Effective Date: August 09, 2023
Expiration Date: August 09, 2038

GSFC JPSS CMO
November 08, 2023
Released

474-00448-03-02, Revision A
Joint Polar Satellite System (JPSS) Code 474

Joint Polar Satellite System Algorithm Specification Volume III: Operational Algorithm Description (OAD) for the ATMS RDR/TDR/SDR

Goddard Space Flight Center
Greenbelt, Maryland

NOAA / NASA

Joint Polar Satellite System
Algorithm Specification Volume III:
Operational Algorithm Description (OAD) for the ATMS RDR/TDR/SDR
Review/Signature/Approval Page

Prepared by:

LEO Ground Services Project SE

Approved by:

Kellyann Jeletic
LEO Ground Services Project SEIT Lead

Nicolaie Todirita
LEO Ground Services Project Manager

Electronic Approval available on-line at: https://jpssmis.gsfc.nasa.gov/frontmenu_dsp.cfm

Preface

This document is under JPSS Ground configuration control. Once this document is approved, JPSS approved changes are handled in accordance with Class I and Class II change control requirements as described in the JPSS Configuration Management Procedures, and changes to this document shall be made by complete revision.

Any questions should be addressed to:

JPSS Configuration Management Office
NASA/GSFC
Code 474
Greenbelt, MD 20771

Change History Log

Revision	Effective Date	**Description of Changes** (Reference the CCR & CCB/ERB Approval Date)
Rev -	Jul 26, 2013	This was approved by the JPSS Ground ERB via **474-CCR-13-1098** on the effective date shown.
Rev -1	Oct 23, 2014	This version incorporates **474-CCR-14-2094** which was approved by JPSS Ground ERB on the effective date shown. This version was baselined in support of Change Order 10.
Rev -1 - Revalidated	Dec 11, 2019	The expiration date on this document has been extended five years by 474-CCR-19-4789, which was approved by the JPSS Ground AERB on Dec 11, 2019. The effective date and the revision/version on this document will remain the same, only the expiration date has been changed.
Rev A	Aug 09, 2023	This version incorporates 474-CCR-23-6516 which was approved by the Ground ERB on May 01, 2023 and the Ground CCB on May 24, 2023 and 474-CCR-23-6762 which was approved by the Ground Segment ERB on Aug 04,2023 and by the JPSS Ground Segment CCB on the effective date shown.

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

1.1 Objective

The purpose of this Operational Algorithm Description (OAD) document is to express, in computer-science terms, the remote sensing algorithms that produce the Joint Polar Satellite System (JPSS) end-user data products. These products are individually known as Raw Data Records (RDRs), Temperature Data Records (TDRs), Sensor Data Records (SDRs) and Environmental Data Records (EDRs). In addition, any Intermediate Products (IPs) produced in the process. The OAD provides a software description of that science as implemented in the operational ground system.

The purpose of an OAD is two-fold:

1. Provide initial implementation design guidance to the operational software developer.
2. Capture the “as-built” operational implementation of the algorithm reflecting any changes needed to meet operational performance/design requirements.

An individual OAD document describes one or more algorithms used in the production of one or more data products. This particular document describes operational software implementation for the Advanced Technology Microwave Sounder (ATMS) Sensor Data Record (SDR).

1.2 Scope

The scope of this document is limited to the description of the core operational algorithm(s) required to create the ATMS SDR. The basis for the geolocation algorithm is described in this document.

2 RELATED DOCUMENTATION

The latest versions of all documents listed below should be used. The latest JPSS documents can be obtained from URL: https://jpssmis.gsfc.nasa.gov/frontmenu_dsp.cfm. JPSS Project documents have a document number starting with 470, 472 or 474 indicating the governing Configuration Control Board (CCB) (Program, Flight or Ground) that has the control authority of the document.

2.1 Parent Documents

The following document is the Parent Document(s) from which this document has been derived. Any modification to a Parent Document will be reviewed to identify the impact upon this document. In the event of a conflict between a Parent Document and the content of this document, the JPSS Program Configuration Control Board has the final authority for conflict resolution.

Document Number	Title
474-00448-01-02	JPSS Algorithm Specification Volume I: SRS for the ATMS RDR/TDR/SDR

2.2 Applicable Documents

The following documents(are the Applicable Documents from which this document has been derived. Any modification to an Applicable Document will be reviewed to identify the impact upon this document. In the event of conflict between an Applicable Document and the content of this document, the JPSS Program Control Board has the final authority for conflict resolution.

Document Number	Title
429-05-02-42	NPP Mission Data Format Control Book (MDFCB)
429-05-02-42-02	NPP MDFCB Appendix A
472-00251	Mission Data Format Control Book (MDFCB) Joint Polar Satellite System-1 (JPSS-1)
474-00448-02-02	JPSS Algorithm Specification Volume II: Data Dictionary for the ATMS RDR/TDR/SDR
474-00448-04-02	JPSS Algorithm Specification Volume IV: SRS Parameter File (SRSPF) for the ATMS RDR/TDR/SDR

3 ALGORITHM OVERVIEW

This document presents the theoretical basis of the ATMS Sensor Data Record (SDR) Algorithms. The functional flow of algorithms required to transform a Raw Data Record (RDR) coming from the satellite into a Temperature Data Record (TDR) and SDR is described. Engineering, calibration, and science data from the ATMS sensor, along with spacecraft attitude and ephemeris, and user-supplied tunable processing parameters, are processed by the ATMS SDR Module to produce the ATMS SDR consisting of scene brightness temperate. The ATMS SDR Module also produces the ATMS TDR, ATMS GEO, and ATMS IP outputs. Figure 3-1 shows the processing chain.

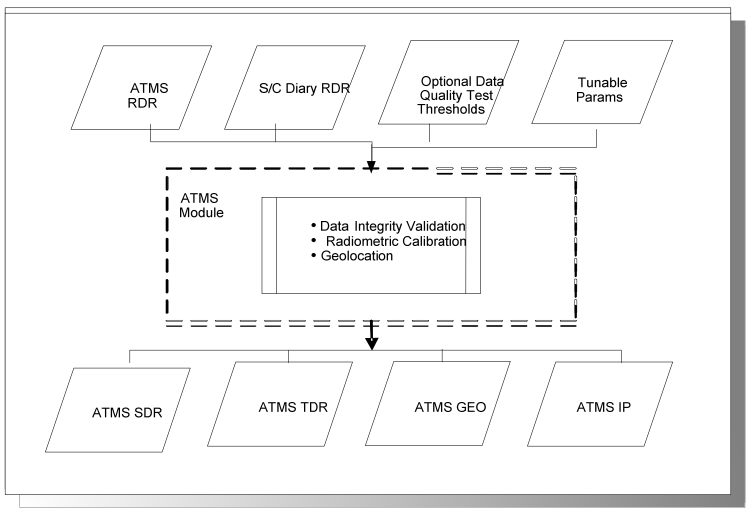

Figure 3-1. Processing Chain

ATMS SDR processing is handled by two separate algorithms, the Verified RDR algorithm and the Calibration and Geolocation algorithm. Both algorithms are managed within a Controller "algorithm", which is run as a single process. To begin data processing, the Infrastructure (INF) Subsystem Software Item (SI) initiates the ATMS algorithms. The INF SI provides tasking information to the algorithms indicating which granule to process. The Data Management Subsystem (DMS) SI provides data storage and retrieval capability. ATMS SDR processing is retaskable, so instead of shutting down after processing, it requests additional tasking information from INF and continues processing with this information. A library of C++ classes, depicted in Figure 3-2, is used to implement the SI interfaces.

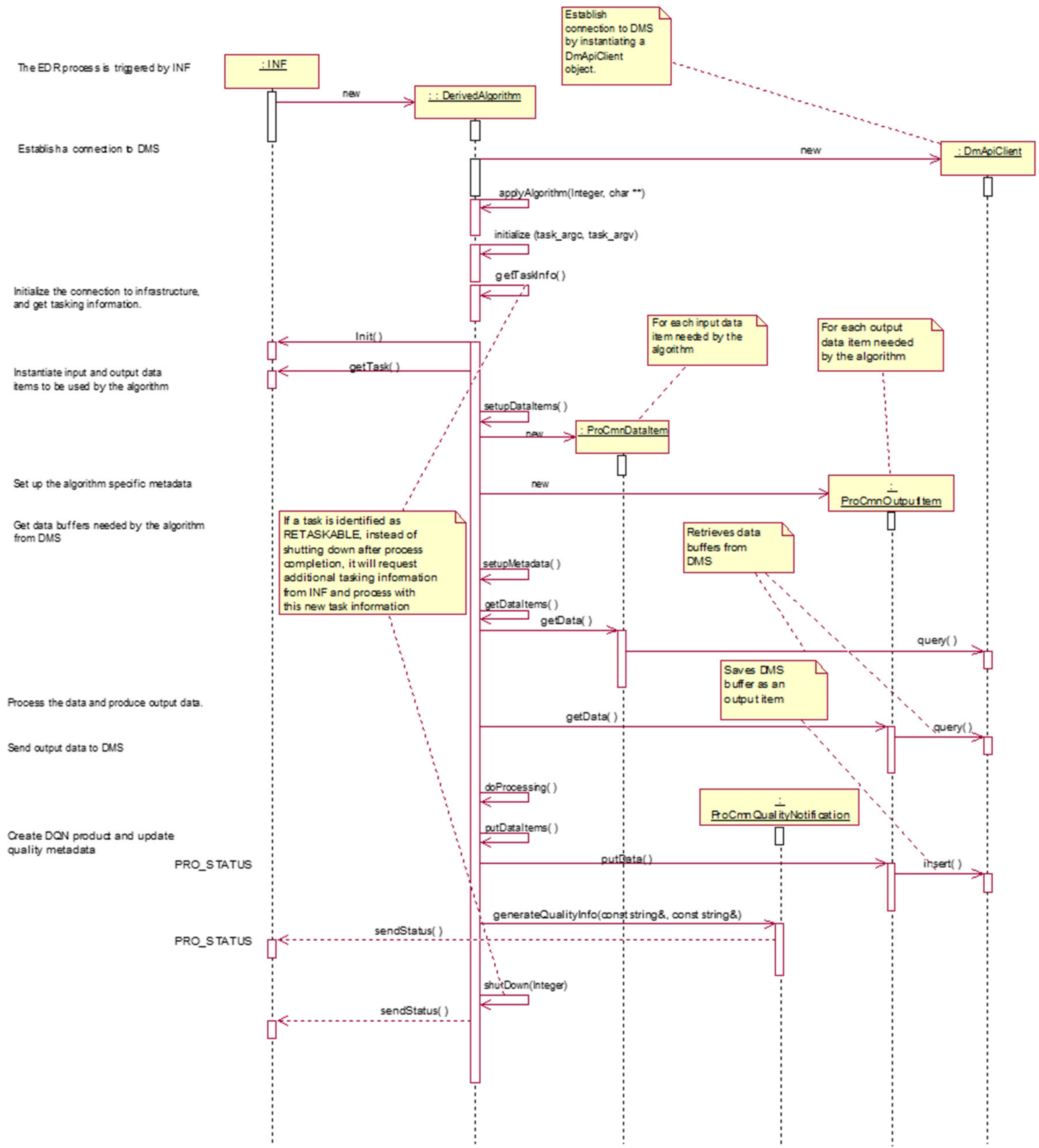

Figure 3-2. IPO Model Interface to INF and DMS

3.1 ATMS Verified RDR Description

3.1.1 Interfaces

3.1.1.1 Inputs

The inputs to the ATMS Verified RDR algorithm are ATMS Science RDRs and the ATMS Configurable Coefficients file as shown in Table 3.1.1.1-1.

Table 3.1.1.1-1. ATMS Verified RDR Algorithm Inputs

Input	Description	Reference Document
ATMS science RDR, along with adjacent intrack RDRs.	Science, calibration and engineering sensor data	474-00448-02-02_JPSS-DD-Vol-II-Part-2
ATMS SDR Coefficients.	Adjustable ATMS SDR coefficients from DMS.	474-00448-02-02_JPSS-DD-Vol-II-Part-2

3.1.1.2 Outputs

The internal output of the ATMS Verified RDR algorithm is a byte-aligned version of the ATMS Science RDR.

3.1.2 Algorithm Processing

ATMS Verified RDR processing retrieves the ATMS Science RDRs from DMS and unpacks and byte-aligns the RDRs to create a Verified RDR. ATMS RDR data in DMS is in CCSDS packet format. This byte-aligned data is used by the Calibration and Geolocation algorithm to create the ATMS SDR and TDR.

The derived algorithm class for the ATMS Verified RDR algorithm, ProSdrAtmsVerifiedRDR, is a subclass of the AutoGeneratedProSdrAtmsVerifiedRDR class, which is a subclass of the ProCmnAlgorithm class. AutoGeneratedProSdrAtmsVerifiedRDR is auto generated at build time and contains member functions for setting up data members with addresses to the input and output products in DMS. The derived algorithm class creates a list of input data items read from DMS and passes required data into the algorithm. An internal Verified RDR is created once the algorithm finishes processing this data, and is then passed to the Geolocation and Calibration algorithm as an input.

3.1.2.1 Created Verified RDRs (ProSdrAtms::createVerifiedAtmsRdr())

The Create Verified ATMS RDR flow diagram is shown in Figure 3.1.2.1-1.

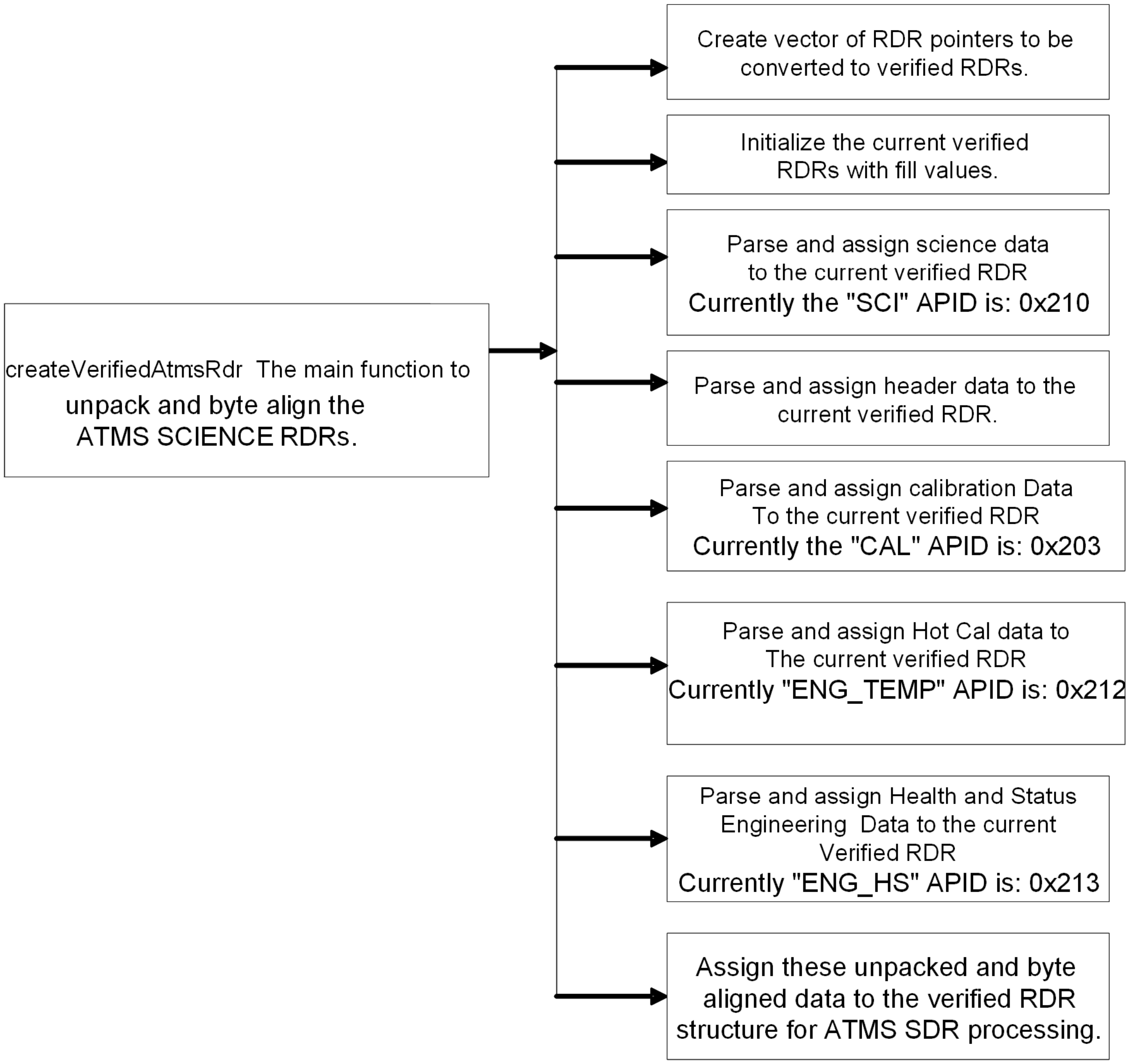

Figure 3.1.2.1-1. Create Verified ATMS RDR Flow Diagram

ATMS Science RDRs contain science, calibration, and engineering application packets. A Verified RDR contains all the data from an RDR that is required for the SDR processing of a tasked granule. The data has been extracted, unpacked and byte aligned from the CCSDS application packets contained in the RDR. In other words, a Verified RDR is a byte-aligned RDR. This unpacked and byte-aligned data is assigned to the internal verified RDR structure, and ATMS SDR processing can use the data from that structure to do further processing.

3.1.3 Graceful Degradation

No graceful degradation is performed.

3.1.4 Exception Handling

No exception handling is performed.

3.1.5 Data Quality Monitoring

No data quality monitoring is performed.

3.1.6 Computational Precision Requirements

There are no computational precision requirements for the ATMS Verified RDR algorithm.

3.1.7 Algorithm Support Considerations

INF and DMS must be running before execution of the algorithm.

3.1.8 Assumptions and Limitations

None.

3.2 ATMS SDR Geolocation and Calibration Description

3.2.1 Interfaces

Many of the input and output parameters are in "counts" or unsigned integer data types. Before any of the main SDR processing tasks are performed, conversion of the input into engineering units must first take place. On output, the inverse operation is used to convert the parameters back to their original units, if so specified in Table 3.2.1.2-1. The "Counts to Engineering Units Conversions" are listed in Table 3.2.1-1.

Table 3.2.1-1. Counts to Engineering Units Conversions

Var. Name	Description	Data Type	Conversion	Units / Range
Time Conversions				
scan_start_time health_time beam_time	Time parameters	real*64	Time * 1.0D-6	IET time
Beam Angle Conversion				
beam_angle	scan angle	real*32	(360.0/(2^{16} – 1)) * (beam_angle_counts – resolver_offset)	deg / 0.0° – 360.0 °
Calibration Parameter Conversions				
pam_kav	PAM resistance for KAV band group	real*32	2300.0 + (0.006*pam_kav_scaled)	ohms /
pam_wg	PAM resistance for WG band group	real*32	2300.0 + (0.006*pam_wg_scaled)	ohms /
prt_coeff_kav (1:ncoef,1:num_prt_kav) where ncoef=4 num_prt_kav=8 **Note**: outermost loop on PRTs; innermost loop on coefs	(1, n) resistance Ro of the n*th* KAV PRT at the ice point (aka 4-W_PRT_KAV_*n*_R0),	num_prt_kav x real*32	Coeff Ro: 1900.0 + (0.003*prt_coeff_kav_scaled(1,1:num_prt_kav))	ohms/
	(2, n) constant α measured for the n*th* KAV PRT (aka 4-W_PRT_KAV_*n*_alpha),	num_prt_kav x real*32	Coeff α: 0.002 + (5.0e-8 * prt_coeff_kav_scaled(2, 1:num_prt_kav))	°C^{-1} /

Var. Name	Description	Data Type	Conversion	Units / Range
	(3, n) constant δ measured for the n^{th} KAV PRT (aka 4-W_PRT_KAV_*n*_delta),	num_prt_kav x real*32	Coeff δ: 5.0e-5 * prt_coeff_kav_scaled(3, 1:num_prt_kav)	ºC /
	(4, n) constant β measured for the n^{th} KAV PRT (aka 4-W_PRT_KAV_*n*_beta)	num_prt_kav x real*32	Coeff β: (3.0e-5 * prt_coeff_kav_scaled(4, 1:num_prt_kav)) – 1.0	ºC /
prt_coeff_wg (1:ncoef,1:num_prt_wg) where ncoef=4 num_prt_wg=7 **Note**: outermost loop on PRTs; innermost loop on coefs	(1, n) resistance Ro of the n^{th} WG PRT at the ice point (aka 4-W_PRT_WG_*n*_R0),	num_prt_wg x real*32	Coeff Ro: 1900.0 + (0.003*prt_coeff_wg_scaled(1,1:num_prt_wg))	ohms/
	(2, n) constant α measured for the n^{th} WG PRT (aka 4-W_PRT_WG_*n*_alpha),	num_prt_wg x real*32	Coeff α: 0.002 + (5.0e-8 * prt_coeff_wg_scaled(2, 1:num_prt_wg))	ºC^{-1} /
	(3, n) constant δ measured for the n^{th} WG PRT (aka 4-W_PRT_WG_*n*_delta),	num_prt_wg x real*32	Coeff δ: 5.0e-5 * prt_coeff_wg_scaled(3, 1:num_prt_wg)	ºC /
	(4, n) constant β measured for the n^{th} WG PRT (aka 4-W_PRT_WG_*n*_beta)	num_prt_wg x real*32	Coeff β: (3.0e-5 * prt_coeff_wg_scaled(4, 1:num_prt_wg)) – 1.0	ºC /
warm_bias(1:5)	PRT warm bias: (1) for channel 1 (K band) (2) for channel 2 (Ka band) (3) for channels 3 to 15 (V band) (4) for channel 16 (W band) (5) for channels 17 to 22 (G band)	5 x real*32	– 7.5e-6 * warm_bias_scaled(1:5)	ºC /
cold_bias(1:5)	PRT cold bias: (1) for channel 1 (K band) (2) for channel 2 (Ka band) (3) for channels 3 to 15 (V band) (4) for channel 16 (W band) (5) for channels 17 to 22 (G band)	5 x real*32	1.5e-5 * cold_bias_scaled(1:5)	ºC /

Var. Name	Description	Data Type	Conversion	Units / Range
quadratic_coeff (1:num_channels) where num_channels=22	Quadratic BT correction factors for each channel (Peak nonlinearity as determined from calibration tests)	num_channels x real*32	(2.6e-5 * quadratic_coeff_scaled(1:num_channels)) – 0.85	K
beam_alignment (1:attitude, 1:beamPosOffset, 1:bands). where attitude = {X, Y, Z} beamPosOffset = {01, 48, 96} bands = 5 **Note**: outermost loop on band; innermost loop on attitude	Alignment offsets (roll, pitch, yaw offsets per band)	3 x 3 x 5 x real*32	(2.0e-5 * beam_alignment_scaled(:,:,:)) – 0.655	deg
prt_coeff_shelf (1:ncoef, *feed*) where ncoef=4, *feed* = {K, V, W, G} **Note**: outermost loop on feed; innermost loop on coefs	(1, *feed*) resistance Ro of *feed* shelf PRT at the ice point (aka *feed* _SHELF_PRT_R0),	4 x real*32	Coeff Ro: 1900 + (0.003*prt_coeff_shelf_scaled(1,*feed*))	ohms/
	(2, *feed*) constant α measured for *feed* shelf PRT (aka *feed* _SHELF_PRT_alpha),	4 x real*32	Coeff α: 0.002 + (5.0e-8 * prt_coeff_shelf_scaled(2, *feed*))	$^{o}C^{-1}$ /
	(3, *feed*) constant δ measured for *feed* shelf PRT (aka *feed* _SHELF_PRT_delta),	4 x real*32	Coeff δ: 5.0e-5 * prt_coeff_shelf_scaled(3, *feed*)	°C /
	(4, *feed*) cable resistance Rc to *feed* shelf PRT (aka *feed*_SHELF_PRT_RC)	4 x real*32	Coeff Rc: (0.0003 * prt_coeff_shelf_scaled(4, *feed*)	ohms

3.2.1.1 Inputs

The implementation of ATMS SDR requires an ATMS Verified RDR, Spacecraft Diary RDRs and adjustable ATMS SDR coefficients (tunable parameters). A general description of the inputs used for ATMS SDR can be found in Table 3.2.1.1-1.

Table 3.2.1.1-1. ATMS SDR Algorithm Inputs

Input	Description	Reference Document
ATMS Verified RDR.	Byte-aligned data from the ATMS Science RDRs.	474-00448-02-02_JPSS-DD-Vol-II-Part-2

Spacecraft Diary RDR, along with adjacent intrack RDRs.	Ephemeris and Attitude Data from DMS.	474-00448-02-02_JPSS-DD-Vol-II-Part-2
ATMS SDR Coefficients.	Adjustable ATMS SDR coefficients from DMS.	474-00448-02-02_JPSS-DD-Vol-II-Part-2
Optional data quality test thresholds.	Data quality notification threshold values.	474-00448-02-02_JPSS-DD-Vol-II-Part-2

The following tunable parameters were removed from the Processing Coefficients file and made constants in the source code. This was done so array dimension constants could be defined and used in array declarations and the fact these parameters can affect the number of cross-granules required for processing.

num_scan_prt	int*16	Number of scans for PRT averaging	-- / 1 – 10
num_scan_wc	int*16	Number of scans of hot calibration target data used in calibration; the maximum number of scans for averaging hot calibration of any channel	-- / 1 – 20
num_scan_cc	int*16	Number of scans of cold calibration target data used in calibration; the maximum number of scans for averaging cold calibration of any channel	-- / 1 – 20

Some parameters may be adjusted after some operational experience has been gained after launch.

3.2.1.2 Outputs

The ATMS SDR algorithm creates an ATMS Temperature Data Record (TDR) (uncorrected brightness temperature), ATMS SDR (corrected brightness temperature), ATMS SDR Geolocation Data, and ATMS SDR Intermediate Product (IP) and stores the outputs in DMS. Table 3.2.1.2-1 shows the ATMS SDR outputs.

Table 3.2.1.2-1. ATMS SDR Outputs

Output	Description	Reference Document
ATMS SDR Data	ATMS SDR contains corrected brightness temperature, noise equivalent delta temperature for cold and warm views, calibration gain factors and quality flags.	474-00448-02-02_JPSS-DD-Vol-II-Part-2
ATMS TDR Data	ATMS TDR contains uncorrected brightness temperature and quality flags.	474-00448-02-02_JPSS-DD-Vol-II-Part-2
ATMS SDR GEO Data	ATMS SDR Geolocation contains scan's start and mid times;Solar Zenith Angle, Solar Azimuth Angle, Sensor Zenith Angle, Sensor Azimuth Angle, and Terrain Height, Range; Spacecraft's	474-00448-02-02_JPSS-DD-Vol-II-Part-2

Output	Description	Reference Document
	position, velocity and attitude at scan's mid-time; and geodetic latitude and longitude values. Also this contains the Geolocation Error Flag.	
ATMS SDR IP	Processing coefficients and other values calculated which are used by cal/val. Warm bias, cold bias, PRT KAV Coefficients, PRT WG Coefficients, PAM resistance counts for KAV and WG, PRT counts for KAV and WG from hot calibration data etc.	Refer to Table 3.2.1.2-2

Note: The values for the constants referenced in Table 3.2.1.2-2 are as follows:

NUM_ATMS_SCANS 12

NUM_CHANNELS 22

NUM_CAL_SCANS 4

NUM_PRT_COEFFS 4

NUM_PRT_KAV 8

NUM_PRT_WG 7

Table 3.2.1.2-2. ATMS SDR IP

Output	Type/size	Description	Units/valid range
Scan level			
PrtKavCoeffs	Int32 * NUM_CAL_SCANS * NUM_PRT_COEFFS * NUM_PRT_KAV	PRT(KAV) counts-to-temperature conversion coefficients (4x8)	Unitless
prtWgCoeffs	Int32 * NUM_CAL_SCANS * NUM_PRT_COEFFS * NUM_PRT_WG	PRT(WG) counts-to_temperature conversion coefficients (4x7)	Unitless
pamKav	Int32 * NUM_CAL_SCANS	PAM resistance counts for KAV	Unitless
pamWg	Int32 * NUM_CAL_SCANS	PAM resistance counts for WG	Unitless
prtKavCounts	Int32 * NUM_ATMS_SCANS * NUM_PRT_KAV	PRT counts (KAV) from hot cal data	Unitless
prtWgCounts	Int32 * NUM_ATMS_SCANS * NUM_PRT_WG	PRT counts (WG) from hot cal data	Unitless

Output	Type/size	Description	Units/valid range
pamKavCounts	Int32 * NUM_ATMS_SCANS	PAM counts (KAV) calibration data	Unitless
pamWgCounts	Int32 * NUM_ATMS_SCANS	PAM counts (WG) calibration data	Unitless
multiplexRef	Int32 * NUM_ATMS_SCANS	Multiplexer reference	Unitless
Channel level			
warmBias	Int32 * NUM_CAL_SCANS * NUM_CHANNELS	Warm target bias corrections	Unitless
coldBias	Int32 * NUM_CAL_SCANS * NUM_CHANNELS	Cold space bias corrections	Unitless
quadraticCoeff	Int32 * NUM_CAL_SCANS * NUM_CHANNELS	Calibration nonlinearity (quadratic) factors	Unitless

3.2.2 Algorithm Processing

The algorithm uses a Common Geolocation (CMN GEO) library of functions to calculate geodetic latitude and longitude along with sun azimuth and zenith angles. Geolocation is performed using a combination of sensor specific functions and the Common Geolocation library of functions.

The data integrity module checks the quality of radiometric and other data. If the data are not suitable for processing, the proper flags are set and the processor continues with the calibration module. The calibration module converts raw radiometric scene information into brightness temperatures.

The overall ATMS SDR top-level flow diagram is illustrated in Figure 3.2.2-1.

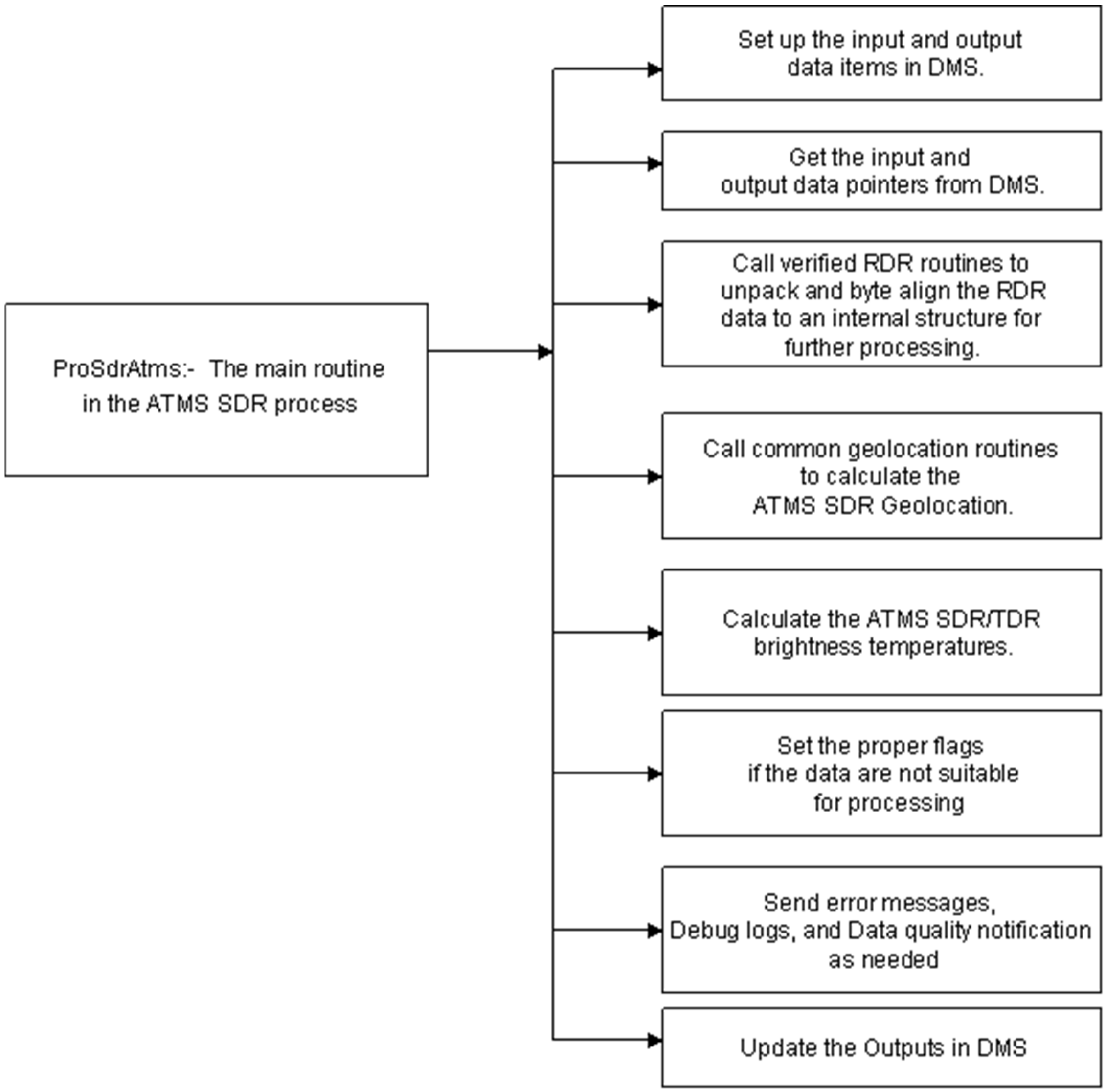

Figure 3.2.2-1. ATMS SDR Top Level Flow Diagram

3.2.2.1 Detailed Design of Full Radiance Calibration

ATMS SDR Calibration processing is calculating full radiance as sum of linear part and nonlinear part and then transfer radiance to temperature. See details below. Implementation of ATMS SDR Full Radiance Calibration is:

- The spectral radiance of cold end is determined at side lobe corrected cosmic background temperature of 2.73K
- The spectral radiance of warm target radiance is calculated at bias corrected warm load temperature
- Compute calibration gain in radiance, by which the linear part of scene radiance can be derived
- Calculate "µ" parameter from receiver temperature (in ºC), from which the maximum nonlinearity Q^{max} can be derived.

- Derive the nonlinear part of scene radiance from Q^{max}, find the calibrated scene radiance from sum of linear and nonlinear part
- Transfer spectral scene radiance back to brightness temperature by inverse Planck function

The scene radiance is derived as the sum of linear part and nonlinear part:

$$R_b = R_{b,1} + Q_b$$

where the linear term is

$$R_{b,1} = R_w + G_b^{-1}(c_s - c_w)$$

$$G_b = \frac{c_w - c_c}{R_w - R_c}$$

The brightness temperature can also be corrected for nonlinearity by the addition of a quadratic term. This correction is applied when the user-input variable use_quadratic_term is set to *.true.*

And nonlinear term is

$$Q_b = \mu G_b^{-2}(C_s - C_W)(C_s - C_c) = \mu(R_w - R_c)^2 x(x - 1)$$

$$x = \frac{C_s - C_c}{C_w - C_c}$$

Using Taylor expansion for f(x) = x(x-1) at x0=0.5, Nonlinearity term can be expressed as function of the maximum nonlinearity:

$$Q_b = Q^{max}[4 \cdot (x - 0.5)^2 - 1]$$
$$Q^{max} = \frac{1}{4} . \mu . (R_w - R_c)^2$$

"μ" is a function of instrument temperature and can be determined from TVAC test

$$\mu = aT^2 + bT + c$$

3.2.2.1.1 PRT Conversion

The hot target physical temperature is determined as a weighted average value derived from the embedded PRTs plus bias-like correction factors. The KAV target contains num_prt_kav (eight) PRTs and the WG target contains num_prt_wg (seven) PRTs. In addition, the instrument is also equipped with shelf PRTs that provide readings of the receiver's physical temperature which is used to determine the hot target bias and to interpolate the quadratic correction coefficients. Conversion of PRT counts to resistance and resistance to physical temperatures are shown below.

3.2.2.1.1.1 Conversion from Counts to Resistance

Two forms of the counts to resistance equations are used, one for hot target PRTs (4-wire PRTs) and one for the receiver shelf 2-wire PRTs:

$$R_i = R_{ref}\left(\frac{C_i - C_{off}}{C_{ref} - C_{off}}\right), \text{ for hot target 4-wire PRTs}$$

$$R_i = R_{ref}\left(\frac{C_i - C_{off}}{C_{ref} - C_{off}}\right) - R_c, \text{ for receiver shelf 2-wire PRTs}$$

where

C_i	=	Number of PRT counts for the i^{th} PRT
C_{ref}	=	PAM counts
C_{off}	=	Multiplexer reference counts (shorted input) obtained from the most recent Engineering Health and Status data packet, variable 4W_GND_A or 4W_GND_B
R_{ref}	=	PAM (Precision Analog Monitor) resistance in ohms
R_c	=	Cable resistance to the shelf PRT in ohms
R_i	=	PRT resistance in ohms for the i^{th} PRT

All inputs are obtained from telemetry. The case of division-by-zero when $C_{ref} = C_{off}$ occurs is flagged in prt_conversion_kav_flag, prt_conversion_wg_flag or prt_conversion_shelf_flag.

3.2.2.1.1.2 Conversion from Resistance to Physical Temperature

The resistance, R, calculated in Section 3.2.2.1.1.1 is related to the physical temperature, T_c, of a given PRT by the Callendar-Van Dusen equation:

$$R_i = R_{o,i}\left[1 + \alpha_i\left(T_c - \delta_i\left(\frac{T_c}{100} - 1\right)\left(\frac{T_c}{100}\right) - \beta_i\left(\frac{T_c}{100} - 1\right)\left(\frac{T_c}{100}\right)^3\right)\right]$$

where

T_c	=	the physical temperature, in degrees Centigrade
R_i	=	PRT resistance in ohms calculated in Section 3.2.2.1.1
$R_{o,i}$	=	the measured resistance in ohms of the specified PRT at the ice point
α_i, δ_i, β_i	=	constants measured for the specified (subscript *i*) PRT. Note for the shelf PRTs, β_i is 0.

Note that all PRT constants, Ro, α, δ, and β, are obtained from telemetered data except for the shelf PRT constant β which must be set to zero. For a bad PRT, the flight software sets Ro,i to zero.

The above equation is solved for Tc by iteration using the Newton-Raphson method with an initial guess of Tc as

$$T_c = \frac{R_i - R_{o,i}}{R_{o,i}\alpha_i}$$

and a loop count initialization of iloop = 0.

The Newton Raphson method of solution is implemented by performing the algorithm below:

$$do$$

$$f(T_c) = R_{o,i}\left[1 + \alpha_i\left(T_c - \delta_i\left(\frac{T_c}{100} - 1\right)\left(\frac{T_c}{100}\right) - \beta_i\left(\frac{T_c}{100} - 1\right)\left(\frac{T_c}{100}\right)^3\right)\right] - R_i$$

$$f'(T_c) = R_{o,i}\alpha_i\left(1 - \delta_i\left(\frac{T_c}{5000} - 0.01\right) - \beta_i\left(\frac{T_c^3}{2.5\times10^7} - \frac{3T_c^2}{1\times10^6}\right)\right)$$

$$T_c = T_c - \frac{f(T_c)}{f'(T_c)}$$

$$iloop = iloop + 1$$

$$while\left(\left(\left|\frac{f(T_c)}{f'(T_c)}\right| > prt\_convergence\right) \| (iloop < prt\_loops)\right)$$

To avoid the off-nominal condition of an infinite iteration, a limit on the number of iteration loops, prt_loops, should be set and the loop exited with an error condition if the maximum number of iterations is exceeded. When this happens it is flagged in prt_conversion_kav_flag, prt_conversion_wg_flag or prt_conversion_shelf_flag.

3.2.2.1.2 Calculation of Average Hot Target PRT Temperature, $\overline{T}_W$

Separate average hot target temperatures are calculated for the KAV and WG targets. These temperatures are determined from multiple PRT temperatures that are weight-averaged over Np (aka, num_scan_prt) scans. Nominally, the current scan, isc, is bounded by istart = isc – (num_scan_prt/2) and iend = istart + (num_scan_prt - 1) scans.

The basic equation for calculating the average hot target temperature for a specified target (i.e., KAV or WG) is

$$\overline{T}_W = \frac{\sum_{i}^{num\_prt}\sum_{n}^{N_p} T_{K,n,i}W_{n,i}}{\sum_{i}^{num\_prt}\sum_{n}^{N_p} W_{n,i}}$$

where

$T_{K,n,i}$ = PRT temperature in Kelvin for the i^{th} PRT at the n^{th} scan
$W_{n,i}$ = Scan weights for the i^{th} PRT at the n^{th} scan

If a PRT is deemed to be bad permanently by the user, its weights in the parameter file should be set to zero. It is then excluded from the calibration process and quality checks.

For the calculation of the hot target temperature, the limits check and the self consistency check can be turned on (or off) via chk_consistency_prt in the parameter file. The data sufficiency check is designed to be always on.

1) PRT quality check – limits
The converted warm load PRT temperatures are checked against predetermined gross limits: low_limit_prt and upp_limit_prt. Those which fall outside the limits are considered "bad" and flagged by setting the quality flag (prt_limit_kav_flag or prt_limit_wg_flag) on. Then the PRT temperatures, along with their weights, are set to zeros:

$$T_i < T_{low} \text{ or } T_i > T_{upp} \rightarrow \text{"bad-}T_i\text{"}$$

2) PRT quality check – self consistency
The PRT temperatures are next checked for internal consistency. This is done by comparing all temperatures not flagged as bad with each other. Any PRT's temperature that differs by more than a fixed limit, max_var_prt, from at least two other PRTs' readings is considered "bad" and flagged by setting the quality flag (prt_consistency_kav_flag or prt_consistency_wg_flag) on. Then the PRT temperatures, along with their weights, are set to zeros:

$$|T_i - T_j| > \Delta T_{max} \text{ and } |T_i - T_n| > \Delta T_{max} \rightarrow \text{"bad-}T_i\text{"}$$

The number of "good" PRTs is then checked. If there are less than num_threshold_prt[1] "good" PRTs for KAV or num_threshold_prt[2] for WG, all PRTs within that group are considered "bad". The PRT temperatures, along with their weights, are set to zeros. The quality flag (prt_consistency_kav_flag or prt_consistency_wg_flag) is set on.

3) PRT quality check – data sufficiency
If the weight-sum of all (KAV or WG) PRT readings over all num_scan_prt scans not flagged as "bad" falls below a specified percentage, wt_threshold_prt, of the total weights for PRT averaging, it is not possible to reliably determine this warm load temperature for the current calibration cycle:

$$(\textstyle\sum_i W_i)/W_{total} < W_{threshold\_prt} \rightarrow \text{"bad-}\overline{T}_W\text{"}$$

where $W_i = 0$ for "bad-T_i". The cycle is then flagged by turning prt_data_sufficiency_kav_flag or prt_data_sufficiency_kav_flag on. Failing the data sufficiency test results in an unsuccessful calibration cycle.

3.2.2.1.3 Shelf Temperature Calculations, T_{SHELF}

Besides being used to determine the quadratic correction to the radiometric transfer function, the receiver temperature is also used to determine a correction to the average hot target temperature determined above. This is implemented by computing the bias as a quadratic polynomial function of the relevant PRT temperature. There are four shelf PRTs: KKA_SHELF_PRT, V_SHELF_PRT, W_SHELF_PRT and G_SHELF_PRT.

To calculate the shelf PRT temperature for a specified receiver shelf, the following steps are performed:

Convert the PRT counts to resistance according to the 2-wire shelf PRT equation in Section 3.2.2.1.1.1.

Convert the PRT resistance to physical temperature, TSHELF, according to the algorithm discussed in Section 3.2.2.1.1.2.

If division-by-zero condition exists, or if the computation loop fails to converge, the condition is flagged in prt_conversion_shelf_flag. The most recent shelf temperature saved in the buffer bptemp_backup is then used. If the temperature is out of range, it is assigned the upper or lower bound. The shelf temperature in the common buffer bptemp_backup is then updated with the current value.

3.2.2.1.4 Calculation of Effective Hot Target Temperature

An effective hot target temperature in Kelvin, with bias added, is calculated for each of the 22 ATMS channels according to the equation below:

$$T_{WC,chan} = \overline{T}_W + \begin{cases} warmBias_{band} \\ a_{chan,1} + a_{chan,2} T_{SHELF} + a_{chan,3} T_{SHELF}^2 \end{cases}$$

where

$\overline{T}_W$	=	Averaged warm load PRT temperature in Kelvin (Section 3.2.2.1.2)
T_{SHELF}	=	Shelf PRT temperature in degrees Centigrade (Section 3.2.2.1.3)
$warmBias_{band}$	=	PRT warm bias in degrees Centigrade for each of the five bands from telemetry file
$a_{chan,k}$	=	Quadratic correction coefficients, k=1:3, for each ATMS channel from user input

Note that the $warmBias_{band}$ variable is telemetered data; the other variables are user inputs. The option is controlled by the switch use_warm_bias_tele in the parameter file. If a shelf PRT temperature is unavailable (filled value), the last saved shelf temperature is used. If that is not available either, then the temperature dependent corrections are ignored by assuming $T_{SHELF} = 0$.

3.2.2.1.5 Calculation of Effective Cold Calibration Brightness Temperature

The cold calibration brightness temperature for each ATMS channel is determined as follows:

Where

$T_{cbc,chan}$	=	Cosmic background radiometric temperature for each channel in Kelvin
$coldBias_{band}$	=	Cold calibration offset bias for each of the five bands from telemetry

$coldBias_{chan,sv}$ = Cold calibration offset bias for each of the 22 channels and each of the four space views from user input

The cold bias variable, $coldBias_{band}$, is telemetered data and $coldBias_{chan,sv}$ is user input from the parameter file. This accounts for antenna sidelobe energy. Four sets of $coldBias_{chan,sv}$ parameters corresponding to four space view groups are stored in the parameter file. The space view group currently used as indicated in the telemetry data (Scan Pattern ID bits of Instrument Mode Word in Engineering H&S packet) determines the appropriate set of $coldBias_{chan,sv}$ to be used. The choice between the bias in the telemetry and that in the parameter file is controlled by the switch use_cold_bias_tele in the parameter file.

3.2.2.1.6 Average Number of Counts from the Warm Calibration View, CWCA

The limits check, the self consistency check and the gain error check are performed if the user-specified parameter chk_consistency_wc_cc in the parameter file is turned on. The data sufficiency check is designed to be always on. These checks on warm counts are:

1) Quality check – limits
The warm counts are checked against pre-defined channel-specific gross limits. Those which fall outside the limits are flagged as "bad" by setting them to zero.

$$Cw_i < Cw_{low} \text{ or } Cw_i > Cw_{upp} \rightarrow \text{"bad-}Cw_i\text{"}$$

2) Quality check – self consistency
The counts are next checked for internal consistency. This is done by checking each count not flagged as "bad" against other counts in the same scan. Any count that differs by more than a fixed limit, max_var_wc, from at least two other counts is flagged as "bad". The count is set to zero.

$$|Cw_i - Cw_j| > \Delta Cw_{max} \text{ and } |Cw_i - Cw_k| > \Delta Cw_{max} \rightarrow \text{"bad-}Cw_i\text{"}$$

The number of "good" samples is then checked. If there are less than three "good" samples, all counts within this scan are flagged as "bad" and the counts are set to zeros.

3) Quality check – gain error
If the lowest "good" warm count is smaller than or equal to the highest "good" cold count, all the warm counts and cold counts with this scan are flagged as "bad" and the counts are set to zeros.

4) Quality check – data sufficiency
The counts from the warm calibration views are weight-averaged over N_{wc} scans. Within each scan, at least three "good" warm target samples are present. Otherwise, that scan is excluded in the count averaging. If the weight-sum of all scans not flagged as "bad" falls below a specified percentage, wt_threshold_wc, of the total weight for warm count averaging, it is deemed not possible to reliably determine the averaged warm count of this channel for the current calibration cycle. This channel is then flagged by turning wc_data_sufficiency_flag on. Failing the data sufficiency test shall result in an unsuccessful

calibration cycle for this channel. The brightness temperatures (TK) and corrected brightness temperatures (TKB) are then filled with fill value ERR_FLOAT32_FILL.

The process is represented as follows:

$$C_{WC,chan} = \sum_{n}^{Nwc}\left(scanWeight\_wc_{n,chan}\left(\frac{1}{num\_samples_{n,chan}}\right)^{num\_samples_{n,chan}} \sum_{j} C_{RWC,j,n,chan}\right)$$

$$W_{WC,chan} = \sum_{n}^{Nwc}\left(scanWeight\_wc_{n,chan}\right)$$

$$if\,(W_{WC,chan} \geq Weight\_threshold\_wc)$$

$$C_{WCA,chan} = \frac{C_{WC,chan}}{W_{WC,chan}}$$

$$else$$

$$C_{WCA,chan} = 0$$

$$wc\_data\_sufficiency\_flag = on$$

$$endif$$

where

$num\_samples_{n,chan}$	=	Number of good warm target samples: three or four.
$C_{RWC,j,n,chan}$	=	Number of raw counts from each of the num_warm_samples hot calibration target at each of the Nwc scans and each channel.
$scanWeight\_wc_{n,chan}$	=	Normalized weight applied for each scan. Weights are re-set to zero for “bad” or “missing”scans, therefore, $\sum_{n} scanWeight\_wc_{n,chan} \leq 1.0$
$Weight\_threshold\_wc$	=	Minimum weight percentage required to accept $C_{WCA,chan}$. It is a tunable parameter from user input.

3.2.2.1.7 Average Number of Counts from the Cold Calibration View, CCCA

The limits check, the self consistency check and the gain error check are performed if the user-specified parameter chk_consistency_wc_cc in the parameter file is turned on. The data sufficiency check is designed to be always on. Similar to the warm count quality checks, these checks on cold counts are:

1) Quality check – limits

The cold counts are checked against pre-defined channel-specific gross limits. Those which fall outside the limits are flagged as “bad” by setting them to zero.

$Cc_i < Cc_{low}$ or $Cc_i > Cc_{upp} \rightarrow$ "bad-Cc_i".

2) Quality check – self consistency
The counts are next checked for internal consistency. This is done by checking each count not flagged as "bad" against other counts in the same scan. Any count that differs by more than a fixed limit, max_var_cc, from at least two other counts is flagged as "bad". The count is set to zero.

$|Cc_i - Cc_j| > \Delta Cc_{max}$ and $|Cc_i - Cc_k| > \Delta Cc_{max} \rightarrow$ "bad-Cc_i"

The number of "good" samples is then checked. If there are less than three "good" samples, all counts within this scan are flagged as "bad" and the counts are set to zeros.

3) Quality check – gain error
If the lowest "good" warm count is smaller than or equal to the highest "good" cold count, all the warm counts and cold counts with this scan are flagged as "bad" and the counts are set to zeros.

4) Quality check – data sufficiency
The counts from the cold calibration views are averaged over N_{cc} scans. Within each scan, at least three "good" cold target samples are present. Otherwise, that scan is excluded in the count averaging. If the weight-sum of all scans not flagged as "bad" falls below a specified percentage, wt_threshold_cc, of the total weight for cold count averaging, it is deemed not possible to reliably determine the averaged cold count of this channel for the current calibration cycle. This channel is then flagged by turning cc_data_sufficiency_flag on. Failing the data sufficiency test shall result in an unsuccessful calibration cycle for this channel. The brightness temperatures (TK) and corrected brightness temperatures (TKB) are then filled with fill value ERR_FLOAT32_FILL.

The process is represented as follows:

$$C_{CC,chan} = \sum_{n}^{Ncc} \left(scanWeight\_cc_{n,chan} \left(\frac{1}{num\_samples_{n,chan}} \right)^{num\_samples_{n,chan}} \sum_{j} C_{RCC,j,n,chan} \right)$$

$$W_{CC,chan} = \sum_{n}^{Ncc} \left(scanWeight\_cc_{n,chan} \right)$$

$$if\,(W_{CC,chan} \geq Weight\_threshold\_cc)$$

$$\quad C_{CCA,chan} = \frac{C_{CC,chan}}{W_{CC,chan}}$$

$$else$$

$$\quad C_{CCA,chan} = 0$$

$$\quad cc\_data\_sufficiency\_flag = on$$

$$endif$$

where

$num\_samples_{n,chan}$	=	Number of good cold target samples: three or four.
$C_{RCC,j,n,chan}$	=	Number of raw counts from each of the num_cold_samples cold calibration targets for each of the Ncc scans and each ATMS channel.
$scanWeight\_cc_{n,\ chan}$	=	Normalized weight applied for each scan. Weights are reset to zero for “bad” or “missing”scans, therefore, $\sum_n scanWeight\_cc_{n,chan} \le 1.0$
$Weight\_threshold\_cc$	=	Minimum weight percentage required to accept $C_{CCA,chan}$. It is a tunable parameter from user input.

3.2.2.1.8 Scan Position Dependent Correction to Brightness Temperature

Once the brightness temperature is calculated for each channel, it is finally adjusted by scan position-dependent correction terms:

$$T_{BB,chan,sp} = beamEfficiencyCorrection_{chan,sp} T_{B,chan,sp} + scanBias_{chan,sp}$$

where

$\mathrm{beamEfficiencyCorrection}_{chan,\ sp}$	=	Scan-position dependent beam efficiency correction factor for each channel
$\mathrm{scanBias}_{chan,\ sp}$	=	Scan-position dependent bias for each channel

Note that *beamEfficiencyCorrection* and *scanBias* are user-input variables.

3.2.2.1.9 Missing PRT Count, Warm Count or Cold Count

The algorithm recognizes a missing PRT count, warm count or cold count by comparing it to the fill value (assumed to be zero). Missing count is excluded in the calibration.

3.2.2.2 ProSdrAtms (ProSdrAtms.cpp)

This is the derived algorithm class for the ATMS SDR Calibration and Geolocation algorithm and is a subclass of the AutoGeneratedProSdrAtms class. ProSdrAtms reads all data items required by the algorithm from DMS and then geolocates and calibrates the data.

ATMS SDR algorithm uses common geolocation routines to calculate the ATMS SDR geolocation. Since the common geolocation is done in C++, ATMS SDR calculates the data necessary for the common geolocation process in the derived algorithm (ProSdrAtms). ATMS SDR calls the common geolocation routines to get the ATMS SDR GEO and it is stored in the DMS with the corresponding granule ID.

After the SDR geolocation, it passes the data into the FORTRAN side of the algorithm to do calibration. The createAtmsSdr subroutine calls the other functions to calculate brightness temperatures. The status returned from the algorithm is checked and if the algorithm did not complete successfully, the derived algorithm sends status to INF that the SDR was not generated successfully via the common code. If the algorithm completed successfully, the SDR, TDR, IP, and GEO are stored in DMS with the corresponding Granule ID. Also, it checks the quality of the SDR products and sends Data Quality Notification accordingly.

3.2.2.3 calcAtmsSdrs (calcAtmsSdrs.f)

In FORTRAN, this is the main module for the ATMS SDR calibration process. First associate the pointers in atms_struct module with addresses passed in from the derived algorithm (these are pointers to DMS memory). Since the ATMS SDR calibration is done in FORTRAN, the pointers from the CPP side need to be associated to FORTRAN structures to access DMS memory so that data from DMS can be used for further processing. Then subroutines are called to convert the data from counts to corresponding engineering units and calculate the brightness temperatures.

3.2.2.4 Initialize the ATMS SDR Outputs and Flags (init_atms_sdr_flags.f)

init_atms_sdr_flags initializes all the sdr flags. Latitude and longitude values are initialized to float fill values in the ProSdrAtms.cpp. Corrected and uncorrected brightness temperatures are also initialized to float fill values.

3.2.2.5 NeDT Calculation (sdrs.f - routine compute_nedt)

The NEDT calculation is done on the 4 cold sky readings and 4 warm load readings of an individual scan. With only four counts, it has a large variance between scans and needs to be used for quality control with care. It is not averaged over multiple scans. It is not corrected to the 300 K specification value. The gain is calculated using the weighted scan profiles contained in the tunable processing coefficient files. The equation is:

NEDT = G* SQRT { S [(xi-xbar)2]/3}

where G is the gain of the system, xi are the raw counts, and xbar is the mean value of the four readings.

3.2.2.6 Calibration Module (c2k_new.f)

The radiometric calibration algorithm converts radiometric counts from each channel and ATMS beam position into calibrated brightness temperatures, in units of Kelvin. The SDR calibration flowchart of this module is shown in Figure 3.2.2.6-1. Calibration inputs and calibration outputs are listed in Table 3.2.2.6-1 and Table 3.2.2.6-2, respectively.

Each ATMS scan is comprised of radiometric measurements for 22 channels collected from 96 ATMS beam positions and four warm and four cold target calibration positions. The physical temperature of the in-flight hot calibration target is determined from the averaged read-outs from Platinum Resistance Temperature (PRT) sensors. A precision resistor, referred to as a Precision Analog Monitor (PAM), provides a calibration of the resistance-to-digital counts conversion. Warm calibration target readings and cold calibration space view readings are averaged over a number of scans. The number is optimized for the warm and cold readings separately and for each channel. PRT readings are also averaged over a number of scans, num_scan_prt. Use of the quadratic term can be enabled or disabled by user through the variable use_quadratic_term in the parameter file. If use of quadratic term is enabled, scene radiance is derived as the sum of linear part and nonlinear part. Then spectral scene radiance will be transfer back to brightness temperature by inverse Planck function.

In the case of a bad PRT, the flight software has the vendor-supplied PRT constant, R0 (resistance at ice point of the PRT, see Calibration Packet parameters in the NPP MDFCB, 429-05-02-42), set as zero. If a PRT is deemed to be bad permanently by the user, its weights in the parameter file should be set to zero. It is then excluded from the calibration process and quality checks.

Limits check and self consistency check can be turned on or off through users' parameter input. The data sufficiency check is always on. Those failing to pass the quality checks are flagged and discarded from the average PRT temperature calculation.

Similarly, the same types of quality checks are applied to warm target counts and cold space view counts. Those failing to pass their quality checks are eliminated from the calibration process and their corresponding quality flags are set accordingly.

Error flags set in this module are explained in 474-00448-02-02_JPSS-DD-Vol-II-Part-02, Section 6.2.2 which contains information about the quality flags set in the ATMS SDR outputs.

The science code implementation of calibration is contained in subroutine c2k_new.f which is called from subroutine sdrone.f.

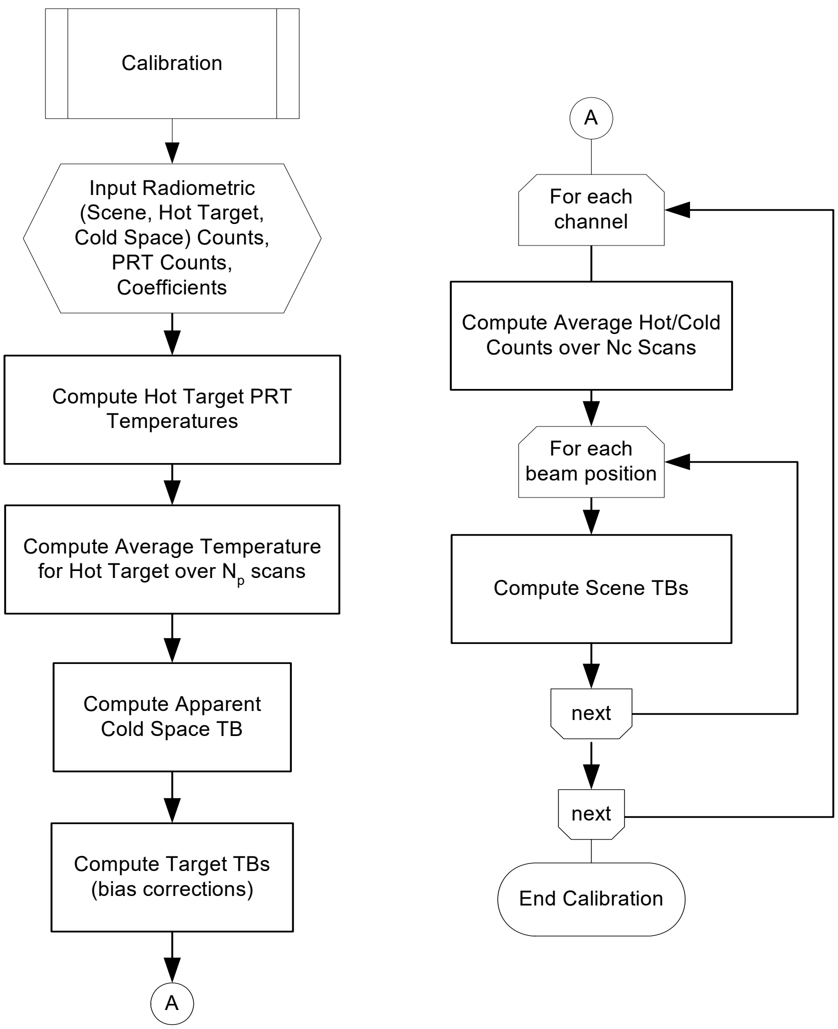

Figure 3.2.2.6-1. ATMS SDR Calibration Flow

Table 3.2.2.6-1. Calibration Module Inputs

Input Var. Name	Description	Type	Units
scnt	Scene counts for the current scan	(NUM_CHANNELS * NUM_BEAM_POSITIONS)* REAL * 8	counts
crw	Raw warm counts for each channel over a specified number of scans, num_scans_wc, bounding the current scan	(NUM_CHANNELS * NUM_WARM_SAMPLES * NUM_SCAN_WC) * REAL * 8	counts
crc	Raw cold counts for each channel over a specified number of scans, num_scans_wc, bounding the current scan	(NUM_CHANNELS* NUM_COLD_SAMPLES* NUM_SCAN_WC) * REAL *8	counts
num_prts	Number of PRTs (KAV bands or WG bands)	INTEGER	counts

Input Var. Name	Description	Type	Units
prtw	Number of counts measured for each KAV or WG PRT over a specified num of scans, num_scans_prt, bounding the current scan	(num-ptrs * NUM_SCAN_PTR) INTEGER	counts
scan_weights_prt	Weights for scan averaging of PRTs	(num_prts* NUM_SCAN_PRT) REAL	counts
prt_coefficients	PRT coefficients (R0, Alpha, etc)	(NUM_PRT_COEFFS* num_prts) REAL	counts
prt_bp	PRT “counts” for baseplate	INTEGER	counts
prt_coeff_bp	PRT coefficients for baseplate	(NUM_PRT_COEFFS) REAL	counts
coff	Reference counts for PRT computation	INTEGER	counts
cref	PAM counts for PRT computation	INTEGER	counts
rref	PAM resistance for PRT computation	REAL*8	counts
warm_bias	Warm load bias	(NUM_CHANNELS) REAL	°C
cold_bias	Cold load bias	(NUM_CHANNELS) REAL	°C
quadratic_factors	Quadratic BT correction factors	(NUM_CHANNELS) REAL	K
isc	Current scan number	INTEGER	Index-No unit

Table 3.2.2.6-2. Calibration Module Outputs

Output Var. Name	Description	Type	Units
tk	Brightness temperature without any corrections applied	num_channels x num_beam_positions x real*32	K
tkb	Brightness temperature with a scan-position dependent beam-efficiency correction and bias applied	num_channels x num_beam_positions x real*32	K

3.2.2.7 ATMS SDR Geolocation (ProSdrAtms::calcGeolocation())

The ATMS SDR geolocation flow diagram is shown in Figure 3.2.2.7-1.

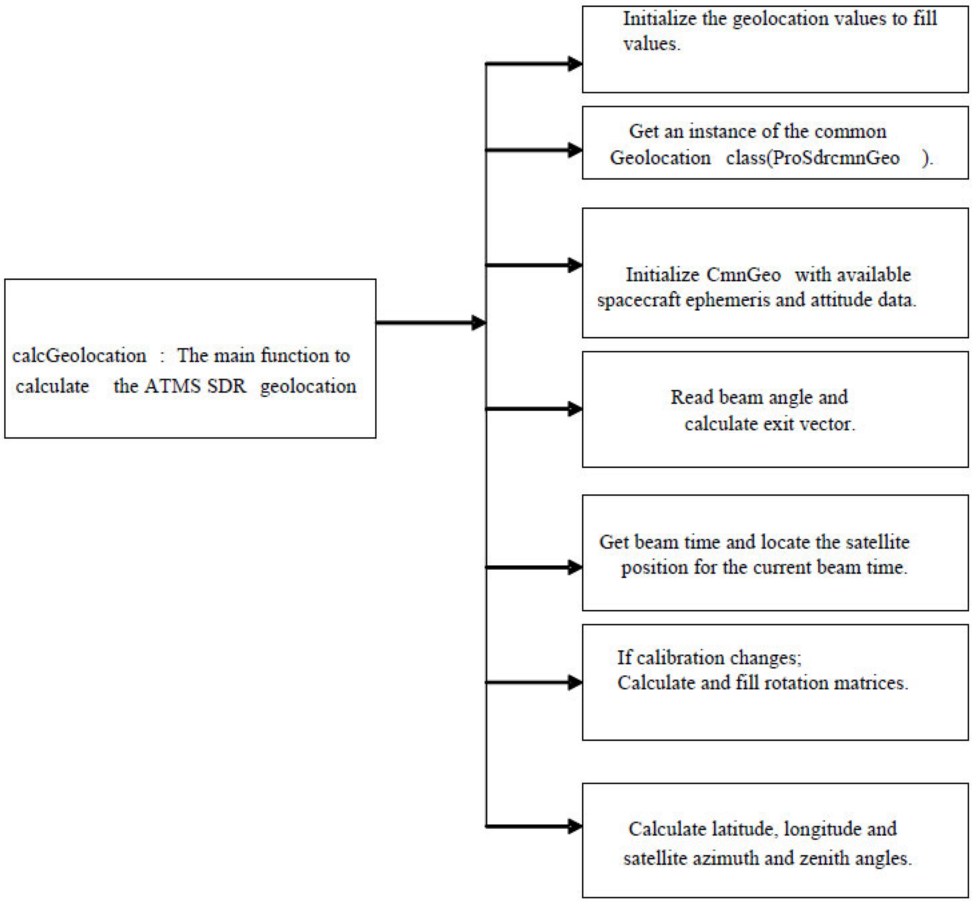

Figure 3.2.2.7-1. ATMS SDR Geolocation Flow Diagram

IDPS designed and developed a library of functions called Common Geolocation that are used by all the SDR algorithms to calculate geodetic latitude and longitude as well as solar and lunar geometries. For more details on the common geolocation library of functions please refer to the Common Geolocation OAD, 474-00448-03-08. ATMS SDR retrieves the Spacecraft Diary RDR Data from DMS, unpacks and byte aligns the spacecraft diary to create the verified spacecraft RDR. ATMS SDR initializes common geolocation structure with satellite attitude and ephemeris data. ATMS SDR gets the beam angle for a given beam position and calculates the exit vector for that beam position. Beam alignment offsets are obtained from RDR calibration packet data for the 5 bands (K, Ka, V, W and G) or from the tunable parameters file for the 22 channels. The option to use beam alignment offsets from either calibration packet data or the tunable parameters file is controlled by a switch in the tunable parameters file. The rotation matrix is calculated using the beam alignment offsets; Common Geolocation functions are then used to apply the rotation matrix to the exit vector (resulting in corrected exit vectors) and calculate the ATMS SDR geodetic latitude, longitude, and other geolocation values. This process is repeated for each earth-view beam position within each scan.

When the band-level beam alignment offsets are used, the corrected exit vectors and geolocation values are calculated at the band-level, where channels belonging to the same band will have identical geolocation. When the channel-level beam alignment offsets are used, the corrected exit vectors and geolocation values are calculated at the channel-level.

Additionally, latitude and longitude for a subset of channels (currently channels 1, 2, 3, 16, 17 - representing the 5 bands) are stored in the GEO output product.

The following subsections provide additional details on ATMS geolocation.

3.2.2.7.1 Calculation of Mid-scan Spacecraft Position, Velocity and Attitude

Spacecraft position, velocity and attitude are calculated for the middle of every scan in the granule. This data is obtained by calling the ProSdrCmnGeo::satPosAtt() function and passing it the mid-scan (beam 47) time of a given scan. When the mid-scan time is missing, an estimated mid-scan time is used. The estimated time is calculated based on an offset from the scan start time.

3.2.2.7.2 Calculation of Geolocation

Geolocation is calculated for each channel/earth-view beam/scan in the granule. The following section outlines the process.

First, a sensor exit vector is calculated for a given earth-view beam position. The beam angle associated with the beam position is used to calculate a sensor exit vector for that beam position. A sensor exit vector is calculated as follows:

X component = 0.0
Y component = sin(beamAngle)
Z component = cos(beamAngle)

Note: The X component is always zero because the scan is in the Y-Z plane.

Next, satellite attitude and ephemeris data is calculated for the given earth-view beam position. This data is obtained by calling the ProSdrCmnGeo::satPosAtt() function and passing it the beam time of the beam position.

A direction cosine matrix for sensor-to-spacecraft rotation (sensor-to-spacecraft rotation matrix) is calculated from the mounting errors in the tunable parameters file. Then a calibration offset rotation matrix is calculated by using the beam alignment offsets for the five bands from verified RDR calibration data or the offsets for the 22 channels from the tunable parameters file. Alignment offsets for each beam position are calculated by interpolating between the 3 beam (1, 48 and 96) offsets - using the first 2 offsets for first half of the scan and last 2 offsets for second half of the scan. Then, the roll, pitch and yaw calibration offsets are translated into a rotation matrix (calibration offset rotation matrix) for the beam position.

The option to use beam alignment offsets from either calibration packet data or the tunable parameters file is controlled by the useBeamAlignTele switch. If the switch is set, the code uses

alignment offsets from calibration data with the closest time to the given scan start time. In this case, channels belonging to the same band will have identical geolocation.
Next, the sensor-to-spacecraft rotation matrix is multiplied times the calibration offset rotation matrix to produce a combined rotation matrix. The combined rotation matrix is passed to the Common Geolocation function ProSdrCmnGeo::ellipIntersect(), along with the sensor exit vector to create the geolocation data.

ProSdrCmnGeo::ellipIntersect() uses the combined rotation matrix to transform the sensor exit vector in sensor coordinates, to the sensor exit vector in spacecraft coordinates. It performs many other calculations to create the final geolocation data (see Common Geolocation OAD, 474-00448-03-08, for details).

NOTE: The steps to account for Sensor-to-Spacecraft Rotation are not yet implemented in the operational code.

Any desired rotation matrix is calculated by the following:

```
roll, pitch, yaw → attitude offsets from the calibration data
cosRoll = cos(roll);
sinRoll = sin(roll);
cosPitch = cos(pitch);
sinPitch = sin(pitch);
cosYaw = cos(yaw);
sinYaw = sin(yaw);

rotation[0][0] = (cosYaw*cosPitch) – (sinYaw*sinRoll*sinPitch)
rotation[0][1] = (sinYaw*cosPitch) + (cosYaw*sinRoll*sinPitch)
rotation[0][2] = (-cosRoll*sinPitch)
rotation[1][0] = (-sinYaw*cosRoll)
rotation[1][1] = (cosYaw*cosRoll)
rotation[1][2] = (sinRoll)
rotation[2][0] = (cosYaw*sinPitch) + (sinYaw*sinRoll*cosPitch)
rotation[2][1] = (sinYaw*sinPitch) – (cosYaw*sinRoll*cosPitch)
rotation[2][2] = (cosRoll*cosPitch)
```

Note: Before the above calculations are performed, the matrices are initialized to be identity matrices, i.e., no rotation.

Finally, the geolocation is obtained for each channel/band of a given beam position by calling the ProSdrCmnGeo::ellipIntersect() function and passing it the previously calculated satellite attitude and ephemeris data, rotation matrices and sensor exit vector for the given channel/band / at the given beam position. This function combines the sensor exit vector with the roll, pitch and yaw of the attitude and ephemeris point. This function calculates the geodetic latitude, longitude, satellite azimuth and zenith angles, satellite range.

The solar azimuth and solar zenith are obtained for each channel/band of the given beam position by calling the ProSdrCmnGeo::sunAngles() function and passing it the beam time, latitude and longitude of the channel/band at the given earth view beam position.

3.2.2.7.3 Lunar Intrusion Detection

The lunar detection algorithm detects the occurrence of lunar contamination of the cold space view. The check is performed for each channel/cold space view/scan in the granule. Additional details can be found in the Lunar Intrusion Handling Tech Memo (NP-EMD-2007.510.0018). The following section outlines the process.

First, a sensor exit vector is calculated for a given cold space beam position. The beam angle of a given space view beam position is used to calculate a sensor exit vector for the given position. A sensor exit vector is calculated as follows:

X component = 0.0
Y component = sin(beamAngle)
Z component = cos(beamAngle)
Note: The X component is always zero because the scan is in the Y-Z plane.

Next, satellite attitude and ephemeris data is calculated for a given cold space view beam position. This data is obtained by calling the ProSdrCmnGeo::satPosAtt() function and passing it the beam time of the beam position.

Next, lunar angles, phase and illumination fraction are calculated for a given cold space view beam position. This data is obtained by calling the ProSdrCmnGeo::moonAngles() function and passing it the beam time of the beam position and the attitude and ephemeris data.

The separation angle between the moon and sun is given by:

$\Theta = \pi$ - moonPhase

The angle between the cold space view and the moon is obtained by calling the ProSdrCmnGeo::moonInView() function and passing it the beam time, the rotation matrix, and the sensor exit vector of the beam position.

The cold space view temperature increase caused by the lunar contamination is given as:

$$\Delta T_c = \exp\left[-\frac{\gamma^2}{2\gamma_s^2}\right] \times \beta \times T_{moon}$$

Where γ : angle between the cold space view and the moon [degrees]

$\gamma_s = \frac{\gamma_{3dB}}{2.35}$ where γ_{3dB} is the 3dB beamwidth [degrees]

$\beta = \frac{1}{2}\left(\frac{r_{moon}}{\gamma_s}\right)^2$ where radius of the moon: r_{moon} = 0.255 deg

Where β : nominal area ration of moon to FOV

T_{moon} : effective moon temperature [K]

$$= 95.21 + 104.63(1 - \cos\theta) + 11.62(1 + \cos 2\theta)$$

θ: separation angle between moon and sun (θ=180° in case of Full Moon)

Finally, the cold space view temperature increase caused by the lunar contamination is calculated. If the lunar intrusion temperature increase (ΔT_c) is larger than the threshold (ΔT_0) for the given channel, then the cold space view count is set to a fill value (NA_INT32_FILL) in the verified RDR and the Moon In View Quality Flag is set for this scan and channel. The ΔT_0 values are implemented as tunable parameters in the Processing Coefficients file. For now, all values are set to 0.2 K.

3.2.3 Graceful Degradation

No graceful degradation is performed.

3.2.4 Exception Handling

ATMS SDR checks for division-by-zero errors and other error conditions. In the c2K routine it checks for PRT counts for each scan, and if the count is zero then the KAV PRT Error and WG PRT Error quality flags are set. Also it checks to make sure the offset counts and PRT counts are not equal, which can introduce division-by-zero error. In that case also the KAV PRT and WG PRT Error flags are set. There are checks for infinite loops to make sure the PRT temperatures are converging.

3.2.5 Data Quality Monitoring

Data quality tests are performed on the ATMS SDR and TDR and each test can produce a Data Quality Notification (DQN). If the thresholds are met, the algorithm stores a DQN to DMS indicating the tests that failed and the number of failures. The DQN criteria is adjustable and contained in a data quality threshold table (DQTT). If the ATMS SDR algorithm cannot obtain the DQTT, the algorithm still executes but no DQN tests are run. Refer to 474-00448-02-01_JPSS-DD-Vol-II-Part-1 for more information on Data Quality Notifications.

Table 3.2.5. Data Quality Notification Criteria

Test Description	Text	Action
Summary ATMS SDR Quality – Checks if the number of good quality retrievals in the granule is less than a configurable amount.	ATMS Summary Quality Test	Send DQN if less than threshold.
Health and Status – Checks if the number of out-of-range Health & Status items in the granule is greater than a configurable amount.	ATMS Health & Status Quality Test	Send DQN if greater than threshold.
Gain Error – Checks if the number of gain errors in the granule is greater than a configurable amount.	ATMS Gain Error Quality Test	Send DQN if greater than threshold.

3.2.5.1 Quality Flags

474-00448-02-02_JPSS-DD-Vol-II-Part-02, Section 6.2.2 contains information about the quality flags set in the ATMS SDR outputs.

3.2.6 Computational Precision Requirements

The code uses double precision real variables whenever necessary for computational accuracy.

3.2.7 Algorithm Support Considerations

INF and DMS must be running before execution of the algorithm.

3.2.8 Assumptions and Limitations

None.

4 GLOSSARY/ACRONYM LIST

4.1 Glossary

Below is a glossary of terms most applicable for this OAD.

Term	Description
Algorithm	A formula or set of steps for solving a particular problem. Algorithms can be expressed in any language, from natural languages like English to mathematical expressions to programming languages like FORTRAN. On JPSS, an algorithm consists of: 1. A theoretical description (i.e., science/mathematical basis) 2. A computer implementation description (i.e., method of solution) 3. A computer implementation (i.e., code)
Algorithm Engineering Review Board (AERB)	Interdisciplinary board of scientific and engineering personnel responsible for the approval and disposition of algorithm acceptance, verification, development and testing transitions. Chaired by the Data Process Algorithm Lead, members include representatives from STAR, DPMS, IDPS, and Raytheon.
Algorithm Verification	Science-grade software delivered by an algorithm provider is verified for compliance with data quality and timeliness requirements by Algorithm Team science personnel. This activity is nominally performed at the GRAVITE facility. Delivered code is executed on compatible GRAVITE computing platforms. Minor hosting modifications may be made to allow code execution. Optionally, verification may be performed at the Algorithm Provider's facility if warranted due to technical, schedule or cost considerations.
Ancillary Data	Any data which is not produced by the JPSS System, but which is acquired from external providers and used by the JPSS system in the production of JPSS data products.
Auxiliary Data	Auxiliary Data is defined as data, other than data included in the sensor application packets, which is produced internally by the JPSS system, and used to produce the NPOESS deliverable data products.
EDR Algorithm	Scientific description and corresponding software and test data necessary to produce one or more environmental data records. The scientific computational basis for the production of each data record is described in an ATBD. At a minimum, implemented software is science-grade and includes test data demonstrating data quality compliance.
Environmental Data Record (EDR)	*[IORD Definition]* Data record produced when an algorithm is used to convert Raw Data Records (RDRs) to geophysical parameters (including ancillary parameters, e.g., cloud clear radiation, etc.). *[Supplementary Definition]* An Environmental Data Record (EDR) represents the state of the environment, and the related information needed to access and understand the record. Specifically, it is a set of related data items that describe one or more related estimated environmental parameters over a limited time-space range. The parameters are located by time and Earth coordinates. EDRs may have been resampled if they are created from multiple data sources with different sampling patterns. An EDR is created from one or more JPSS SDRs or EDRs, plus ancillary environmental data provided by others. EDR metadata contains references to its processing history, spatial and temporal coverage, and quality.
Operational Code	Verified science-grade software, delivered by an algorithm provider and verified by GRAVITE , is developed into operational-grade code by the IDPS IPT.
Operational-Grade Software	Code that produces data records compliant with the System Specification requirements for data quality and IDPS timeliness and operational infrastructure. The software is modular relative to the IDPS infrastructure and compliant with IDPS application programming interfaces (APIs) as specified for TDR/SDR or EDR code.

Term	Description
Raw Data Record (RDR)	*[IORD Definition]* Full resolution digital sensor data, time referenced and earth located, with absolute radiometric and geometric calibration coefficients appended, but not applied, to the data. Aggregates (sums or weighted averages) of detector samples are considered to be full resolution data if the aggregation is normally performed to meet resolution and other requirements. Sensor data shall be unprocessed with the following exceptions: time delay and integration (TDI), detector array non-uniformity correction (i.e., offset and responsivity equalization), and data compression are allowed. Lossy data compression is allowed only if the total measurement error is dominated by error sources other than the data compression algorithm. All calibration data will be retained and communicated to the ground without lossy compression. *[Supplementary Definition]* A Raw Data Record (RDR) is a logical grouping of raw data output by a sensor, and related information needed to process the record into an SDR or TDR. Specifically, it is a set of unmodified raw data (mission and housekeeping) produced by a sensor suite, one sensor, or a reasonable subset of a sensor (e.g., channel or channel group), over a specified, limited time range. Along with the sensor data, the RDR includes auxiliary data from other portions of JPSS (space or ground) needed to recreate the sensor measurement, to correct the measurement for known distortions, and to locate the measurement in time and space, through subsequent processing. Metadata is associated with the sensor and auxiliary data to permit its effective use.
Retrieval Algorithm	A science-based algorithm used to 'retrieve' a set of environmental/geophysical parameters (EDR) from calibrated and geolocated sensor data (SDR). Synonym for EDR processing.
Science Algorithm	The theoretical description and a corresponding software implementation needed to produce an NPP/JPSS data product (TDR, SDR or EDR). The former is described in an ATBD. The latter is typically developed for a research setting and characterized as "science-grade".
Science Algorithm Provider	Organization responsible for development and/or delivery of TDR/SDR or EDR algorithms associated with a given sensor.
Science-Grade Software	Code that produces data records in accordance with the science algorithm data quality requirements. This code, typically, has no software requirements for implementation language, targeted operating system, modularity, input and output data format or any other design discipline or assumed infrastructure.
SDR/TDR Algorithm	Scientific description and corresponding software and test data necessary to produce a Temperature Data Record and/or Sensor Data Record given a sensor's Raw Data Record. The scientific computational basis for the production of each data record is described in an Algorithm Theoretical Basis Document (ATBD). At a minimum, implemented software is science-grade and includes test data demonstrating data quality compliance.
Sensor Data Record (SDR)	*[IORD Definition]* Data record produced when an algorithm is used to convert Raw Data Records (RDRs) to calibrated brightness temperatures with associated ephemeris data. The existence of the SDRs provides reversible data tracking back from the EDRs to the Raw data. *[Supplementary Definition]* A Sensor Data Record (SDR) is the recreated input to a sensor, and the related information needed to access and understand the record. Specifically, it is a set of incident flux estimates made by a sensor, over a limited time interval, with annotations that permit its effective use. The environmental flux estimates at the sensor aperture are corrected for sensor effects. The estimates are reported in physically meaningful units, usually in terms of an angular or spatial and temporal distribution at the sensor location, as a function of spectrum, polarization, or delay, and always at full resolution. When meaningful, the flux is also associated with the point on the Earth geoid from which it apparently originated. Also, when meaningful, the sensor flux is converted to an equivalent top-of-atmosphere (TOA) brightness. The associated metadata includes a record of the processing and sources from which the SDR was created, and other information needed to understand the data.

Term	Description
Temperature Data Record (TDR)	*[IORD Definition]* Temperature Data Records (TDRs) are geolocated, antenna temperatures with all relevant calibration data counts and ephemeris data to revert from T-sub-a into counts. *[Supplementary Definition]* A Temperature Data Record (TDR) is the brightness temperature value measured by a microwave sensor, and the related information needed to access and understand the record. Specifically, it is a set of the corrected radiometric measurements made by an imaging microwave sensor, over a limited time range, with annotation that permits its effective use. A TDR is a partially-processed variant of an SDR. Instead of reporting the estimated microwave flux from a specified direction, it reports the observed antenna brightness temperature in that direction.
Model Validation	The process of determining the degree to which a model is an accurate representation of the real-world from the perspective of the intended uses of the model.
Model Verification	The process of determining that a model implementation accurately represents the developer's conceptual description and specifications.

4.2 Acronyms

Below is a list of acronyms most applicable for this OAD.

Acronym	Description
ACO	Atmospheric Correction over Ocean
ADCS	Advanced Data Collection System
ADS	Archive and Distribution Segment
AFB	Air Force Base
AFM	Airborne Fluxes and Meteorology Group
AFSCN	Air Force Satellite Control Network
AFWA	Air Force Weather Agency
AFWWS	Air Force Weather Weapon System
AGE	Aerospace Ground Equipment
AIAA	American Institute of Aeronautics and Astronautics
ANSI	American National Standards Institute
Ao	Operational Availability
AOS	Acquisition of Signal
ATMS	Advanced Technology Microwave Sounder
BIT	Built-in Test
BITE	Built-in Test Equipment
BMMC	Backup Mission Management Center
C2	Command and Control
C3S	Command, Control, and Communications Segment
CCSDS	Consultative Committee for Space Data Systems
CDA	Command and Data Acquisition
CDDIS	Crustal Dynamics Data Information System
CDR	Climate Data Records
CERES	Cloud and Earth Radiant Energy System
CGMS	Coordination Group for Meteorological Satellites
CI	Configured Item
CLASS	Comprehensive Large-Array data Stewardship System
CMIS	Conical Microwave Imager Sounder
CMN GEO	Common Geolocation

Acronym	Description
CMOC	Cheyenne Mountain Operations Center
COMSAT	Communications Satellite
COMSEC	Communications Security
CONUS	Continental United States
COTS	Commercial Off the Shelf
CrIS	Cross-Track Infrared Sounder
CSCI	Computer Software Configured Item
DCP	Data Collection Platforms
DES	Digital Encryption System
DFCB	Data Format Control Book
DHN	Data Handling Node
DMSP	Defense Meteorological Satellite Program
DOC	Department of Commerce
DoD	Department of Defense
DRR	Data Routing and Retrieval
EDR	Environmental Data Records
EELV	Evolved Expendable Launch Vehicle
EMC	Electromagnetic Compatibility
EMD	Engineering and Manufacturing Development
EOL	End of Life
EOS	Earth Observing System
ERBS	Earth Radiation Budget Suite
ESD	Electrostatic Discharge
EUMETSAT	European Organization for the Exploitation of Meteorological Satellites
EWR	Eastern and Western Ranges
FFRDC	Federally Funded Research and Development Center
FMH	Federal Meteorological Handbook
FNMOC	Fleet Numerical Meteorology and Oceanography Center
FOC	Full Operational Capability
FTS	Field Terminal Segment
FVS	Flight Vehicle Simulator
GFE	Government Furnished Equipment
GIID	General Instrument Interface Document
GN	NASA Ground Network
GPS	Global Positioning System
GPSOS	GPS Occultation Suite
GSE	Ground Support Equipment
GSFC	Goddard Space Flight Center
HIJACK	Data Conversion Software
HRD	High Rate Data
IAW	In Accordance With
ICD	Interface Control Document
IDPS	Interface Data Processor Segment
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IGS	International GPS Service
IJPS	Initial Joint Polar System
ILS	Integrated Logistics Support
IOC	Initial Operational Capability

Acronym	Description
IORD	Integrated Operational Requirements Document
IOT&E	Initial Operational Tests & Evaluation
IP	Intermediate Product
IPL	Integrated Priority List
IPO	Integrated Program Office
IRD	Interface Requirements Document
ISO	International Standards Organization
ITRF	International Terrestrial Reference Frame
ITU	International Telecommunications Union
JPS	Joint Polar System
JSC	Johnson Space Center
JTA	Joint Technical Architecture
km	kilometer
LEO&A	Launch, Early Orbit, & Anomaly Resolution
LOS	Loss of Signal
LRD	Low Rate Data
LSS	Launch Support Segment
LST	Local Solar Time
LUT	Look-Up Table or Local User Terminal
LV	Launch Vehicle
MDFCB	Mission Data Format Control Book
MDT	Mean Down Time
Metop	Meteorological Operational Program
MMC	Mission Management Center
MOU	Memorandum of Understanding
MSS	Mission System Simulator
MTBCF	Mean Time Between Critical Failures
MTBDE	Mean Time Between Downing Events
MTTRF	Mean Time to Restore Function
NA	Non-Applicable
NACSEM	NPOESS Acquisition Cost Estimating Model
NASA	National Aeronautics and Space Administration
NAVOCEANO	Naval Oceanographic Office
NCA	National Command Authority
NCEP	National Centers for Environmental Prediction
NDT	Nitrate-Depletion Temperature
NESDIS	National Environmental Satellite, Data, and Information Service
NOAA	National Oceanic and Atmospheric Administration
NORAD	North American Aerospace Defense Command
NPOESS	National Polar-orbiting Operational Environmental Satellite System
NPP	NPOESS Preparatory Program
NSA	National Security Agency
NTIA	National Telecommunications Information Agency
OC/C	Ocean Color/Chlorophyll
O&M	Operations and Maintenance
OMPS	Ozone Mapping and Profiling Suite
P3I	Potential Pre-planned Product Improvements
PAM	Precision Analog Monitor
PHS&T	Packaging, Handling, Storage, and Transportation

Acronym	Description
PIP	Program Implementation Plan
PM&P	Parts, Materials, and Processes
PMT	Portable Mission Terminal
POD	Precise Orbit Determination
POES	Polar Orbiting Environmental Satellite
PRT	Platinum Resistance Temperature
RDR	Raw Data Records
RPIE	Real Property Installed Equipment
RSR	Remote Sensing Reflectance
S&R	Search and Rescue
SARSAT	Search and Rescue Satellite Aided Tracking
SCA	Satellite Control Authority
SDC	Surface Data Collection
SDE	Selective Data Encryption
SDP	Software Development Plan
SDR	Sensor Data Records
SDS	Science Data Segment
SESS	Space Environmental Sensor Suite
SGI®	Silicon Graphics, Inc.
SI	International System of Units
SMD	Stored Mission Data
SN	NASA Space Network
SOC	Satellite Operations Center
SRD	Sensor Requirements Documents
SS	Space Segment
SST	Sea Surface Temperature
STDN	Spaceflight Tracking and Data Network
SVE	Space Vehicle Equipment
TBD	To Be Determined
TBR	To Be Resolved
TBS	To Be Supplied
TDR	Temperature Data Records
TDRSS	Tracking and Data Relay Satellite System
TEMPEST	Telecommunications Electronics Material Protected from Emanating Spurious Transmissions
TOA	Top of the Atmosphere
TRD	Technical Requirements Document
TSIS	Total Solar Irradiance Sensor
USAF	United States Air Force
USB	Unified S-band
USG	United States Government
UTC	Universal Time Coordinated
VIIRS	Visible/Infrared Imager Radiometer Suite