ESA ESTEC
Keplerlaan 1
2201 AZ Noordwijk
The Netherlands

SENTINEL-1 IMPROVED AIS PROCESSOR

Prepared by	Stefan Graham
Document Type	TN - Technical Note
Reference	S1CD-TN-ESA-PM00-0763
Issue/Revision	0 . 0
Date of Issue	14/07/2025
Status	Draft

APPROVAL

Title	Sentinel-1 Improved AIS Processor		
Issue Number	0	Revision Number	0
Author	Stefan Graham	Date	14/07/2025
Approved By		Date of Approval	

CHANGE LOG

Reason for change	Issue Nr	Revision Number	Date

CHANGE RECORD

Issue Number 0	Revision Number 0		
Reason for change	Date	Pages	Paragraph(s)

DISTRIBUTION

Name/Organisational Unit

Table of Contents

1. INTRODUCTION

The following document outlines the usage, algorithmic baseline and interfaces for the Sentinel-1 Improved AIS Processor developed at ESA-ESTEC. The algorithms are intended to be implemented in the EOF Copernicus Ground Segment and to replace the current MATLAB AIS processor currently in use. The algorithms are licensed under the Apache License 2.0.

2. REFERENCE DOCUMENTS

[RD1] S1CD-DD-TAI-IN02-0001 AIS P/L Design Description & Performance Report

[RD2] S1CD-RD-TAI-PM18-0123 Sentinel-1 AIS payload Packet Format

[RD3] G. Colavolpe, T. Foggi, A. Ugolini, J. Lizarraga, S. Cioni, and A. Ginesi, A highly efficient receiver for satellite-based Automatic Identification System signal detection, 2014.

[RD4] S1CD-RP-ESA-SY00-0730 Sentinel-1 In-orbit Commissioning Report

[RD5] S1CD-DD-ESA SY00-0494 AIS Processing Algorithms

[RD6] S1CD-DD-ESA-PL00-0518 Confidential Annex: AIS Processing Algorithms

[RD7] S1CD-RS-TAI-SC01-0047 AIS Processing Technical Note

[RD8] Barbieri A, Colavolpe G. Simplified soft-output detection of CPM signals over coherent and phase noise channels, IEEE Trans. Wireless Commun. 2007; 6:2486–2496

[RD9] U. Mengali, M. Morelli, Data-aided frequency estimation for burst digital transmission, IEEE Trans. Commun., vol. 45, pp. 23-25, Jan. 1997.

[RD10] M. Morelli and U. Mengali, Joint frequency and timing recovery for MSK-type modulation, IEEE Trans. Commun., vol. 47, pp. 938-946, June 1999.

[RD11] *Receiving Method and Receiver for Timing and Frequency Offset Correction of Continuous Phase Demodulation in Satellite-Based Automatic Identification Systems* (PCT/EP2014/051273).

3. USER MANUAL

This section describes the usage, installation and compilation for the Sentinel-1 AIS processing algorithms.

3.1. End-to-end AIS Processor

Figure 1 shows the high-level overview of the AIS processing chain with the interfaces for each step. The processor takes as input the Sentinel-1 AIS Instrument Source Packets (ISPs) contained in the .dat file within the SAFE file format. The ISPs are extracted using Python and the raw IQ samples for each channel are exported as .wav files for use by the ESA AIS receiver executable. The ESA AIS receiver performs the demodulation of the raw data to produce the AIS binary decode messages. The decoded AIS messages are then passed back to Python or the AIS Message Parsing step. A more detailed overview of each step of the processing can be found in Section 4.

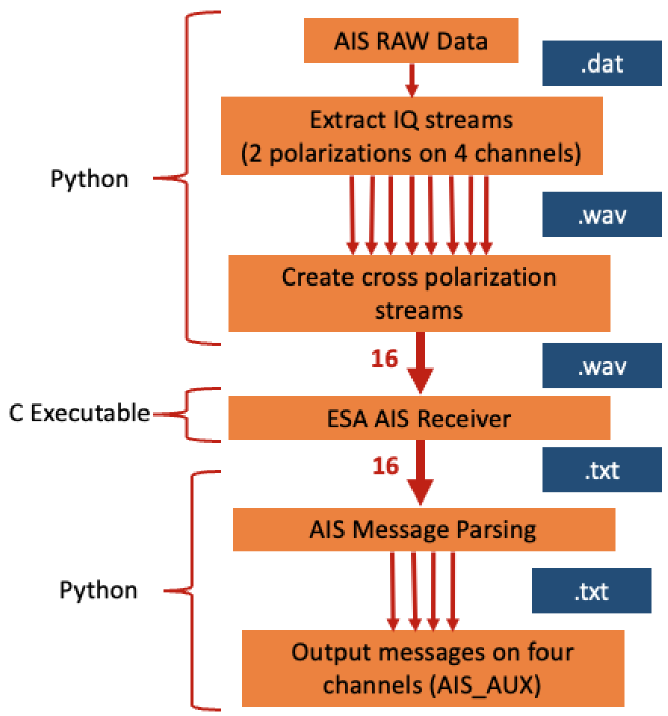

Figure 1 Interface diagram for the Sentinel-1 AIS Processing.

Usage

The main script, `process_s1_ais_raw.py`, performs the end-to-end AIS processing chain of Figure 1. The Raw AIS data is extracted and demodulated for all frequency channels and polarizations, including linear combinations at ±45 degrees.

To run the main Python script the following command structure is used:

```
python process_s1_ais_raw.py <input_folder> -o <output_folder>
```

<input_folder>: Path to the Sentinel-1 AIS SAFE folder containing the binary `.dat` file. <output_folder>: Path where the processed output files will be saved.

Installation

To install the repository locally execute the following:

- Clone the repository: `git clone https://github.com/esa/sentinel1-ais-processor`
- Navigate to the folder: `cd sentinel1-ais-processor`
- Install dependencies: `pip install -r requirements.txt`

Example test case

The `/Testcase/` folder contains a simulated AIS dataset in Sentinel-1 ISP format.

To run the script on this test data, use:

```
python process_s1_ais_raw.py
./Testcase/S1C_AI_RAW__0____20220531T155630_20220531T155744_000016________2D48.SAF
E/ -o ./Testcase/output
```

The testcase can be used to verify that the processor has been correctly installed. The expected output can be found in the directory `/Testcase/output`. Note: for the specific testcase no detections are expected to be demodulated from the H-V input channel.

Output Files

- Extracted ISP headers and de-interleaved raw data (.wav).
- Decoded AIS messages, named: `S1C_AI_L1_YYYYMMDDTHHMM_YYYYMMDDTHHMM.txt`.
- Binary detections from each AIS channel.
- Summary report with statistics for each input stream.
- List of messages detected with an invalid message type.

3.2. ESA AIS Receiver Subfunction

The ESA AIS Receiver subfunction performs the demodulation of each AIS channel. The processing algorithms used for the ESA AIS Receiver are based on the ESA-patented algorithm, *Receiving Method And Receiver for Timing and Frequency Offset Correction of Continuous Phase Demodulation in Satellite-Based Automatic Identification Systems* (PCT/EP2014/051273).

Compilation

The source code implementing the ESA-patented AIS algorithm is located in the directory */src/ESA_AIS_receiver_code/* and can be compiled using the `Makefile` by running:

```
make
```

This will generate the *AIS_receiver* executable used by the Python script. To run the executable within the Sentinel-1 AIS end-to-end processor (*process_s1_ais_raw.py*), the executable must be located in the main directory.

Standalone Execution:

The compiled ESA AIS receiver can also be run independently from the Python script for a single channel as follows:

On MacOS/Linux

```
./AIS_receiver <data_len> <outputFile> <inputWavfile>
```

Or on Windows as:

```
AIS_receiver.exe <data_len> <outputFile> <inputWavfile>
```

Input parameters are defined as follows:

Table 1 Input Parameters for the AIS executable

Parameter	Description
`<data_len>`	AIS message bit length: 168 for heritage AIS Channels (162.025 MHz and 161.975 MHz) or 96 for SAT-AIS Channels (156.825 MHz and 156.775).
`<outputFile>`	Output directory and filename (with .txt extension) where the AIS detections will be saved.

Parameter	Description
`<inputWavfile>`	Input two-channel WAV file containing the raw IQ AIS data sampled at 28.8 kHz.

The executable requires the following files to be present in the main directory before running:

Table 2 Input files for the ESA AIS Receiver subfunction.

Filename	Description
`<h1.dat>`	Low-pass filter coefficients (7-tap Blackman-Harris).
`<h2.dat>`	Matched-filter coefficients (Principal pulse of the Laurent decomposition).
`<h3.dat>`	Low-pass filter (zonal) coefficients (7-tap Hamming)
`<h4.dat>`	Low-pass filter coefficients (7-tap Hamming for frequency estimation).
`<pulse.dat>`	GMSK pulse coefficients (only used for experimental digital remodulation).
`<error_files>`	Directory containing the syndrome error correction files.

Output

The output file from the standalone receiver is a comma separated .txt file containing the estimated Doppler frequency offset (Hz), detection index and the binary AIS message. The detection index is later used in the processing to determine the time of detection from the AIFM timestamp samples. The binary AIS message length will be either 168 or 96 depending on the selected `data_len` input variable/input channel.

An example of one row is shown below:

```
383.1, 95688, 011011011110110010011010000100010010011100010001000101101000010100011001010000110000000110010100
```

Adaptions for Ground Segment Processing

The ESA AIS Receiver subfunction is intended to replace the MATLAB AIS processing algorithms currently implemented in the EOF Copernicus Ground Segment. The ESA receiver subfunction provides the same output format as the current MATLAB implementation. The only difference in input is that the executable requires the data length parameter which is 168 for the channels at 162.025 MHz (channel 0) and 161.975 MHz (channel 1). For the channels 156.825 MHz (channel2) and 156.775 MHz (channel 3) the data length should be set to 96 bits. A separate branch `asc-version` has been created to ingest the .asc/.txt files used in the ground segment processing instead of the .wav files.

4. ALGORITHM BASELINE

4.1. Packet Extraction

The first step of the processing involves the instrument source packet extraction to retrieve the AIS IQ samples and the AIFM timestamp. The structure of the AIS CCSDS packets is shown in Figure 1. A detailed description of the extraction procedure can be found in [RD5].

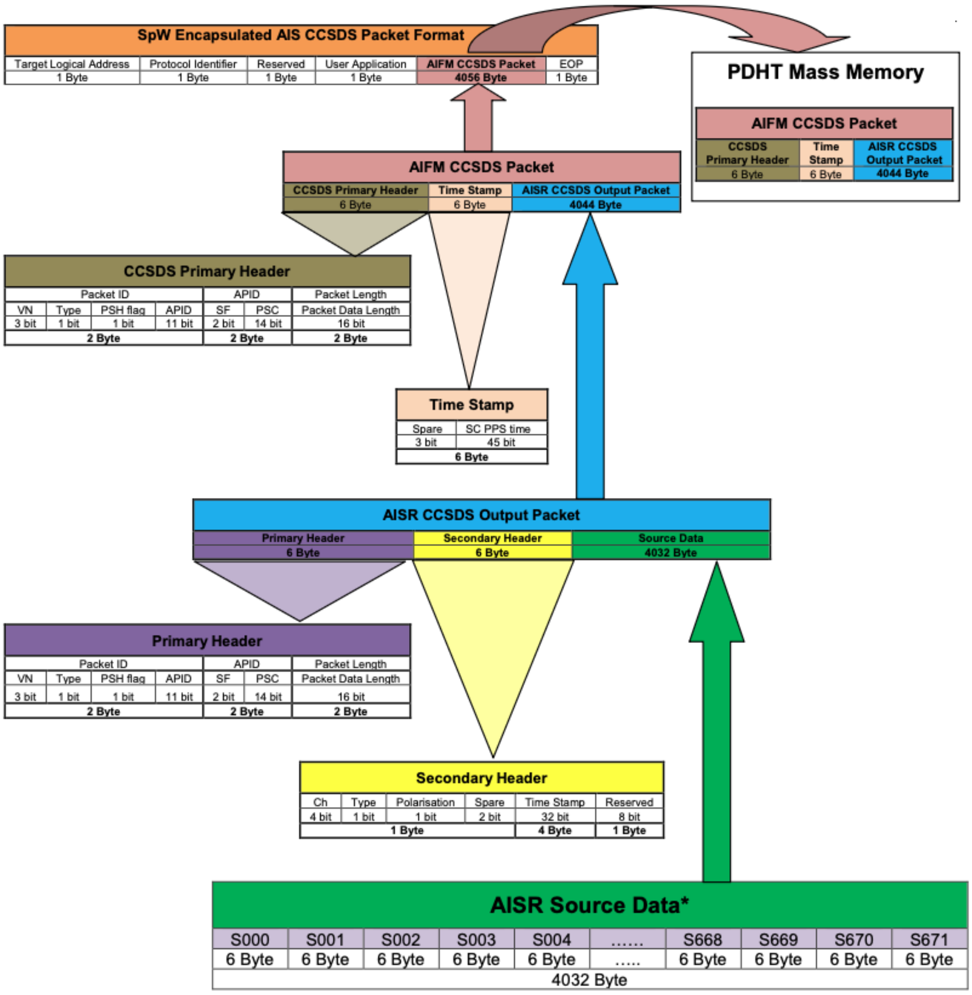

Figure 2 CCSDS packetization for the Sentinel-1 AIS Payload data.

4.2. Linear Polarization Combinations

The Sentinel-1 C&D AIS antennas receive and downlink two orthogonal polarization channels, H and V. Following the extraction of the data streams we combine the two orthogonal polarization channels into two additional data streams that synthesize the RF detection at a 45-degree angle. The θ = ±45degree linear combinations between H and V are performed as follows:

$$streamHV_{+\theta} \; = streamH * \sin(\theta) + streamV * \cos(\theta)$$
$$streamHV_{-\theta} \; = streamH * \sin(-\theta) + streamV * \cos(-\theta)$$

With the combinations at ±45degrees, we in total have 16 RF-input streams i.e. four for each frequency channel for demodulation. In the specific implementation the RF streams are saved as two channels .wav files to be passed to the ESA AIS Receiver subfunction.
Results from the Sentinel-1C IOCR have shown that the performance can be increased further using additional combinations as shown in Figure 3.

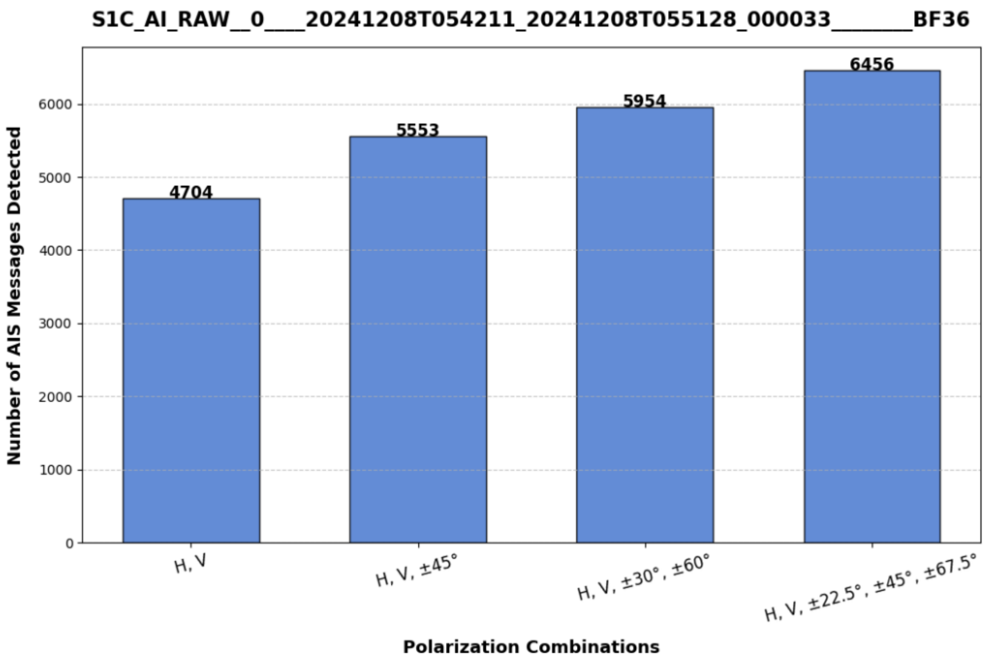

Figure 3 Comparison of total number of AIS messages detected with a range of linear polarization combinations. From left to right ([H, V], [H, V, ±45], [H, V, ±30, ±60], [H, V, ±22.5, ±45, ±67.5]).

4.3. ESA AIS Receiver

The ESA AIS receiver is an advanced AIS channel estimation and demodulation algorithm developed over several years through the ARTES Satcom technology development program. The ARTES technology development program resulted in an ESA owned patent on the AIS processing algorithms. This section gives a high-level overview of the algorithm processing flows and implementation details. A detailed description and the theoretical background can be found in [RD3] and [RD11].

Stream Processing

The ESA AIS Receiver subfunction takes a Sentinel-1 raw AIS channel as input (e.g. Channel 1, polarization V) and process the entire IQ stream to output the decoded binary AIS messages. The current implementation takes a two channel .wav file as input and outputs a .txt file as shown in Figure 4.

The processing of the RF-stream is performed in segments of frames of 330 symbols. Each frame is passed to the three zonal demodulators. After processing the first frame, the next frame is ingested but with a large overlap of 240 symbols (73%) from the previous frame. The large overlap between successive frames is essential to improve the detection of colliding AIS messages, however it results in an increase in duplicated detections. The duplicated detections are removed after processing the entire stream in the AIS Message Parsing step.

The stream processing does not pre-detect the modulated training sequence, instead the demodulation is attempted for all input frames. Each of the 16 input streams are processed in parallel on CPU, while the frames are loaded sequentially until the end of the RF-stream is reached.

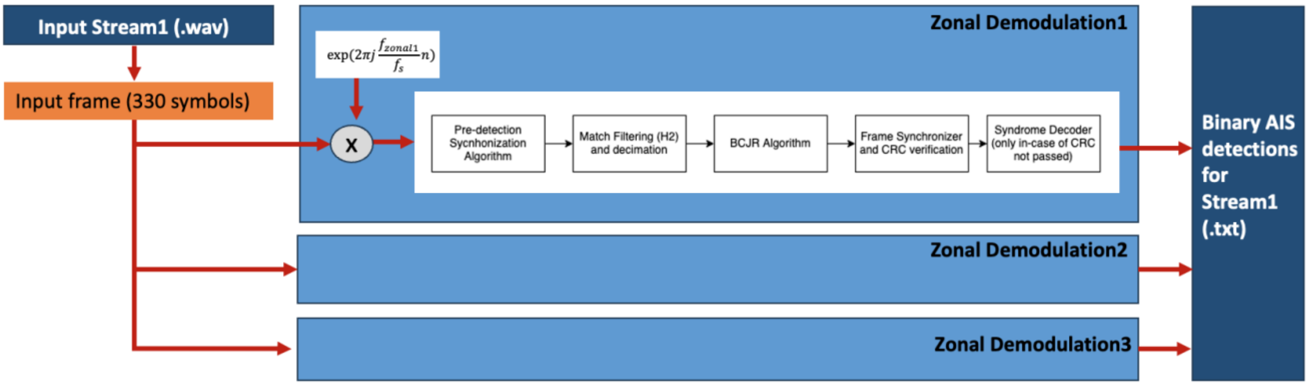

Figure 4 IQ Stream processing diagram for the ESA AIS receiver subfunction

Zonal demodulation

Each AIS frame is split into three zones, centred at different frequencies and processed independently. The current implementation includes one zonal demodulator processed at the centre frequency and two zonal demodulations centred at ±1/3 symbol rate (i.e. ±9600 Hz/3 = ±3168Hz). The zonal range is selected because the optimal frequency estimation range of the frequency estimator is less than ±0.21/T (±2.016 kHz) while the frequency offset of the AIS message can be up to a maximum value of ±5.8 kHz [RD3].

In each zonal demodulation, the channel estimation is performed on the AIS frame as shown in Figure 5. The AIS frame is filtered using the low pass filter coefficients provided and listed in Table 2. The sub-symbol time offset is estimated and corrected for in the interpolation step. The implementation of the time offset estimation is based on the Mengali-Morelli algorithm first proposed in [RD10]. Using the time offset corrected signal, a more accurate frequency offset is estimated and corrected using the algorithm described in [RD9].

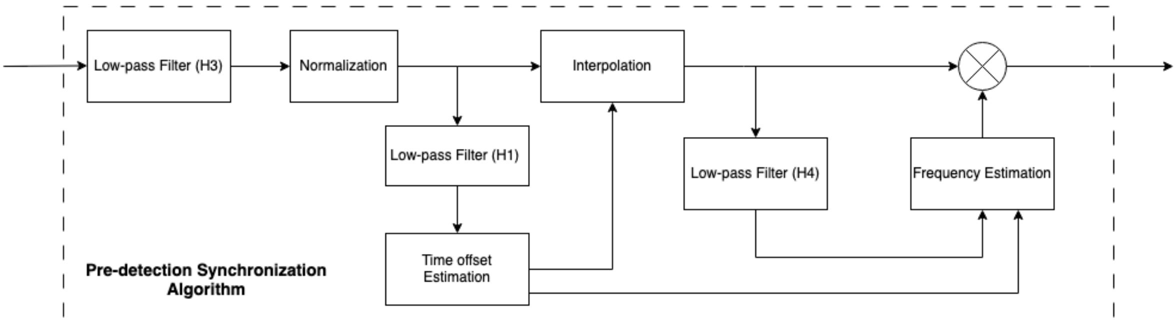

Figure 5 Pre-detection synchronization used by the ESA AIS receiver.

After the pre-detection synchronization algorithm, a matched filter (Gaussian Low pass filter) is applied to the AIS frame and the packet is decimated from three to one sample-per-symbol. The filtered and decimated signal is input to the BCJR algorithm for demodulation. The Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm does not require the phase offset parameter since the detection algorithm performs an implicit phase estimation during the construction of the trellis [RD8]. In comparison to a classic Viterbi algorithm, the BCJR algorithm has the added value of providing the soft-output Logarithmic Likelihood ratio (LLR) indicating the reliability of each decoded bit. The reliability of each output bits is later used for the Syndrome Error-correction steps.

In the frame synchronization step the first 86 bits of the demodulated output are checked to identify a potential preamble. During the preamble search only the last 6 bits of the training sequence and 8-bit start flag are used. If a preamble is identified, with a margin of up to 3-bit errors, the CRC is calculated to verify it the decoded output is a validate AIS message.

If the CRC fails a Syndrome decoding step is performed to correct for possible bit errors. The syndrome decoding uses a look up table consisting of the relationship between the check-sum values and a given error pattern. Given a check-sum value, a certain bit pattern is inverted, for example the bit sequence 010 is inverted to give 111. After one pattern has been inverted, the CRC is re-calculated to verify if the new sequence is a valid AIS message. To improve the reliability and avoid false detections only bit couples with a log-likelihood ratio less than a given threshold (25.0) are inverted.

When an AIS message with a valid CRC is detected, the binary AIS message, estimated frequency offset and detection time index are saved in a .txt file. Once the entire stream processing is complete for all 16 input IQ sample streams the output files is input into the Python AIS Message Parsing step.

4.4. AIS Message Parsing

After demodulation, the binary AIS messages with a correct CRC are passed to the AIS Message Parsing block. Due to both the overlapping detection range of the three zonal demodulators and the large overlap in AIS frames, it often occurs that the same message is detected multiple times. Likewise, the use of the linear polarization combination can result in duplicated detections. These duplicate detections are discarded at this stage. The AIS Message Parsing block uses the open-source Python package *pyais* to retrieve the message content from the binary AIS messages. The AIFM timestamp along with the index of detection is used to provide an accurate timestamp for each AIS message.

5. CONCLUSIONS

This document has described the usage, algorithmic baseline and interfaces for the Sentinel-1 AIS Processor developed at ESA-ESTEC for the Sentinel-1 EOF Ground Segment. The main demodulation algorithm is based on the outcomes of previous ARTSE development activities resulting in the ESA-patented algorithm. The ESA AIS algorithm has been adapted and incorporated into the data processing pipeline for processing the Sentinel-1 raw AIS data.