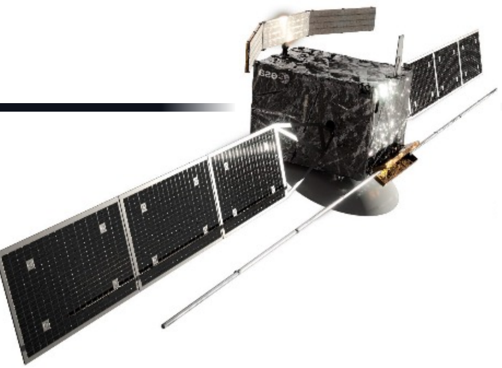


OFF-A

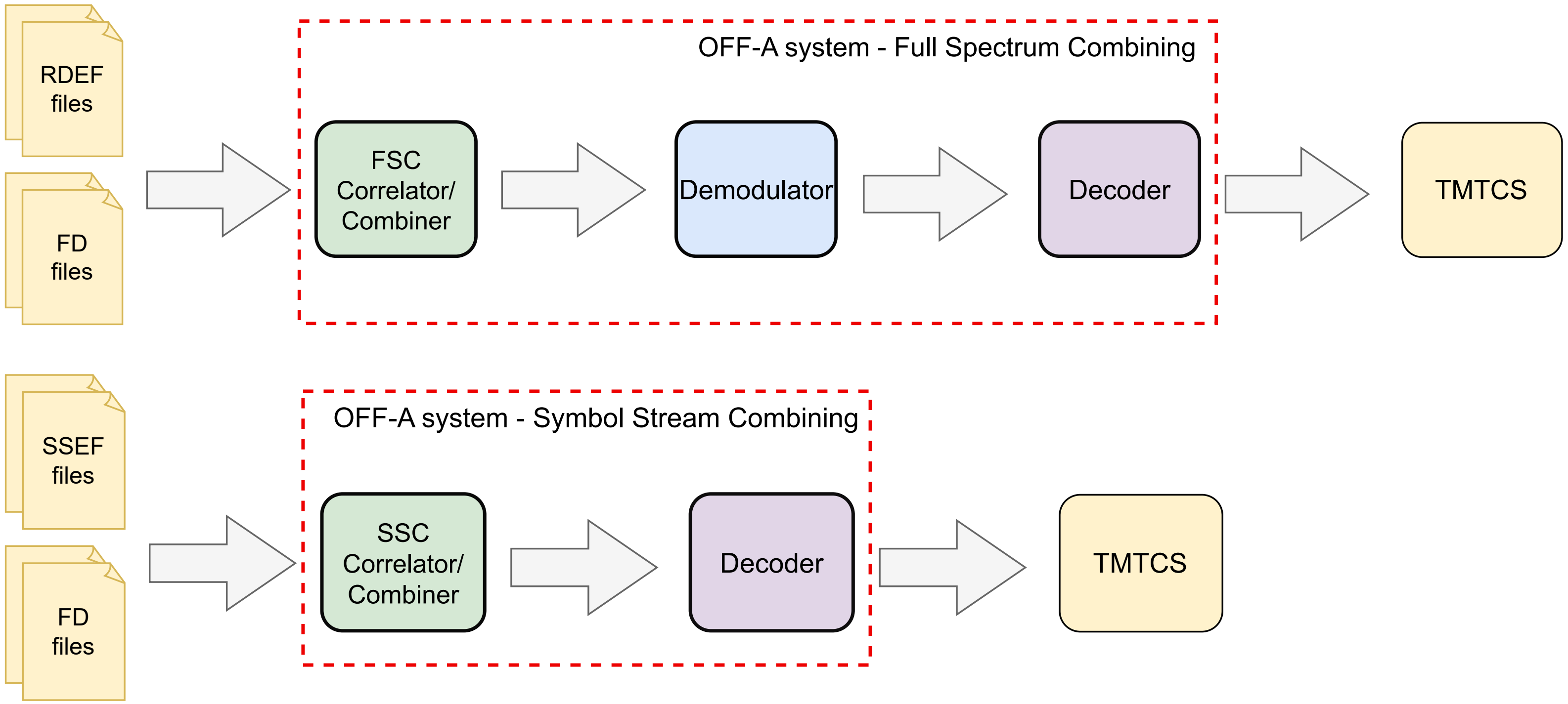
Offline Antenna Correlator for EnVision

Final Presentation– 31st Jan 2024

SCOPE OF THE STUDY

Antenna array: signal acquisition at two GS and combination to increase the SNR

OFF-A is a Software Offline Correlator for Antenna Array



Primary objectives:

- Combination from 2 stations (FSC/SSC)
- Demodulation
  - GMSK
  - PCM/BPSK/PM
  - PCM/SP-L
- Decoder
  - Turbo Code rate 1/2, 1/4, 1/6 k=8920

Secondary objectives:

- Extension to (up to) 4 stations
- Quasi-real time (collocated stations)

Target TRL: 5

Breadboard tested in a relevant environment

LEGACY

The legacy of OFF-A is the project PROTOCOL (Prototype of Offline Correlator for Arraying of Large Aperture Antennas).

Changes/challenges with respect to PROTOCOL:

- GMSK with very high rate for Envision – 40 Msps (I&Q samples at 43.75 MHz)
- Front end
- Interface with TMTCS
- Real-time oriented
- Stringent processing time capabilities
  - → internal parallelization of correlator/combiner, demodulator and decoder
  - → global parallelization (core services working in parallel)
  - → challenging infrastructure (nodes, pods, queues)
  - → smart memory handling
  - → smart "communication" between services

Sequential Mono-thread Resource-limited	**New paradigm** → **> 50k lines of code**	Distributed system Parallelized Multi-thread

Code completely written again in C++, Python and JavaScript.

Team

Arpsoft	Management&Support Correlator/Combiner Decoder optimization Front end Infrastructure	Alessandro Ardito Giulia Moretti Stefano Finocchiaro Edoardo Bini Stefano Milani and Alessio Parmeggiani
Zelinda	Demodulator	Paul Maguire, Anbazhagan Aroumont
Università Politecnica delle Marche	Support for decoder*	Franco Chiaraluce, Marco Baldi, Massimo Battaglioni
esa	Support	Marco Lanucara, Marco Menapace, Jorge Quintanilla, Maria Montagna, André Lofaldli, Roman Sevcik, Holger Dreihahn

* Design of new decoder was out the scope. Decoder from PROTOCOL has been optimized.

Outline

Schedule and WPs

Front-end (video)

Infrastructure

Core Services

- Correlator/combiner
- Demodulator
- Decoder

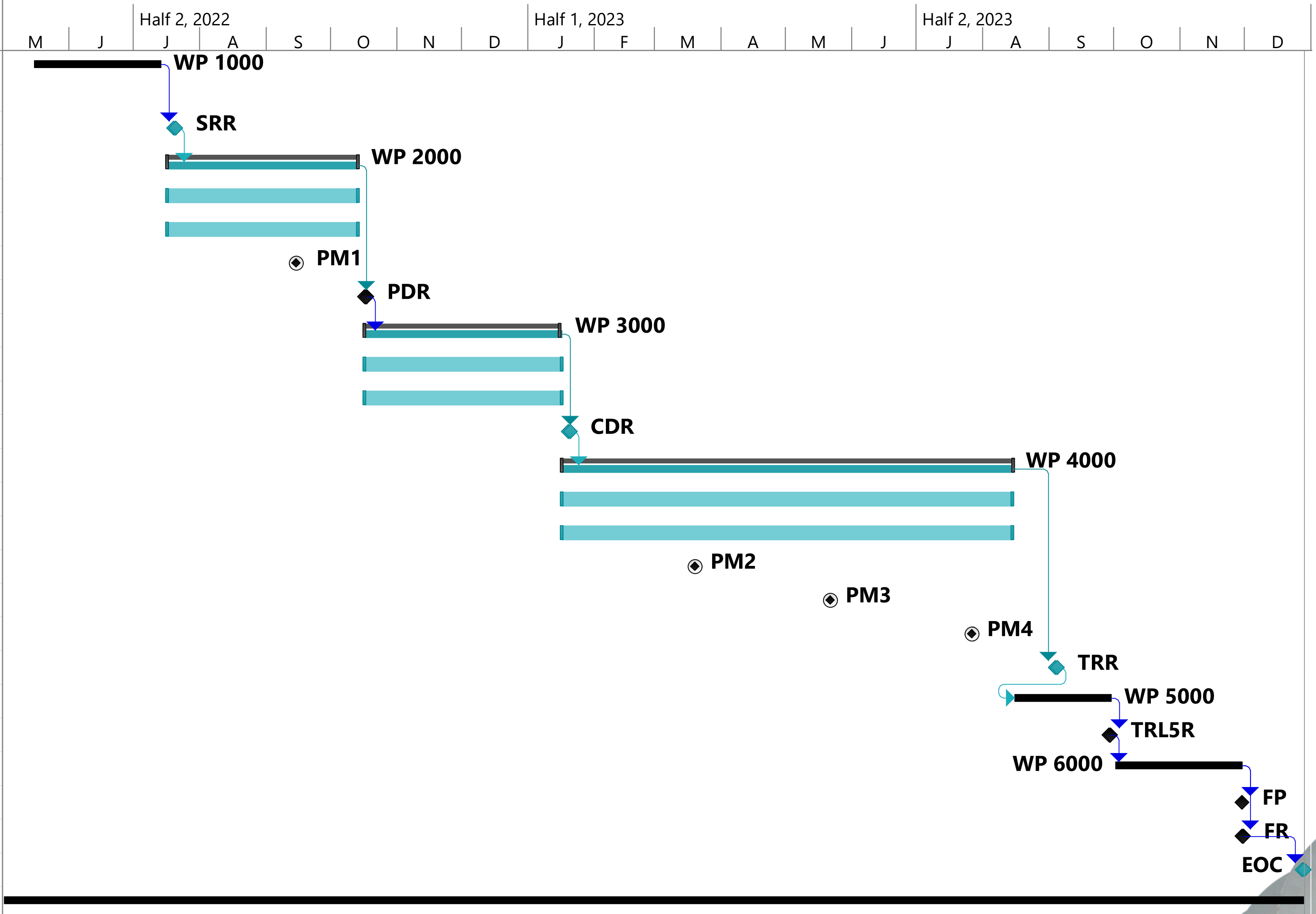
End-to-end tests and performances

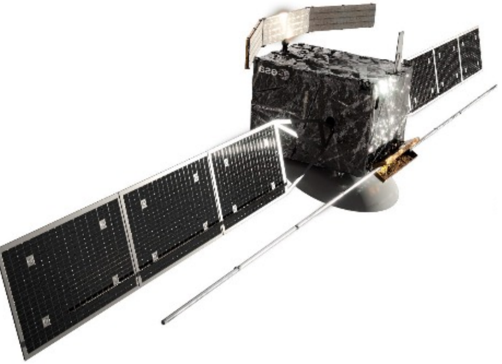
Tests with Exomars and SolO

Considerations and future work

Schedule and WPs

Task Name	Start	Finish
WP 1000 Requirements consolidation and system trade-off	**16/05/22**	**14/07/22**
SRR (System Requirements Review)	20/07/22	21/07/22
WP 2000 System specifications	**17/07/22**	**14/10/22**
WP 2100 Offline correlator specifications	17/07/22	14/10/22
WP 2200 Simulator specifications	17/07/22	14/10/22
PM1 (Progress Meeting 1)	15/09/22	15/09/22
PDR (Preliminary Design Review)	17/10/22	18/10/22
WP 3000 System design	**17/10/22**	**16/01/23**
WP 3100 Offline correlator design	17/10/22	17/01/23
WP 3200 Simulator design	17/10/22	17/01/23
CDR (Critical Design Review)	20/01/23	20/01/23
WP 4000 System implementation	**17/01/23**	**15/08/23**
WP 4100 Offline correlator implementation	17/01/23	15/08/23
WP 4200 Simulator implementation	17/01/23	15/08/23
PM2 (Progress Meeting 2)	20/03/23	21/03/23
PM3 (Progress Meeting 3)	22/05/23	22/05/23
PM4 (Progress Meeting 4)	27/07/23	28/07/23
TRR (Test Readiness Review)	04/09/23	04/09/23
WP 5000 Test and Validation	**16/08/23**	**29/09/23**
TRL5R (TRL-5 Achievement Review)	29/09/23	29/09/23
WP 6000 Final presentation preparation	**02/10/23**	**29/11/23**
FP (Final Presentation)	30/11/23	30/11/23
FR (Final Review)	30/11/23	30/11/23
EOC (End Of Contract)	28/12/23	28/12/23
WP 7000 Management	**02/05/22**	**28/12/23**





OFF-A Front-end

Frontend

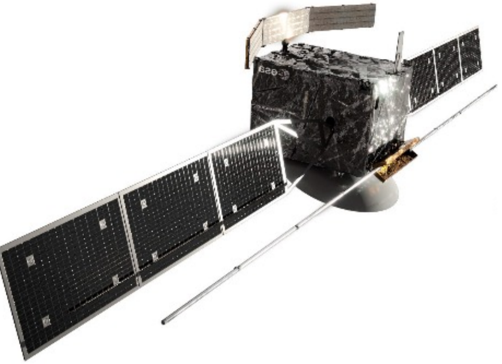
The Frontend manages the interactions between the user and the tool, mainly during the job creation phase and its execution.

Main objectives:

- Easy to use and immediate for the user;
- Provide the best user experience;
- Responsiveness and best performance (loading time/rendering).

Technologies:

- Framework7 v7.1.5
- React v18.2.0



OFF-A Infrastructure

Hardware & Cluster Setup

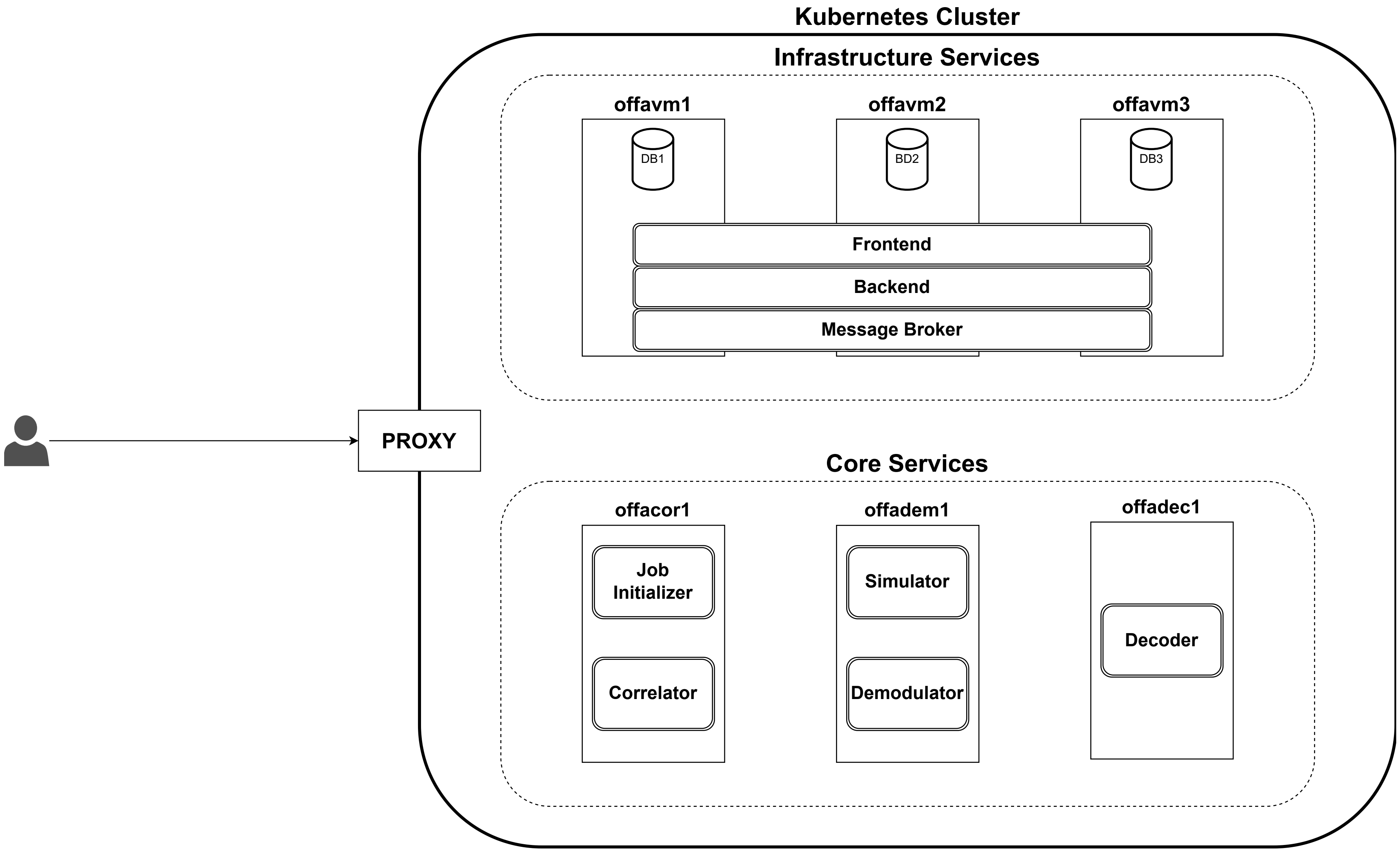
OFF-A is deployed on a total of **6 machines**:

- 3 Virtual Machines
  - Red Hat Enterprise Linux 9.2 (Plow) - 5.14.0-284.11.1.el9_2.x86_64
  - 4 vCPUs
  - 16 GB RAM
  - 500 GB HHD Storage
- 3 Physical Machines
  - Red Hat Enterprise Linux 9.2 (Plow) - 5.14.0-284.25.1.el9_2.x86_64
  - 16 Core, 32 Thread 4.5GHz (AMD Ryzen 9 7950X)
  - 64 GB RAM
  - 4000 GB SSD Storage (WD_BLACK SN850X)

The OFF-A cluster is created using **microk8s** (v1.26), a low-ops, minimal production implementation of **Kubernetes**.

- Distributed system
- Easy to scale
- Technology independent
- Services and components scaling and distribution

Components Overview



Database Management System - CouchDB

Apache CouchDB is an open-source document-oriented **NoSQL database**.

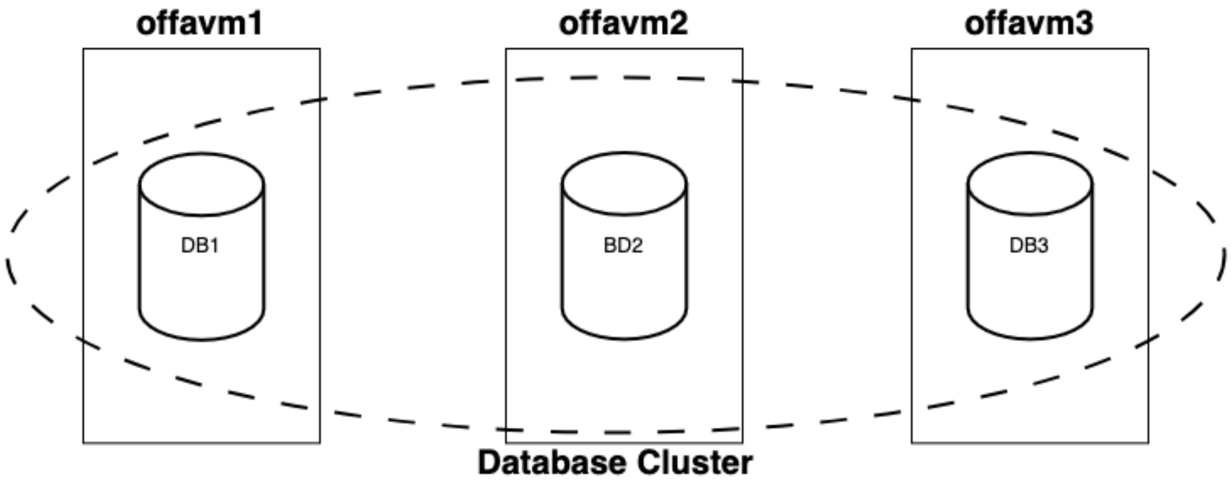
- JSON documents organized in Databases.
- Mango Queries for optimized search.

It stores all the necessary data.

- Users
- Jobs Input Data and Results
- Auxiliary data

Cluster of 3 instances

- One instance per VM
- **Redundancy**
- Fault resistant (1 node down)

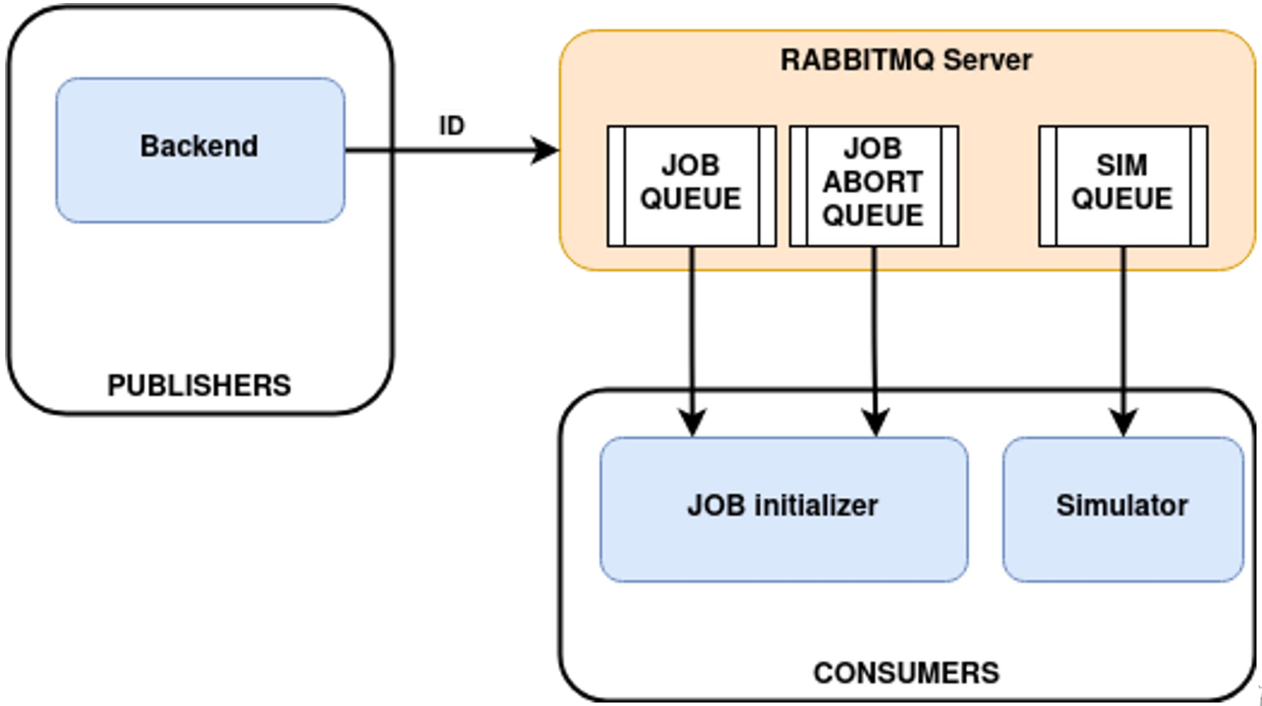


MESSAGE BROKER - RABBITMQ

RabbitMQ is an open-source message-broker software implementing the **Advanced Message Queuing Protocol** (AMQP). Deployed on one of the VM (managed by K8S).

Used to exchange **asynchronous messages** between the OFF-A Services, using different queues for each task.

- job_queue: maintain submitted jobs not yet started. (By design one job is started only after the previous one finished to avoid conflicts with different modalities for subsequent jobs.)
- job_abort_queue: maintain the jobs that the users aborted.
- sim_queue: maintain submitted simulation not yet started.



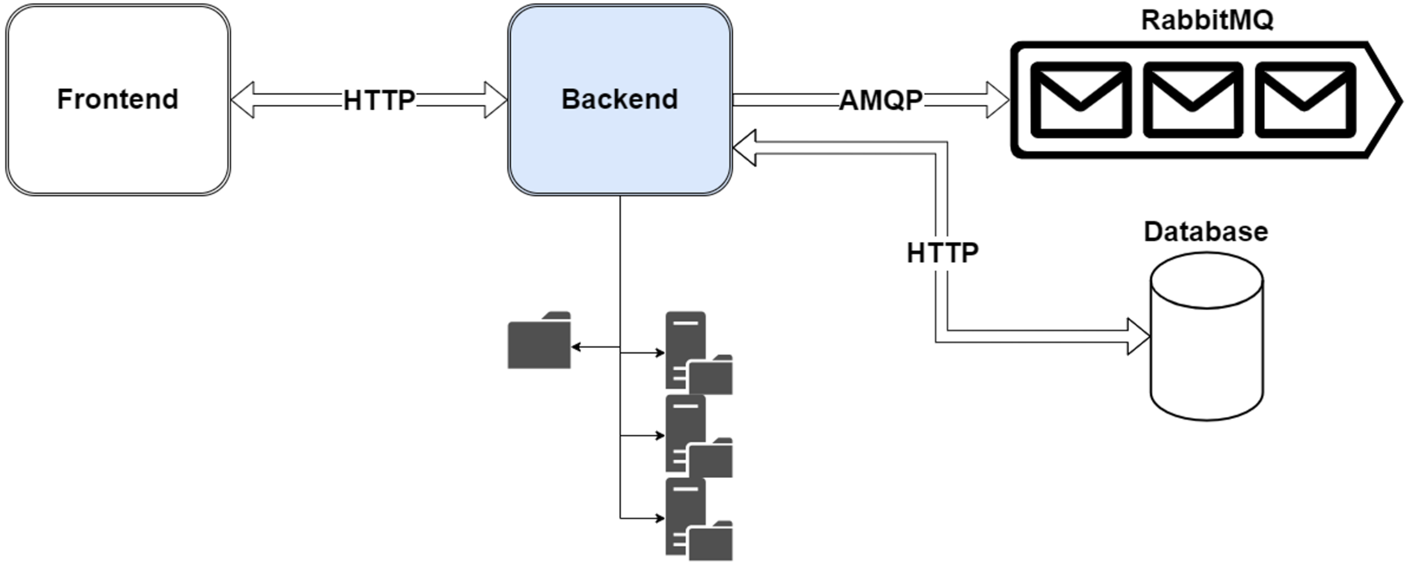
Backend - Overview

The Backend is developed in Python 3. It is responsible for:

- Handling **authentication** and authorization of users.
- Offering **API** for Fronted (e.g.: retrieve job list, start a run, etc.).
- **Querying** the DB and format data for GUI rendering.
- Handling the **search** of job-related files, both in local and remote machines.
- **Generating** PDF reports.
- **Triggering** the run of a process.

The backend offers the following interfaces:

- **HTTP**: for communication from/to Frontend and Database.
- **AMQP**: for publishing messages in the Message Broker.



BACKEND - BLANK SCHEMA

The Backend makes use of a **Blank Schema approach** to create and organize the job and simulation creation form.

Pros:

- The GUI handles only the rendering of the form.
- Allows **dynamic** and reactive form.
- Allows to handle all the **attributes** of the input fields, depending on previous inputs:
    - Visibility
    - Enable/Disable
    - Admissible range
    - Input type (text, number, etc.)
    - Pre-filling

Cons:

- Add a **layer of complexity** wrt static form rendering.
- Need interaction FE-BE at each change in the form.
- Need of **fine optimization** in the backend not to disrupt the UX.

Core Component Data Exchange - RPC & ZeroC ICE

The Job Initializer and the Core Components make use of the **Remote Procedure Protocol** (RPC) to exchange the data. RPC is a request–response protocol. An RPC is initiated by the client, which sends a request message to a known remote server to execute a specified procedure with supplied parameters.

The **Internet Communications Engine** (ICE) is an object-oriented RPC framework that helps us build distributed applications with minimal effort. ICE manages:

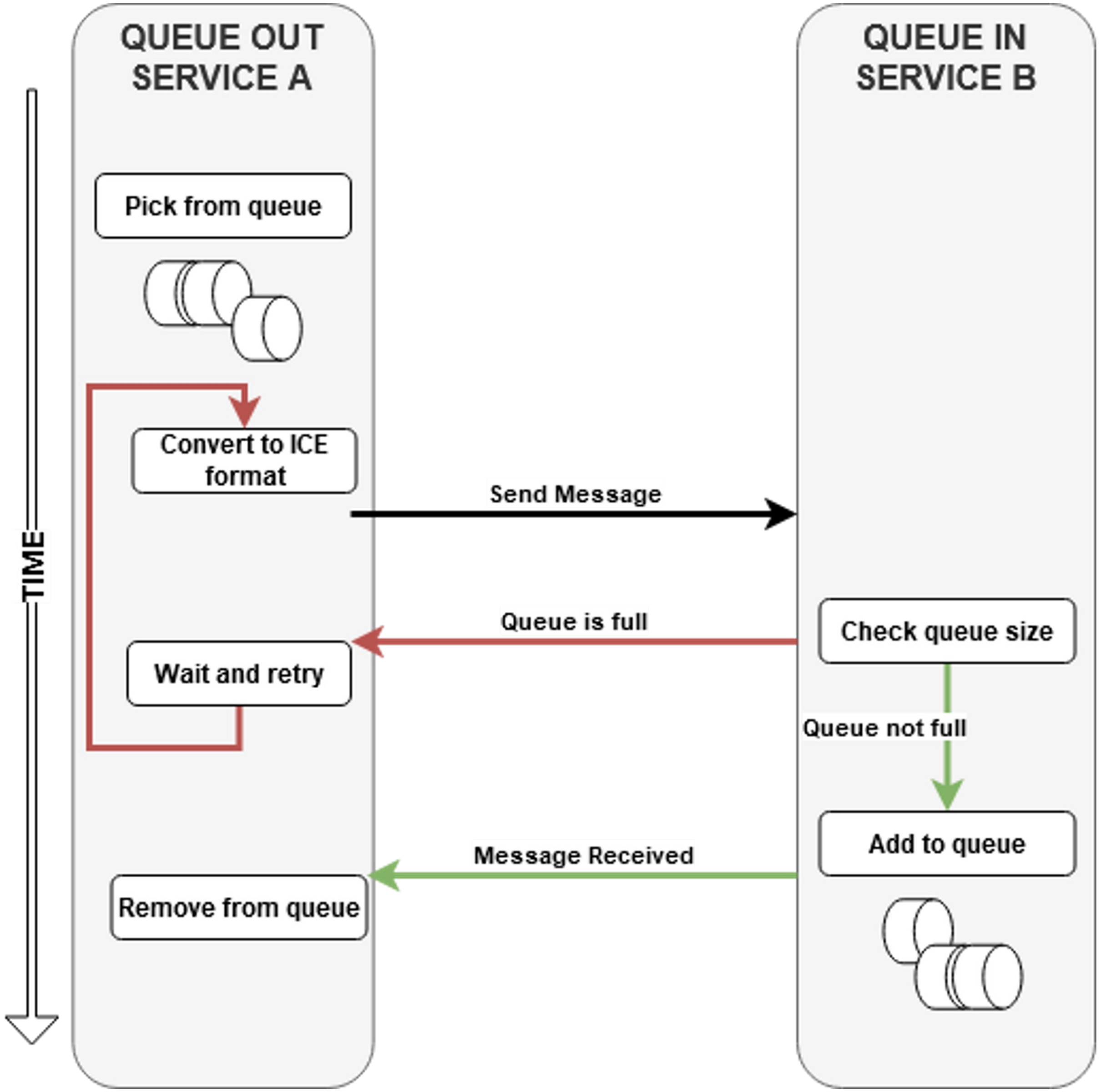
- Connection loss
- Retry
- Timeout

The data is organized in **Chunks and Records**. Each Chunk has some parameters shared between all the records it is composed of.

RPC and ICE are used to:

- Initialization of a new chunk.
- Reception of a new record.
- Notify a job abortion.
- Notify a job failure.

Core Component Data Exchange - FIFO Queue

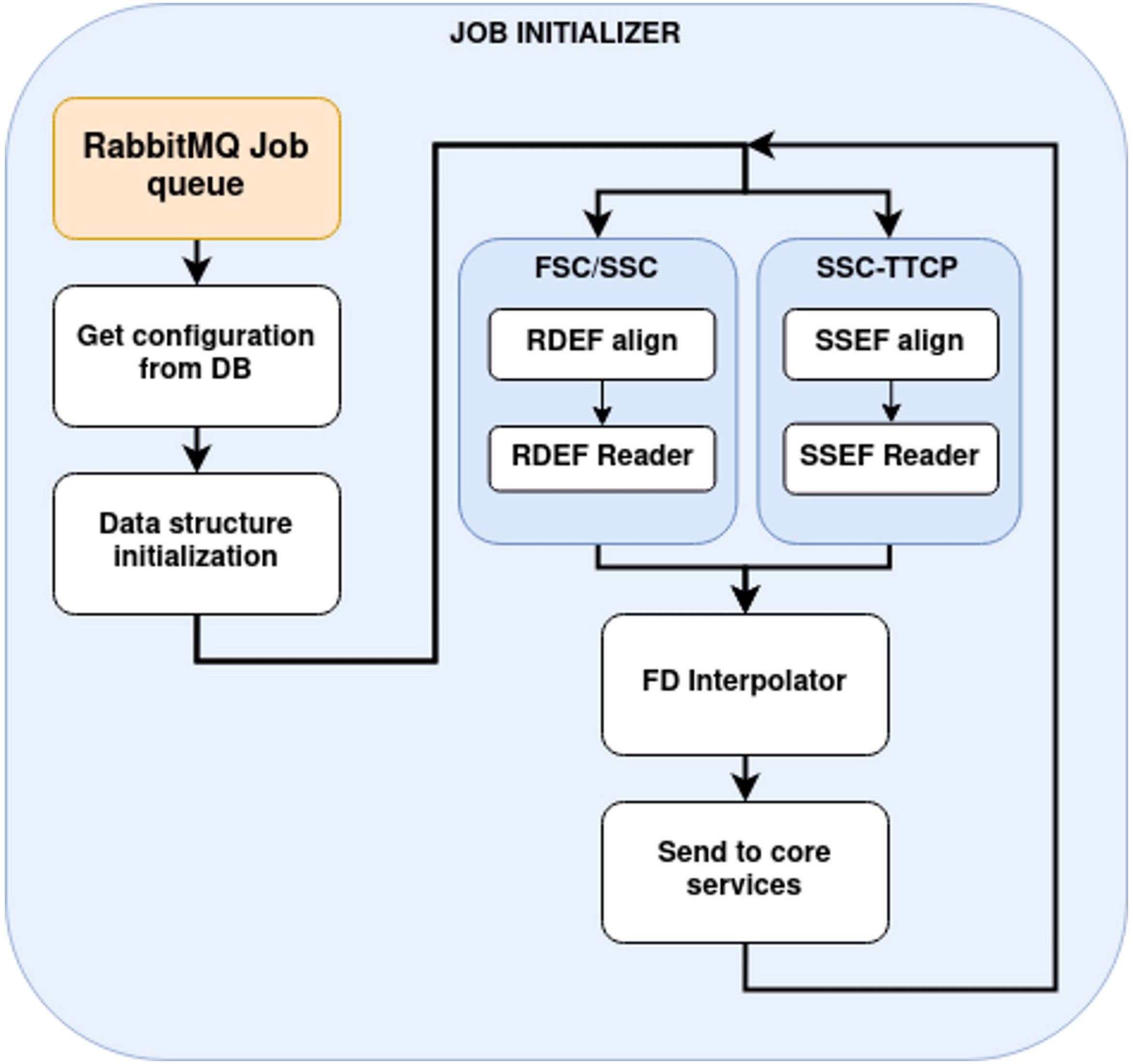


On top of RPC and ICE a **FIFO Queue** is used to store and manage both new chunks and records.

The FIFO Queue:

- Receives and sends data **without blocking** the core components computations.
- Allows **parallel computations** of the core components.
- Intercepts and manages failures while sending (not handled by Ice).
- Checks the memory (RAM) usage **avoiding Out of Memory (OOM)** states.
- **Mitigates bottlenecks** on data transfer.

Job Initializer



Job Initializer pipeline

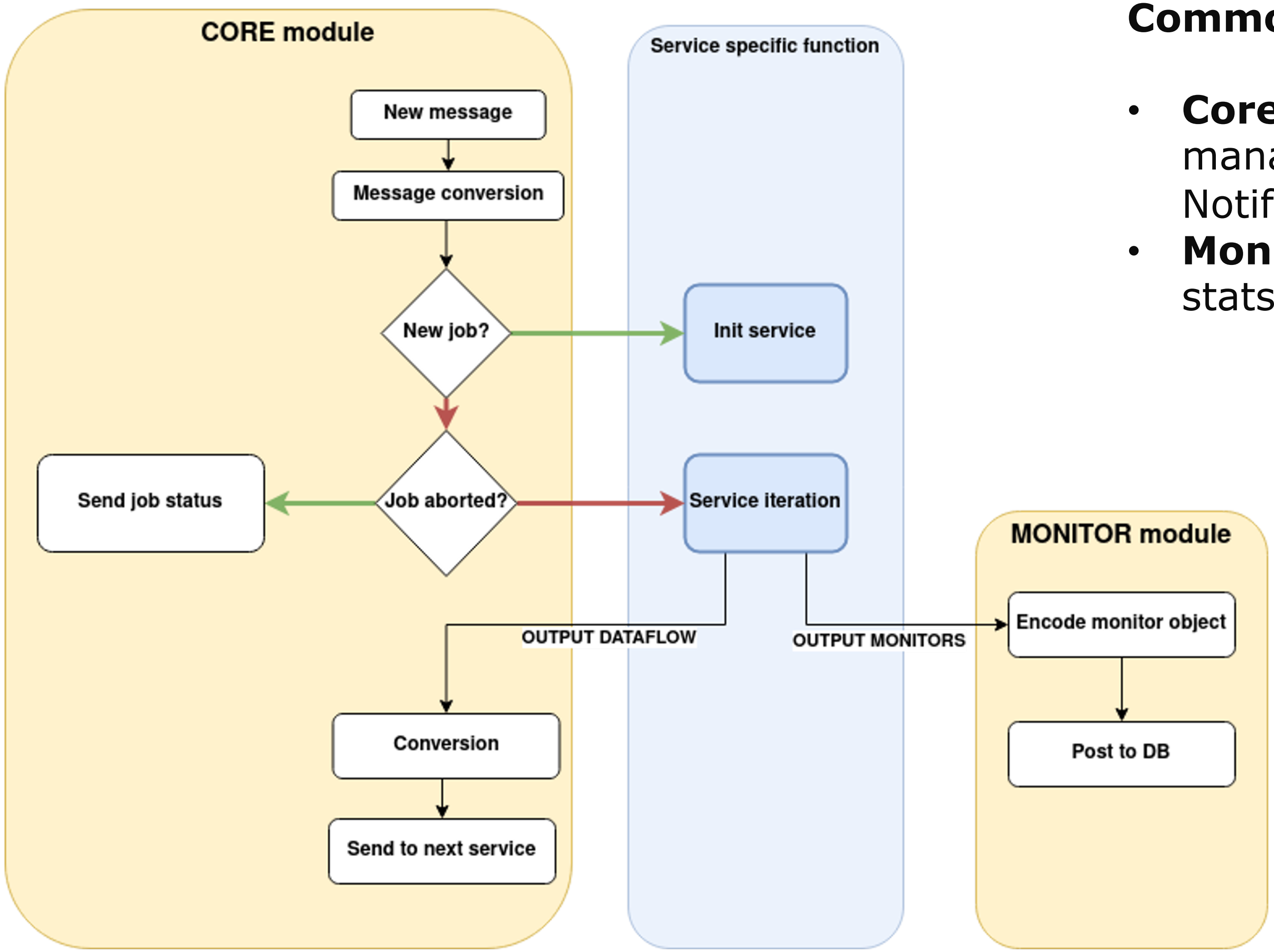
Initializes data structure

- Reads RDEF files
- Stores useful parameters about read RDEF files, to prefill subsequent jobs' form using the same file/files.
- Reads SSEF files
- Manages Radio Science Mode with RDEF files containing partial or multiple dataflows.

Manages the different aspects of job progression

- Start job
- Job aborting
- Job failure

Core Services Integration

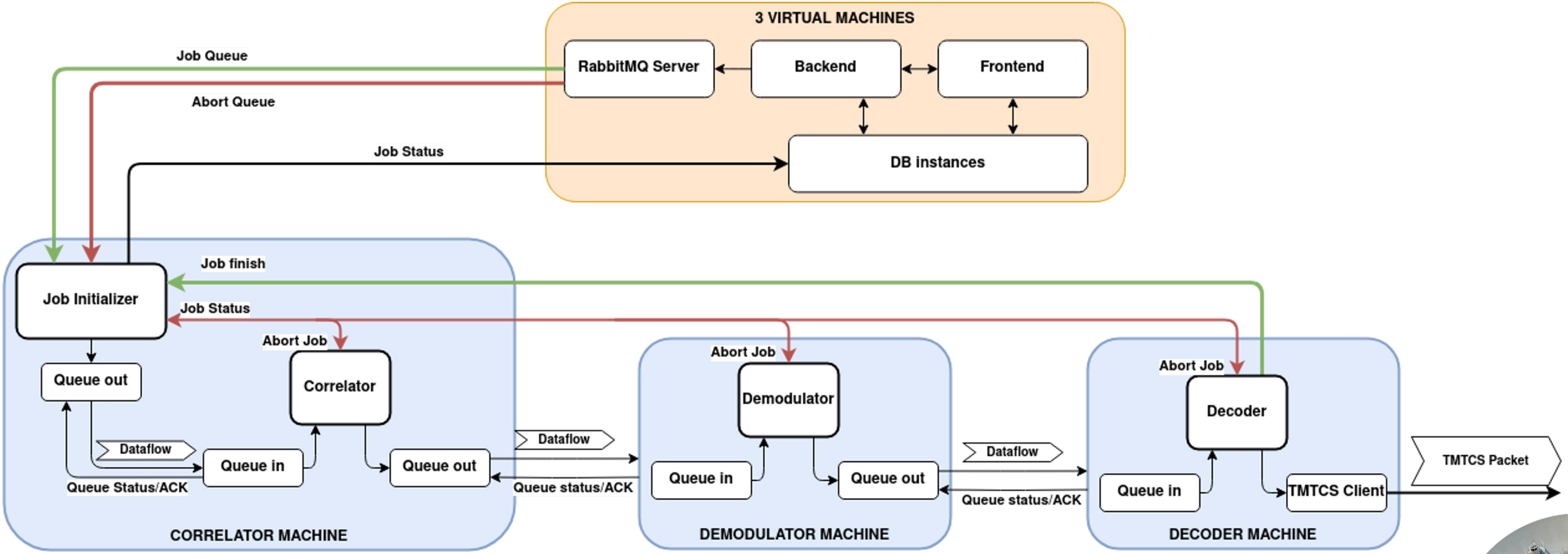


Common modules between all services.

- **Core Module**: handles service initialization and manage iterations.
  Notify other services if job is aborted or failed.
- **Monitor Module**: get monitor objects with service stats and add them to the DB.

Process Flow: FSC Job

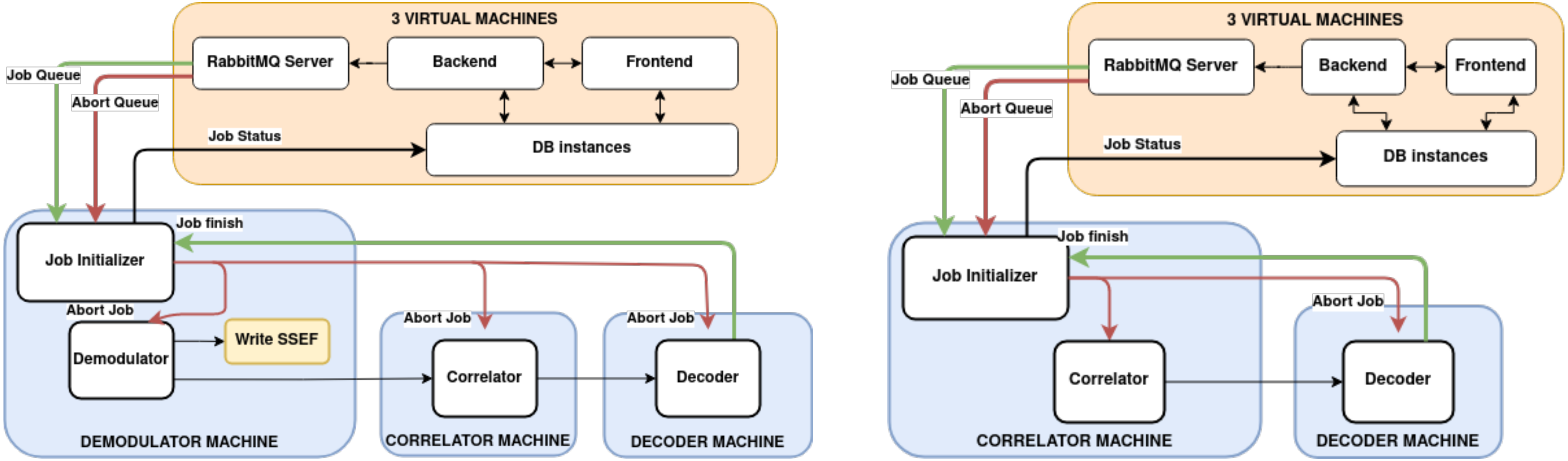
Process Flow in case of **FSC Job**.
The core services can run totally in parallel minimizing the processing time.



OFF-A architecture and message flow for an FSC job

Process Flow: SSC Job

Process Flow in case of **SSC Job.** Currently SSC can also run from RDEF.
The core services can run totally in parallel minimizing the processing time.



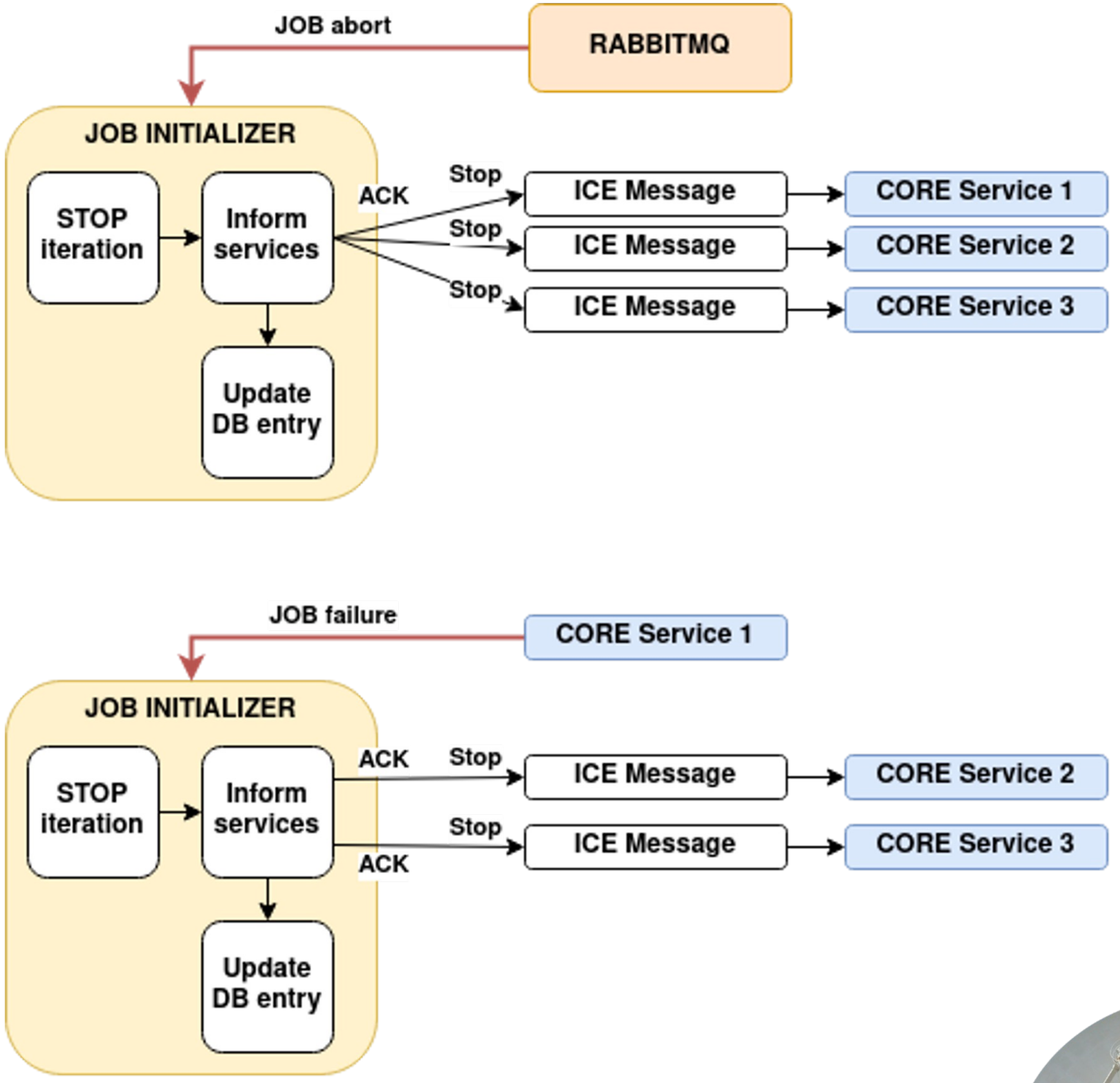
OFF-A architecture and message flow for an SSC job

Abort & Job failure

Due to the different processes and architecture complexity managing of errors is not trivial.

A job can stop in different ways:

- It finishes **successfully**
- It is **aborted** by the user
- It fails because of a **handled error**
- It fails because of an **unexpected error**
- Cluster and pod properties are exploited to manage these cases

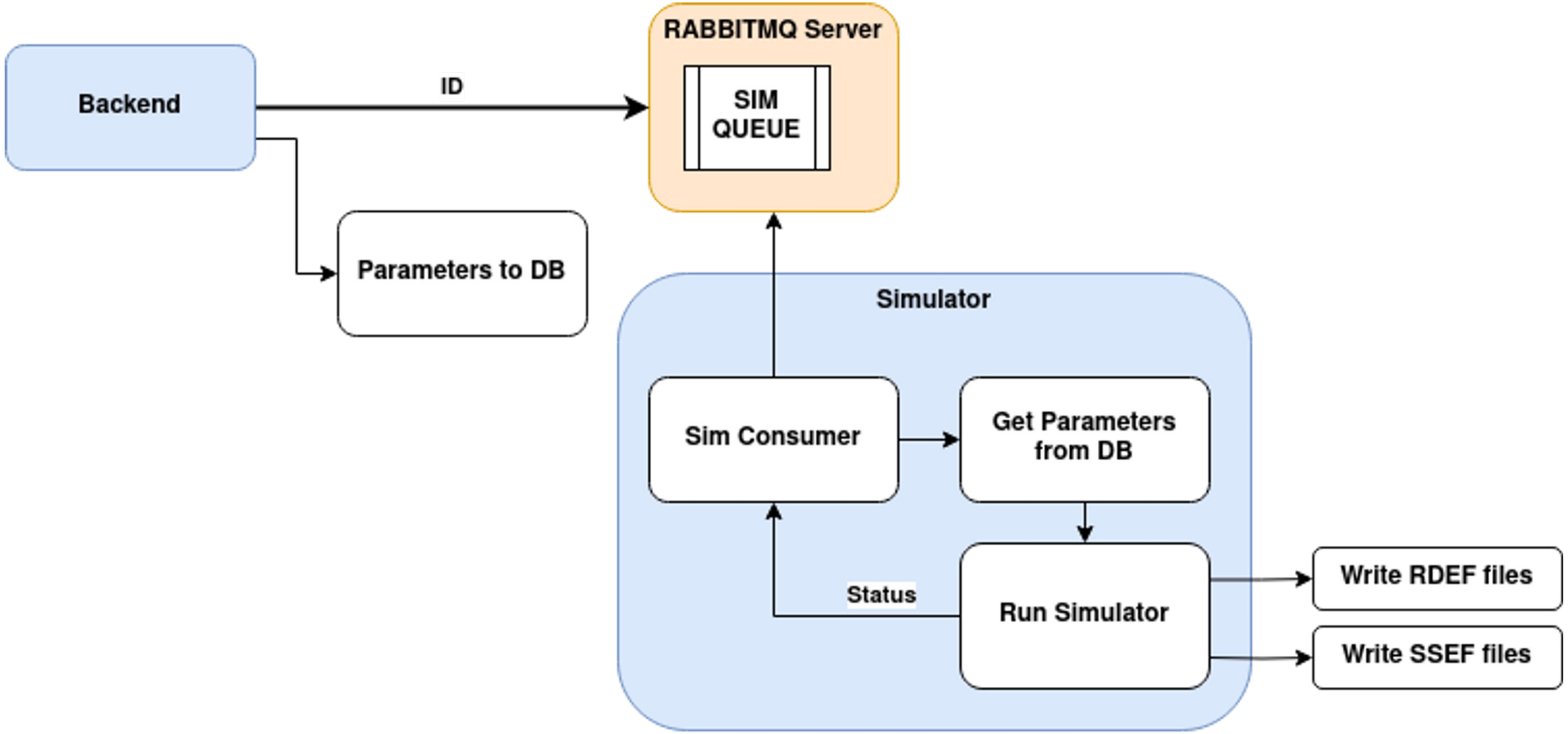


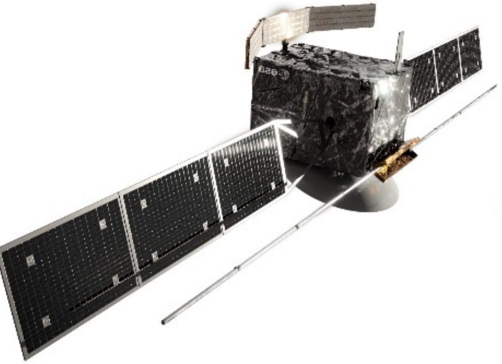
Behavior in case of abort or job failure

SIMULATOR

The simulator uses a similar approach.

- RabbitMQ Queue for **storing scheduled simulations**.
- Consumer to **retrieve parameters** from RabbitMQ and DB.
- Managing of success and failure





OFF-A core services: correlator/combiner

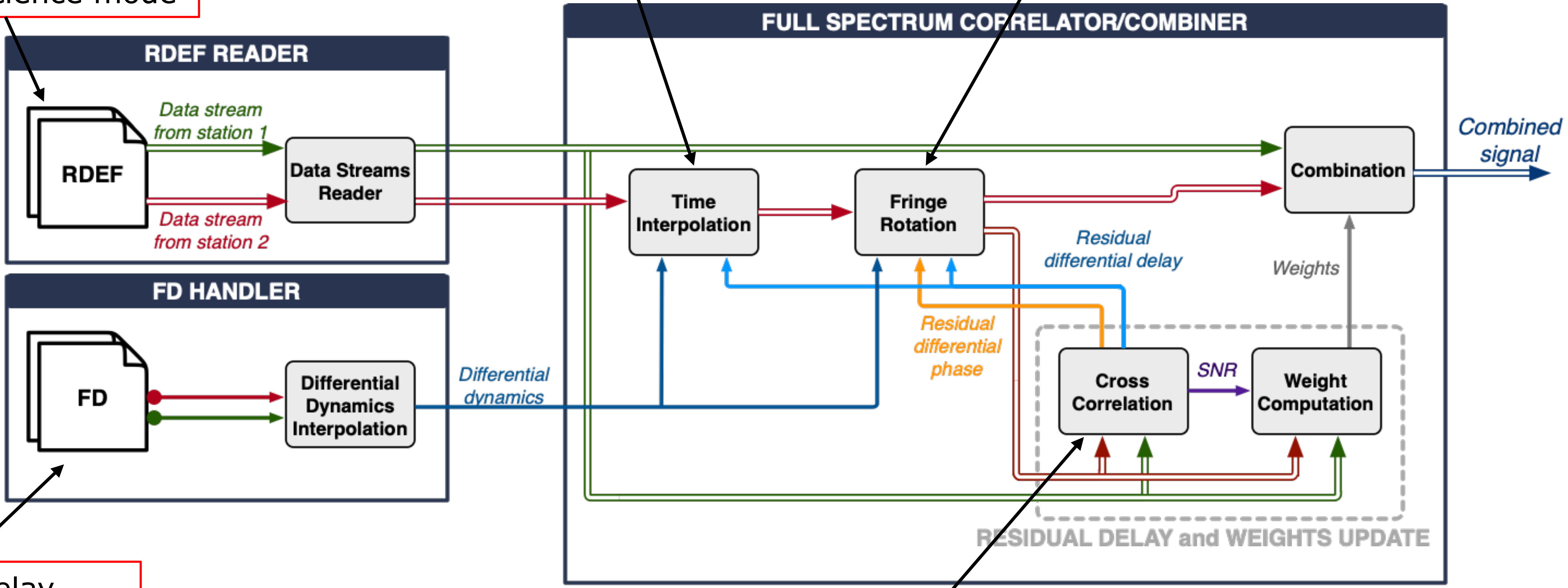
CORRELATOR/COMBINER FSC STANDARD: ALGORITHM

FSC standard: residual delay/phase computed with cross-correlation. Used for GMSK.

CCSDS Blue Book
- DDOR mode
- Radio Science mode

Poli-phase filter with 128 bins and 64 coefficients

Phase adjustment and LO alignment



Differential delay computed iteratively from the range at both stations (provided by FD as DDOR input files)

High symbol/sampling rate → low correlation time

Correlation function interpolated with a parabola with points up to half of the maximum

CORRELATOR/COMBINER FSC STANDARD: TIME CONSTANTS

To **automate** as much as possible the correlation process, some time constants have been introduced.

- Integration time for the cross-correlation is computed with a target correlation SNR of 14 dB. This define a sub-chunk.

$$SNR_{corr} = SNR_1 \cdot SNR_2 \cdot 2BT = \frac{S}{N_{01}} \cdot \frac{S}{N_{02}} \cdot \frac{2T}{B}$$

  Integration time is conveniently quantized to 50 ms, 200 ms, 1 s or ceil(T) if > 1 s (since the RDEF stores data in batches of 1 second)

- The chunk size is defined as the time span containing 100 sub-chunks for the weights’ computation (if G/T tables are used time span is 4 times the correlation if T>1 s and 4 records if T<1 s).

- Phase/delay update rate is set to min(1,correlation time). No need to update more frequently than 1 s.

- Weights are computed every 300 seconds.

Correlator/Combiner FSC Standard: weights computation (I)

The weights that maximize the combination gain are A/P_N (being A the amplitude of the signal and P_N the noise power).

The formulation for the computation of the weights and SNR has been provided by ESA.

Recorded signals

$$x_1 = A_1 y + n_1$$
$$x_2 = A_2 y + n_2$$

Build 1-bit signals

$$u_k = f(x_{k,r}) + i \cdot f(x_{k,i}), k = 1,2 \quad \text{f=sign()}$$

Cross-correlation

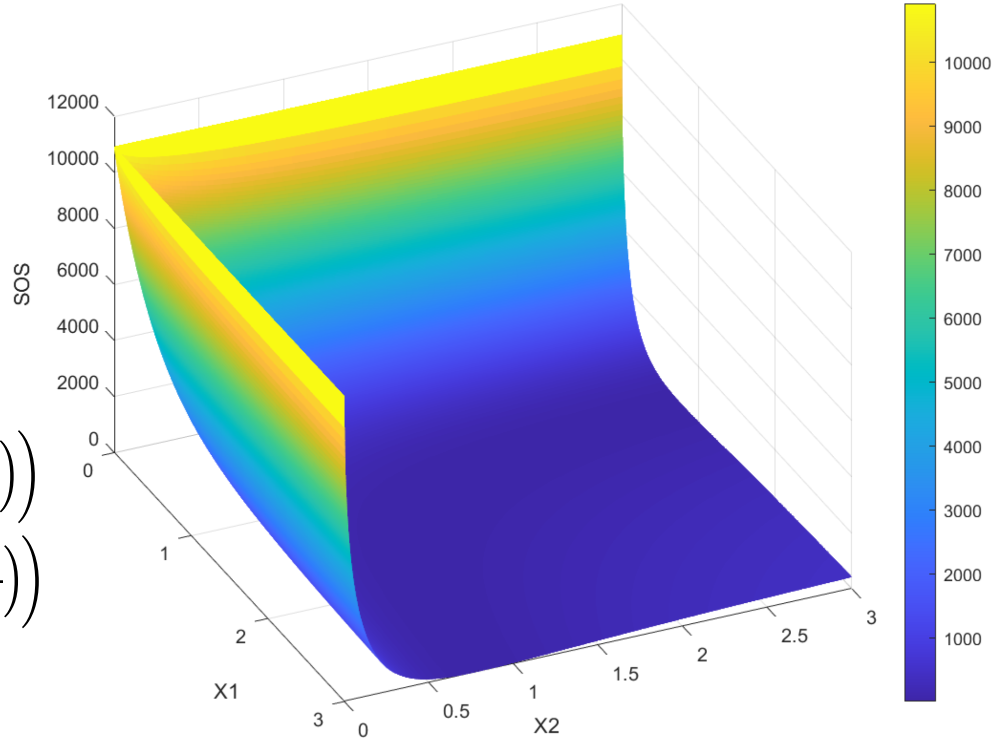
$$\rho_{11} = E[u_1 x_1^*]$$
$$\rho_{22} = E[u_2 x_2^*]$$
$$\rho_{12} = E[u_1 x_2^*]$$
$$\rho_{21} = E[u_2 x_1^*]$$

Non-linear system

$$\rho_{11} = \frac{2\sqrt{P_{N,1}}}{\sqrt{\pi}} e^{-\frac{SNR_1}{2}} \left((1 + SNR_1) I_0\left(\frac{SNR_1}{2}\right) + SNR_1 I_1\left(\frac{SNR_1}{2}\right) \right)$$
$$\rho_{22} = \frac{2\sqrt{P_{N,2}}}{\sqrt{\pi}} e^{-\frac{SNR_2}{2}} \left((1 + SNR_2) I_0\left(\frac{SNR_2}{2}\right) + SNR_2 I_1\left(\frac{SNR_2}{2}\right) \right)$$
$$\rho_{12} = \frac{2\sqrt{P_{N,2}}}{\sqrt{\pi}} \sqrt{SNR_1 SNR_2} e^{-\frac{SNR_1}{2}} \left(I_0\left(\frac{SNR_1}{2}\right) + I_1\left(\frac{SNR_1}{2}\right) \right)$$
$$\rho_{21} = \frac{2\sqrt{P_{N,1}}}{\sqrt{\pi}} \sqrt{SNR_1 SNR_2} e^{-\frac{SNR_2}{2}} \left(I_0\left(\frac{SNR_2}{2}\right) + I_1\left(\frac{SNR_2}{2}\right) \right)$$

Cost function (SOS)

$$\sum (\rho - \rho_{est})^2$$



CORRELATOR/COMBINER FSC STANDARD: WEIGHTS COMPUTATION (2)

Implementation (self consistent, no information on the SNR from the user):

For the first computation:

- SNR_{min} computed from symbol rate and coding scheme, using the Es/No at FER=1E-5

- Search range from $SNR_{min}/2$ to $SNR_{min}*16$ (mostly for test purposes where the Es/No may be much larger than the one required for FER=1E-5)

- For 8- and 4- bits data, system of 2 equations with 2 unknows (SNR1 and SNR2, using the auto-correlation function to express the noise power P_N a function of the SNR). For 2- bits, system of 4 equations with 4 unknows (noise power P_N search range from 60% to 140% of the reference value computed with the auto-correlation function). For 2- bits re-quantization is needed.

- Computation of the SOS and determination of the minimum.

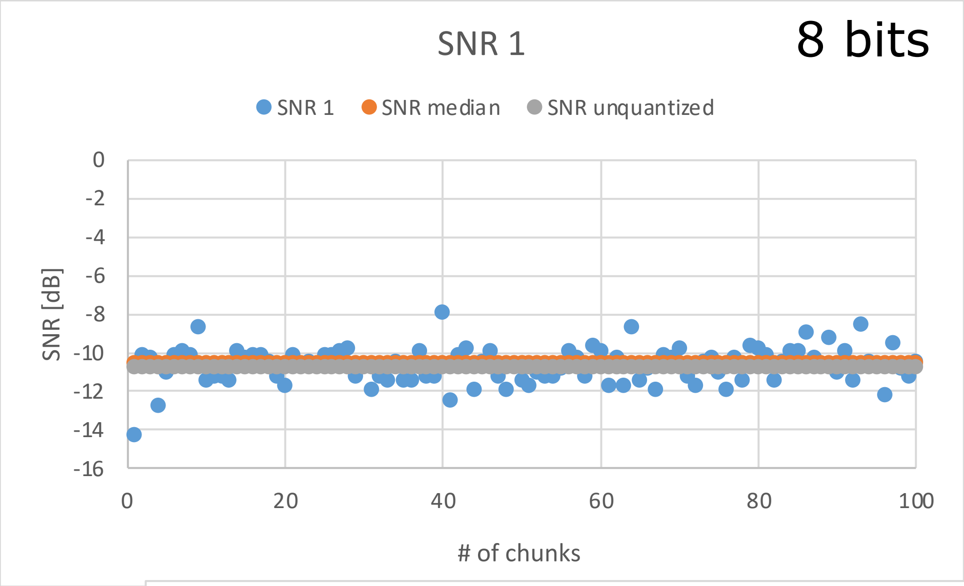
- 100 points are taken and the median computed.

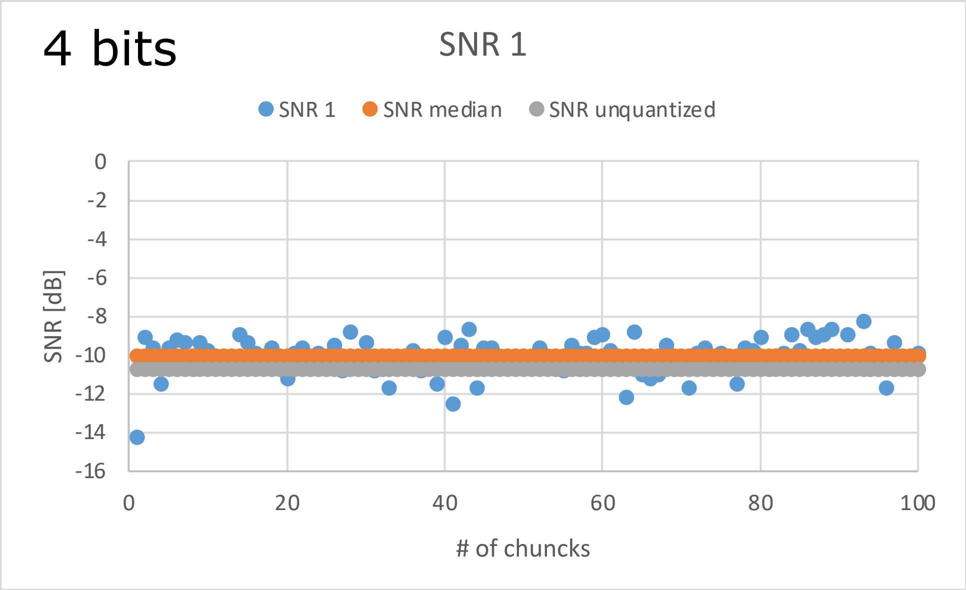
For the subsequent computations:

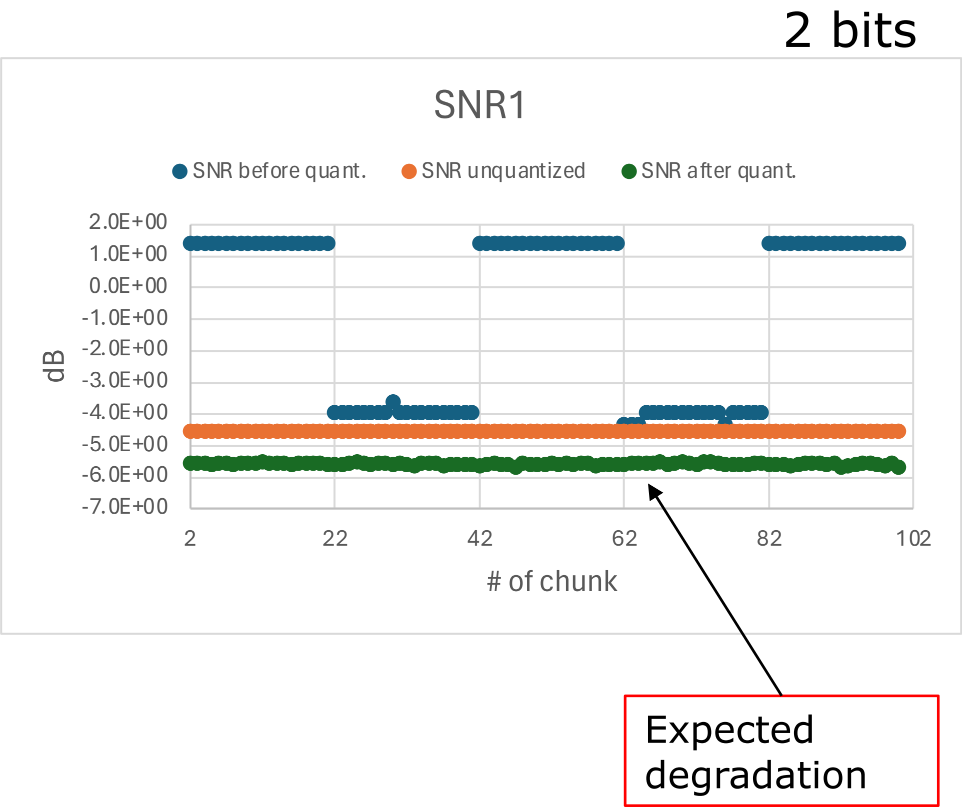
- Search range from SNR/2 to SNR*2, with SNR found at the previous step.

Assuming AWGN, from the cross-correlation of the combined signal we derive the SNR of the combined signal and the gain of the array.

Correlator/Combiner FSC Standard: weights computation (3)







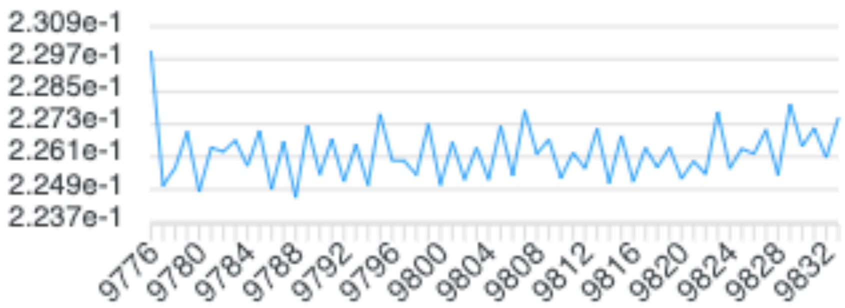
Setup	
S/No	65.71 dBHz
Sampling rate	43.75 MHz
Symbol rate	40 Msymb/s

CORRELATOR/COMBINER FSC STANDARD: RESULTS HIGH RATE

Symbol rate	Info rate	Sampling rate	Coding	Simulated Es/N0
40 Msymbols/s	20 Mb/s	43.75 MHz	TC 1/2 k=8920	-4.17 dB

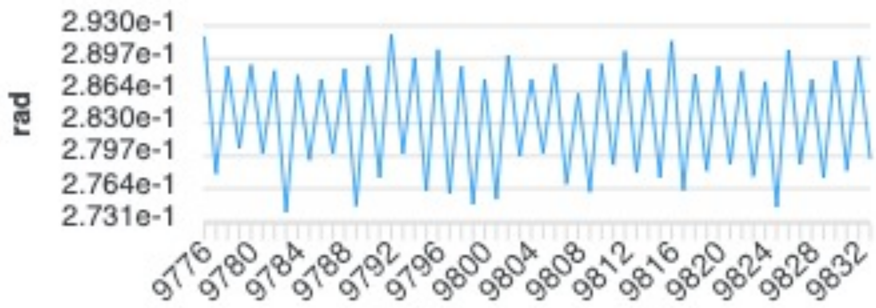
High corr. SNR → 50 ms corr. time

Peak of the cross-correlation function



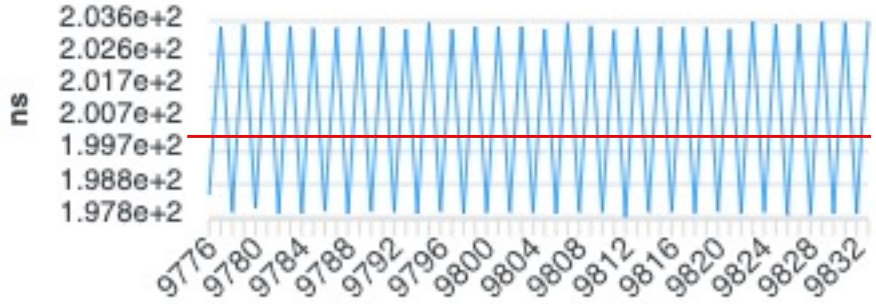
Time [Seconds past midnight]

Residual Phase from cross-correlation



Time [Seconds past midnight]

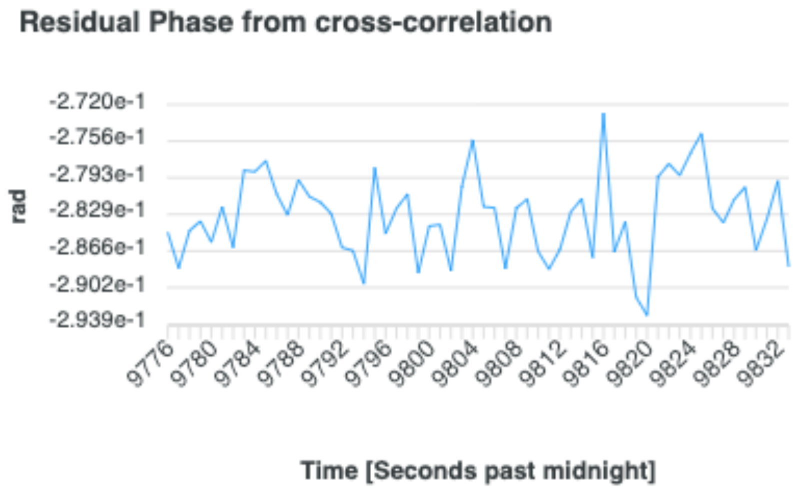
Residual Delay from cross-correlation

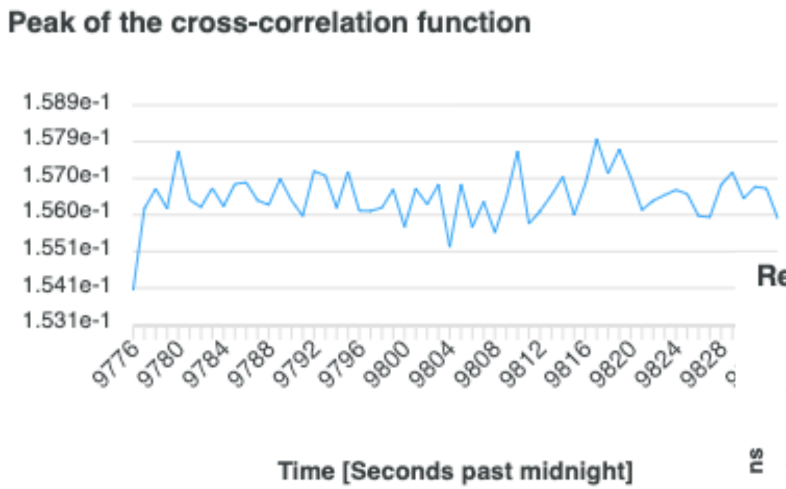


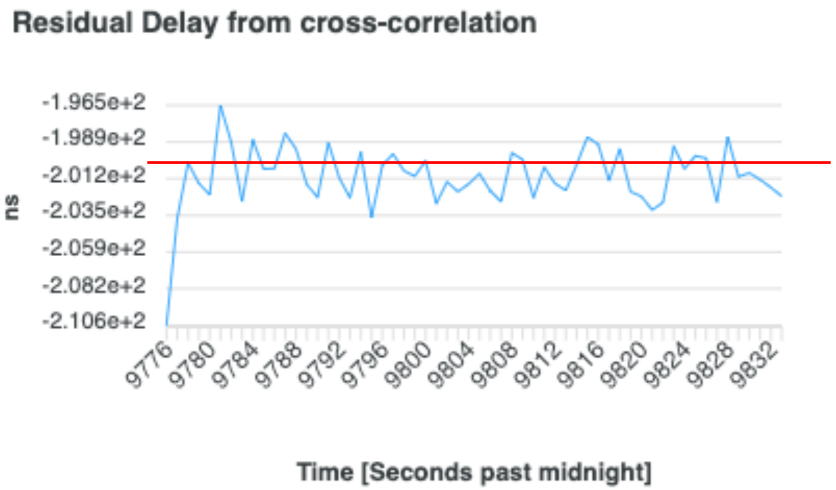
Time [Seconds past midnight]

Correlator/Combiner FSC Standard: Results low rate

Symbol rate	Info rate	Sampling rate	Coding	Simulated Es/N0
2 Msymbols/s	1 Mb/s	4 MHz	TC 1/2 k=8920	-4.17 dB



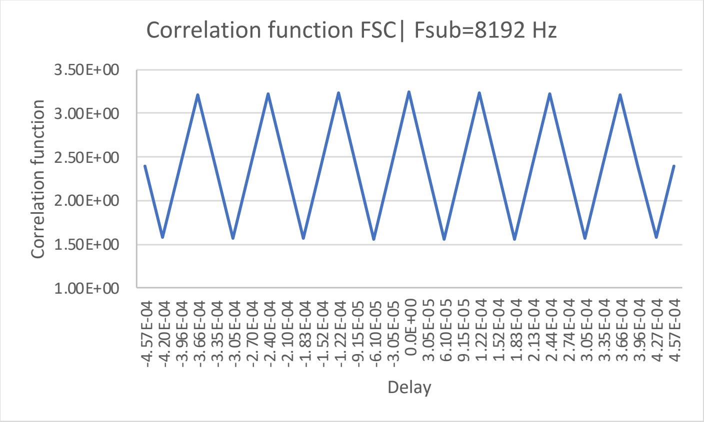




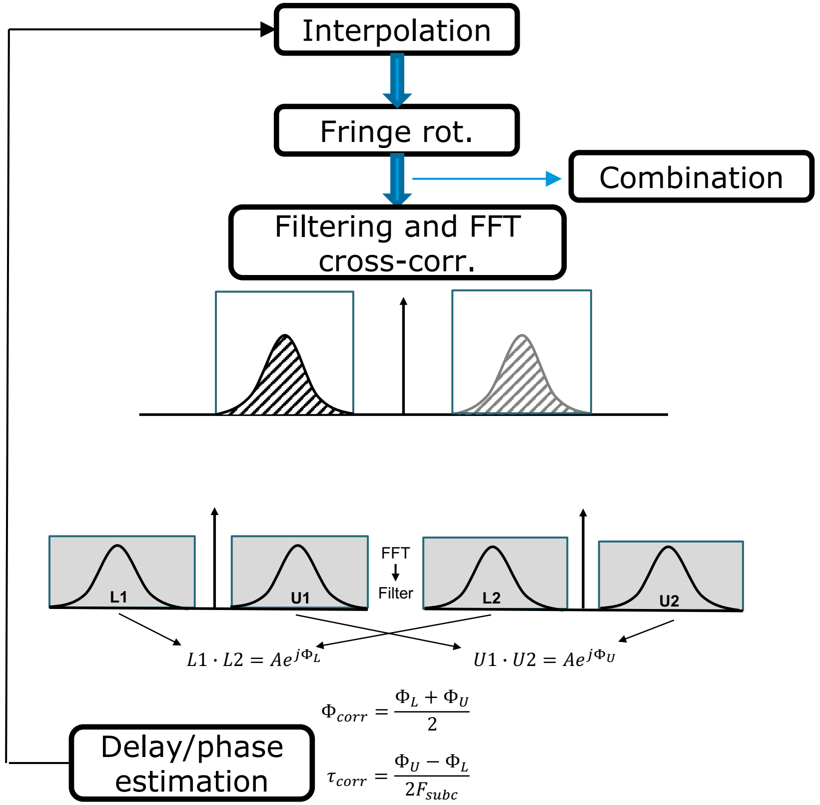
Measured combination loss from weights computation < 0.1 dB.

CORRELATOR/COMBINER FSC FILTERED: ALGORITHM

FSC filtered: residual delay/phase computed with cross-correlation of the filtered FFT of the signals. Used for remnant carrier modulation schemes.



Period: $1/(F_c - F_{sub})$
Critical for low-rate scenarios



Period: $1/2(F_c - F_{sub})$

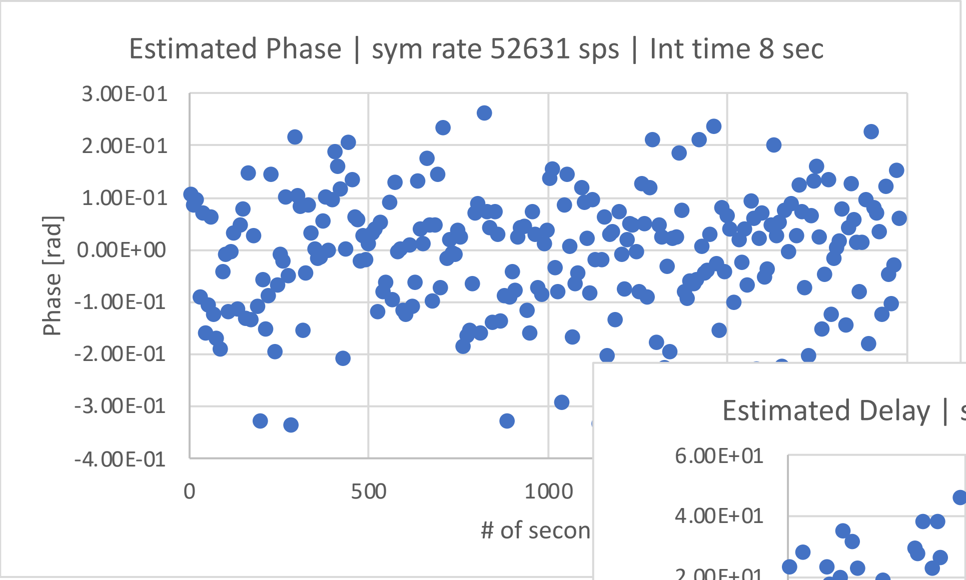
Monograph 5 DESCANSO
Antenna Arraying Techniques on the DSN

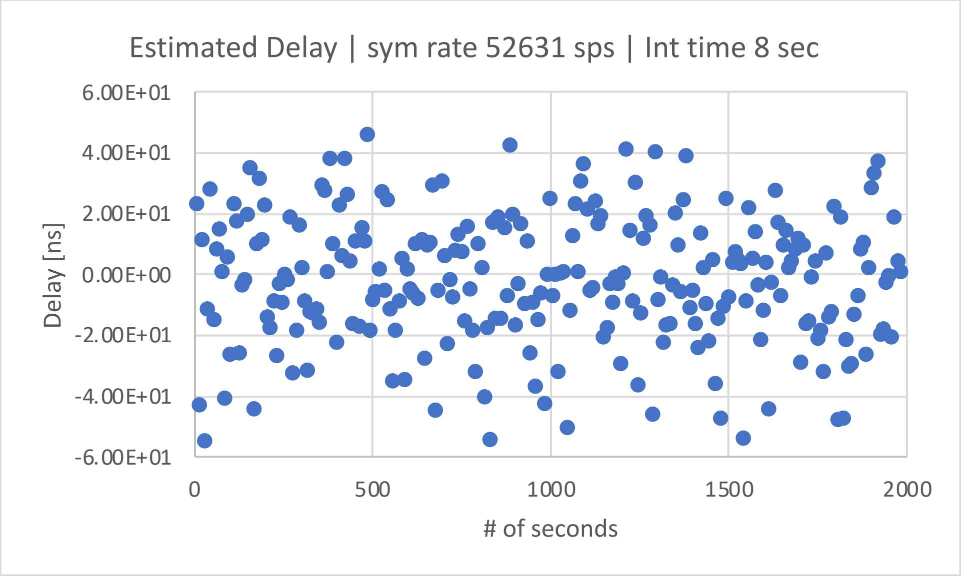
The algorithm acts on the sidebands
→ **For low rates large integration time**.

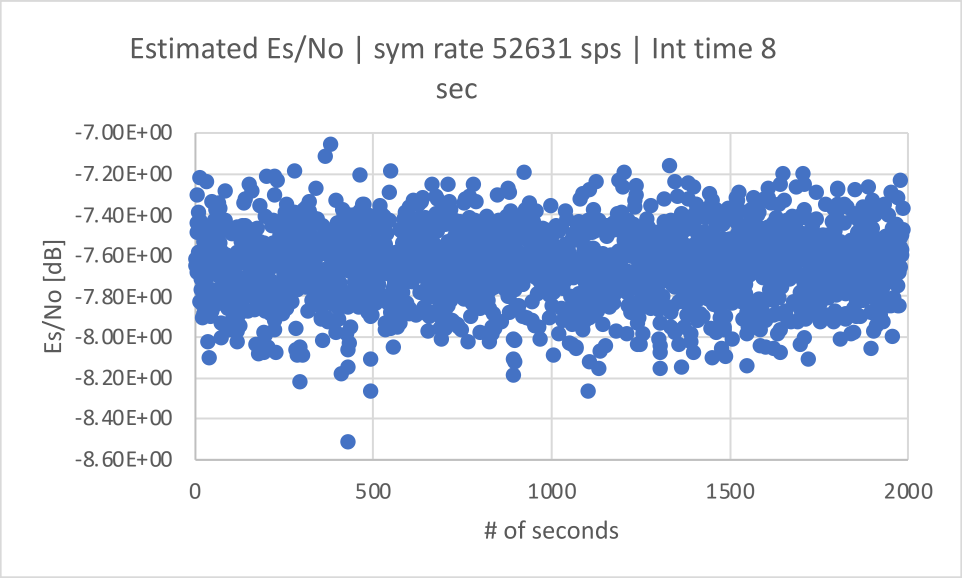
Less power (carrier power lost) but smaller correlation bandwidth → SNR_{corr} increases

Correlator/Combiner FSC filtered: Results

Symbol rate	Info rate	Sampling rate	Subcarrier Freq.	Coding	Simulated Es/N0
52 ksymbols/s	8736.67 b/s	4 MHz	263 kHz	TC 1/6 k=8920	-10.3 dB







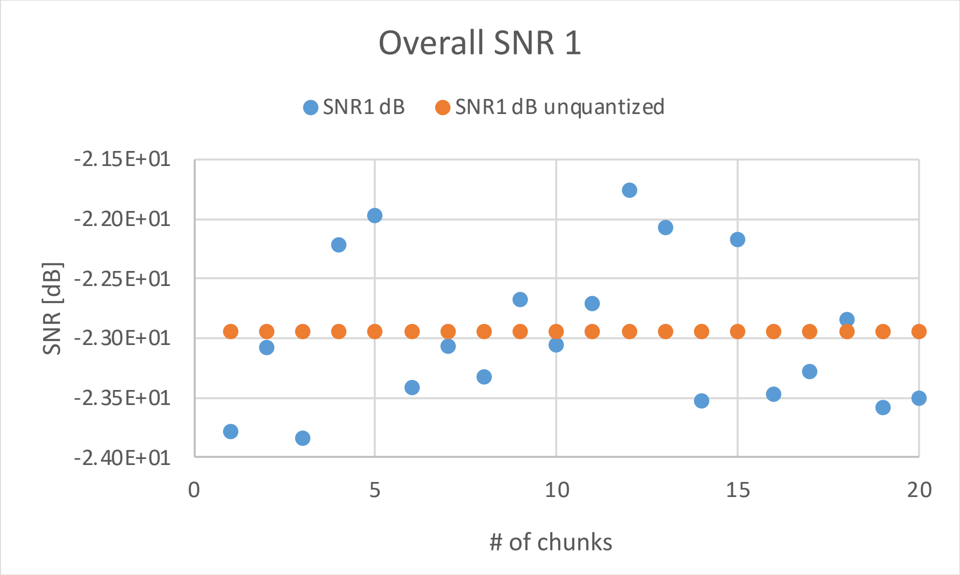
Coding	Loss (measure from demodulator)
TC 1/2	0.05 dB
TC 1/4	0.1 dB
TC 1/6	0.2 dB

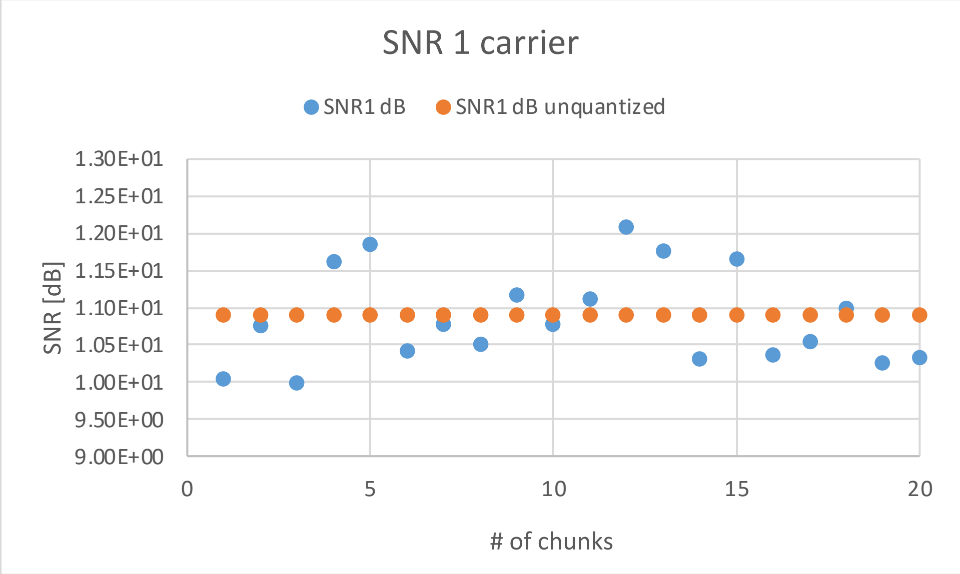
Correlator/Combiner FSC Filtered: weights computation

In some scenarios the SNR can be < -35 dB. Filtering about the carrier allows to estimate the SNR (and the weight) of the carrier with a narrow bandwidth.

$$W = \frac{A}{\sigma^2}$$

Setup	
S/No	37.06 dBHz
Subcarrier Freq.	263 kHz
Symbol rate	52 ksps
Filter BW	60 points



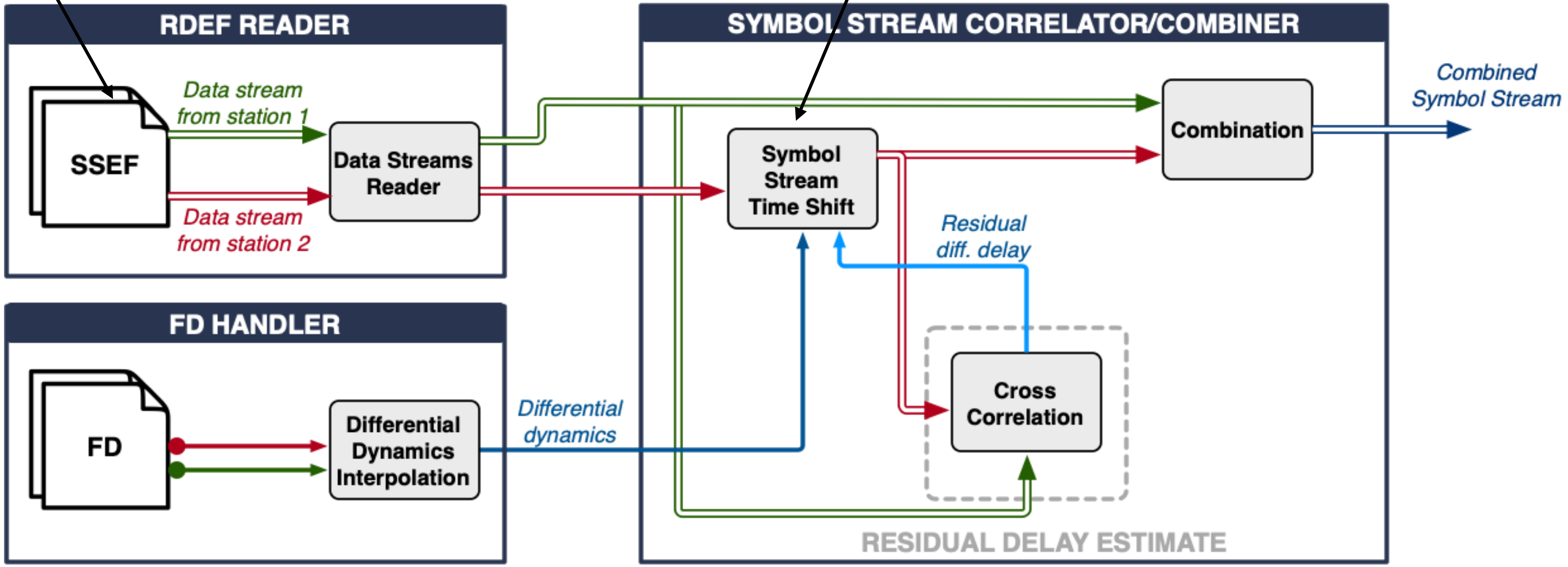


Used for PCM/BPSK/PM.
For PCM/SP-L same algorithm as for GMSK.

SSC: residual delay computed with cross-correlation on symbols.

SSEF based on Vita 49 standard

Only symbol shift, no time interpolation or fringe rotation



Cross-correlation may be frequent to detect slip bits and/or unlock of the demodulator
Cross-correlation must solve for carrier ambiguity
Handling different number of bits per chunk

Correlator/Combiner SSC: weights computation

The quantized matched filter output $y_i^{(1)}$ and $y_i^{(2)}$ are combined using a weighted sum.

- Weights are proportional to Es/N0 and normalized such that $w_1 + w_1 = 1$.

$$\text{Quantization Factor: } q_f = 4\left(\frac{Es}{N0}\right) * LLR_{scale}$$

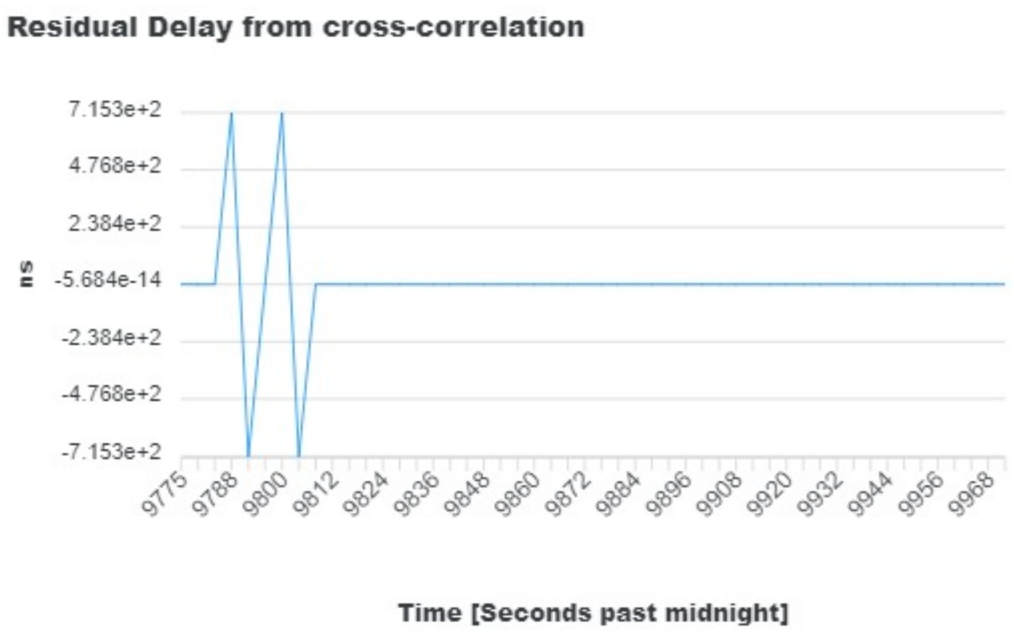
$$LLR(y_i) = 4\left(\frac{Es}{N0}\right) MF(y_i) = 4\left(\frac{Es}{N0}\right) * LLR_{scale} * y_i$$

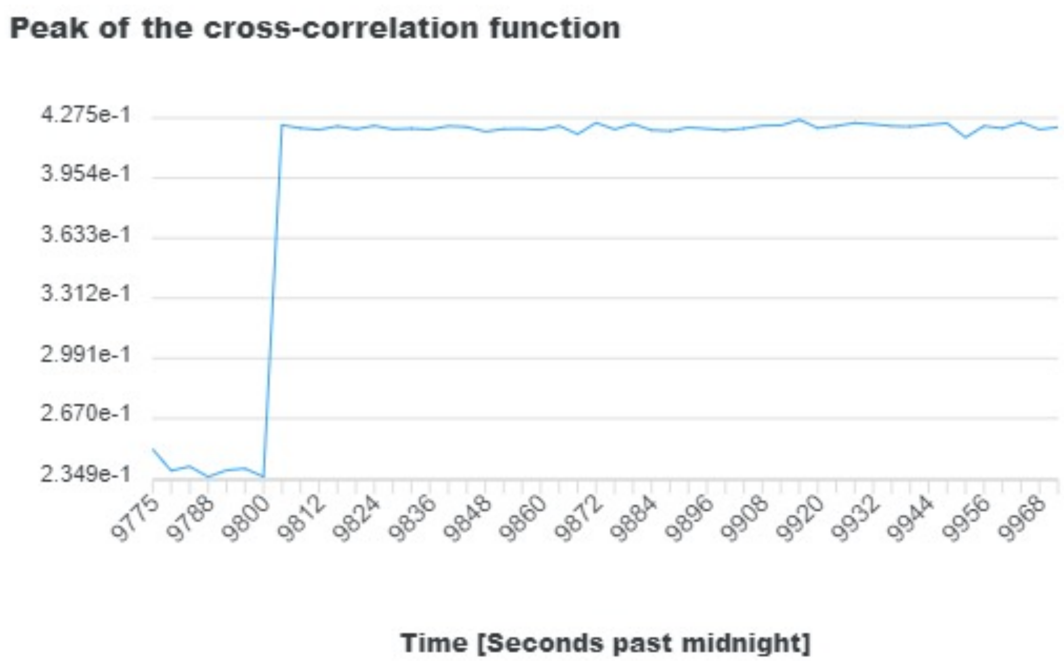
$$Combined\ MF = \frac{4\left(\frac{Es}{N0}\right)^{(1)} * LLR_{scale}^{(1)} * y_i^{(1)} + 4\left(\frac{Es}{N0}\right)^{(2)} * LLR_{scale}^{(2)} * y_i^{(2)}}{4\left(\frac{Es}{N0}\right)^{(1)} + 4\left(\frac{Es}{N0}\right)^{(2)}}$$

- The $\frac{Es}{N0}$ of the combined stream is computed with the same formulation used in the demodulator.
- The combined MF output is quantized, and the new quantization factor is passed to the decoder.

CORRELATOR/COMBINER SSC: RESULTS (1)

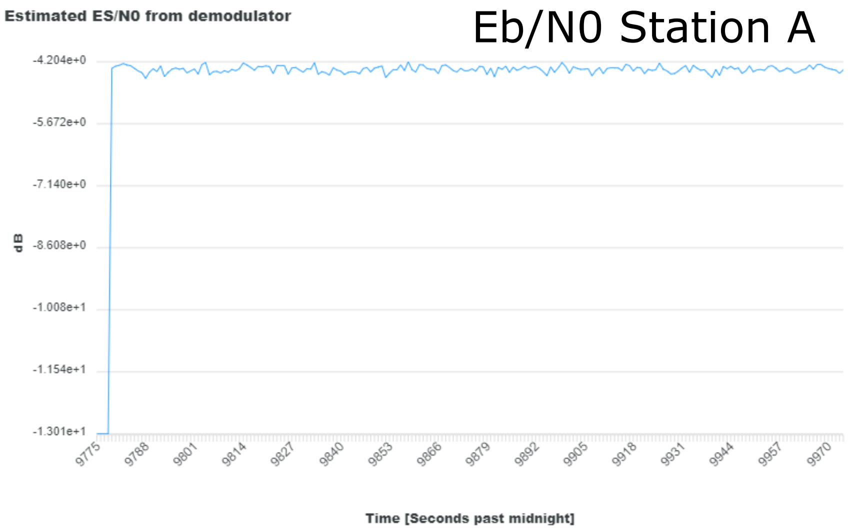
Symbol rate	Info rate	Sampling rate	Coding	Modulation index	Simulated Es/N0	Acquisition BW	Tracking BW
1398.10 ksymbols/s	699.05 kb/s	16 MHz	TC 1/2 k=8920	1 rad	-4.17 dB	100 Hz	25 Hz



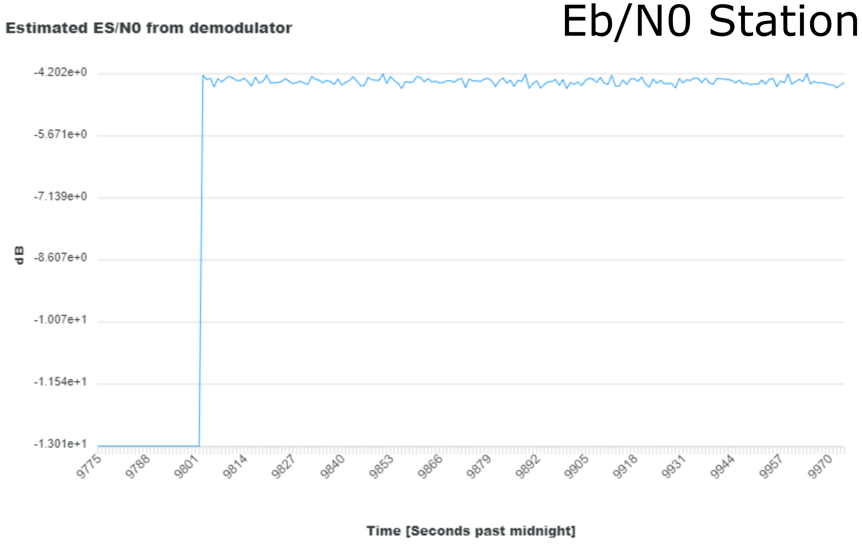


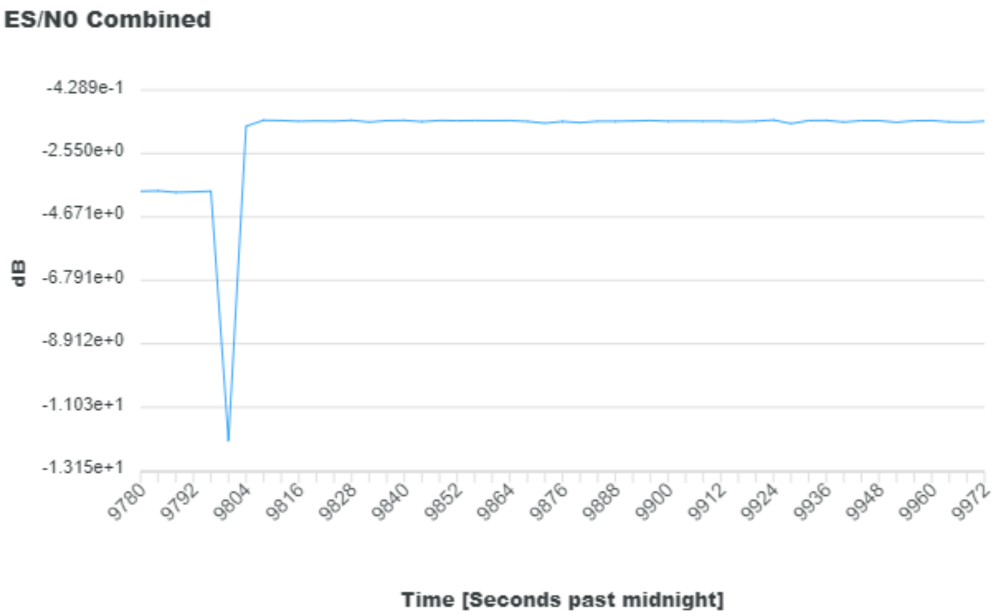
CORRELATOR/COMBINER SSC: RESULTS (2)

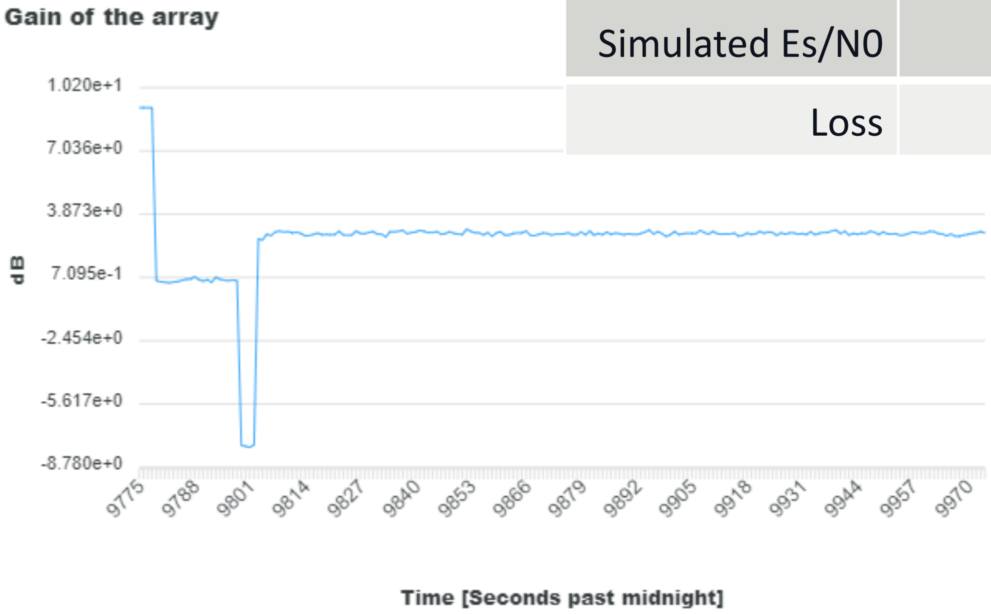
Eb/N0 Station A



Eb/N0 Station B







Simulated Es/N0	-1.17 dB
Loss	0.1 dB

Arpsoft® Zelinda

Correlator/Combiner: technology (I)

In WP1000 a trade-off between CPU and/or GPU (general purpose GPU, GPGPU) has been carried out.

Modern CPUs are characterized by having 16 to 64 cores working at around 5 GHz (turbo). The GPUs, on the other hand, are characterized by a much larger number of cores, nowadays greater than 10000, working at around 1.5 GHz. The usage of GPUs, however, entails a time overhead needed to transfer the data between the host (CPU RAM), where the data must be used by the overall tool, and the device (GPU DRAM), where the data must be located for GPU cores processing task. Moreover, an additional overhead is given by the device initialization.

Desktop Machine used for test	
CPU	AMD Ryzen 5950X 3.4 GHz 16 Core
MB	AMD X570
RAM	64 GB DDR4 3200 MHz
Storage	SSD M2.1 1TB Samsung Evo Pro
GPU	Nvidia GTX 3080 Ti 12 GB
Market price	3 kEuro

	GPU (c++ and CUDA)	**CPU (c++ and openMP)**
GPU Initialization	*120 ms*	-
HOST/DEVICE Memory transfer	60 ms	-
Time Interpolation, Fringe rotation, Combination	110 ms	300 ms
Cross-correlation, auto-correlation, weights update (on 30 ms)	20 ms (CPU)	20 ms
Total Time	190 ms [initialization time is not considered]	320 ms

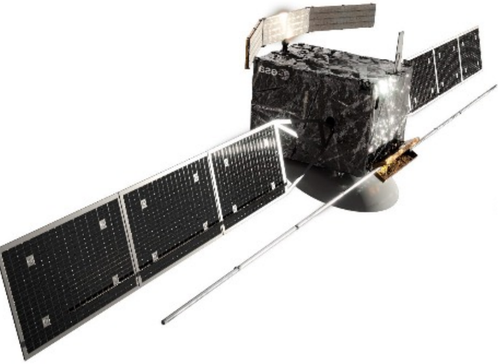
Correlator/Combiner: technology (2)

Trade-off CPU/GPU

	Pros	Cons
CPU	• No additional dedicated HW • Portability	• Machine exclusively dedicated to it • Higher processing time
GPU	• CPU available for other computations or ancillary services • Lower processing time	• Constraint of single manufacturer product (Nvidia) • More complex to code

For the OFF-A primary objective (2 stations, off-line) both CPU and GPU approaches provide adequate results.

Real time applications for ESA at the moment are for 2 stations. Therefore, CPU approach was chosen for the tool.



OFF-A core services: demodulator

Near real time software demodulator

- 3 Modulation schemes: GMSK BT=0.5; PCM/BPSK/PM; SPL/PM
- GMSK Symbol rates upto ~40 Msps – subject to technology dependent sampling rate maximum
- Low loss
- Operation down to Es/No = -12 dB (turbo 1/6 with external soft decision combination, SSC)
- Flexible input format (8 bit or 2 bit near baseband I&Q)
- Soft-decision outputs (LLRs) of variable number of bits.
- Modest Doppler uncertainty.

SOFTWARE DEMODULATOR - GMSK BANDWIDTHS

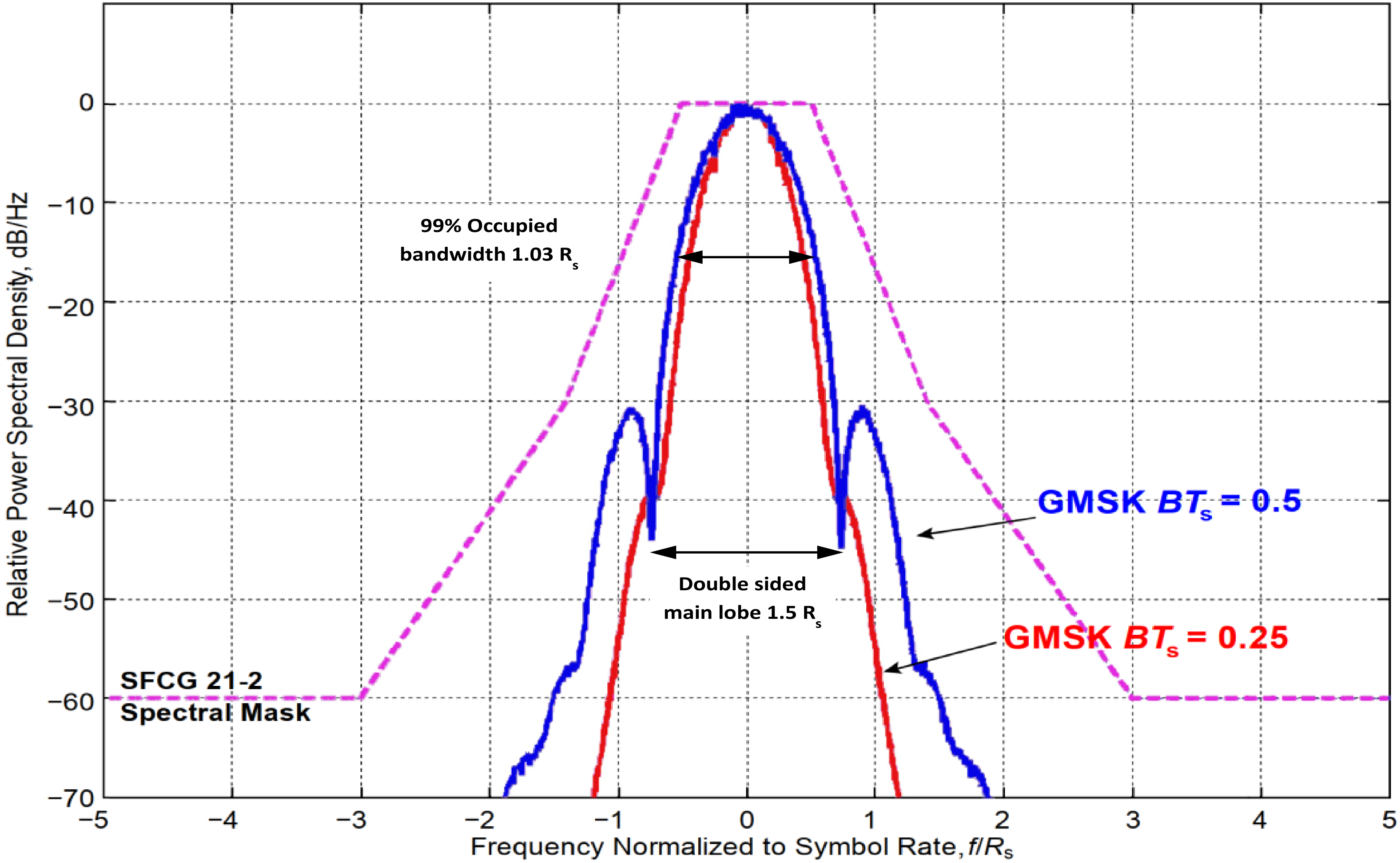
Note to transmit R_{es} = 40 M encoded bits per second :

- (unfiltered) OQPSK would have a double sided main lobe bandwidth of Rs (noting 2 bits are sent per channel symbol) so would be **40 MHz** (but much wider 99% bandwidth);
- SRRC filtered OQPSK with α=0.5 has main of $\frac{1}{2}(1+\alpha)R_s = 0.75\ R_s$, or **30 MHz** (but not constant envelope).
- GMSK main lobe is 1.5 Rs = **60 MHz**, twice the width of SRRC OQPSK, despite the modulations being quite similar (in the sense that GMSK can be demodulated with an OQPSK demodulator) – but is constant envelope.

Software Demodulator - GMSK Bandwidths 2

Sampling rate

A trade off between losses (aliasing and signal loss) and computational load

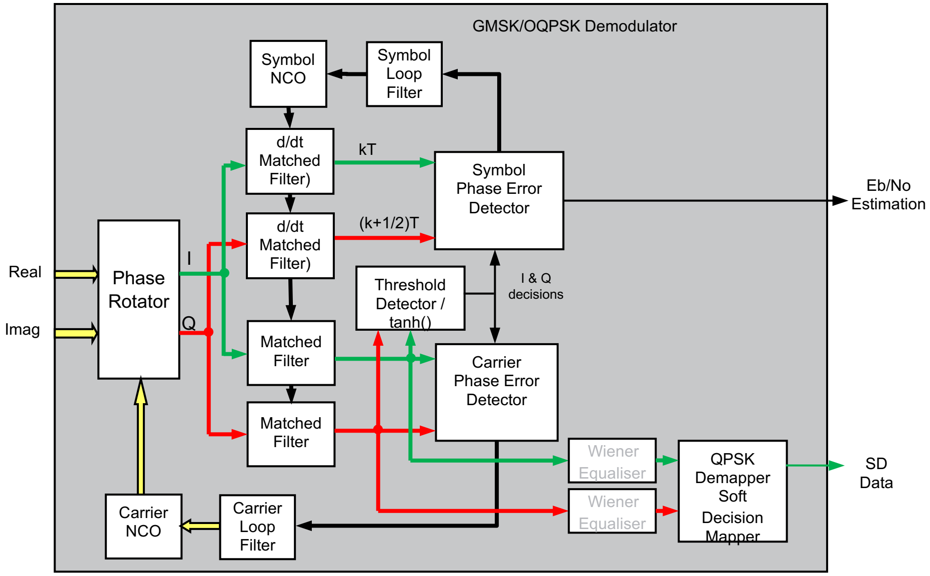


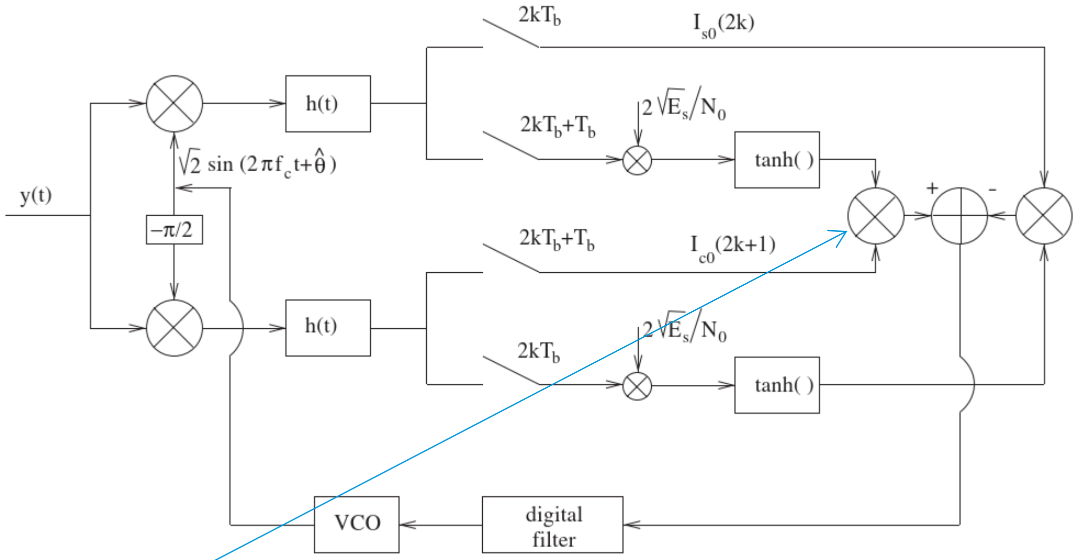
CCSDS 413.0-G-3

Stand alone Software Test Modulator - GMSK

- Abandoned MatLab
  - Very messy to have non integer arbitrary ratio between sampling rate and symbol rate (interpolators and decimators needed)
  - Could get implementation loss below 0.7 dB (MatLab Modulator -> MatLab Demodulator).
- Wrote true phase modulator in C++ using LUT (without interpolation) for GMSK symbol phase
- AWGN Noise generator (using library Mersenne Twister random number generator and the Box–Muller transform)
- Quantisation and clipping of the floating point samples into signed integers of required number of bits.

Software Demodulator - GMSK





- Carrier phase and symbol timing synchronization use maximum likelihood (ML) estimation algorithm considering only the first Laurent pulse (one-Amp).

  Approximation for tanh(x) yields Carrier and Timing phase errors :

  Low SNR approximation (non data-aided):

  $$e_P(k) = I(k)Q(k) - Q(k+1/2)I(k+1/2)$$

  $$e_T(k) = I(k)I'(k) + Q(k+1/2)Q'(k+1/2)$$ (using Diferential Matched filters, $I'(k)$ and $Q'(k+1/2)$)

  High SNR approximation (decision directed)

  $$e_P(k) = \hat{a}_k Q(k) - \hat{b}_k I(k+1/2)$$

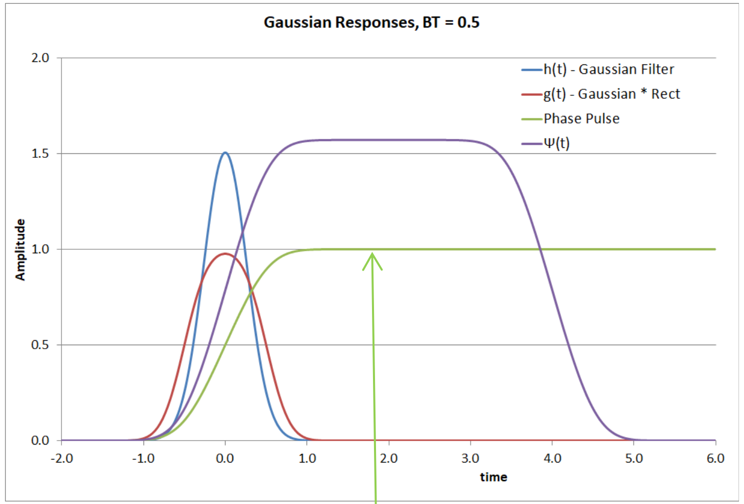
  $$e_T(k) = \hat{a}_k I'(k) + \hat{b}_k Q'(k+1/2)$$

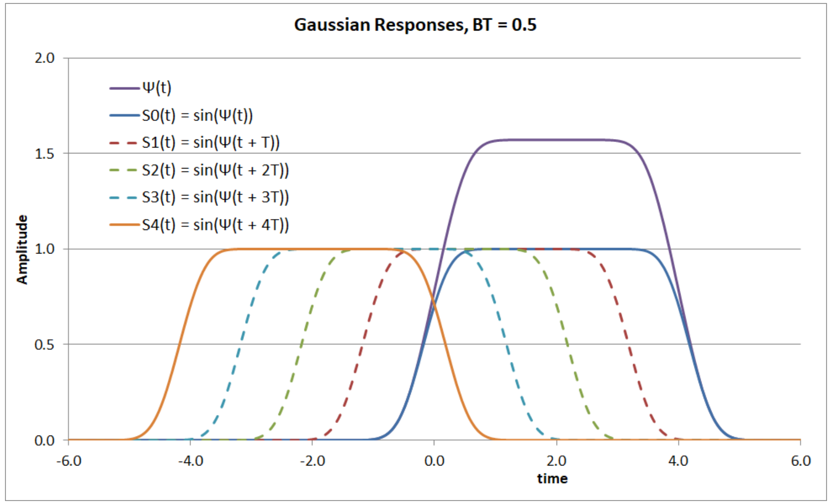
  Note that this is effectively treating each channel (I or Q) as independent BPSK and then adding the phase errors (OQPSK has better carrier phase estimate than QPSK)

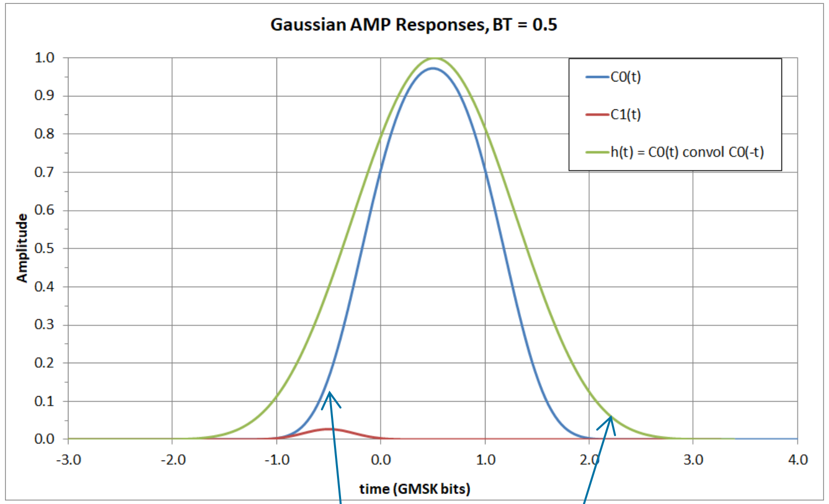
- E_s/N_0, uses ratio of 2nd and 4th raw moments to estimate $\theta = \frac{\mu}{\sigma} = \sqrt{2\, E_s/N_0}$, $\hat{\theta} = \sqrt{\frac{-(2\hat{C}-6)\pm\sqrt{(2\hat{C}-6)^2-4(\hat{C}-1)(\hat{C}-3)}}{2(\hat{C}-1)}}$, Where $\hat{C} = \frac{\hat{\mu}_4'}{\hat{\mu}_2'^2}$ $\hat{\mu}_2' = \frac{1}{N_{sym}}\sum_{i=1}^{N_{sym}} y_{m_i}^2$

GMSK – Phase Pulse, Laurent Pulse, Matched and Differential Filters

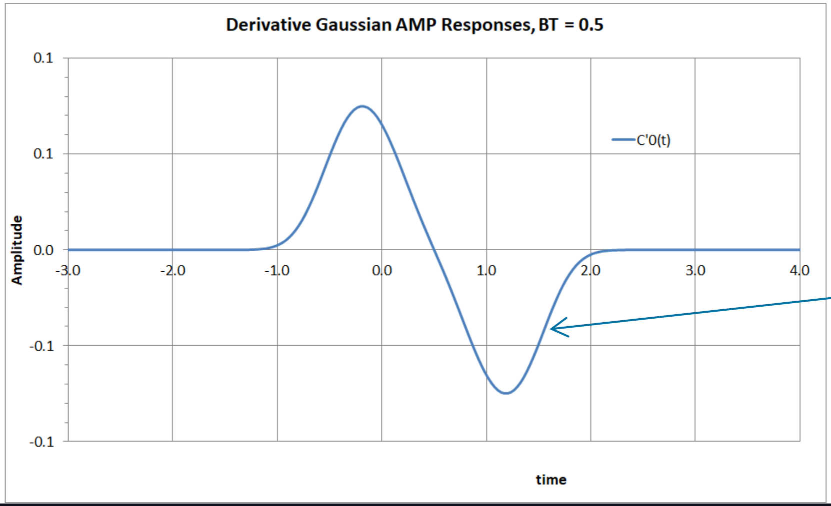
- Excel used to generate a LUT 4 symbols long of the **symbol phase** at 64 samples per GMSK symbol for Test Modulator
- and used to generate Laurent Pulse for Demodulator Matched filters + Derivative Matched filters







Test Modulator Phase Pulse



Demodulator Matched Filter response

Used in Phase Detector Gain Calculation

Demodulator Differential Matched Filter

Software Demodulator – Software Implementation

Coded in GNU 17 C++ with Code::Blocks IDE

Vectorisation using SIMD with Agner Fog libraries which provide:

- Vector functions without assembler instructions.
- Very fast (vector) math functions using unusual algorithms (eg cos & sin).
- Almost all library code in "in-line" (without function calls).
- Source code provided and modifiable (so can change speed vs accuracy trade-off).

Demodulator Implementation:

- All vectors 8 elements (256 bits, AVX-2) to allow use on both **intel and AMD targets**.
- AVX-512 not used as Intel devices (unpredictably) throttle to prevent overheating and not supported by AMD.
- Predominantly 32 bit floating point vectors, integer vectors for addresses (eg in Laurent AMP LUT)
- Matched Filter & Derivative Matched filter are the most complicated:
  - Processes vector of 8 complex samples together with vector of 8 symbol phases in single instructions,
  - Performs 11 correlations (8 samples + L=3)
  - Detects symbol boundaries
  - Outputs up to 4 soft decisions per channel (8 soft decisions per 8 samples)
- All NCOs have "Doppler predictor" which can be programmed with the Doppler rate for Doppler prediction.
- All loop automatically narrowed from Acquisition Bandwidth to Tracing bandwidth in geometric steps of $\sqrt[4]{2}$

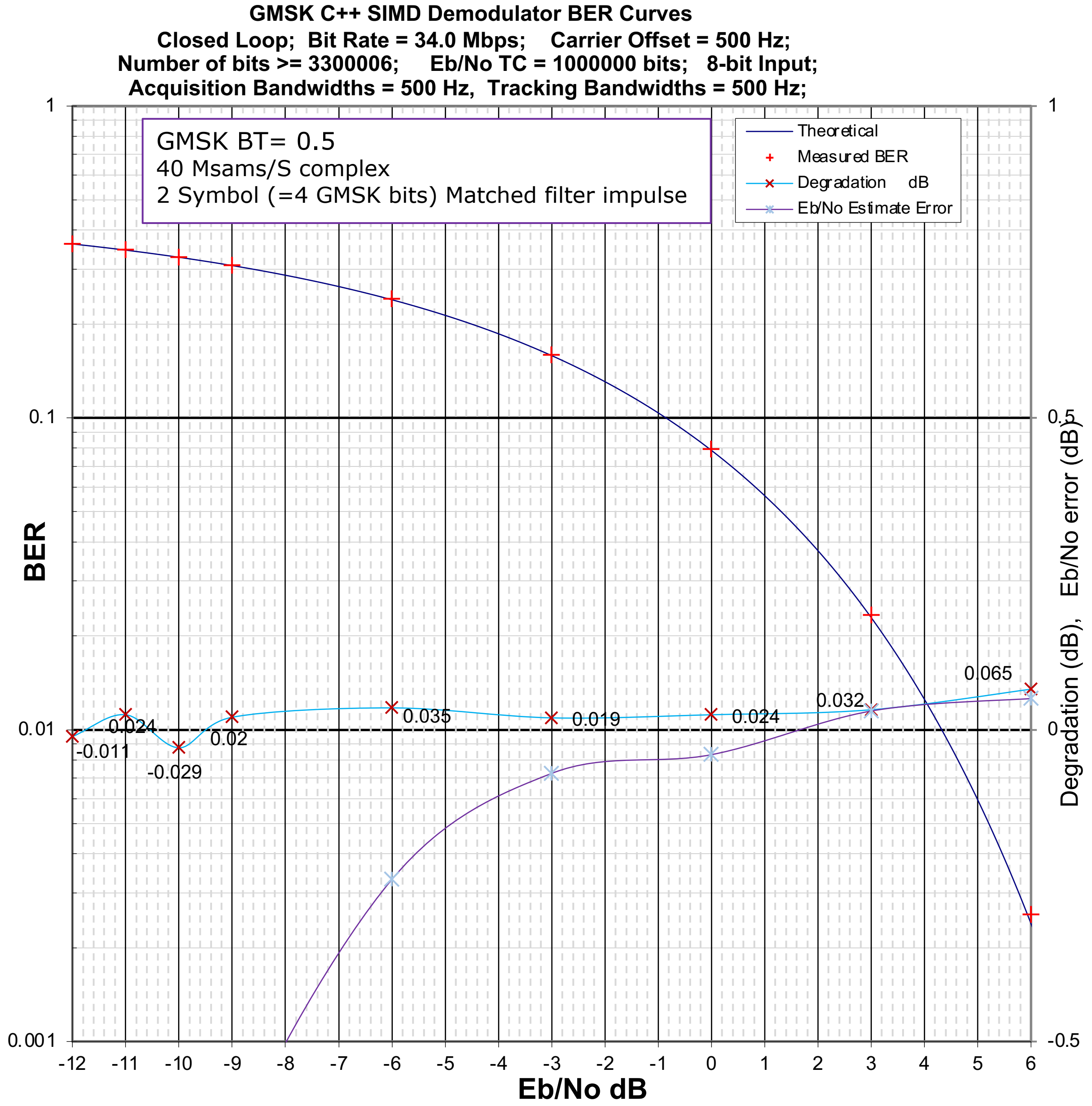
Design uses a single thread on a single core (No parallel processing at Demodulator level)

Software Demodulator – Vector routines performance

Downconversion Benchmarks

Processor **Date** **SIMD**	**Intel(R) Xeon(R) CPU E5-1620 0 @ 3.60GHz** **Launched Q1 2012** **Intel AVX**		
OS	Windows 10 Pro		
Compiler	g++ (x86_64-posix-seh-rev0, Built by MinGW-W64 project) 8.1.0		
Vector size	8 floats		
Data Block size	8k complex samples		
Operation	Coding syle	Execution Rate M Complex Samples/Second	Note
Complex Multiply	non-vector	643	
	Vector	1767	
Downconvert	non-vector	13.3	With GNU C++ 17 sin and cosine computation (float)
	Vector	317	with vector sin() & cos()
	Vector	316	with function "sincos()" that generates sin and cos simultaneously
	Vector	376	with function "fastsincos()" that operates over a reduced angular range (of a few hundred revolutions)
	Vector using Cordic	317	8 stages of Cordic (~1/2 degree accuracy)

Demodulator Testing – GMSK Closed Loop 8 Bit Quantised



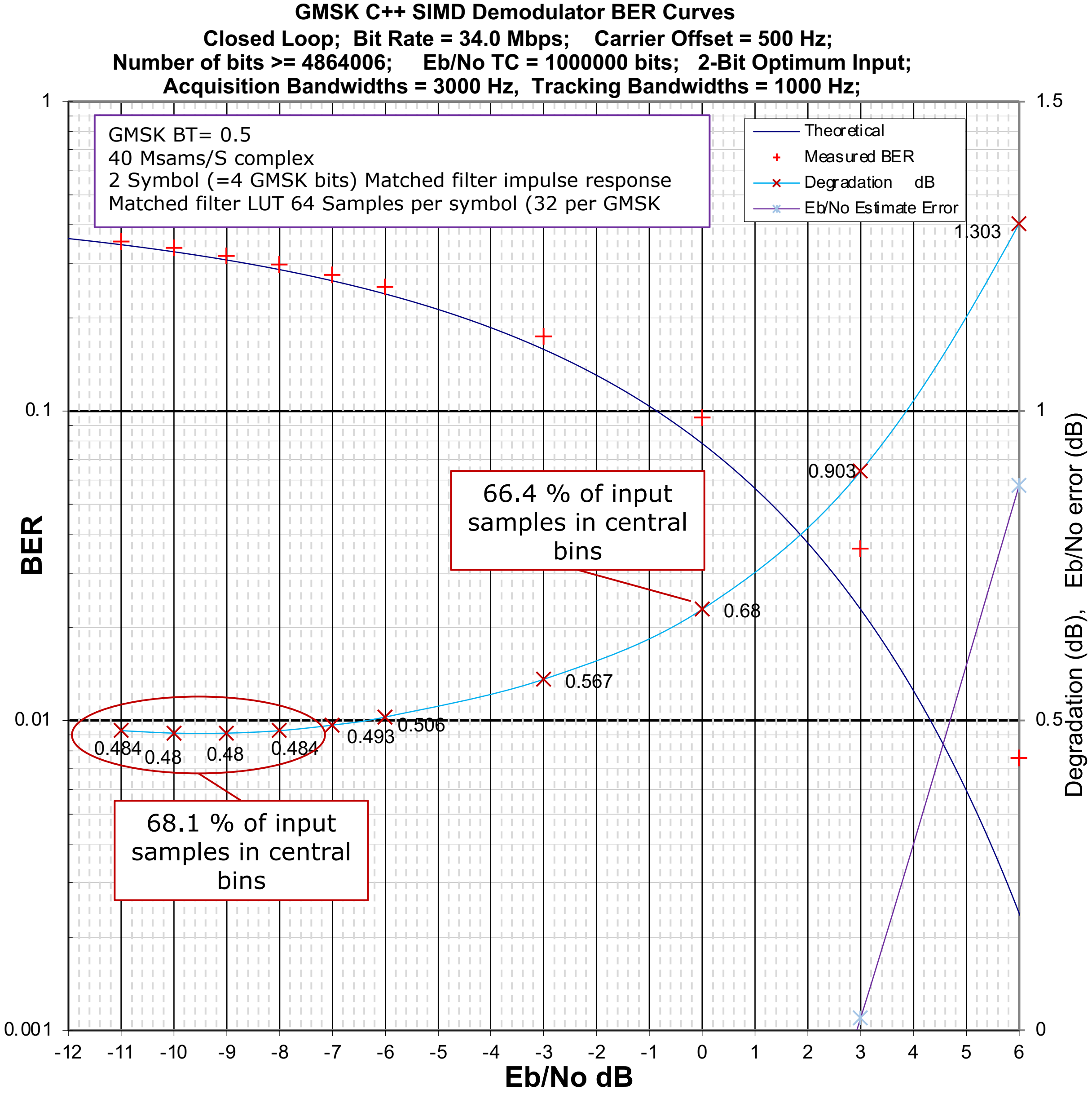
- Processed in Chunks of 40,000 complex samples
- Timing Acquisition from ½ bit error
- Carrier offset upto 3 kHz
- Acquisition bandwidths 500 Hz (Damping factor = 0.8)
- Tracking bandwidths 500 Hz

- BER degradation < 0.07 dB, <0.03 dB at Es/No of interest
- Eb/No estimate needs more simulation time

- Processing rate 17.1 complex MSamples/S on Xeon E5-16200 @ 3.60 GHz, Single Thread Rating : 1774
  vs AMD Ryzen 9 5950X @3.4 GHz Single Thread Rating: 3470
  vs AMD Ryzen 9 7950X3D @4.2 HGz Single Thread Rating: 4203
  - Sample generation ~ 8 hours/sec
  - Demodulator I/O and analysis ~4 hours/sec (Slow BER)
  - Demodulation ~2 secs/sec

Equals or exceeds real time rate on fast processors

Demodulator Testing – GMSK Closed Loop 2 Bit Optimum Quantised



- Reconstruction levels of ±0.4528, ±1.5100
- Processed in Chunks of 40,000 complex samples
- Timing Acquisition from ½ bit error
- Carrier offset upto 3 kHz
- Acquisition bandwidths 3000 Hz (Damping factor = 0.8)
- Tracking bandwidths 1000 Hz

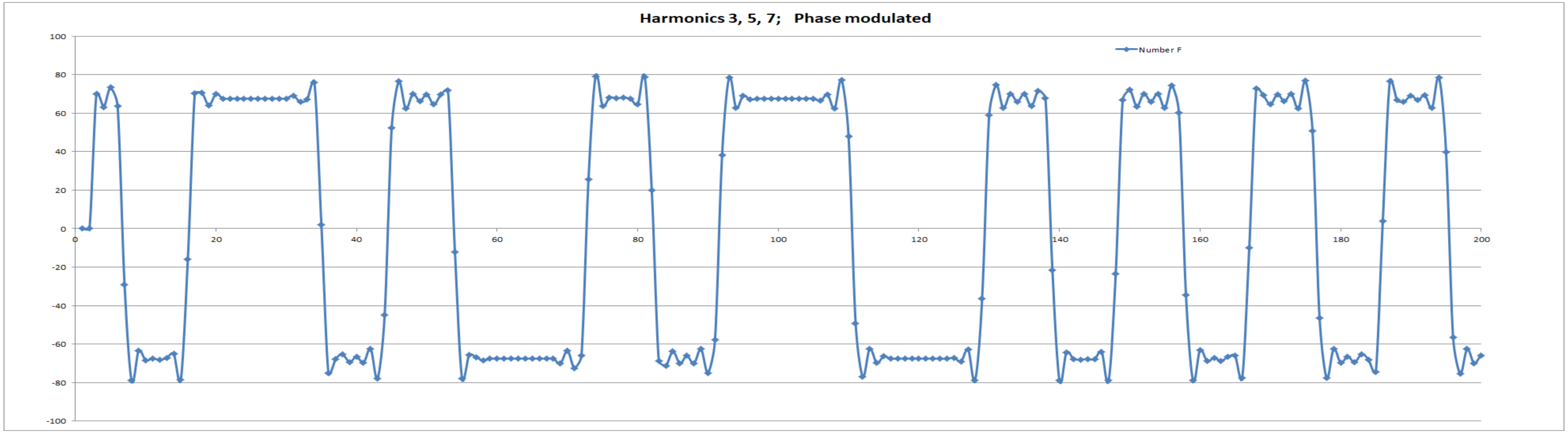
- BER degradation < 0.7 dB, <0.03 dB at Es/No of interest
- Eb/No estimate needs more simulation time

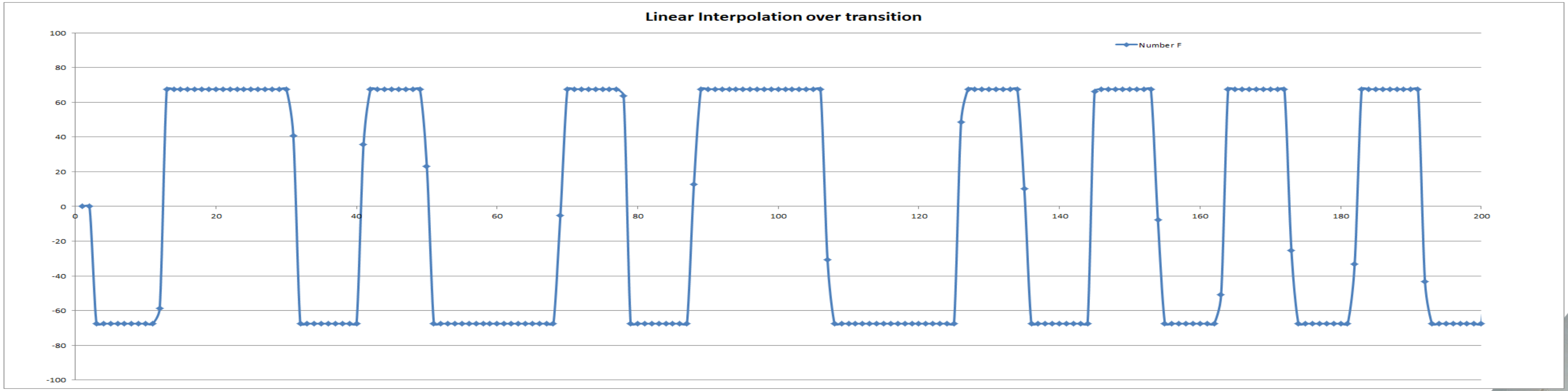
- **Degradation ~0.5 dB worse than 8 bit at low SNR** (in line with 1960 paper by Joel Max)

Stand alone Software Test Modulator - PCM/BPSK/PM & SPL I

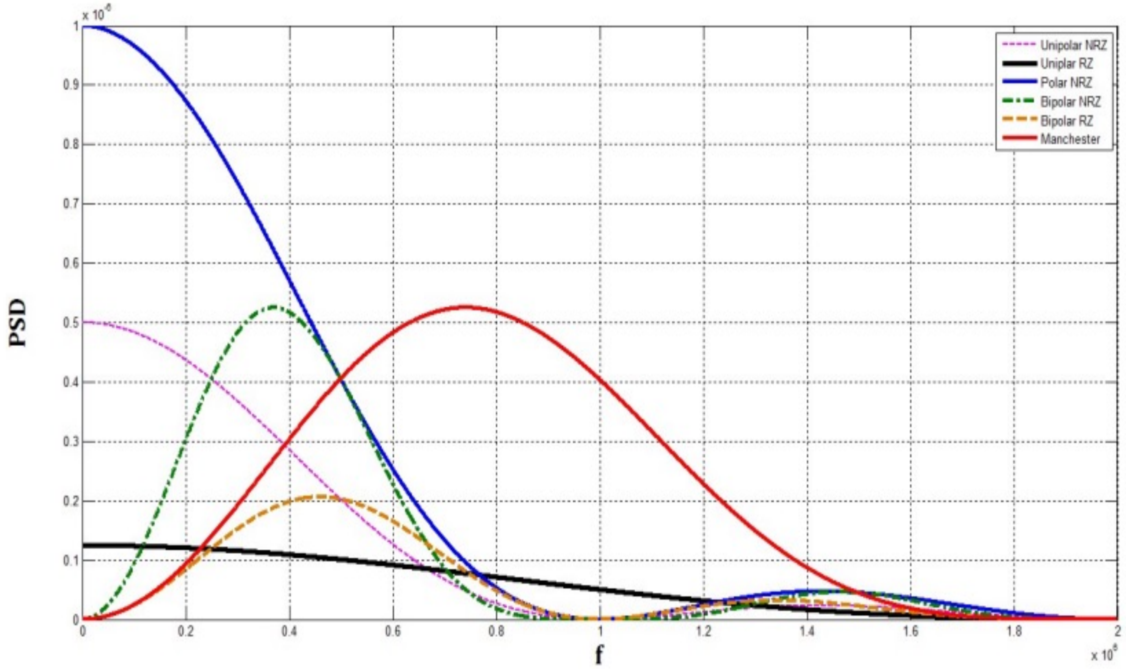
- Sine or "Square" subcarrier
- True square subcarrier has infinite bandwidth so samples of it would contain aliased harmonics, so Modulator (and demodulator) must synthesise a band-limited "square" wave , implemented several options :
  - "Square" wave synthesized "on the fly" by adding only (user defined) non-aliasing harmonics : "$\sin(\omega t) + 1/3 \sin(3\omega t) + 1/5 \sin(5\omega t)\ldots$"
  - If >= 9 harmonics wanted, instead only subcarrier transitions are filtered with a linear interpolation
  - If data modulation "removes" a subcarrier transition, synthesised subcarrier replaced with constant amplitude over the transition
    - "square" wave can be phase modulated on to carrrier which gives $Q = \sin(\sin(\omega t) + 1/3 \sin(3\omega t) + 1/5 \sin(5\omega t)\ldots)$
      - which at high modulation index re-introduces aliased harmonics - Bessel
    - Or "square" wave directly used as Q channel and I synthesized to $I = \sqrt{1 - Q^2}$ (brickwall bandlimited)
- SPL uses same Test Modulator with 1 subcarrier cycle per bit.
- PCM/BPSK/PM Demodulator has the same options for harmonics in local subcarrier generation.
- SPL Demod use traditional approach using samples of a square wave (with aliases)

Stand alone Software Test Modulator - PCM/BPSK/PM & SPL 2





Demodulator Testing – SPL/PM

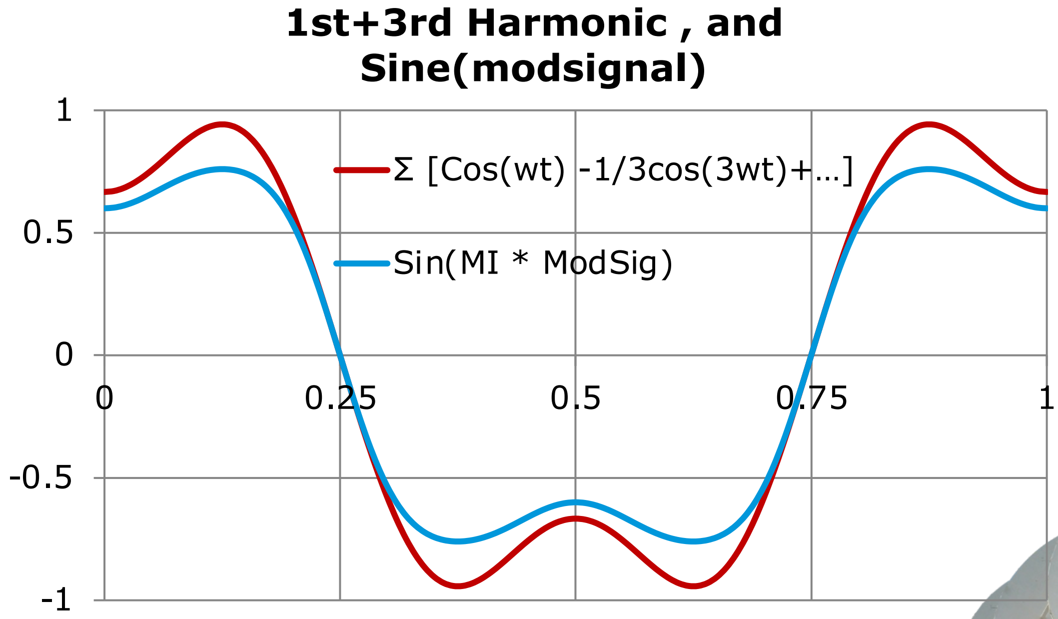


SPL has main lobe Null bandwidth -2Rb to +2Rb, twice that of NRZ

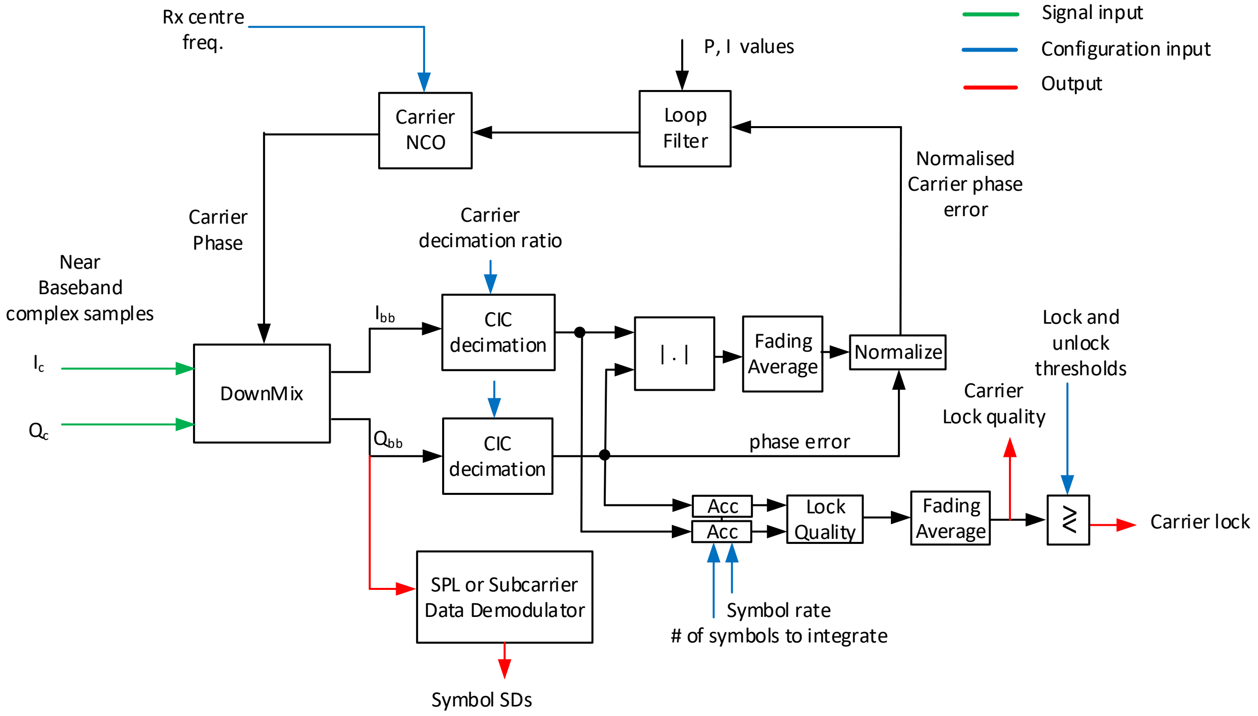
SPL Demodulation options implemented

- Sampling_Rate < 4 x Bit_Rate – Use Sin wave Subcarrier Demod
- Sampling_Rate < 6 x Bit_Rate – Use Sine + 3rd harmonic Subcarrier
- Sampling _Rate > 6 x Bit_rate – Use SPL Demod

PCM/BPSK/PM Subcarrier options include “sine($\sum \cos wt + 1/3 \cos 3wt$ …)"



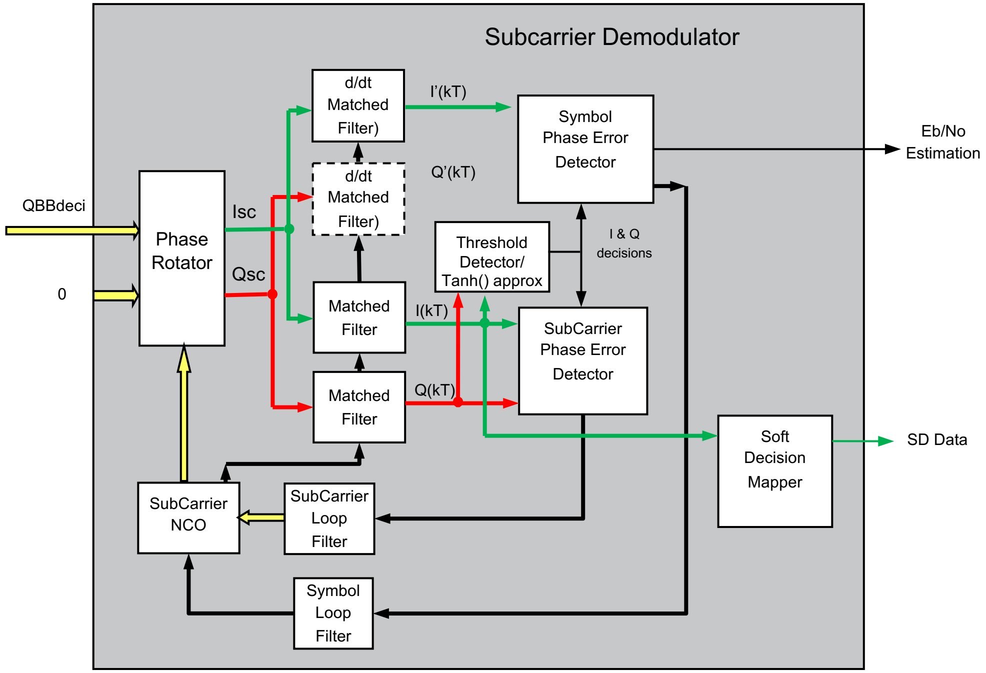
Software Demodulator – Remnant Carrier, PCM/BPSK/PM & SPL



The same Remnant Carrier Demodulator used for PCM/BPSK/PM & SPL.

- Conventional design
- CIC decimator used to get SNR > 0dB to allow C/No estimate and hence correct Phase Detector gain.
- 8 sample "Float" vectors using SIMD instructions in :
  - Carrier NCO
  - Downmix
  - CIC decimator
- Autonomous Carrier Loop narrowing after acquisition

Software Demodulator – PCM/BPSK/PM Demodulator



- "Phase rotator" multiplies real input by

$$Isc = \text{Qin} * \sum(\cos wt + 1/3\cos 3wt + 1/5\cos 5wt \ldots)$$

$$Qsc = \text{Qin} * \sum(\sin wt - \sin 3wt + \cos 5wt \ldots)$$

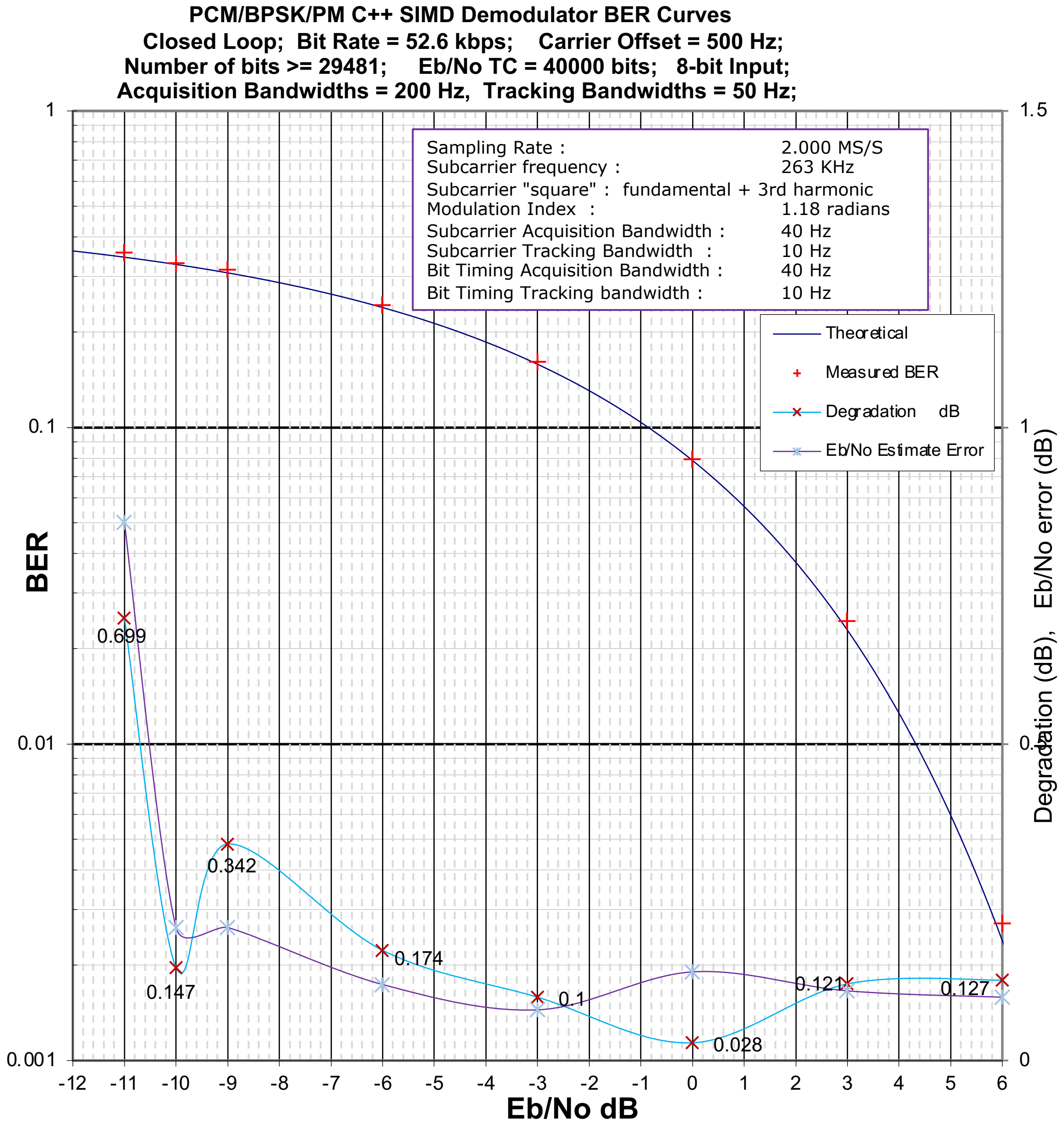
- Similar to a conventional BPSK down-conversion with the Q output, the error signal, multiplying by the differential of the inphase Local oscilator signal. Effectively the sub carrier phase error is formed by integration over the carrier transition.
- I & Q Integrate and Dump Matched filters provide Sub-Carrier power estimate
- I differential matched filter (integration over symbol start – integration over symbol end) provides timing error.
- Timing loop first order as slaved off Subcarrier NCO.
- Es/No estimate by SSME (within Matched filters)

$$Signal + Noise\ Power = (Acc\ over\ 1st\ half + Acc\ over\ 2nd\ half)^2$$

$$Noise\ Power = (Acc\ over\ 1st\ half - Acc\ over\ 2nd\ half)^2$$

- Es/No estimate used to normalise Phase Detector Gains
- Autonomous Subcarrier & Timinmg Loop narrowing after acquisition

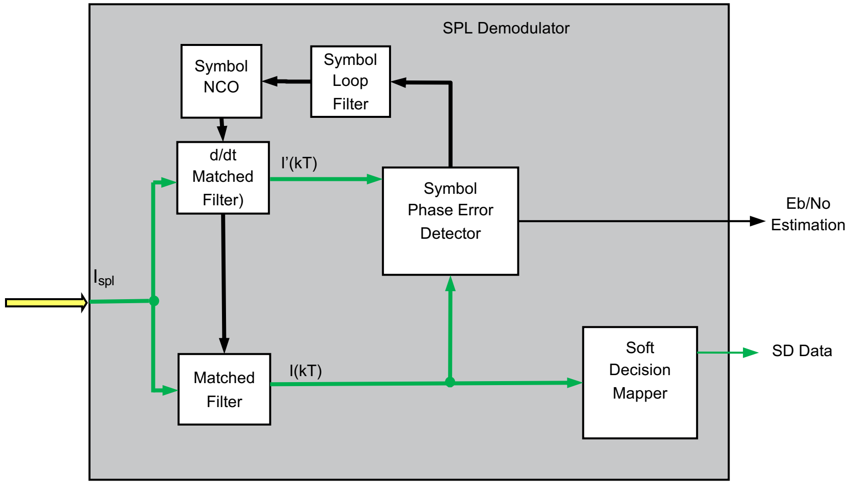
DEMODULATOR TESTING – PCM/BPSK/PM WITH SUBCARRIER



- Processed in Chunks of 80,000 complex samples
- BER degradation < ~ 0.20 dB

- Processing rate 48.6 complex MSamples/S
  on Xeon E5-16200 @ 3.60 GHz Single Thread Rating : 1774
  vs AMD Ryzen 9 5950X @3.4 GHz Single Thread Rating: 3470
  vs AMD Ryzen 9 7950X3D @4.2 HGz Single Thread Rating: 4203
  Approx 50 x real time rate with 2Mcomplex samples/S

Software Demodulator – SPL Demodulator



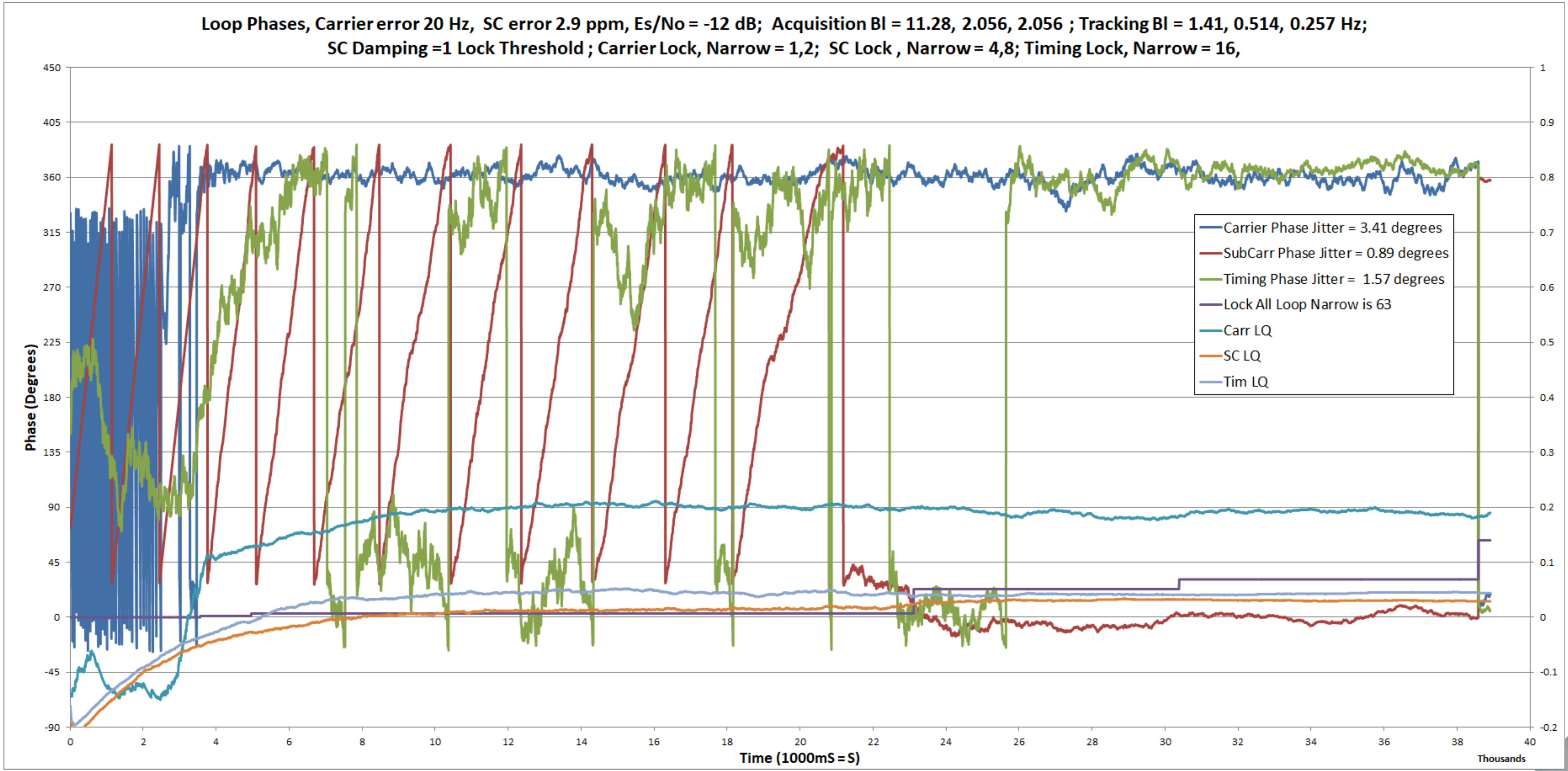
- Conventional design
- Timing Derivative Matched filter uses "on the fly" cosine coefficients over centre ½ of symbol
- Potential Transitions at start and end of symbol not used for timing.
- No SIMD as potentially fewer that 8 samples per symbol
- Es/No estimate by SSME

$$Signal + Noise\ Power = (Acc\ over\ 1st\ half\ -\ Acc\ over\ 2nd\ half)^2$$

$$Noise\ Power = (Acc\ over\ 1st\ half\ +\ Acc\ over\ 2nd\ half)^2$$

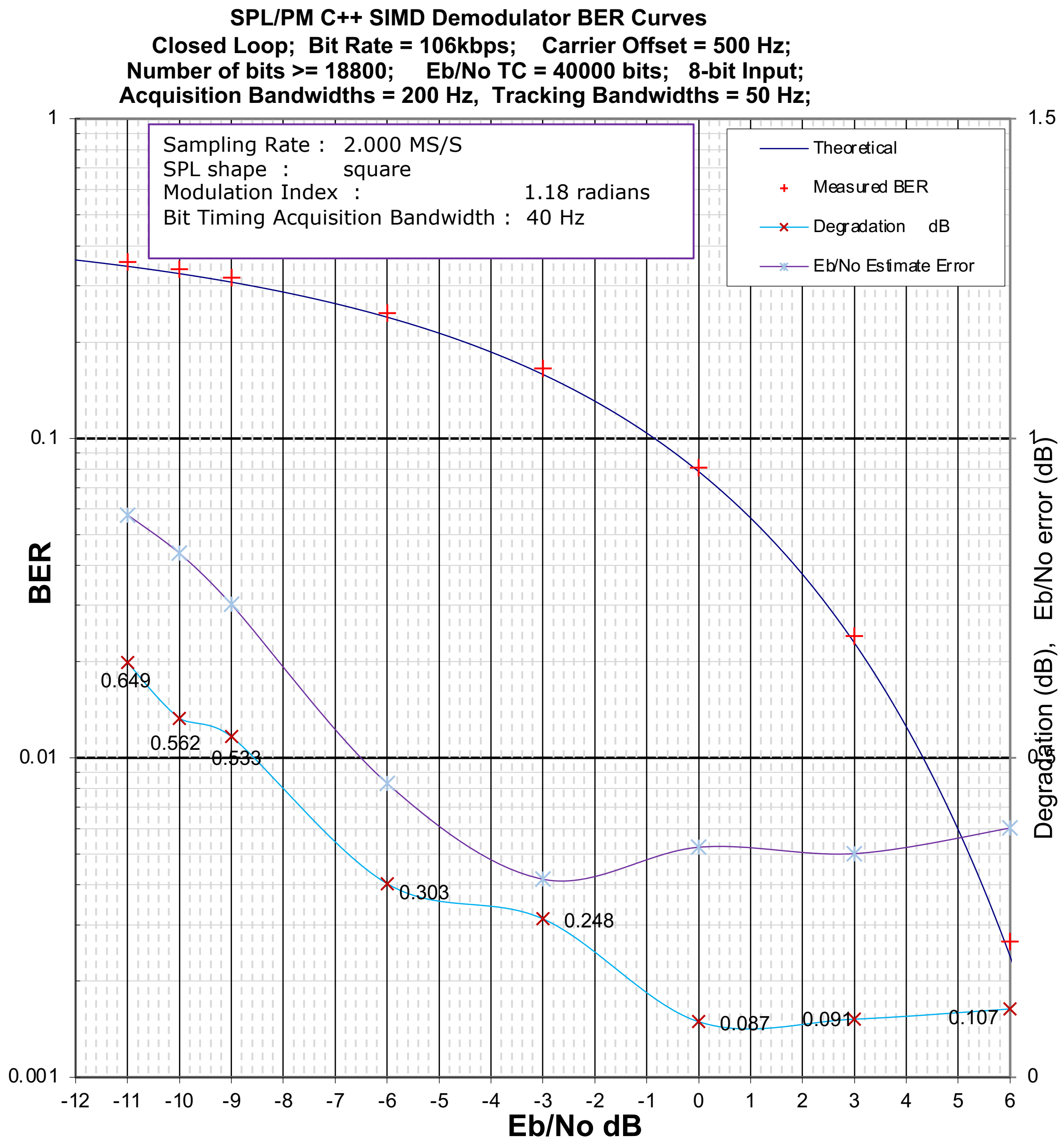
- Es/No estimate used to normalise Phase Detector Gain
- Autonomous Timing Loop narrowing after acquisition

Demodulator Testing – PCM/BPSK/PM Demodulator Acquisition



Subcarrier Frequency 263 kHz, Symbol rate = 2055 bps, 128 Subcarrier cycles per symbol

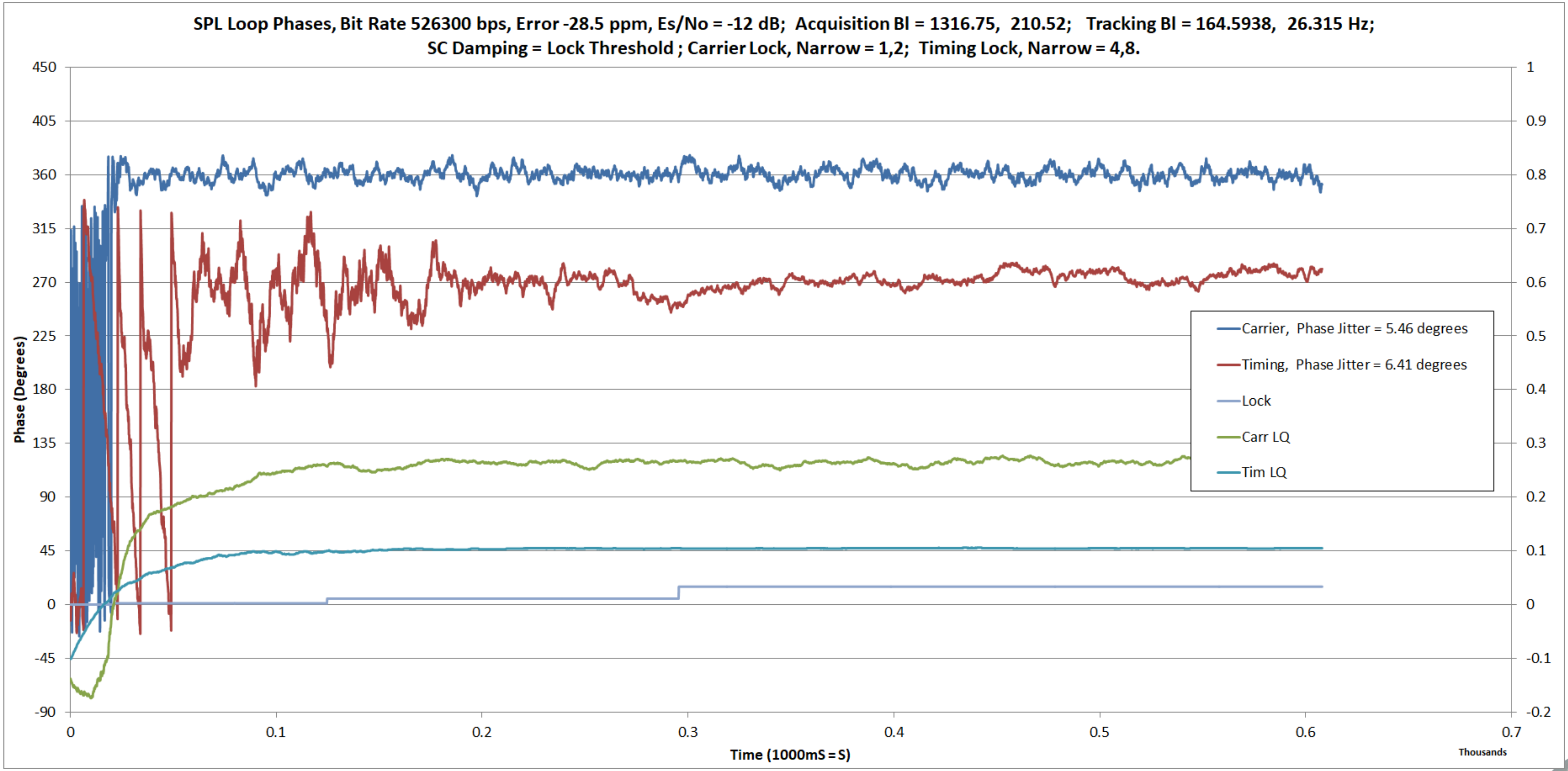
DEMODULATOR TESTING – SPL DEMODULATOR WITH "TRUE" SQUARE MATCHED FILTER

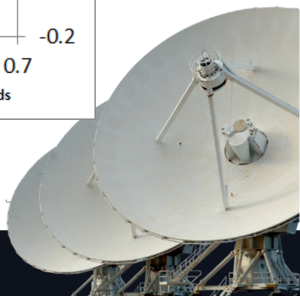


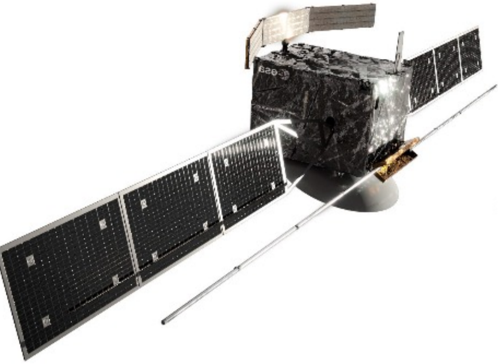
- True Square wave matched filter well matched at high sampling rates
- BER degradation < ~ 0.30 dB (if loop bandwidth low enough)
- BER degradation increases as sampling rate falls due to uncertain matching and ISI.
- Processing rate 64.7 complex MSamples/S

  Approx 65 x real time rate with 2 Mcomplex samples/S

Demodulator Testing – SPL Demodulator Acquisition







OFF-A core services: decoder

Main Core Services – Decoder

OFF-A turbo-codes decoder is inherited from the one used in the PROTOCOL project.

PROTOCOL decoder is a C++ implementation of the BCJR (MAP) algorithm for turbo-codes, tailored for academic use, provided by Università Politecnica delle Marche. It is lossless but characterized by poor execution time performances (53 ksym/s for TurboCode 1/2).

Moreover, the frame-synchronization algorithm relied on a single multi-word correlation. It was therefore not suitable for operational use when re-synchronization is often required due to slip-bits or demodulator unlocking.

In order to meet the OFF-A requirements, the following improvement have been introduced:

- 64-bit compilation
- Automatic compiler code vectorization
- Execution on newer machine

Hardware improvement and compiler options

- Multi-thread (CPU) parallelization by codeword
    - Code refactoring to avoid unnecessary computation
    - From double to floating-point formulation to allow core local caching
    - Use of OpenMP C++ library to manage the codeword loop.

- TTCP-like Frame synchronization strategy.

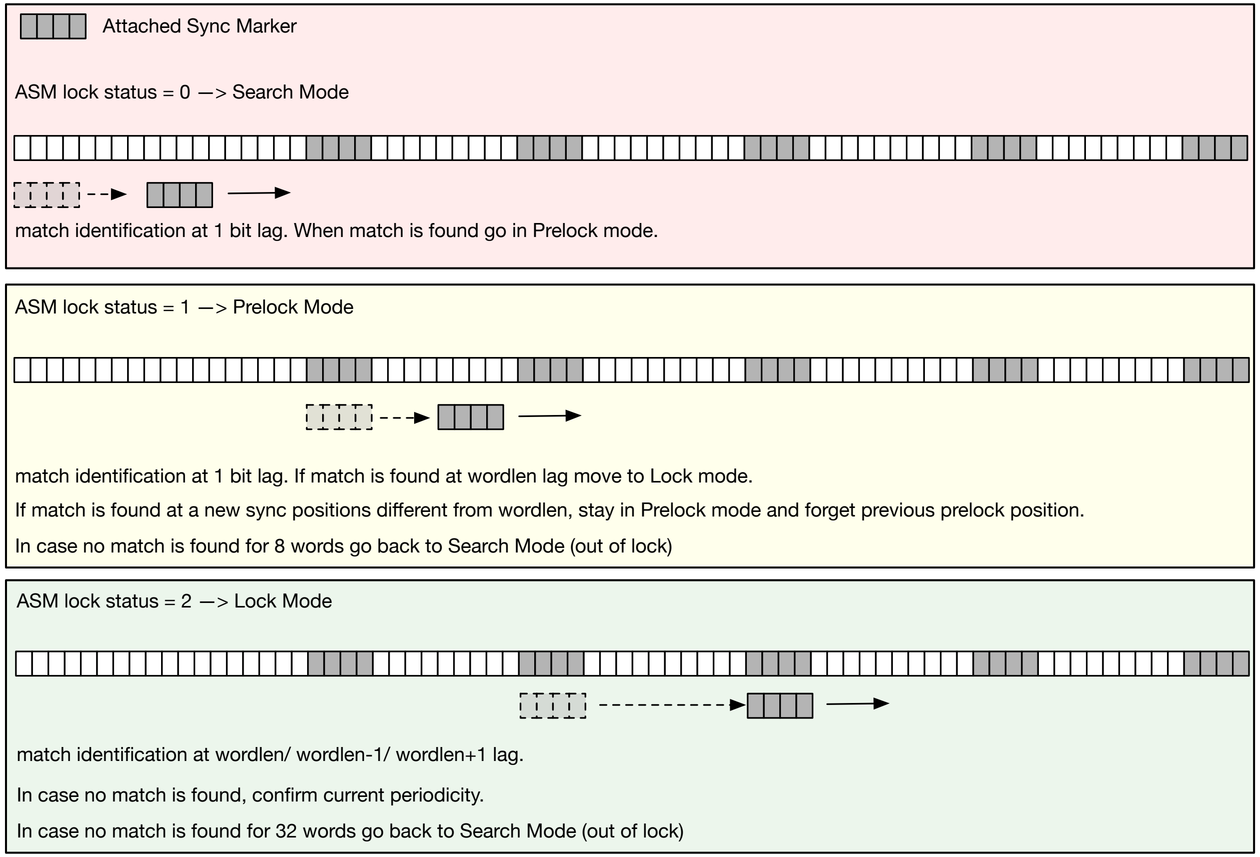
Improved Throughput

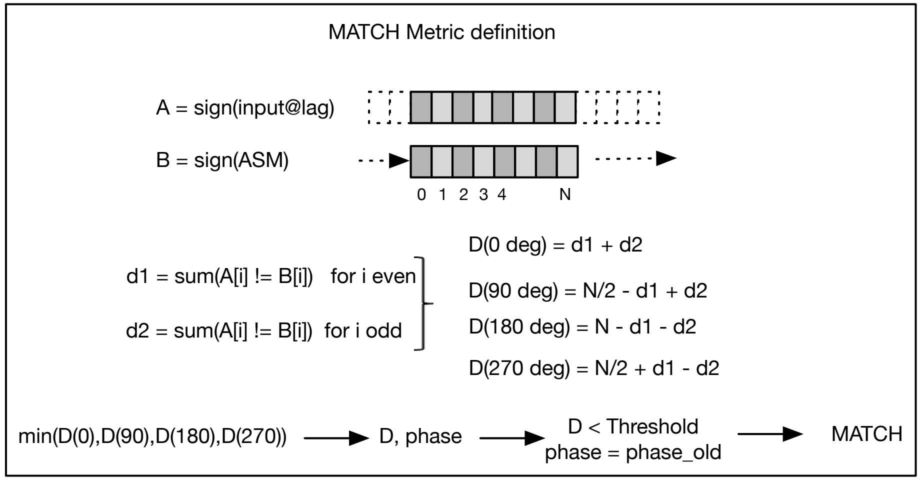
10 Msym/s

Includes FS and implementation losses

Main Core Services – Decoder – Frame Synchronization

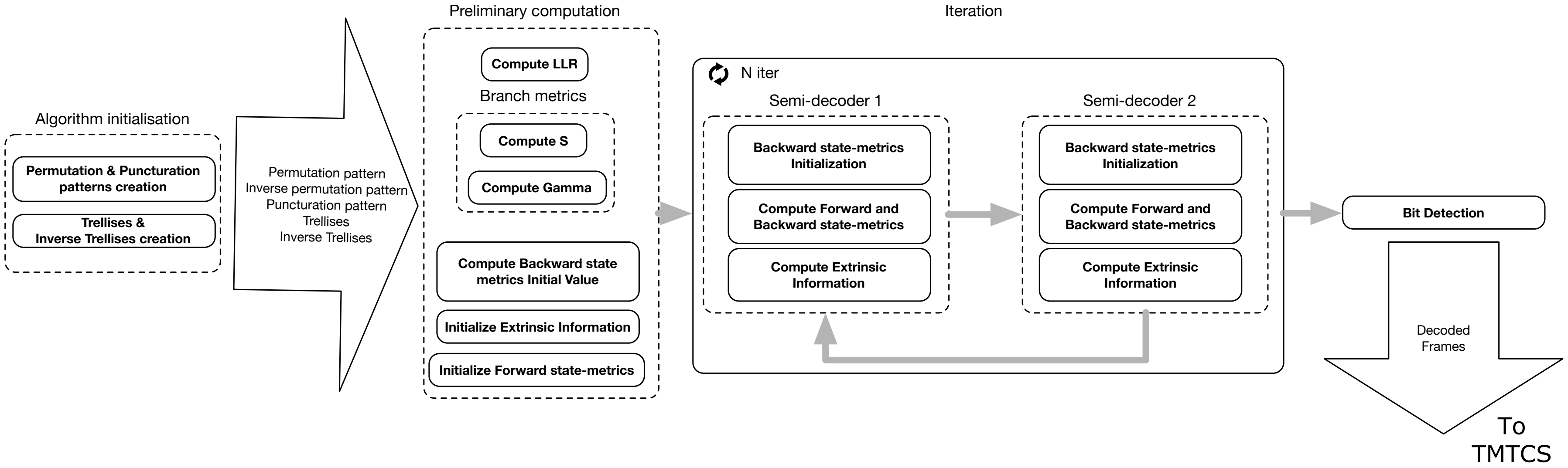
- The Frame Synchronizer cope with bit slips and demodulator re-synchronizations (new sync position and phase ambiguity determination).





Main Core Services – Decoder

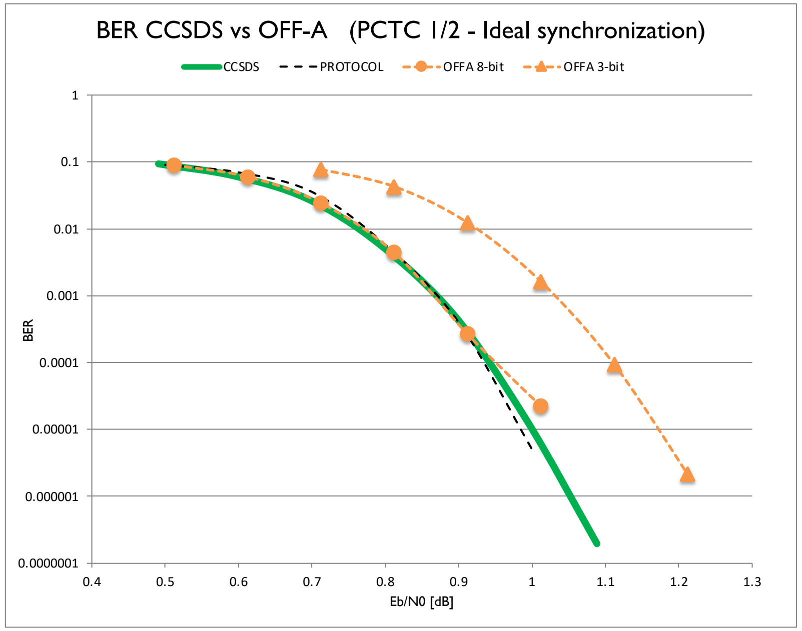
- BCJR algorithm (MAP)
- Floating point LLRs

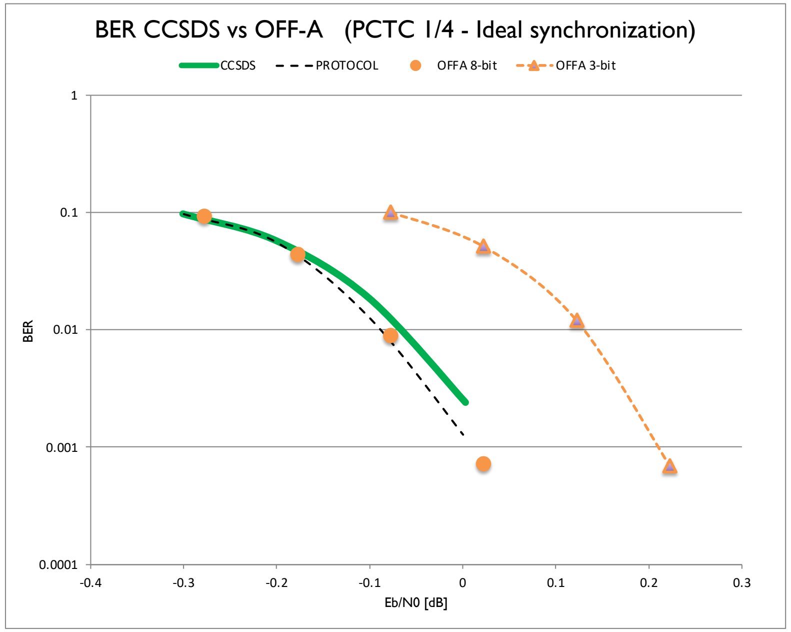


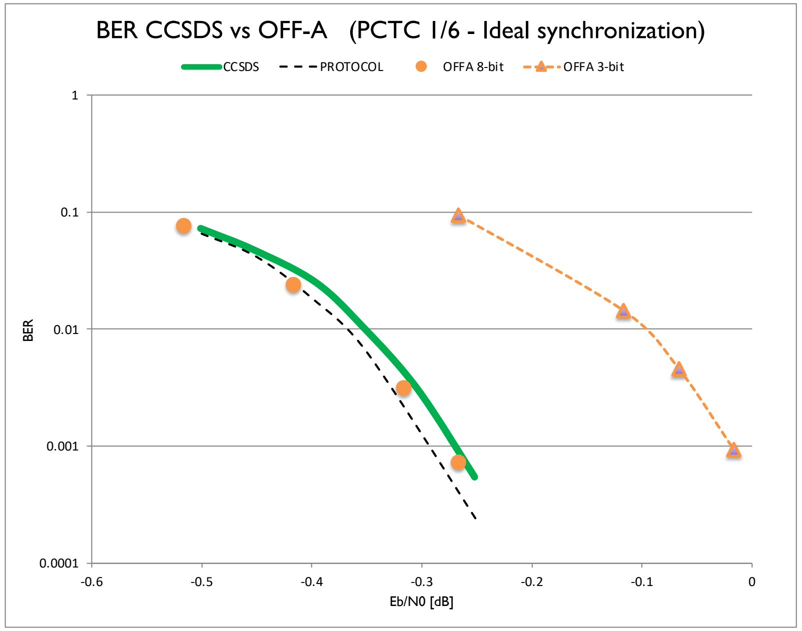
- **Algorithm initialization**: permutation and puncturation pattern, computation of Trellis
- **Preliminary computation**: LLR, branch metrics, forward metrics initialization, backward metrics initial value computation
- **Iteration**: two semi-decoders exchange updated extrinsic information, which is an updated estimate of the a priori LLRs, until a number of iterations is reached.

Main Core Services – Decoder

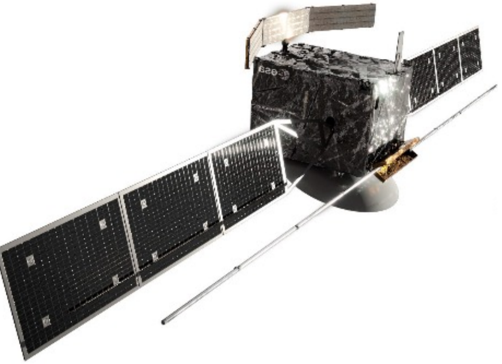
The performance of the decoder standalone has been evaluated assuming ideal synchronization (baseband simulations)







Loss with 3 soft bits is between 0.2 dB and 0.3 dB



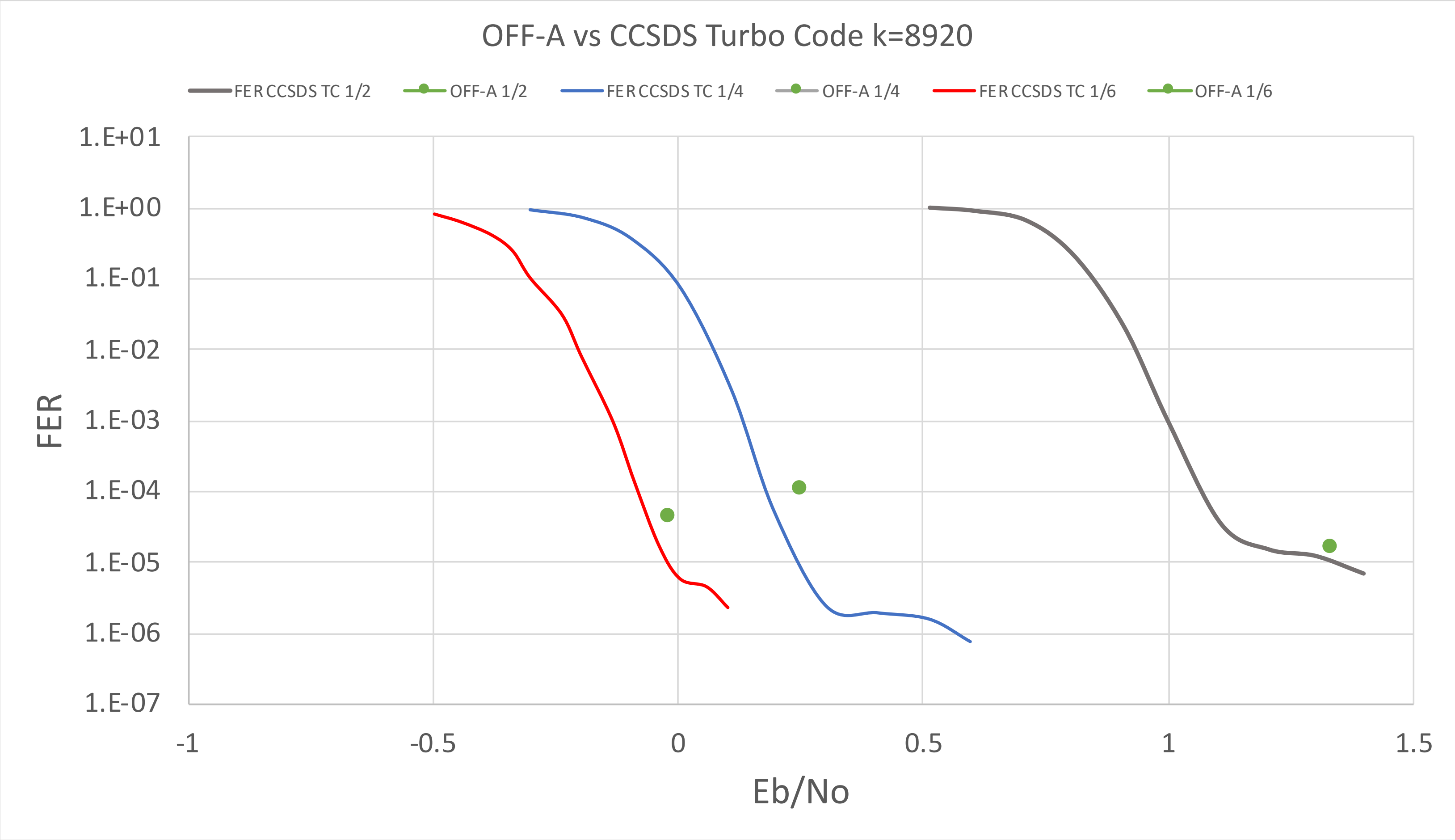
OFF-A E2E tests and performances

OFF-A E2E Tests

Scenario	Freq. band	Mod.	Coding	S/N_0 [dBHz]	Mod. Index [rad-pk]	Subcarrier freq. [Hz]	Info rate [bit/s]	Symbol rate [symb/s]	Single station E_s/N_0 [dB]
1, 5	X	GMSK	TC 1/2	71.85			20000000	40000000	-4.17
2, 6	X	GMSK	TC 1/4	67.76			10000000	40000000	-8.26
3, 7	X	GMSK	TC 1/6	65.71			6666666	40000000	-10.3
4	X	GMSK	TC 1/4	71.91			26000000	104000000	-8.26
8, 12	X	GMSK	TC 1/2	58.84			995980.3	2000000	-4.17
9, 13	X	GMSK	TC 1/4	56.97			829983.6	3333333	-8.26
10, 14	X	GMSK	TC 1/6	54.92			553322.4	3333333	-10.3
11	X	GMSK	TC 1/4	56.97			829983.6	3333333	-8.26
15	X	PCM/PSK/PM	TC 1/6	14.8	0.4305	8190	10.62	63.98	-10.83
16	X	PCM/PSK/PM	TC 1/6	23,879	0.982	263157.9	341.28	2055.92	-10.83
17	X	PCM/PSK/PM	TC 1/6	37.06	1.18	263157.9	8736.67	52631.58	-10.83
18	X	PCM/PSK/PM	TC 1/6	14.8	0.4305	8190	10.62	63.98	-10.83
19	X	PCM/PSK/PM	TC 1/6	23,879	0.982	263157.9	341.28	2055.92	-10.83
20	X	PCM/PSK/PM	TC 1/6	37.06	1.18	263157.9	8736.67	52631.58	-10.83
21	Ka	PCM/SP-L	TC 1/2	58.79	1.0		696240.73	1398101.333	-4.17
22	Ka	PCM/SP-L	TC 1/4	47.42	1.0		65272.57	262144	-8.26
23	Ka	PCM/SP-L	TC 1/6	39.34	1.0		10849.46	65359.4771	-10.3
24	Ka	PCM/SP-L	TC 1/2	58.79	1.0		696240.73	1398101.333	-4.17
25	Ka	PCM/SP-L	TC 1/4	47.42	1.0		65272.57	262144	-8.26
26	Ka	PCM/SP-L	TC 1/6	39.34	1.0		10849.46	65359.4771	-10.3

OFF-A Overall Loss FSC 8 bits GMSK

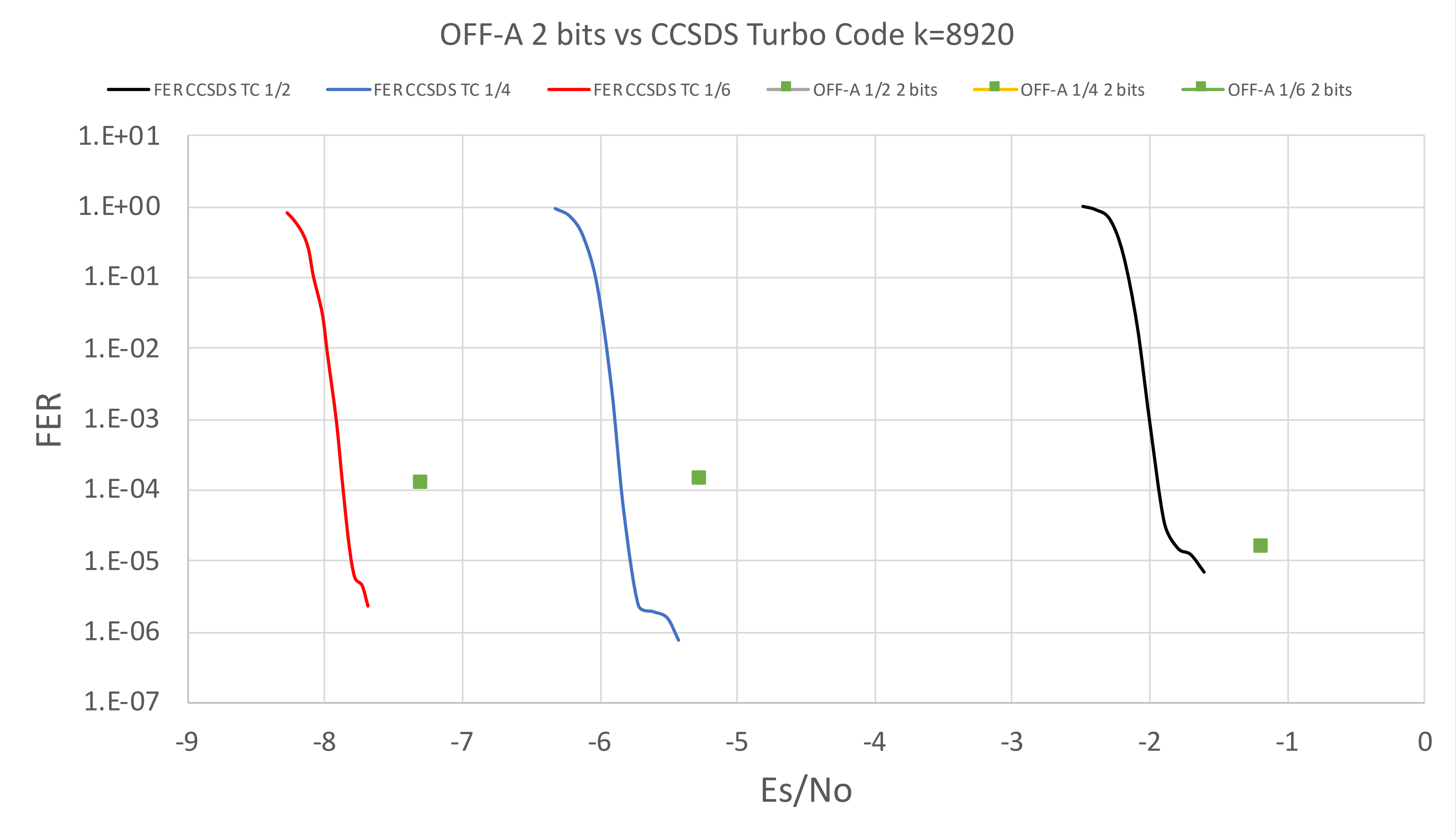
Modulation	Symbol rate	Info rate	Sampling rate	Coding	Combined Es/N0	Simulated Es/N0
GMSK	40 Msymbols/s	20 Mb/s	43.75 MHz	TC 1/2 k=8920	-1.67 dB	-4.67 dB
		10 Mb/s		TC 1/4 k=8920	-5.76 dB	-8.76 dB
		6.6 Mb/s		TC 1/6 k=8920	-7.8 dB	-10.8 dB



	Performance	Requirement
✓	< 0.13 dB	0.4 dB

OFF-A Overall Loss FSC 2 bits GMSK

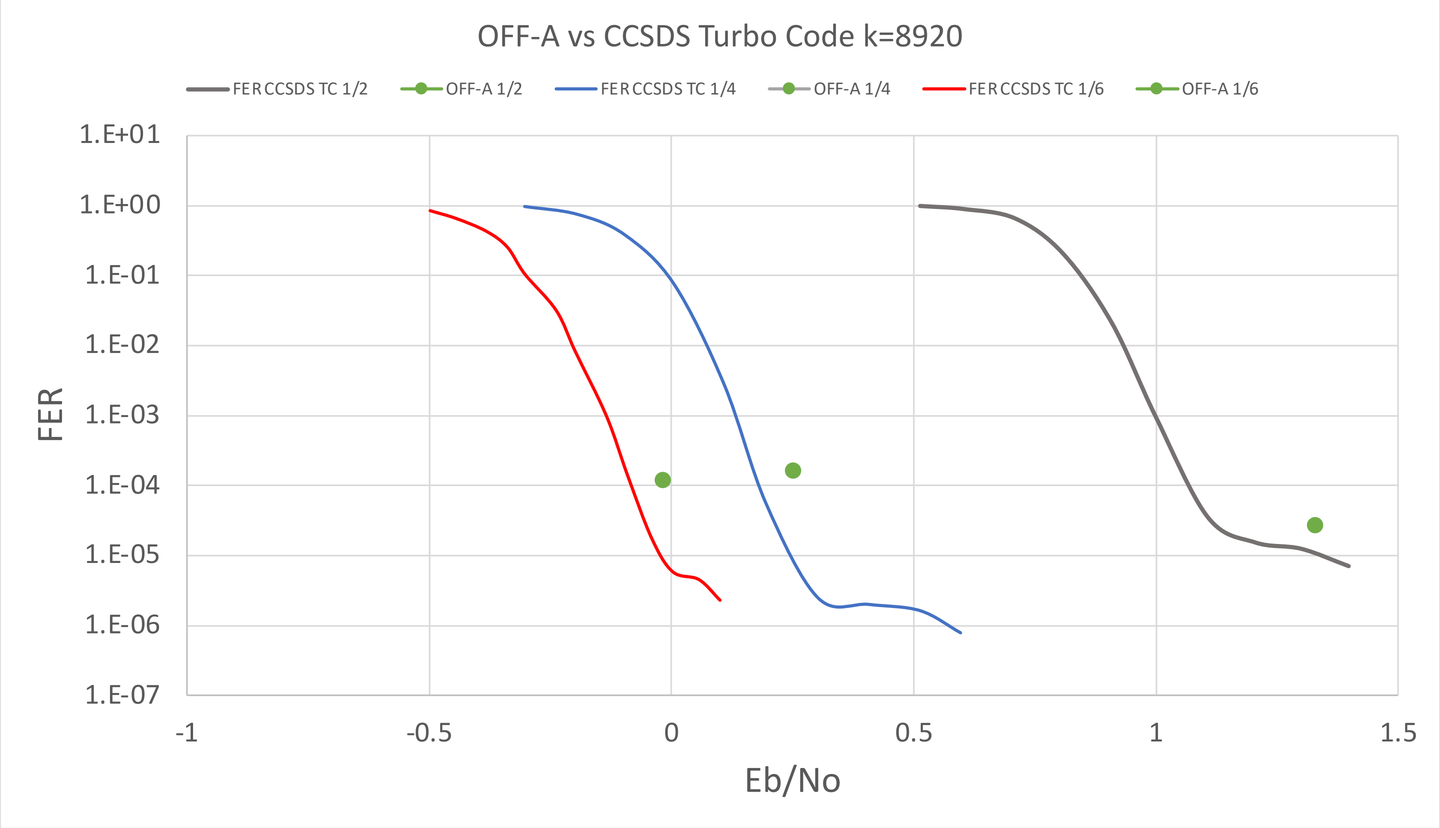
Modulation	Symbol rate	Info rate	Sampling rate	Coding	Combined Es/N0	Simulated Es/N0
GMSK	40 Msymbols/s	20 Mb/s	43.75 MHz	TC 1/2 k=8920	-1.17 dB	-4.17 dB
		10 Mb/s		TC 1/4 k=8920	-5.26 dB	-8.26 dB
		6.6 Mb/s		TC 1/6 k=8920	-7.3 dB	-10.3 dB



	Performance	Requirement
✓	**< 0.63 dB**	0.9 dB

OFF-A Overall Loss SSC 8 bits GMSK

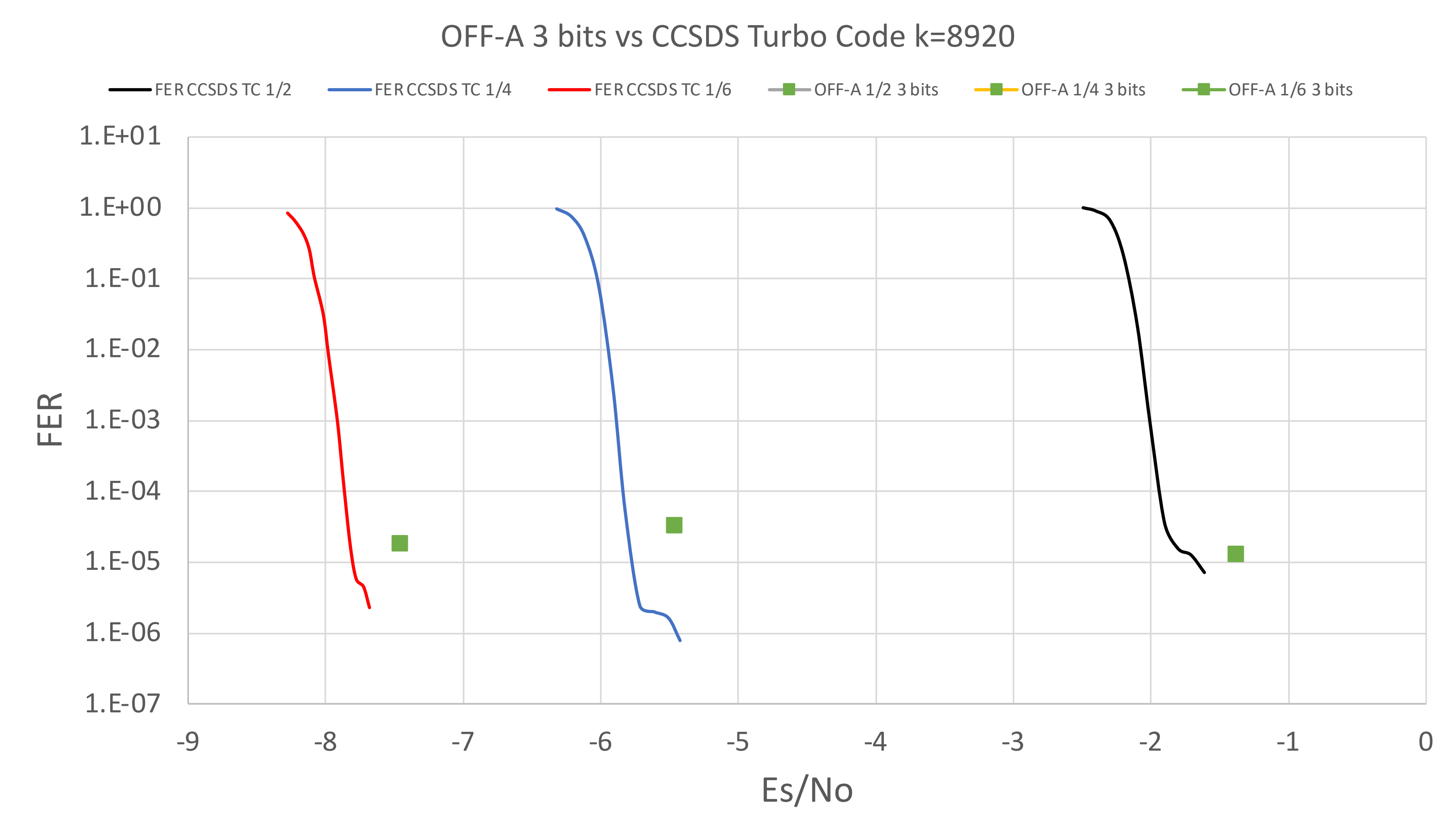
Modulation	Symbol rate	Info rate	Sampling rate	Coding	Combined Es/N0	Simulated Es/N0
GMSK	40 Msymbols/s	20 Mb/s	43.75 MHz	TC 1/2 k=8920	-1.67 dB	-4.67 dB
		10 Mb/s		TC 1/4 k=8920	-5.76 dB	-8.76 dB
		6.6 Mb/s		TC 1/6 k=8920	-7.8 dB	-10.8 dB



	Performance	Requirement
✔	< 0.18 dB	0.4 dB

OFF-A Overall Loss SSC 3 bits GMSK

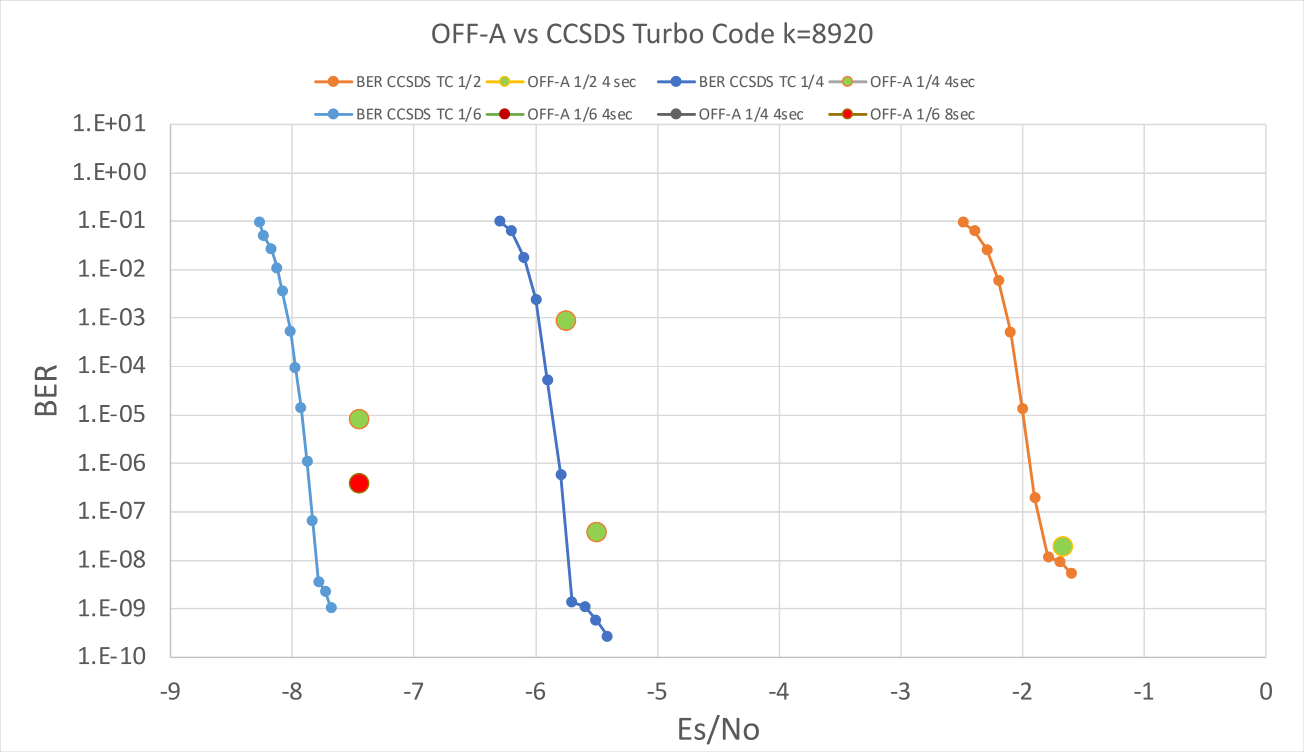
Modulation	Symbol rate	Info rate	Sampling rate	Coding	Combined Es/N0	Simulated Es/N0
GMSK	40 Msymbols/s	20 Mb/s	43.75 MHz	TC 1/2 k=8920	-1.67 dB	-4.67 dB
		10 Mb/s		TC 1/4 k=8920	-5.76 dB	-8.76 dB
		6.6 Mb/s		TC 1/6 k=8920	-7.8 dB	-10.8 dB



	Performance	Requirement
✓	**< 0.35 dB**	0.9 dB

OFF-A Overall Loss FSC PCM/BPSK/PM

Modulation	Sub. Freq.	Symbol rate	Info rate	Sampling rate	Coding	Combined Es/N0	Simulated Es/N0
PCM/BPSK/PM	263157.9 Hz	52631.58 symbols/s	8736.67 b/s	4 MHz	TC 1/2 k=8920	-1.67 dB	-4.67 dB
					TC 1/4 k=8920	-5.76 dB	-8.76 dB
					TC 1/6 k=8920	-7.8 dB	-10.8 dB



	Performance	Requirement
	< 0.4 dB	0.4 dB

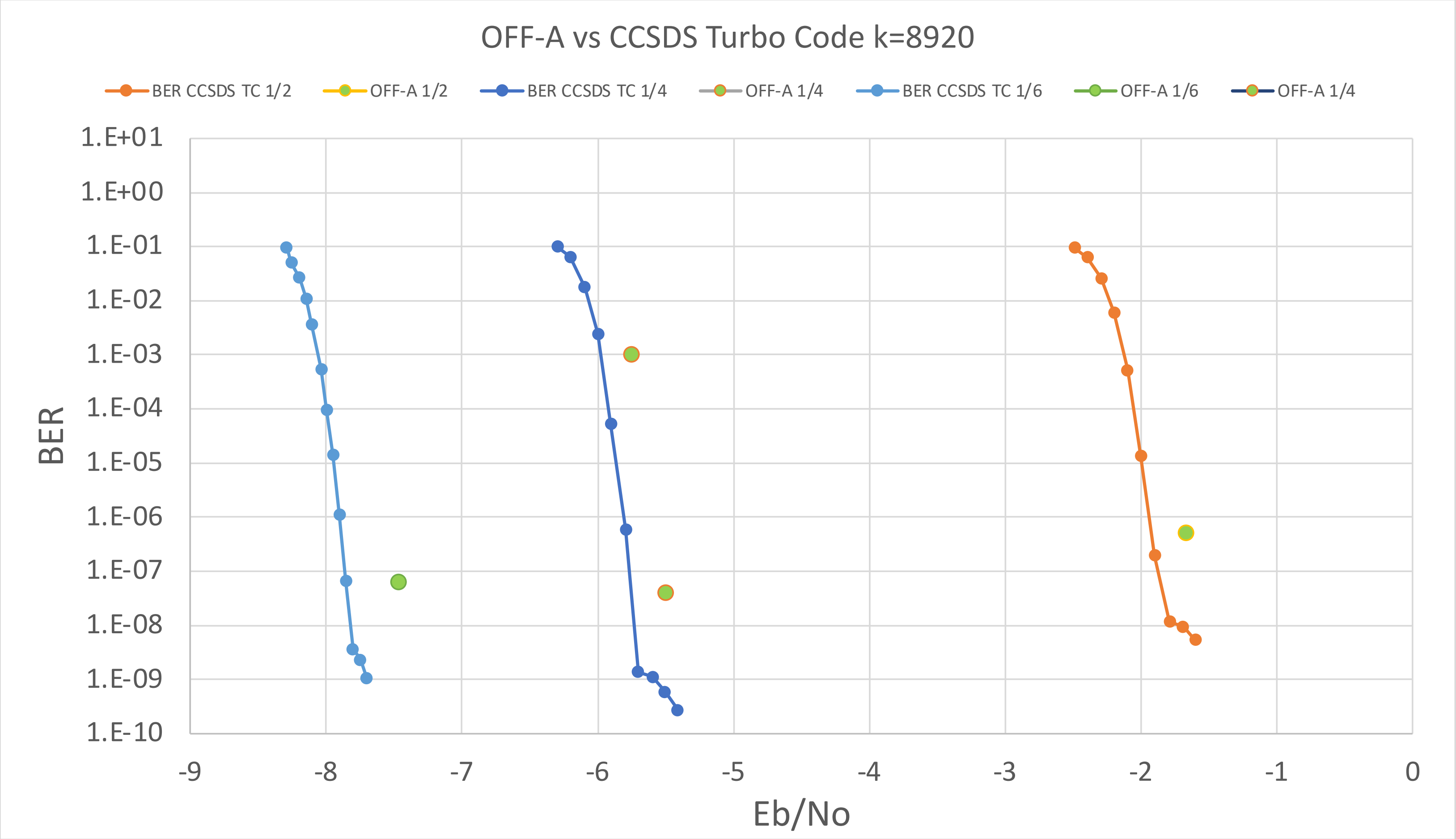
! Larger integration time needed for low rates. Currently not feasible in OFF-A.

Long integration time may be not compatible with phase variations due to troposphere and other sources.

Modulation	Sub. Freq.	Symbol rate	Info rate	Coding	Integration time
PCM/BPSK/PM	263157.9 Hz	2055.92 symbols/s	341.28 b/s	TC 1/6 k=8920	3-4 minutes
	8190 Hz	63.98 symbol/s	10.62 b/s		10-15 minutes

OFF-A Overall Loss SSC PCM/BPSK/PM

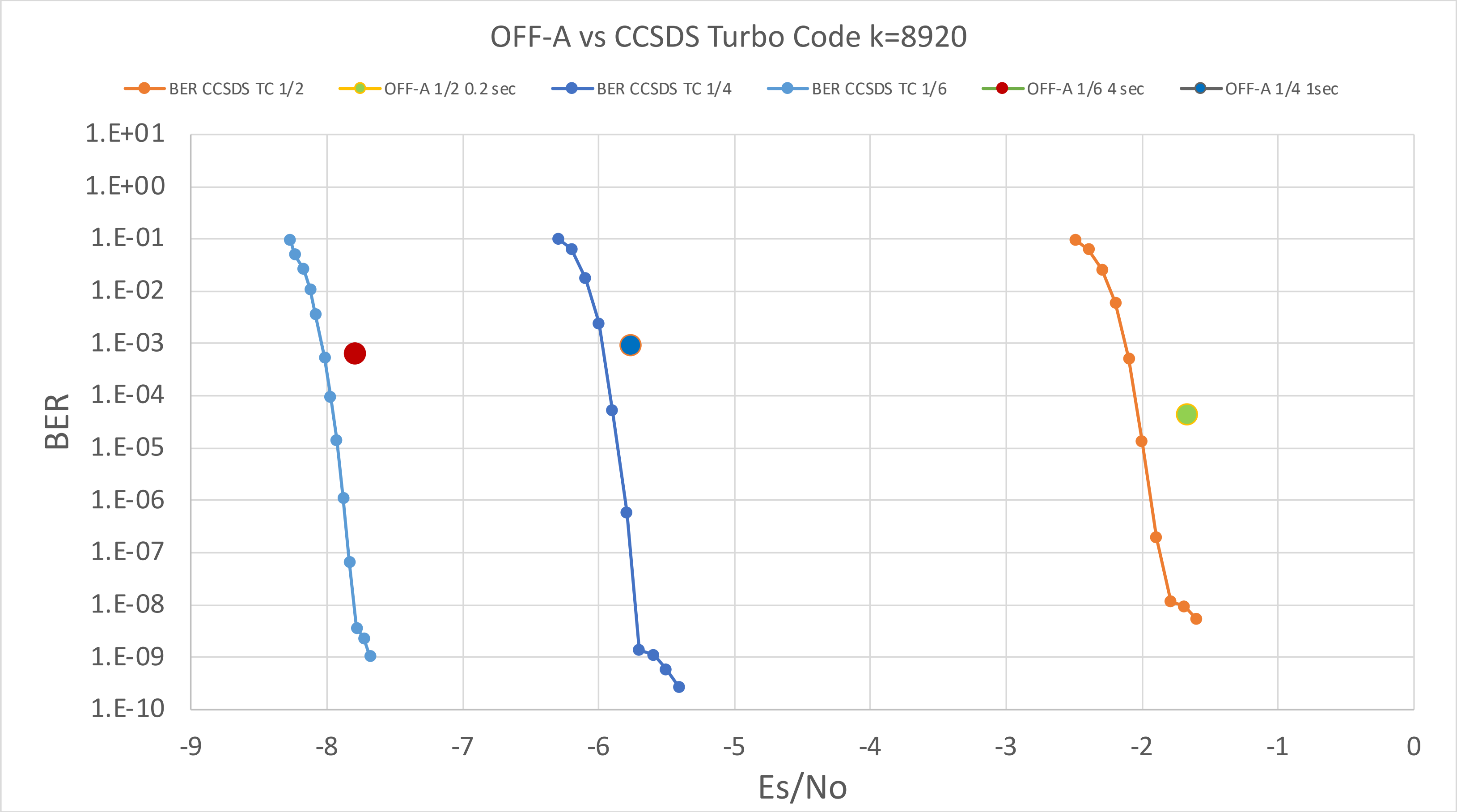
Modulation	Sub. Freq.	Symbol rate	Info rate	Sampling rate	Coding	Combined Es/N0	Simulated Es/N0
PCM/BPSK/PM	263157.9 Hz	52631.58 symbols/s	8736.67 b/s	4 MHz	TC 1/2 k=8920	-1.67 dB	-4.67 dB
					TC 1/4 k=8920	-5.76 dB	-8.76 dB
					TC 1/6 k=8920	-7.8 dB	-10.8 dB



	Performance	Requirement
✓	**< 0.4 dB**	0.4 dB

OFF-A Overall Loss FSC PCM/SP-L

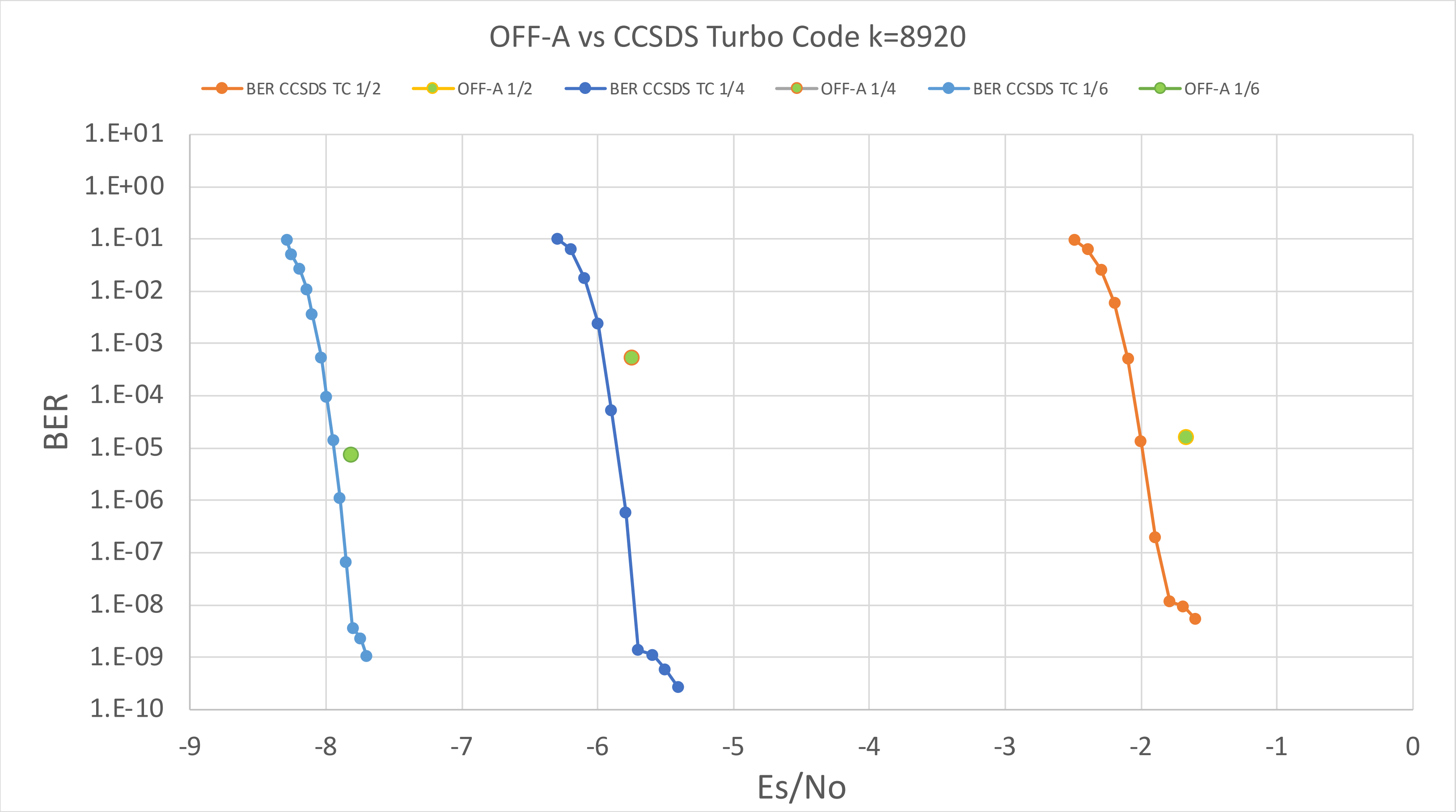
Modulation	Symbol rate	Info rate	Sampling rate	Coding	Combined Es/N0	Simulated Es/N0
PCM/SP-L	1398101 symbols/s	696240 b/s	16 MHz	TC 1/2 k=8920	-1.67 dB	-4.67 dB
	262144 symbols/s	65272 b/s	6 MHz	TC 1/4 k=8920	-5.76 dB	-8.76 dB
	65359 symbols/s	10849 b/s	1 MHz	TC 1/6 k=8920	-7.8 dB	-10.8 dB



Performance	Requirement
< 0.4 dB	0.4 dB

OFF-A Overall Loss SSC PCM/SP-L

Modulation	Symbol rate	Info rate	Sampling rate	Coding	Combined Es/N0	Simulated Es/N0
PCM/SP-L	1398101 symbols/s	696240 b/s	16 MHz	TC 1/2 k=8920	-1.67 dB	-4.67 dB
	262144 symbols/s	65272 b/s	6 MHz	TC 1/4 k=8920	-5.76 dB	-8.76 dB
	65359 symbols/s	10849 b/s	1 MHz	TC 1/6 k=8920	-7.8 dB	-10.8 dB



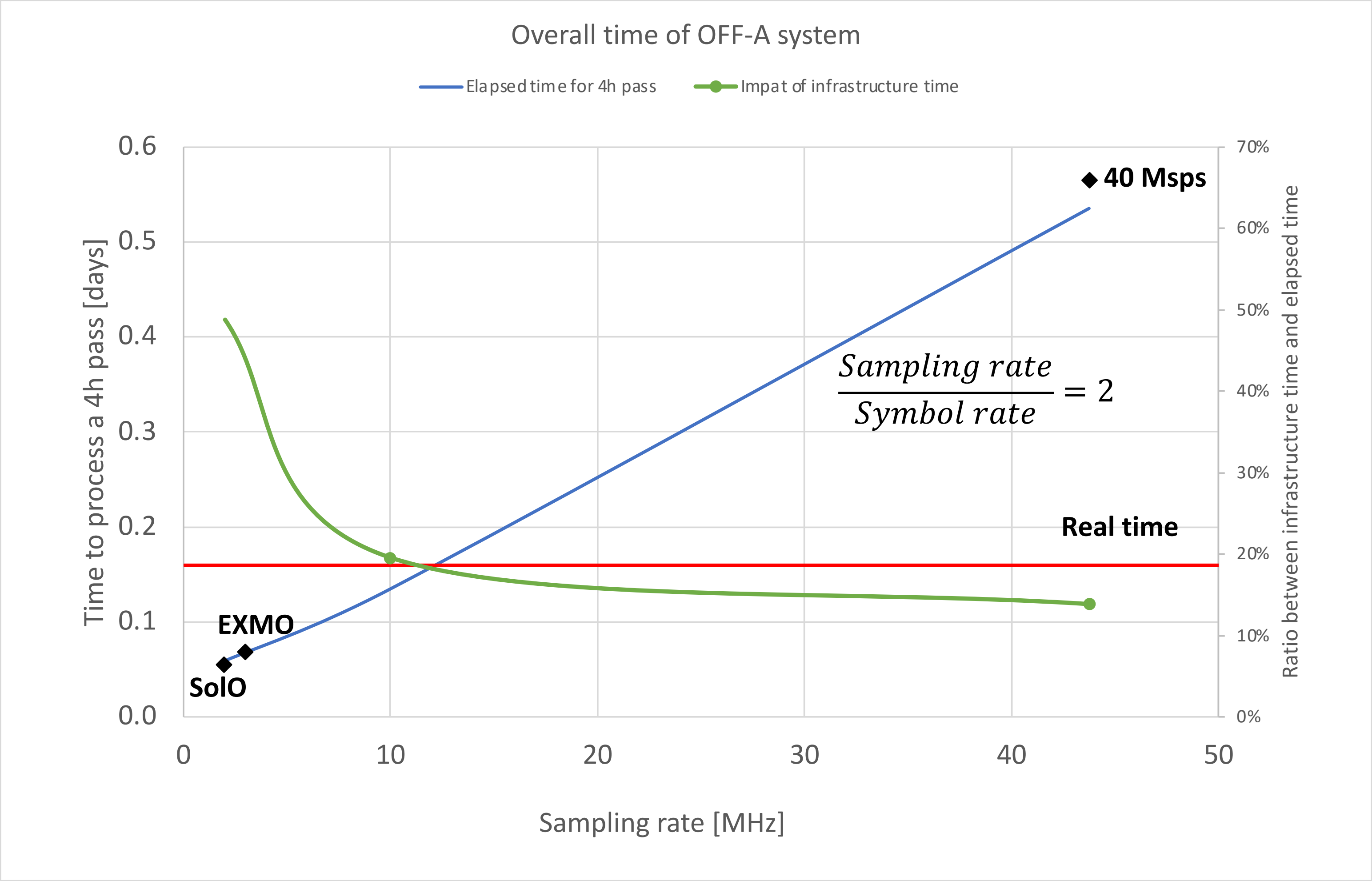
	Performance	Requirement
✓	**< 0.4 dB**	0.4 dB

OFF-A Processing times - GMSK

The overall time is driven by the slowest service.

Elapsed time is composed of:

- Processing time (core services)
- Infrastructure time (infrastructure services)

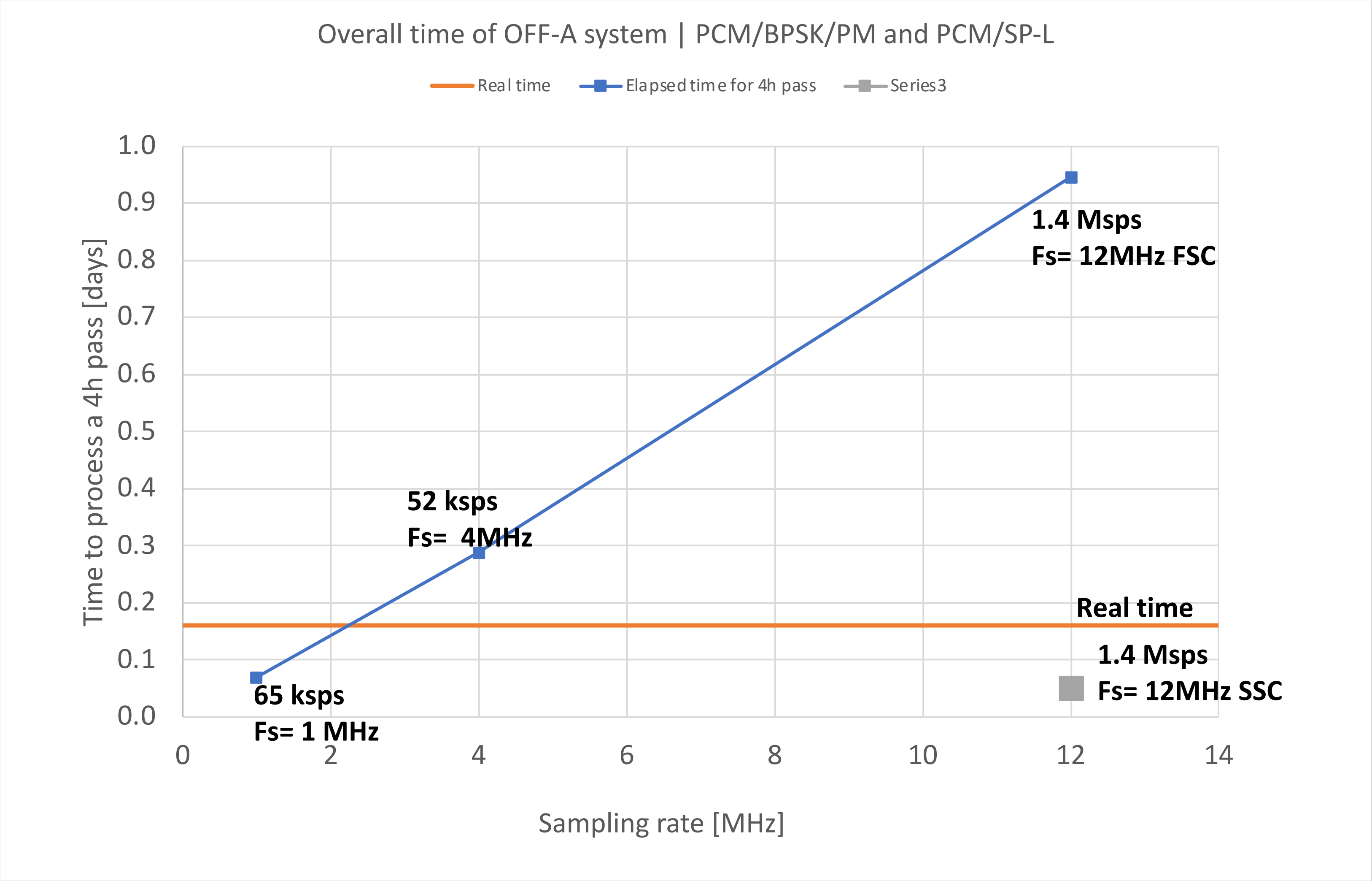


	Symbol rate	Performance	Req.
✓	40 Msymbols/s	**16 hours**	5 days
✓	104 Msymbols/s	**1.8 days**	5 days

No significant gain (10-15%) with SSC (performances of decoder similar to correlator/combiner)

OFF-A Processing times – PCM/BPSK/PM and PCM/SP-L

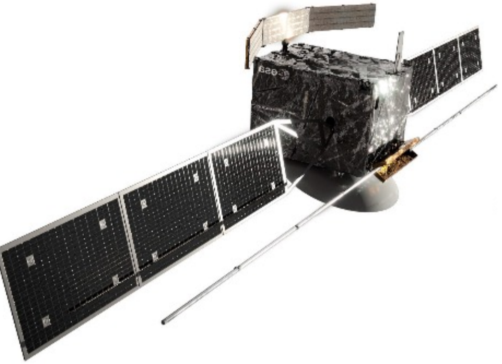
Processing time for FSC driven by sampling rate. Given the large ratio between sampling rate and symbol rate, **SSC is much faster**.



	Symbol rate	Performance	Req.
✓	1.4 Msymbols/s @ 12 MHz	**0.95 days**	1 day

FSC is real-time @ 2 MHz (symbol rate < 250 ksymbols/s).

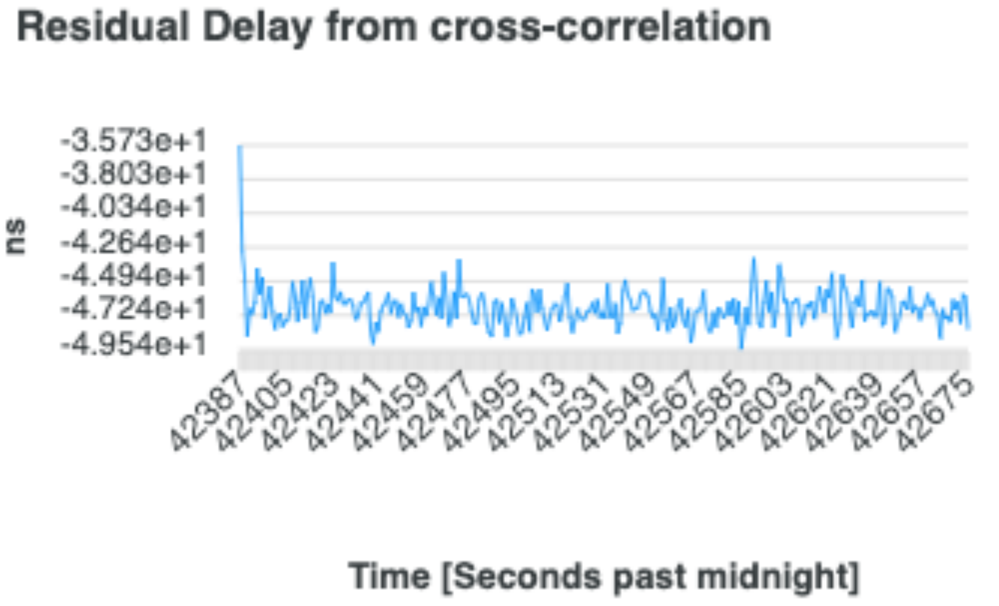
SSC always real-time.

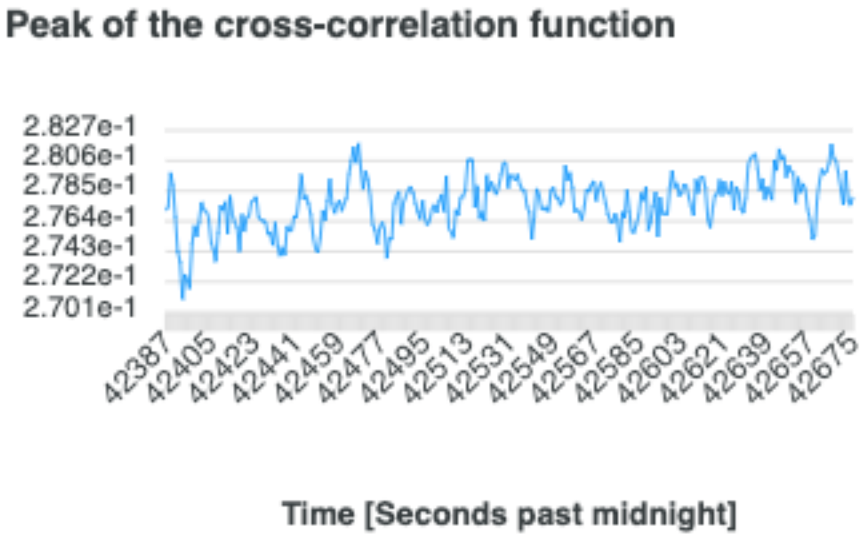


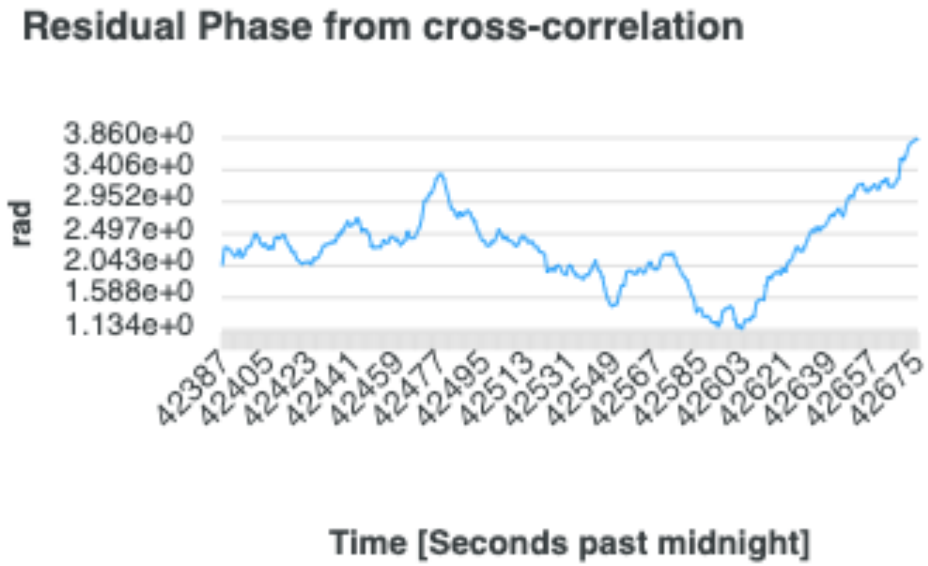
OFF-A tests with Exomars and SolO

DATA FROM SOLO DOY 226 (1)

SolO DOY 226	
Symbol rate	909.909 ksps
Info rate	226.367 kbps
Sampling rate	2 MHz
Quantization	8 bits
Es/No CB (69 deg)	0.4 dB (from TTCP)
ES/No MG (8 deg)	-1.78 dB (from TTCP)

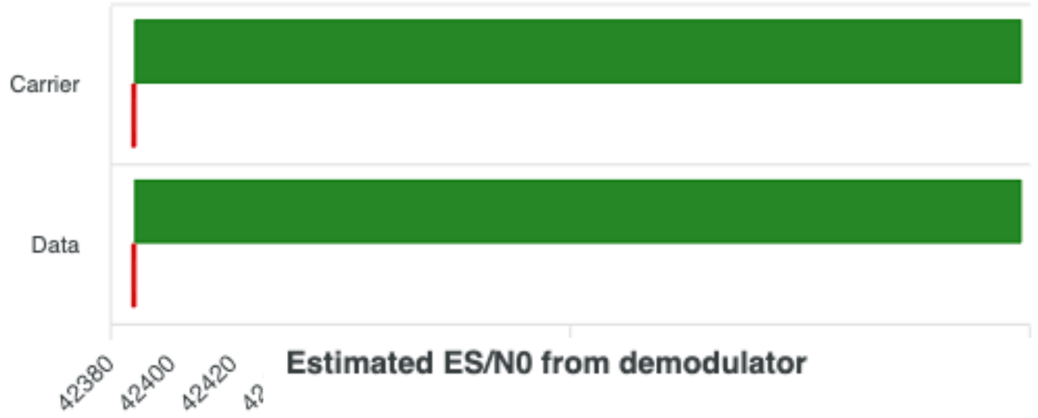




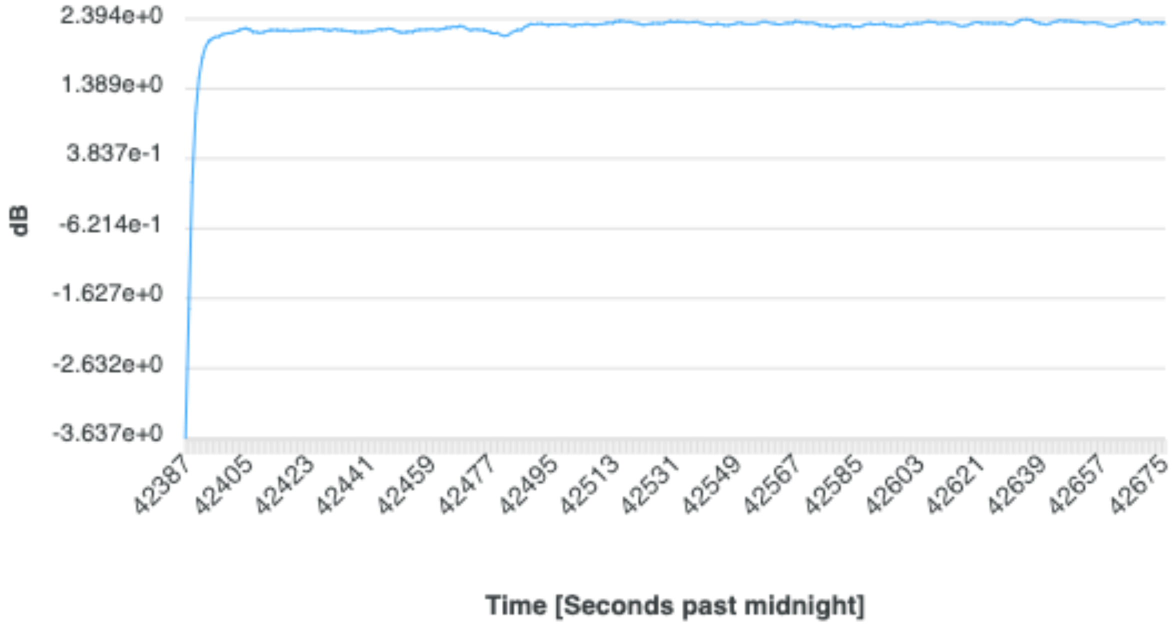


Data from SoLO DOY 226 (2)

Demodulator lock status



Estimated ES/N0 from demodulator




```
00 1010001010 011 1 01000100 00111010
00 1010001010 011 1 01000101 00111011
00 1010001010 011 1 01000110 00111100
00 1010001010 011 1 01000111 00111101
00 1010001010 011 1 01001000 00111110
00 1010001010 011 1 01001001 00111111
00 1010001010 011 1 01001010 01000000
00 1010001010 011 1 01001011 01000001
```

Spacecraft ID: 650

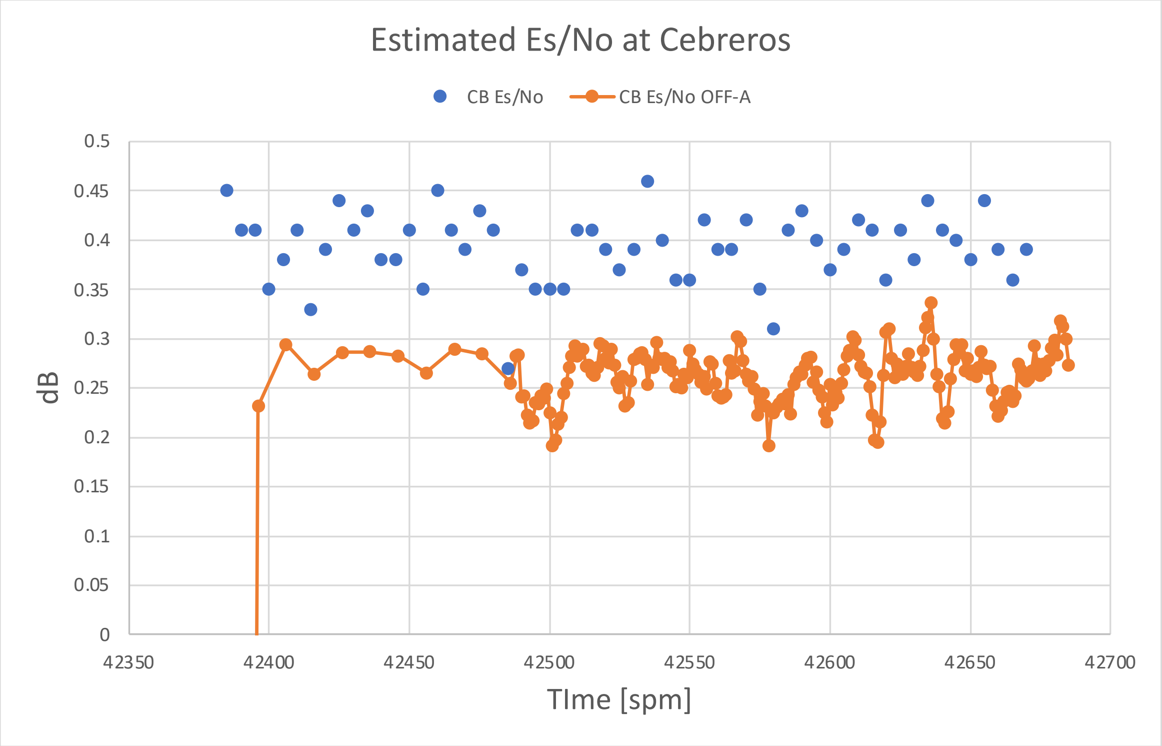
DATA FROM SOLO DOY 226 (3)

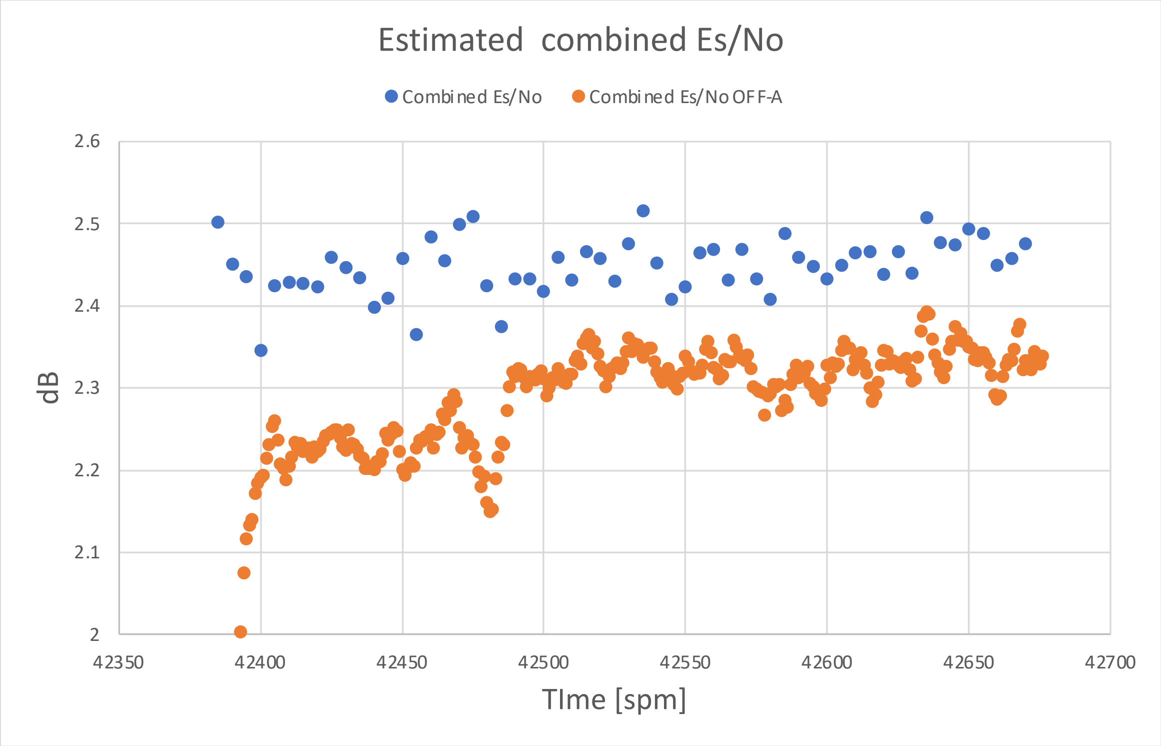
Comparative analysis with TTCP info (ideal combination).

$$\frac{E_S}{N_{0COMB}} = \frac{E_S}{N_{01}} + \frac{E_S}{N_{02}}$$

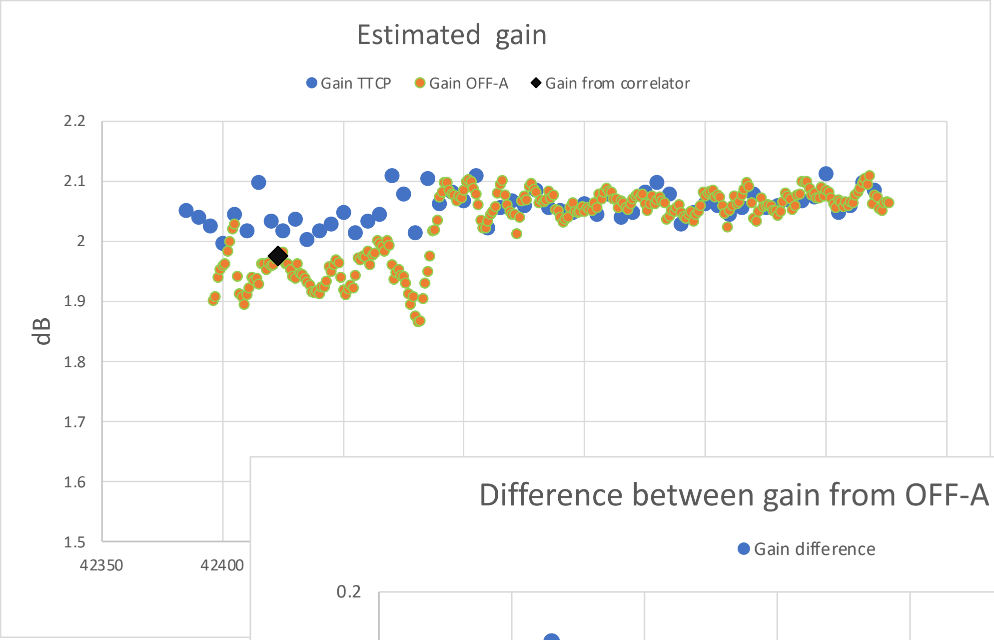
Es/No ➡ combined Es/No, gain, maximum of correlation function.

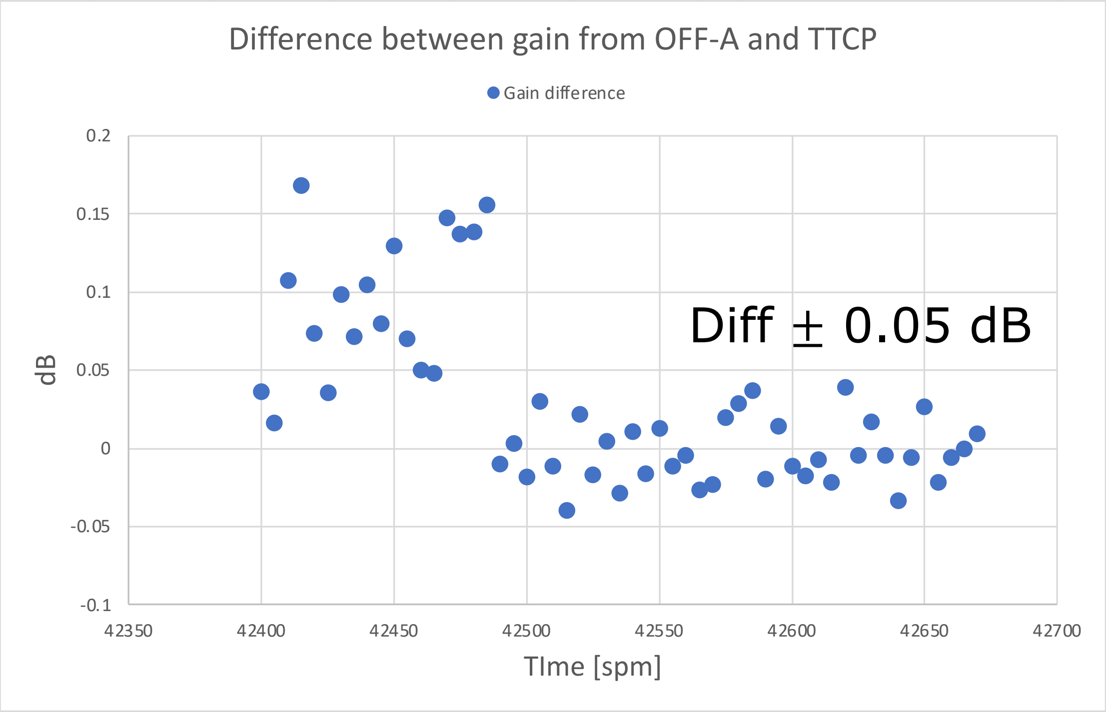
$$\text{SNR} = \frac{E_S}{N_0} \cdot \frac{Symbol\ rate}{Sampling\ rate}$$

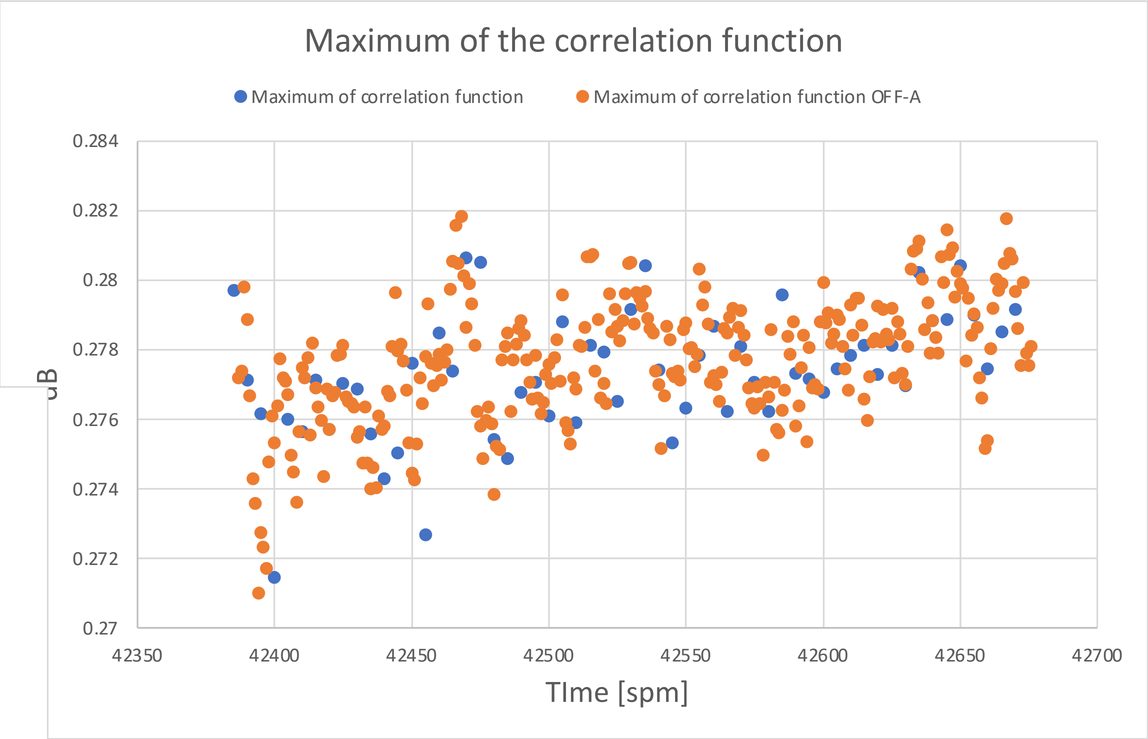




Data from SolO DOY 226 (4)



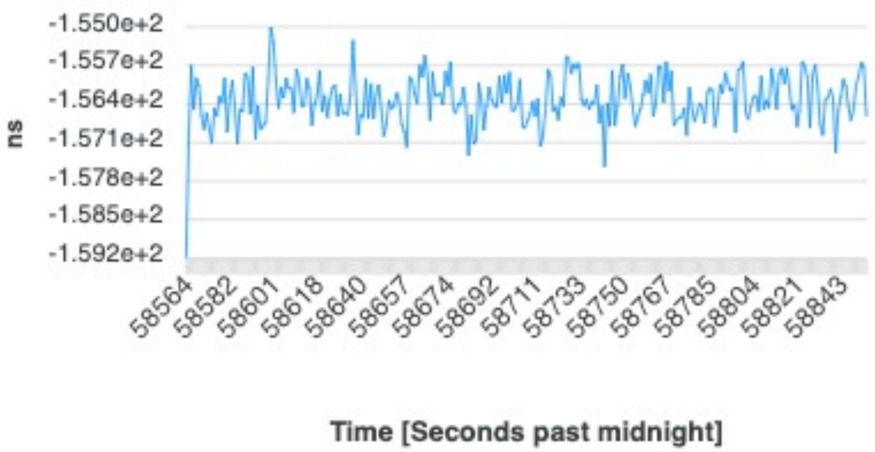




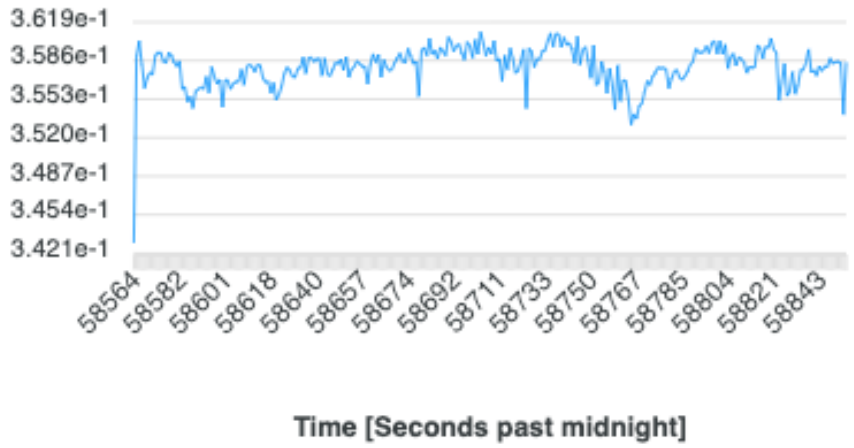
DATA FROM EXOMARS DOY 227 (I)

EXMO DOY 227	
Symbol rate	1666.666 ksps
Info rate	415.006 kbps
Sampling rate	3 MHz
Quantization	8 bits
Es/No CB (46 deg)	0.19 dB (from TTCP)
ES/No MG (36 deg)	-0.03 dB (from TTCP)

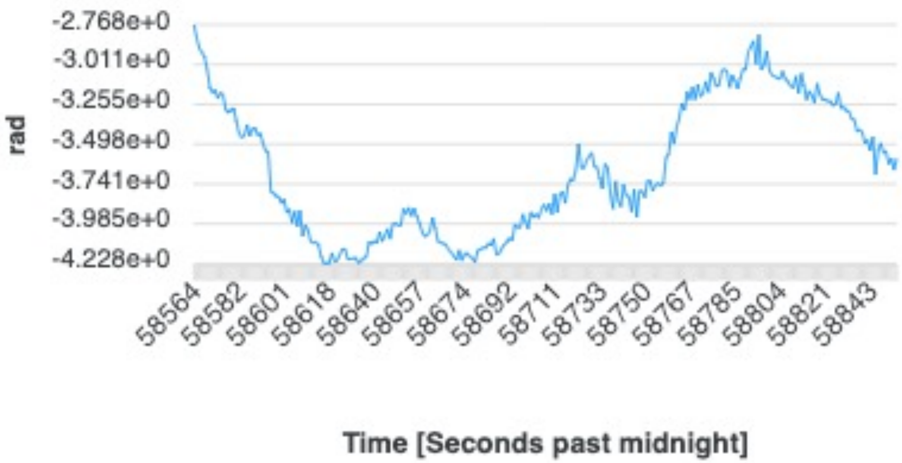
Residual Delay from cross-correlation



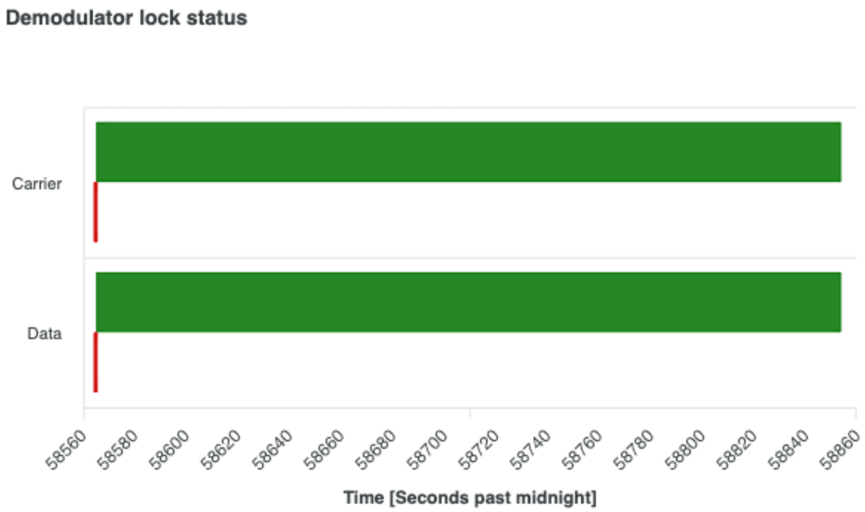
Peak of the cross-correlation function

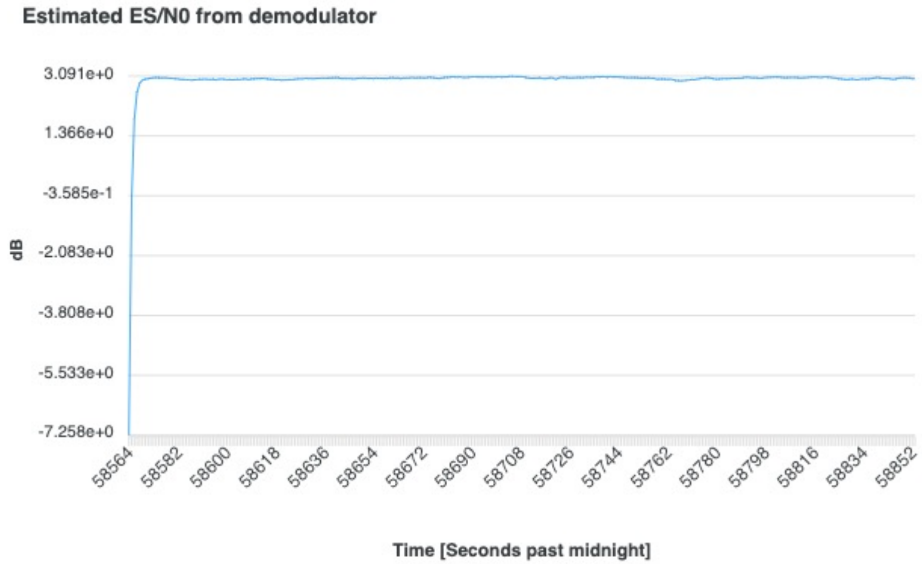


Residual Phase from cross-correlation



Data from Exomars DOY 227 (2)






```
00 1010001111 101 1 10111011 01101011
00 1010001111 110 1 10111100 01100101
00 1010001111 100 1 10111101 11000101
00 1010001111 100 1 10111110 11000110
00 1010001111 110 1 10111111 01100110
00 1010001111 100 1 11000000 11000111
00 1010001111 100 1 11000001 11001000
00 1010001111 110 1 11000010 01100111
00 1010001111 100 1 11000011 11001001
00 1010001111 000 1 11000100 01010100
```

Spacecraft ID: 655

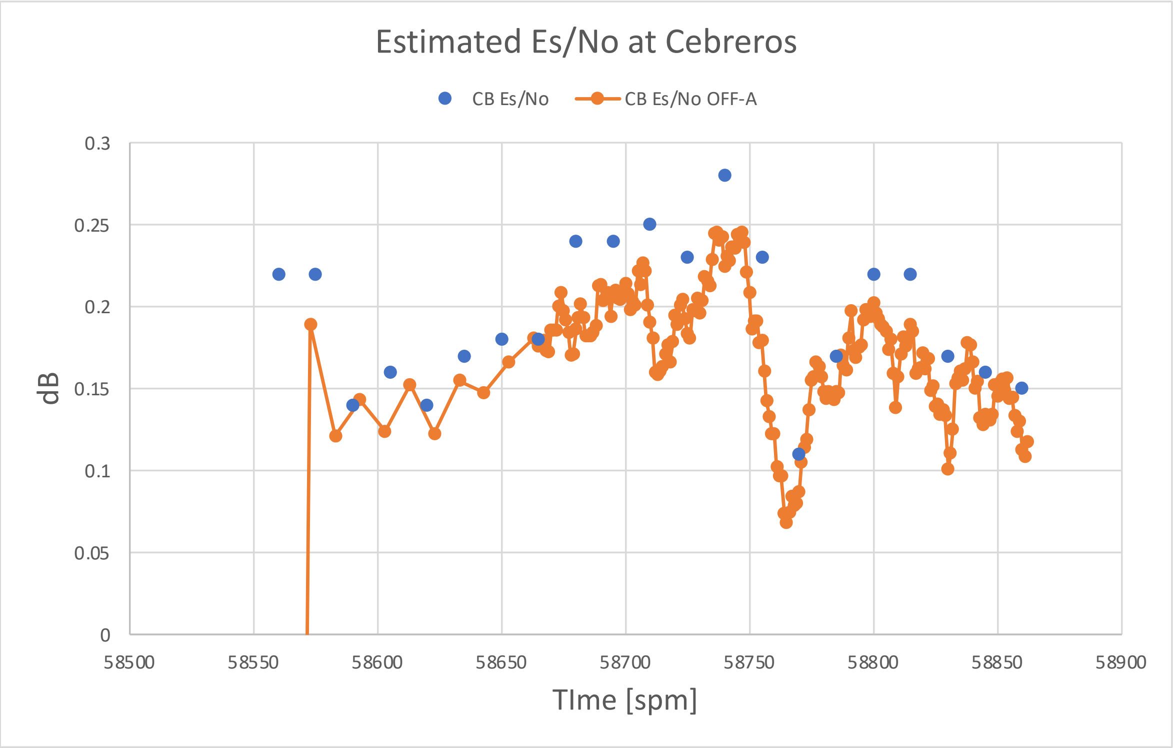
DATA FROM EXOMARS DOY 227 (3)

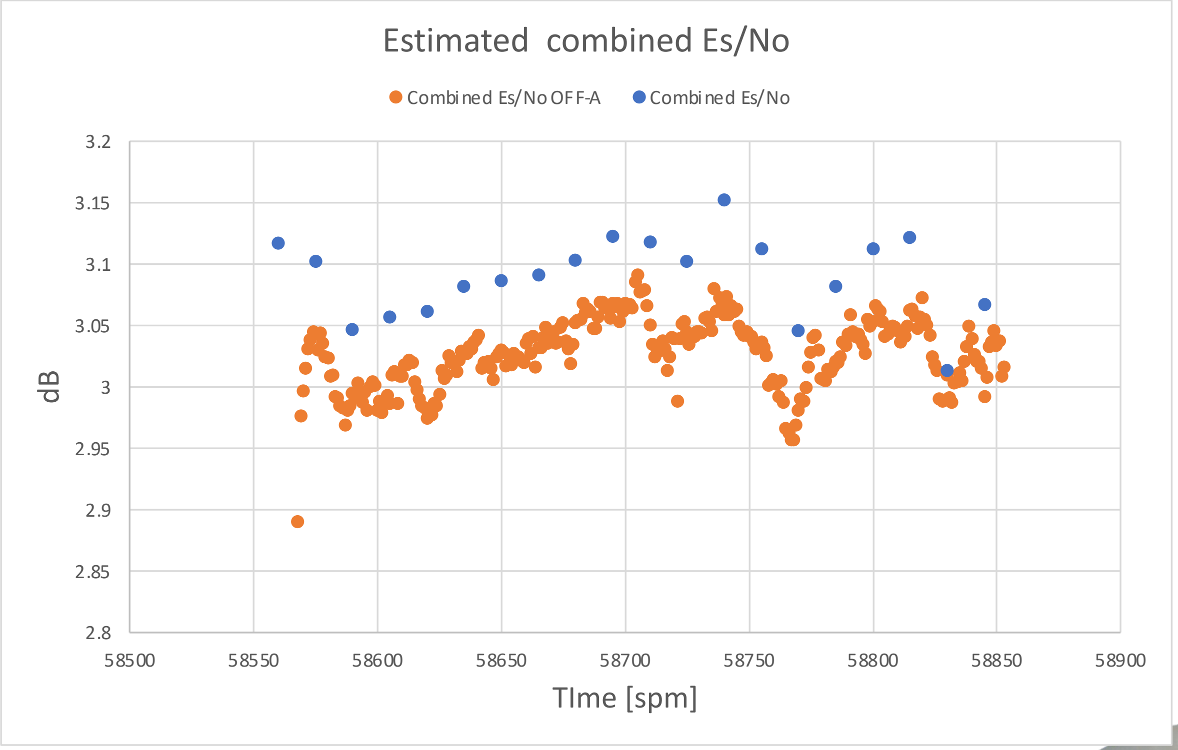
Comparative analysis with TTCP info.

$$\frac{E_S}{N_{0COMB}} = \frac{E_S}{N_{01}} + \frac{E_S}{N_{02}}$$

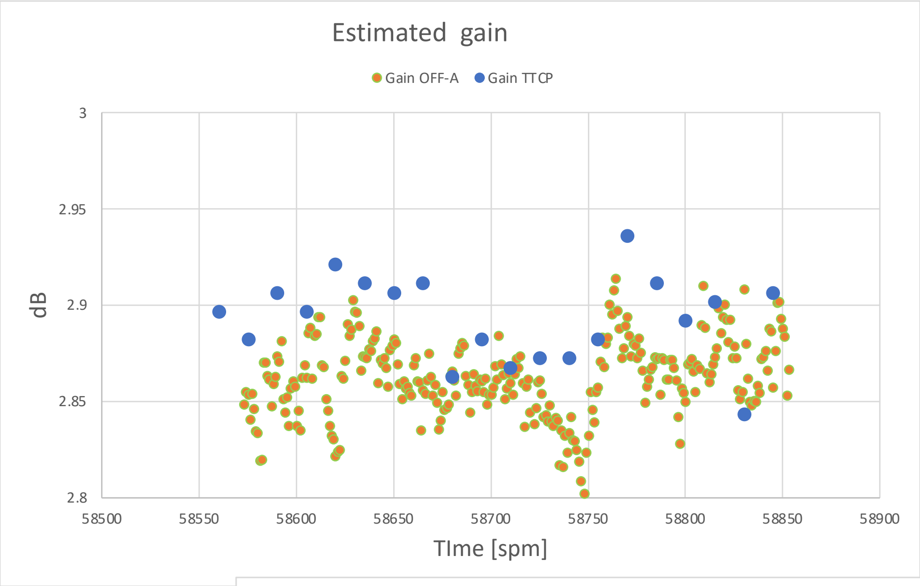
Es/No ➡ combined Es/No, gain, maximum of correlation function.

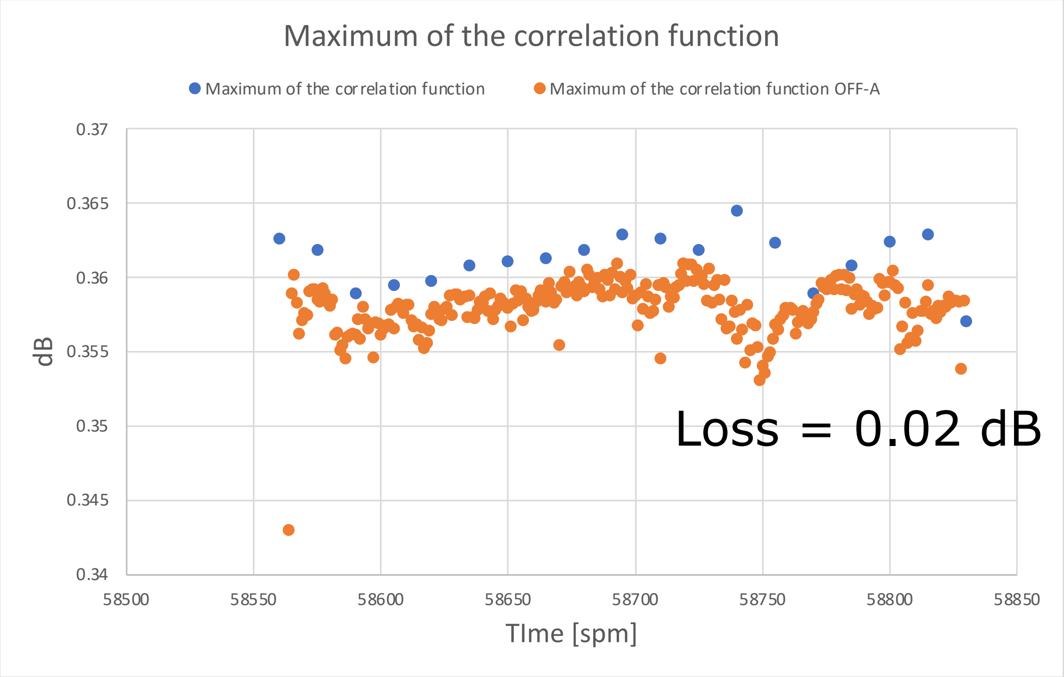
$$\mathrm{SNR} = \frac{E_S}{N_0} \cdot \frac{Symbol\ rate}{Sampling\ rate}$$

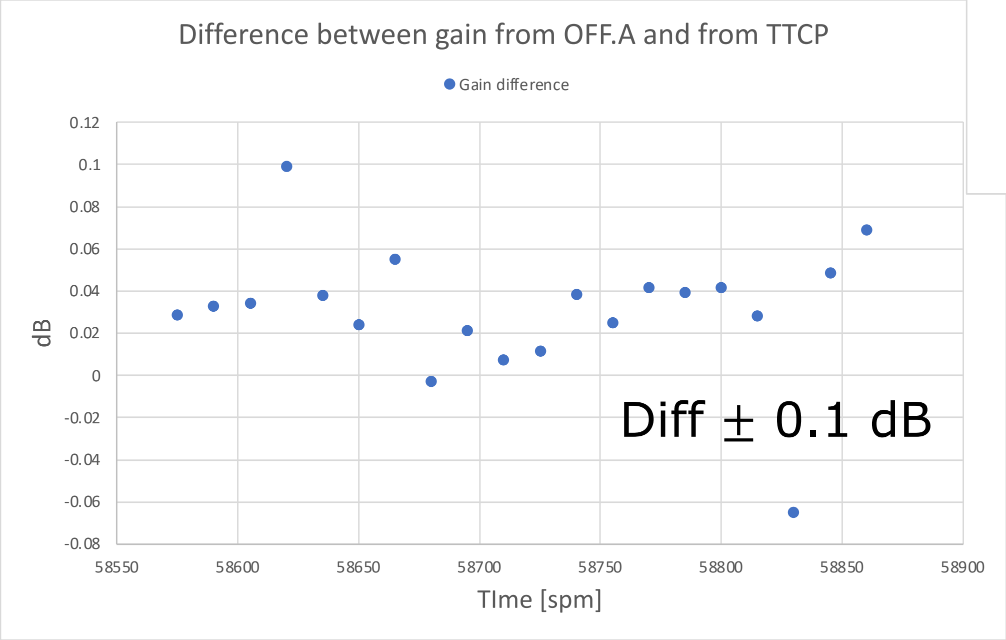




DATA FROM EXOMARS DOY 227 (4)

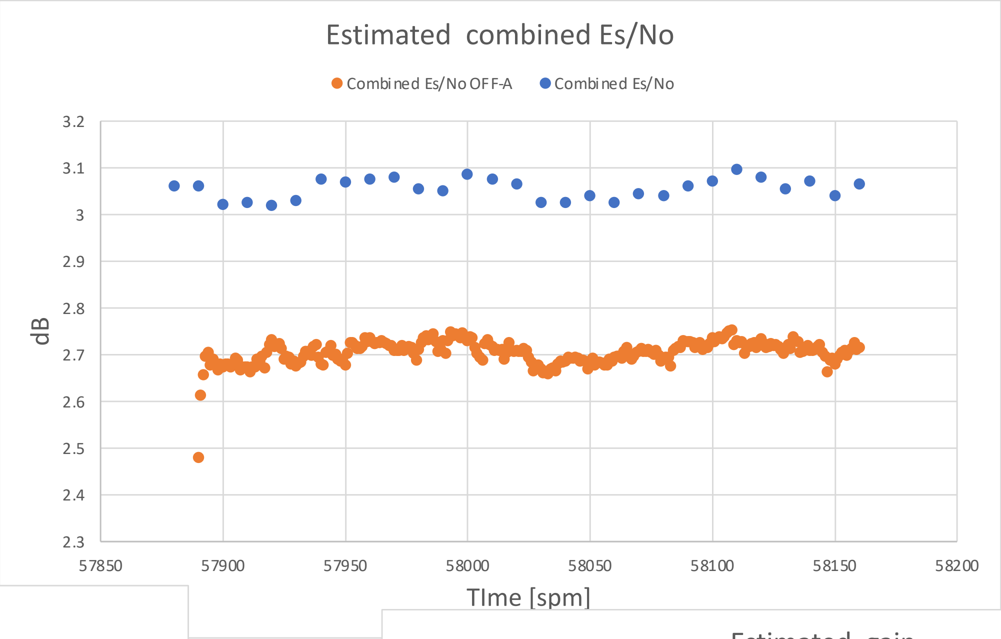


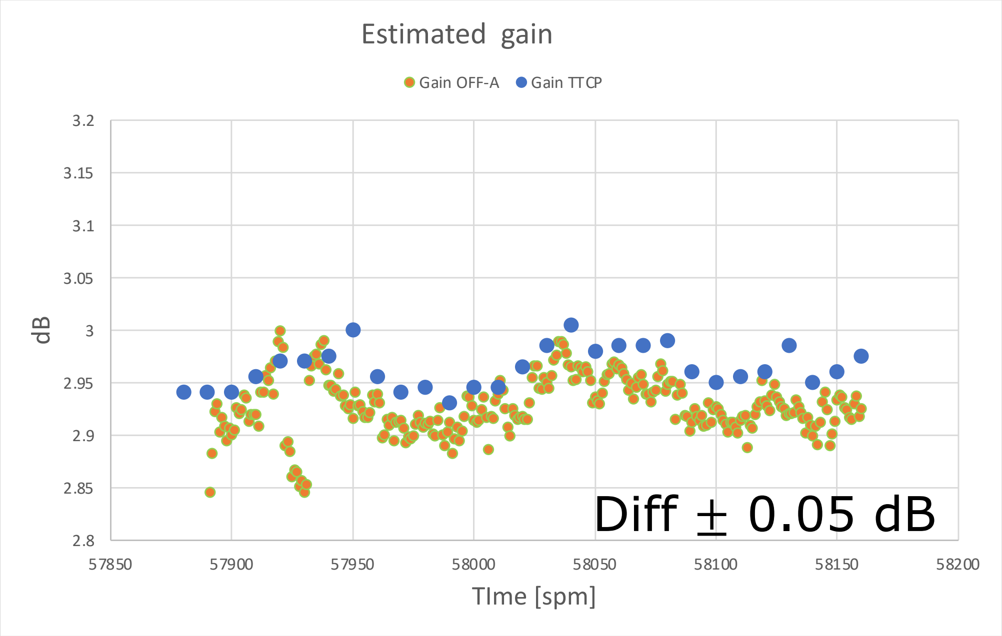


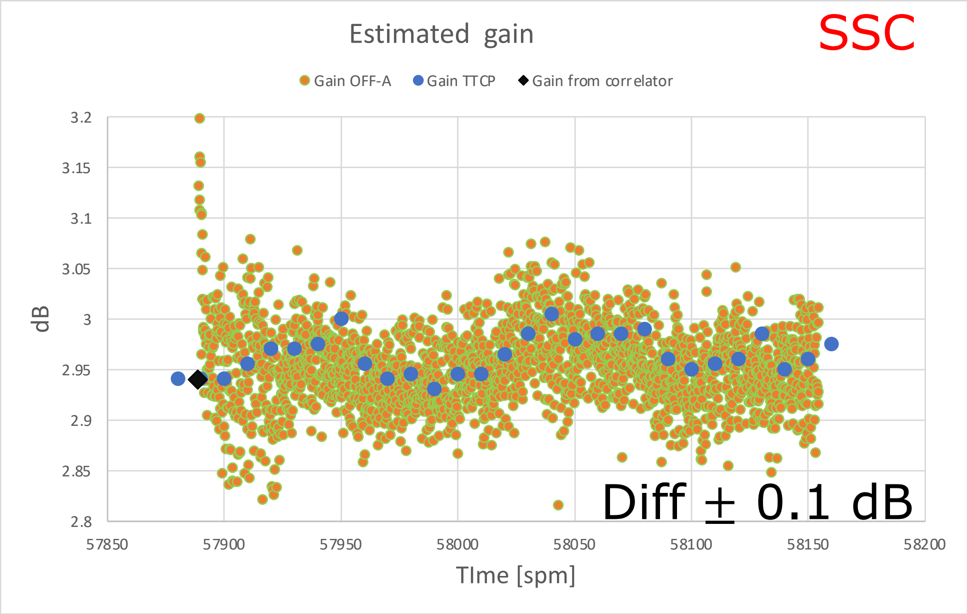


Data from Exomars DOY 230 4 bit

EXMO DOY 227	
Symbol rate	1666.666 ksps
Info rate	415.006 kbps
Sampling rate	3 MHz
Quantization	4 bits
Es/No CB	0 dB (from TTCP)
ES/No MG	0 dB (from TTCP)







Overall OFF-A Performances

Modulation	Mode	Overall loss [dB]	Requirement [dB]	Compliance
GMSK	FSC 8 bits	**<0.13 dB**	0.4	✓
	SSC 8 bits	**< 0.18 dB**	0.4	✓
	FSC 2 bits	**<0.63 dB**	0.9	✓
	SSC 3 bits	**<0.35 dB**	0.9	✓
PCM/BPSK/PM	FSC/SSC	**< 0.4**	0.4	✓
PCM/SP-L	FSC/SSC			

Modulation	Scenario	Elapsed time	Requirement	Check
GMSK FSC	Symbol rate: 40 Msymbols/s Sampling rate: 43.75 MHz	**<16 hours**	5 days	✓
GMSK SSC	Symbol rate: 104 Msymbols/s	**1.8 days**	5 days	✓
PCM/BPSK/PM and PCM/SP-L	FSC PCM/SP-L 1.4 Msps @12 MHz FSC @ < 2MHz and SSC	**0.95 days** **Real-time**	1 day	✓

Considerations

The tool is quite robust for FSC. Some improvements foreseen for SSC.

GMSK

- High rate and power → small integration time, capability of following rapid variations of the phase and good overall performance with FSC.
- Good performance with SSC (slightly worse than FSC)
- No significant gain in processing time with SSC

→ **FSC preferred scheme.**

PCM/SP-L

- Weight estimation with the approach used for GMSK provides good results.
- Integration times with FSC still low enough to follow the variations of the phase
- Similar performance in terms of overall performances between FSC and SSC, but SSC much faster

→ **FSC offers better performance and is real-time for sampling rate < 2 MHz. SSC may be preferred in view of real-time applications.**

PCM/BPSK/PM

- For very low rate (and low S/No) FSC is currently not feasible in OFF-A (implementation can be fixed). In any case, integrating over long times means missing the phase variations over the short-term
- SSC and FSC (sampling rate < 2 MHz) real-time.

→ **SSC preferred scheme.**

FUTURE WORK, FROM PROTOTYPE TO OPERATIONAL

Improvements in the following areas:

- Fixing of FSC for PCM/BPSK/PM with very low rates.
- Optimization of SSC (paradigm of records probably to be discarded).
- Interface with TTCP for SSEF (quantization, scale factor, possible generalization of the SSEF interface).
- Attempt to find a setup that fits all the cases (especially for the demodulator at different rates) partially successful → clear guidelines for different scenarios, also in view of an automation of the tool.
- Guidelines should include be also for acquisition (remnant carrier modulation).
- Improve robustness and reliability of all services (core and infrastructure).
- Test campaign with PCM/BPSK/PM and PCM/SP-L data (following the guidelines for the acquisitions).
- MMI may be improved with other functionalities and refinement of the existing ones (parameters to visualize, report).
- Re-design of the decoder
- Optimization of interface between services (no quantization with 10 Gbps network card, currently useful for very high rate)

New features

- CLI for automation of the tool (with implementation of configuration tables based on guidelines)
- Fault tolerance

FUTURE WORK: EXTENSION TO (UP TO) 4 ANTENNAS

CORRELATION METHOD	PROS	CONS
Simple	• Low complexity • Allows for computation of 2-way observables from demodulator	• Correlation time increases with the number of stations
Sumple	• Allows for reduction of integration times	• Higher complexity • The computation of the 2-way observables must be done after applying the phases and delay computed in the correlation

Integration times for GMSK are of the order of ms. Addition of antennas would have no impact on GMSK performances.

Integration times for PCM/SP-L would still be compatible with variation of delay and phase if more antennas were added. Same is applicable for PCM/BPSK/PM with high rates.

For very low rates, a reduction of the integration time using Sumple if more antenna were added would in any case not be compatible with the variation of phase and delay.

Simple is the preferred scheme.

Future work: from Offline to Real-time

Currently OFF-A is real-time (neglecting latency) for:

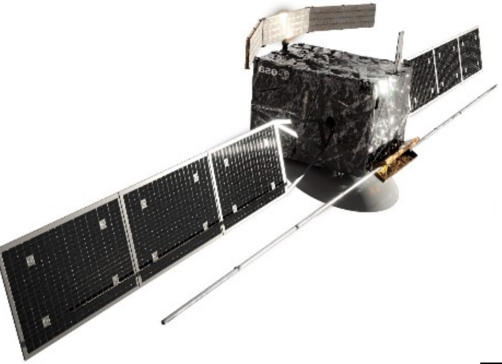
- GMSK @10 MHz
- PCM/SP-L and PCM/BPSK/PM @ 2 MHz

Implementation of a real-time correlator can be done with:

- RDEF files (as for OFF-A): radio science mode (more than one file per pass) must be used to limit latency
- New interface between TTCP and the correlator (I&Q samples). Vita 49 could be used for this purpose.

Changes to the current correlator for a real-time tool include:

- Improvements foreseen for the offline correlator (MMI is probably not needed, CLI instead)
- Possible simplification of correlator
- Change of interface (I&Q samples from TTCP) or new files retrieval/handling (RDEF files)
- Hot redundancy and fault tolerance
- More RAM and machines to reach higher data rate



Thank you for your attention!

Questions?