huld

Techniques for resilient time provision and GNSS spoofer/jammer eraser

Advanced Algorithms and Techniques for Resilient Time Provision (ESA AO/1-10897/21/NL/AS)

Block-Box for an Optimized GNSS Spectrum Monitoring Network Using Artificial Intelligence (ESA AO/1-11383/22/NL/CW)

Ondřej Daniel, ondrej.daniel@huld.io, Huld

United Nations/Finland Workshop on the Applications of GNSS

Helsinki | Finland | 23 - 26 October 2023

huld

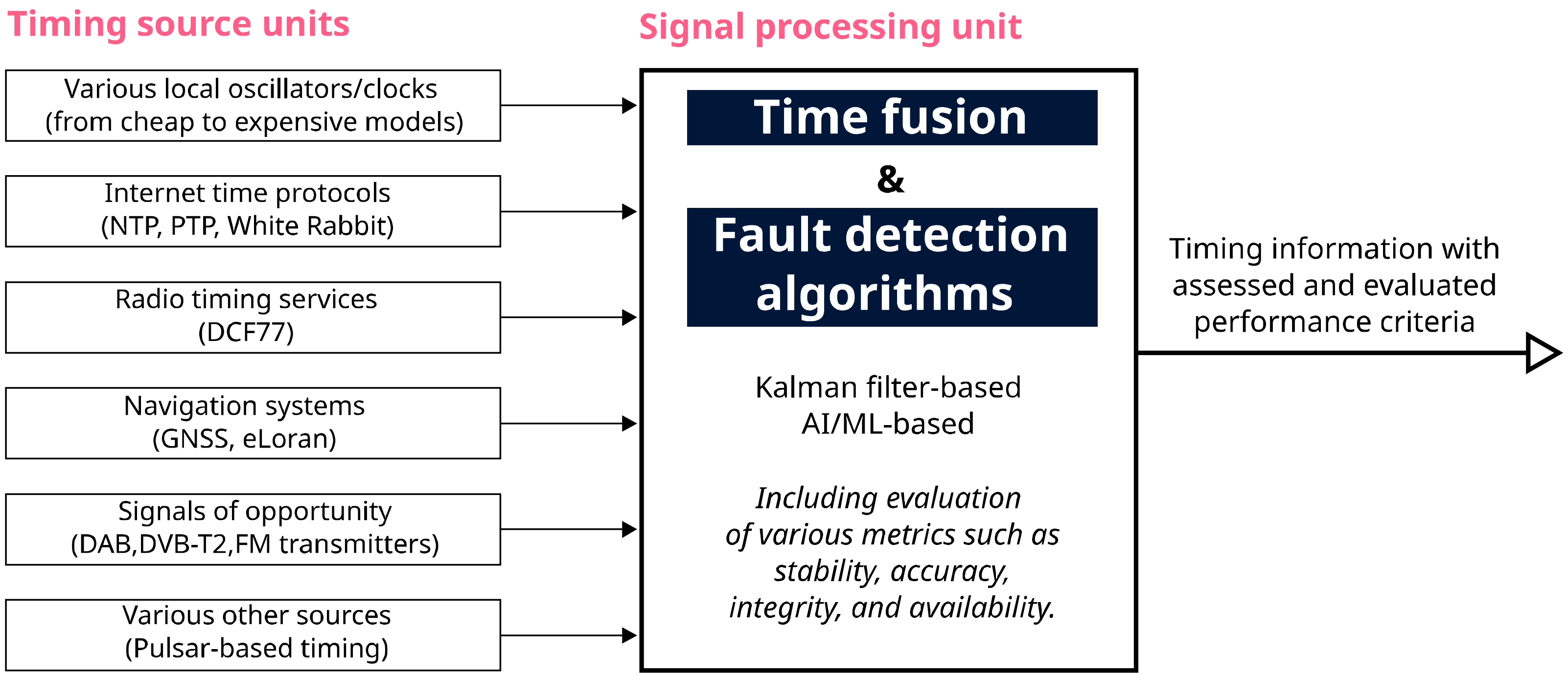
Advanced Algorithms and Techniques for Resilient Time Provision

Huld together with
University of West Bohemia
Czech Republic

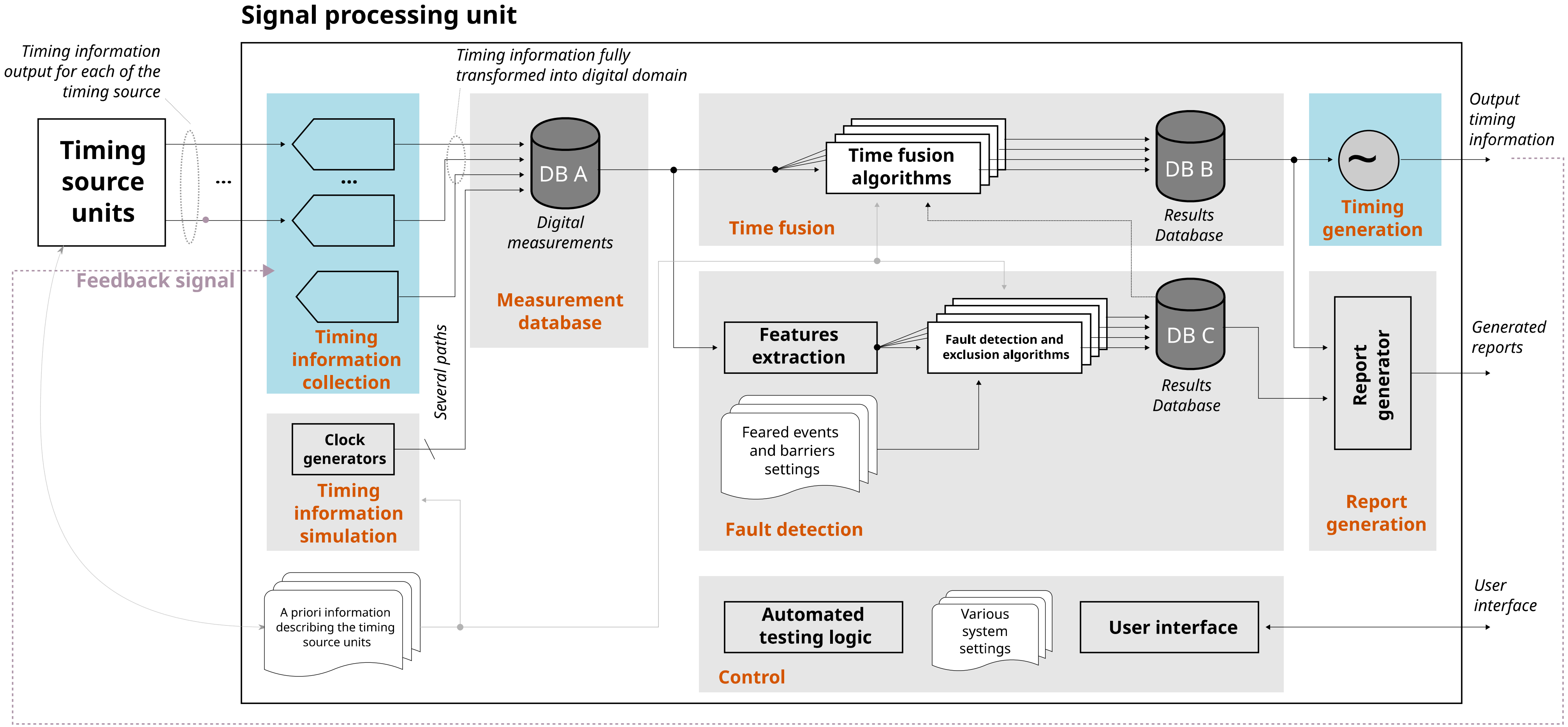
UNIVERSITY OF WEST BOHEMIA

huld

Objectives

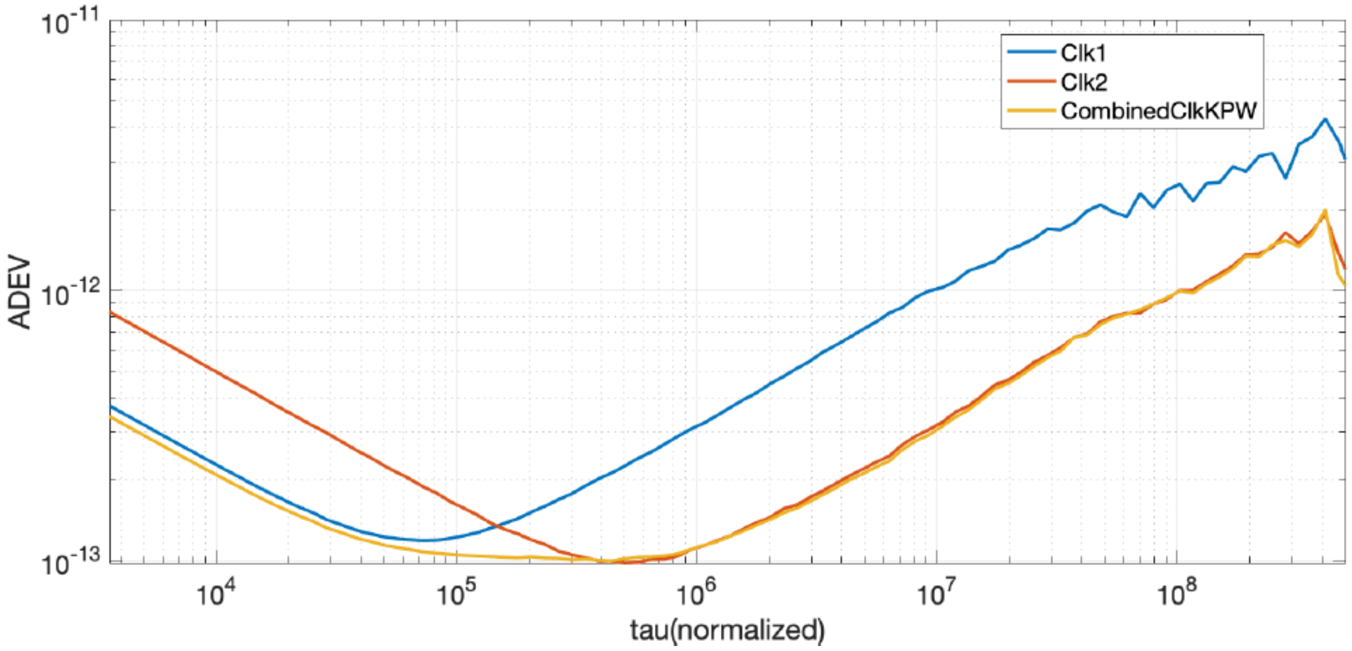


System overview



Clock combination

- Optimal ensemble of clocks leads to better frequency stability, and it is better than any of the individual clocks
- Modern clock combination algorithms are based on Kalman filter (and state-space models)



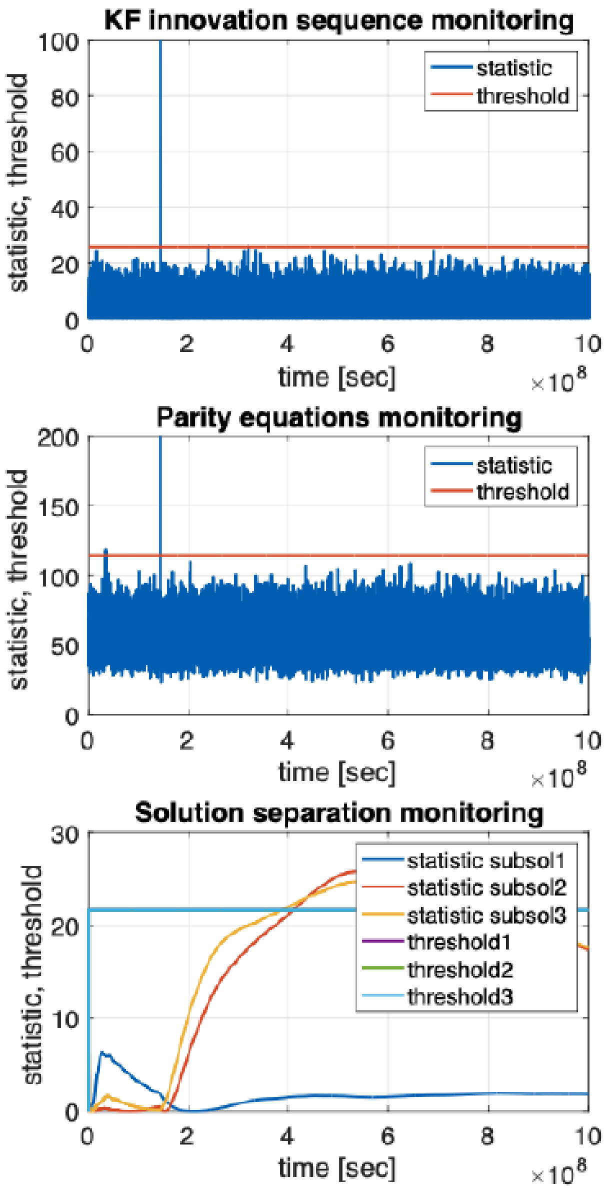
Fault detection

- Reliable clock ensemble requires not only algorithms for clock combination but also algorithms for fault detection of the input clocks
- State-of-the-art literature mentions several signal-based (SB) fault detection techniques, often designed ad-hoc (based on user defined thresholds).
- We have considered additional two groups of fault detection techniques
  - Statistical (model-based)
  - AI/ML (data-based)

Jindrich Duník, Ladislav Král, Ivo Puncochár, Ondrej Straka, Ondrej Daniel, O. Lushchykov: **Fault Detection in Resilient Time Provision.** FUSION 2023: 1-8 hЏld

Fault detection | Statistical tests

- Parity equations
- Kalman filter innovation sequence monitoring
- Solution separation



huld

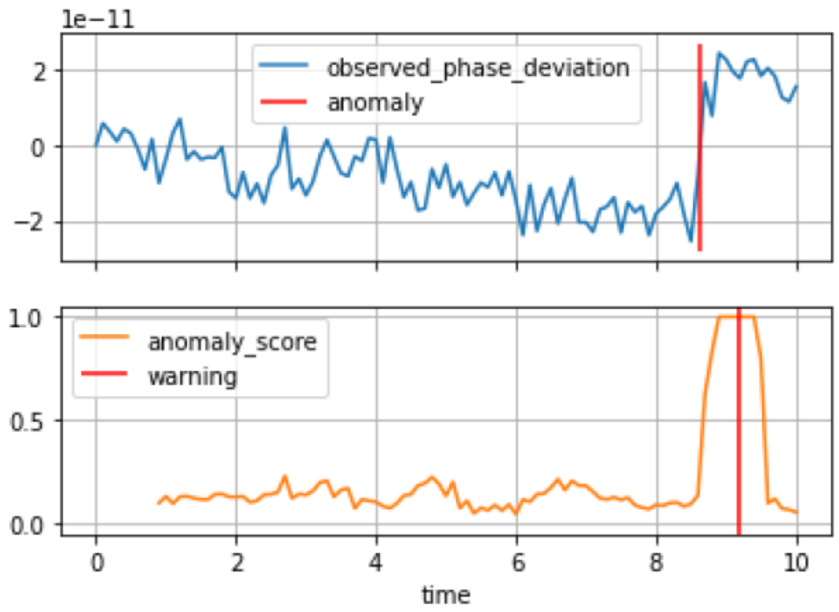
Fault detection | ML-based

Detection ML Algorithms

- *Assumption*: Anomalous (faulty) data are not at disposal for all possible faults
- *Idea*: Train AI/ML model for nominal conditions
- Suitable for abrupt faults
- 10 methods from Python Outlier Detection (PyOD) library

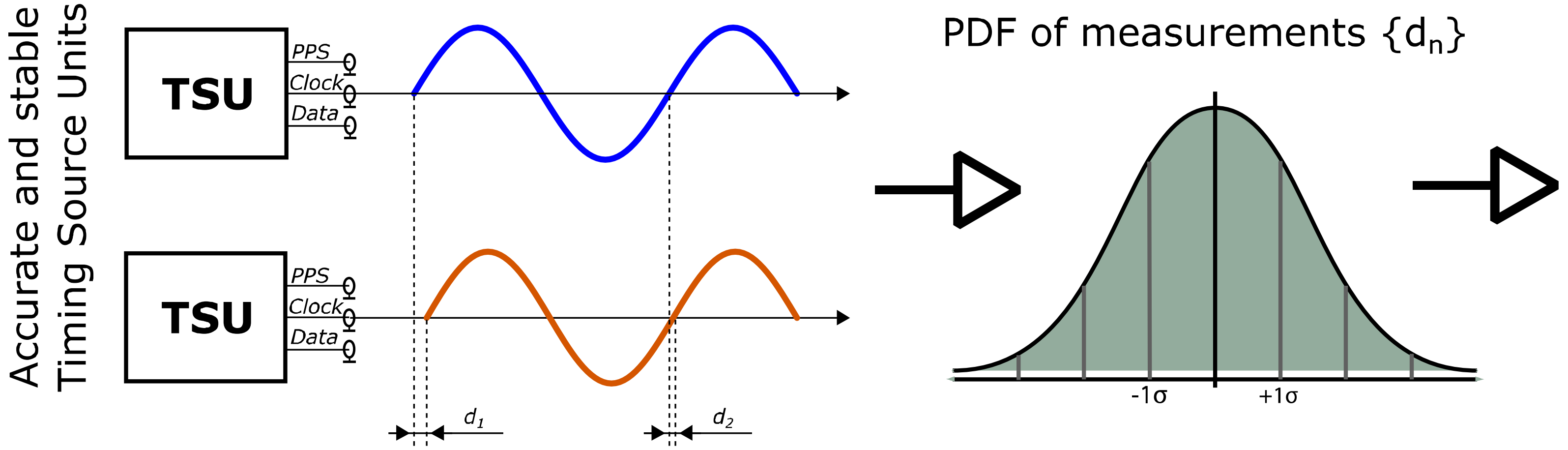
Classification ML Algorithms

- *Assumption*: Anomalous (faulty) data are available for all possible faults
- *Idea*: Train AI/ML model for all faulty conditions
- Sensitive for certain slowly growing errors
- 10 methods from Python Scikit-learn library



Accuracy requirements

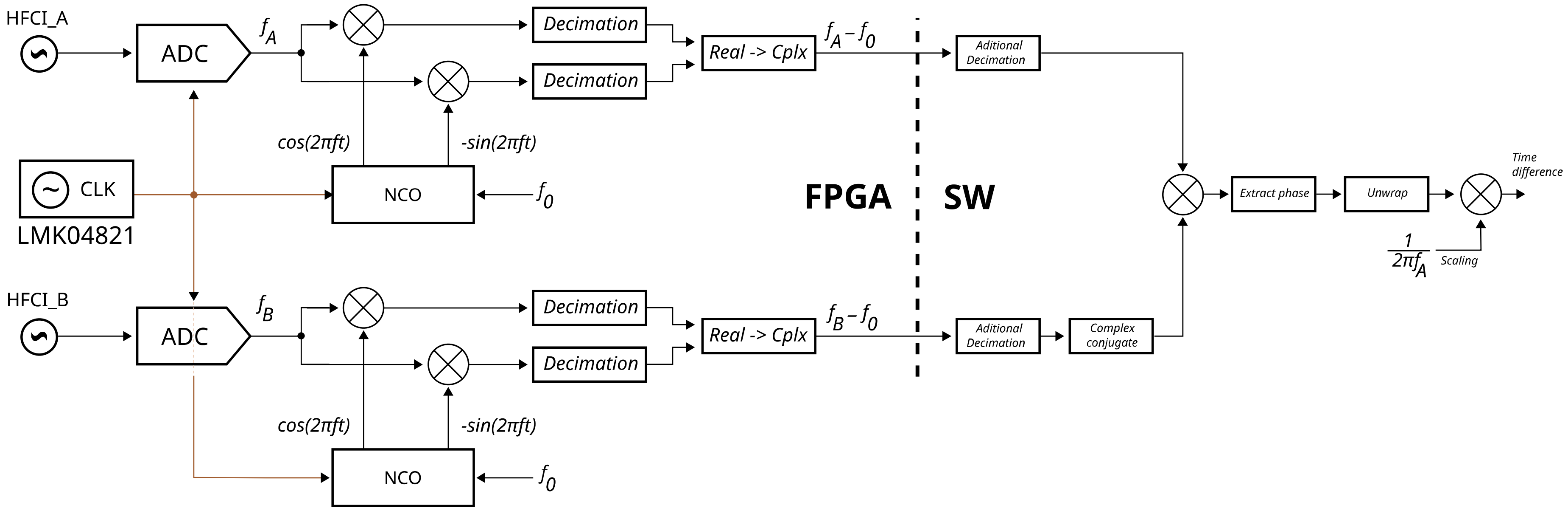
- The developed system shall be useful for generating clock ensemble based on atomic clocks



+1σ shall be smaller than 1 ps

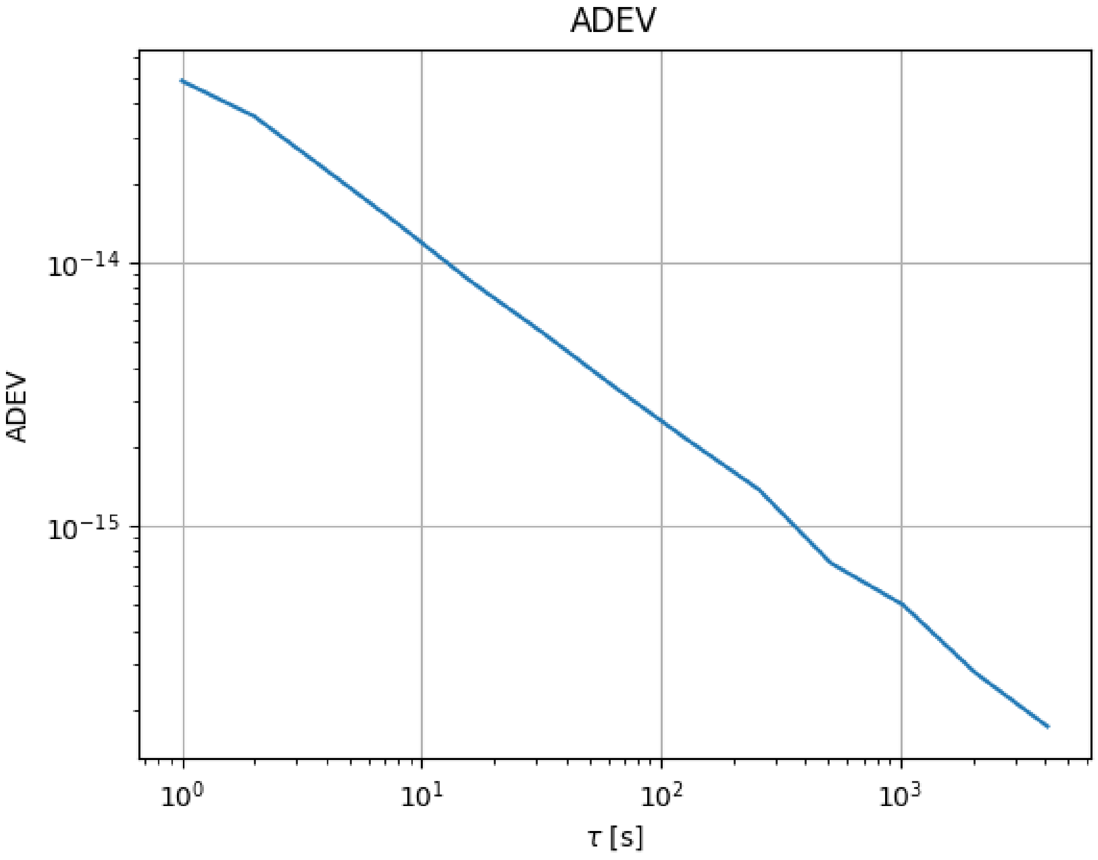
Measurements technique

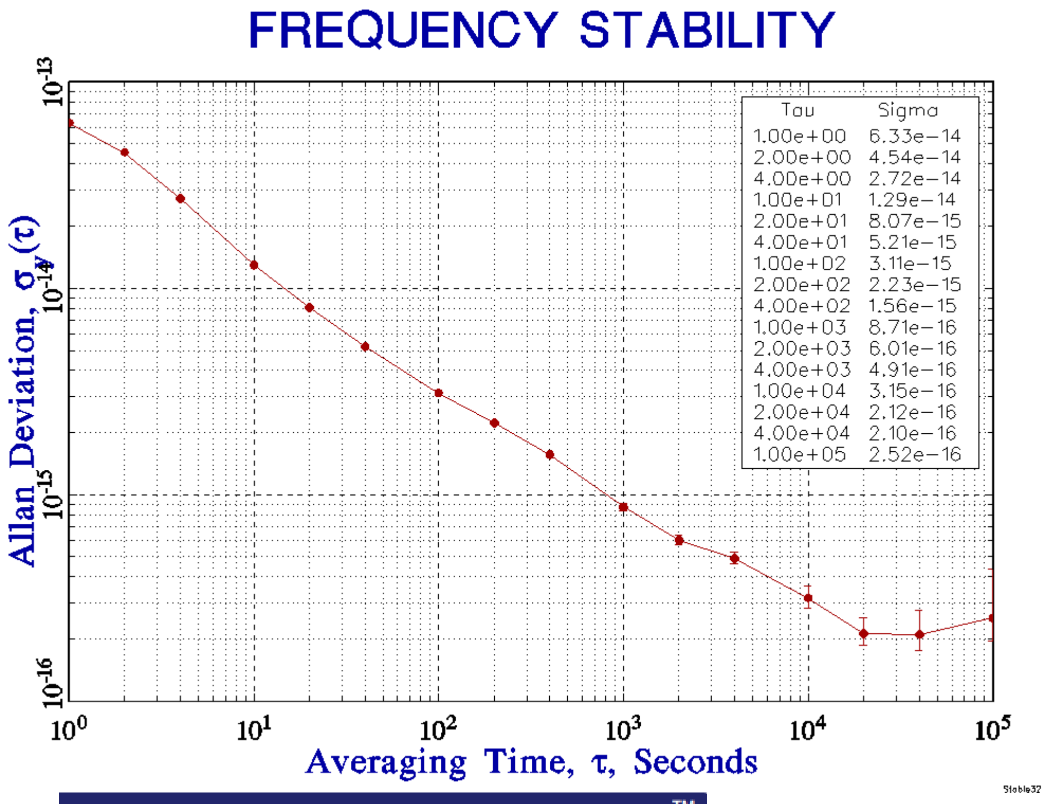
- Dual Mixer Time Difference (DMTD) method
- **Direct analog to digital conversion**



huld

Preliminary characterization



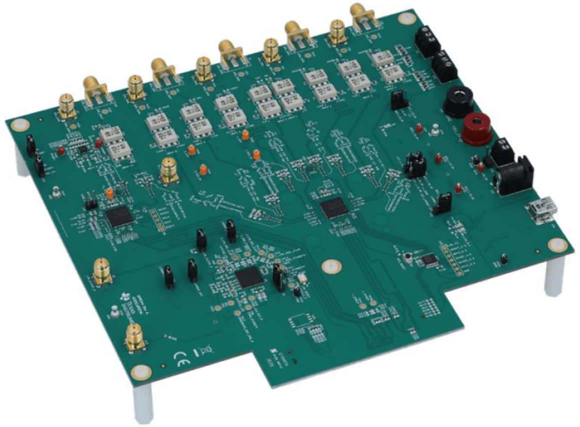


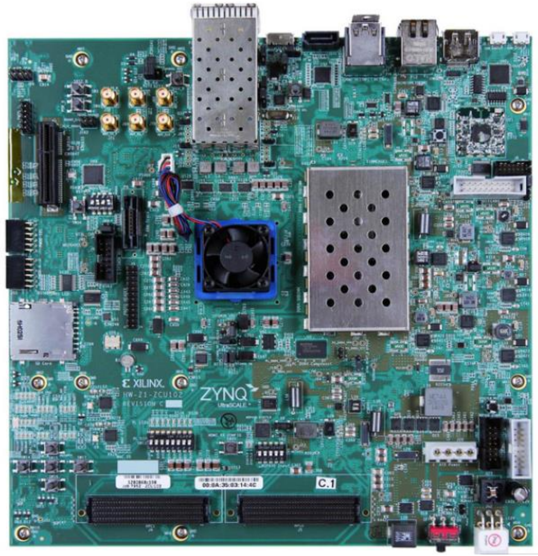
iMaser 3000™

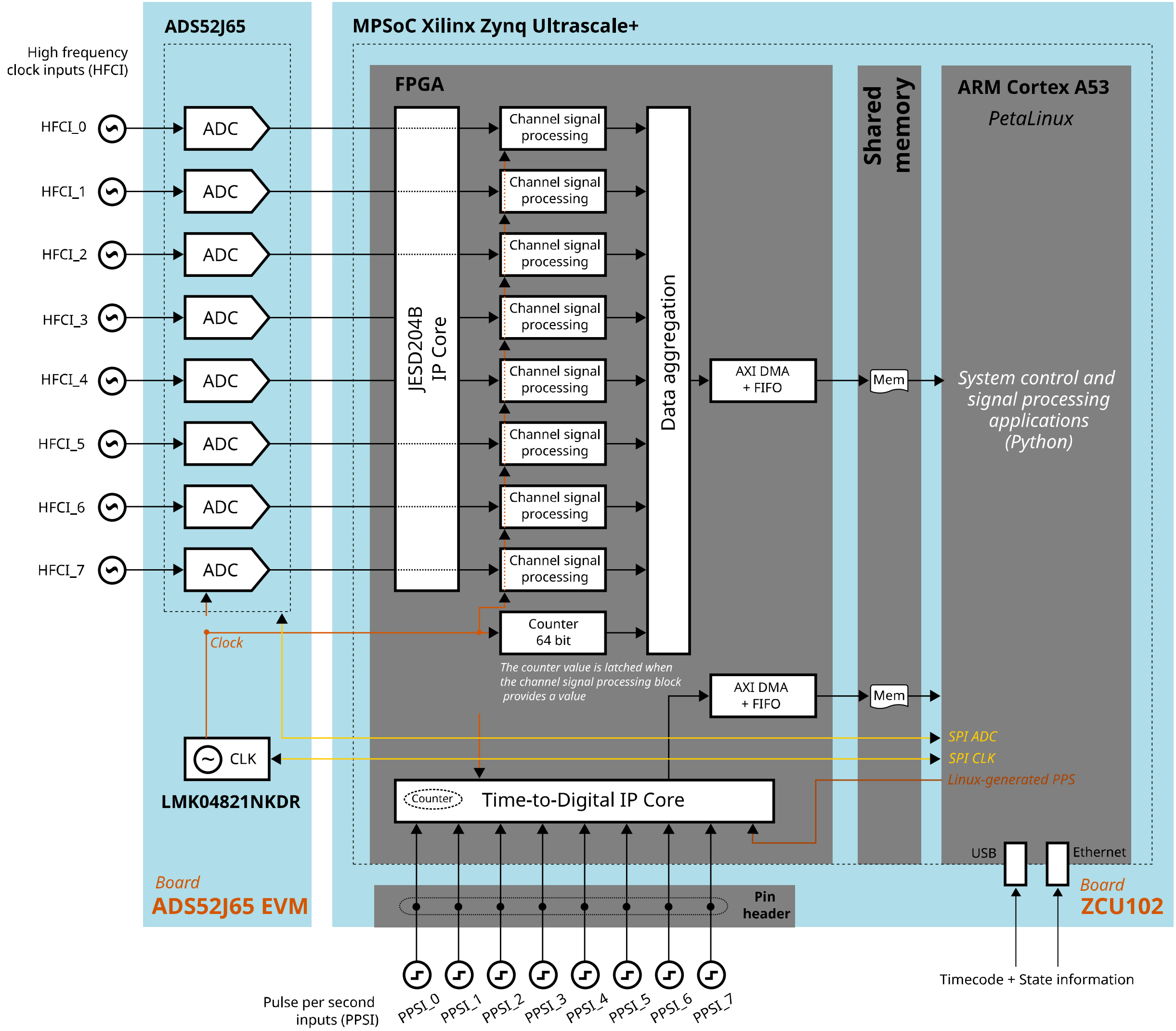
SMART ACTIVE HYDROGEN MASER CLOCK

huld

System





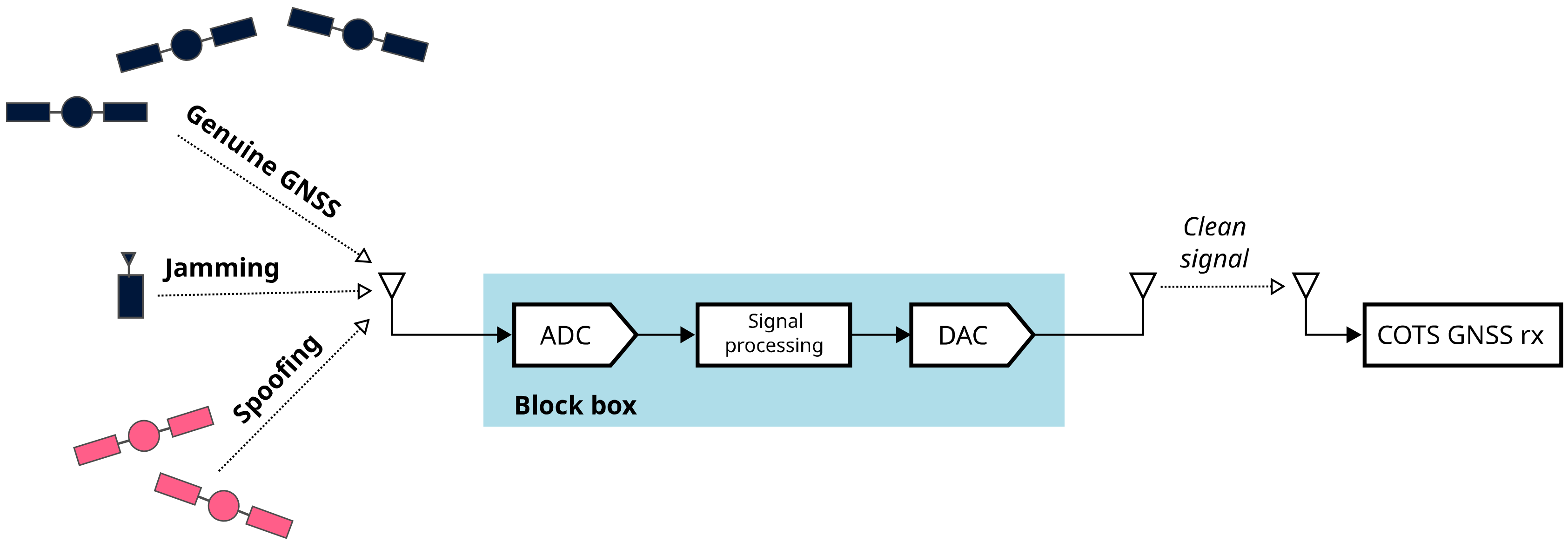

ADS52J65
MPSoC Xilinx Zynq Ultrascale+
High frequency clock inputs (HFCI)
HFCI_0
HFCI_1
HFCI_2
HFCI_3
HFCI_4
HFCI_5
HFCI_6
HFCI_7
ADC
FPGA
JESD204B IP Core
Channel signal processing
Data aggregation
Counter 64 bit
The counter value is latched when the channel signal processing block provides a value
AXI DMA + FIFO
Shared memory
Mem
ARM Cortex A53
PetaLinux
System control and signal processing applications (Python)
Clock
CLK
LMK04821NKDR
SPI ADC
SPI CLK
Linux-generated PPS
Counter
Time-to-Digital IP Core
USB
Ethernet
Board
ADS52J65 EVM
Board
ZCU102
Pin header
Pulse per second inputs (PPSI)
PPSI_0
PPSI_1
PPSI_2
PPSI_3
PPSI_4
PPSI_5
PPSI_6
PPSI_7
Timecode + State information

Block-Box for an Optimized GNSS Spectrum Monitoring Network Using Artificial Intelligence

huld

Objectives

- System capable of detecting and mitigating GNSS jamming and spoofing using state-of-the-art AI/ML techniques
- SW and HW demonstration (Python, FPGA)

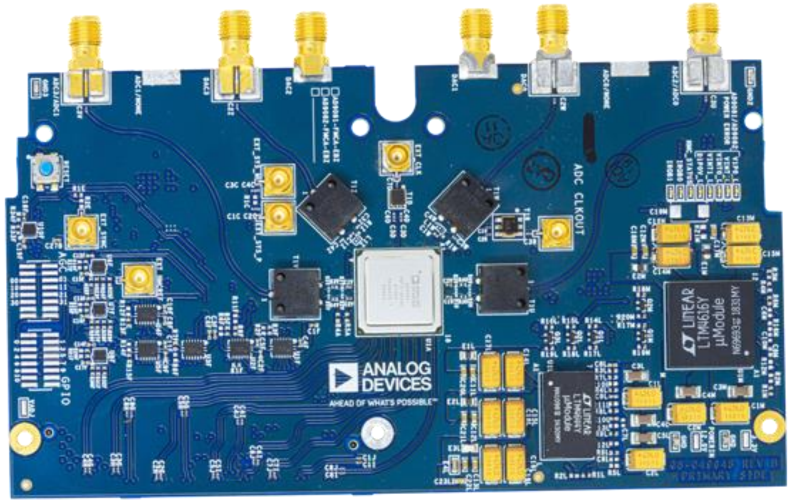


Scope

- Definition of threats, detection metrics and mitigation techniques for GNSS signals
- FPGA as development and validation HW platform (Xilinx Zynq, AD MxFE 9082)
- Execution of AI algorithms, DSP, mitigation, signal retransmission, … in real-time in HW
- Supported frequency bands: E1, E5, E6
- Cloud SW as backend application allowing customized training and control

AD9082

MxFE Quad, 16-Bit, 12 GSPS RF DAC and Dual, 12-Bit, 6 GSPS RF ADC



huld

Development approach

- Preliminary design of **anti-jamming** algorithms based on simulated data
- Preliminary design of **anti-spoofing** techniques based on available spoofing data sets:
  - **OAKBAT** (Oak Ridge Spoofing and Interference Test Battery)
  - **TEXBAT** (Texas Spoofing Test Battery)
- After this preliminary stage, the real signals are used

Machine learning

- Classification
- Segmentation
- Reinforcement learning
- Training with various inputs
  - Spectrogram, AGC, Correlation function, PVT
- Finding dependence between clean signal and jammed/spoofed signal

huld

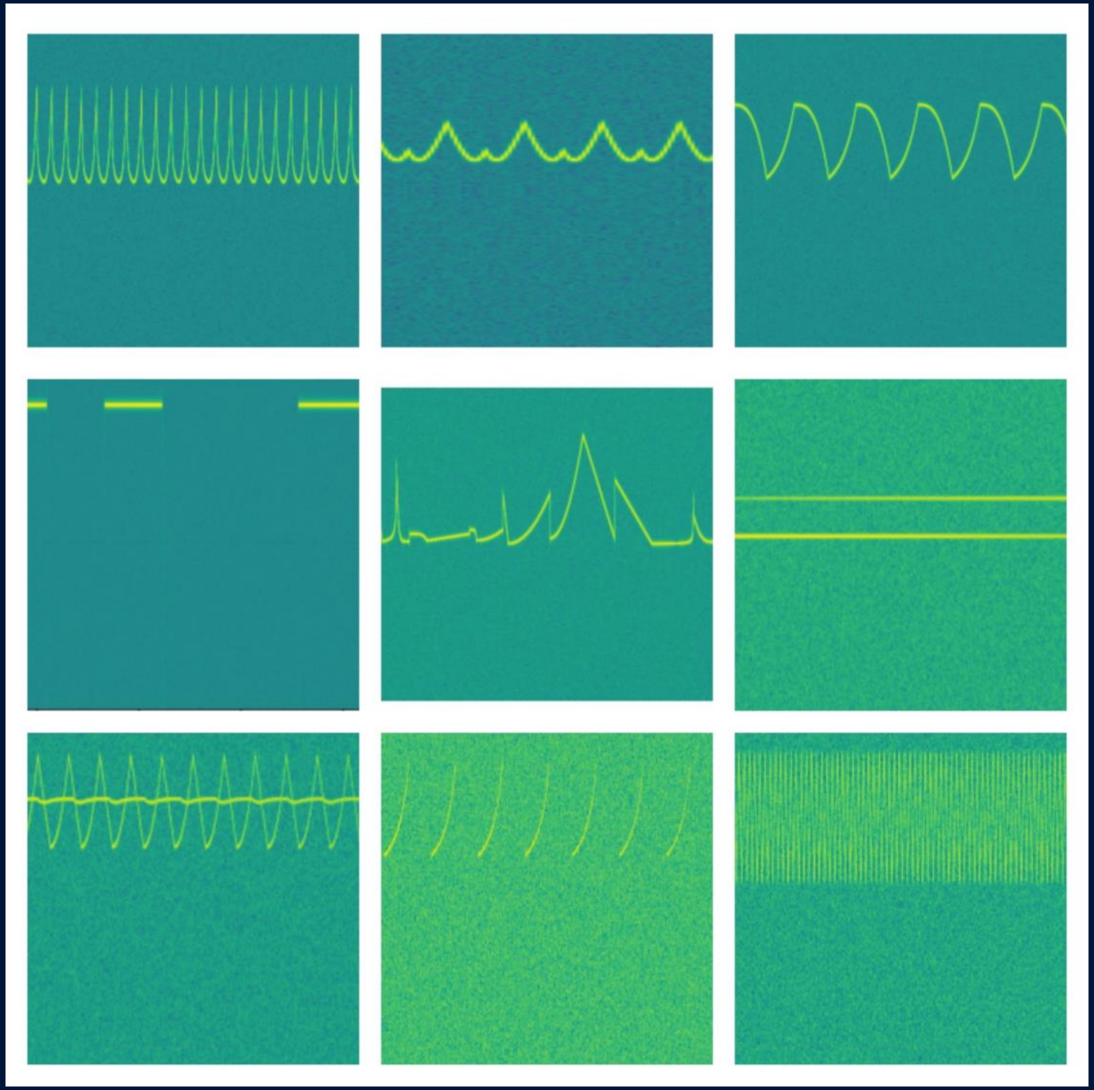
Jamming

Types of jamming

- Chirp
- Pulse
- Single tone
- Constant or variable amplitude
- Combination of above
- Multiple jammers

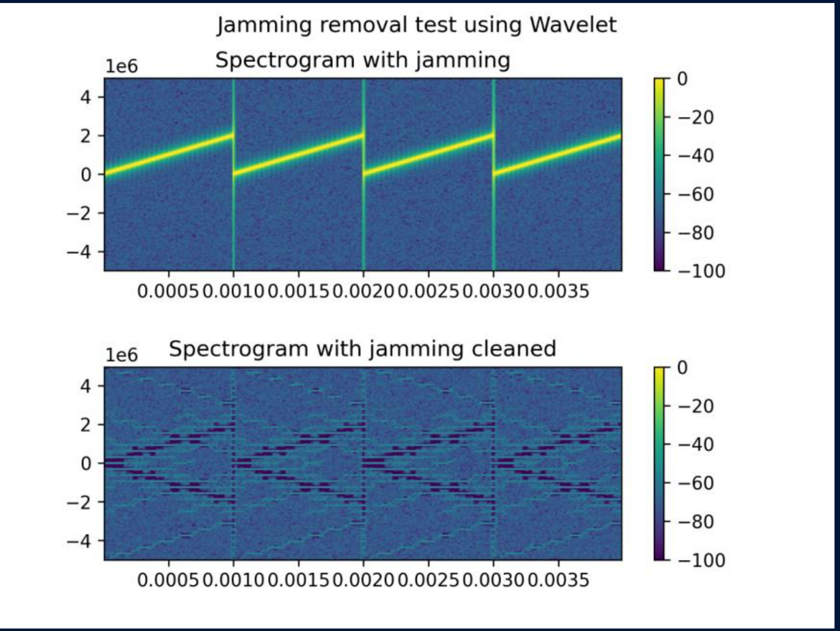
AI input

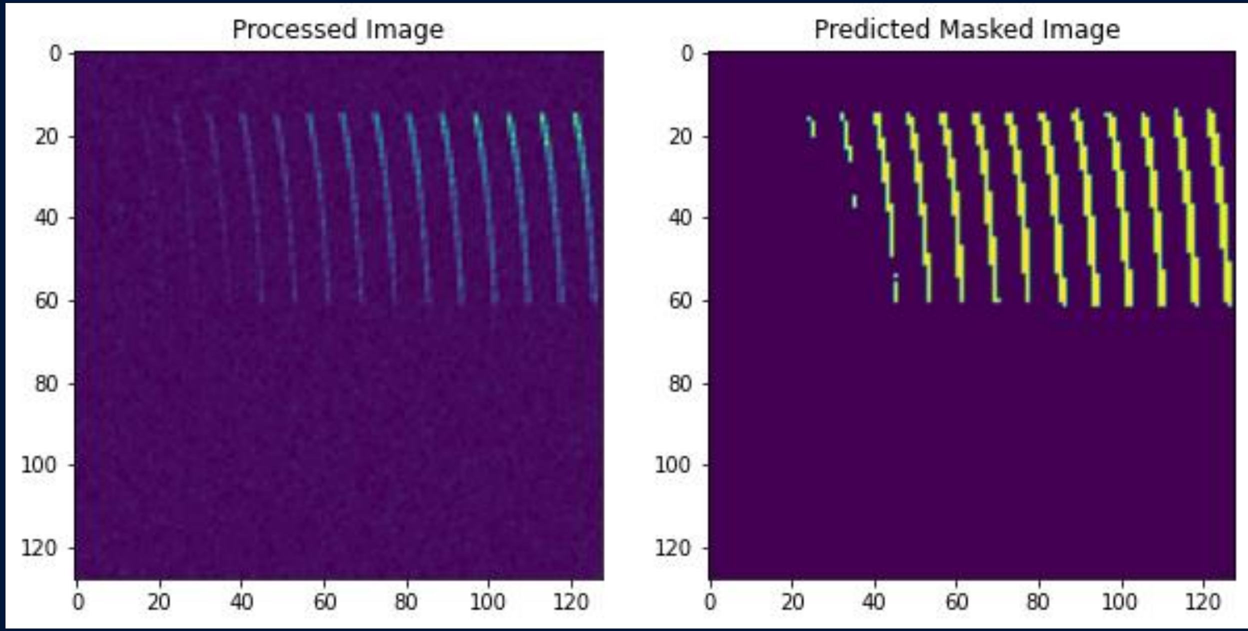
- Spectrogram
- ADC histogram



Jamming mitigation

- DSP with AI-derived parameters
- FDAF (Frequency Domain Adaptive Filtering)
- STFT or Wavelet transformation





huld

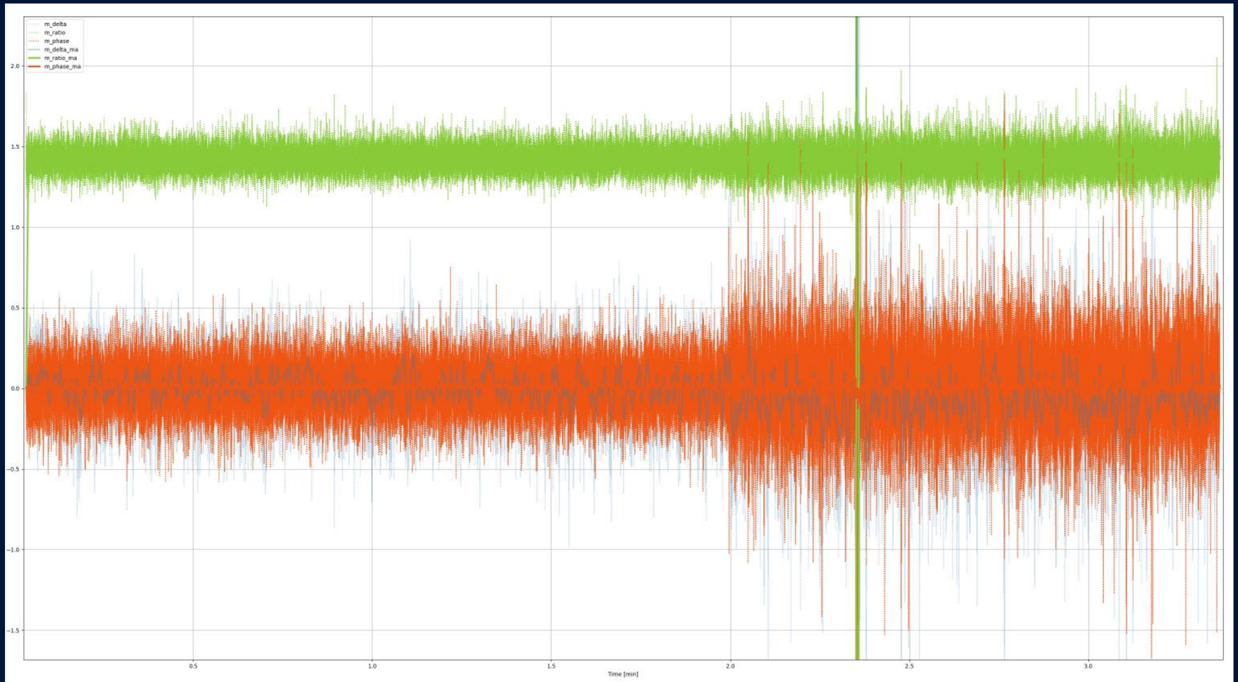
Spoofing

AI spoofing detection based on

- event (change of parameters)
- state (don’t need to catch start of spoofing)

AI input

- Correlation function
- C/N_0
- SQM (Signal quality metric)
  - Delta, Ratio, early/late phase metrics



- Delta Metric

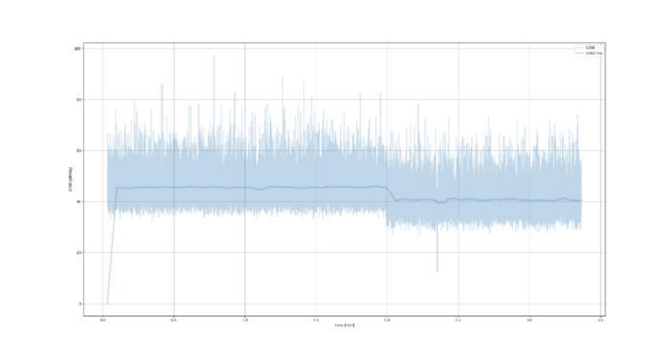
$$m_{delta} = \frac{I_{-d} - I_{+d}}{I_p} \quad (6)$$

- Ratio Metric

$$m_{ratio} = \frac{I_{-d} + I_{+d}}{I_p} \quad (7)$$

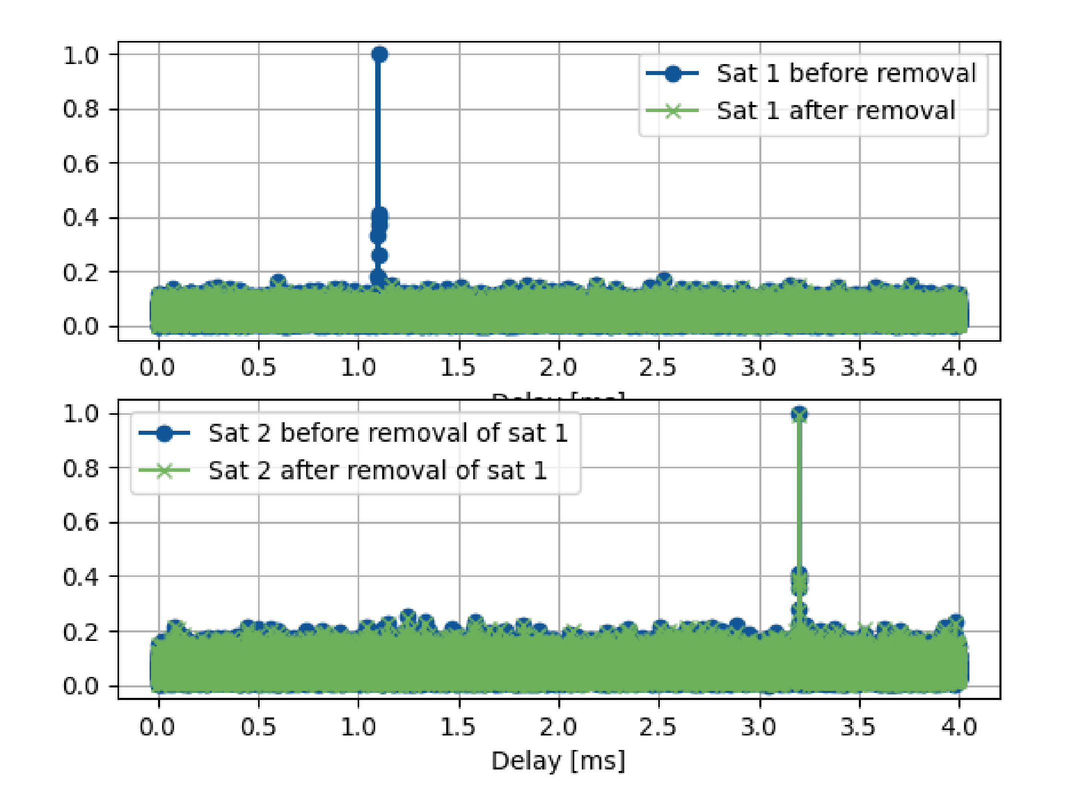
- Early Late Phase Metric

$$m_{elp} = \tan^{-1}\left(\frac{Q_{-d}}{I_{-d}}\right) - \tan^{-1}\left(\frac{Q_{+d}}{I_{+d}}\right) \quad (8)$$



Spoofing mitigation

- Classical tracking of the signal to be mitigated
- Subtracting of the tracked component from the stream of baseband samples



huld

Beyond tomorrow