

SatERR User Manual

V1.1 April 4, 2022
SatERR Team

Contents

1	Introduction	3
1.1	Context	3
1.2	SatERR Overview	4
2	Download and Installation	6
2.1	Download	6
2.2	Installation	6
2.3	Code Directory	6
2.4	Ancillary Data	7
2.5	Third Party Models	8
3	How to Use SatERR	9
3.1	A Quick Start	9
3.2	Setting Error Sources	9
3.3	Setting Modes	11
3.4	Output and Visualization	12
3.5	Advanced Setting	12
3.6	Sequence of Execution	13
3.7	Simulating a New Sensor	16
4	Observation Error Inventory	21
4.1	Algorithm Theoretical Basis	21
4.2	Measurement error	21
4.2.1	Radiometer diagram	21
4.2.2	Generating count, T_A with noise and orbital oscillation	22
4.2.3	Windowing and Calibration	26
4.2.4	Decomposing Noise	28
4.2.5	Examining Noise with PCA	29
4.2.6	Striping	30
4.2.7	Cross-pol Contamination	31
4.2.8	Twist angle offset	31
4.2.9	Spacecraft attitude and geolocation	34
4.2.10	Antenna pattern	35
4.2.11	Faraday rotation	35
4.2.12	Reflector emission	37

4.2.13	Cold-space mirror emission	37
4.2.14	Warm load error	37
4.2.15	FOV intrusion	38
4.2.16	Bandpass spectral response	38
4.2.17	Interference of instrument and geomagnetic field	38
4.2.18	Crosstalk	39
4.2.19	Antenna temperature to brightness temperature	39
4.3	Observation operator error	39
4.3.1	Ancillary models and modules	39
4.4	Representativeness error	40
4.5	Pre-processing Error	40
4.6	Miscellaneous	40
4.6.1	Collocation	40
4.6.2	Data download	41
A	Acronyms and Abbreviations	43
B	Simulating Current and Future Sensors	44
C	List of Error Sources	44
D	List of Scripts	45
	Bibliography	50

1 Introduction

1.1 Context

The error representation and uncertainty quantification associated with satellite observations are critical for hardware development, data assimilation (DA), weather forecast, and environmental and climate data records [Kalnay, 2003; Langland and Baker, 2004; Joo et al., 2013; Geer et al., 2018]. Satellite observations play an important role for all these applications (Figure 1).

In the context of data assimilation, observation errors encompass four categories in terms of measurement error, observation operator error, representativeness error, and pre-processing error [Hollingsworth and Lönnberg, 1986; Desroziers et al., 2005; Stewart et al., 2014; Bormann and Bauer, 2010]. In data assimilation requires quantifying observation error with error covariance for weighing different observations and models. For instance, a number of methods have been developed to diagnose inter-channel correlation and its error covariance, which can improve data assimilation [Weston et al., 2014; Janjić et al., 2018].

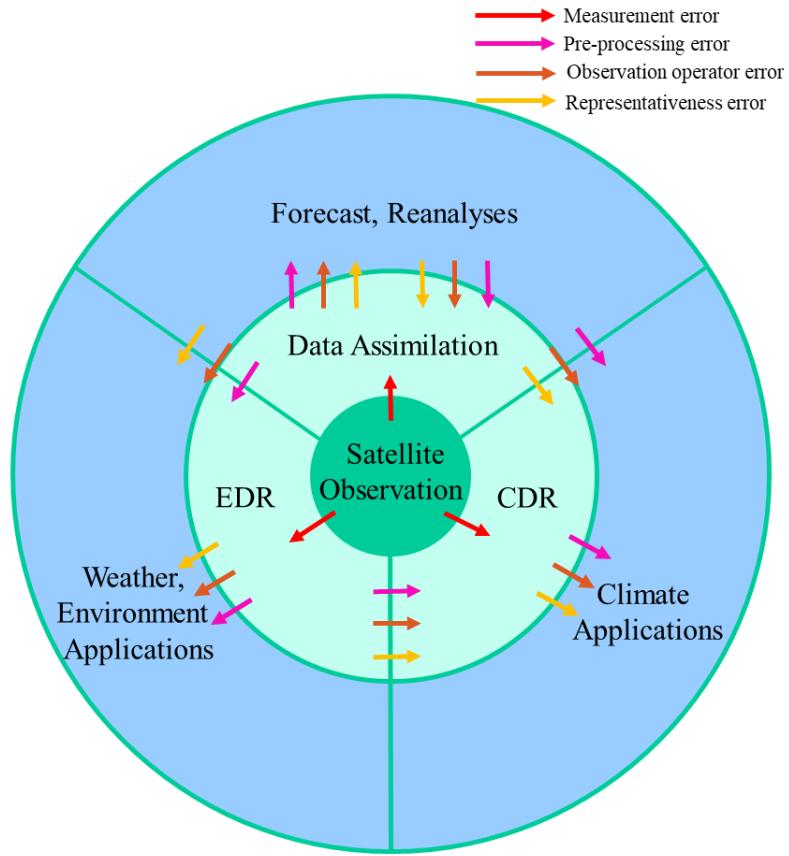


Figure 1: Comparison of the top-down and bottom-up approaches for observation uncertainty quantification. The error inventory SatERR is a bottom-up approach.

Aside from data assimilation, assessing observation error and uncertainty also matters for the development and application of the Environmental Data Record (EDR) and Climate Data Record (CDR) from satellite observations [Hollmann et al., 2013; Ablain et al., 2015; Merchant et al., 2017; Bellprat et al., 2017; Merchant et al., 2019; Gruber et al., 2020]. An accurate uncertainty estimate

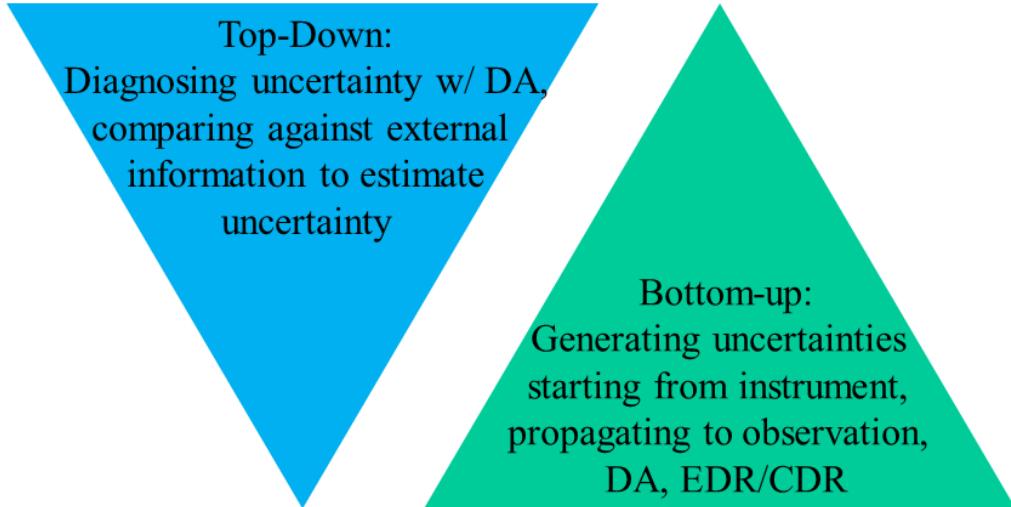


Figure 2: Comparison of the top-down and bottom-up approaches for observation uncertainty quantification. The error inventory SatERR is a bottom-up approach.

can weigh different observations and models for developing high-quality EDR/CDR. While fixed uncertainties are still used in EDR/CDR, simulating error at the pixel level based on physical models can improve error representation.

Also, satellite measurement error can arise from a range of sources related to hardware. Calibration is indispensable for correcting measurement errors, which desires a thorough understanding of error sources and propagation. The measurement error identification and quantification help the development of next-generation sensors.

There are basically two ways for uncertainty quantification, the top-down and bottom-up approaches (Figure 2). Most of DA-based diagnostics fall into the first category, that is, starting from the satellite observations and then iteratively solving for the uncertainty. This approach is affected by observational data, diagnostic techniques, and the quality of external benchmark data. An error inventory is a bottom-up approach and can provide information of error propagation.

1.2 SatERR Overview

The **Satellite Error Representation and Realization** (SatERR) is a satellite observation error inventory. SatERR is a bottom-up approach, with error generating and forward propagation. SatERR can simulate a variety of satellite observation errors. Individual or compound error sources can be turned on and off for applications under different scenarios. The uncertainty information is based on physical models that are a much more accurate representation than the fixed or empirical uncertainties.

Applications of SatERR covers a broad areas such as:

- Simulating error and studying error propagation in assimilating satellite data
- Generating ground truth testbed for validating and improving diagnostics
- Simulating error for EDR/CDR products
- Comparing with ground instrument tests, in-orbit measurement

- Calibrating and intercalibrating satellite sensors

SatERR is applicable to both existing and future satellites. It include a range of sensors such as AMSU-A, AMSU-B, AMSR2, ATMS, GMI, MHS, SMAP, SSMI, SSMIS. It can be used for small satellites such as TROPICS, TEMPEST-D. It can also be extended to new sensors with customized scanning and orbit.

SatERR is developed in MATLAB and Python. The primary codes are developed in MATLAB, without the prerequisite of advanced toolboxes. The simulator is highly vectorized for fast computing such as RTM, orbit, and collocation. It considers the application in Linux-like environment, allowing for breaking down tasks and submitting multiple jobs with hundreds and thousands of cores. The simulator package comprises mostly scripts and several demo data with a total size of 3 MB. The source code has over 30,000 lines of code with over 300 scripts. The code is open source and freely available.

2 Download and Installation

2.1 Download

The SatERR source code is released in a compressed tarball¹ via the GitHub ftp site:

<https://github.com/jxyangrs/saterr>

Documents and literature are also made available on GitHub.

https://github.com/jxyangrs/saterr/blob/main/manual_saterr.pdf

For any issues with access and permission, please contact Dr. John Yang via email jxyang@umd.edu.

2.2 Installation

SatERR can run in Windows, Mac, Linux-like OS. It basically needs no installation but MATLAB and Python. For Matlab, a version of 2016b or later that supports implicit expand for arithmetic operation is recommended. Python 3 is suggested.

After downloading SatERR, change the current directory to a root path such as “~/saterr”. The main script is `saterr_main.m` under the root path.

For a quick check, run the command line after launching MATLAB

```
saterr_main('check')
```

A message will pop out if it went great.

```
'SatERR ran successfully'
```

Figure 8 outlines the structure of SatERR. The simulator comprises a set of modules with a comprehensive error inventory. Various observation errors can be generated. We brief the modules as follows, and more details can be found in the manual online (https://github.com/jxyangrs/saterr/blob/main/manual_saterr.pdf).

2.3 Code Directory

SatERR scripts are organized in a directory structure that subsets scripts per their functionality (Figure 3). It is kept as straightforward as possible. The main script is `saterr_main.m` under the root path of “~/saterr/”.

There are three sub-directories under “~/saterr/”:

- The “src” directory encompasses all source code files.
- The “data” directory contains sample data such as satellite data, land-ocean mask, and re-analyses.
- The “doc” directory includes help documents.

The “src” directory hosts sources code sorted out in a number of sub directories:

- “set” is for setting error sources, satellite, sensors, etc. Scripts in this directory are what users use most frequently.
- “parse” is parsing input and setting.
- “main” contains main scenario scripts.

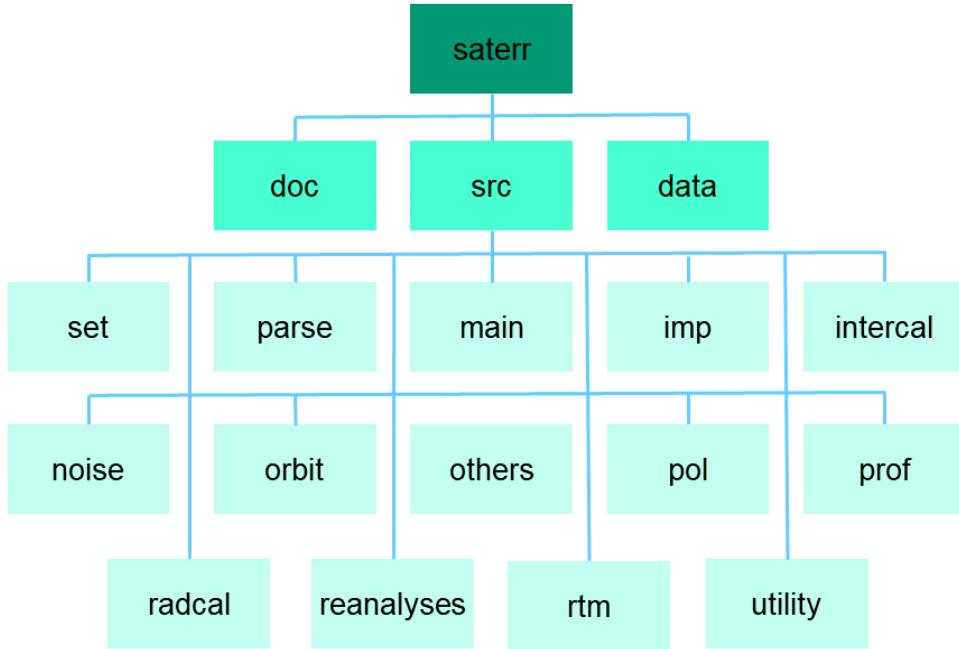


Figure 3: SatERR code directory.

- “imp” includes scripts for function implementing such as error simulation.
- “intercal” directory is for single sensor or multiple sensor calibration and intercalibration.
- “noise” is for sensor noise simulation, decomposition.
- “orbit” is for geolocation and pointing.
- “others” contains third-party models.
- “pol” is about polarization computing.
- “prof” is for reading and processing data of profile like satellite data and reanalyses.
- “radcal” is for processing radiance into counts, antenna temperature (T_A) and brightness temperature (T_B).
- “reanalyses” is for downloading and preprocessing ancillary reanalysis data.
- “rtm” includes a RTM model.
- “utility” contains miscellaneous scripts such as data processing, visualization

2.4 Ancillary Data

SatERR supports the use of ancillary data. These data are not included in SatERR due to their large size, but we provide scripts for downloading, reading and processing those data. Reanalysis

data such as ERA5, GDAS can be used for radiative transfer simulation. Scripts for processing reanalyses can be found under `~/src/reanalyses`.

When Faraday rotation effect is considered, users can use electron profile, geomagnetic field downloaded from external sources. Various satellite data such as those from NOAA and EUMETSAT can be read and fed to the simulator. Users can also develop I/O scripts for reading satellite data with specific formats.

2.5 Third Party Models

There are a variety of models that users can use and combine with SatERR. We are aware of many well established models and do not want to reinvent the wheel. We provide interface for input simulation results from third-party models. SatERR can also generate intermediate files to feed other models.

For example, users can use other RTMs for simulating the Top of Atmosphere (TOA) radiance and feed it to SatERR. A number of RTM models include the Radiative Transfer for TOVS (RTTOV) and the Community Radiative Transfer Model (CRTM). Users can run their RTMs and feed the simulation results to our simulator. In addition, SatERR can also be used to collocate satellite observations with reanalyses, generating input profiles for CRTM and RTTOV.

Once errors are added into the radiance with SatERR, there are lots of models, retrieval algorithms for processing radiance data into EDR/CDR products. For example, the NOAA's Microwave Integrated Retrieval System (MIRS) is one powerful retrieval system based on 1-dimensional variational (1Dvar) inversion. Another application is feeding data assimilation system for diagnosing and quantifying error propagation.

3 How to Use SatERR

3.1 A Quick Start

Once you have obtained and installed SatERR, you want to get familiar with it by running a couple of simple tests. A simple task starts with executing the command line:

```
saterr_main('step1')
```

‘Step1’ is running simulation transforming state vectors to TOA radiance to counts. The default setting can be found in the main script `saterr_main.m` as follows:

```
1 % -----
2 % Input setting
3 % -----
4 % radiometer specification
5 Setting.Rad.sensor = 'amsu-a'; % Demo_Oscillation/amus-a/mhs/amsr2/smap/atms/
   % Customize
6 Setting.Rad.spacecraft = 'metop-a'; % optional for Demo_Oscillation; NOAA-19 for
   % AMSU-A/MHS; GCOM-W for AMSR2
```

where the sensor is `amsu-a` and spacecraft is `metop-a`. Only a part of the setting is illustrated. Go to the main script for details.

The second step is converting counts to T_A , T_B by executing the command line:

```
saterr_main('step2')
```

You are done with the simple task. The results are in the newly created directory

“`~/saterr/output/metop-a/amsu-a/simple`”

where there are two sub-directories of “`/1sim`” and “`/2cal`”. The first one contains simulated orbit, TOA T_B , count, etc. The second one has calibrated T_A, T_B and plots.

3.2 Setting Error Sources

After finishing the simple task, let us take a close look at specific settings. All of the settings are made in files in the directory “`~/saterr/src/setting`”. These settings can be found in the overall setting file `saterr_main_setting.m`, which is in the main script `saterr.m`

The content of the overall setting file `saterr_main_setting.m` is as follows

```
1 function saterr_main_setting
2 % overall setting of error sources, control configuration
3 %   A number of settings can be set in this script. Users can customize these
   %   settings for specific error sources and
4 %   configuration
5 %
6 % Input:
7 %   setting a range of error sources, control configuration
8 %
9 % Output:
10 %   customized settings
11 %
```

```

12 % written by John Xun Yang, University of Maryland, jxyang@umd.edu, or
13 % johnxun@umich.edu, 10/03/2018: original code
14 % revised by John Xun Yang, University of Maryland, jxyang@umd.edu, or
15 % johnxun@umich.edu, 02/13/2020: review
16
17 % =====
18 %% setting
19 % =====
20
21 % -----
22 % Input setting
23 % -----
24
25 % directory and scheme
26 saterr_set_path_scheme
27
28 % parsing input and preprocessing
29 saterr_parse_input
30
31 % -----
32 % radiometer basics
33 % -----
34
35 % radiometer basic specification
36 saterr_set_radspc
37
38 % radiometer channel specification
39 saterr_set_radspc_chan

```

where we only show the first 40 lines of the file. You can see a couple of setting functions, `saterr_set_path_scheme`, `saterr_set_radspc`, and `saterr_set_radspc_chan`. More details about the three settings can be found in the scripts. Overall, the script, `saterr_main_setting.m`, is where you can explore for sub-settings.

Next, we take a look at an example about setting the receiver nonlinearity. The specific setting script is `saterr_set_nonlinear.m`. The content of the script is as follows:

```

1 function saterr_set_nonlinear
2 % setting receiver nonlinearity
3 %
4 % Input:
5 %     Rad.nonlinear.onoff,           turn on/off nonlinearity,          0/1
6 %     T_nl,                         peak nonlinearity of each channel (K), [1,
7 %                           channel]
8 %
9 % Output:
10 %    Rad.nonlinear.onoff,          turn on/off nonlinearity,          0/1
11 %    Rad.nonlinear.T_nl,          peak nonlinearity of each channel(K), [1,
12 %                           channel]
13 %
14 % Description:
15 %     The receiver nonlinearity is written as
16 %             Tas = Tc + X(Tw-Tc) + 4*T_nl*X*(1-X),          (1)

```

```

15 %      where Tas is the scene antenna temperature w/ nonlinearity (K), Tc is
16 %      cold-space TB (K), Tw is warm-load TB (K), T_nl is the peak nonlinearity (K)
17 %      and X=(Cs-Cc)/(Cw-Cc) with Cs,Cc,Cw for scene count, cold count, and
18 %      warm count respectively.
19 %      The first term, Tc + X(Tw-Tc), is the linear part, and 4*T_nl*X*(1-X) is
20 %      the nonlinear part.
21 %      The maximum nonlinearity appears with X=0.5
22 %      Equation 1 is rewritten as
23 %          4*T_nl*X^2 - (Tw-Tc+4*T_nl)*X + Tas-Tc = 0
24 %          a*X^2+b*X+c = 0,                                     (2)
25 %      where a=4*T_nl, b=-(Tw-Tc+4*T_nl), c=Tas-Tc
26 %      X = 1/(2*a)*(-b-sqrt(b^2-4*a*c))
27 %      This is one of the two solutions, but a second solution is not physical
28 %      Cs = X*(Cw-Cc) + Cc,                                     (3)
29 %      Tas often have more cross-track scans, and Tc,Tw,Cc,Cw have less scans
30 %      and are averaged along cross-track and applied to Tas
31 %
32 % written by John Xun Yang, University of Maryland, jxyang@umd.edu, or
33 % johnxun@umich.edu, 01/31/2019: original code
34
35 global Rad
36
37 if Rad.nonlinear.onoff ==1
38
39 % -----
40 % setting
41 % channel-dependent peak nonlinearity TB, [1,channel]
42 % -----
43 T_nl = 5*ones(1,Rad.num_chan);
44
45 % -----
46 % parse
47 % -----
48 saterr_parse_nonlinear
49
50 end

```

To turn on the reflector nonlinearity, you first set `Rad.nonlinear.onoff = 1` as in line 35. Then, set the magnitude of nonlinearity in line 43. This parameter is channel dependent. Details about the nonlinearity definition and derivation are given at the header of the script. The script `saterr_parse_nonlinear` in line 49 is for parsing the input and assuring the input parameters such as variable dimensions are correct.

3.3 Setting Modes

The comprehensive error inventory can be set with turning on/off individual sources and specify their combination. There are a few common modes users may want to work with. Table 1 lists three modes.

- **The Nominal Mode.**

This refers to in-orbit spacecraft working in the nominal mode. The sensor observes Earth, with spillover seeing the cosmic background. In this mode, a couple of modules are turned on, including the orbit module, RTM, etc.

- **The Maneuver Mode**

This mode is similar to the nominal mode, but the spacecraft is flipped over with the sensor looking at the cosmic background and the spillover pointing to Earth. This can be made by setting the targets of mainlobe, sidelobe and spillover.

- **The TVAC Mode**

This is for ground Thermal Vacuum Chamber (TVAC) test. The scene target can be set to a uniform scene, with the RTM module turned off.

Setting	Nominal	Maneuver	TVAC
Orbit (saterr_orbit.m)	On	On	Off
RTM (saterr_set_geoorbit.m)	On	On	Off
Scene Target (saterr_set_modefarfieldTB.m)	On (Earth)	On (Cosmic)	On (Uniform Scene)

Table 1: The observation error inventory in the simulator

3.4 Output and Visualization

The output can have a range of files encompassing data and figures. The output also dependent on what the setting and running modes are. The standard output data include the TOA T_B , counts, calibrated T_A and T_B , noise, etc. A number of figures are generated for visualization of these variables. Figures 4,5,6,7 shows examples of simulated orbit, T_A , noise (total noise and 1/f noise percentage) for MetOp-A AMSU-A.

3.5 Advanced Setting

After getting familiar with the simulator, users can try more advanced setting. Among them include combining a number of error sources. For setting most of error source, go to `saterr_main_setting.m`.

Note that there can be dependency for setting specific error sources. For example, the attitude offset (`saterr_set_attitudeoffset.m`) has dependency for satellite orbit setting (`saterr_set_geoorbit.m`). The dependency requirement is straightforward to find. In the afore-said case, it is described in the `saterr_set_attitudeoffset.m`. In the Appendix, the lists of error sources and scripts are given, respectively, where users can look for settings of interest. Detailed help can be found in the header of each script.

SatERR can be used for extensive simulation such as for processing one-year satellite data with ancillary reanalysis data. Given a Linux OS with a number of computing nodes and cores, users can break down the computing into many sub-task. For example, the one-year data can be subset to different date ranges, with one core for dealing with a specific time range. The code can also be set to break down a large orbital data (of a hundred minutes) into many granule data (of a couple of minutes) to fasten computing with multiple cores. We provide scripts for generating batch job

submitting scripts such as in accordance with qsub, allowing for running hundreds of jobs at the same time.

3.6 Sequence of Execution

There are two major ways of executing the program. For light simulation such as studying error sensitivity, features, etc, the execution sequence is as follows. This example shows one-day simulation from 01/01/2021 to 01/02/2021.

- `saterr_main_('step1','20210101','20210102')`

Running the simulation and generating TOA T_B , counts. Other variables like geolocation, noise are also generated if they are enabled in the setting. `saterr_main_('step1')`

The execution can be speed up by subsetting task into sub time ranges and running them at the same time. For example, `saterr_main_('step1','20210101','20210102')` is for simulation from '20210101' – '20210102', i.e. '20210101'≤date range<'20210102'.

`saterr_main_('step1','20210101','20210102')` is for the next day. The two time ranges can be run at the same time, with one core for each job.

- `saterr_main_('step2','20210102','20210103')`

Calibrating counts to T_A and T_B . Figures are plotted for visualizing a range of parameters.

The second scenario is for heavy simulation working in computer clusters with abundant computing resources. Running the program with dozens, hundreds or thousands of CPU cores. This kind of extensive simulation is suggested to work in the following way. For a task of one-month simulation from 01/01/2021 – 01/31/2021, we go through steps below:

- `saterr_main_('step1_a','20210101','20210131')`

Preprocessing data such as collocating satellite data and reanalyses. Orbital satellite data (e.g. a hundred minutes) can set to split into smaller granule data (e.g. a couple of minutes) for speeding up simulation later.

- `saterr_main_('step1_b_queue','20210101','20210131')`

Writing bash scripts for submitting jobs such as through qsub in Linux servers. An example bash file, `q_step1.bash`, is generated as follows:

```

1 #!/bin/bash
2 #PBS -l walltime=24
3 #PBS -l nodes=1:ppn=24
4 cd ~/errsat/
5 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[1
   1000])" > log/log_1.txt </dev/null
   &
6 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[1
   1001 2000])" > log/log_2.txt </dev/
   null &
7 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[1
   2001 3000])" > log/log_3.txt </dev/
   null &
8 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[1
   3001 4000])" > log/log_4.txt </dev/
   null &
```

```

9 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[  

10    4001 5000])" > log/log_5.txt </dev/  

11    null &  

12 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[  

13    5001 6000])" > log/log_6.txt </dev/  

14    null &  

15 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[  

16    6001 7000])" > log/log_7.txt </dev/  

17    null &  

18 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[  

19    7001 8000])" > log/log_8.txt </dev/  

20    null &  

21 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[  

22    8001 9000])" > log/log_9.txt </dev/  

23    null &  

24 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[  

25    9001 10000])" > log/log_10.txt </  

26    dev/null &  

27 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[  

28    10001 11000])" > log/log_11.txt </  

29    dev/null &  

30 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[  

31    11001 12000])" > log/log_12.txt </  

32    dev/null &  

33 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[  

34    12001 13000])" > log/log_13.txt </  

35    dev/null &  

36 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[  

37    13001 14000])" > log/log_14.txt </  

38    dev/null &  

39 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[  

40    14001 15000])" > log/log_15.txt </  

41    dev/null &  

42 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[  

43    15001 16000])" > log/log_16.txt </  

44    dev/null &  

45 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[  

46    16001 17000])" > log/log_17.txt </  

47    dev/null &  

48 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[  

49    17001 18000])" > log/log_18.txt </  

50    dev/null &  

51 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[  

52    18001 19000])" > log/log_19.txt </  

53    dev/null &  

54 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[  

55    19001 20000])" > log/log_20.txt </  

56    dev/null &  

57 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[  

58    20001 21000])" > log/log_21.txt </  

59    dev/null &  

60 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[  

61    21001 22000])" > log/log_22.txt </  

62    dev/null &  

63 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[  

64    22001 23000])" > log/log_23.txt </  

65    dev/null &

```

```

28 matlab -nodisplay -nodesktop -r "errsat('step1_b','20210101','20210131',[  

29      23001 24150])" > log/log_24.txt </  

30      dev/null &

```

where we specify using 1 node with 24 cores to run simulation from 01/01/2021 – 01/31/2021 with a total of 24150 granule files. The walltime is 24 hours. These granule files are assigned to 24 cores with granule range specified in the last parameter of command line. The first 1 – 23 cores deal with 1000 files, respectively. The last core processes granule files from 23001 – 24150.

- `saterr_main_('step1_b','20210101','20210131')`
`saterr_main_('step1_b','20210101','20210131',[1,1000])`
RTM simulation with many CPU cores. The first command does all the simulation from 01/01/2021 – 01/31/2021.
The second command, `saterr_main_('step1_b','20210101','20210131',[1,1000])`, processes a portion of granule files from 1 – 1000.

You can submit a bash script with qsub. In a Linux environment with proper setup, the command works:

```
qsub q_step1.bash
```

where the `qsub q_step1.bash` is generated in the previous `step1_b_queue`

- `saterr_main_('step1_c','20210101','20210131')`
Merging granule simulation into orbital data.
- `saterr_main_('step1_d','20210101','20210131')`
Converting TOA T_B to counts. Likewise, we can split the time range and run with multiple cores to speed up. Below is for using three cores
`saterr_main_('step1_d','20210101','20210110')`
`saterr_main_('step1_d','20210110','20210120')`
`saterr_main_('step1_d','20210120','20210131')`
- `saterr_main_('step2','20210101','20210131')`
Calibrating counts to T_A and T_B . Figures are plotted for visualizing a range of parameters. This process can also speed up with multiple cores.

The above procedures are for simulating one satellite in terms of counts, T_A, T_B . We can compare simulation and observation for one satellite or implement intercalibration for multiple satellites:

- `saterr_main_('step3_oo','20210101','20210131')`
Comparing two satellite observations.
- `saterr_main_('step3_sd','20210101','20210131')`
Comparing observations and simulation for one satellite. The single difference is analyzed.
- `saterr_main_('step3_dd','20210101','20210131')`
Intercalibrating two satellites and calculating parameters like double difference.

3.7 Simulating a New Sensor

SatERR can be extended for a new sensor. Suppose you want to customize the setting for a new small satellite with a low inclination orbit and cross-track scanning.

The crosstrack scanning can be set in `saterr_set_scanning.m`. Part of the script is shown below. The case ‘customize’ can be modified to set the scanning parameters. Alternatively, a new case can be added.

```

1 % -----
2 % setting
3 % -----
4 switch Rad.sensor
5   case 'customize'
6     Rad.scan.angle_res = 1.11; % angular resolution of scan (degree)
7     Rad.scan.angle_center = [0,73.6,180]; % angular center for cs,cc,cw;
8       left-hand pointing to forward; nadir 0,[-180,180]
9     Rad.scan.period = 8/3; % scanning period per rotation (second)
10    Rad.scan.time_integ = 0.018; % sampling integration time (second)
11
12  case {'demo'}
13    Rad.scan.angle_res = 1.11; % angular resolution of scan (degree);
14    Rad.scan.angle_center = [0,73.6,180]; % angular center for cs,cc,cw;
15      left-hand pointing to forward; nadir 0,[-180,180]
16    Rad.scan.period = 8/3; % scanning period per rotation (second)
17    Rad.scan.time_integ = 0.018; % sampling integration time (second)
18
19  case 'mhs'
20    Rad.scan.angle_res = 1.11; % angular resolution of scan (degree)
21    Rad.scan.angle_center = [0,73.6,180]; % angular center for cs,cc,cw;
22      left-hand pointing to forward; nadir 0,[-180,180]
23    Rad.scan.period = 8/3; % scanning period per rotation (second)
24    Rad.scan.time_integ = 0.018; % sampling integration time (second)
```

Likewise, the channel specification can be added in the script, `saterr_set_radspc_chan.m`

```

1 switch Rad.sensor
2
3   case 'customize'
4     Rad.chanfreq = {23.8,31.4,183.311-1}; % frequency (GHz)
5     Rad.chanpol = {'QV','QV','QH'};
6     Rad.chan_freq_nominal = {'23.8','31.4','183.311-1'}; % channel nominal
7       frequency (GHz), unique name
8     Rad.num_chan = length(Rad.chanfreq); % channel number
9
10  case {'demo'}
11    Rad.chanfreq = {23.8}; % channel frequency (GHz); can have subband
12    Rad.chanpol = {'QV'}; % channel polarization
13    Rad.chan_freq_nominal = {'23.8'}; % channel nominal frequency (GHz),
14      unique name
15    Rad.num_chan = length(Rad.chanfreq); % channel number
16
17  case {'amsr2'}
18    Rad.chanfreq = {6.925,[6.925],[89]}; % channel frequency (GHz); can have
19      subband
```

```

17     Rad.chanpol = {'V','H','V'};
18     Rad.chan_freq_nominal = {6.925,6.925,89}; % channel nominal
19         frequency (GHz), unique name
      Rad.num_chan = length(Rad.chanfreq); % channel number

```

The orbit can be set in the script, `saterr_set_geoorbit.m`

```

1 Orbit.onoff = 1; % orbit setting, 0=off,1=on;
2
3 if Orbit.onoff == 1
4     Orbit.type = 'Keplerian'; % Keplerian/Real_SC/Real_SCFOV/Real_FOV/
5         Real_AllFromObs
6
7     if strcmp(Path.scheme,'B')
8         Orbit.type = 'Real_SCFOV'; % Real_SC/Real_SCFOV/Real_FOV; Real_SCFOV can
9             study attitude;
10    end
11
12    switch Orbit.type
13        case 'Keplerian'
14            % setting Keplerian orbit
15            errsat_set_geoorbit_kepler
16
17        case 'Real_SC'
18            % setting spacecraft geolocation
19            errsat_set_geoorbit_real_sc
20
21        case {'Real_SCFOV','Real_FOV'}
22            % setting spacecraft and footprint FOV geolocation
23            errsat_set_geoorbit_real_scfov
24
25        case {'Real_AllFromObs'}
26            % setting spacecraft and footprint FOV geolocation
27            errsat_set_geoorbit_real_allfromobs
28    end
29
30 end

```

The radiometer related specification can be set in the script, `saterr_set_radspc.m`

```

1 % -----
2 % setting
3 % -----
4 switch Rad.sensor
5
6     case 'customize'
7         Rad.scantype = 'crosstrack'; % crosstrack/conical
8         Rad.name_crosstrack = {'cs','null','cc','null','cw','null'}; % cc=cold
9             count,cw=warm-load count,cs=scene count,null=gaps between cc,cw,cs
10            Rad.num_crosstrack = [90,4,4,19,4,23]; % number of scanning positions in
11                one rotation w.r.t. name_crosstrack
12                Rad.num_alongtrack_1orbit = 2342; % number of alongtrack scanlines/
13                    rotations of one orbit
14                    Rad.num_orbit = 1; % number of orbit; total scanlines/rotations =
15                        num_orbit*num_alongtrack_1orbit

```

```

12
13 case {'demo'}
14     Rad.scantype = 'crosstrack'; % crosstrack/conical
15     Rad.name_crosstrack = {'cs','null','cc','null','cw','null'}; % cc=cold
16         count,cw=warm-load count,cs=scene count,null=gaps between cc,cw,cs
17     Rad.num_crosstrack = [90,4,4,19,4,23]; % number of scanning positions in
18         one rotation w.r.t. name_crosstrack
19     Rad.num_alongtrack_1orbit = 2342; % number of alongtrack scanlines/
20         rotations of one orbit
21     Rad.num_orbit = 1; % number of orbit; total scanlines/rotations =
22         num_orbit*num_alongtrack_1orbit
23
24 case {'amsr2'}
25     Rad.scantype = 'conical'; % crosstrack/conical
26     Rad.name_crosstrack = {'cs','null','cw','null','cc','null'}; % cc=cold
27         count,cw=warm-load count,cs=scene count,null=gaps between cc,cw,cs
28     Rad.num_crosstrack = [243,101,16,101,16,101]; % number of scanning
29         positions in one rotation w.r.t. name_crosstrack;
30         [243,108,16,108,16,108] 6-36;
31     Rad.num_alongtrack_1orbit = 1000;%3956; % number of alongtrack scanlines
32         /rotations of one orbit
33     Rad.num_orbit = 1; % number of orbit; total scanlines/rotations =
34         num_orbit*num_alongtrack_1orbit
35
36 case 'amsu-a'

```

More advanced setting can be found in relevant scripts, with guidance at the headers.

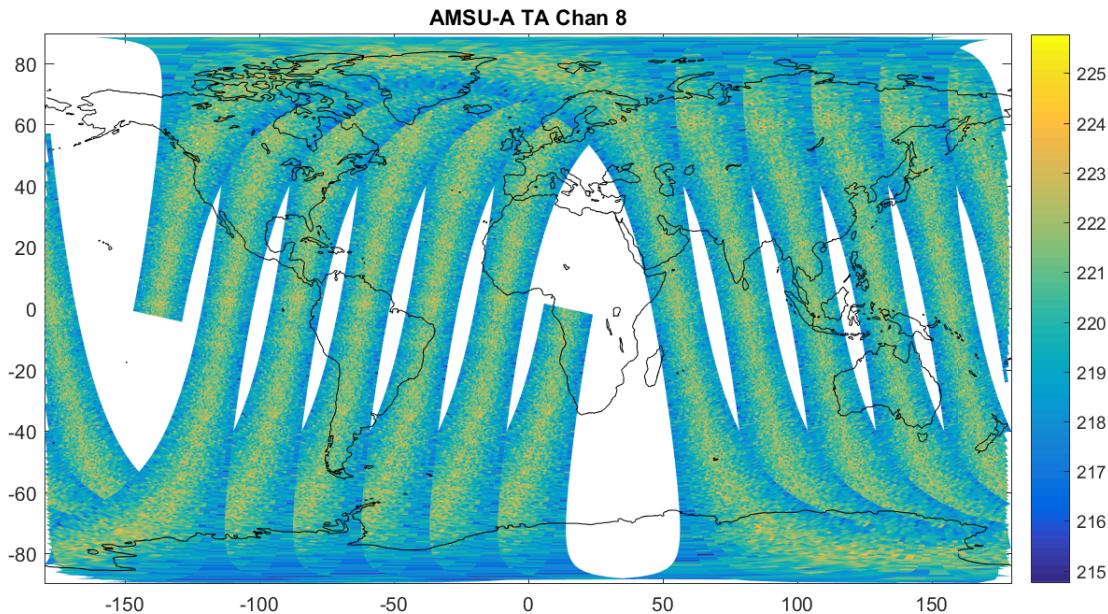


Figure 4: The Metop-A AMSU-A T_A of channel 8, with the simulated orbit and scanning. The atmospheric profile is the Standard U.S. Atmosphere, and thus there is no spatial variation. But ancillary data such as ERA5, GDAS can be used for more realistic simulation.

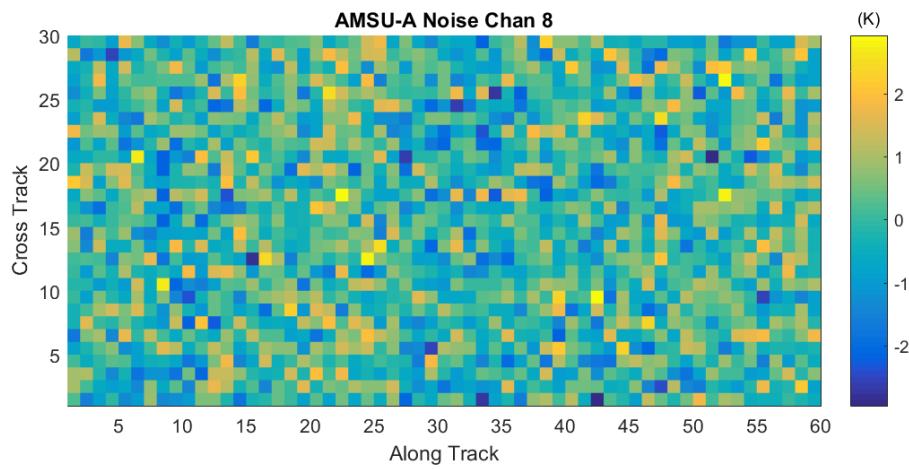


Figure 5: A swath of channel 8 noise of MetOp-A AMSU-A.

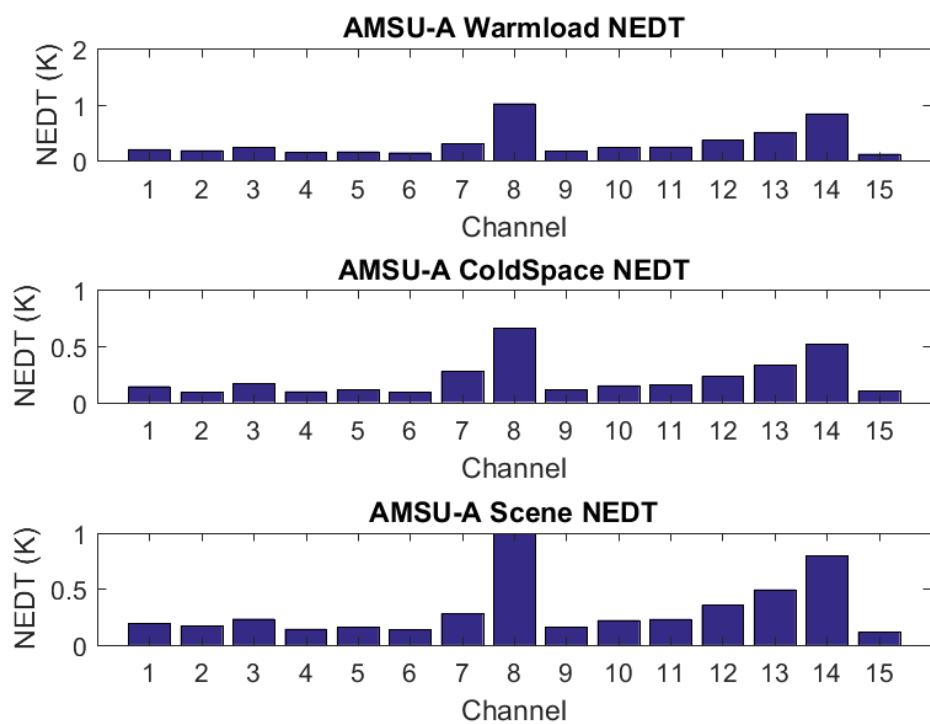


Figure 6: NEDT of MetOp-A AMSU-A.

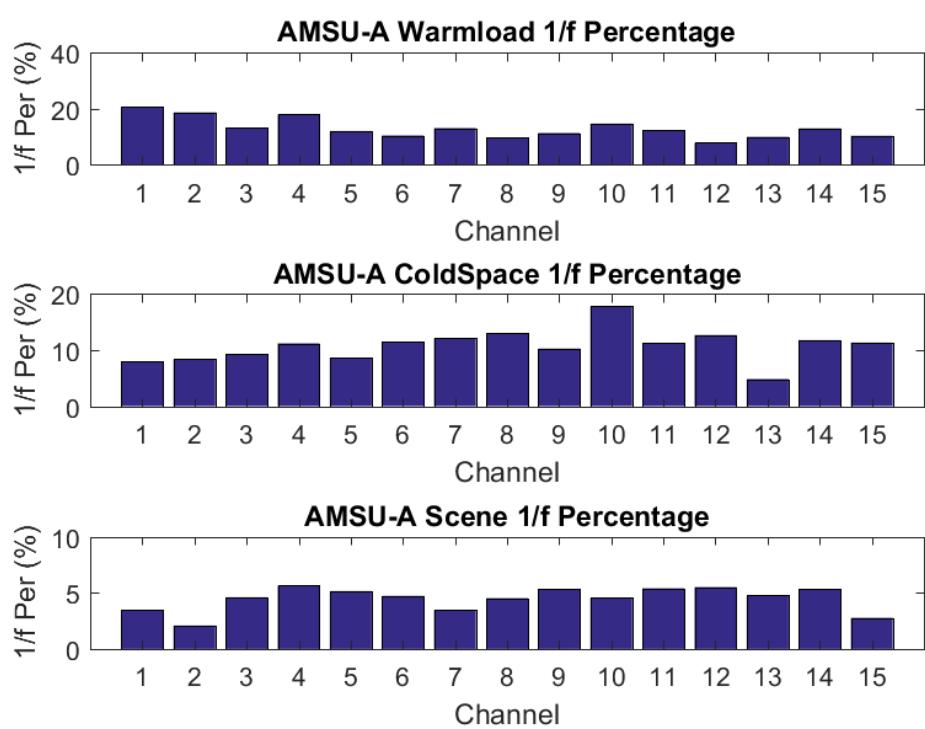


Figure 7: $1/f$ noise percentage.

4 Observation Error Inventory

4.1 Algorithm Theoretical Basis

The algorithm theoretical basis is present in this section. We describe what the algorithms and models are used for different error sources. These models are from a collection of different work, with the literature cited. For error sources that are straightforward, we skip the description in this manual, but more details can be found in the header of the codes. The diagram of SatERR structure is illustrated in Figure 8.

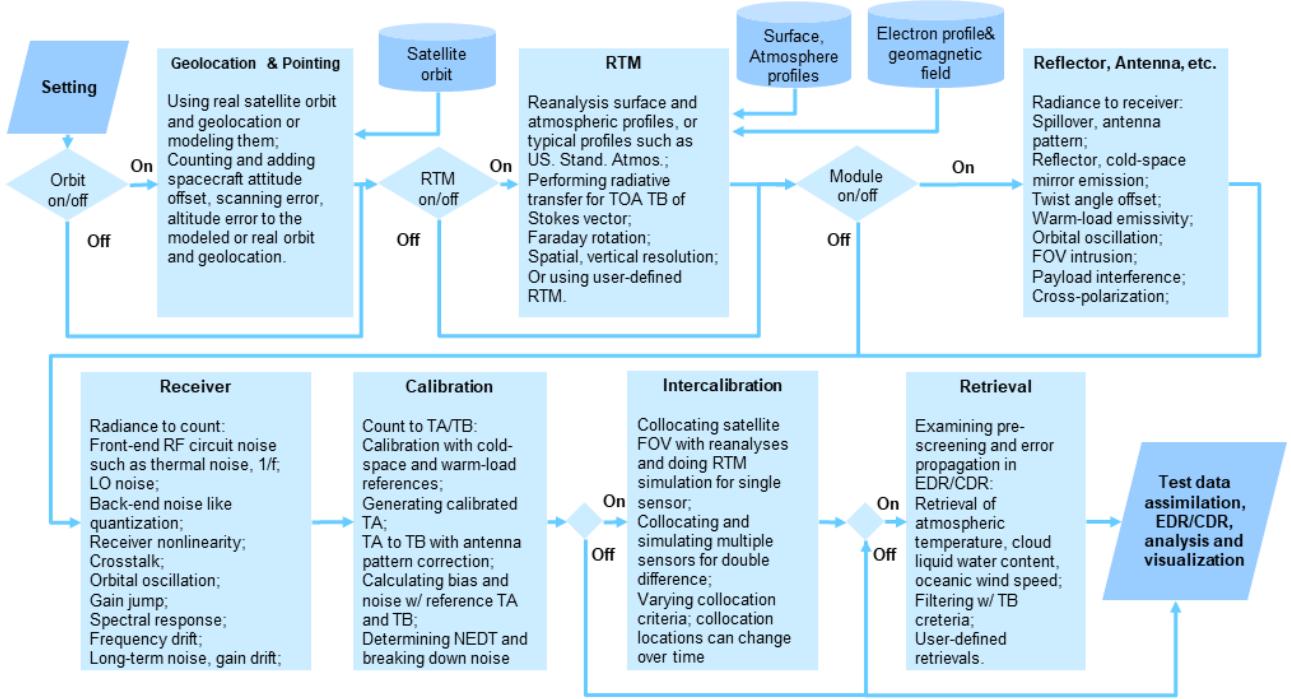


Figure 8: SatERR structures and modules with error sources described.

4.2 Measurement error

4.2.1 Radiometer diagram

Figure 9 show the diagram of a typical radiometer. The reflector and antenna collect incoming radiance that enters the feedhorn followed by diplexers or orthomode transducers (OMT) for different channels. When the radiometer operates in the nominal mode, the radiance alternates between the cold-space, warm load, and Earth scenes, as the reflector spins. A ranges of errors can take place in this part as in the diagram and we will detail them shortly.

The receiver front-end comprises a cascade of a low-noise amplifier (LNA), LO if needed, and band-pass filters (BPF) where the incoming radiance gets amplified and filtered. During the process, electronic noise and other factors can affect the radiometer performance.

The receiver back-end is the video amplifying and recording. It has a square-law detector, video amplifier, and analog to digital converter (ADC). After data downlinking, the raw data are processed into level-1 T_A and T_b as well as other science data in the ground data center.

The small satellites that are emerging in recent years can have a slightly different design. For instance, they may use a noise diode as the warm reference rather than the blackbody. SatERR can still accommodate to simulating small satellites, with the warmload error source substituted.

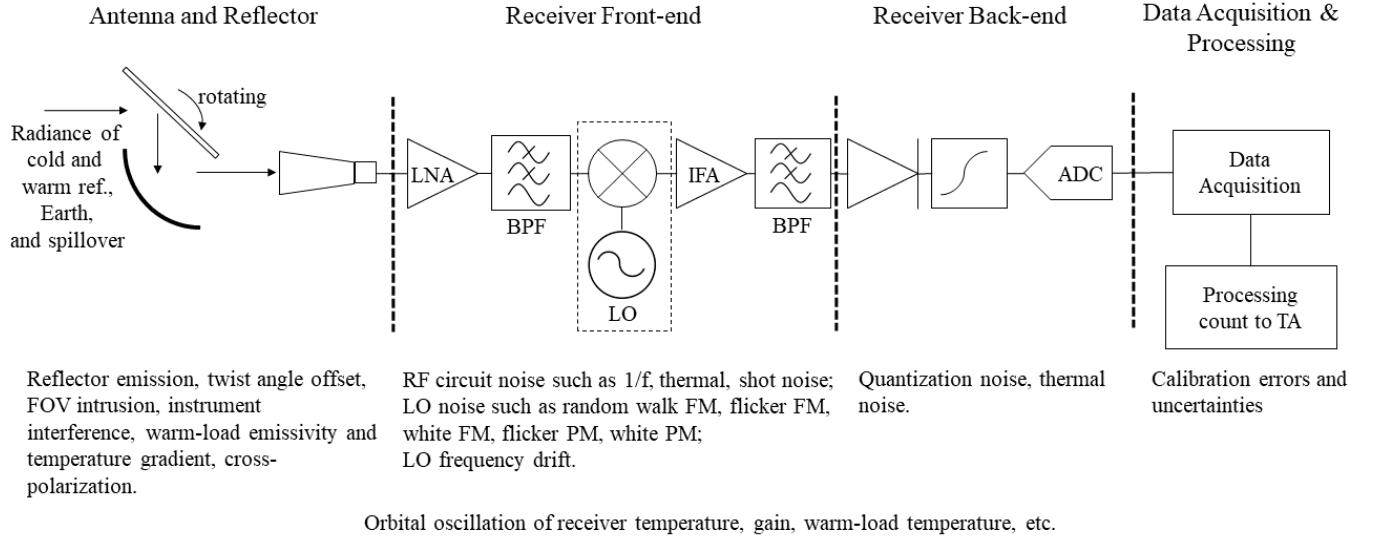


Figure 9: Diagram of a radiometer with noise and error sources illustrated.

4.2.2 Generating count, T_A with noise and orbital oscillation

The receiver processes radiance to counts. With calibration, the count is converted to T_A . Noise from radio frequency (RF) circuit is accompanied with the process.

The noise-free count of a radiometer can be written as [Ulaby *et al.*, 2014]

$$C = (T_A + T_R)G \quad (1)$$

where C is count, T_A is antenna temperature, T_R is receiver temperature, G is the power gain. The system temperature can be defined as $T_{sys} = T_A + T_R$. In the presence of noise, it can be written as

$$C = (T_A + T_R + \sum n_i)G \quad (2)$$

where n_i represents a number of aforementioned individual noise, including both additive and signal-dependent noise. The short-term gain fluctuation also results in noise and can be implicitly counted as an individual n_i . The long-term gain oscillation subject to orbital variation is different from short-term fluctuation and will be introduced shortly.

The noise term n_i and its generating can be associated with two families of additive and signal-dependent noise. The additive noise is written as

$$x = s + n \quad (3)$$

where x is the measurand, s is the signal, n is the additive noise that is signal independent. Figure 10 illustrates a couple of noise in the time and frequency domain.

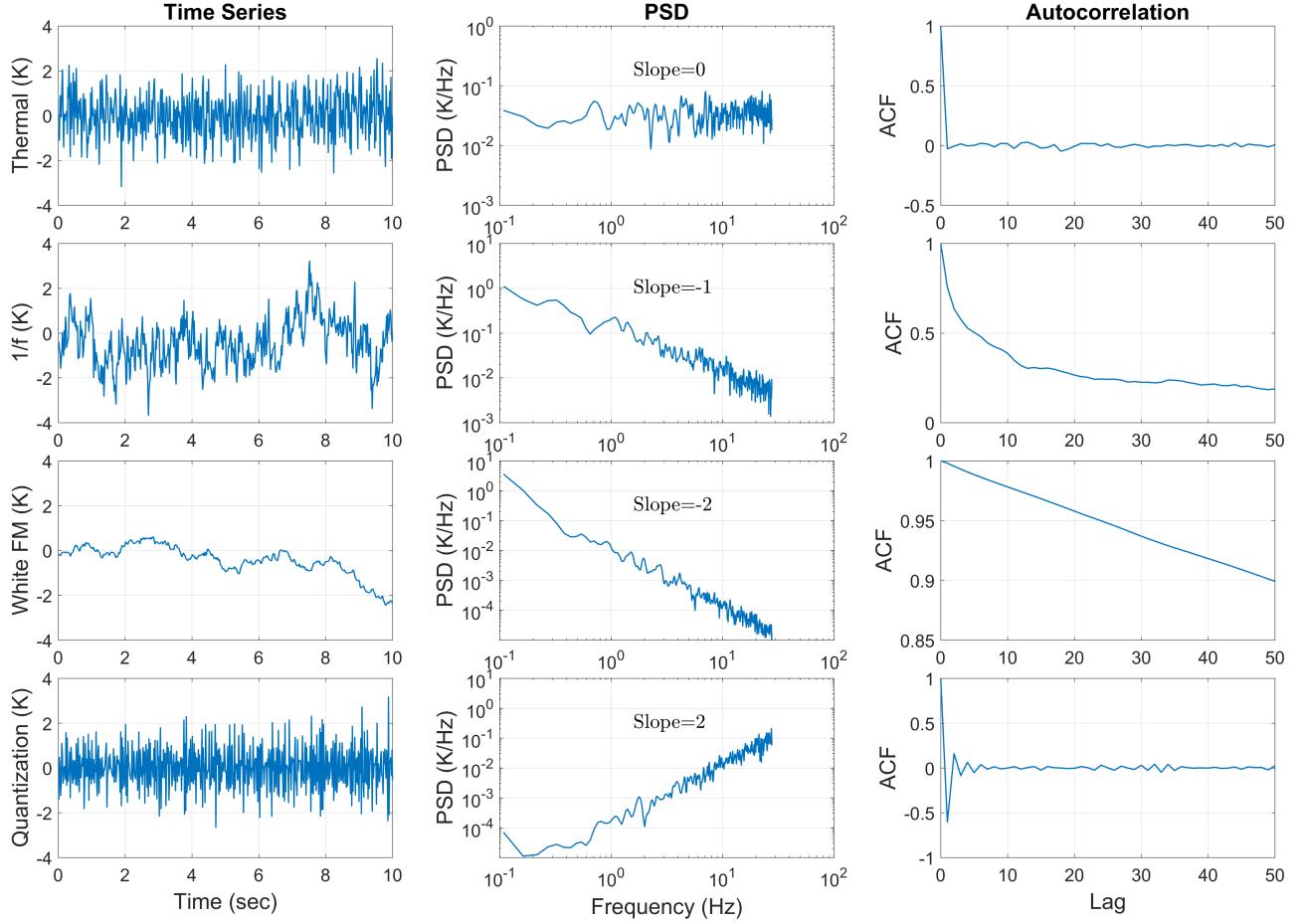


Figure 10: Examples of simulated noise in time series, power spectrum, and auto-correlation. These include thermal noise, 1/f, white FM, and quantization from up to down, showing different characteristics in the time and frequency domain.

Thermal noise is due to the random motion of agitated charger carriers [Lee, 2004; Vasilescu, 2005; Schiek *et al.*, 2006]. The additive model is used to generate thermal noise. Thermal noise is treated as additive white Gaussian noise (AWGN) and generated with a pseudo random number generator (RNG). The default RNG is the Mersenne Twister and AWGN is assumed to be zero-mean [Matsumoto and Nishimura, 1998]. The noise generating is different in different runs. Under certain circumstances, the same noise needs to be generated for testing the sensitivity of other error sources, which can be realized by fixing the RNG and its seed.

1/f noise is also called flicker noise or excess noise. It can arise from different mechanisms. It is found pronounced in electronics sensitive to surface phenomenon related to charge trapping and releasing and carrier generation and recombination [Lee, 2004; Vasilescu, 2005; Schiek *et al.*, 2006]. 1/f noise is produced with Fast Fourier Transform (FFT) [Sullivan *et al.*, 1990; Riley, 2008]. We first generate AWGN in the time domain. FFT transform is applied and multiplied with a term of power-law exponent of -1. The inverse FFT is applied for random numbers back in the time domain

$$\begin{aligned} h(\omega) &= \mathcal{F}(n_{\text{AWGN}}(t)) \\ n(t) &= \mathcal{F}^{-1}(h(\omega)f^\alpha) \end{aligned} \tag{4}$$

where the first equation denotes the Fourier transform of $n_{\text{AWGN}}(t)$, and the second one is the inverse transform with an additional power law term $h(\alpha)f^\alpha$. For 1/f noise, $\alpha = -1$.

Phase and amplitude noise can come from a LO. This is very common for V/W/G bands, as LO performs heterodyning. A category of power-law noise can be produced, including the random walk FM noise ($\alpha = -4$) related to oscillator environment, flicker FM ($\alpha = -3$) from resonance mechanism, white FM ($\alpha = -2$) from resonator frequency standards, flicker PM ($\alpha = -1$) due to noisy electronics, white PM ($\alpha = 0$) from phase noise in stages of amplification [Sullivan *et al.*, 1990]. The causes of noise are complex in practice and can come from multiple mechanisms. Oscillator noise is generated the same way with the FFT method.

Quantization noise is due to ADC with $\alpha = 2$ [Sullivan *et al.*, 1990; Schiek *et al.*, 2006]. It is also generated with the FFT method. Other power-law noise can also be generated in the simulator based on FFT transform with positive α [Riley, 2008].

The signal-dependent noise is produced in a different way. Shot noise is due to the discrete nature of charge, and it is Poisson noise that depends on the signal [Lee, 2004]. It is generated with a Poisson RNG as [Kuan *et al.*, 1985]

$$n_{\text{shot}} = P_\lambda(T_{\text{sys}}) \quad (5)$$

where shot noise is a function of T_{sys} and P_λ is the Poisson RNG with λ as the Poisson distribution parameter.

The payload environment usually has an orbital oscillation as the spacecraft orbits the Earth [Gaiser *et al.*, 2004; Yang *et al.*, 2015; Piepmeier *et al.*, 2017; Yang and Yang, 2021]. It is due to the change of solar incidence angle and environment temperature along the orbit. The radiometer gain and warm-load temperature oscillate with a period same as the spacecraft orbital period [Yang and Yang, 2021]. For most sun-synchronous satellites, the orbital period is ~ 100 minutes. The orbital oscillation can be described as

$$\begin{aligned} T_W &= T_{W0} + \Delta T_W(t, \lambda, a, \phi) \\ G &= G_0 + \Delta G(t, \lambda, a, \phi) \end{aligned} \quad (6)$$

where T_{W0} and G_0 are constant, and ΔT_W and ΔG are oscillation in terms of waveform as a function of time t , wavelength λ , amplitude a , and phase ϕ . A variety of waveforms can be simulated. We can also import empirical orbital oscillations from observations as the oscillation term. In this study, we use sinusoidal wave with the empirical magnitudes from observation [Gaiser *et al.*, 2004].

A radiometer observes both the cold and warm reference targets and the Earth scene during one complete rotation of the reflector. We generate counts as follows

$$\begin{aligned} C_W &= (T_{A,W} + T_R + n_W)G \\ C_C &= (T_{A,C} + T_R + n_C)G \\ C_S &= (T_{A,S} + T_R + n_S)G \end{aligned} \quad (7)$$

where $T_{A,W}$, $T_{A,C}$, and $T_{A,S}$ are the warm-load, cold-space, and Earth scene temperature, respectively. n_W , n_C , and n_S are corresponding noise, and C_C , C_W , C_S are counts. If the warm-load is a perfect blackbody, there is $T_{A,W} = T_W$. And $T_{A,C}$ equals the cosmic temperature T_C , if the cold-space view has no contamination from the reflector emission or field of view (FOV) intrusion.

The scanning mechanism of a radiometer determines the way of data sampling that should be counted in simulation. Figure 11 shows the diagram of MHS scan. As the reflector spins, the

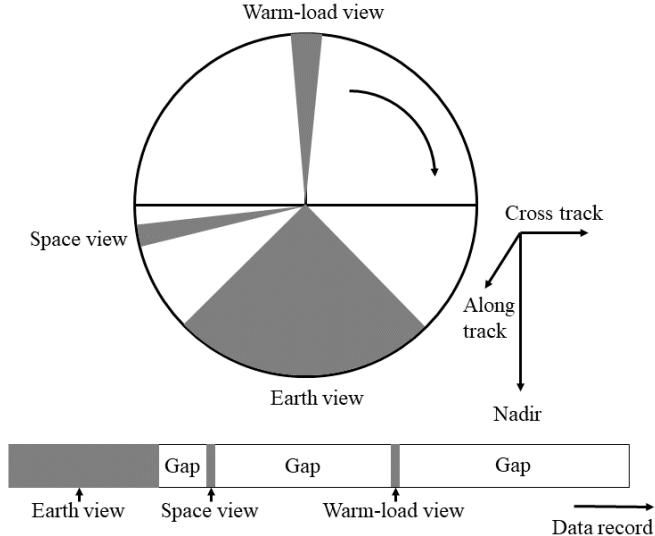


Figure 11: The scanning mechanism of MHS. In the nominal operating, the receiver sees the Earth, cold space, and warm load in turn with gaps in between as the reflector rotates. The duty cycle of the effective target views is about 68%. The data records are discrete chunks in time series, which is a conditional sampling. The data streaming is counted in the simulator.

radiance received by the receiver alternates between the cold space, warm load, and Earth views. In between the effective objects, the radiometer is still working but sees the radiometer itself and the data are not usable. MHS has a rotation period of 2.67 seconds with a duty cycle of 68% for viewing Earth and reference targets [Robel and Graumann, 2014].

The scanning produces data as a conditional sampling with only a portion of data used. The recorded data are discrete chunks with gaps in between, and so is the noise. The conditional sampling has a profound influence. For instance, 1/f noise has a non-stationary mean that changes with time. The 68% duty cycle means longer time ($1/0.68$) and larger variability for the same amount of data with 1/f noise compared to that without gaps. In addition, the discrete data with gaps can result in a difference between population statistics and sample statistics. Our simulator counts the scanning and sampling.

We simulate the null count as

$$C_N = (T_{A,N} + T_R + n_N)G \quad (8)$$

where C_N , $T_{A,N}$, and n_N denote the null count, scene temperature, and noise, respectively. The null simulation is not used but better represents MHS work mode of a conditional sampling.

After generating count for scene, cold and warm references, T_A is calculated

$$T_{A,S} = \frac{(T_w - T_c)}{(C_w - C_c)}(C_s - C_c) + T_c \quad (9)$$

where $T_{A,S}$ is the scene temperature, T_w for warm load temperature, T_c is cold space temperature. C_w , C_c , C_s are count for warm load, cold space, and scene respectively.

If nonlinearity is considered, there is [Robel and Graumann, 2014]

$$T'_{A,S} = T_{A,S} + 4T_n \frac{(C_s - C_c)}{(C_w - C_c)} \left[1 - \frac{(C_s - C_c)}{(C_w - C_c)} \right] \quad (10)$$

where $T'_{A,S}$ is after nonlinear correction, $T_{A,S}$ is from linear calibration, T_n is the term of peak nonlinearity that is often a look-up table from prelaunch test.

4.2.3 Windowing and Calibration

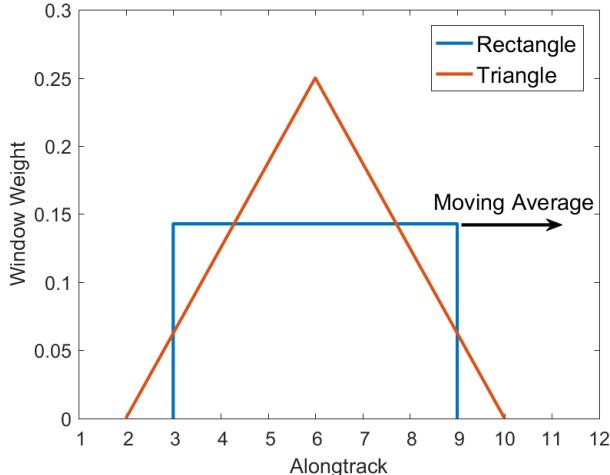


Figure 12: Examples of rectangular and triangular windows. The windows span a finite along-track length and are used for smoothing radiometer counts in deriving gain and brightness temperature.

Windowing is used for smoothing cold-space and warm-load counts. The cold-space counts are smoothed as

$$\overline{C_{Cj}} = \frac{1}{n_{cc}} \sum_{k=j-n_1}^{j+n_2} w_k \sum_{i=1}^{n_{cc}} C_{Ci,j}, \quad (11)$$

where $C_{Ci,j}$ is the cold-space counts as a function of scan positions in terms of cross-track i and along-track j . We refer cross-track measurements as *samples* and along-track measurements as *scans* hereafter. n_{cc} is the number of cross-track samples of cold counts. For example, ATMS and MHS have $n_{cc} = 4$. n_1 and n_2 are related to the window length L through the floor function as

$$n_1 = \lfloor \frac{L-1}{2} \rfloor \quad \text{and} \quad n_2 = \lfloor \frac{L}{2} \rfloor. \quad (12)$$

w_k is a window function with a length of L for smoothing counts along track. w_k can be a rectangular window as

$$w_k = \frac{1}{L}, \quad 0 \leq k \leq L-1. \quad (13)$$

A triangular window is also often used as

$$w_k = \begin{cases} \frac{2}{L+1} \left(1 - \frac{|2k-L+1|}{L+1}\right), & 0 \leq k \leq L-1, \text{ odd } L, \\ \frac{2}{L} \left(1 - \frac{|2k-L+1|}{L}\right), & 0 \leq k \leq L-1, \text{ even } L. \end{cases} \quad (14)$$

A triangular window of $L = 7$ has been used for all MHS and AMSU-A/B at EUMETSAT and NOAA [Mo, 1996; Weng et al., 2013; EUM, 2013b, a; Robel and Graumann, 2014]. Figure 12 shows examples of triangular and rectangular windows.

Likewise, $C_{W_{i,j}}$ is smoothed

$$\overline{C_{W_j}} = \frac{1}{n_{cw}} \sum_{k=j-n_1}^{j+n_2} w_k \sum_{i=1}^{n_{cw}} C_{W_{i,j}}, \quad (15)$$

where n_{cw} is the cross-track number of warm-load counts.

The warm-load PRT temperature, from the average of multiple PRTs, is smoothed as

$$\overline{T_{W_j}} = \sum_{k=j-n_1}^{j+n_2} w_k T_{W_j}. \quad (16)$$

Then, the power gain $\overline{G_{S_j}}$ is given by

$$\overline{G_j} = \frac{\overline{C_{W_j}} - \overline{C_{C_j}}}{\overline{T_{W_j}} - T_C}, \quad (17)$$

where T_C is the cosmic temperature.

Earth scene antenna temperature is

$$T_{A,S_{i,j}} = \frac{1}{\overline{G_j}} (C_{S_{i,j}} - \overline{C_{C_j}}) + T_C, \quad 1 \leq i \leq n_{cs}. \quad (18)$$

where n_{cs} is the cross-track number of Earth scene.

In addition to Earth scene, it is useful to derive warm-load and cold-space antenna temperature for determining the Noise Equivalent Delta Temperature (NEDT) and decomposing noise. In contrast to the cold-space counts, we split the cross-track warm-load counts into two subsets for deriving the gain and warm-load antenna temperature $T_{A,W}$. Doing so assures the independency between the derived gain and $T_{A,W}$, avoiding generating a pseudo f^2 in the frequency domain due to the inappropriate signal processing. A pseudo f^2 noise spectrum underestimates 1/f noise since its low-frequency regime is ignored [Yang and Yang, 2021]. We suggest reader refer to [Yang and Yang, 2021; Yang et al., 2021] for details.

We first smooth half of cross-track $C_{W_{i,j}}$ as

$$\overline{C_{W_{i,j}}} = \frac{2}{n_{cw}} \sum_{k=j-n_1}^{j+n_2} w_k \sum_{i=1}^{\frac{n_{cw}}{2}} C_{W_{i,j}}, \quad 1 \leq i \leq \frac{n_{cw}}{2}, \quad (19)$$

where n_{cw} is number of cross-track warm counts, which is four for MHS. Then, the power gain $\overline{G_{CW_j}}$ is given by

$$\overline{G_{CW_j}} = \frac{\overline{C_{W_{i,j}}} - \overline{C_{C_j}}}{\overline{T_{W_j}} - T_C}, \quad 1 \leq i \leq \frac{n_{cw}}{2}, \quad (20)$$

where T_C is the cosmic temperature.

The other half of cross-track $C_{W_{i,j}}$ is used for deriving the warm-load antenna temperature $T_{A,W}$

$$T_{A,W_{i,j}} = \frac{1}{\overline{G_{CW_j}}} (C_{W_{i,j}} - \overline{C_{C_j}}) + T_C, \quad \frac{n_{cw}}{2} + 1 \leq i \leq n_{cw}. \quad (21)$$

4.2.4 Decomposing Noise

In the above section, we obtain the cold-space or warm-load antenna temperature. We proceed to derive the noise. The orbital oscillations is removed with the warm-load PRT

$$\Delta T_{A,W,i,j} = T_{A,W,i,j} - \overline{T_{W,j}}, \quad \frac{n_{cw}}{2} + 1 \leq i \leq n_{cw}, \quad (22)$$

where $\Delta T_{A,W}$ is the warm-load noise.

NEDT is the unbiased second central moment of noise [Yang and Yang, 2021]

$$\text{NEDT}_{\text{total}} = \sqrt{\frac{1}{MN-1} \sum_{j=1}^N \sum_{i=1}^M (\Delta T_{A,W,i,j} - \overline{\Delta T_{A,W}})^2}, \quad (23)$$

where $\Delta T_{A,W}$ is a $M \times N$ matrix, and $\overline{\Delta T_{A,W}}$ is

$$\overline{\Delta T_{A,W}} = \frac{1}{MN} \sum_{j=1}^N \sum_{i=1}^M \Delta T_{A,W,i,j}. \quad (24)$$

We can break down total noise into thermal and 1/f noise based on their spectra information, since thermal noise dominates on a small time scale while 1/f noise is significant on a large time scale [Lee, 2004]. Thermal noise is calculated as [Yang et al., 2021]

$$\text{NEDT}_{\text{thermal}} = \sqrt{\frac{1}{2N(M-1)} \sum_{j=1}^N \sum_{i=1}^{M-1} (\Delta T_{A,W,i+1,j} - \Delta T_{A,W,i,j})^2}. \quad (25)$$

The above equation calculates the unbiased standard deviation of adjacent cross-track samples of a time scale of a couple of milli-second. The above equation is based on Allan deviation, which is an unbiased estimate of thermal noise that is Gaussian additive white noise [Yang et al., 2021].

1/f noise is given by

$$\text{NEDT}_{1/f} = \sqrt{(\text{NEDT}_{\text{total}}^2 - \text{NEDT}_{\text{thermal}}^2)}. \quad (26)$$

In the presence of noise other than thermal and 1/f noise, this equation gives the magnitude of non-thermal noise.

The 1/f percentage is defined as [Yang et al., 2021]

$$P_{1/f} = \frac{\text{NEDT}_{1/f}^2}{\text{NEDT}_{\text{total}}^2} \times 100\%. \quad (27)$$

$P_{1/f}$ ranges from 0 to 100%. In [Yang et al., 2021], the partitioning of thermal and 1/f noise has been validated with simulation and observation. While we have calculated noise of warm-load, the similar procedure can be applied to derive the cold-space noise as well. The derivation can be applied to calculate scene noise in TVAC.

4.2.5 Examining Noise with PCA

The brightness temperature or noise can be written as

$$A = \begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{bmatrix} \quad (28)$$

where the rows are for cross-track with the columns for the along-track. We remove the columnar mean of A so that A has zero mean for each column. Its covariance matrix is

$$C = \frac{1}{(n-1)} A^T A \quad (29)$$

PCA is the eigen-decomposition of the covariance matrix

$$C = E \Lambda E^T \quad (30)$$

where E is the eigenvector matrix with columns as eigenvectors, and Λ is a diagonal matrix of variance.

The covariance matrix size is the square of the along-track number n and becomes computationally difficult to solve with a large n . In practice, we solve PCA with the single value decomposition (SVD). SVD does not need to construct the covariance matrix. Also, SVD can solve for the first few eigenvectors rather than all of them, leveraging computational efficiency. A can be factorized by SVD as

$$A = U S V^T \quad (31)$$

where U is an $m \times n$ orthogonal matrix, V is an $n \times n$ orthogonal matrix, and S is an $m \times n$ rectangular diagonal matrix with nonnegative diagonal entries. The diagonal entries of S are singular values of A . We can define S^2 as

$$S^2 = S^T S = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \sigma_n^2 \end{bmatrix} \quad (32)$$

where S^2 can be sorted with the diagonal entries from high to low such that $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$.

SVD is related to eigen-decomposition as

$$\begin{aligned} A^T A &= (U S V^T)^T (U S V^T) \\ &= V S^T U^T U S V^T \\ &= V S^T I S V^T \\ &= V S^2 V^T \end{aligned} \quad (33)$$

Combining Equations 25-29, there are $\Lambda = \frac{1}{(n-1)} S^2$ and $E = V$. Λ can be written as

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} \quad (34)$$

where $\lambda_i = \frac{1}{(n-1)}\sigma_i^2$.

The projection of A onto the eigenvector basis is

$$P = AV \quad (35)$$

where P is an $n \times n$ matrix with the columns of P as the principal components (PCs). For the i th PC, the corresponding variance is λ_i .

The explained variance can be written in ratio by normalization

$$R_{var}(i) = \frac{\lambda_i}{\sum_i^m \lambda_i} \quad (36)$$

The cumulative explained variance ratio is the discrete integration of $R_{var}(i)$.

4.2.6 Striping

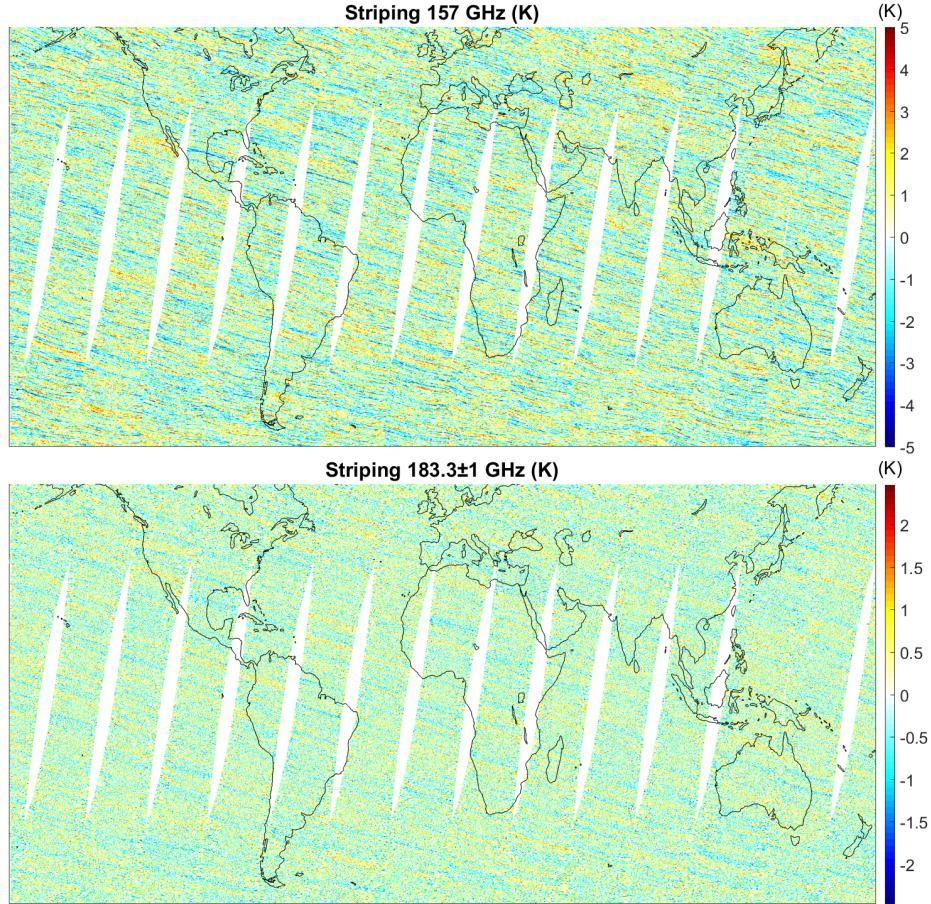


Figure 13: Simulated striping for MetOp-C MHS channel 157 GHz on July 1st, 2019. Empirical parameters include magnitudes of thermal, 1/f noise, and orbital oscillation based on observation.

Striping features sharp, non-periodic stripe patterns with large fluctuation in both radiance data and high-level science products [Jarnot *et al.*, 1996; Kim *et al.*, 2014; Qin *et al.*, 2013; Li and Liu, 2016; Yang *et al.*, 2021]. Striping appears non-stationary, which can severely degrade radiometer

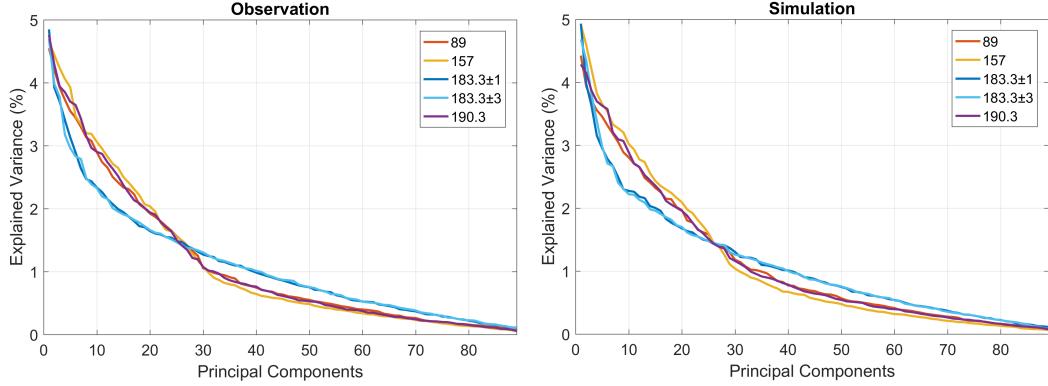


Figure 14: Comparison of explained variance of observation and simulation of MetOp-A MHS.

performance and data applications. Striping simulation is described in the calibration and noise section.

An example of SatERR-simulated striping is shown in Figure 13 below. It is for MHS, with the striping magnitudes derived from the in-orbit measurement [Yang *et al.*, 2021]. In the simulation, 1/f and thermal noise are added as well as the orbital oscillation. Figure 14 compares the PCA-derived explained variance between observation and simulation. Figure 15 shows the PSD of observation and simulation. The observation and simulation are in good agreement.

4.2.7 Cross-pol Contamination

The antenna main polarization (co-pol) sees some radiation leakage from the orthogonal polarization plane (cross-pol). The cross-pol leakage can be due to the antenna asymmetry, defects, excitation of higher-order modes [Ruf, 1998; Davis and Agarwal, 2011; Balanis, 2016]. Cross-pol correction is a necessary step in APC. In the simulator, the cross-pol is a 4×4 matrix as

$$\begin{bmatrix} T'_v \\ T'_h \\ T'_3 \\ T'_4 \end{bmatrix} = \begin{bmatrix} C_{vv} & C_{vh} & C_{v3} & C_{v4} \\ C_{hv} & C_{hh} & C_{h3} & C_{h4} \\ C_{3v} & C_{3h} & C_{33} & C_{34} \\ C_{4v} & C_{4h} & C_{43} & C_{44} \end{bmatrix} \begin{bmatrix} T_v \\ T_h \\ T_3 \\ T_4 \end{bmatrix} \quad (37)$$

where the $C_{i,j}$ is cross-pol coefficients that transforms the cross-pol free Stokes (right) to the new Stokes (left). The values of the coefficients can be customized. In the case of cross-pol free, the matrix is the identity matrix. Examples of cross-pol sensitivity are presented in the results.

4.2.8 Twist angle offset

The twist angles refer to a set of reflector and polarization angles that affect radiance and polarization. For a cross-track scanning radiometer like AMSU-A and MHS the channel T_b is [Weng *et al.*, 2003]:

$$\begin{aligned} T_{b,qv} &= A^2 T_{b,v} + B^2 T_{b,h} \\ T_{b,qh} &= A^2 T_{b,h} + B^2 T_{b,v} \end{aligned} \quad (38)$$

where $T_{b,qv}$ and $T_{b,qh}$ are the quasi-V and quasi-H brightness temperature respectively, and $T_{b,v}$ and $T_{b,h}$ are T_b of the vertical polarization (v-pol) and horizontal (h-pol) respectively. Without any angular offset, there is $A = \cos \phi$ and $B = \sin \phi$ with ϕ as the scan angle.

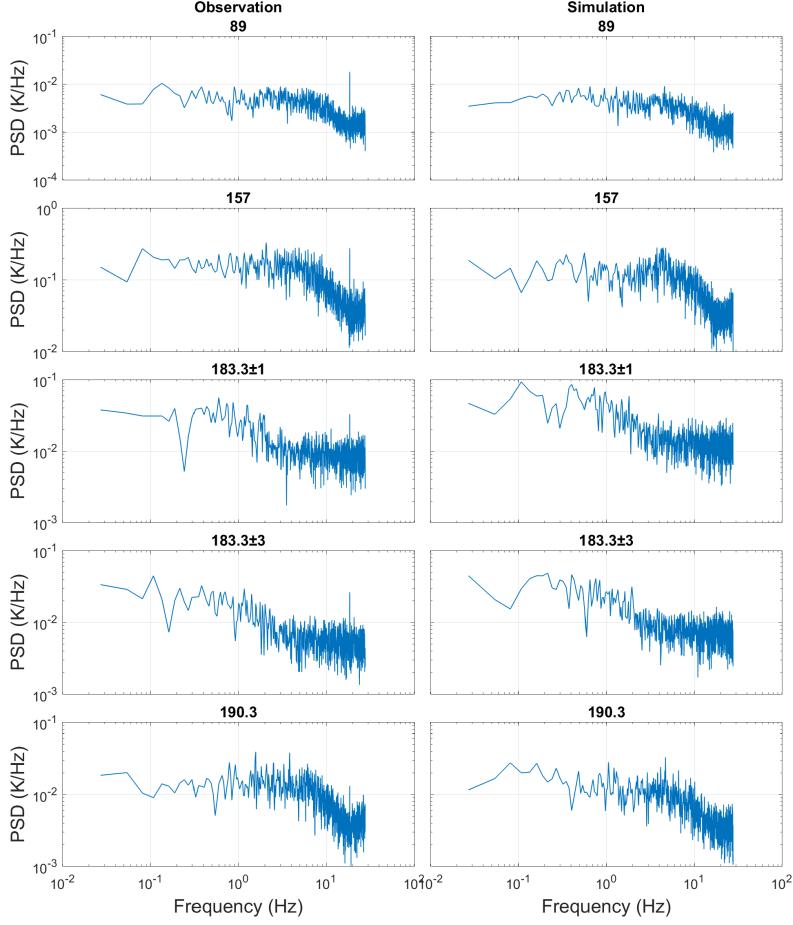


Figure 15: Comparison of power spectral density of observation and simulation of MetOp-A MHS.

The equation can be rewritten as a function of twist angles as

$$A = \frac{1}{\sqrt{1 - \sin^2 2\theta \cos^2 \phi}} \cdot [\cos 2\theta \sin \psi - 2 \sin \theta \cos(\phi + \psi) \sin \phi] \quad (39)$$

$$B = \frac{1}{\sqrt{1 - \sin^2 2\theta \cos^2 \phi}} \cdot \{ (\cos^2 2\theta + \sin^2 2\theta \sin^2 \phi) \cdot [2 \sin^2 \theta \cos(\phi + \psi) \cos \phi - \cos \psi] - \frac{1}{2} \sin^2 2\theta \sin 2\phi [2 \sin^2 \theta \cos(\phi + \psi) \sin \phi + \sin \psi] - \frac{1}{2} \sin 4\theta \cos \phi \sin 2\theta \cos(\phi + \psi) \} \quad (40)$$

where twist angles include θ of the reflector angle, ψ of the polarization alignment angle, and ϕ of the scan angle. The nominal reflector and polarization alignment angles without offset are $\theta = 45^\circ$ and $\psi = 90^\circ$, and the above two equations reduce to $A = \cos \phi$ and $B = \sin \phi$. The twist angles can be tuned to examine their impact on T_b . An example is shown in Figure 17.

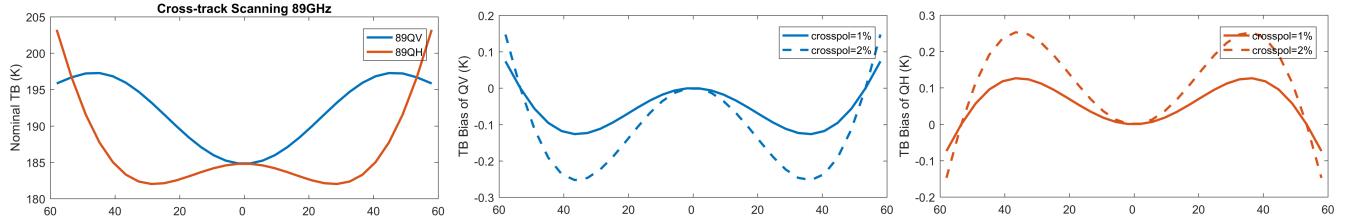


Figure 16: Effect of cross-pol leakage for cross-track scanning radiometers at 89 GHz of QV and QH. The panels show the (a) nominal TBs, (b) bias at QV, and (c) bias at QH. The RTM simulation assumes mid-latitude winter atmospheric profiles with a SST of 290 K and a wind speed of 5 m/s.

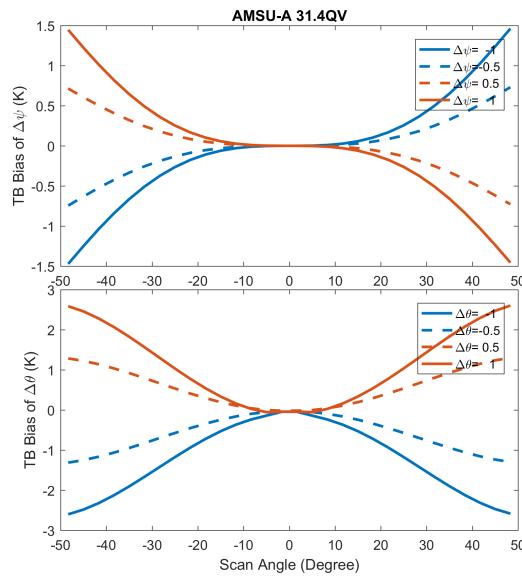


Figure 17: Sensitivity of twist angle offset for AMSU-A channel-2 of 31.4 GHz QV. a) The sensitivity of polarization alignment angle ψ and b) the sensitivity of reflector angle θ .

Unlike cross-track scanning, a conical scanning radiometer has constant scan angles. We only consider the offset in the polarization alignment angle ψ .

4.2.9 Spacecraft attitude and geolocation

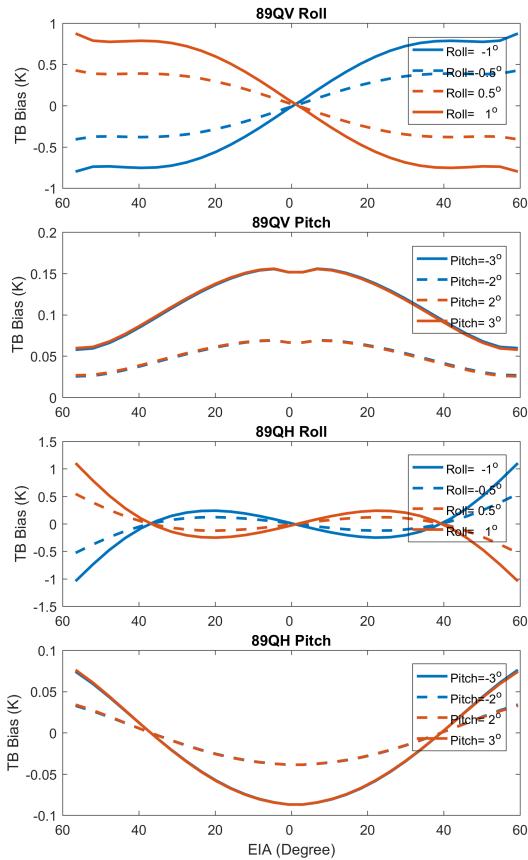


Figure 18: Attitude biases at 89 GHz QV and QH of cross-track scanning radiometers for (upper two panels) roll and (lower two panels) pitch.

A spacecraft and onboard sensor can have attitude errors in terms of roll, pitch, and yaw. The attitude changes the FOV geolocation and incidence angle and therefore the scene temperature [Meissner and Wentz, 2006; Kroodsma *et al.*, 2012]. Attitude is also linked to other error sources such as Faraday rotation which is dependent on FOV geolocation and angles.

A module is developed for performing coordinate systems and transformation [Cai *et al.*, 2011]. The steps for coordinate transformation are briefed as follows. The error-free cross-track or conical scanning vector is computed in the local spacecraft frame of North-East-Down (NED) coordinate. The rotation matrix for the attitude is calculated and applied to the scan vector. The new scan NED vector is transformed to the Earth-centered Earth-fixed (ECEF) geocentric coordinate. The line-of-sight intersection on the ground is computed, and the FOV geolocation is determined. The geodetic coordinate, including FOV geolocation and azimuth and incidence angle, is obtained by transforming the FOV vector from geocentric to geodetic coordinate. The Earth ellipsoid model of WGS-84 is used, while a customized ellipsoid can also be used. The FOV geolocation and angles on the ionosphere shell are computed in the same way but with new ellipsoid parameters counting ionosphere altitude.

The afore-said transforms are from the radiometer to the Earth. It can also be inverted: the input can be realistic FOV geolocations and spacecraft orbit, and the scan NED vector can be determined. Then the attitude can be applied. By doing so, attitude is applied to any operating radiometers. In addition to observational data, the simulator also includes a Keplerian orbit model that simulates a spacecraft orbit [Griffin, 2004]. The orbit can be designed with the six orbit elements. This allows studying attitude in a more idealized but versatile way since the orbit and scan are flexible. Figure 18 presents the sensitivity of attitude offset over the ocean.

4.2.10 Antenna pattern

Antenna pattern determines the weighted radiance a radiometer receives in all solid angles [Balanis, 2015]. APC is necessary in calibration to process weighted T_A to the scene T_b . Antenna pattern can be measured on the ground, and a hybrid one with both measurement and modeling is often used. Errors in the antenna pattern result in T_b bias when there is an overestimate or underestimate of sidelobe and spillover. And the bias is not a constant but dependent on the scene temperature since APC has a scaling effect.

A module is included in the simulator to study the antenna pattern impact. Antenna pattern can be parameterized for the mainlobe, sidelobe, and spillover. The fractions of them can be customized with respect to specific radiometers. Different FOV targets can be set for the mainlobe, sidelobe and spillover. For instance, the nominal operating can be set to have mainlobe and sidelobe viewing the ocean with the cosmic background for spillover. To simulate the deep-space maneuver when the radiometer is flipped upside down, the mainlobe and sidelobe can be set viewing the cosmic with the spillover toward the Earth.

4.2.11 Faraday rotation

Faraday rotation takes place when a polarized EM wave goes through the ionosphere with the geomagnetic field [Meissner and Wentz, 2006; Peng et al., 2017] (Figure 19). The E-field vector of the EM wave rotates and the Stokes vector is altered. Faraday rotation is dependent on the EM frequency, ionosphere electron content, and the geomagnetic field. It is pronounced at low-frequencies like L-band and for polarimetric radiometers like WindSat where the second and third Stokes are measured. It can be written by [Meissner and Wentz, 2006; Peng et al., 2017]

$$\Omega = \frac{2.365 \cdot 10^2}{f^2} \cdot B \cos \gamma \cdot \text{VTEC} \sec \theta \quad (41)$$

where Ω is in radian, f is the frequency in GHz, B is the geomagnetic field in Tesla, and γ in radian is the angle between the EM wave and B field. VTEC is the integral of vertical total electron content in TEC unit (TECU), where 1 TECU is 10^{16} electron/m². And θ in radian is the angle between the EM wave and ionosphere.

Faraday rotation can be written in the modified Stokes

$$\begin{bmatrix} T'_v \\ T'_h \\ T'_3 \\ T'_4 \end{bmatrix} = \begin{bmatrix} \cos^2 \Omega & \sin^2 \Omega & 1/2 \sin 2\Omega & 0 \\ \sin^2 \Omega & \cos^2 \Omega & -1/2 \sin 2\Omega & 0 \\ -\sin 2\Omega & \sin 2\Omega & \cos 2\Omega & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} T_v \\ T_h \\ T_3 \\ T_4 \end{bmatrix} \quad (42)$$

where the right side Stokes is before entering ionosphere and the left side one is with Faraday rotation effect.

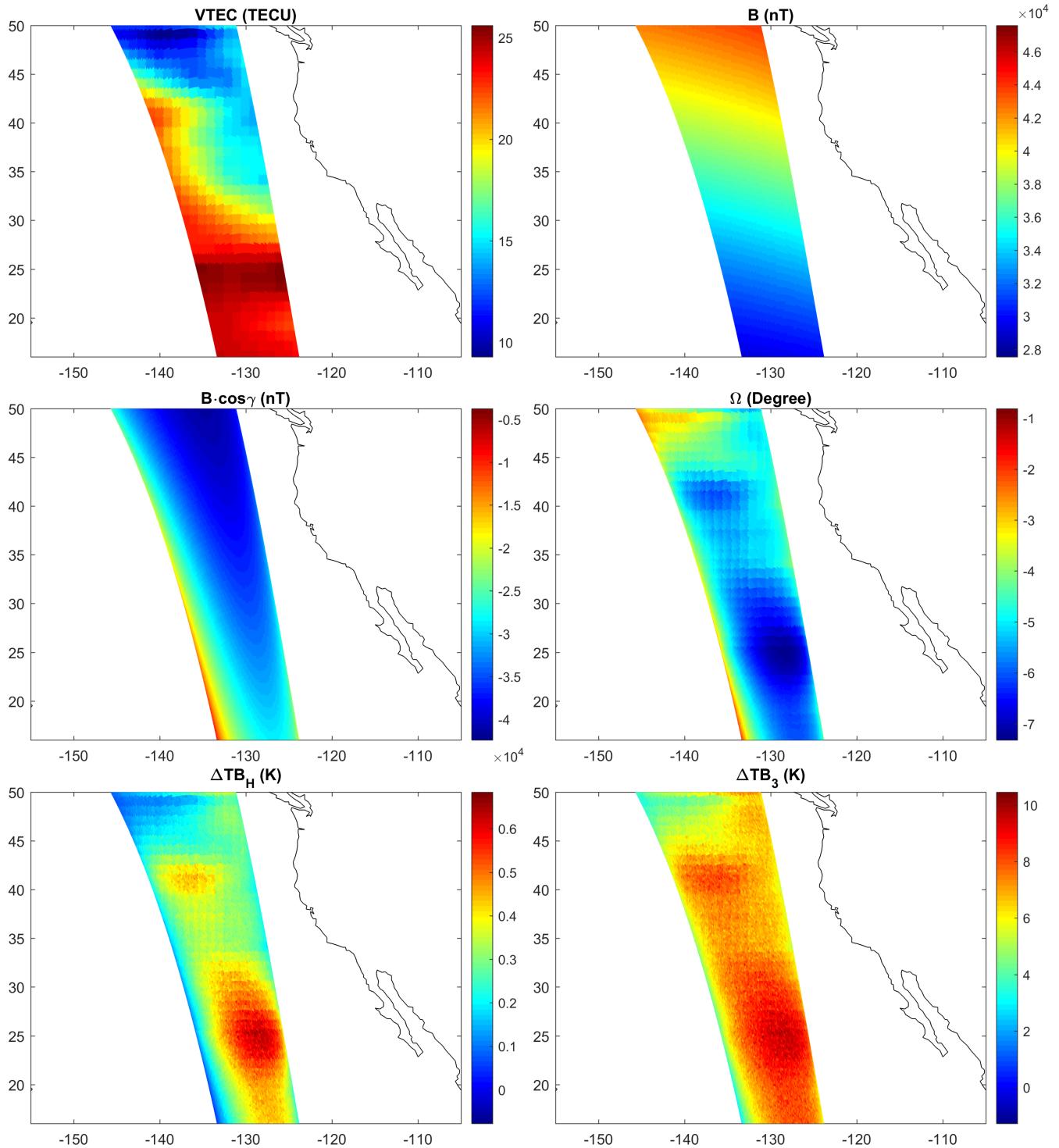


Figure 19: Simulated parameters of Faraday rotation for SMAP.

The simulator treats the electron content as a shell from the ground [Meissner and Wentz, 2006; Peng *et al.*, 2017]. The altitude of the shell is adjustable and the default setting is 400 km. The ionosphere altitude can also be varied to account for the fact it has a lower altitude toward the poles. The geolocation of EM waves on the ionosphere is calculated with the Earth mapping module provided the spacecraft orbit and scan geometry. The IGRF model is included as a module for computing the geomagnetic field [Thébault *et al.*, 2015]. The total electron content can be obtained from observations or models and imported to the simulator [Bilitza *et al.*, 2017].

4.2.12 Reflector emission

An ideal reflector has no self-emission with emissivity equal to zero. In practice, its emissivity can be slightly larger than zero with self-emission seen by the receiver [Wentz *et al.*, 2001; Gopalan *et al.*, 2009]. The reflector emission introduces errors as it is mixed with radiance of Earth scene, cold and warm references. The emission can be negligible if the emissivity is small enough. An assessment of the reflector emissivity is always necessary.

A module in the simulator is for the reflector emission. It includes the setting for reflector emissivity and physical temperature. The reflector emissivity is dependent on frequency and polarization. For cross-track scanning, the emissivity is dependent on scan angle as it changes in different scan positions. It is insensitive to scan angle for a conical scanning radiometer as the scan angle is very constant. The reflector temperature is dependent on scan and orbit position.

4.2.13 Cold-space mirror emission

For conical scanning radiometers, a cold-space mirror is often employed for reflecting the cosmic background to the feedhorn. Its self-emission can lead to error in the cold reference. But the warmload and Earth scenes are unaffected, which is different from the reflector emission.

$$T_C = T_{\text{mirror}}\epsilon_{\text{mirror}} + T_C(1 - \epsilon_{\text{mirror}}) \quad (43)$$

where T_{mirror} is the mirror temperature, T_C is cosmic background, ϵ_{mirror} is mirror emissivity.

4.2.14 Warm load error

Warm load measurements can have errors. The onboard warm load is a blackbody. It is expected that it is a perfect blackbody with emissivity of one. This is opposite to the expectation of the reflector. The emissivity is less than one in practice [Twarog *et al.*, 2006; ATB, 2016; Alquaied *et al.*, 2018]. A correction is needed if the error is large enough. Additionally, the warm load error can come from the temperature gradient, which makes the PRT measured temperature deviate from the actual temperature. The situation can be pronounced when there is a solar intrusion on the warm load target, which induces a large temperature gradient [Twarog *et al.*, 2006; ATB, 2016; Alquaied *et al.*, 2018].

We provide a module for parameterizing the warm load error. The warm load effective temperature is a function of physical temperature, emissivity, and a bias term. The emissivity multiplied by the physical temperature is the apparent temperature. The emissivity ranges from zero to one. The bias term is a constant, and negative bias means the warm load temperature would be underestimated and positive bias means an overestimate of warm load temperature. The settings are channel dependent.

4.2.15 FOV intrusion

FOV intrusion, or beam spoiling, occurs when the antenna FOV is obstructed by objects like the spacecraft or payloads. The effect has been found such as for SSM/I, TMI, AMSU-A [*Colton and Poe, 1999; Wentz et al., 2001; McKague et al., 2010*]. It is usually obvious at the edge of the scan. The intrusion errors can be parameterized by the temperature of the intrusion object and fraction of intrusion FOV [*McKague et al., 2010*]

$$T_b = T_{b,intr}\Omega_{intr} + T_{b,scene}(1 - \Omega_{intr}) \quad (44)$$

where $T_{b,intr}$ is the brightness temperature of the intrusion object, $T_{b,scene}$ is for the Earth scene, Ω_{intr} is the intrusion solid angle fraction. Ω_{intr} is a function of scan position. Ω_{intr} can be the error function assuming antenna pattern is a Gaussian function. And it can be further simplified as an exponential function when the FOV intrusion is small. $T_{b,intr}$ can be set as a function of both cross-track and along-track scan.

4.2.16 Bandpass spectral response

The spectral response is the bandpass responsivity of the finite bandwidth around the center frequency [*Kim et al., 2014; ATB, 2016*]. The spectral response includes the response of amplifiers and BPFs and is dependent on the hardware. It usually exhibits channel-dependent responsivity that appears irregular than a flat spectrum. It is often measured in prelaunch tests. The effective channel radiance is the incoming radiance weighted by the spectral response.

The spectral response can be counted in the simulator. The channel bandpass can be split to plenty of sub-frequencies. A channel can have double sidebands or more subbands. RTM simulation is performed at every sub-frequency, and the channel effective brightness temperature is the convolution of T_b with the spectral response.

$$T_{b,chan} = \sum_{i=1}^n \int_{f_i-B_i/2}^{f_i+B_i/2} T_b(f) S_i(f) df \quad (45)$$

where $T_{b,chan}$ is the channel effective brightness temperature, f is the sub-frequency, i is number of subband, f_i is the subband center frequency, B is the subband bandwidth, $S_i(f)$ is the subband spectral response.

Doing so enables more accurate RTM simulations [*Yang and Yang, 2018*], since the atmospheric absorption is usually nonlinear in the finite bandpass. When the spectral response or atmospheric absorption nonlinearity is pronounced, a noticeable difference can be found between accurate simulation across the bandwidth and at only the center frequency of a channel.

4.2.17 Interference of instrument and geomagnetic field

Interference can take place due to the onboard instrument. For example, in the early flight phase of GMI, instrumental induced magnetic biases are found in all channels. The 10.65 GHz channel has a pronounced bias of 1.2 K of peak-to-peak. The bias is scan-dependent but is time invariant. The cause was later identified as instrumental interference of the reflector spin and feed switches on the launch restraints [*Draper et al., 2015; Wentz and Draper, 2016*]. The variation of the geomagnetic field was also found to affect the radiometer. In our simulator, we consider the instrumental bias as an additive signal that is channel dependent. The additive bias can be customized and is added to T_A .

4.2.18 Crosstalk

The inter-channel correlation can stem from crosstalk of hardware [Ruf, 1998; Corbella *et al.*, 2002; Kim *et al.*, 2014].

Crosstalk can be modeled in different ways. One simple way is

$$T_A = M \cdot T_A \gamma + T_A(1 - \gamma) \quad (46)$$

where M is the crosstalk matrix, γ is the reflection coefficient. The reflection can take place in the antenna sub-system.

4.2.19 Antenna temperature to brightness temperature

The conversion of T_A to T_B is often referred to as the antenna pattern correction (APC). There are a number of steps in APC such as the correction for spillover, cross-pol contamination. Additional corrections are also needed in case of reflector emission, warmload error, FOV intrusion, etc. These corrections are included in the simulator, and can be turned on or off to study the sensitivity. Correction modules include:

- Removing instrument inference
- Edge-of-scan intrusion correction
- Spillover correction
- Main reflector emission correction
- Cold-space mirror emission correction
- Cross-pol contamination correction
- Crosstalk correction
- Faraday rotation correction

4.3 Observation operator error

4.3.1 Ancillary models and modules

A 1-D RTM is included in the simulator [Yang *et al.*, 2016; Yang and Yang, 2018]. The atmospheric gaseous absorption is based on the Rosenkranz model [Rosenkranz, 2017; Rosenkranz and Cimini, 2019]. An oceanic emission model is included [Elsaesser and Kummerow, 2008]. The users can also choose other surface and atmospheric absorption model and import the computed variables to the simulator.

The IGRF model is included for computing the geomagnetic field. It is the 12th generation of the model that covers from 1990 to the end of 2019 [Thébault *et al.*, 2015]. It is a series of models describing the geomagnetic field and is published by the International Association of Geomagnetism and Aeronomy (IAGA). The geomagnetic field is derived as the gradient of the magnetic scalar potential. The input for IGRF is geodetic coordinates, and the output is the vector of the geomagnetic field. Our simulator can also import and use geomagnetic fields produced by other models.

A Keplerian orbit model is used for simulating the spacecraft orbit [Griffin, 2004]. The input is the six orbit elements, and the output is the spacecraft geolocation. Orbital perturbations are not

considered in the model, but it is sufficient to study effects like spacecraft attitude. Observational orbits can be imported to the simulator.

4.4 Representativeness error

Representativeness is due to the unresolved scales and processes, where the Numerical Weather Predicting (NWP) model resolution is usually coarser than satellite observations and has empirical parameterizations.

We develop a module for collocating satellite FOVs and the reanalyses, with the horizontal and vertical resolution adjustable. Accordingly, the RTM simulation can be conducted with a spatial resolution that the users define through the collocation. The resolution can be flexibly coarsened. A resolution finer than the reanalyses is realized by linear interpolation.

4.5 Pre-processing Error

The pre-processing is for screening data, such as rejecting observations that cannot be adequately modeled. Pre-processing can be involved with retrieval for converting radiance to state variables. Due to the limitation of filters and retrieval, pre-processing can give rise to error and uncertainties. We include a number of filters based on measured radiance for screening cloud, precipitation.

The retrieval module includes several retrieval algorithms for processing radiance data into science products. It is for evaluating the impact of radiance errors on science products. For instance, empirical algorithms are included for retrieving atmospheric temperature, cloud liquid water content, oceanic surface wind speed [Goodberlet *et al.*, 1989; Zhu and Weng, 2013; Rosenkranz, 2006]. The retrieval can also be used for studying pre-screening in intercalibration.

Users can also refer to many established algorithms for producing EDR and CDR products, such as those at NOAA and EUMETSAT. For example, the NOAA's Microwave Integrated Retrieval System (MIRS) is a comprehensive package based on 1D-variational (1DVar) inversion. MIRS can be used for studying error propagation in the retrieval.

4.6 Miscellaneous

4.6.1 Collocation

A set of scripts are provided for collocating satellites, satellite-reanalysis. The criteria of collocation can be adjusted for distance, time. The distance can be set in a coordinate of degree or kilometer, the latter is of a conversion from degree to kilometer. The code is vectorized with a speed close to a Fortran code. An example script is:

```
1 function [cID,clat,clon,ctime,IDX1,IDX2] = sub_collocate(Time,Grid,lat1,
2   lon1,time1,lat2,lon2,time2,varargin)
3 % spatialtemporal collocation
4 % (output is sorted by default)
5 %
6 % Input (1D [n,1], double precision):
7 %     Time,      time resolution (minute)
8 %     Grid,      grid resolution (km/degree)
9 %     lat1,      x1 location
10 %    lon1,      y1 location
11 %    lat2,      x2 location
```

```

11 %           lon2,      y2 location
12 %           varargin, sorted/stable (default is sorted); specify range of x&y
13 %
14 % Output:
15 %           cID,       ID in common
16 %           ID1,       ID for group 1
17 %           idx1,     logical for ismember
18 %           ID2,       ID for group 2
19 %           idx2,     logical for ismember
20 %
21 %
22 % History:
23 % written by John Xun Yang, University of Maryland, jxyang@umd.edu, or
24 % johnxun@umich.edu, 11/22/2016
25 % revised by John Xun Yang, University of Maryland, jxyang@umd.edu, or
26 % johnxun@umich.edu, 12/26/2016: varargin for range
27 % revised by John Xun Yang, University of Maryland, jxyang@umd.edu, or
28 % johnxun@umich.edu, 01/18/2016: stable option
29
30 % setting
31 w = 24*60/Time; % windows for matching time; e.g., 24*60/5=within 5 minutes
32
33 % preprocess
34 time1org = time1; % coordinate 1 as output
35 lat1org = lat1;
36 lon1org = lon1;
37
38 if ~isempty(varargin) % use given range
39     Sort = varargin{1};
40     RangeLat = varargin{2};
41     RangeLon = varargin{3};
42 else
43     Sort = 'sorted';
44     RangeLat = max(max(lat1),max(lat2)) - min(min(lat1),min(lat2));
45     RangeLon = max(max(lon1),max(lon2)) - min(min(lon1),min(lon2));
46 end
47
48 IDgeoNum = round(RangeLat/Grid+1)*round(RangeLon/Grid+1);
49 LatNum = round(RangeLat/Grid+1);
50
51 temp = floor(min(min(time1),min(time2)));
52 time1 = time1-temp;
53 time2 = time2-temp;

```

4.6.2 Data download

We provide scripts for data download such as ECMWF reanalyses. An example script is:

```

1 #!/usr/bin/env python
2
3 """
4 Dowloading ERA5 data from ECMWF
5     sources are CDS or MARS
6     CDS has 37-pressure-level or single-level
7     MARS has 137-model level (MARS's surface is identical to CDS single-level)
8     CDS downloading is fast, while MARS is slow
9

```

```

10 Examples:
11     date_begin = '20140101' # beginning day
12     date_end = '20140201' # ending day
13     var_names = ['tp','skt','sst']
14     data_source = 'CDS'
15     data_level = 'sl'
16     running the code: python main_era5.py
17
18 @author: John Xun Yang, jxyang@umd.edu, or johnxun@umich.edu, 04/17/2020,
19                               original code
20             John Xun Yang, jxyang@umd.edu, or johnxun@umich.edu, 05/30/2021,
21                               flexible input
22             John Xun Yang, jxyang@umd.edu, or johnxun@umich.edu, 10/10/2021, CDS &
23                               MARS
24 """
25
26
27 from mod_era5_dn import era5_ec_dn
28
29 # =====
30 # setting
31 # =====
32
33 # date range and variables
34 # date_begin<= date <date_end
35 date_begin = '20140101' # beginning day
36 date_end = '20140201' # ending day
37 var_names = ['tp'] # short name per ECMWF convention; refer to era5_var_list['
38                               short_name']; https://apps.ecmwf.int/
39                               codes/grib/param-db
40
41
42 # =====
43 # CDS/MARS
44 #   CDS: pl(37-level) and sl; fast
45 #   MARS: ml(137-level); slow
46 #   ml/pl/sl = model-level/pressure-level/single-level
47 data_source = 'CDS'
48 data_level = 'sl'
49
50
51 # =====
52 # downloading
53 # =====
54 era5_ec_dn(date_begin,date_end,var_names,data_source,data_level)

```

Given the date range and variables, the data can be download during this period.

The ECMWF reanalyses can have 37-level reanalysis (ERA5), 137-level at model level. The prerequisite is the installation of the ECMWF CDS code. Refer to <https://cds.climate.copernicus.eu/api-how-to>.

The NOAA GDAS data is available from UCAR: <https://rda.ucar.edu/datasets/ds083.2/>.

A Acronyms and Abbreviations

Acronyms	Description
ACF	Auto Correlation Function
ADC	Analog-To-Digit Converter
AGWN	Additive Gaussian White Noise
AMSU-A/B	Advanced Microwave Sounding Unit
APC	Antenna Pattern Correction
ATMS	Advanced Technology Microwave Sounder
BPF	Band Pass Filter
CDR	Climate Data Record
CLWP	Cloud Liquid Water Path
Cross-pol	Cross Polarization
CRTM	Community Radiative Transfer Model
EDR	Environmental Data Record
EM	Electromagnetic
ERA5	ECMWF Reanalysis V5
SatERR	Error Representation and Realization of Satellite Observation
FM	Frequency Modulation
FOV	Field Of View
GDAS	Global Data Assimilation System
IGRF	International Geomagnetic Reference Field
LNA	Low Noise Amplifier
LO	Local Oscillator
MHS	Microwave Humidity Sounder
MIRS	Microwave Integrated Retrieval System
NEDT	Noise Equivalent Delta Temperature
NPP	National Polar-orbiting Partnership
O-A	Observation Minus Analysis
O-B	Observation Minus Background
OMT	Orthomode Transducer
PC	Principal Component
PCA	Principal Component Analysis
PM	Phase Modulation
PRT	Platinum Resistance Thermometers
PSD	Power Spectral Density
QH	Quasi Horizontal
QV	Quasi Vertical
RFIC	Radio Frequency Integrated Circuit
RNG	Random Nonise Generator
RTTOV	Radiative Transfer For TOVS
SVD	Singular Value Decomposition
T_A	Antenna Temperature
T_B	Brightness Temperature
TEC	Total Electron Content
TIROS	Television Infrared Observation Satellite
TOA	Top Of Atmosphere

TOVS	TIROS Operational Vertical Sounder
TVAC	Thermal Vacuum Chamber
VTEC	Vertical Total Electron Content

B Simulating Current and Future Sensors

Conical Scanning	Crosstrack Scanning
AMSR2 GMI SMAP SSMI SSMIS customized	AMSU-A AMSU-B ATMS MHS TEMPEST-D TROPICS customized

C List of Error Sources

No.	Error Sources	Description
1	RFIC stationary noise	thermal noise, quantization noise in ADC, etc
2	RFIC non-stationary noise	1/f noise, shot noise, etc
3	local-oscillator noise	LO noise such as random walk FM, flicker FM, white FM, flicker PM
4	warm-load noise	noise in the warm load
5	gain jump	abrupt changes in the power gain
6	receiver nonlinearity	radiometric nonlinearity of the receiver
7	striping	radiometric stripes related to 1/f noise, thermal noise, and calibration
8	spectral response	bandpass spectral characteristics that are channel dependent
9	scanning and pointing	scanning angle error for viewing cold/warm reference, Earth scene
10	orbital oscillation	oscillation of warm load, receiver temperature, gain
11	twist angle offset	offsets in polarization alignment angle, reflector angle, scan angle
12	cross polarization	cross polarization leakage
13	cross talk	channel cross talk
14	reflector emission	main reflector emission
15	cold-space mirror emission	cold-space mirror emission
16	warm load emissivity	non-ideal blackbody emissivity, bias
17	spillover	spillover seeing cosmic background
18	FOV intrusion	field of view intrusion due to the spacecraft or payloads
19	payload interference	interference from onboard instruments
20	spacecraft attitude offset	roll, pitch, yaw
21	Faraday rotation	Faraday rotation due to ionosphere and Earth magnetic field
22	count to antenna temperature	calibration frequency, windowing function, etc
23	antenna pattern correction	calibrating antenna temperature to brightness temperature
24	gain and noise long-term change	long-term degradation in terms of noise and gain
25	frequency drift	channel center frequency drift
26	orbit drift	long-term satellite altitude change
27	horizontal spatial correlation	FOV overlap horizontally

28	vertical spatial correlation	vertical overlap of channel weighting functions
29	surface and atmospheric profiles uncertainties	error in surface and atmospheric profiles
30	RTM error	RTM deficiencies in surface, atmospheric models
31	spatial resolution in RTM	vertical or horizontal resolutions affecting RTM simulation
32	collocation	collocation criteria for single or multiple satellite
33	intercalibration	errors due to sensor difference in frequency, scanning angle, collocation criteria, RTM
34	pre-screening	filtering and screening for satellite data
35	retrieval	inversion based retrieval affecting EDR and CDR
36	external models, algorithms	combining our simulator with external RTM, retrieval algorithms

D List of Scripts

Script Name	Script Name
imp:	polanglecoeff_cross.m
saterr_imp_1fadjust.m	polmix_cross.m
saterr_imp_1fper.m	stokes_reflconical.m
saterr_imp_attitudeoffset.m	stokes_reflct.m
saterr_imp_band2chan_inout.m	
saterr_imp_band2chan.m	prof:
saterr_imp_band2chan_sat_ana.m	col_prof2grid.m
saterr_imp_conic_mirrorcold.m	col_prof2grid_twice.m
saterr_imp_count.m	col_uniquegrid.m
saterr_imp_crosspol.m	ECMWF_pres_sfc2atm_L137.m
saterr_imp_crosstalk.m	ECMWF_pres_sfc2atm_L91.m
saterr_imp_faradayangle.m	ec_read_era5_ana_ml_137_sfc_umd.m
saterr_imp_faradystokes.m	filter_landmask.m
saterr_imp_farfieldTB_fun.m	ind_filter1D.m
saterr_imp_farfieldTB.m	ind_startend_bin.m
saterr_imp_file_more_count.m	ind_startend_cum.m
saterr_imp_file_simorbit.m	interp_lin_2D_asc.m
saterr_imp_fovaer.m	interp_lin_3D_asc.m
saterr_imp_fovintrusion.m	movingavg_2Dwin_1conv.m
saterr_imp_initcw.m	parse_sensorname.m
saterr_imp_interference.m	prof_atmsfc.m
saterr_imp_loadgranel.m	prof_interp_atm.m
saterr_imp_noiseadditive.m	prof_read_4sim_bin.m
saterr_imp_noisePRT.m	prof_read_4sim_mat.m
saterr_imp_noiserng.m	prof_write_4sim_bin.m
saterr_imp_nonlinearta.m	prof_write_4sim_mat.m
saterr_imp_orbit.m	prof_write_4sim_nc.m
saterr_imp_orbitscanning.m	prof_write_dailyinfo.m
saterr_imp_orbitsingle.m	qsub_matlab_one.m
saterr_imp_oscil_gain.m	read_bin_AMSU_A.m
saterr_imp_oscil_PRT.m	read_bin_KLM_MHS_L1B.m

saterr_imp_oscil_tr.m	read_bin_MHS.m
saterr_imp_poisson.m	read_data_cal_tatb_filesdaily.m
saterr_imp_poloffset.m	read_hdf5_atms_daily_geo.m
saterr_imp_prof_files2orbit.m	read_hdf5_atms_daily_sdr_geo.m
saterr_imp_prof_filescal.m	read_hdf5_atms_daily_sdr.m
saterr_imp_prof_files.m	read_hdf5_atms_daily_tdr.m
saterr_imp_prof_filessat.m	read_hdf5_atms_sdr_geo.m
saterr_imp_prof_filessimgran.m	read_sat_filesdaily.m
saterr_imp_prof_filessimorbit.m	read_sim_filesdaily.m
saterr_imp_prof_ini.m	scp_qsub.m
saterr_imp_prof_load.m	set_data_sim_filedaily.m
saterr_imp_prof_read4singleorbit.m	set_FileDaily.m
saterr_imp_receiver.m	set_Read1File.m
saterr_imp_receiverpre_tas.m	ucar_read_era5_ana_pl_37_sfc_umd.m
saterr_imp_receiverpre_tcw.m	write_CRTM_bin_prof.m
saterr_imp_refl_tacw.m	
saterr_imp_refl_tas.m	radcal:
saterr_imp_ref.m	cal2point_win.m
saterr_imp_rtm_atm.m	saterr_cal_2point.m
saterr_imp_save_A.m	saterr_cal_conic_faraday.m
saterr_imp_save_B.m	saterr_cal_conic_fovintrusion.m
saterr_imp_scanning_1fromscanset.m	saterr_cal_conic_mirrorcold.m
saterr_imp_scanning_2fromobsretr.m	saterr_cal_conic_refl_main.m
saterr_imp_scanning_3combo.m	saterr_cal_conic_xpol.m
saterr_imp_scanning_4allfromobs.m	saterr_cal_count2ta.m
saterr_imp_scanning.m	saterr_cal_crosschan.m
saterr_imp_simioout.m	saterr_cal_crosstalk.m
saterr_imp_simioout_more_count.m	saterr_cal_ct_refl_main.m
saterr_imp_simioout_more_toatb.m	saterr_cal_ct_xpol.m
saterr_imp_simiooutpre.m	saterr_cal_interference.m
saterr_imp_simreceiver.m	saterr_cal_nedt.m
saterr_imp_spillover.m	saterr_cal_nonlinear.m
saterr_imp_sr.m	saterr_cal_output.m
saterr_imp_targetTB.m	saterr_cal_output_more.m
saterr_imp_targetTB_prof.m	saterr_cal_output_save.m
saterr_imp_TB_mainsidespillover.m	saterr_cal_output_save_more.m
saterr_imp_TBsource_sidelobe.m	saterr_cal_para_customize.m
saterr_imp_TBsourcesim_load.m	saterr_cal_para_fromsim.m
saterr_imp_TBsourcesim_model.m	saterr_cal_para.m
saterr_imp_TBsource_waveform.m	saterr_cal_para_parse_spillover.m
saterr_imp_TOAtb.m	saterr_cal_read_count.m
saterr_imp_uniqfreq.m	saterr_cal_read_counttatb_less.m
saterr_imp_warmloaderror.m	saterr_cal_read_counttatb_less_setting.m
intercal:	saterr_cal_read_counttatb_more.m
saterr_3intercal_dd.m	saterr_cal_read_counttatb_more_setting.m
saterr_3intercal_dd_plot.m	saterr_cal_spillover.m
	saterr_cal_ta2tb.m

saterr_3intercal_oo.m	saterr_cal_warmloaderror.m
saterr_3intercal_oo_plot.m	saterr_plot_count.m
saterr_3intercal_sd.m	saterr_plot_faraday.m
saterr_3intercal_sd_plot.m	saterr_plot_map_scene.m
saterr_intercal_dd_collocate.m	saterr_plot_nedt.m
saterr_intercal_dd_filter.m	saterr_plot_noiseacf.m
saterr_intercal_dd_output.m	saterr_plot_noisepca.m
saterr_intercal_dd_process.m	saterr_plot_noisepsd.m
saterr_intercal_dd_setting.m	saterr_plot_scandep.m
saterr_intercal_dd_stats.m	saterr_plot_tabias.m
saterr_intercal_oo_collocate.m	saterr_plot_ta.m
saterr_intercal_oo_filter.m	saterr_plot_tanoise.m
saterr_intercal_oo_output.m	
saterr_intercal_oo_process.m	reanalyses:
saterr_intercal_oo_setting.m	ec_main_write_era5_ana_ml_137_sfc_umd.m
saterr_intercal_oo_stats.m	ec_read_era5_ana_ml_137.m
saterr_intercal_plot_sd.m	ec_read_era5_ana_ml_137_sfc_umd.m
saterr_intercal_sd_output.m	ec_read_era5_ana_sfc.m
saterr_intercal_sd_process.m	ec_write_era5_ana_ml_137_sfc_umd.m
saterr_intercal_sd_setting.m	main_era5.py
orbit_period2sat.m	mod_date.py
plot_crosstrack_dd.m	mod_era5_dn.py
plot_crosstrack_oo.m	scaleoffset.m
plot_crosstrack_sd.m	ucar_clw_main_write_era5_ana_pl_37_sfc_umd.m
plot_hist1_dd.m	ucar_clw_read_era5_ana_pl_37_sfc_ucar.m
plot_hist1_oo.m	ucar_clw_read_era5_ana_pl_37_sfc_umd.m
plot_hist1_sd.m	ucar_clw_write_era5_ana_pl_37_sfc_umd.m
plot_hist2_tb_dd.m	
plot_hist2_tb_dif.m	rtm:
plot_hist2_tb_sd.m	atm_6prof.m
plot_map_tb_dd.m	atm_ph_139.m
plot_map_tb_dif.m	humconvert.m
plot_map_tb_sd.m	press2h_interp.m
rad_spc_set.m	rtm_abs_liq.m
xcold_1obs.m	rtm_abs_N2.m
xcold_1obssim.m	rtm_abs_O2.m
xcold_2cal_more.m	rtm_abs_wvp.m
	rtm_atm_abs.m
main:	rtm_atm_updnwell_profmult.m
saterr_main_1sim_imp.m	rtm_atm_updnwell_profone.m
saterr_main_1sim.m	rtm_prof.m
saterr_main_1sim_more_count.m	rtm_tbupdn.m
saterr_main_1sim_more_granule2orbit.m	rtm_toa.m
saterr_main_1sim_more_profcollate.m	
saterr_main_1sim_more_prof.m	set:
saterr_main_1sim_more_prof_queue.m	saterr_set_antennapattern.m
saterr_main_2cal_less.m	saterr_set_attitudeoffset.m

```

saterr_main_2cal.m
saterr_main_2cal_more_1cal.m
saterr_main_2cal_more_2plot.m
saterr_main_3intercal.m
saterr_main_retrieval.m
saterr_main_setting.m
saterr_profcolligate.m

noise:
allandev_2sample_2D.m
decomp_NEDT_1D.m
decomp_NEDT_2D.m
saterr_noise_fft1.m
saterr_noise_poisson.m
saterr_noise_scale.m
saterr_noise_sim_sub.m
saterr_noise_source.m

orbit:
ang2enu.m
ang2ned.m
coscan2az.m
ctscan2az.m
ctscan2enu.m
ctscan2tilt.m
earth_wgs84.m
enu2ang.m
enuo2gc.m
eulerrotate.m
gc2gd.m
gco2enu.m
gd2gc.m
gd_linesightx_ned_ell.m
gd_linesightx_ned.m
interlos.m
ned2ang.m
orbit_kepler.m
orbit_scan2gd_co.m
orbit_scan2gd_ct.m
scan2ned.m
scfov2scaninterp.m
scfov2scaninterp_scheme2.m
scfov2scan.m
scfov2scan_scheme2.m
scgeo2scaz.m
scgeointerp.m
sphere_eia2nadir.m

```

```

saterr_set_conic_mirrorcold.m
saterr_set_crosspol.m
saterr_set_crosstalk.m
saterr_set_faraday.m
saterr_set_farfieldTB.m
saterr_set_fovintrusion.m
saterr_set_geoorbit_kepler.m
saterr_set_geoorbit.m
saterr_set_geoorbit_real_allfromobs.m
saterr_set_geoorbit_real_scfov.m
saterr_set_geoorbit_real_sc.m
saterr_set_interference.m
saterr_set_jump.m
saterr_set_modefarfieldTB.m
saterr_set_noiseadditive_cust.m
saterr_set_noiseadditive.m
saterr_set_noiseadditive_sensor.m
saterr_set_noisePRT.m
saterr_set_noiserng.m
saterr_set_noisesigdep.m
saterr_set_nonlinear.m
saterr_set_oscil.m
saterr_set_oscil_sub_cust.m
saterr_set_oscil_sub_demo.m
saterr_set_oscil_sub_sensor.m
saterr_set_oscil_sub_sensor_table.m
saterr_set_path_scheme.m
saterr_set_polooffset.m
saterr_set_profile.m
saterr_set_rad_fe.m
saterr_set_radspc_adv.m
saterr_set_radspc_chan.m
saterr_set_radspc.m
saterr_set_rad_sr.m
saterr_set_reflcemiss.m
saterr_set_reflcemiss_sat_ana.m
saterr_set_review.m
saterr_set_scanning.m
saterr_set_TBsource.m
saterr_set_TBsource_sidelobe.m
saterr_set_tmpdep.m
saterr_set_warmloadererror.m

utility:
coeff_crosstalk_corr.m
dist_earthxy.m
saterr_const.m

```

sphere_nadir2eia.m	saterr_plot_observational.m
xenu2ned.m	filter_clwc_183.m
others:	filter_precip_land.m
igrf	filter_precip_ocean.m
parse:	idx_orbit_ascdes.m
saterr_parse_antennapattern.m	ind_startend_2ind.m
saterr_parse_attitudeoffset.m	ind_startend_bin.m
saterr_parse_conic_mirrorcold.m	ind_startend_binnum.m
saterr_parse_crosspol.m	ind_startend_cum.m
saterr_parse_crosstalk.m	interp_lin_2D_asc.m
saterr_parse_faraday.m	math_window.m
saterr_parse_farfieldTB.m	movingavg_cal.m
saterr_parse_fovintrusion.m	movingavg_edge_lr.m
saterr_parse_fovintrusionsub.m	movingavg_idx_1D.m
saterr_parse_geoorbit_kepler.m	mtimes_2d3d.m
saterr_parse_geoorbit_real_scfov.m	mtimes_3d3d.m
saterr_parse_geoorbit_real_sc.m	ocean_emissivity.m
saterr_parse_granuel.m	pca_svds.m
saterr_parse_indCT.m	plot_colormap_whitejet.m
saterr_parse_input.m	plot_map_coastline.m
saterr_parse_interference.m	plot_map_lonedgenan.m
saterr_parse_no_alongtrack.m	plot_map_tb_ascdes.m
saterr_parse_noiseadditive_cust.m	plot_subplotnum.m
saterr_parse_noiseadditive.m	plot_subplotnum_rectangle.m
saterr_parse_noiseadditive_sensor.m	polyfitweighted.m
saterr_parse_nonlinear.m	read_bin_KLM_MHS_L1B.m
saterr_parse_orbit.m	read_MHS_chan1_89GHz.m
saterr_parse_path_scheme.m	retr_clwc_amsua.m
saterr_parse_poloffset.m	retr_tmp_atms.m
saterr_parse_profile.m	retr_wind_ssmi.m
saterr_parse_radspc_chan.m	scan_ind2ang.m
saterr_parse_radspc.m	scanpos2angle.m
saterr_parse_rad_sr.m	stat_stdhist.m
saterr_parse_reflmiss.m	sub_collocate.m
saterr_parse_reflmiss_sat_ana.m	sub_filterInd1D.m
saterr_parse_scanning.m	sub_filterInd2D.m
saterr_parse_TBsource.m	sub_gridMapMean_vargrid.m
saterr_parse_warmloaderror.m	sub_hist2.m
pol:	sub_plotCrossSubChan.m
mueller_refl_ct.m	sub_plotDirectHist2DFit.m
	sub_uniqmean.m
	sub_uniqmeanstd.m

Bibliography

- AMSU-A Level 1 Product Generation Specification, *Tech. rep.*, MetOp, European Organisation for the Exploitation of Meteorological Satellites, 2013a.
- MHS Level 1 Product Generation Specification, *Tech. rep.*, MetOp, European Organisation for the Exploitation of Meteorological Satellites, 2013b.
- Nasa global precipitation measurement (gpm) microwave imager (gmi) level 1b (l1b), *Algorithm Theoretical Basis Document (ATBD)*, 2016.
- Ablain, M., et al., Improved sea level record over the satellite altimetry era (1993–2010) from the climate change initiative project, *Ocean Science*, 11(1), 67–82, 2015.
- Alquaied, F., R. Chen, and W. L. Jones, Hot load temperature correction for trmm microwave imager in the legacy brightness temperature, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(6), 1923–1931, 2018.
- Balanis, C., Antenna theory: Analysis and design, 318, 2016.
- Balanis, C. A., *Antenna theory: analysis and design*, John wiley & sons, 2015.
- Bellprat, O., F. Massonet, S. Siegert, C. Prodhomme, D. Macias-Gómez, V. Guemas, and F. Doblas-Reyes, Uncertainty propagation in observational references to climate model scales, *Remote Sensing of Environment*, 203, 101–108, 2017.
- Bilitza, D., D. Altadill, V. Truhlik, V. Shubin, I. Galkin, B. Reinisch, and X. Huang, International reference ionosphere 2016: From ionospheric climate to real-time weather predictions, *Space weather*, 15(2), 418–429, 2017.
- Bormann, N., and P. Bauer, Estimates of spatial and interchannel observation-error characteristics for current sounder radiances for numerical weather prediction. i: Methods and application to atovs data, *Quarterly Journal of the Royal Meteorological Society*, 136(649), 1036–1050, 2010.
- Cai, G., B. M. Chen, and T. H. Lee, *Unmanned rotorcraft systems*, Springer Science & Business Media, 2011.
- Colton, M. C., and G. A. Poe, Intersensor calibration of dmsp ssm/i's: F-8 to f-14, 1987-1997, *IEEE Transactions on Geoscience and Remote Sensing*, 37(1), 418–439, 1999.
- Corbella, I., A. J. Gasiewski, M. Klein, V. Leuski, A. J. Francavilla, and J. R. Piepmeier, On-board accurate calibration of dual-channel radiometers using internal and external references, *IEEE transactions on microwave theory and techniques*, 50(7), 1816–1820, 2002.
- Davis, W. A., and K. K. Agarwal, *Radio Frequency Circuit Design*, 2nd ed., Wiley, New York, doi:10.1002/0471200689, 2011.
- Desroziers, G., L. Berre, B. Chapnik, and P. Poli, Diagnosis of observation, background and analysis-error statistics in observation space, *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 131(613), 3385–3396, 2005.

Draper, D. W., D. A. Newell, F. J. Wentz, S. Krimchansky, and G. M. Skofronick-Jackson, The global precipitation measurement (gpm) microwave imager (gmi): Instrument overview and early on-orbit performance, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(7), 3452–3462, 2015.

Elsaesser, G. S., and C. D. Kummerow, Toward a fully parametric retrieval of the nonrainning parameters over the global oceans, *Journal of applied meteorology and climatology*, 47(6), 1599–1618, 2008.

Gaiser, P. W., et al., The windsat spaceborne polarimetric microwave radiometer: Sensor description and early orbit performance, *IEEE Transactions on Geoscience and Remote Sensing*, 42(11), 2347–2361, 2004.

Geer, A. J., et al., All-sky satellite data assimilation at operational weather forecasting centres, *Quarterly Journal of the Royal Meteorological Society*, 144(713), 1191–1217, 2018.

Goodberlet, M., C. Swift, and J. Wilkerson, Remote sensing of ocean surface winds with the special sensor microwave/imager, *Journal of Geophysical Research: Oceans*, 94(C10), 14,547–14,555, 1989.

Gopalan, K., W. L. Jones, S. Biswas, S. Bilanow, T. Wilheit, and T. Kasparis, A time-varying radiometric bias correction for the trmm microwave imager, *IEEE transactions on geoscience and remote sensing*, 47(11), 3722–3730, 2009.

Griffin, M. D., *Space vehicle design*, AIAA, 2004.

Gruber, A., et al., Validation practices for satellite soil moisture retrievals: What are (the) errors?, *Remote sensing of environment*, 244, 111,806, 2020.

Hollingsworth, A., and P. Lönnberg, The statistical structure of short-range forecast errors as determined from radiosonde data. part i: The wind field, *Tellus A*, 38(2), 111–136, 1986.

Hollmann, R., et al., The esa climate change initiative: Satellite data records for essential climate variables, *Bulletin of the American Meteorological Society*, 94(10), 1541–1552, 2013.

Janjić, T., et al., On the representation error in data assimilation, *Quarterly Journal of the Royal Meteorological Society*, 144(713), 1257–1278, 2018.

Jarnot, R., R. Cofield, J. Waters, D. Flower, and G. E. Peckham, Calibration of the microwave limb sounder on the upper atmosphere research satellite, *Journal of Geophysical Research: Atmospheres*, 101(D6), 9957–9982, 1996.

Joo, S., J. Eyre, and R. Marriott, The impact of metop and other satellite data within the met office global nwp system using an adjoint-based sensitivity method, *Monthly weather review*, 141(10), 3331–3342, 2013.

Kalnay, E., *Atmospheric modeling, data assimilation and predictability*, Cambridge university press, 2003.

Kim, E., C.-H. J. Lyu, K. Anderson, R. Vincent Leslie, and W. J. Blackwell, S-npp atm instrument prelaunch and on-orbit performance evaluation, *Journal of Geophysical Research: Atmospheres*, 119(9), 5653–5670, 2014.

- Kroodsma, R. A., D. S. McKague, and C. S. Ruf, Inter-calibration of microwave radiometers using the vicarious cold calibration double difference method, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(3), 1006–1013, 2012.
- Kuan, D. T., A. A. Sawchuk, T. C. Strand, and P. Chavel, Adaptive noise smoothing filter for images with signal-dependent noise, *IEEE transactions on pattern analysis and machine intelligence*, (2), 165–177, 1985.
- Langland, R. H., and N. L. Baker, Estimation of observation impact using the nrl atmospheric variational data assimilation adjoint system, *Tellus A: Dynamic Meteorology and Oceanography*, 56(3), 189–201, 2004.
- Lee, T. H., *The Design of CMOS Radio-Frequency Integrated Circuits*, Cambridge Univ, doi:10.1017/cbo9780511817281, 2004.
- Li, J., and G. Liu, Direct assimilation of chinese fy-3c microwave temperature sounder-2 radiances in the global grapes system, *Atmospheric Measurement Techniques*, 9(7), 3095–3113, 2016.
- Matsumoto, M., and T. Nishimura, Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator, *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 8(1), 3–30, 1998.
- McKague, D. S., C. S. Ruf, and J. J. Puckett, Beam spoiling correction for spaceborne microwave radiometers using the two-point vicarious calibration method, *IEEE transactions on geoscience and remote sensing*, 49(1), 21–27, 2010.
- Meissner, T., and F. J. Wentz, Polarization rotation and the third stokes parameter: The effects of spacecraft attitude and faraday rotation, *IEEE Transactions on Geoscience and Remote Sensing*, 44(3), 506–515, 2006.
- Merchant, C. J., et al., Uncertainty information in climate data records from earth observation, *Earth System Science Data*, 9(2), 511–527, 2017.
- Merchant, C. J., et al., Satellite-based time-series of sea-surface temperature since 1981 for climate applications, *Scientific data*, 6(1), 1–18, 2019.
- Mo, T., Prelaunch calibration of the advanced microwave sounding unit-A for NOAA-K, *IEEE Transactions on Microwave theory and Techniques*, 44(8), 1460–1469, 1996.
- Peng, J., et al., Soil moisture active/pассив l-band microwave radiometer postlaunch calibration, *IEEE Transactions on Geoscience and Remote Sensing*, 55(9), 5339–5354, 2017.
- Piepmeier, J. R., et al., Smap l-band microwave radiometer: Instrument design and first year on orbit, *IEEE Transactions on Geoscience and Remote Sensing*, 55(4), 1954–1966, 2017.
- Qin, Z., X. Zou, and F. Weng, Analysis of atms striping noise from its earth scene observations, *Journal of Geophysical Research: Atmospheres*, 118(23), 13–214, 2013.
- Riley, W. J., Nist special publication 1065, *Handbook of frequency stability analysis*, p. 60, 2008.
- Robel, J., and A. Graumann, Noaa klm user's guide with noaa-n, *N Prime, and MetOp SUPPLE MEN*: https://www.nasa.gov/pdf/111742main_noaa_n_booklet.pdf, 2014.

- Rosenkranz, P., Line-by-line microwave radiative transfer (non-scattering), *Remote Sensing Code Library*, https://doi.org/10.21982/M_81013, 2017.
- Rosenkranz, P. W., Cloud liquid-water profile retrieval algorithm and validation, *Journal of Geophysical Research: Atmospheres*, 111(D9), 2006.
- Rosenkranz, P. W., and D. Cimini, Speed dependence of 22-and 118-ghz line shapes for tropospheric remote sensing, *IEEE Transactions on Geoscience and Remote Sensing*, 57(12), 9702–9708, 2019.
- Ruf, C. S., Constraints on the polarization purity of a stokes microwave radiometer, *Radio Science*, 33(6), 1617–1639, 1998.
- Schiek, B., H.-J. Siweris, and I. Rolfes, *Noise in high-frequency circuits and oscillators*, John Wiley & Sons, 2006.
- Stewart, L., S. L. Dance, N. K. Nichols, J. Eyre, and J. Cameron, Estimating interchannel observation-error correlations for iasi radiance data in the met office system, *Quarterly Journal of the Royal Meteorological Society*, 140(681), 1236–1244, 2014.
- Sullivan, D. B., D. W. Allan, D. A. Howe, and F. L. Walls, *Characterization of clocks and oscillators*, National Institute of Standards and Technology Technical Note, 1990.
- Thébault, E., et al., International geomagnetic reference field: the 12th generation, *Earth, Planets and Space*, 67(1), 1–19, 2015.
- Twarog, E. M., W. E. Purdy, P. W. Gaiser, K. H. Cheung, and B. E. Kelm, Windsat on-orbit warm load calibration, *IEEE Transactions on Geoscience and Remote Sensing*, 44(3), 516–529, 2006.
- Ulaby, F. T., D. G. Long, W. J. Blackwell, C. Elachi, A. K. Fung, C. Ruf, K. Sarabandi, H. A. Zebker, and J. Van Zyl, *Microwave radar and radiometric remote sensing*, vol. 4, University of Michigan Press Ann Arbor, 2014.
- Vasilescu, G., *Electronic noise and interfering signals: principles and applications*, Springer Science & Business Media, 2005.
- Weng, F., L. Zhao, R. R. Ferraro, G. Poe, X. Li, and N. C. Grody, Advanced microwave sounding unit cloud and precipitation algorithms, *Radio Science*, 38(4), 2003.
- Weng, F., X. Zou, N. Sun, H. Yang, M. Tian, W. J. Blackwell, X. Wang, L. Lin, and K. Anderson, Calibration of suomi national polar-orbiting partnership advanced technology microwave sounder, *Journal of Geophysical Research: Atmospheres*, 118(19), 11–187, 2013.
- Wentz, F. J., and D. Draper, On-orbit absolute calibration of the global precipitation measurement microwave imager, *Journal of Atmospheric and Oceanic Technology*, 33(7), 1393–1412, 2016.
- Wentz, F. J., P. Ashcroft, and C. Gentemann, Post-launch calibration of the trmm microwave imager, *IEEE Transactions on Geoscience and Remote Sensing*, 39(2), 415–422, 2001.
- Weston, P., W. Bell, and J. Eyre, Accounting for correlated error in the assimilation of high-resolution sounder data, *Quarterly Journal of the Royal Meteorological Society*, 140(685), 2420–2429, 2014.

- Yang, J. X., and H. Yang, Radiometry calibration with high-resolution profiles of gpm: Application to atms 183-ghz water vapor channels and comparison against reanalysis profiles, *IEEE Transactions on Geoscience and Remote Sensing*, 57(2), 829–838, 2018.
- Yang, J. X., and H. Yang, A new algorithm for determining the noise equivalent delta temperature of in-orbit microwave radiometers, *IEEE Transactions on Geoscience and Remote Sensing*, 2021.
- Yang, J. X., D. S. McKague, and C. S. Ruf, Identifying and resolving a calibration issue with gmi, in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 2568–2571, IEEE, 2015.
- Yang, J. X., D. S. McKague, and C. S. Ruf, Uncertainties in radiometer intercalibration associated with variability in geophysical parameters, *Journal of Geophysical Research: Atmospheres*, 121(19), 11–348, 2016.
- Yang, J. X., Y. You, W. Blackwell, S. Misra, and R. A. Kroodsma, Quantifying and characterizing striping of microwave humidity sounder with observation and simulation, *IEEE Transactions on Geoscience and Remote Sensing*, 2021.
- Zhu, T., and F. Weng, Hurricane sandy warm-core structure observed from advanced technology microwave sounder, *Geophysical Research Letters*, 40(12), 3325–3330, 2013.