Odin/SMR

Input/Output Data Definition Document: Level2 processor

Author(s): Bengt Rydberg, Patrick Eriksson, Jimmy Petersson, Andreas Skyman, and Donal Murtagh
Contact: Bengt Rydberg <bengt.rydberg@molflow.com>
Version: 1.1
Date: 2020-09-08

Chalmers University of Technology
Department of Space, Earth and Environment

Contents

Chapter 1 | Introduction

1.1 Aim and scope of this document

Odin/SMR has been in operation since 2001 and performs passive limb measurements of the atmosphere, mainly at frequencies around 500 GHz. From these measurements, profiles of species that are of interest for studying stratospheric and mesospheric chemistry and dynamics can be derived, such as O_3, ClO, N_2O, HNO_3, H_2O, CO, and isotopologues of H_2O, and O_3. These profiles are denoted as the Level2 products of Odin/SMR. The Level2 products are generated by a Level2 processor. The input to the Level2 processor is geolocated and calibrated measurements (Odin/SMR Level1B data, described in Rydberg et al. (2020a)) and dynamic and static auxilliary/ancilliary data.

The aim of this Input/Output Data Definition Document (IODD) is to describe the input and output data used and generated by the Level2 processor. The details of the algorithms applied by the Level2 processor is described in an Algorithms Theoretical Basis Document (ATBD) - Level 2 processing (Eriksson, 2020), and are not covered by this document. The Level2 processor is a component of a Level2 processing system, and the aim is also to describe this processing system.

1.2 Document structure

This document is organized as follows: Chapter 2 describes the design of the Odin/SMR Level2 processing system. Chapter 3 describes the input data used and the output data generated by the Odin/SMR Level2 processor.

Chapter 2 | Level2 processing system

2.1 Design of the Level2 processing system

The processing system consists of three parts:

- Job Service: An api that keeps track of which jobs that are available.
- Workers: Fetch jobs from the job service, send them to the Level2 Processor and notify the job service when the jobs are done.
- Level2 Processor: Fetches the dynamic input data from the odin api and sends the resulting level2 data to the odin api.

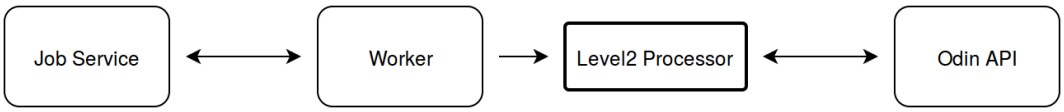

Figure 2.1: The Level2 processing system.

The inter-component communication is realised by exchanging JSON objects through a authenticated REST API over *https* (Sect. 3.2.1). This implies that the involved pool of workers do not need to be in the same network, neither as each other, nor as the server. This makes the processing system flexible and easily scalable and re-scalable on demand, as the needs for processing power changes. The main bottle neck is the access to the database, which is therefore located on a dedicated server.

Chapter 3 | Level2 processor and I/O data

3.1 Level2 processor overview

The Odin/SMR Level2 processor is designed to process measurements from a scan of Odin/SMR in order to retrieve profiles of atmospheric species or temperature. The Level2 processor is an optimal estimation method (OEM) implementation, which combines measurement information with Ancillary/Auxiliary data and applies a forward model, and an iterative scheme is used to find a best solution. The details of the algorithms within the Odin/SMR Level2 processor is described in Odin/SMR Level2 ATBD (Eriksson, 2020).

3.2 Level2 processor input data

Figure 3.1 describes, on a high level, the input data used by the Level2 processor and the generated output data. The input data can be divided into two groups, dynamic (or semi-dynamic) and static input data,

- dynamic (or semi-dynamic) input data:
  - *Level1B data*: the most basic input data to the Level2 processor is the Level1B data, i.e. geolocated and calibrated spectra from a scan of Odin/SMR.
  - *Climatological apriori data*: climatology data, covering all species of interest is required, as the OEM implementation needs a starting estimate for each profile to be retrieved. The climatology covers vertical, seasonal, and latitudinal variations of each species and is based on data from the "Bourdeaux" climatology.
  - *ZPT data*: The Level2 processor requires external ZPT data, (altitude, pressure, and temperature). The ZPT data is extracted from the ERA-Interim and ERA-5 datasets for the geolocation of the scan. ERA-Interim is a global atmospheric reanalysis with coverage from 1979 until 2019-08-31. ERA-Interim is replaced by ERA-5 data after this date in the Level2 processing. The ERA-Interim / ERA-5 data used has a 0.75° resolution in latitude and longitude and 6 h in time.
- static input data:
  - *Sensor characteristic data*: The forward model simulation of the Level2 processor takes into account sensor characteristic data, i.e. the antenna angular response, sideband filtering, and frequency response of each spectrometer channel.

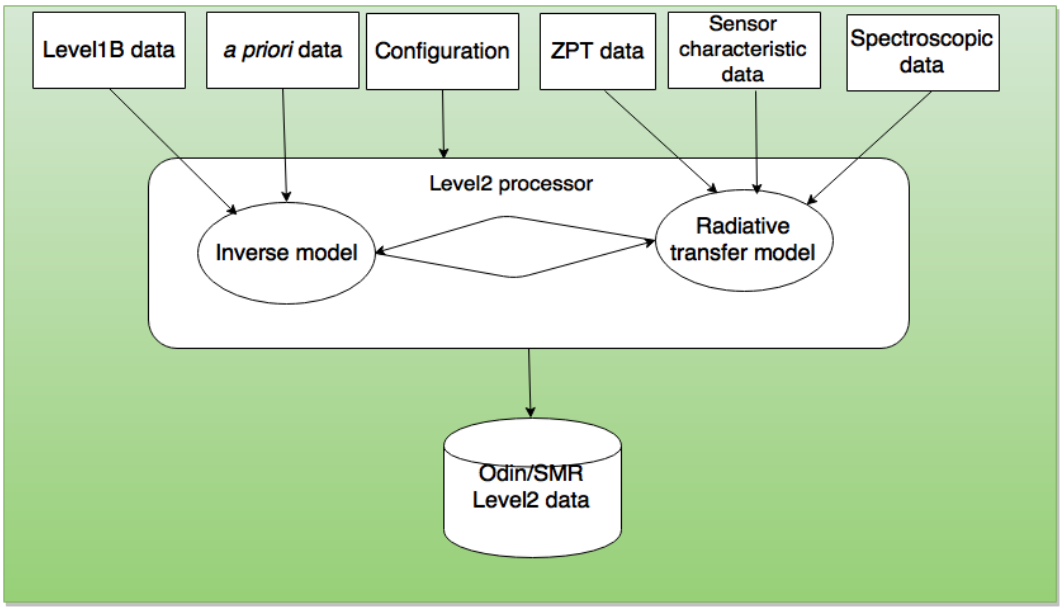

Figure 3.1: Schematic of the I/O data used and generated by the Odin/SMR Level2 processor.

- *Spectroscopic data*: Spectroscopic data is basic input to the forward model of the Level2 processor.
- *Configuration*: Odin/SMR applies a number of different observation modes (different frequency intervall coverage and scanning altitudes) and for each mode a different configuration is applied of the Level2 processor.

The Level2 processor obtains the dynamic and semi-dynamic input data through a hierarcical REST API, and the API is described in Sect. 3.2.1. The static input data is integrated into the Level2 processor, as described in Figure 3.2. The detailed data formats of the dynamic and static data are described in Sect. 3.2.1 and Sect. 3.2.2.

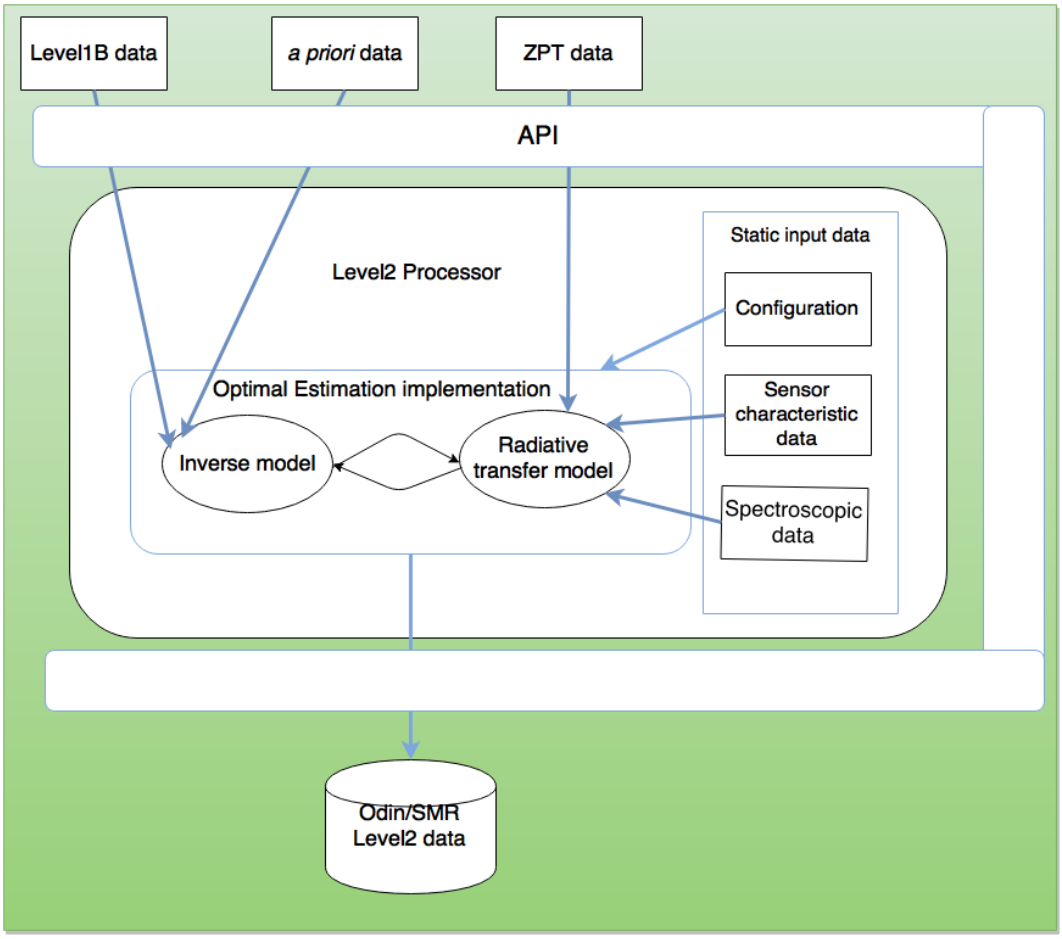

Figure 3.2: Schematic of the I/O data used and generated by the Odin/SMR Level2 processor.

3.2.1 API and dynamic input data

The Level2 processor obtains dynamic input data through a hierarcical REST API, where deeper URIs return more specific data. An online API documentation can be found at: http://odin.rss.chalmers.se/apidocs/index.html#/. All GET calls return JSON objects, and the ones of concern here are:

- /rest_api/<version>/level1/<freqmode>/<scanno>/Log/: Get log info for a scan from cached table
- /rest_api/<version>/level1/<freqmode>/<scanno>/L1b/: Get level1 data for a scan
- /rest_api/<version>/level1/<freqmode>/<scanno>/ptz/: Get ptz data for a scan
- /rest_api/<version>/level1/<freqmode>/<scanno>/apriori/<species>/: Get apriori data for a scan and species

The data returned by the four various GET calls listed above is described in the following sections. Key/value pairs are listed as name of the key along with the type of the corresponding value within parantheses, followed by a brief description of the contents.

3.2.1.1 Level1B overview data

/rest_api/<version>/level1/<freqmode>/<scanno>/Log/

Method: *GET*

Returns object, which describes log data of the scan for the given freqmode and scan number with the following keys:

- AltEnd *(float)*: Tangent point altitude ([m]) for last spectrum in scan
- AltStart *(float)*: Tangent point altitude for first spectrum in scan
- DateTime *(str)*: Mean UTC datetime (datetime(first) + datetime(last))/2 of scan
- FreqMode *(Int)*: Deployed frequency mode
- LatEnd *(float)*: Latitude of tangent point for last spectrum in scan
- LatStart *(float)*: Latitude at tangent point for first spectrum in scan
- LonEnd *(float)*: Longitude of tangent point for last spectrum in scan
- LonStart *(float)*: Longitude at tangent point for first spectrum in scan
- MJDEnd *(float)*: Modified julian date for last spectrum in scan
- MJDStart *(float)*: Modified julian date for first spectrum in scan
- NumSpec *(int)*: Number of atmospheric spectra in scan
- Quality *(int)*: A quality estimate of the scan
- ScanID *(int)*: Satellite time word identifier of scan

- SunZD *(float)*: Mean solar zenith angle (SunZD(first) + SunZD(last))/2 for scan
- URLS a list of key/value describing URIs containing more detailed data
  - URL-apriori-<species> *(URI)*: A URI for getting *a priori* data of the scan (`/rest_api/<version>/level1/<freqmode>/<scanno>/apriori/<species>/`)
  - URL-ptz *(URI)*: A URI for getting ptz data for the scan (`/rest_api/<version>/level1/<freqmode>/<scanno>/ptz/`)
  - URL-spectra *(URI)*: A URI for getting Level1B spectra of the scan (`/rest_api/<version>/level1/<freqmode>/<scanno>/L1b/`)

3.2.1.2 Level1B data

`/rest_api/<version>/level1/<freqmode>/<scanno>/L1b/`

Method: *GET*

Returns an object that describes the Level1B data of the scan, with the following keys (See Rydberg et al. (2020a) for more details):

- Altitude *(Array of doubles)*: Tangent point altitude [m]
- Apodization *(Array of integers)*: Filter used when converting auto-correlations to power spectrum. 1 = no filter, 1 = Hanning
- AttitudeVersion *(Array of integers)*: The version number of the attitude reconstruction software (SODA) used at SSC during production of attitiude files
- Backend *(Array of integers)*: The backend used for this observation: 1 = AC1 and 2 = AC2
- Channels *(Array of integers)*: The number of channels in this spectrum
- Dec2000 *(Array of doubles)*: The declination (J2000) of the direction of pointing in degrees
- EffTime *(Array of doubles)*: The effective integration time in seconds, i.e. you will get the noise level in this spectrum by using this time and the receiver noise temperature from above and the usual radiometer formula:

  ```
  dT = Trec/sqrt(df * EffTime)
  ```

  where `df` is the bandwidth (FreqRes) of the spectrometer
- FreqCal *(2-D Array of doubles)*: These are the four local oscillator frequencies of the SSB modules
- FreqMode *(Array of integers)*: Frequency mode applied
- FreqRes *(Array of doubles)*: The spacing in Hz between neighbouring channels for this spectrum
- Frequency *(Object)*: an object that contains information that can be used to create frequency grids for the spectra and consists of following keys:

  - LOFreq *(Array of doubles)*: Local oscillator frequency in Hz in the rest frame of the observed object, i.e Doppler corrected
  - AppliedDopplerCorr *(Array of doubles)* The applied Doppler correction in Hz
  - IFreqGrid *(Array of doubles)*: The frequency grid can be obtained by

    ```
    frequency = LOFreq + IFreqGrid
    ```

  - SubBandIndex *(2-D Array of integers)*: The SubBandIndex can be used to identify from which sub-band a given channel belongs to. An example SubBandIndex array is [[-1, -1, 420, 509, 111, 1, 310, 200], [-1, -1, 508, 618, 199, 110, 419, 309]]. The first and second row indicate the start and end positions, respectively, of the eight sub-bands in the frequency sorted spectrum. The first and second index in these rows corresponds to SubBand 1 and 2, and so on. E.g sub-band 3 data can be found between index 420 to 508. -1 indicates that data from this band is not used.
  - ChannelsID *(Array of integers)*: Channel identifier describes the location of the sorted channels in the original unsorted spectra

- Frontend *(Array of integers)*: The frontend used for this observation: 1 = 555, 2 = 495, 3 = 572, 4 = 549, and 5 = 119
- GPSpos *(2-D Array of doubles)*: The geocentric position X,Y,Z in meter of the satellite
- GPSvel *(2-D Array of doubles)*: The geocentric velocity $\dot{X}$, $\dot{Y}$, $\dot{Z}$ in meter per second of the satellite
- IntTime *(Array of doubles)*: The integration time in seconds, i.e. the duration of this observation
- Latitude *(Array of doubles)*: Geodetic latitude of the tangent point
- Longitude *(Array of doubles)*: Geodetic longitude of the tangent point
- MJD *(Array of doubles)*: Modified julian date of observation
- MoonPos *(2-D Array of doubles)*: The geocentric position of the Moon in meter
- Orbit *(Array of doubles)*: Orbit number, including fractional part
- Quality *(Array of integers)*: Quality indicator of scan/spectrum
- RA2000 *(Array of doubles)*: The right ascension (J2000) of the direction of pointing in degrees
- SBpath *(Array of doubles)*: The path length in meter of the SSB diplexer
- ScanID *(Array of integers)*: satellite time word scan identifier
- STW *(Array of integers)*: Satellite time word
- Spectrum *(2-D Array of doubles)*: Frequency sorted and intensity calibrated spectra in Rayleigh Jeans temperature in Kelvin
- SunPos *(2-D Array of doubles)*: The geocentric position of the Sun in meter

- SunZD *(Array of doubles)*: The solar zenith angle in degrees
- TSpill *(Array of doubles)*: Estimated Tspill in Kelvin
- Tcal *(Array of doubles)*: The temperature in Kelvin of the calibration load
- Trec *(Array of doubles)*: The mean value of the receiver nosie temperature
- TrecSpectrum *(Array of doubles)*: Frequency sorted receiver noise temperature spectrum determined and used in calibration of this scan
- Version *(Array of integers)*: Calibration version
- Vgeo *(Array of doubles)*: The velocity of the satellite with respect to the Earth in meter per second
- ZeroLagVar *(2-D Array of doubles)*: Zerolag variation ([%]) of the surrounding reference measurements for all sub-bands.

  ```
  ZeroLagVar =
  abs(diff(ZeroLag))/mean(ZeroLag)
  ```

  ZeroLag is the measured coefficient of the first channel of the auto-correlator, and is proportional to the total power of the band

3.2.1.3 PTZ data

r/rest_api/<version>/level1/<freqmode>/<scanno>/ptz/

Method: *GET*

Returns an object that describes the ptz data of the scan, with the following attributes:

- Altitude *(array of doubles)*: Altitude vector([m])
- Latitude *(double)*: Latitude of PTZ data
- Longitude *(double)*: Longitude of PTZ data
- MJD *(double)*: Modified Julian Date of PTZ data
- Pressure *(array of doubles)*: Pressure vector ([Pa])
- Temperature *(array of doubles)*: Temperature vector([K])

3.2.1.4 Climatological *apriori* data

/rest_api/<version>/level1/<freqmode>/<scanno>/apriori/<species>/

Method: *GET*

Returns an object that describes the *a priori* data of the scan, with the following attributes:

- Altitude *(array of doubles)*: Altitude vector([m])
- Pressure *(array of doubles)*: Pressure vector ([Pa])

- Species *(String)*: description of species, i.e. one of 'BrO', 'Cl2O2', 'CO', 'HCl', 'HO2', 'NO2', 'OCS', 'C2H2', 'ClO', 'H2CO', 'HCN', 'HOBr', 'NO', 'OH', 'C2H6', 'ClONO2', 'H2O2', 'HCOOH', 'HOCl', 'O2', 'SF6', 'CH3Cl', 'ClOOCl', 'H2O', 'HF', 'N2', 'O3', 'SO2','CH3CN', 'CO2', 'H2S', 'HI', 'N2O', 'OBrO', 'CH4', 'COF2', 'HBr', 'HNO3', 'NH3', or 'OClO'.
- VMR *(array of doubles)*: Climatology volume mixing ratio vector [−] of the species

3.2.2 Static input data

Static data (sensor characteristic data, spectroscopic data, and configuration) is required by the Level2 processor, although this data is in a strict sense not input data to the Level2 processor, as this data is contained by the Level2 processor as explained by Figure 3.2. The sensor characteristic and spectroscopic data is used by the forward model (ARTS) of the Level2 processor, and the configuration controls the complete retrieval calculation including the forward model calculation. The static data used by the Level2 processor is described in the following sections.

3.2.2.1 Configuration

The Odin/SMR retrievals are performed by a Matlab implementation of OEM, that is a part of the Atmlab package (available at `www.radiativetransfer.org/tools`). The main OEM function in Atmlab is *oem.m*. The Atmlab package contains the required functions to interface *oem.m* with ARTS (i.e. read/write ARTS output/input files). The OEM implementation is controlled by a Matlab object (a Matlab structure denoted as Q), and a description of this structure is found in Rydberg et al. (2020b). Part of this setting is general for all of the various observation modes applied by Odin/SMR while some of the settings are observation mode specific.

3.2.2.2 Sensor characteristic data

Antenna angular response

A number of data files describing the antenna pattern are contained by the Level2 processor. A given file is valid for a specific frequency mode and integration time, and the file naming convention is:
`qsmr/DataFiles/Antenna/antenna_fmode<XX>_tint<YYYY>ms.xml`
where *XX* describes the frequency mode (e.g 01, 02, and so on) and *YYYY* describes the integration time (e.g 0860, 1860, or 3860). These files are valid for an "azimuthally collapsed" antenna. The file format is ARTS xml GriddedField4 (see ARTS user guide (Eriksson and Buehler, 2016)), where the grids are *Polarisation*, *Frequency*, *Zenith angle*, and *Azimuth angle*, and the only non-scalar grid is *Zenith angle*. An example is given below:

```
<?xml version="1.0"?>
<arts format="ascii" version="1">
  <GriddedField4 name="Antenna response function">
    <Array name="Polarisation" type="String" nelem="1">
      <String>
          "1"
      </String>
    </Array>
```

```
    <Vector name="Frequency" nelem="1">
      5.4482500e+11
    </Vector>
    <Vector name="Zenith angle" nelem="161">
      -2.0000000e-01
      -1.9750000e-01
      ...
      2.0000000e-01
    </Vector>
    <Vector name="Azimuth angle" nelem="1">
       0.0000000e+00
    </Vector>
    <Tensor4 name="Response" nbooks="1" npages="1" nrows="161" ncols="1">
      3.2714490e-04
      3.4765317e-04
      ...
      3.4187888e-04
    </Tensor4>
  </GriddedField4>
</arts>
```

Sideband filtering

Odin/SMR operates as a single sideband receiver. The nominal sideband suppression is better than 19 dB across the image band, but the actual in-flight sideband leakage is not perfectly known. The sideband leakage has been estimated to vary between the frequency modes. The sideband leakage assumed in the retrieval caclulation is controlled by the configuration parameter *SIDEBAND_LEAKAGE*, which describes the relative contribution of the sideband. So far the sideband leakage is assumed to be flat over each frequency band.

Frequency response of each spectrometer channel

The frequency response function of each spectrometer channel is contained by the file `qsmr/DataFiles/Backend/backend_df1000kHz_withHan.xml` of the Level2 processor. The response function by itself is determined from the algorithm that is applied to calibrate Odin/SMR spectra, i.e. if a Hanning smoothing is applied or not. The file format is ARTS xml GriddedField1 (see ARTS user guide (Eriksson and Buehler, 2016)), where the grid is *Frequency*, and an example is given below:

```
<?xml version="1.0"?>
<arts format="ascii" version="1">
  <Array type="GriddedField1" nelem="1">
    <GriddedField1 name="Backend channel response function">
      <Vector name="Frequency" nelem="45">
        -4.8125000e+06
        -4.5937500e+06
        ...
        4.8125000e+06
      </Vector>
```

```
      <Vector name="Response" nelem="45">
        -1.6581976e-03
        -3.2984437e-03
        ...
        -1.6581976e-03
      </Vector>
    </GriddedField1>
  </Array>
</arts>
```

3.2.2.3 Spectroscopic data

Spectroscopic data is basic input to the forward model (ARTS) applied. The absorption data used in the calculation is extracted from an absorption cross-section look-up table for a computational efficiency reason. The abosrption look-up table is contained by the Level2 processor. ARTS provides methods to create, read and write, and extract data from the lookup table. The format of the absorption look-up table and how to create it is described in details in Eriksson and Buehler (2016).

3.3 Level2 processor Output Data

The output of the OEM implementation of the Level2 processor is two Matlab objcets (L2 and L2I), where L2 contains the main results of the retrieval and L2I describes auxiliary instrument and retrieval data (Eriksson, 2020). The Level2 processor posts this data to the API (the *POST* URL is `rest_api/<version>/level2`) which in turns store the data in a Odin/SMR Level2 database. The L2 and L2I objects are described in Perot et al. (2020).

Bibliography

P. Eriksson. Odin/SMR algorithms theoretical basis document - Level2 processing. Technical report, Department of Earth and Space Sciences, Chalmers University of Technology, 2020.

P. Eriksson and S. Buehler. ARTS user guide. Technical report, Department of Earth and Space Sciences, Chalmers University of Technology, 2016.

K. Perot, P. Eriksson, D. Murtagh, and B. Rydberg. Odin/SMR: L2 data - format and overview. Technical report, Department of Earth and Space Sciences, Chalmers University of Technology, 2020.

B. Rydberg, P. Eriksson, J. Kiviranta, A. Skyman, and D. Murtagh. Odin/SMR: Algorithms Theoretical Basis Document - Level 1b data. Technical report, Department of Earth and Space Sciences, Chalmers University of Technology, 2020a.

B. Rydberg, P. Eriksson, and D. Murtagh. Odin/SMR: Level2 processing configuration. Technical report, Department of Earth and Space Sciences, Chalmers University of Technology, 2020b.