros located differently which effectively suppresses the out-of-band partly constructive components and consequently worsens the overall BER characteristic. Additionally, since the signal power is spread, after filtration the  $E_b/N_0$  ratio is worse than expected. The same effect works for the half-band versions with the difference that the Goertzel downconverter filtration characteristic is no longer matched to the input signal.



**Figure 4.45:** BER simulation results of the Goertzel and modified Goertzel downconverters with no matched filter on the transmission side. The full-band version on the left hand side and the half-band version on the right hand side.



**Figure 4.46:** BER simulation results of the Goertzel and modified Goertzel downconverters and both full-band and half-band versions with the Raised-Cosine matched filter on the transmission side. The simulation for the filter with the roll-off factor 0.5 on the left hand side and for the roll-off factor 1.0 on the right hand side.

The simulation results are obviously different in Figure 4.46, where the Raised-Cosine filter is used on the transmission side and the output power is corrected to the nominal power. In these cases the modified Goertzel downconverter performs better. We suppose that the roll-off factor of values around 0.5 is more desirable than 1.0, because there is sufficient attenuation of the side lobes, and in this case the downconverters perform in expected order. The modified Goertzel half-band downconverter gave us the best results while both full-band downconverters are the worst. The characteristic of the baseband transmission is shown for comparison and checking the power level. The performance of the downconverters with the filter with a roll-off factor of 1.0 is again influenced by constructive spectral leakage caused by weak attenuation of the closest side lobes.

#### 4.5. Summary

- The downconverter employed in the Electra SDR was introduced at the beginning of this section. The main focus was aimed at the analysis of the CIC filter as a benchmark for the planned performance comparison.
- The Goertzel algorithm as an implementation form of the Fourier transform was introduced in a new role of the downconverter providing



bandpass filtration and undersampling based frequency conversion. The algorithm was initially tested on the AM modulation system of DCF77, where it simultaneously served as an AM demodulator by the efficient signal power estimation.

- In the next important section it was shown that the Goertzel algorithm, used as a filter, can be used in a special constellation which allows elimination of the decimation property of the algorithm independently on its filtration characteristics. Further it was predicted and subsequently proved that its property may improve aliasing distortion within the downconverters.
- The undersampling based downconversion suffers from discretization of frequency shifts that are directly dependent on the sampling rate conversion. It was shown that bandpass filtration provided by the Goertzel downconverter is independent of the frequency conversion and may be tuned to whatever in-band frequency. The novel algorithm for the efficient correction of residual frequency shifts after the undersampling downconversion was introduced. The algorithm has the largest importance in signal power estimation of the downconverted signal.
- The analysis of the impulse response of the Goertzel filter showed that there is a possibility to improve its filtration characteristics by smoothing the impulse response by additional recursive branches. Indeed, the combination with the second Goertzel filter and precise tuning at a defined frequency led to a novel filter with improved frequency characteristics.
- The proposed algorithms were analyzed from three main points of view. Firstly, insufficient attenuation of the stopband of the filtration characteristic leads to aliasing during downconversion and it was shown how this contributes to BER performance degradation. Our objective and thus requirement for the novel downconverter was to lower implementation demands. It was shown that, depending on the available resources, the Goertzel algorithm is more efficient. The price paid for better characteristics of the improved versions is a higher need for computational resources compared to the first order CIC filter. Finally, for the implementation itself, it is needed to know the performance of the real implementation with finite number accuracy processing, for this purpose a general analysis was also performed.
- All of the proposed algorithms were modeled in MATLAB Simulink and the models were simulated under real conditions (excluding implementation performance). The discussion of the given results says that the downconverters provide the expected performance under the most useful conditions.

## 5. Carrier synchronization

The correct demodulation of received radio frequency signals lies in the knowledge of carrier parameters. Static systems with a non-critical working environment are satisfied with non-coherent demodulation. In the case of listening to far objects, typically satellites and deep-space probes, which are often moving so fast, that the Doppler frequency shift becomes a significant degrading aspect, using coherent and partially coherent demodulation giving a margin in the detection error performance is inevitable. Such digital systems need special algorithms for carrier parameter estimation.

### 5.1. Cramer-Rao bounds

We assume that the inspected signal can be modeled as

$$r(t) = s(t) + w(t) \quad (5.1)$$

where  $s(t)$  is a general modulated signal and  $w(t)$  is additive white Gaussian noise (AWGN). The task is to find the closest estimation  $\tilde{s}(t)$  from the received signal  $r(t)$ . How close the estimation is to the transmitted signal  $s(t)$  depends on the estimator performance. AWGN makes the received signal  $r(t)$  be a random variable and because the estimation  $\tilde{s}(t)$  is a function of  $r(t)$ , it is also a random variable. The assessment of the estimator performance, therefore, must be done statistically; particularly we have to inspect the variance of the signal estimate through an amount of different noise realizations.

The question now arises concerning the existence of a bound of the lowest achievable estimation variance of an ultimate estimator. Such a bound would be the target performance of the design and the reference against which to compare. An available tool is the Cramer-Rao lower bound (CRB) [45]. Unfortunately, CRB encounters difficulties with synchronization applications, and therefore we restrict our focus on a modified CRB (MCRB) [44] when applying the binary PSK modulated signal.

Assume the transmitted signal to be

$$s(t) = e^{j(2\pi f_c t + \varphi_c)} \sum_k c_k g(t - kT - \tau) \quad (5.2)$$

where the carrier frequency and phase, denoted by  $f_c$  and  $\varphi_c$  respectively, is modulated by data symbols  $c_k$ . The symbols are shaped by a pulse function  $g(t)$ , where  $T$  stands for symbol period and  $\tau$  for symbol timing uncertainty. In [44], MCRB is defined as

$$MCRB(\lambda) = \frac{N_0}{E_u \left\{ \int_0^{T_0} \left| \frac{\partial s(t, \lambda, u)}{\partial \lambda} \right|^2 dt \right\}} \quad (5.3)$$

where  $\lambda$  are estimated parameters and  $u$  is a vector of known or averaged parameters. The expectation  $E_u$  depends on the estimation period  $T_0$  over which the signal properties are averaged.

In our case we assume estimation of the carrier frequency  $f_c$  and averaging over the symbols and timing uncertainty, thus for the modulation signal we can write

$$m(t) = \sum_k c_k g(t - kT - \tau). \quad (5.4)$$

Then the expectation has the form

$$E_u \left\{ \int_{t_0}^{t_0+T_0} \left| \frac{\partial e^{j(2\pi f_c t + \varphi_c)} m(t)}{\partial f_c} \right|^2 dt \right\} = 4\pi^2 \int_{t_0}^{t_0+T_0} t^2 E_u \{|m(t)|^2\} dt. \quad (5.5)$$

The variable  $t_0$  was implemented to allow further inspection of the estimation interval. To average over the symbols, we can apply

$$E\{c_k c_l\} = \begin{cases} C & \text{for } k = l \\ 0 & \text{otherwise} \end{cases} \quad (5.6)$$

and for averaging over the possible timing uncertainty, the Poisson re-summation yields

$$\sum_k g^2(t - kT - \tau) = \frac{1}{T} \sum_n G\left(\frac{n}{T}\right) e^{j2\pi n \frac{t-\tau}{T}} \quad (5.7)$$

and after averaging of  $\tau$ , we can write

$$E\{|m(t)|^2\} = \frac{C}{T} \int_{-\infty}^{\infty} |G(f)|^2 df = \frac{E_S}{T} \quad (5.8)$$

where  $G(f)$  is a Fourier spectrum of the pulse function power, i.e.  $G(f) = \mathcal{F}\{g^2(t)\}$ . The estimation after averaging is proportional to the symbol energy  $E_S$ . The expectation (5.5) then can be expressed as

$$E_u \left\{ \int_{t_0}^{t_0+T_0} \left| \frac{\partial s(t, f, u)}{\partial f} \right|^2 dt \right\} = 4\pi^2 C E_S L_0^3 T^2 \frac{(t_0 + T_0)^3 - t_0^3}{3}. \quad (5.9)$$

It is more convenient to express the estimation period as an integer multiple of  $T$ , i.e.  $T_0 = L_0 T$ , where  $L_0 \in \mathbb{N}$ . It is obvious that the value of (5.9) is highly dependent on the starting time  $t_0$ . From the derivative

$$\frac{1}{3} \frac{\partial (t_0 + T_0)^3 - t_0^3}{\partial t_0} = 2t_0 T_0 + T_0^2 \quad (5.10)$$

it was found that MCRB is minimum for  $t_0 = -\frac{T_0}{2}$  and this assumption together with (5.3) and (5.9) yields the minimum MCRB limit

$$MCRB(f) = \frac{N_0}{4\pi^2 C E_S L_0^3 T^2 \frac{T_0^3}{12}} = \frac{3}{2\pi^2} \frac{1}{L_0^3 T^2} \frac{1}{E_S/N_0}. \quad (5.11)$$

This expression will serve us as the reference performance limit to compare with at the end of this chapter.

## 5.2. Maximum likelihood estimation

The theoretical approach to finding an optimal solution for carrier synchronization comes up with the formulation of the problem. Concerning probability theory, we need to find the frequency and the phase of a harmonic signal which is most probably the closest to the real carrier, more or less lost in the interfering signals, i.e. noise. This can be formulated by the likelihood function and finding its maximum will lead us to the optimal solution from the probability point of view.

Interfering signals are at most, and generally always are, unknown and act as contributions to noise. In a very general case, we assume that noise has a flat power spectral density over the whole valid range, i.e. is white, and has Normal (or Gaussian) distribution. The Normal distribution is defined by

$$f(x|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \quad (5.12)$$

where the probability  $f$  of the receiving signal  $x$  is conditioned by the mean  $\mu$  and the standard deviation  $\sigma$ .

The discrete phase modulated signal has been defined earlier as an equation (5.2) and for our purpose we made slight modifications resulting in

$$\tilde{s}(t) = \sqrt{2P} e^{j(\tilde{\omega}_c t + \tilde{\varphi}_c)} \sum_k \tilde{c}_k g(t - kT - \tau) \quad (5.13)$$

where the symbols which are supposed to be estimated are marked by  $\sim$  symbol, so the symbol  $\tilde{s}(t)$  in (5.13) means the estimation of the PSK signal as a function of time  $t$ . The symbol  $\tau$  stands for the time offset needed to keep symbol synchronization; this is supposed to be known in this case.

Since the representation of the expected signal is known, it can be put into the probability function to formulate the joint likelihood function

$$p(r|\tilde{\varphi}_c, \tilde{\omega}_c, \tilde{c}_k) = \frac{1}{\sqrt{N_0 2\pi}} \exp\left(-\frac{1}{2N_0} \int_{kT}^{(k+1)T} |r(t) - \tilde{s}(t)|^2 dt\right). \quad (5.14)$$

The expected signal  $\tilde{s}(t)$  is the mean in the Normal distribution function and we are interested only in the absolute value of the complex difference between the expectation and the real received signal  $r(t)$ . Additionally, the result should be constant over the symbol period, thus the difference function is integrated over the symbol period to maintain average. The variable  $k$  then indexes the symbol. It is good practice to interpret noise by its power spectral density, and therefore the standard deviation is rewritten according to the relation  $\sigma^2 = N_0$ , where  $N_0$  is a single-sided noise power spectral density.

The noise parameters exactly correspond to the real white noise measurement only in the case where time of acquisition goes to infinity, therefore the probability function calculated over a few samples included in one symbol could not give accurate results, especially for low SNR cases. The acquisition over an infinite number of

symbols is also useless, but the number of symbols used for averaging can be optimized to the useful number  $K$  as

$$\Lambda(r|\tilde{\varphi}_c, \tilde{\omega}_c, \tilde{c}_k) \cong \prod_{k=0}^{K-1} \exp \left( \frac{1}{2N_0} \int_{kT}^{(k+1)T} 2\text{Re}\{r(t)\tilde{s}(t)^*\} - |\tilde{s}(t)|^2 dt \right). \quad (5.15)$$

The integrand in (5.14) has been simplified and the term  $|r(t)|^2$  has been omitted due to its independence of estimated parameters  $\tilde{\varphi}_c$ ,  $\tilde{\omega}_c$  and  $\tilde{c}_k$ . The second term in the right-hand side is rewritten as

$$|\tilde{s}(t)|^2 = 2P \left| \sum_k \tilde{c}_k g(t - kT) \right|^2 \quad (5.16)$$

and it is clear that the absolute value discards the dependency on the frequency and phase estimation. As for general cases, we will suppose that the data is completely unknown and no estimation of received symbols will be provided. Then (5.16) can be omitted yielding

$$\Lambda(r|\tilde{\varphi}_c, \tilde{\omega}_c) \cong C \exp \left( \sum_{k=0}^{K-1} \frac{1}{N_0} \int_{kT}^{(k+1)T} \text{Re}\{r(t)\tilde{s}(t)^*\} dt \right). \quad (5.17)$$

This is a very important relation and will be treated separately for frequency and phase estimation, respectively.

### 5.3. Frequency estimation

The focus is firstly aimed at the parameter that is the first object of the estimation within two stage estimators, i.e. the frequency estimation. In the following, the ML estimator is derived as a benchmark for the novel spectrum averaging estimators proposed in the next sections.

#### 5.3.1. Maximum likelihood frequency estimator

The derivation of the maximum likelihood frequency estimator is based on the recently found likelihood function (5.17). Our goal is to average over the symbol estimate  $\tilde{c}_k$  and the phase estimate  $\tilde{\varphi}_c$

First, substitution of the phase modulated signal (5.13) yields

$$\Lambda(r|\tilde{\varphi}_c, \tilde{\omega}_c) \cong \exp \left( \frac{\sqrt{2P}}{N_0} \sum_{k=0}^{K-1} \text{Re}\{\tilde{c}_k^* x(kT) e^{j\tilde{\varphi}_c}\} \right) \quad (5.18)$$

where  $x(t)$  is a substitution of

$$x(t) = \int_T r(\xi) g(\xi - t) e^{-j\tilde{\omega}_c \xi} d\xi. \quad (5.19)$$

Averaging over  $\tilde{c}_k$  in this form is not a task suitable for implementation, it is better to approximate the exponential function by the Taylor series. Substituting the exponent as

$$X_{rs} = Re \left\{ \sum_k \tilde{c}_k^* x(kT) e^{-j\tilde{\varphi}_c} \right\} = |Y| \cos(\arg(Y) - \tilde{\varphi}_c) \quad (5.20)$$

the Taylor series for three fractions is

$$\Lambda(r|\tilde{\varphi}_c, \tilde{\omega}_c) = 1 + \frac{\sqrt{2P}}{N_0} X_{rs} + \frac{2P}{4N_0^2} X_{rs}^2. \quad (5.21)$$

The real part of the function is expressed in a more convenient form by (5.20) and substituting  $Y$

$$Y = \sum_k \tilde{c}_k^* x(kT). \quad (5.22)$$

The form of (5.22) helps to separate the phase estimation (argument) from the frequency estimation (magnitude) and allows averaging over the phase. Equations

$$E\{X_{rs}\} = E \left\{ \frac{1}{2} (Y e^{-j\tilde{\varphi}_c} + Y^* e^{j\tilde{\varphi}_c}) \right\} = 0 \quad (5.23)$$

and

$$E\{X_{rs}^2\} = E \left\{ \frac{1}{2} Y Y^* + \frac{1}{4} Y^2 e^{-j2\tilde{\varphi}_c} + \frac{1}{4} (Y^*)^2 e^{j2\tilde{\varphi}_c} \right\} = E \left\{ \frac{1}{2} Y Y^* \right\} \quad (5.24)$$

evaluate the average of  $X_{rs}$  and its power in (5.21); the mean of the first one is zero and can be omitted. Averaging over symbols assumes that the symbols are uncorrelated and then applies

$$E\{c_k c_i\} = \begin{cases} C & k = i \\ 0 & otherwise \end{cases}. \quad (5.25)$$

The term  $X_{rs}^2$  is then proportional only to the power of the integral (5.19) magnitude and the constant mean symbol power

$$E \left\{ \frac{1}{2} Y Y^* \right\} = E \left\{ \frac{1}{2} \sum_k \sum_i \tilde{c}_k^* \tilde{c}_i x(kT) x(iT)^* \right\} = \frac{C}{2} \sum_k |x(kT)|^2. \quad (5.26)$$

The probability function for the frequency estimation then simplifies to

$$\Lambda(r|\tilde{\omega}_c) \cong \sum_k |x_k(kT)|^2. \quad (5.27)$$

Finally, maximization of the probability function gives the optimal frequency estimation

$$e(\tilde{\omega}_c) = \frac{d}{d\tilde{\omega}_c} \Lambda(r|\tilde{\omega}_c) \cong \sum_k \frac{d}{d\tilde{\omega}_c} |x(kT)|^2 = 2 \sum_k Re \left\{ x(kT) \frac{dx(kT)^*}{d\tilde{\omega}_c} \right\}. \quad (5.28)$$

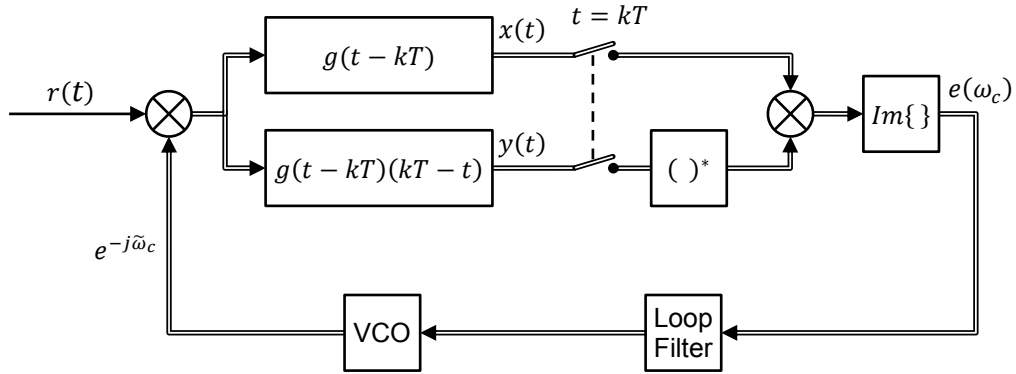
It is rather useful for open loop estimators. For the closed loop estimators, it is more convenient to compute the error signal by finding its derivative (5.27). The derivation after some manipulation yields the equation

$$e(\tilde{\omega}_c) \cong \sum_k \text{Im}\{x(kT)y(kT)^*\} \quad (5.29)$$

with substitution of

$$y(t) = \int_T r(\xi)g(\xi - t)(t - \xi)e^{-j\tilde{\omega}_c\xi}d\xi. \quad (5.30)$$

In Figure 5.1 there is a graphical representation of the maximum likelihood frequency estimator put into the loop. The averaging summation is provided by the Loop Filter, which is usually designed together with the desired locking and tracking characteristics. The complex harmonic signal is generated generally by the voltage control oscillator (VCO) or in the digital form numerically controlled oscillator (NCO) dependent on the filtered error signal.



**Figure 5.1:** Closed loop ML frequency estimator.

### 5.3.2. Spectrum averaging frequency estimator

The specific spectral characteristics of the receiving signals may allow us to find another approach to carrier frequency estimation. The BPSK signal has been defined in equation (2.22) and in a more general form for a specific modulation index (4.109), including the modulation function; the complete BPSK signal with the modulation index  $\beta$  of carrier frequency  $\omega_c$  and phase  $\varphi_c$  is defined by

$$s_{BPSK}(t) = \sqrt{2P} \cos(\beta) \sin(\omega_c t + \varphi_c) + \sqrt{2P} \sin(\beta) \sum_{k=-\infty}^{\infty} c_k p(t - kT_R) \cos(\omega_c t + \varphi_c). \quad (5.31)$$

The  $P$  stands for RMS signal power, which is, by the modulation index, divided into carrier power and data symbol  $c_k$  power. The symbols are modeled by the pulse function  $p(t)$ , where  $T_R$  is the symbol period. The index  $R$  is used to prevent it being mixed up with the sampling period  $T_S$ .

To find the spectrum of the BPSK signal it may be dealt with separately. First, the carrier spectrum is obviously one spectral line at the carrier frequency and its conjugate on the opposite side of the frequency axis. By isolating a single symbol, the average spectral density of the data spectral component or the spectral density of the suppressed carrier BPSK signal was found

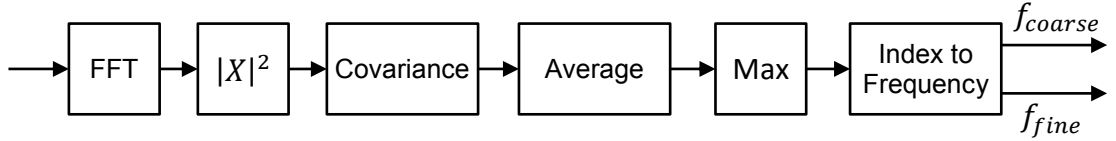
$$\begin{aligned}
 S_{SC}(\omega) &= \frac{1}{T_R} \int_{-\frac{T_R}{2}}^{\frac{T_R}{2}} \cos(\omega_c t + \varphi_c) e^{-j\omega t} d\omega \\
 &= \frac{e^{j\varphi_c}}{2} \frac{\sin\left((\omega - \omega_c) \frac{T_R}{2}\right)}{(\omega - \omega_c) \frac{T_R}{2}} + \frac{e^{-j\varphi_c}}{2} \frac{\sin\left((\omega + \omega_c) \frac{T_R}{2}\right)}{(\omega + \omega_c) \frac{T_R}{2}}.
 \end{aligned} \tag{5.32}$$

The baseband BPSK signal spectrum is then obtained by assumption of the zero carrier frequency and the zero carrier phase

$$S_{SCBB}(\omega) = \frac{\sin\left(\omega \frac{T_R}{2}\right)}{\omega \frac{T_R}{2}}. \tag{5.33}$$

The spectrum of the BPSK signal is symmetric; it is clear as from the function (5.33) as from the fact that the baseband BPSK signal is purely real. It is the key property of the proposed algorithm, firstly because we suppose that the symmetry of the spectrum is unnatural, and secondly it uses the whole signal power for detection, which is important for detection in the noisy environment.

In Figure 5.2, there is shown the block diagram of the proposed algorithm.



**Figure 5.2:** Spectrum averaging frequency estimator.

The spectrum is estimated by the fast Fourier transform (FFT). Since the transform is applied to the finite length generally non-deterministic signal, the estimation more or less differs from reality. It is usual to use windowing technique to improve the estimation parameters; the spectral resolution and the spectral leakage are two key contradictory parameters of the windows. In our case the high dynamic range of the signal is also important, because of the focus on the signals with a high level of additional noise, and every window more or less significantly decreases the dynamic of the analyzed signal. A rectangular window has very high spectral leakage, but excellent resolution characteristic for the signals of comparable strength and signal power is not weighted. That is why the rectangular window is a good choice for the spectral analysis within the frequency estimator, at least in the first step of the development.

To see the exact response of the FFT to the real harmonic signal we applied the discrete Fourier transform to the carrier of the BPSK signal resulting in the spectrum



$$\begin{aligned}
 S_c(k) &= \frac{\sqrt{2P}}{N} \cos(\beta) \sum_{n=0}^{N-1} \sin(\omega_c T_s n + \varphi_c) e^{-j2\pi \frac{k}{N} n} \\
 &= \frac{\sqrt{2P}}{2Nj} \cos(\beta) \left( e^{j\varphi_c} \sum_{n=0}^{N-1} e^{j(\omega_c T_s - 2\pi \frac{k}{N})n} - e^{-j\varphi_c} \sum_{n=0}^{N-1} e^{-j(\omega_c T_s + 2\pi \frac{k}{N})n} \right) \\
 &= \frac{\sqrt{2P}}{2Nj} \cos(\beta) \left( e^{j\varphi_c} \frac{1 - e^{j(\omega_c T_s - 2\pi \frac{k}{N})N}}{1 - e^{j(\omega_c T_s - 2\pi \frac{k}{N})}} - e^{-j\varphi_c} \frac{1 - e^{-j(\omega_c T_s + 2\pi \frac{k}{N})N}}{1 - e^{-j(\omega_c T_s + 2\pi \frac{k}{N})}} \right) \quad (5.34) \\
 &\cong \frac{\sqrt{2P}}{2N} \cos(\beta) \left( e^{j(\varphi_c - \frac{\pi}{2})} \frac{\sin\left(\omega_c \frac{T_s}{2} N - \pi k\right)}{\sin\left(\omega_c \frac{T_s}{2} - \pi k\right)} + e^{-j(\varphi_c - \frac{\pi}{2})} \frac{\sin\left(\omega_c \frac{T_s}{2} N + \pi k\right)}{\sin\left(\omega_c \frac{T_s}{2} + \pi k\right)} \right).
 \end{aligned}$$

In the last expression, the phase of the uncentered rectangular window is omitted. As expected, the result is the sum of two “sinc comb” functions at frequencies  $\pm\omega_c$ .

The result of FFT applied to the real signal has a symmetric magnitude to which the correlation can be applied. The correlation function will result in a high peak if the signals are similar to each other. If the spectrum contains a BPSK signal, i.e. a signal with symmetric spectrum magnitude, the correlation with its image rotated in frequency should result in a high peak at the position of the BPSK signal. The image rotated in frequency is actually contained in the negative frequency side, so auto-correlation may be used. As proposed in [60], the correlation does not converge for signals with a non-zero mean, which the magnitude spectrum generally does not have. A better way is to use the auto-covariance function, where the mean is estimated and subtracted first. The algorithm then has the following steps; the power spectrum estimation

$$P(k) = |S(k)|^2 = \text{Re}\{S(k)\}^2 + \text{Im}\{S(k)\}^2, \quad (5.35)$$

mean estimation

$$\bar{P} = \frac{1}{N} \sum_{n=0}^{N-1} P(n) \quad (5.36)$$

and covariance function estimation

$$C(k) = \sum_{n=-\infty}^{\infty} (P(n) - \bar{P})(P(n+k) - \bar{P}). \quad (5.37)$$

It should be noted that realization is more effective using auto-correlation of the AC component of the power spectrum signal as expressed by

$$P_{AC}(k) = P(k) - \bar{P} \quad (5.38)$$

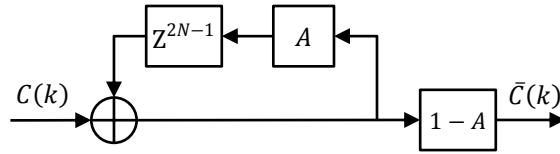
and

$$C(k) = \sum_{n=-\infty}^{\infty} P_{AC}(n)P_{AC}(n+k). \quad (5.39)$$

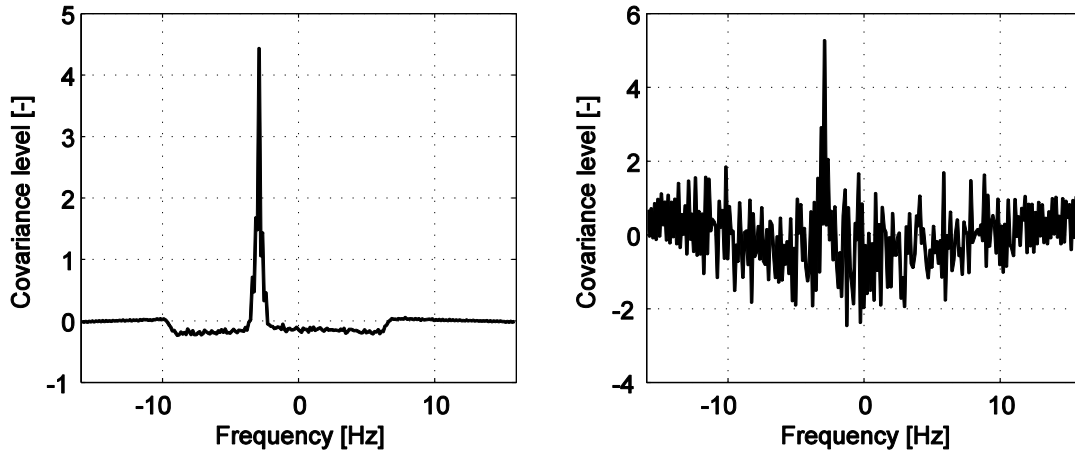
The limits of the general expression for covariance are infinite, but the signal to which it is applied is limited to  $N$  samples and therefore the result of the covariance estimation is non-zero only within a precisely defined range. The limits of the correlation sum then shrink

$$C(k) = \sum_{n=0}^{N-k} P_{AC}(n)P_{AC}(n+k). \quad (5.40)$$

If the input signal contains strong noise, the noise will also be contained in the covariance output, but the signal can be easily averaged to accentuate the correlation peak. The averaging algorithm used is a simple exponential cumulation for the frames of  $2N$  samples as shown in Figure 5.3. The resulting averaged covariance peak is shown in Figure 5.4, where the two examples for different noise levels are depicted.



**Figure 5.3:** Covariance estimation cumulation algorithm.



**Figure 5.4** Averaged covariance function for  $\frac{E_b}{N_0} = 10\text{dB}$  (left) and  $\frac{E_b}{N_0} = 0\text{dB}$  (right).

Finally, the position of the peak defines the carrier frequency of the BPSK signal. It is found by seeking the index of the sample of maximal value. The desired frequency  $f_{cc}$  from the index  $k_c$  is then calculated using the formula

$$f_{cc} = f_s \frac{N - k_c - 1}{2N}. \quad (5.41)$$

$N$  defines the length of FFT, and because spectrum estimation is discretized in frequency, the resulting frequency estimation suffers from quantization error directly, which equals the FFT resolution  $\Delta f = \frac{f_s}{N}$ . The result is then considered as a coarse estimate and is useful, e.g., in connection with the Goertzel pre-downconversion. On the other hand, when the carrier frequency lies in between the FFT bins, it will cause the huge and predictable leakage to neighboring bins. The frequency can then be exactly evaluated from the characteristic (5.34), but the calculation and resulting formula is

expected to be too complex for implementation, and therefore an empirical approximation formula was found for  $\bar{C}(k_l) > \bar{C}(k_r)$

$$f_c = f_{cc} + \frac{\Delta f}{2} \left( 1 - \frac{\bar{C}(k_r)}{\bar{C}(k_l)} \right) \quad (5.42)$$

and for  $\bar{C}(k_l) < \bar{C}(k_r)$

$$f_c = f_{cc} - \frac{\Delta f}{2} \left( 1 - \frac{\bar{C}(k_l)}{\bar{C}(k_r)} \right) \quad (5.43)$$

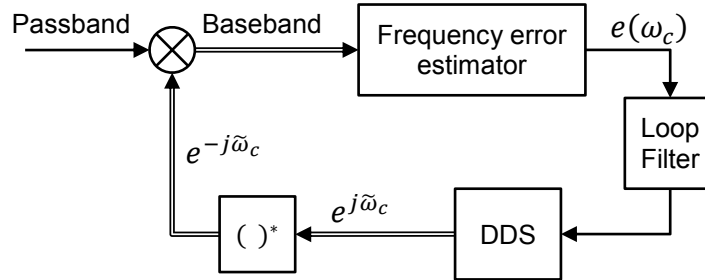
where  $\bar{C}$  is the averaged covariance output and  $l$  and  $r$  indexes the left hand neighbor sample from the coarse estimation and the right hand sample respectively.

### 5.3.3. Closed loop spectrum averaging frequency estimator

This chapter briefly introduces the closed loop system used for frequency estimation; many authors discuss this topic in their books, e.g., phase lock loops are discussed in [53] and purely digital way of constructing carrier synchronizers in [61].

The typical structure of the closed loop frequency estimators is depicted in Figure 5.5. First, the input signal is led to a complex mixer, where the passband signal is downconverted ideally to the baseband. The local oscillator frequency for the mixer is estimated by the loop, so the loop has to be locked first to find the carrier frequency; this phase is usually called locking or acquisition. Since the system is locked, the signal can be used for demodulation and the loop comes to its tracking phase, where it corrects relatively slight variations in frequency.

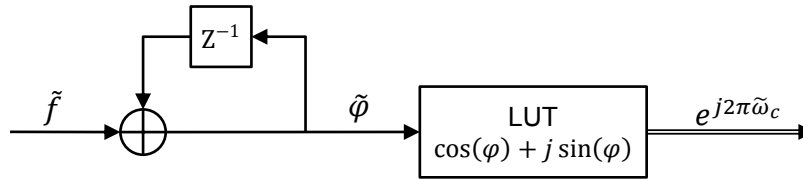
The local oscillator needs an error signal, which defines the deviation of the oscillator frequency from the real carrier frequency. Error signal estimation is usually provided by special filters with additional processing as is used, e.g., in the ML closed loop estimator in Figure 5.1.



**Figure 5.5:** General closed loop frequency estimator.

The key block of the loop, which determines its behavior, is the loop filter. The filter defines the loop bandwidth, i.e. how fast the loop reactions will be for the carrier variations and simultaneously how immune it will be to noise. The transfer function of the filter determines the order and the type of the loop. The order of the loop directly corresponds to the loop transfer function order, whereas the type of loop is defined by the number of included integrators. Since the order of the loop can be higher than the number of integrators contained, the type may be generally lower than the order, but never higher [53]. Figure 5.6 shows an example of usually used discrete oscillators based on direct digital synthesis. The oscillator is designed to keep the output frequency

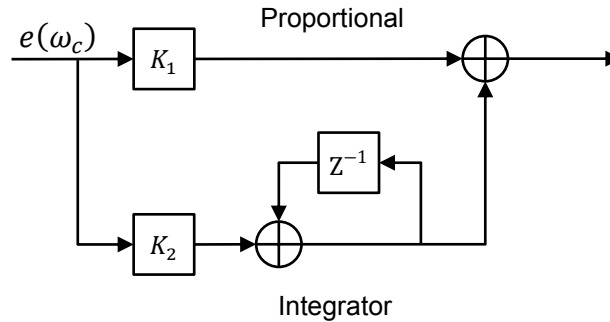
the same if the input signal is constant, but the output signal is not based directly on the input frequency signal, but on the actual phase of the harmonic function. The frequency is the derivation of the phase, thus, the discrete integrator is inserted prior to the look up table (LUT). The integrator included increases the type of loop and if the simplest possible loop filter, i.e. constant, is used, the loop will be of type 1 (and also of first order) and is able to track the first derivative of the input signal phase, i.e. its constant frequency. Do not be misled, a type 1 loop is not able to follow changing frequency, it supposes that the frequency is exactly known and the parameter corrected is only the phase. A signal with different and changing frequency can be tracked by higher type loops, i.e. the filter has to be replaced with one containing one or more integrators. A simple rule says that each derivation of the input signal phase variation must be compensated by an individual integrator within the loop filter including DDS.



**Figure 5.6** Direct digital synthesizer [61].

Type 2 loops are common within terrestrial systems, where the frequency is supposed to be constant or varying very slightly within mobile systems [53]. Satellite systems, e.g. GPS, on the other hand, compete against relatively rapid Doppler shifts, which are dependent on the satellite's actual position. To be able to track the signals exactly, type 3 or higher loops are commonly used [61].

The proportional-plus-integrator loop filter is commonly used in second order type 2 loops and is characterized by the signal diagram in Figure 5.7 [61].



**Figure 5.7** Second order proportional-plus-integrator loop filter [61].

The constants of the filter  $K_1$  and  $K_2$  are, together with the loop constants DDS gain  $K_0$  and the frequency error estimator and the mixer gain  $K_p$ , critical parameters of the loop and are responsible for possible instability of the loop. From the loop analysis in [61], the constants of the loop are defined by

$$K_p K_0 K_1 = \frac{4\zeta\theta_n}{1 + 2\zeta\theta_n + \theta_n^2} \quad (5.44)$$

and

$$K_p K_0 K_2 = \frac{4\theta_n^2}{1 + 2\zeta\theta_n + \theta_n^2} \quad (5.45)$$

where  $\zeta$  is the so-called damping ratio and for a proportional-plus-integrator filter

$$\theta_n = \frac{B_n T_s}{\zeta + \frac{1}{4\zeta}}. \quad (5.46)$$

$T_s$  is, as usual, the sampling period and  $B_n$  is the equivalent loop bandwidth defined the same way as the noise bandwidth as the width of the equivalent rectangular filter with the same power transfer. For more details see [61].

The spectral averaging frequency estimator proposed in the previous chapter gives the absolute frequency of the BPSK signal. The frequency can actually be used as the error signal of the closed loop estimator too. The input signal is slowly moved to the baseband and then the frequency estimator gives zero frequency, which is correct. The only complication is the averaging algorithm within the estimator, since the input signal is moving in the frequency domain, the averaged spectrum will be disrupted. The solutions for such an issue are to make the loop slow enough to allow the estimator to efficiently track the spectral changes or include frequency correction (DDS input) also to the spectral averaging algorithm. The second way seems to be too complex to be solved within this thesis, but definitely deserves future investigation.

#### 5.3.4. Modified closed loop spectrum averaging frequency estimator

The complications with the closed loop estimator based on the spectrum averaging frequency estimator come from the non-linearity of the maximum function. The function selects the peak of the maximal value and it causes long jumps in frequency to which the immunity of the loop filter is naturally low and therefore causes jumps of the DDS frequency which corrupts averaging even more. This is the reason why the fully linear modified version of the closed loop estimator was developed.

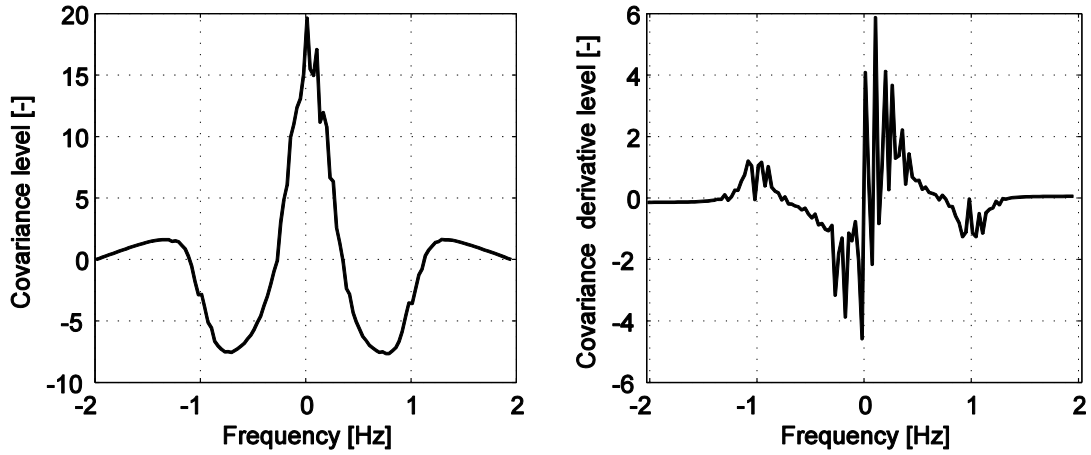
The modified estimator block diagram is shown in Figure 5.8. The basis of the algorithm stays the same, as in the original averaging estimator, but the frequency error is not the absolute frequency estimation, but is provided by additional linear operations. First, the covariance peak is differentiated to linearly transform the maximum to the zero crossing. Naturally, the desired maximum indicates the carrier frequency, thus, the frequency error should be zero.



**Figure 5.8** Modified spectrum averaging frequency estimator.

Figure 5.9 shows a single instance of the covariance peak and its derivative. The noise level is 20 dB lower than the BPSK signal for better clarity of the reached shape. Even though the noise level is low, the derivative is very noisy and the simple zero crossing determination is impossible. On the other hand, the center sample (at the zero frequency) can be taken as the error signal and its variance can be filtered by the narrow loop filter.

A closer look at the derivative curve shows us that the shape is antisymmetric around the zero frequency and thus the integral over the frequency interval is on average zero. To eliminate the parasitic ripple at the curve's edges, which would work against us, the integration range is narrowed to the range approximately from the main negative peak to the main positive peak. Then the locking range of the closed loop estimator will be widened. Every deviation from the nominal carrier frequency will cause integration more over the positive or negative peak of the derivative and it will result in a positive or negative error deviation. The loop will then force to compensate the error.



**Figure 5.9** Covariance peak of the modified spectrum averaging algorithm (left) and its derivative (right).

At the very beginning of the block schematic in Figure 5.8, there is a pulse filter, which is typical also for the ML estimator and plays several roles. First, the filter acts as a matched filter, which is necessary especially for filters like root RC filters. It is a low pass filter, which suppresses noise at higher frequencies and significantly improves the SNR of the signal. On the other hand, the low pass filter narrows the range of loop acquisition and if the signal is not pre-downconverted correctly, the loop will not be able to lock anymore. Finally, the pulse filter is very important for covariance peak shaping. The filter is set to downsample the input signal to fit the FFT resolution to the desired bandwidth and it noticeably widens the covariance pulse. Let us compare the peak in Figure 5.4, where for precise estimation the peak is desired to be as sharp as possible with the wide closed loop case in Figure 5.9. The peak width has an essential impact on the shape of the derivative and subsequently the shape of the tracking characteristic, i.e. S-curve.

Since averaging of signal noise was concentrated only on the loop filter, the design of the loop coefficients according to the desired characteristics is easier.

#### **5.4. Maximum likelihood phase estimation**

The derivation of the phase maximum likelihood estimator starts also from equation (5.17). It is supposed that the received signal is a type of the M-PSK modulated signal, i.e. the symbols in (5.13) are now substituted by a more exact expression as

$$\tilde{s}_{PSK}(t) = \sqrt{2P} e^{j(\tilde{\omega}_c t + \tilde{\varphi}_c)} \frac{1}{M} \sum_{m=0}^{M-1} e^{j2\pi \frac{m}{M}} g(t - kT - \tau). \quad (5.47)$$

The signal can be easily substituted into (5.17) resulting in the equation

$$\begin{aligned} \Lambda(r|\tilde{\varphi}_c, \tilde{\omega}_c) &\cong \frac{1}{M} \prod_{k=0}^{K-1} \sum_{m=0}^{M-1} \exp \left( \frac{\sqrt{2P}}{N_0} \operatorname{Re} \left\{ x(kT) e^{-j\tilde{\varphi}_c} e^{j2\pi \frac{m}{M}} \right\} \right) \\ &= \frac{2}{M} \prod_{k=0}^{K-1} \sum_{m=0}^{\frac{M}{2}-1} \cosh \left( \frac{\sqrt{2P}}{N_0} \operatorname{Re} \left\{ x(kT) e^{-j\tilde{\varphi}_c} e^{j2\pi \frac{m}{M}} \right\} \right) \end{aligned} \quad (5.48)$$

where again for simplicity the substitution of  $x(kT)$  is in accordance with (5.19). The following simplification of the equation is provided by splitting of the sum into two parts and applying the rule of complex exponential symmetry,  $\exp \left( j2\pi \frac{m+M/2}{M} \right) = -\exp \left( j2\pi \frac{m}{M} \right)$ , then the two halves can be expressed as a hyperbolic cosine.

The product is hard to be dealt with, better is to convert it to the sum by the logarithm, as literature also recommends. The function

$$\lambda(r|\tilde{\varphi}_c, \tilde{\omega}_c) \cong \sum_{k=0}^{K-1} \ln \left( \frac{2}{M} \sum_{m=0}^{M-1} \cosh \left( \frac{\sqrt{2P}}{N_0} \operatorname{Re} \left\{ x(kT) e^{-j\tilde{\varphi}_c} e^{j2\pi \frac{m}{M}} \right\} \right) \right), \quad (5.49)$$

usually called a log-likelihood function, is in addition still monotonous and it is easier to provide further calculations.

Finally, let the derivative of (5.49) to be zero to find maximum of the likelihood function. The result can be identified as an error signal within the closed loop phase estimator

$$\begin{aligned} e(\tilde{\varphi}_c) &= \frac{d\lambda(r|\tilde{\varphi}_c, \tilde{\omega}_c)}{d\tilde{\varphi}_c} \\ &= \frac{\sqrt{2P}}{N_0} \sum_{k=0}^{K-1} \frac{\sum_{m=0}^{\frac{M}{2}-1} \operatorname{Im} \left\{ x(kT) e^{-j\tilde{\varphi}_c} e^{j2\pi \frac{m}{M}} \right\} \sinh \left( \frac{\sqrt{2P}}{N_0} \operatorname{Re} \left\{ x(kT) e^{-j\tilde{\varphi}_c} e^{j2\pi \frac{m}{M}} \right\} \right)}{\sum_{m=0}^{\frac{M}{2}-1} \cosh \left( \frac{\sqrt{2P}}{N_0} \operatorname{Re} \left\{ x(kT) e^{-j\tilde{\varphi}_c} e^{j2\pi \frac{m}{M}} \right\} \right)}. \end{aligned} \quad (5.50)$$

The equation (5.50) has been derived for the general M-PSK modulation, but if we suppose a certain case of the BPSK modulation, where  $M = 2$ , the error function simplifies to

$$e_{BPSK}(\tilde{\varphi}_c) = \sum_{k=0}^{K-1} \operatorname{Im} \left\{ \frac{\sqrt{2P}}{N_0} x(kT) e^{-j\tilde{\varphi}_c} \right\} \tanh \left( \operatorname{Re} \left\{ \frac{\sqrt{2P}}{N_0} x(kT) e^{-j\tilde{\varphi}_c} \right\} \right). \quad (5.51)$$

The block diagram of the estimator is shown in Figure 5.10. It is very similar to the well-known Costas loop. The implementation of the hyperbolic tangent is complicated and is usually approximated by the linear amplifier for the low SNR signals or by a linear amplifier with limitation for both high and low SNR signals. The

summation in (5.51) is again provided by the loop filter of which design includes requirements to the loop behavior.

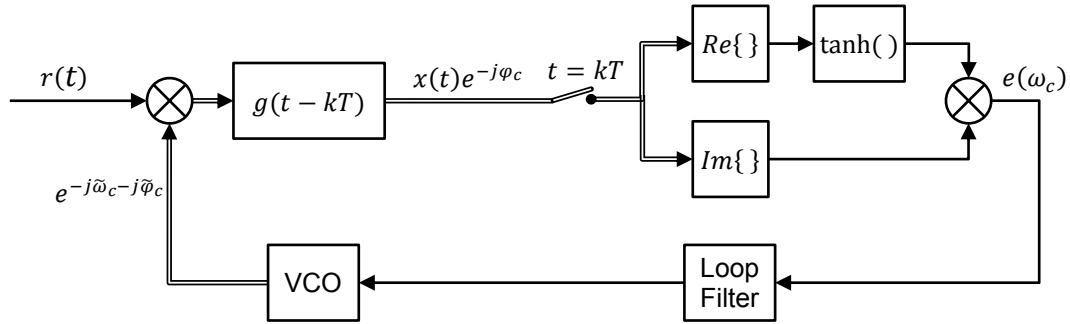


Figure 5.10 Closed loop ML phase estimator.

### 5.5. Simulink tests

The tests of the proposed frequency estimators and comparison with the theoretical limits were achieved by simulations in MATLAB Simulink. The models were composed mainly from available design blocks combined with MATLAB m-file functions. The following text describes each model and this section is concluded with the simulation results and discussion.

Carrier synchronization systems are usually characterized by an error estimator transfer function, the so-called S-curve, and a frequency estimate variance dependent on the signal noise performance. The chain of the general simulation model of the frequency estimators is shown in Figure 5.11. The S-curve characterizes the system from the carrier acquisition and the tracking point of view and the variance characteristic gives the information how the produced carrier estimate may differ from the original one, when the received signal is distorted by the AWGN. To test the real functionality of the proposed synchronizers the bit error ratio is also evaluated.

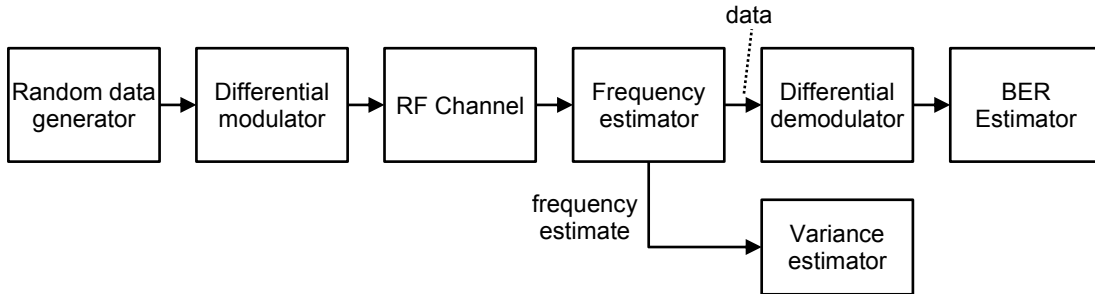


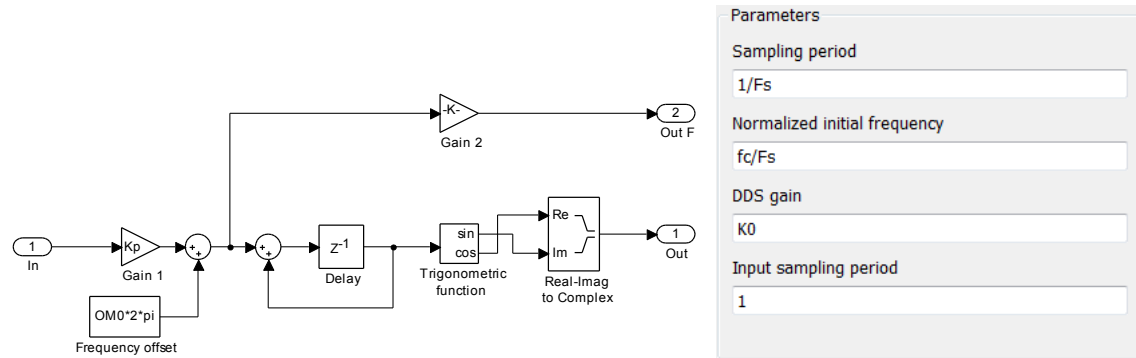
Figure 5.11 General schematic of the frequency estimator simulation model.

#### 5.5.1. DDS model

Typically the discrete oscillator is realized by LUT, CORDIC or the recursive harmonic generator depending on the target device and application. For our purpose for the Simulink tests in the model of DDS in Figure 5.12 the Math function block has been used and to keep the floating point output, no approximation method has been used. The quadrature and the inphase components are linked together as a complex harmonic carrier. As already mentioned, the harmonic signal is controlled by the applied phase, and therefore the integrator is included. The input frequency is scaled by the  $K_p$



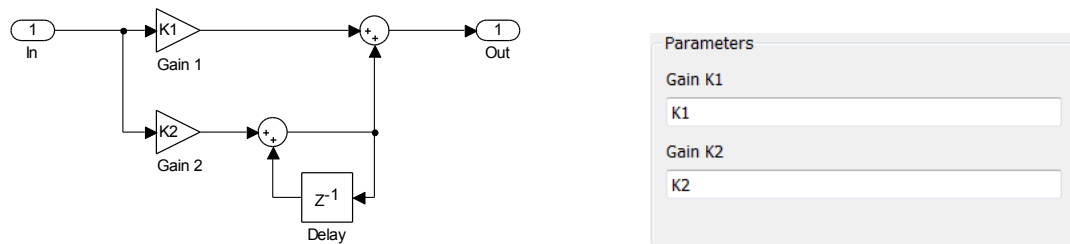
coefficient to provide selectable DDS gain. The constant added to the input scaled frequency serves as an initial frequency or the frequency offset of DDS. The model mask that is shown in Figure 5.12 allows modifying the initial frequency normalized to the sampling period and DDS gain is also adjustable.



**Figure 5.12** DDS Simulink model (left) and its configuration mask (right).

### 5.5.2. Loop filter model

The model of the proportional-plus-integrator filter is simple according to its signal diagram in Figure 5.7 and is shown in Figure 5.13. The filter is adjustable using the coefficients  $K_1$  and  $K_2$ , and when the coefficient  $K_2$  is set to zero, the integrator is disconnected and the loop becomes type 1 from type 2.



**Figure 5.13** Loop filter model (left) and its configuration mask (right).

### 5.5.3. DBPSK modulator and demodulator

Since the frequency estimators are not generally able to synchronize to the carrier phase, with limited knowledge demodulation to the carrier frequency is non-coherent. To be able to correctly demodulate the signal, an easy way is to use differential modulation; in our case differential BPSK (DBPSK).

The DBPSK modulator in Figure 5.14 is a simplified version of the BPSK model in Figure 4.32. The model uses a DBPSK modulator model instead of the BPSK model to perform differential coding. The modulation index setting has been removed for simplicity and additionally the modulator produces a complex single side band signal. It is the signal which is supposed to be given by the pre-downconverter in a complete system, but it is not necessary to implement the pre-downconverter in the frequency estimation simulation.

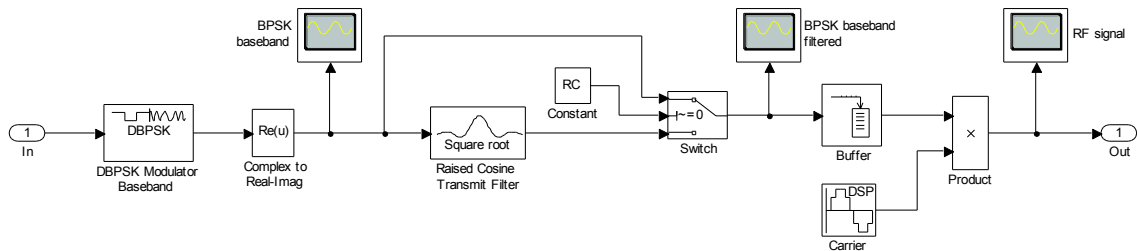


Figure 5.14 DBPSK modulator model.

The DBPSK demodulator is shown in Figure 5.15. It does not include the DBPSK demodulator model to perform differential decoding, but the signal is decoded first in the delay-conjugate circuit right after decimation performed by the Root Raised Cosine pulse filter.

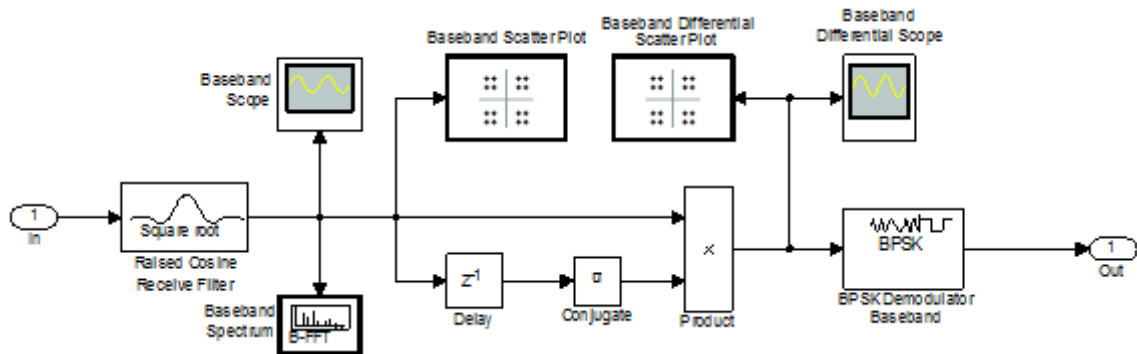


Figure 5.15 DBPSK demodulator model.

The settings of the models are shown in Figure 5.16. It is supposed that the input signal of the demodulator is already downconverted to the baseband by the frequency estimator block, therefore the setting is reduced to the only pulse filter roll-off factor.

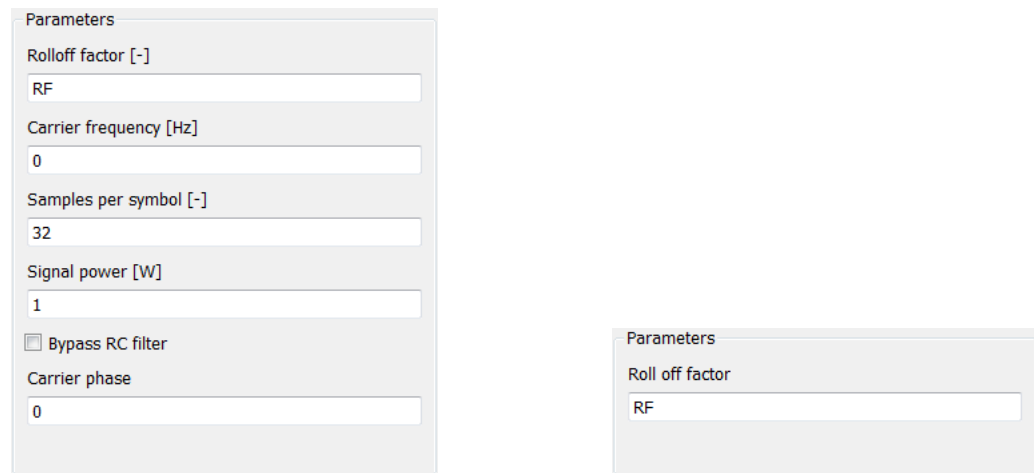


Figure 5.16 DBPSK modulator settings (left) and demodulator settings (right).

#### 5.5.4. Maximum likelihood frequency estimator model

The model of the closed loop ML frequency estimator is shown in Figure 5.17 and is composed according to the block diagram in Figure 5.1. Downconversion is provided by the mixer realized with the simple complex multiplication with a conjugated version of the reconstructed carrier from the DDS block. Buffers make from the sample flow from the input and the DDS frames of the same length to be multipliable and the size of

the frames is as long as is the decimation factor of the forthcoming blocks to produce a single sample per frame at the frequency error level.

The parallel filters are FIR type defined by the impulse response calculated within the initialization code in Listing 5.1 dependent on the parameter settings, Figure 5.18. The filters themselves do not include downsampling as was common in the RC pulse filters, but it is implemented as a separate block after integration.

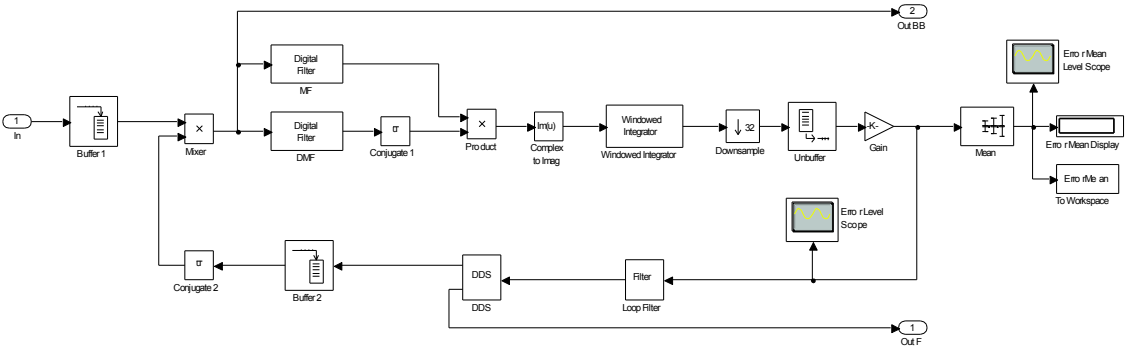


Figure 5.17 ML closed loop frequency estimator.

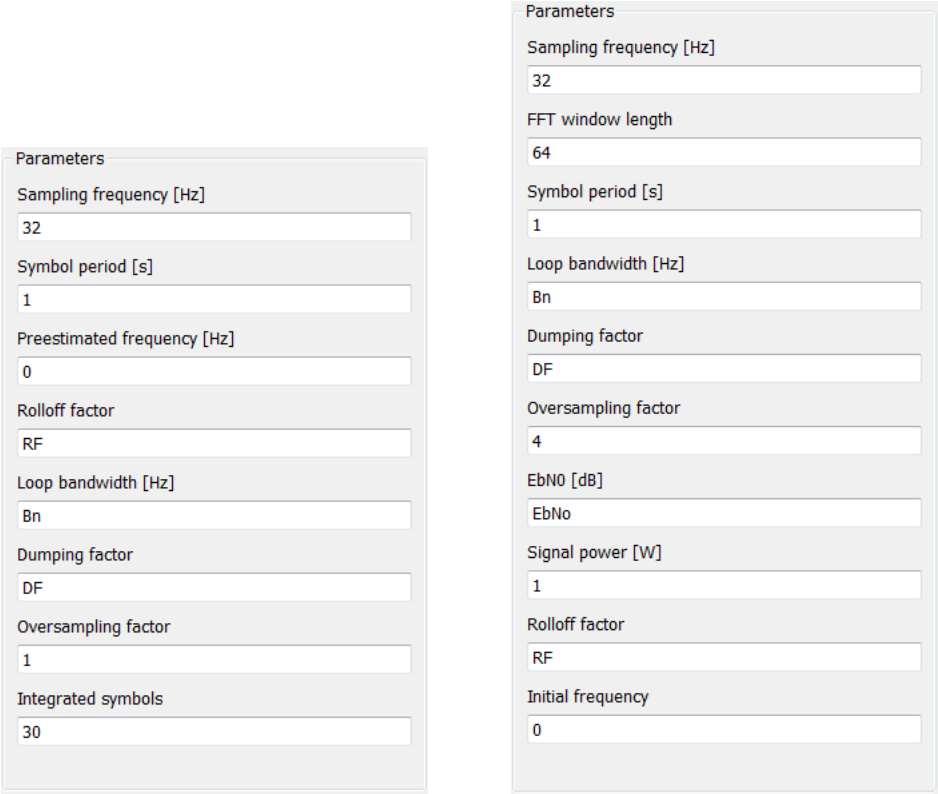


Figure 5.18 ML estimator settings (left) and spectrum averaging frequency estimator settings (right).

```

Gp = 12;

Nsym = 6;

MFSpec = fdesign.pulseshaping(Ts*Fs, 'Square Root Raised Cosine',
    'Nsym,beta', Nsym, RF);
MFFlt = design(MFSpec);

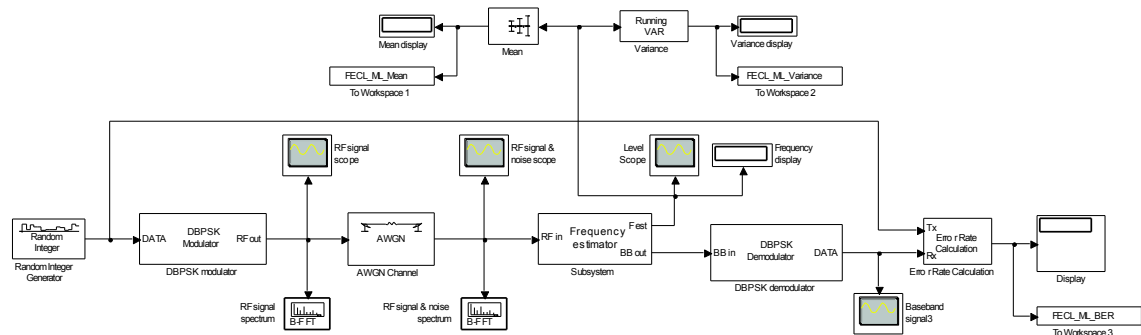
DMFFlt = design(MFSpec);
t = -Ts*Fs*Nsym/2:1:Ts*Fs*Nsym/2;
DMFFlt.Numerator = DMFFlt.Numerator.*t;

Kp = 1;
K0 = 1;
nf = Bn/Fs/(df + 1/(4*df));
K1 = 4*df*nf/((1+2*df*nf+nf^2)*Kp*K0);
K2 = 4*(nf^2)/((1+2*df*nf+nf^2)*Kp*K0);
    
```

**Listing 5.1** ML frequency estimator initialization code.

The integrator is realized as a moving average filter with the length the same as the number of samples per symbol. This is straightforward implementation of the estimator with known symbol timing. In such a case, the frame composed within the model has to contain the samples of a single symbol, and the integrated symbol parameter in the setting window has to be set to one. If we suppose the symbol timing is not known, the integrator may be replaced by an unsynchronized lowpass filter as is discussed in [1]. Thus, we can set the moving average length to a higher number of symbols and provide the symbol averaging, which makes the algorithm independent of the symbol synchronization. The last part of the error estimator, the gain block, sets the output level to the normalized values and includes the  $K_p$  gain. The error signal is then led to the proportional-plus-integrator loop filter and DDS to produce the carrier replica.

Listing 5.1 shows the code that is run once within the model initialization sequence and serves for the impulse response of the filters calculation and calculation of the loop gain coefficients. The setting of the filters is dependent on the sampling period, the roll-off factor and the symbol period. These parameters are adjustable by the user, and therefore the impulse responses are not fixed. Additionally, the derivative of the impulse response of the DMF filter cannot be calculated by any Simulink model and a special MATLAB code had to be written. The loop gain coefficients are estimated by the code providing the calculations according to equations (5.44) through (5.46) and allow an easy way of setting the loop.

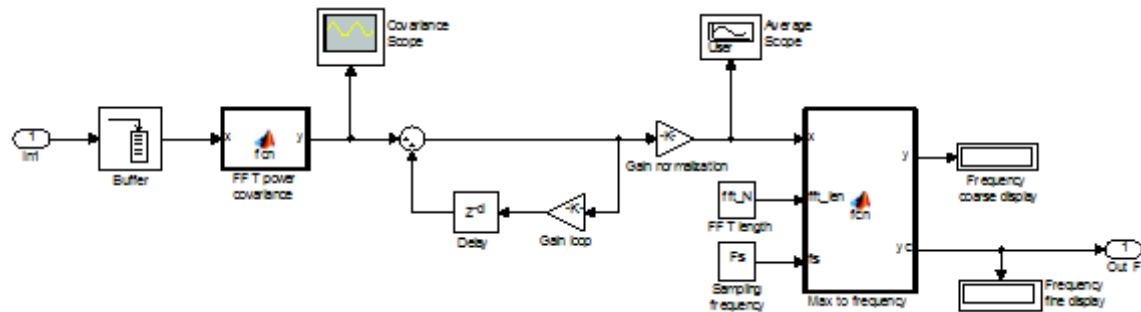


**Figure 5.19** ML closed loop estimator model.

The overall model for the frequency estimation tests is very similar to the downconverter models in the previous chapter, but the signal is downconverted by the frequency estimator. Then the downconverted and demodulated signal is processed to estimate BER. Additionally, there are typical measurements provided for the carrier synchronizers as is the frequency estimation variance and the mean for the locking check. The model of the estimator in Figure 5.17 contains the measurement of the frequency error mean which serves for the S-curve simulation with a disconnected loop. A loop disconnection may be easily made by changing the DDS gain  $K_0$  in the initialization code or in the DDS setting window to zero.

### 5.5.5. Spectrum averaging frequency estimator model

The model of the open loop spectrum averaging frequency estimator is shown in Figure 5.20. It starts with the buffer followed by the m-file function of which its code is depicted in Listing 5.2. The size of the buffer indirectly sets the size of the FFT window, which can be seen in the code, where the size is extracted.



**Figure 5.20** Open loop spectrum averaging frequency estimator.

```
function y = fcn(x)

len = length(x);
y = zeros(2*len - 1, 1);

x = (abs(fftshift(fft(x)))/(len/2)).^2;

x = x - sum(x)/length(x);
xA = x;
xB = [zeros(len - 1, 1); flipud(x); zeros(len - 1, 1)];

for i = 1:(2*len - 1)
    y(i) = 0;
    for j = 1:len
        y(i) = y(i) + xA(j)*xB(j + i - 1);
    end
end
```

**Listing 5.2** FFT power covariance code.

As is expected from the function block name, the code estimates the covariance function of the signal spectrum. First, the normalized power magnitude spectrum is estimated by the FFT. The covariance is calculated according to equations (5.38) and (5.40), so the mean is calculated first and it is followed by the correlation estimation. Since the model is able to process a complex input signal of a single sided spectrum, the input signal is correlated with its spectrum-flipped version. The function outputs the full version of the covariance function, i.e. all nonzero components are included, and it

causes the output data rate to be  $N:2N - 1$  as is also readable from the code of  $y$  initialization.

The covariance function bearing the carrier frequency in the form of a conspicuous peak is averaged by exponential cumulation with a period of  $2N - 1$  samples. The degree of averaging is adjustable through the Gain loop block by the coefficient in the range  $\langle 0; 1 \rangle$ ; the coefficient is once more used for the amplitude normalization.

The last m-file function block contains the code for locating the peak position and recalculating the position into the frequency estimate. Locating the peak position is based on searching the maximum position index and the discrete frequency calculated from (5.41) is then refined with the inclusion of the maximum value and values of the nearest neighbor samples according to equations (5.42) and (5.43). The code is shown in Listing 5.3.

```
function [y, yc] = fcn(x, fft_len, fs)

x_len = length(x);
[~, idx] = max(x);

y = fs/(2*fft_len)*(fft_len - idx - 1);
df = fs/fft_len;

idx_left = idx - 1;
idx_right = idx + 1;

if idx_left == 0
    idx_left = x_len;
end
if idx_right > x_len
    idx_right = 0;
end

l = x(idx_left);
r = x(idx_right);
yc = y;

if l > r
    yc = y + df*(1 - r/l)/2;
else
    if r > l
        yc = y - df*(1 - l/r)/2;
    end
end
```

**Listing 5.3** The code of the maximum position to frequency conversion function.

The setting window of the open loop spectrum averaging frequency estimator is shown in Figure 5.21. The window length sets the size of the input buffer and thus the FFT size and the averaging coefficient is the exponential cumulation coefficient.

**Parameters**

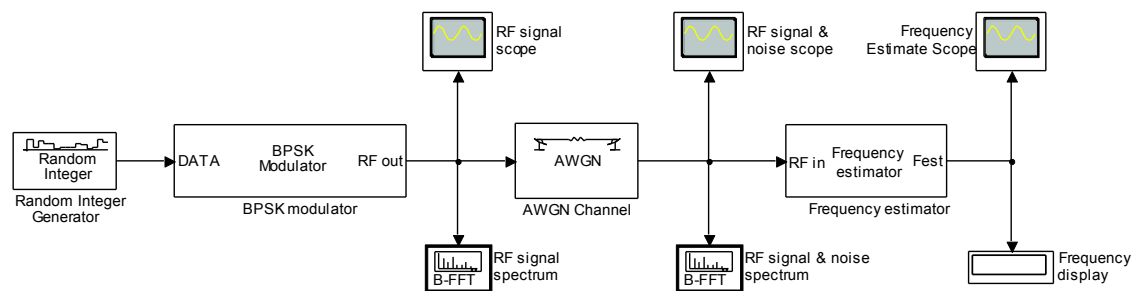
Window length  
128

Average coefficient  
0.995

Sampling frequency  
32

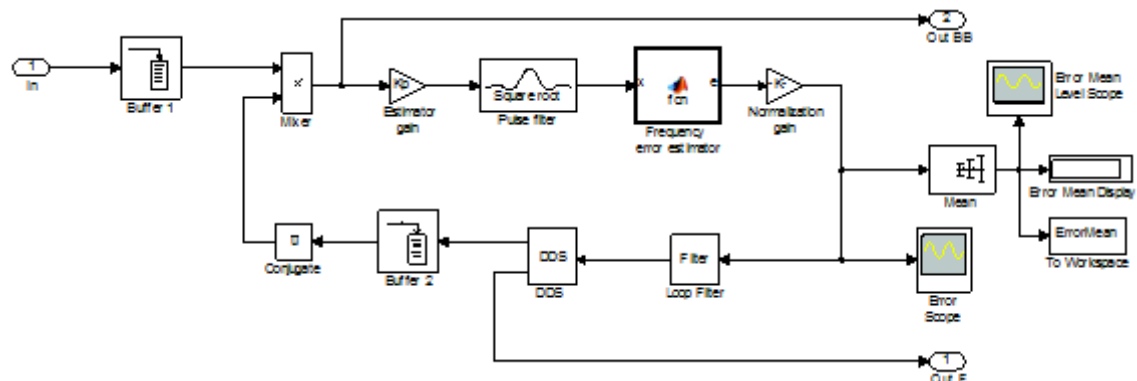
**Figure 5.21** Open loop spectrum averaging frequency estimator settings.

The Simulink model of the frequency estimator using the open loop spectrum averaging frequency estimator is shown in Figure 5.22. The model does not include BER and variance estimation blocks because of its insufficient accuracy in very noisy environments. It is rather supposed to use the algorithm in connection with a narrowband carrier frequency/phase estimator.



**Figure 5.22** Open loop spectrum averaging frequency estimator model.

The closed loop spectrum averaging frequency estimator model is based only on its modified version according to Figure 5.8. The Frequency error estimator is prepended by the estimator gain block and the pulse filter, and appended by the normalization gain. The pulse filter is a general low-pass filter minimizing intersymbol interferences (ISI), in our case the square root filter. The decimation factor of the pulse filter is adjustable in this model dependent on the desired error estimator bandwidth and noise immunity.



**Figure 5.23** Closed loop spectrum averaging frequency estimator.

The code of the error estimator is listed in Listing 5.4 and is based on the code of the open loop estimator in Listing 5.3, but with several modifications. Since the decimation factor of the pulse filter is adjustable, due to covariance peak shaping, the spectrum of the oversampled signal will contain spectral components at irrelevant frequencies. These components are removed from the covariance calculation. The

covariance estimate is then differentiated to convert the value to the error function. Finally, the function is filtered by integration, realized by the sum over a specified interval centered on the zero frequency.

```
function e = fcn(x)

len = length(x);
f_c = 1/2;
LP = floor(len*f_c);

x = abs(fftshift(fft(x)));
x = x(len/2 - LP + 1:len/2 + LP);

len = length(x);
C = zeros(2*len - 1, 1);

x = x - sum(x)/length(x);
xA = x;
xB = [zeros(len - 1, 1); flipud(x); zeros(len - 1, 1)];

for i = 1:(2*len - 1)
    C(i) = 0;
    for j = 1:len
        C(i) = C(i) + xA(j)*xB(j + i - 1);
    end
end
C = C/len;
y = [C; 0] - [0; C];

e = sum(y(len-14:len+13));
```

**Listing 5.4** Frequency error estimator.

The model has to be initialized by the code in Listing 5.5 similarly as the ML closed loop frequency estimator model. The initialization contains the calculation of the loop coefficients dependent on the loop bandwidth and the dumping ratio setting. The frequency error estimator gain is dependent on the input power and especially on the noise power when used in a very noisy environment, thus, the gain is corrected by the introduced dependence of  $K_p$  on the noise power to give comparable results with the ML estimator.

```
NP = 2^10;
C_avg = 0.9;

Kp = 1/(Ps*(1 + 10^(-ebno/10)))^0.5;
K0 = 1;
nf = Bn/Fs/(df + 1/(4*df));
K1 = 4*df*nf/((1+2*df*nf+nf^2)*Kp*K0);
K2 = 4*(nf^2)/((1+2*df*nf+nf^2)*Kp*K0);
```

**Listing 5.5** Closed loop spectrum averaging frequency estimator initialization code.

The setting window of the closed loop spectrum averaging frequency estimator is shown in Figure 5.18. The loop characteristic is adjustable by the bandwidth and the dumping factor, the gain calculations also needs the noise power estimate. The square root raised cosine filter has an independently adjustable roll off factor. Covariance shaping is provided by the Oversampling factor and the FFT window length parameters. The oversampling factor defines the number of samples per symbol period, which



indirectly sets the decimation factor of the pulse filter. The Fourier window size affects the spectrum estimation accuracy and the spectral leakage; it is usually set to cover several multiples of the symbols. The FFT size, together with other time parameters, defines the overall frequency error estimator decimation factor.

The testing model for the closed loop spectrum averaging frequency estimator in Figure 5.24 is very similar to the one for the ML estimator in Figure 5.19.

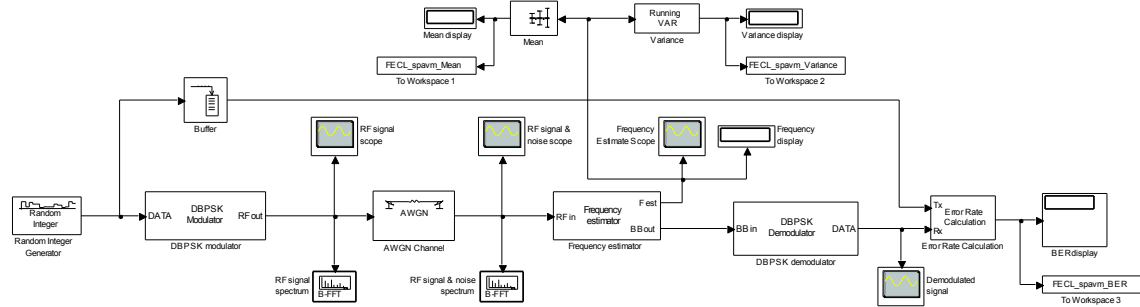


Figure 5.24 Closed loop spectrum averaging frequency estimator model.

### 5.5.6. Maximum likelihood phase estimator model

The maximum likelihood phase estimator – the Costas loop – is shown in Figure 5.25. The model is constructed according its theoretical base in Figure 5.10. The pulse filter is replaced by the synchronized windowed integrator, thus, the ISI minimization filtration is supposed to be placed before the Costas loop on the receiver side or as a single pulse shaping filter on the transmitter side. The hyperbolic tangent function is implemented as a Trigonometric function block and no implicit approximation is used.

The model is supposed to serve as support for system testing and as a carrier synchronization reference.

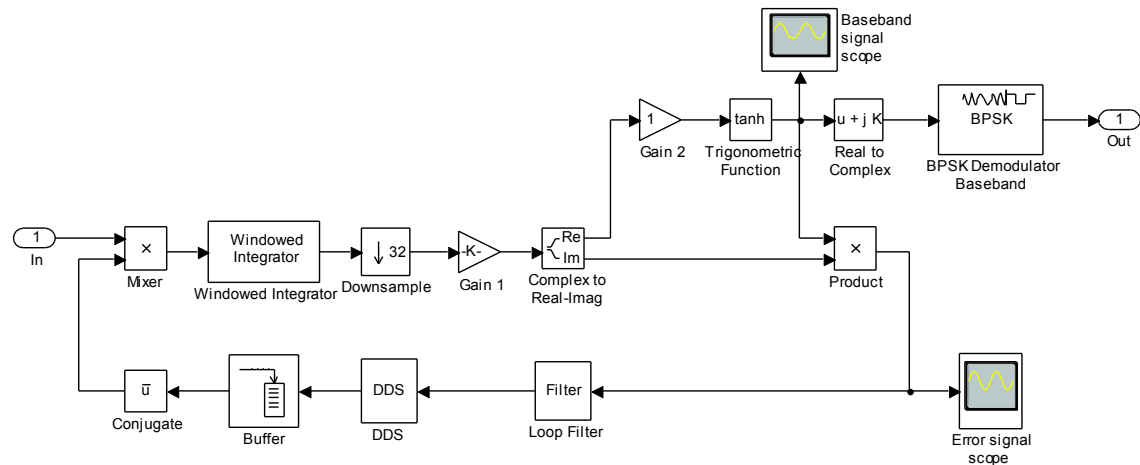


Figure 5.25 Closed loop ML phase estimator - Costas loop.

### 5.5.7. Estimation simulation results

In Figure 5.26 there are shown the simulated S-curves of the ML closed loop and spectrum averaging closed loop frequency estimators. The shape of the characteristics is typical and that is the origin of its name by its similarity to the letter “S”. The characteristics are measured at the output of the frequency error estimator in the system with a disconnected loop while the signal carrier frequency is swept. The function is the

immediate reaction of the error estimator to the carrier frequency variations. It is possible to read out the range of acquisition if the carrier frequency is out of the S-curve; the error estimator provides no reaction to such an offset and thus the system is not able to synchronize. The slope of the linear region crossing the zero level in the middle of the S-curve defines the tracking performance of the system.

Comparing the ML and spectrum averaging estimator S-curves shows that the systems have nearly identical acquisition and tracking characteristics with the difference that the spectrum averaging system has a slightly narrower acquisition region.

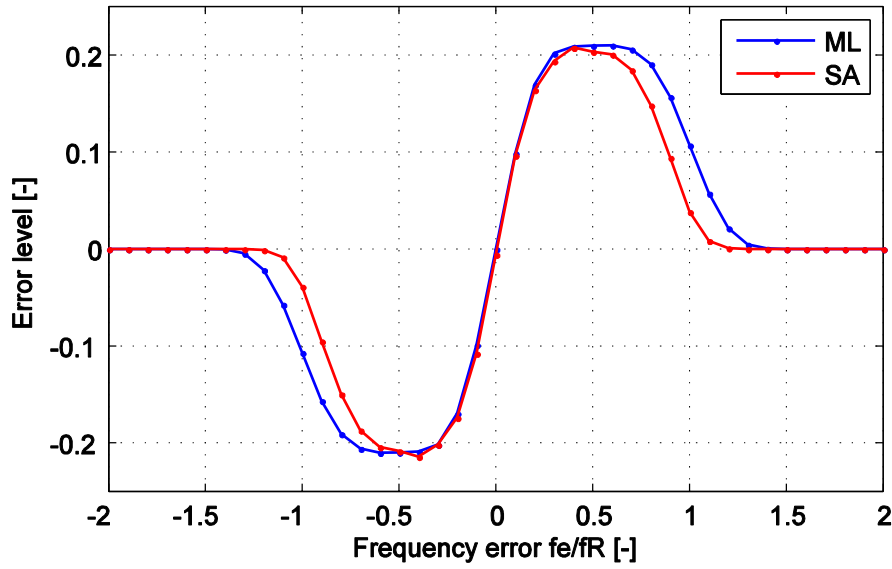


Figure 5.26 Frequency estimators S-curves.

In the previous chapter, we simulated BER within the system of various downconverters and coherent BPSK demodulation was performed. The systems for testing frequency estimators have to use non-coherent demodulation of the DBPSK signal firstly because the estimation gives no information about the carrier phase, and secondly because the frequency estimation variance is high in the interested noisy environment.

Unlike the BPSK signal, the theoretical BER of DBPSK modulation is slightly different. The difference lies in the fact that the single detection error usually leads to two bit errors. For BPSK, the error at the single bit position may cause detection error, which affects only the corresponding bit, but when the data is coded differentially, such an error causes miss-detection of the bit change, which must follow incorrectly at the position of one of the next data bits, and for which the decision will also be wrong. Thus, the single detection error for differentially coded modulation results in two, not necessarily consecutive, data bits. The bit error probability for the differentially coherent DBPSK is [62]

$$BEP = \frac{1}{2} \exp\left(-\frac{E_b}{N_0}\right). \quad (5.52)$$

The error probability functions with its description and derivation are presented in [63] and with the analysis over fading channels in [62].

The bit error rates measured by the simulation of the frequency estimator models are plotted in Figure 5.27 (left). It is nicely seen that the simulated BER of both models

are in good conformity with the theoretical error probability BEP and it indicates that both estimators work fine and do not have a big impact on detection.

The right plot of Figure 5.27 shows the variance performance of the models compared to the modified Cramer-Rao bound defined in section 5.1. Both estimators are quite far from ideal performance. There are many averaged parameters, and therefore our expectation could not be more optimistic. To improve performance, more information is needed for the estimator; in [44] there are the derived data aided estimators with at least partially known data sequence or symbol synchronized estimators with minimal timing uncertainty. Such estimators have rightfully better performance, but the price paid is obviously more complex system and possibly lower useful channel capacity, when using training sequences.

The spectrum averaging frequency estimator was designed for carrier synchronization of signals in very noisy environments within non-aided systems (without data aid or symbol timing). Comparing to the maximum likelihood estimator, the evolution is different, for higher  $E_b/N_0$  ratios the ML estimator has undoubtedly better performance. For higher noise power, however, the performance of the introduced novel estimator comes closer and even crosses the ML line, where it shows better performance. Of course, we have to be critical and state that the region of better performance is very close to the useless signal to noise ratios in usual systems. On the other hand, the algorithm seemingly has potential for further interests.

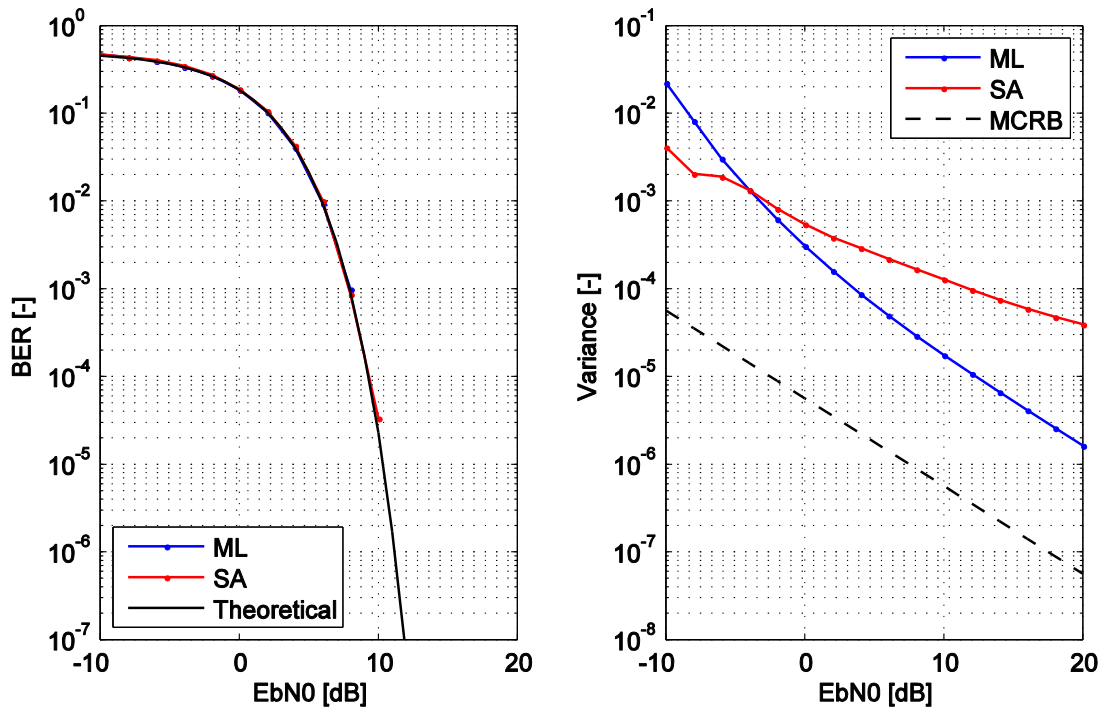


Figure 5.27 Frequency estimators BER results (left) and carrier variance (right).

## 5.6. Summary

- MCRB was introduced at the beginning of the chapter and we stated this limit as a theoretical benchmark.

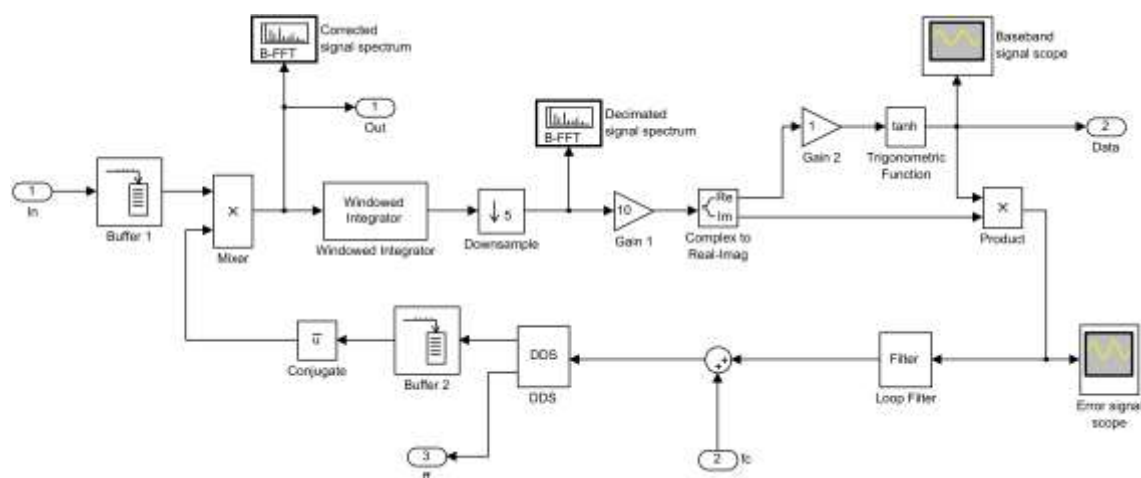
- The typical algorithm for frequency synchronization based on the maximum likelihood function was introduced for clarity also with its derivation from the likelihood function. The algorithm later serves as a practical benchmark.
- The open loop spectrum averaging frequency estimator is a novel algorithm based on seeking spectrum symmetry in the received signal. The algorithm has a big advantage of having a very wide acquisition band.
- The closed loop frequency estimator based on the frequency estimator from the open loop version was proved to be inefficient. The modified closed loop spectrum averaging algorithm inherited the core from the open loop version, but was extended by a differentiator and an integrator to form a frequency error estimator with better performance.
- All the presented algorithms were simulated in MATLAB Simulink and the performance testing models were prepared. The results of the simulation finally show that the spectrum averaging algorithms could not beat the ML based estimator for moderate noise powers, but proved to be even better in very noisy environments, which it was designed for.

## 6. Experimental tests

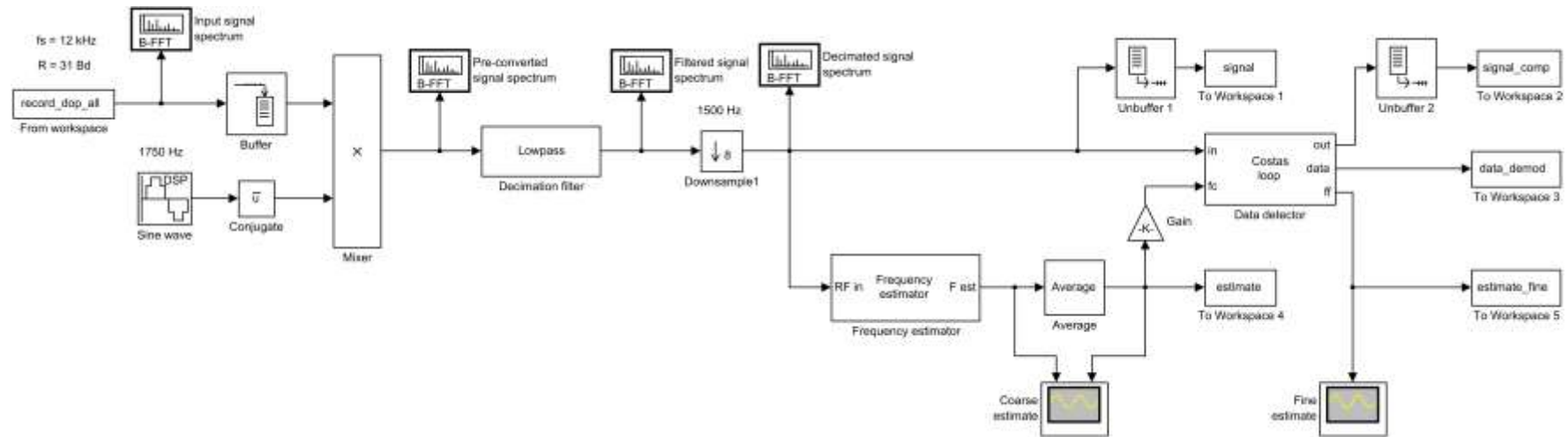
The methods discussed previously are intended for signal processing within the reception of the narrowband satellite signals. We obtained a record of a wideband signal from a fixed frequency SDR system which includes the desired narrowband BPSK modulated signal from a PSAT satellite distorted by the Doppler shift during flyover. The PSK31 transponder carried by the PSAT satellite was designed at the department of Radioelectronics in collaboration with United States Naval Academy and the satellite was launched on May 20, 2015 [65]. In the following, the signal record is subjected to algorithms in order to obtain a transmitted message.

### 6.1. Frequency synchronization

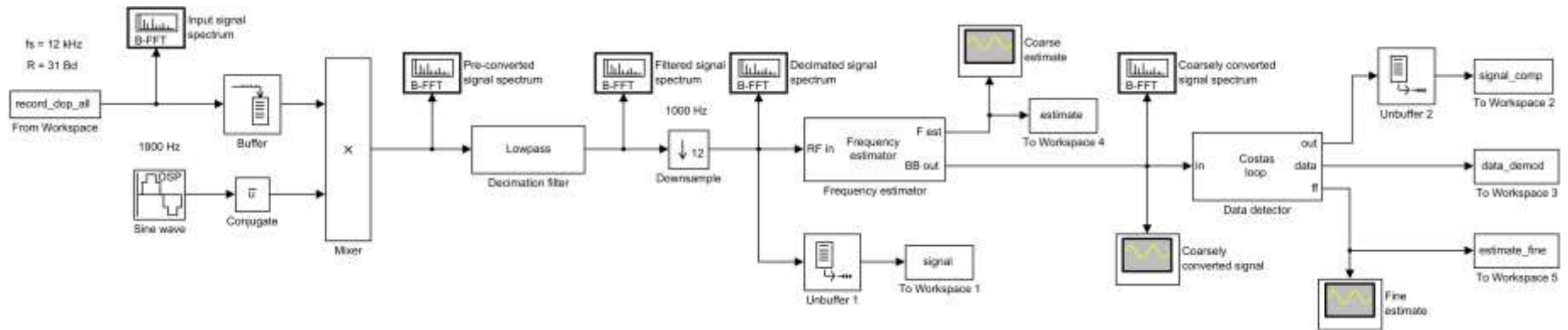
The recorded signal was loaded from a \*.wav file to the MATLAB workspace as a timeseries variable “record\_dop\_all”. Simulink models are able to process the timeseries and also store the results to another timeseries in its workspace. The model for the open loop spectrum averaging frequency estimator is shown in Figure 6.2 and the model for both closed loop spectrum averaging and maximum likelihood frequency estimators is shown in Figure 6.3. Firstly, the signal is pre-processed to limit its bandwidth to the range of the possible Doppler shift with its center frequency near the satellite transmission frequency. All used frequency estimators are not able to synchronize the phase that is needed for coherent demodulation, but are used as a necessary frequency aid since the Costas loop, acting as the phase synchronizer, has a very limited acquisition range. The frequency aid from the open loop estimator is led to the Costas loop directly by addition to the DDS tuning signal. The model of the Costas loop is therefore slightly modified as shown in Figure 6.1. This is an efficient solution because the frequency conversion is performed by a single mixer within the Costas loop. The situation is different for the closed loop frequency estimators where the loop requires its own mixer, and therefore the frequency correction is prepended to the Costas loop and the system requires two independent mixers in each loop.



**Figure 6.1:** Simulink model of the Costas loop with frequency aid.

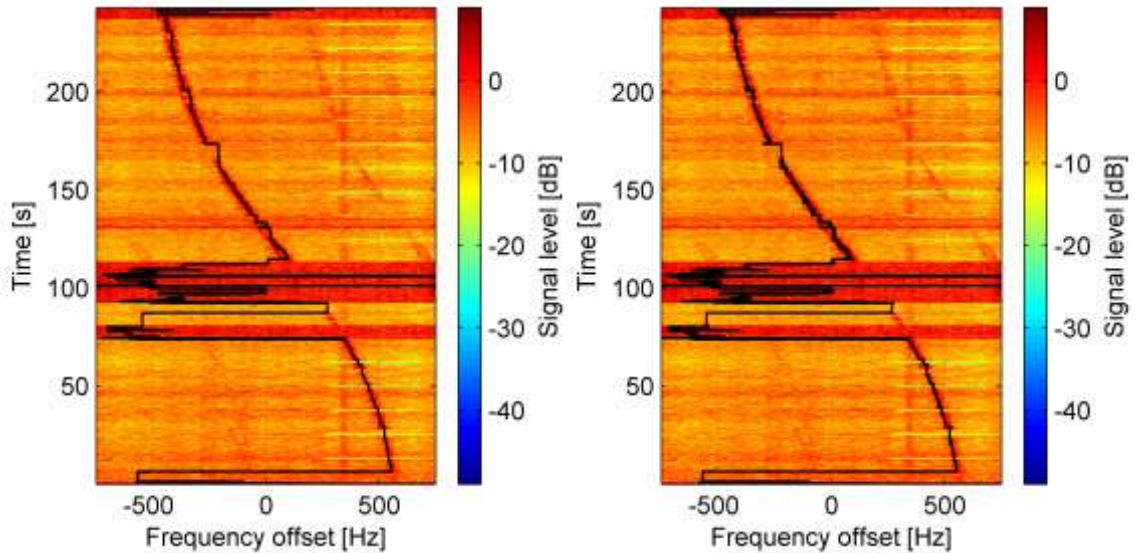


**Figure 6.2:** Simulink model of the open loop frequency and Costas loop phase synchronizers.

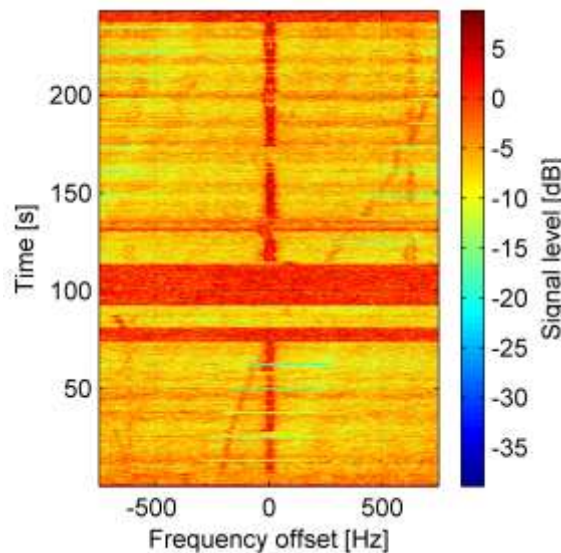


**Figure 6.3:** Simulink model of the closed loop frequency and Costas loop phase synchronizers.

The transmitted signal is the PSK31, which is used for very slow communication of keyboard typing speed, from this radio-teletype (RTTY) [64]. The carrier signal is modulated by BPSK modulation by differentially coded data of speed 31.25 Bd. The useful bandwidth of the PSK31 signal is therefore about 60 Hz. In contrast, the Doppler shift at carrier frequency 435 MHz from the PSAT satellite at low Earth orbit (LEO) reaches  $\pm 600$  Hz. The huge ratio between the signal bandwidth and the Doppler shift is a challenging task for frequency synchronizers.



**Figure 6.4:** Waterfall spectrogram with the frequency estimate from the open loop spectrum averaging estimator – coarse estimate (left) and from the Costas loop – fine estimate (right).

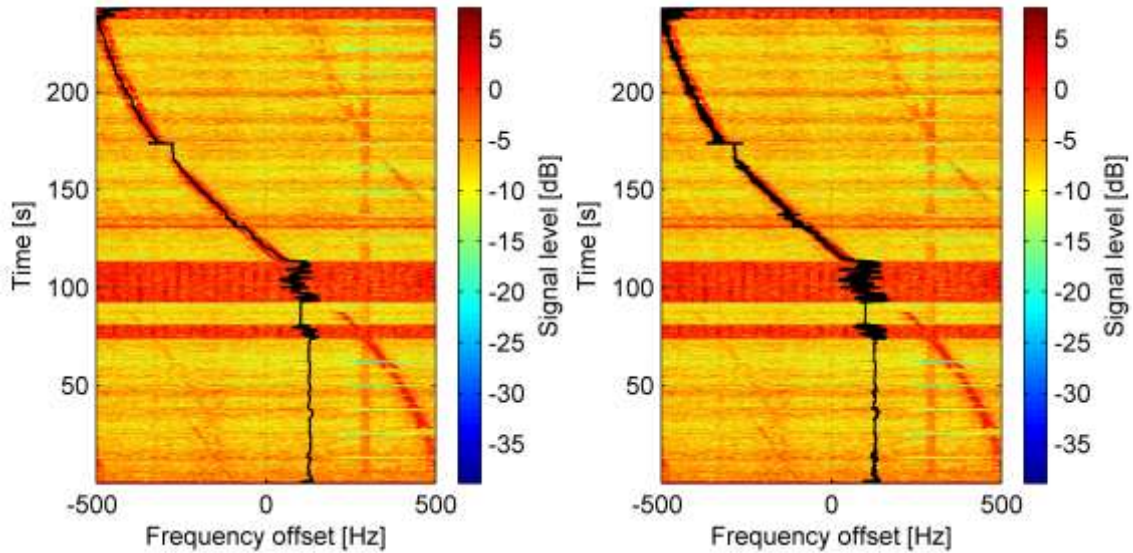


**Figure 6.5:** Waterfall spectrogram frequency compensated by the coarse estimate of the open loop spectrum averaging frequency estimator.

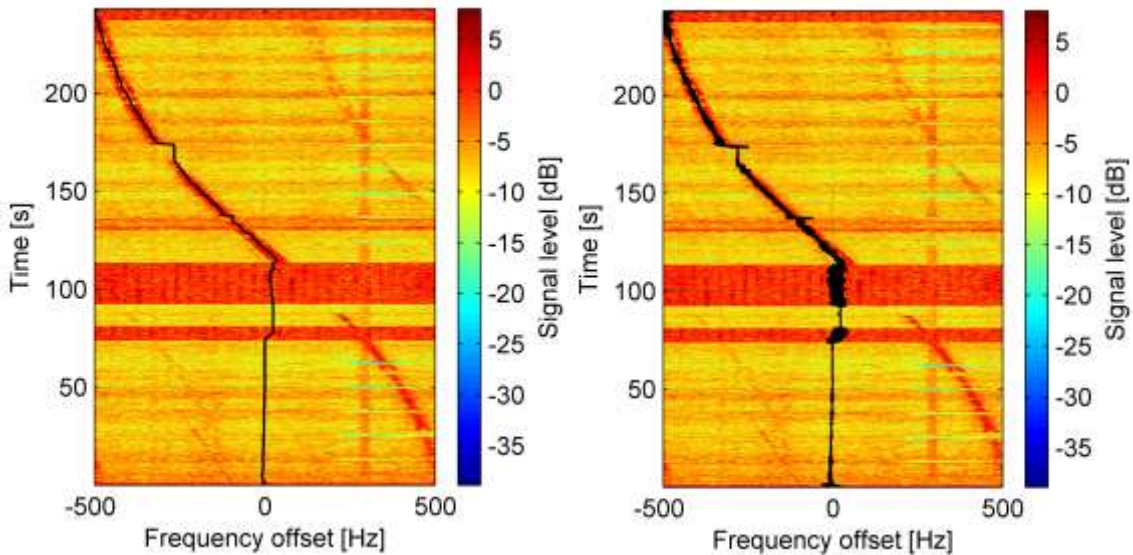
In Figure 6.4, there are shown the waterfall spectrograms of the recorded sample signal of the sampled bandwidth 1500 Hz. This signal was led unfiltered to the open loop spectrum averaging frequency estimator to obtain the frequency estimate. The left-hand side figure includes the line showing the actual frequency estimate from the frequency estimator – coarse estimate. The fine frequency estimate given by the Costas loop is shown in the right hand side figure. The waterfall spectrogram of a carrier



synchronized signal is shown in Figure 6.5. The estimator does not need any setting of initial frequency, it uses all the input bandwidth. The results show that the coarse estimation is very successful. In the signal, there are noisy parts with a lack of the PSK31 signal; this is caused by receiver disconnection from the antenna during the control transmission of the ground station. This causes estimation errors, but recovery after receiver reconnection is fairly fast.



**Figure 6.6:** Waterfall spectrogram with the frequency estimate from the closed loop spectrum averaging estimator – coarse estimate (left) and from the Costas loop – fine estimate (right) with 100 Hz initial frequency.

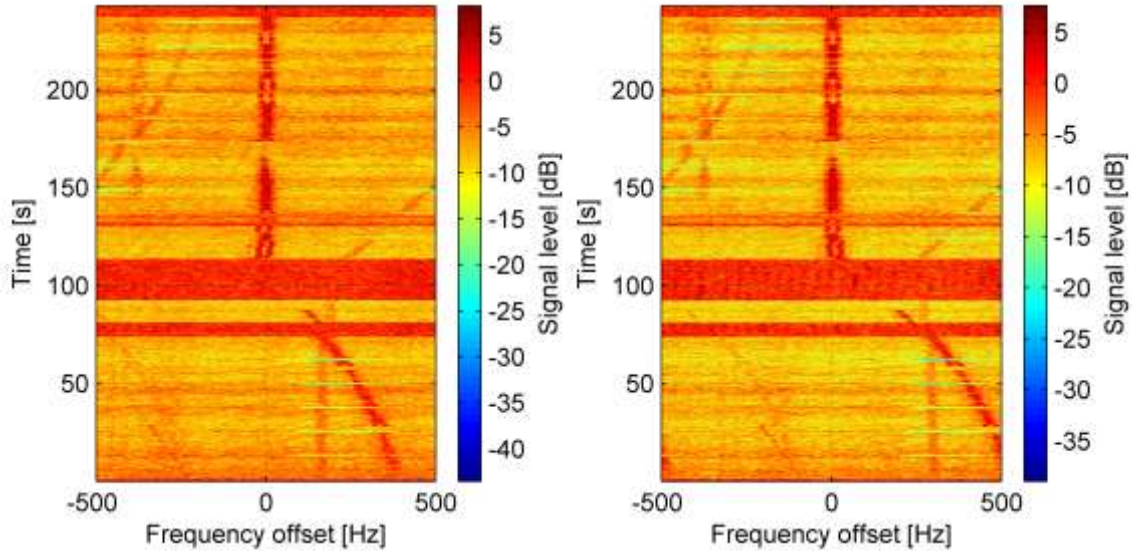


**Figure 6.7:** Waterfall spectrogram with the frequency estimate from the closed loop maximum likelihood estimator – coarse estimate (left) and from the Costas loop – fine estimate (right) with 0 Hz initial frequency.

On the contrary, the closed loop frequency estimators employ low-pass filtration at its input and this significantly narrows the acquisition range. Initial frequency setting is therefore important for correct acquisition. The simulation results for the closed loop spectrum averaging and closed loop maximum likelihood frequency estimators are shown in Figure 6.6 and Figure 6.7 respectively. Both estimators provide successful acquisition after more than 100 seconds, while the signal approaches the acquisition

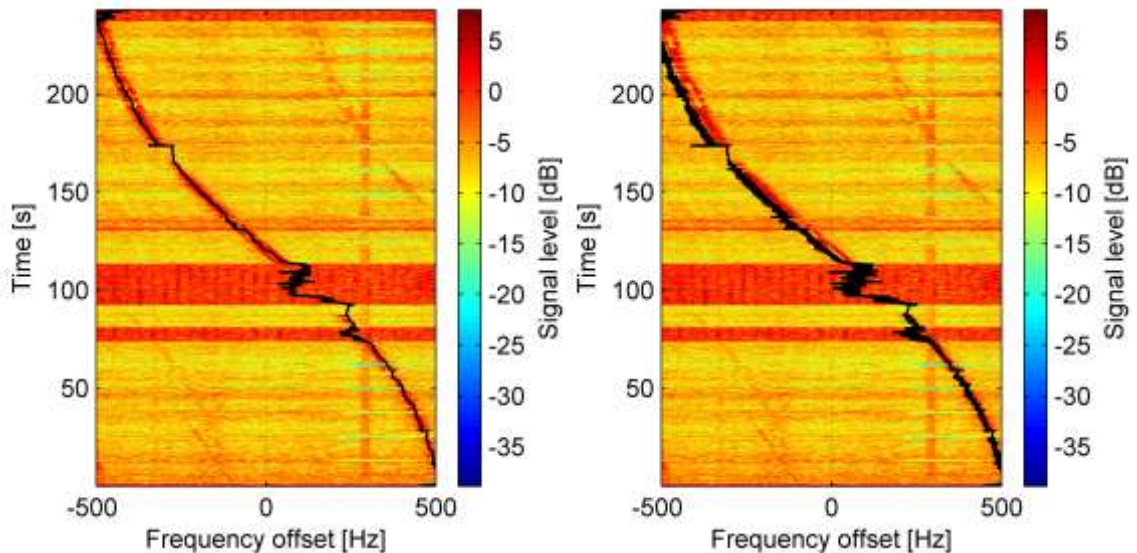


range. The matched filter bandwidth of the CL SA estimator is set to be slightly wider, and therefore is able to lock with 100 Hz initial frequency; this is not the case of the CL ML estimator, where the highly widened matched filter and frequency matched filter cause strong degradation to the estimate. On the other hand, the CL ML estimate is noticeably smoother and obviously has less noise compared to the OL SA and CL SA estimators. The resulting corrected signal spectrograms are shown in Figure 6.8.

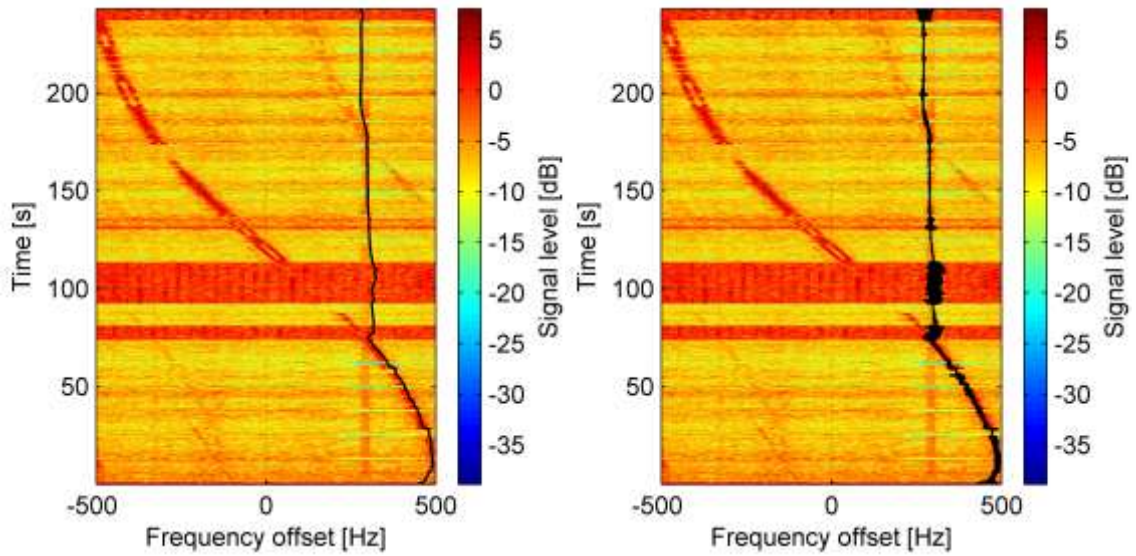


**Figure 6.8:** Waterfall spectrograms frequency compensated by the coarse estimate of the closed loop frequency estimators – spectrum averaging with 100 Hz initial frequency (left) and maximum likelihood with 0 Hz initial frequency (right).

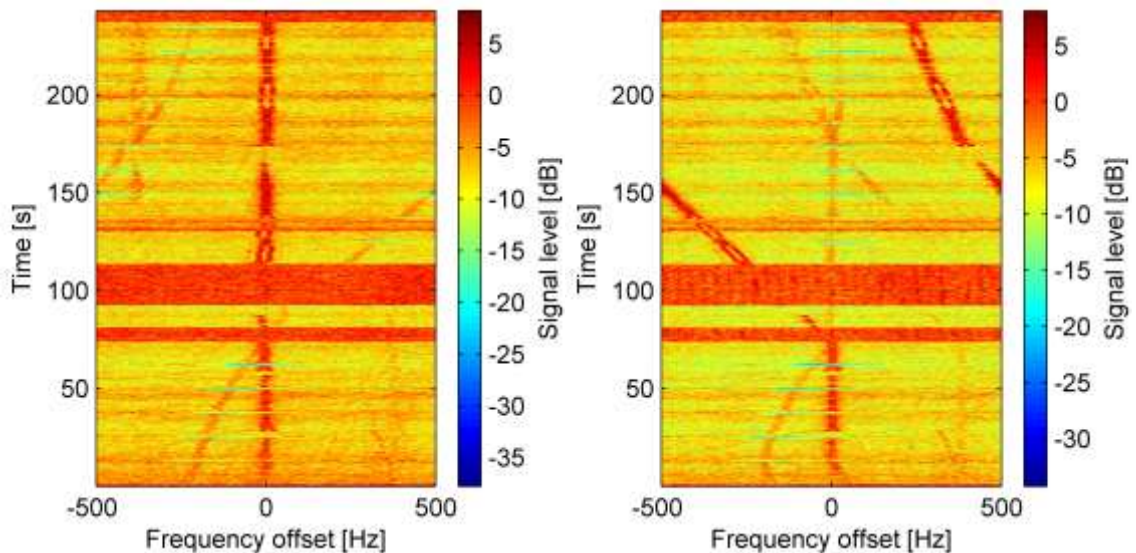
Figure 6.9 and Figure 6.10 show the results for 450 Hz initial frequency setting, which can be supposed as near the expected frequency offset at the beginning of the satellite flyover. The corrected signal spectrograms are shown in Figure 6.11.



**Figure 6.9:** Waterfall spectrogram with the frequency estimate from the closed loop spectrum averaging estimator – coarse estimate (left) and from the Costas loop – fine estimate (right) with 450 Hz initial frequency.



**Figure 6.10:** Waterfall spectrogram with the frequency estimate from the closed loop maximum likelihood estimator – coarse estimate (left) and from the Costas loop – fine estimate (right) with 450 Hz initial frequency.



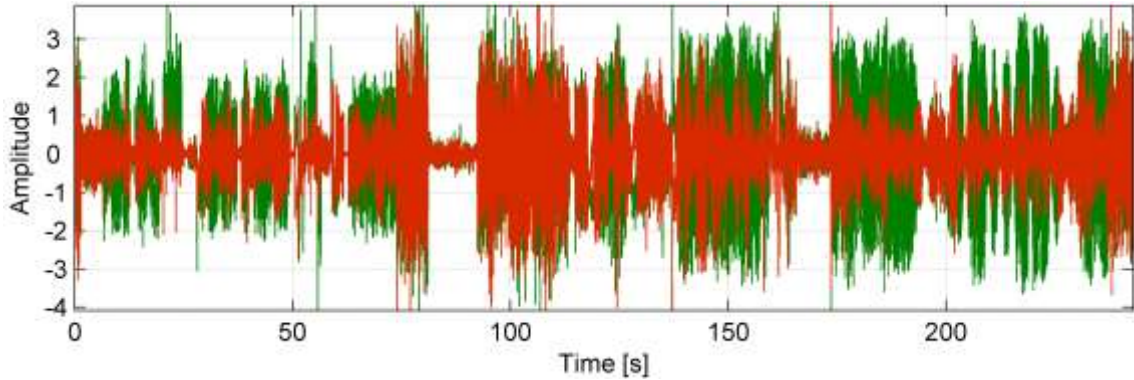
**Figure 6.11:** Waterfall spectrograms frequency compensated by the coarse estimate of the closed loop frequency estimators with 450 Hz initial frequency – spectrum averaging (left) and maximum likelihood (right).

It is obvious that unlike the CL SA estimator, the CL ML estimator has lost lock even after a short period of signal loss and therefore using CL ML is disqualified for frequency synchronization of such signals in this straightforward version. However, the results show that with a properly set initial frequency, the CL SA estimator is applicable for signals with strong Doppler shifts and limited drop-out times. One can notice the loss of synchronization of the fine estimate in Figure 6.9. This is caused by the Costas loop lock loss during signal drop-out. The intelligence of the control system should be able to overcome this issue by switching the Costas loop to the acquisition mode.

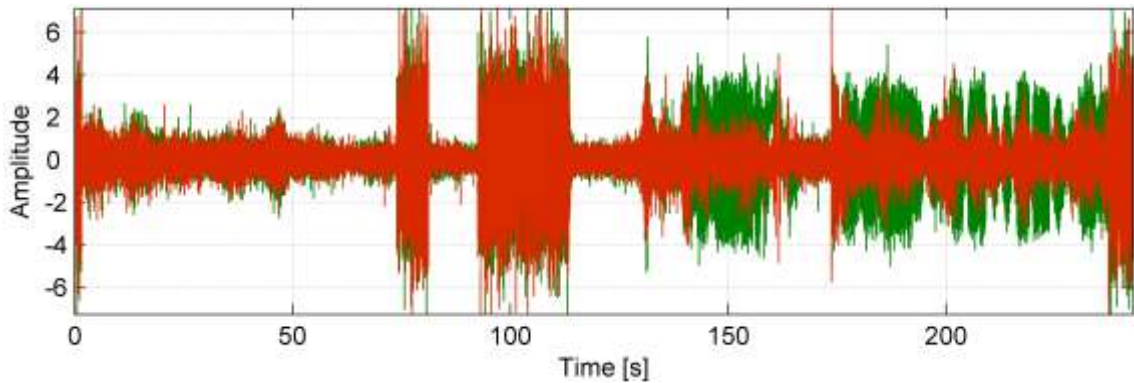
The benefits of the OL SA frequency estimator now seem to be undisputable, but there is an additional point. The curves of fine estimations of each method show different residual noise and this naturally affects the detection error rate. In Figure 6.12, Figure 6.13 and Figure 6.14 the baseband complex signal after carrier synchronization



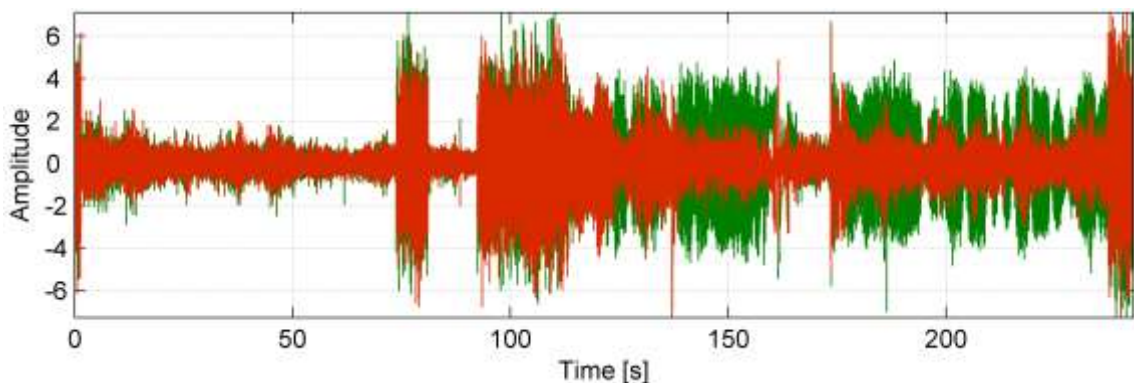
is shown. The real and imaginary parts are colored in such a way that the more green color in the plot means less detection errors. Strictly speaking, it is not true since the noise in the real part (green) may also unpredictably degrade detection, but for this record sample the analogy corresponds to the error rate in the decoded message. The signals show the main drawback of the spectrum averaging methods – high estimation noise. Comparing the closed loop estimators, the Costas loop locking times are significantly faster in the case of the ML estimator. The noise influence is even more visible in the case of the open loop estimator, especially by a time of 150 s.



**Figure 6.12:** Carrier synchronized complex (real part – green, imaginary part – red) signal of the open loop spectrum averaging estimator and Costas loop.



**Figure 6.13:** Carrier synchronized complex (real part – green, imaginary part – red) signal of the closed loop spectrum averaging estimator and Costas loop.

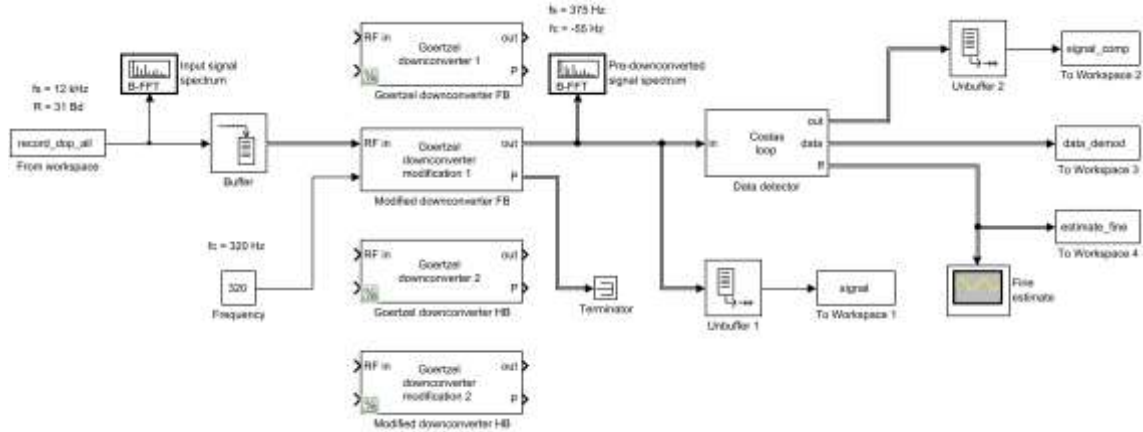


**Figure 6.14:** Carrier synchronized complex (real part – green, imaginary part – red) signal of the closed loop maximum likelihood estimator and Costas loop.

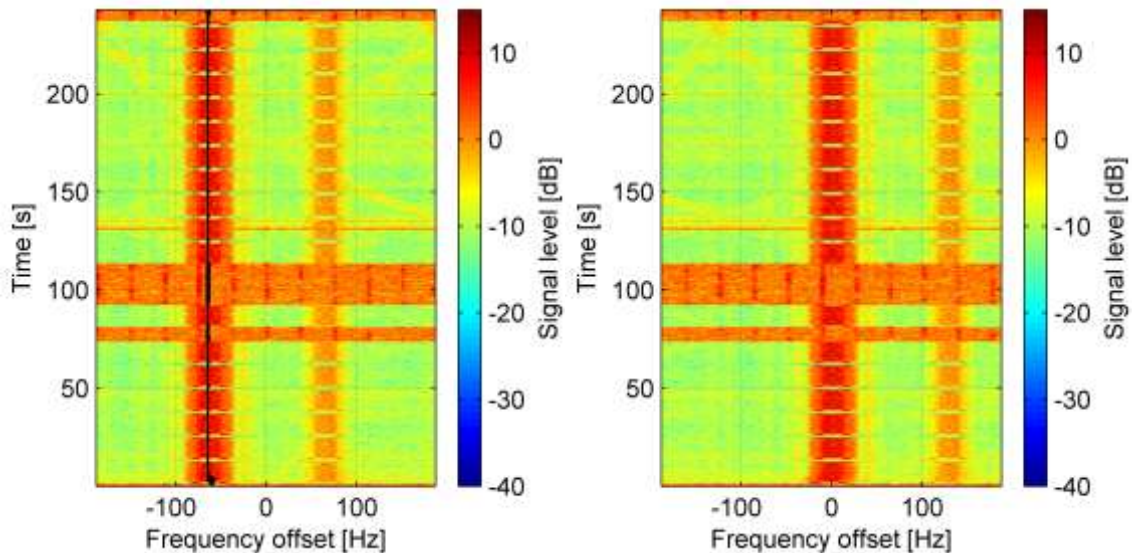
## 6.2. Digital downconversion

Signal recording of the PSAT satellite signal analyzed in the previous section also includes a signal from the ground radio-beacon. The signal is also PSK31 modulated and has a bandwidth of about 60 Hz, but with no Doppler frequency shift. The signal bandwidth to the sampled bandwidth ratio is around 100 and it is an ideal opportunity for applying the Goertzel downconverter.

In Figure 6.15, a Simulink model of the system for Goertzel downconverter experiments is shown. The record is processed from the MATLAB workspace the same way as in the previous section. The beacon signal center frequency is about 320 Hz in the 12 kHz sampled record. All the developed downconverters are included in the model (unemployed are commented out) and are set to decimation factor 32, thus, the resulting signal has a sampling frequency of 375 Hz and the signal of interest is aliased to the center frequency of -55 Hz. The signal is demodulated by the Costas loop with predefined carrier frequency and the results are stored to the workspace.



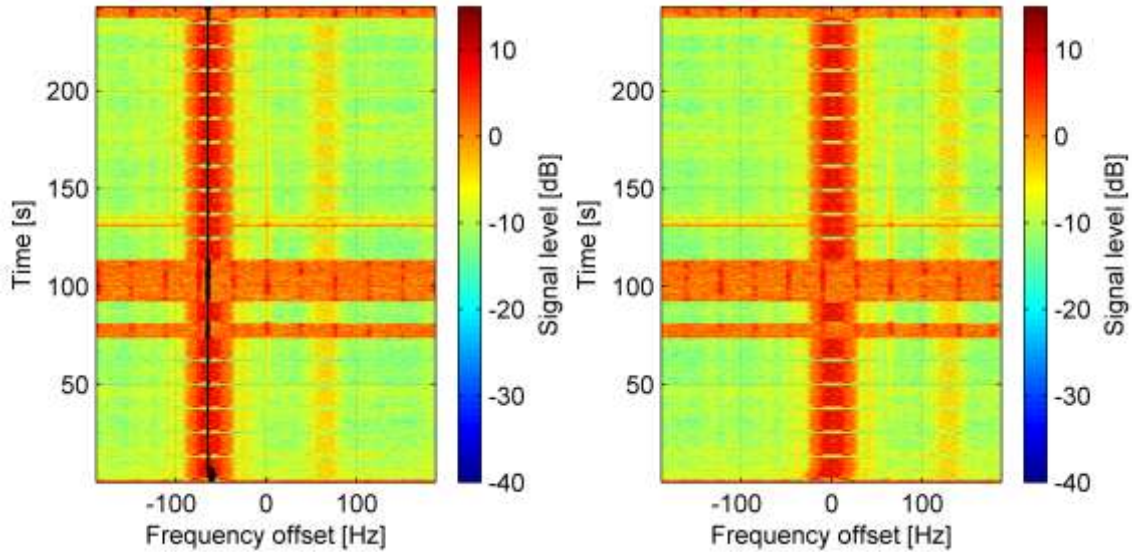
**Figure 6.15:** Simulink model of the experimental demodulator with Goertzel downconverter.



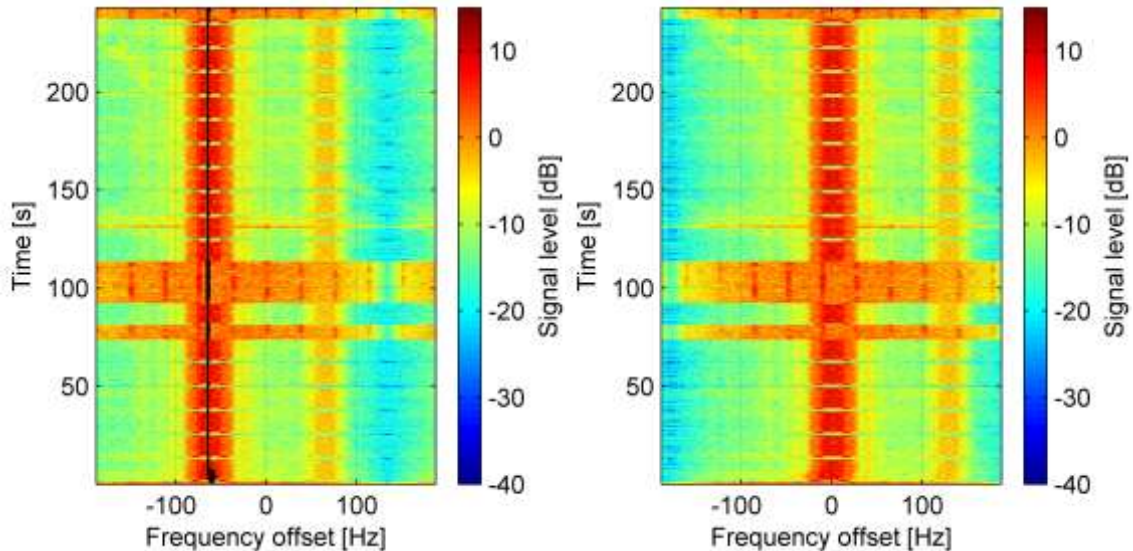
**Figure 6.16:** Waterfall signal spectrogram after downconversion by the Goertzel full-band downconverter (left) and carrier synchronized output of the Costas loop (right).

The filtration characteristics of the downconverters are visualized by the waterfall spectrograms. Figure 6.16 shows the spectrogram of the full-band Goertzel downconverter with the Costas loop frequency estimate (left with a black line) and the

spectrogram of the baseband signal. There is also visible a weaker signal at a complementary frequency, this is actually the -320 Hz signal complement of the real beacon signal. The beacon signal is at relatively low frequency within the sampled band, and therefore the two complements are very close and thus hard to be filtered out.



**Figure 6.17:** Waterfall signal spectrogram after downconversion by the Goertzel modified full-band downconverter (left) and carrier synchronized output of the Costas loop (right).

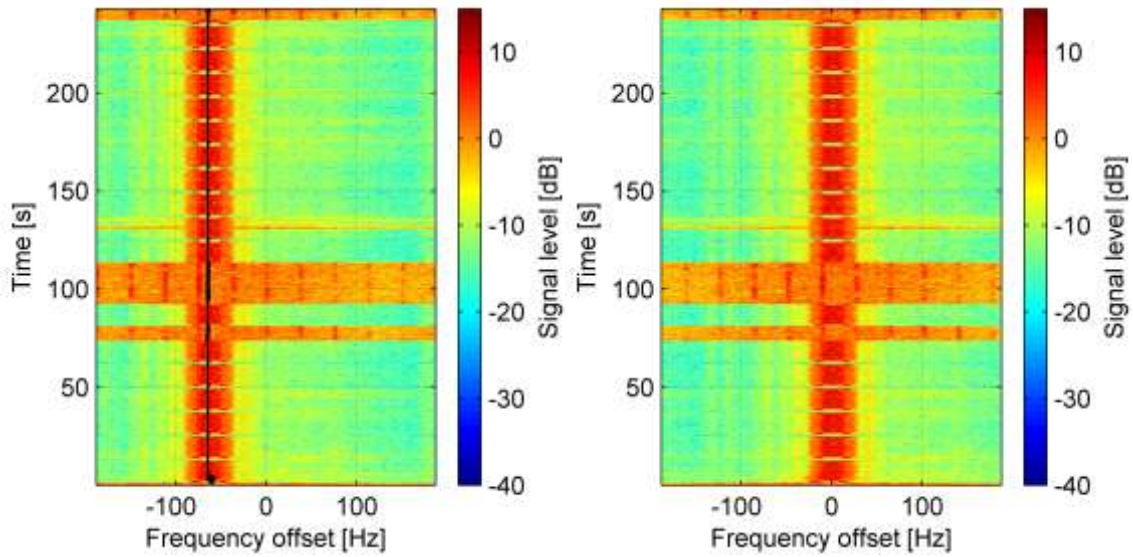


**Figure 6.18:** Waterfall signal spectrogram after downconversion by the Goertzel half-band downconverter (left) and carrier synchronized output of the Costas loop (right).

The following figures show the characteristics of each downconverter. The modified Goertzel downconverter has sharper attenuation of undesired bands and this is confirmed by a significantly lower level of an interfering component in Figure 6.17. The same situation can be observed in Figure 6.18 and Figure 6.19 depicting the results of the half-band (HB) versions of the downconverter. The Goertzel HB downconverter performs strong attenuation at the edges of the sampled region caused by the well-defined zeros at its transfer function. Unlike the modified Goertzel HB version, the interfering component stays recognizable. The modified version has strong attenuation

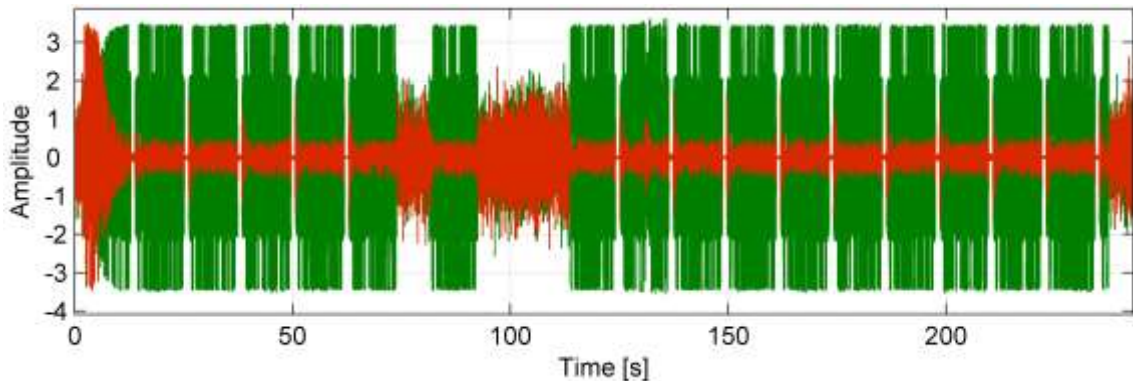


of the side bands, however, it should be remembered that the zeros are not defined well for aliasing minimization.



**Figure 6.19:** Waterfall signal spectrogram after downconversion by the Goertzel modified half-band downconverter (left) and carrier synchronized output of the Costas loop (right).

Figure 6.20 shows the sample signal after decimation and carrier synchronization. Due to the signal strength, the synchronization is performed excellently and there is no visible influence of any proposed downconverters.



**Figure 6.20:** Output synchronized signal from the Goertzel modified half-band downconverter and Costas loop phase synchronizer.

### 6.3. Summary

- The proposed frequency estimators – open and closed loop spectrum averaging and reference maximum likelihood – were tested on the recorded real signal from the PSAT satellite distorted by the strong Doppler frequency shift. The results showed that the open loop spectrum averaging has an unusual coverage bandwidth. The closed loop version has a much narrower acquisition bandwidth, but compared to the reference maximum likelihood, performs better acquisition with strong Doppler shifts. The price paid is worse noise levels of frequency estimates of both open and closed loop spectrum averaging estimators.

- Similar tests were performed on the downconversion algorithms based on the Goertzel algorithm proposed as a novel effective approach. The tests were performed on all four versions of the downconverters – Goertzel full-band, modified Goertzel full-band, Goertzel half-band and modified Goertzel half-band. Good functionality of all algorithms was proved and also expected filtration characteristics were confirmed to meet theory. The modified Goertzel downconverters are more suitable for wideband signals due to high attenuation of all aliased sidebands. The well-defined zeros of the Goertzel downconverters, on the other hand, provide extreme attenuation of aliased signals for narrowband signals.

## 7. Conclusion

The dissertation is focused on the development of new digital signal processing algorithms for software defined receivers of narrowband satellite signals. Particularly, the thesis firstly deals with effective downconversion including filtration and downsampling and secondly with carrier frequency estimation of signals distorted by a strong Doppler shift and AWGN. These are the two main objectives of the thesis and the third one comprises the modeling of the developed algorithms and experimental tests.

The first part of the thesis is an introduction to downconversion and frequency estimation algorithms described in literature and used in practice. There are also laid the foundations such as sampling theory and used spectral effective modulations for space communications.

Digital downconversion is discussed in section 4. The novel approach is based on the Goertzel algorithm, which is a special form of the discrete Fourier transform calculation. Downconversion is provided by undersampling and bandpass filtration. This approach is typically avoided, but the Goertzel algorithm offers operations to be performed very effectively. It has been shown that the filtration characteristic is comparable to the CIC filter, which is often used in FPGA-based systems. The Goertzel algorithm allows effective signal power estimation that is useful for AM detection and it was proved by the DCF77 receiver application. The drawback of undersampling based downconversion is the discrete conversion frequency suggesting the algorithm to be employed in multirate downconverters. It is possible to tune the filtration characteristic to the real frequency, but the power calculation is affected by strong fluctuations in this case. This issue was solved by special coefficients included in the calculation. The calculation is performed on the low-rate side that keeps the algorithm's efficiency high. The bandwidth of the bandpass filter and the decimation factor are inseparable characteristics. It was found that the special constellation of Goertzel filters allows to set the decimation factor independently on the filter bandwidth. The constellation of two parallel Goertzel filters was employed as the half-decimation Goertzel downconverters and helped to suppress aliasing during downsampling. The impulse response of the downconverter was improved by doubling the zero and pole of the original Goertzel filter. Consequently, the side-band attenuation of the frequency response was significantly increased. From these discoveries, four different downconverters were derived – the full-decimation Goertzel downconverter, the half-decimation Goertzel downconverter employing the parallelization of two Goertzel downconverters, the full-decimation and half-decimation modified Goertzel downconverters employing the improved version of the Goertzel downconverter. The algorithms were theoretically analyzed from the points of view of BER performance, computational demands, and quantization errors. The algorithms were modeled in MATLAB Simulink and the theoretical results were confronted with simulation results over the AWGN channel. Finally, the downconverters were subjected to tests with a real signal. All the results showed that the algorithms can be considered as useful for downconversion of narrowband signals within appropriate platforms as the first stage of multirate downconversion.



In section 5, we focused on carrier frequency estimation of BPSK modulated input signals. Firstly, the known theoretical foundations such as Cramer-Rao bounds and maximum likelihood approach to carrier parameter estimation were laid. The main contribution to the objective of establishing new frequency estimators for narrowband signals with strong Doppler shifts and a high level of AWGN is based on the cumulation of the covariance function of the signal spectrum magnitude and from this we call the algorithms “spectrum averaging”. Both open and closed loop versions of the spectrum averaging frequency estimators were introduced. The closed loop estimator was, at first, based on the open loop estimator put into the control loop, but strong nonlinearity in the form of maximum search function made adjustability hard and the overall performance of the estimator was poor compared to the ML estimator. This, finally, disqualified this approach. In the second version of the closed loop estimator, the maximum search function was replaced by a set of linear functions and it allowed averaging to be performed by the loop filter. This resulted in an estimator with parameters more comparable to the ML version. For simulations and tests on the real signal, the algorithms were modeled in the MATLAB Simulink and also for decoding the models of the Costas loop and the codes of offline decoders were prepared. The new SA open loop and SA closed loop estimators and the reference ML closed loop estimators were tested on a real satellite signal with a strong Doppler shift. Both ML open and closed loop estimators employ a pulse filter that removes as much noise as possible. The estimate is then fine, but the narrow pulse filter also narrows the acquisition range. The acquisition range is the field in which the SA estimators can compete and even beat the ML ones. The price paid is then higher levels of estimation noise.

At this point, all the objectives of the thesis can be considered as accomplished. However, the novel algorithms presented in the thesis will be studied deeply in real-time applications. There could be many unknown issues connected with the implementation of the algorithms and its transformation to fixed-point architecture.

## 8. References

- [1] HAMKINS, J., SIMON, M. K. *Autonomous software-defined radio receivers for deep space applications* [online]. Hoboken, N.J.: Wiley-Interscience, 2006, xxii, 435 pages. [cit. 2013-04-29]. ISBN 978-047-0082-126. Available at: [http://descanso.jpl.nasa.gov/Monograph/series9/Descanso9\\_Full\\_rev.pdf](http://descanso.jpl.nasa.gov/Monograph/series9/Descanso9_Full_rev.pdf)
- [2] INTERNATIONAL TELECOMMUNICATIONS UNION. *Radio regulations*. Edition of 2012. Geneva: ITU/RR/1.177, 2012. ISBN 978-926-1140-212.
- [3] TAYLOR, J., LEE, D. K., SHAMBAYATI, S. *Mars Reconnaissance Orbiter Telecommunications* [online]. Pasadena, California: Jet Propulsion Laboratory, 2006, 76 pages. [cit. 2015-03-04]. Available at: [http://descanso.jpl.nasa.gov/DPSummary/MRO\\_092106.pdf](http://descanso.jpl.nasa.gov/DPSummary/MRO_092106.pdf)
- [4] EDWARDS, Ch. D., Jr., JEDREY, T. C., SCHWARTZBAUM, E., DEVEREAUX, A. S., DEPAULA, R., DAPORE, M., FISCHER, T. W. The Electra proximity link payload for Mars relay telecommunications and navigation. *54th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law*. 2003.
- [5] REBOUD, O., DENIS, M., ORMSTON, T. MEX and the NASA landers and rovers on Mars: Sustaining a backup relay in an interplanetary network. *SpaceOps 2012 Conference*. 2012.
- [6] MAKOVSKY, A., ILOTT, P., TAYLOR, J. *Mars science laboratory telecommunications system design* [online]. Pasadena, California: Jet Propulsion Laboratory, 2009, 149 pages. [cit. 2015-03-21]. Available at: [http://descanso.jpl.nasa.gov/DPSummary/Descanso14\\_MSL\\_Telecom.pdf](http://descanso.jpl.nasa.gov/DPSummary/Descanso14_MSL_Telecom.pdf)
- [7] ExoMars orbiter core module completed. EUROPEAN SPACE AGENCY. *Robotic exploration of Mars* [online]. 2014. [cit. 2015-03-20]. Available at: <http://exploration.esa.int/jump.cfm?oid=53642>
- [8] TAYLOR, J. *Dawn telecommunications* [online]. Pasadena, California: Jet Propulsion Laboratory, 2009, 80 pages. [cit. 2015-03-21]. DESCANSO: Design and performance summary series. Available at: <http://descanso.jpl.nasa.gov/DPSummary/090924dawn-FinalCorrex--update5G.pdf>
- [9] *Space telecommunications radio system (STRS): Architecture standard*. Rev. 1.02.1, 2012. Cleveland, Ohio: National Aeronautics and Space Administration.
- [10] OBERG, J. Titan Calling. *IEEE Spectrum*. New York, 2004, vol. 41, No. 10: p. 28-33. DOI: 10.1109/MSPEC.2004.1338782. ISSN 0018-9235.
- [11] JONES, R. *Mars coordinated science networking demonstration* [online]. 2009, 9 pages. [cit. 2015-04-02]. Available at: <http://www.wkiri.com/research/papers/jones-mars-coordsci-09.pdf>
- [12] BEADLE, E. R., DYSON, T. Software-based reconfigurable computing platform (AppSTAR) for multi-mission payloads in spaceborne and near-space vehicles. In: *ERSA'12: The International conference on engineering of reconfigurable systems and algorithms*. Las Vegas, USA: WORLDCOMP, 2012, p. 25-31.
- [13] DOWNEY, J., KACPURA, T. Pre-flight testing and performance of a Ka-band software defined radio. *30th AIAA International Communications Satellite System Conference (ICSSC)*. 2012: 13 pages.

- [14] CINCINNATI ELECTRONICS. *C/TT-505 Transceiver: White paper* [online]. 2005, 2 pages. [cit. 2015-03-21]. Available at: [http://www.cinele.com/images/space\\_datasheets/c-tt-505.pdf](http://www.cinele.com/images/space_datasheets/c-tt-505.pdf)
- [15] CINCINNATI ELECTRONICS. *C/TT-510 Electra-lite: White paper* [online]. 2007, 2 pages. [cit. 2015-03-21]. Available at: [http://www.cinele.com/images/space\\_datasheets/c-tt-510.pdf](http://www.cinele.com/images/space_datasheets/c-tt-510.pdf)
- [16] NYQUIST, H. Certain topics in telegraph transmission theory. *Transaction A. I. E. E.* 1928, vol. 47, No. 2, p. 617-644. DOI: 10.1109/5.989875. ISSN: 0018-9219.
- [17] PAPOULIS, A. Generalized sampling expansion. *IEEE Transactions on Circuits and Systems*. 1977, vol. 24, No. 11, p. 652-654. DOI: 10.1109/TCS.1977.1084284. ISSN: 0098-4094.
- [18] KOHLENBERG, A. Exact interpolation of band-limited functions. *Journal of Applied Physics*. United States of America: American Institute of Physics, 1953, vol. 24, No. 12, p. 1432-1436. ISSN 1089-7550.
- [19] LINDEN, D. A. A discussion of sampling theorems. *Proceedings of the IRE*. 1959, vol. 47, No. 7, p. 1219-1226. DOI: 10.1109/JRPROC.1959.287354. ISSN: 0096-8390.
- [20] ZAPLATA, F., KASAL, M. Architektury a základní vzorkovací techniky SDR. *Elektrorevue* [online]. 2012, No. 43 [cit. 2013-04-14]. ISSN 1213-1539. Available at: <http://www.elektrorevue.cz/cz/download/architektury-a-zakladni-vzorkovaci-techniky-sdr>
- [21] BROWN, J. L. jr. First-order sampling of bandpass signals - A new approach. *IEEE Transactions on Information Theory*. 1980, vol. 26, No. 5, p. 613-615. DOI: 10.1109/TIT.1980.1056240. ISSN: 0018-9448.
- [22] VAUGHAN, R. G., SCOTT, N. L., WHITE, D. R. The theory of bandpass sampling. *IEEE transactions on signal processing: a publication of the IEEE Signal Processing Society*. 1991, vol. 39, No. 9, p. 1973-1984. DOI: 10.1109/78.134430. ISSN 1053-587x.
- [23] YEN, J. L. On Nonuniform Sampling of Bandwidth-Limited Signals. *IRE Transactions on Circuit Theory*. 1956, vol. 3, No. 4, p. 251-257. DOI: 10.1109/TCT.1956.1086325. ISSN: 0096-2007.
- [24] COULSON, A. J. A Generalization of Nonuniform Bandpass Sampling. *IEEE transactions on signal processing: a publication of the IEEE Signal Processing Society*. 1995, vol. 43, No. 3. DOI: 10.1109/78.370623. ISSN 1053-587x.
- [25] BEUTLER, F. J. Alias-Free Randomly Timed Sampling of Stochastic Processes. *IEEE Transactions on Information Theory*. 1970, vol. 16, No. 2, p. 147-152. DOI: 10.1109/TIT.1970.1054435. ISSN: 0018-9448.
- [26] MASRY, E. Alias-Free Sampling: An Alternative Conceptualization and Its Applications. *IEEE Transactions on Information Theory*. 1978, vol. 24, No. 3. DOI: 10.1109/TIT.1978.1055889. ISSN: 0018-9448.
- [27] SAYINER, N., SORENSEN, H. V., VISWANATHAN, T. R. A New Signal Acquisition Technique. *Proceedings of the 35th Midwest Symposium on: Circuits and Systems*. 1992, p. 1140-1142. DOI: 10.1109/MWSCAS.1992.271169.
- [28] SHARMA, N., SREENIVAS, T. V. Sparse Signal Reconstruction Based on Signal Dependent Non-Uniform Samples. *IEEE International Conference on: Acoustics, Speech and Signal Processing (ICASSP)*. 2012, p. 3453-3456. DOI: 10.1109/ICASSP.2012.6288659. ISSN: 1520-6149.
- [29] TROPP, J. A., LASKA, J. N., DUARTE, M. F., ROMBERG, J. K., BARANIUK, R. G. Beyond Nyquist: Efficient Sampling of Sparse Bandlimited Signals. *IEEE Transactions on Information Theory*. 2010, vol. 56, No. 1, p. 520-544. DOI: 10.1109/TIT.2009.2034811. ISSN: 0018-9448.
- [30] SIMON, M. K. *Bandwidth-Efficient Digital Modulation with Application to Deep-Space Communications* [online]. Jet Propulsion Laboratory, California Institute of Technology, 2001, 237 pages. [cit. 2013-04-12]. Available at: <http://descanso.jpl.nasa.gov/Monograph/series3/complete1.pdf>

- [31] *Space engineering: Radio frequency and modulation*. The Netherlands: ESA Requirements and Standards Division, 2009, 81 pages. Available at: [http://www.ecss.nl/forums/ecss/dispatch.cgi/standards/showFile/100742/d20090309095455/No/ECSS-E-ST-50-05C\\_Rev.1%286March2009%29.pdf](http://www.ecss.nl/forums/ecss/dispatch.cgi/standards/showFile/100742/d20090309095455/No/ECSS-E-ST-50-05C_Rev.1%286March2009%29.pdf)
- [32] BURNS, P. *Software defined radio for 3G*. Boston: Artech House, 2003, xix, 279 pages. ISBN 15-805-3347-7.
- [33] CREANEY, S., KOSTARNOV, I. *Designing efficient digital up and down converters for narrowband systems: Application Note XAP1113 v. 1.0* [online]. Xilinx, 2008. [cit. 2015-03-20]. Available at: [http://www.xilinx.com/support/documentation/application\\_notes/xapp1113.pdf](http://www.xilinx.com/support/documentation/application_notes/xapp1113.pdf)
- [34] SCHILCHER, T. RF applications in digital signal processing [online]. 2008: 35 pages. [cit. 2015-07-01]. Available at: <http://cds.cern.ch/record/1100538/files/p249.pdf>
- [35] TUTTLEBEE, W., (ed.). by. *Software defined radio enabling technologies: enabling technologies*. New York: J. Wiley, 2002, xxxvii, 402 pages. Wiley series in software radio. ISBN 04-708-4600-3.
- [36] KENINGTON, P. B. *RF and baseband techniques for software defined radio*. Boston: Artech House, 2005, xiii, 332 pages. Artech House mobile communications series. ISBN 1-58053-793-6.
- [37] ARSLAN, H., (ed.). *Cognitive radio, software defined radio, and adaptive wireless systems*. Online-Ausg. Dordrecht: Springer, 2007. ISBN 978-140-2055-423.
- [38] KHAN, S. A. *Digital design of signal processing systems: A practical approach*. Hoboken: John Wiley, 2011. ISBN 978-047-0974-698.
- [39] ZAPLATA, F. *Algoritmy zpracování signálu na platformě AVR32*. Brno, 2011, 65 pages, Diploma thesis. FEEC, Brno University of Technology.
- [40] LYONS, R. G. *Streamlining digital signal processing: A tricks of the trade guidebook*. Hoboken, N.J.: Wiley-Interscience, 2007, xiv, 322 pages. ISBN 978-047-0131-572.
- [41] MEYR, H., MOENECLAHEY, M. *Digital Communication Receivers: Synchronization, Channel Estimation, and Signal Processing*. vol. 2. Hoboken, NJ: Wiley-Interscience, 1997, 827 pages. ISBN 04-712-0057-3.
- [42] HOGENAUER, E. An economical class of digital filters for decimation and interpolation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. 1981, vol. 29, No. 2: p. 155-162. DOI: 10.1109/TASSP.1981.1163535. ISSN 0096-3518.
- [43] XIONG, F. *Digital modulation techniques*. 2nd ed. Boston, MA: Artech House, 2006, xxi, 1017 pages. Artech House telecommunications library. ISBN 1-58053-863-0.
- [44] MENGALI, U., D'ANDREA, A. N. *Synchronization techniques for digital receivers*. New York: Plenum Press, 1997, xiii, 520 pages. ISBN 03-064-5725-3.
- [45] KAY, S. M. *Fundamentals of statistical signal processing: estimation theory*. Upper Saddle River, N.J.: Prentice Hall PTR, 1993, xii, 595 pages. Prentice Hall signal processing series. ISBN 01-334-5711-7.
- [46] MESSERSCHMITT, D. Frequency Detectors for PLL Acquisition in Timing and Carrier Recovery. *IEEE Transactions on Communications*. 1979, vol. 27, No. 9: p. 1288-1295. DOI: 10.1109/TCOM.1979.1094553. ISSN 0096-2244.
- [47] NATALI, F. AFC Tracking Algorithms. *IEEE Transactions on Communications*. 1984, vol. 32, No. 8: p. 935-947. DOI: 10.1109/TCOM.1984.1096152. ISSN 0096-2244.
- [48] CLASSEN, F., MEYR, H., SEHIER, P. Maximum likelihood open loop carrier synchronizer for digital radio. *Proceedings of ICC '93 - IEEE International Conference on Communications*. IEEE, 1993, vol. 1: p. 493-497. DOI: 10.1109/ICC.1993.397313. ISBN 0-7803-0950-2.

- [49] TRETTER, S. Estimating the frequency of a noisy sinusoid by linear regression (Corresp.). *IEEE Transactions on Information Theory*. 1985, vol. 31, No. 6: p. 832-835. DOI: 10.1109/TIT.1985.1057115. ISSN 0018-9448.
- [50] KAY, S. A fast and accurate single frequency estimator. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. 1989, vol. 37, No. 12: p. 1987-1990. DOI: 10.1109/29.45547. ISSN 00963518.
- [51] FITZ, M. P. Planar filtered techniques for burst mode carrier synchronization. *IEEE Global Telecommunications Conference GLOBECOM '91: Countdown to the New Millennium. Conference Record*. IEEE, 1991, vol. 1: p. 365-369. DOI: 10.1109/GLOCOM.1991.188412. ISBN 0-87942-697-7.
- [52] LUISE, M., REGGIANNINI, R. Carrier frequency recovery in all-digital modems for burst-mode transmissions. *IEEE Transactions on Communications*. 1995, vol. 43, No. 2/3/4: p. 1169-1178. DOI: 10.1109/26.380149. ISSN 0090-6778.
- [53] GARDNER, F. M. *Phaselock techniques*. 3rd ed. Hoboken, NJ: John Wiley, 2005, xxii, 425 pages. ISBN 04-714-3063-3.
- [54] OPPENHEIM, A. V., SCHAFER, R. W., BUCK, J. R. *Discrete-time signal processing*. 2nd ed. Upper Saddle River, N.J.: Prentice Hall, 1999, xxvi, 870 pages. ISBN 01-375-4920-2.
- [55] DONADIO, M. P. *CIC filter introduction* [online]. 2000, 6 pages. [cit. 2011-11-26]. Available at: <http://dspguru.com/sites/dspguru/files/cic.pdf>
- [56] ZAPLATA, F., KASAL, M. Software defined DCF77 receiver. *Radioengineering*, 2013, No. 4, p. 1211-1217. ISSN: 1210- 2512.
- [57] ZAPLATA, F., KASAL, M. SDR implementation for DCF77. *23rd International Conference Radioelektronika*. IEEE, 2013, p. 340-345. DOI: 10.1109/RadioElek.2013.6530943. ISBN 978-1-4673-5519-3.
- [58] DE JESUS, M. A., TEIXEIRA, M., VICENTE, L., RODRIGUEZ, Y. Nonuniform Discrete short-time Fourier transform a Goertzel filter bank versus a FIR filtering approach. *49th IEEE International Midwest Symposium on Circuits and Systems*. IEEE, 2006, p. 188-192. DOI: 10.1109/MWSCAS.2006.382241. ISBN 1-4244-0172-0. ISSN 1548-3746.
- [59] ZAPLATA, F., KASAL, M. Efficient Spectral Power Estimation on an Arbitrary Frequency Scale. *Radioengineering*. 2015, vol. 24: p. 178-184. DOI: 10.13164/re.2015.0178. ISSN 1210-2512.
- [60] JAN, J. *Medical image processing, reconstruction, and restoration: Concepts and methods*. Boca Raton: Dekker/CRC Press, 2005, xxiii, 730 pages. Signal processing and communications, 24. ISBN 08-247-5849-8.
- [61] RICE, M. *Digital communications: a discrete-time approach*. Upper Saddle River, N.J.: Pearson/Prentice Hall, 2009, xxii, 778 pages. ISBN 978-013-0304-971.
- [62] SIMON M. K., ALOUINI K. M.-S. *Digital communication over fading channels: A unified approach to performance analysis*. New York: John Wiley, 2000. ISBN 04-712-0069-7.
- [63] SKLAR, B. *Digital communications: fundamentals and applications*. 2nd ed. Upper Saddle River, N.J.: Prentice-Hall PTR, 2001, xxiv, 1079 pages. ISBN 01-308-4788-7.
- [64] MARTINEZ, P. PSK31: A new radio-teletype mode. *ARRL* [online]. 1999, p. 3-9. [cit. 2015-06-04] Available at: <http://www.arrl.org/files/file/Technology/tis/info/pdf/x9907003.pdf>
- [65] PSAT Transponder WEB Specification. *Experimental satellites laboratory, DREL, FEEC Brno University of Technology* [online]. 2015. [cit. 2015-07-22]. Available at: <http://www.urel.feec.vutbr.cz/esl/files/Projects/PSAT/P%20sat%20transponder%20WEB%20spec02.htm>

## Curriculum Vitae

Name: Filip ZÁPLATA  
Born: March 28<sup>th</sup> 1987 in Chrudim  
Contact: f.zaplata@gmail.com

### Education

2011 – 15 **Brno University of Technology / Department of Radio Electronics**  
Ph.D. study of Electronics and Communication  
Dissertation on Advanced algorithms for satellite communication signal processing

2009 – 11 **Brno University of Technology / Department of Radio Electronics**  
Master's study of Electronics and Communication  
Diploma thesis on Signal Processing Algorithms on AVR32 Platform

2006 – 09 **Brno University of Technology / Department of Radio Electronics**  
Bachelor's study of Electronics and Communication  
Bachelor's thesis on Digital filtration with 8-bit processors

### Experience

9/11 – 11/12 **VESLA s.r.o. Pardubice**  
R&D engineer developing embedded software for signal processing and control in commercial DVB systems

### Languages

English, Russian