Direction des Systèmes Orbitaux
Sous-Direction Etudes Systèmes et Développements
Division Altimétrie et Localisation Précise
Département Missions Systèmes
18, avenue Edouard Belin
31401 TOULOUSE CEDEX 4

Toulouse, January 15, 1999

Pasadena, March 22, 2001

SSALTO

CCI JMR LEVEL 1.0 PROCESSING

Preparation coordinated by:	J. STUM and J.M. ZIGNA (CLS) S. DESAI, A. KITIYAKARA, and T. PAVLOVITCH (JPL)	
Accepted by:	P. VINCENT, CNES J. NOUBEL, CNES J. PERBOS, CNES	
Approved by:	S. COUTIN-FAYE, CNES B. HAINES, JPL	

Document ref:	SMM-ST-M1-EA-11577-CN		Issue: 3	Update: 0

For application	DS2	DS4	DS5	DH2	DPP	DJ1	TP	TP2	F-PAC	DOR	AVI	SOL
								X				

Configuration controlled Document	YES	by: CCM SSALTO	Since: Nov. 13th, 98

TYPING FACILITIES

Original typing	Storage	Softwares
EUROSAV	t:\docenlign\smm\m1\11577000.doc	Office 97
JPL		Office 98

DOCUMENT STATUS

Page	i	ii	iii		1	2	3	4	5	6	7	8	9	10	11	12	13	14
Issue	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0
Update	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0

Page	15	16	17	18	19	20	21	22	23	24	25	26	27		A1.1	A1.2	A1.3
Issue	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
Update	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0

Page	A1.4	A1.5	A1.6	A1.7	A1.8	A1.9	A1.10	A1.11	A1.12		A2.1					
Issue	0	0	0	0	0	0	0	0	0		0					
Update	0	0	0	0	0	0	0	0	0		0					

DOCUMENT CHANGE RECORD

Issue	Update	Date	Modifications	Visa
0	0	30/07/98	Document creation	
1	0	15/01/99	Adding the JMR mode 1 processing, transferring the antenna temperature calculation from level 1b, and removing some TBD/TBC present in the previous issue.	
2	0	18/06/99	Algorithm specifications added by JPL	
2	1	20/10/99	Algorithm specification updated by JPL using comments from CNES	
2	2	13/12/99	Algorithms updated to allow for case when both 23.8 GHz channels are active.	
2	3	20/01/00	Added references for user product document and data management and control algorithms.	
2	4	17/04/00	Changed JMR_NRMCNT_01 to account for overflowed reference counts when altimeter blanking not present.	
3	0	22/03/01	Use default value for latitude and longitude when Everett interpolation fails, but continue processing of JMR data.	

			Add algorithm JMR_PKCHK_01. Update packet definition to allow for CNT to have maximum value of 50150 instead of 50000. If CNT out of range use value of 0 when computing time.	

ABBREVIATIONS

Sigle	Definition
ADx	Applicable Document x
CCI	Centre de Contrôle des Instruments
CLS	Collecte Localisation Satellites
CNES	Centre National d'Etudes Spatiales
DAD	Dynamic Auxiliary Data
GPS	Global Positioning System
JPL	Jet Propulsion Laboratory
JTCCS	JASON Telemetry Command and Control System
PGGS	PROTEUS Ground Segment
PLTM	PayLoad TeleMetry
RDx	Reference Document x
SAD	Static Auxiliary Data
SSALTO	Segment Sol Altimétrie et Orbitographie
TAI	Temps Atomique International
TBC	To Be Confirmed
TBD	To Be Defined
TM	TeleMetry
UTC	Universal Time Coordinated

APPLICABLE AND REFERENCE DOCUMENTS

Reference	Document title
TP2-SB-J0-102-CNES	AD1 JASON-1 Science and Operational Requirements
SMM-SP-M-EA-10600-CN	AD2 Spécifications techniques de besoins du segment sol multimissions SSALTO
TP2-JS-IF-600-CNES	AD3 JASON-1 Ground system interfaces
PRO-LS-DC-10090-CNES	AD4 PGGS Internal and External Interfaces Specifications
SMM-IF-M-EA-20054-CN	AD5 Manuel d'Interface SSALTO
SMM-ST-M2-EA-11010-CN	AD6 Algorithm Definition, Accuracy and Specification: Volume 9: CMA mechanism
TP2-UG-PL-JMR-3201-JPL	AD7 JPL Jason JMR Users Guide
TP2-SS-JGS-JMRL1-5207-JPL	RD1 JMR Level 1.0 Software Specification: Data Management and Control Algorithms

Reference	Document title
SMM-ST-M-EA-12081-CN	RD2 SSALTO Products Specification: JMR Level 1.0 Data Product
TP2-LB-O-SP-0107-ASC (JPL D-17103)	RD3 Jason-1 Project: Payload Instrument General Design and Interfaces Specification (GDIS)

TBC AND TBD LIST

TBC/TBD	Paragraph	Brief description

CONTENTS

1. INTRODUCTION

The purpose of this document is to provide definitions and specifications of the algorithms that are used in the nominal level 1.0 processing of the JASON-1 microwave radiometer (JMR) data. The JMR level 1.0 data processing is performed within the CCI processor. The scientific and operational requirements placed by the JASON Science Working Team on the JASON project are provided in AD1, and the requirements that define the CCI facility within the SSALTO system are provided in AD2.

Definition and Specification of the JMR level 1.0 algorithms

The definition of the JMR level 1.0 algorithms identifies the main functions and provides a broad overview of the algorithms in the context of a general mathematical statement, while the specification of the algorithms provides the detailed information necessary for software development and implementation of the respective algorithm definition.

The JMR level 1.0 algorithms are considered to lie within one of the following two categories.

- Those algorithms that specifically process the data in the JMR PLTM packets into the JMR level 1.0 product.
- “Data management and control” algorithms, which perform operations such as:
- Extracting the input data
- Preparing the data to be processed
- Performing general checks such as ensuring the presence of input files, the data conformity, or the compatibility of input data with the data set to be processed.
- Building the output product(s)
- Managing the processing

Only those algorithms that specifically process data from the JMR PLTM packets into the JMR level 1.0 product are defined and specified in this document. Data management and control algorithms are described in RD1.

Organization of the document

The JMR level 1.0 algorithms are separated into the following two subsets of algorithms:

- The telemetry algorithms, which extract raw JMR measurements and engineering data from JMR PLTM packets.
- The JMR science data algorithms, which generate the JMR level 1.0 product from the raw JMR measurements.

In this document, the algorithms required to generate a set of JMR level 1.0 parameters from JMR PLTM packets are presented as a linear set of functions. This representation is chosen only to ease the understanding of the data processing procedure, but it does not anticipate the organization or the sequencing of the algorithms within the CCI processor. The interfaces of the algorithms, namely the input and output data, are defined in section 2, the telemetry extraction algorithms are described in section 3, and the science data algorithms are described in section 4.

The definition and specification of each algorithm are separated as follows:

- Name and identifier of the algorithm
- Function
- Algorithm definition:
  - Input data
  - Output data
  - Mathematical statement
- Algorithm specification:
  - Input data
  - Output data
  - Processing
- Accuracy (if any)
- Comments (if any)
- References (if any)

Basic Rules

The following basic rules are applied to the algorithm definitions and specifications.

- The specifications of an algorithm are always relevant to the processing of one data point and not to one set of data points.
- Elementary functions, which are common to several algorithms (also called “mechanisms”), are specified in AD6.
- The input and output data are always identified by a precise description, an explicit name (which could be used in the coding phase), a unit, and if necessary a reference system.

- Each algorithm systematically outputs an execution status to indicate if any errors may have occurred during the execution of the algorithm. Execution status flags may be used interchangeably with their associated integer values in the discussion:
  - Execution Status "Good", has integer value 0
  - Execution Status "Bad" has integer value 1
- Flags that are generated by any algorithm are initially set to an "invalid/bad" status, and are only set to a "valid/good" status when values are successfully computed and considered to be reasonable. Data flags will adopt the following conventions:
  - Data flag "valid" or "good", has integer value 0
  - Data flag "invalid" or "bad", has integer value 1
- Bit fields always number the bits from the least significant bit to the most significant bit (right to left).
  - The least significant bit at location 0, and the most significant bit at location 7, for a one byte bitfield.
  - The least significant bit at location 0, and the most significant bit at location 15, for a two byte bitfield.
  - The least significant bit at location 0, and the most significant bit at location 31, for a four byte bitfield.
- The following conventions are used to represent tables:
  - $X[N_1:N_2]$ represents a one-dimensional table whose elements are X(i) (or X_i), with $i \in [N_1:N_2]$.
  - $X[N_1:N_2][M_1:M_2]$ represents a two-dimensional table whose elements are X(i,j) (or X_{ij}) with $i \in [N_1:N_2]$ and $j \in [M_1:M_2]$.
  - $X[N_1:N_2][M_1:M_2][L_1:L_2]$ represents a three-dimensional table whose elements are X(i,j,k) (or X_{ijk}) with $i \in [N_1:N_2]$, $j \in [M_1:M_2]$, and $k \in [L_1:L_2]$.

Conventions

The JMR has four operating channels:

- Channel 1: 18.7H with a frequency of 18.7 GHz
- Channel 2: 23.8V with a frequency of 23.8 GHz (redundant)
- Channel 3: 23.8H with a frequency of 23.8 GHz (nominal)
- Channel 4: 34.0H with a frequency of 34.0 GHz

where H denotes horizontal polarization and V denotes vertical polarization. Channel 2 (23.8V) is the redundant 23.8 GHz channel, and channel 3 (23.8H) is the nominal 23.8 GHz channel. The nominal configuration of the JMR instrument has only one of channels 2 or 3 in operation at any given time. However, it is possible for the JMR instrument to have both 23.8 GHz channels active at the same time. Presently, it is thought that both channels would be commanded to be active simultaneously ONLY for diagnostic and calibration evaluation purposes, and this should be a rare occurrence if at all. Note that all of the algorithms in this document allow for this unusual configuration of the JMR instrument by performing all computations on both 23.8 GHz channels if both appear to be active. However, in the nominal configuration where only one 23.8 GHz channel is active at any given time, computations are only performed on the active 23.8 GHz channel. Note that when the JMR Level 1.0 product provides a valid antenna temperature measurement from both 23.8 GHz channels, subsequent users of the antenna temperature measurements that expect a single antenna temperature measurement at the 23.8 GHz frequency should ignore the antenna temperature from Channel 2, and should use the antenna temperature from Channel 3 as the nominal antenna temperature measured at the 23.8 GHz frequency. Unless specifically noted, this document will always list channel dependent parameters in the sequence indicated above, and frequency-dependent parameters in order of increasing frequency: i.e., 18.7 GHz, 23.8 GHz, and 34.0 GHz.

2. INPUT AND OUTPUT DATA

2.1 INPUT DATA

Two types of input data are distinguished:

- Product data, which correspond to measurements performed by the JMR.
- Auxiliary data, which correspond to instrument characterization data and processing parameters.

2.1.1 PRODUCT DATA

For the JMR level 1.0 processing the product data corresponds to the data on the JMR PLTM packets. The detailed format of the JMR PLTM packets is provided in section 3.

2.1.2 AUXILIARY DATA

Auxiliary data are considered to be either dynamic or static, where dynamic auxiliary data are time dependent, and static data are time independent.

- **Dynamic auxiliary data:**

Dynamic auxiliary data required for the JMR level 1.0 algorithms consist of:

  – JASON-1 orbit data (precise, DORIS preliminary or DORIS navigator orbit).

- **Static auxiliary data:**

For JMR level 1.0 algorithms static auxiliary data consist of:

  – Processing parameters (all of the global constants used in the algorithms, including Universal constants)

  – JMR instrument characterization data

Table 1 defines the format of the ASCII file that provides the JMR static auxiliary data. This file provides: 1) global constants including Universal constants; 2) coefficients that are required for the conversion of raw thermistor counts into temperatures in Kelvin; 3) calibration constants required to correct the noise diode temperatures; 4) calibration coefficients that are required by the antenna temperature retrieval algorithm; 5) other instrument characterization data required by the JMR level 1.0 algorithms.

The static auxiliary data file has a header section of arbitrary size, followed by the data section, which is of fixed size. The header section contains lines on which the first character is an asterisk. These lines are ignored by the software that reads the file. Any number of header lines may be included in the file, as long as they all precede the data lines. Information about the version number and date of creation of any particular JMR level 1.0 static auxiliary data file will be found in the header lines.

Table 1. JMR Level 1.0 Static Auxiliary Data File

```
* JJ1 CHD AXVJPL vvvvmmdd hhmmss vvvvmmdd hhmmss vvvvmmdd hhmmss
* JMR Level 1.0 Static Auxiliary Data
* mm/dd/yyyy Version 1.0
* The thermistor data are always in mux address order.
dtpkgap = DTPKGAP
cntfre = CNTFRE
semi_major_axis = SM_Axis
earth_flattening = Flattening
dt_temp = DT_TEMP
dt_cal1 = DT_CAL1
defcnt = DEFCNT
min_tolerance_counts = MINTOLCNT
waveguide4_mode1_antenna_temps = WGNUM
radiometer_count_renorm_knorm = K_NORM
noise_source_thermistor = NSRCNUM
thermistor_calib_resist_rlo1 = RLO(1)
thermistor_calib_resist_rlo2 = RLO(2)
thermistor_calib_resist_rhi1 = RHI(1)
thermistor_calib_resist_rhi2 = RHI(2)
thermistor_ref1_polyn_coeffs   = A(1), B(1), C(1), D(1)
thermistor_ref2_polyn_coeffs   = A(2), B(2), C(2), D(2)
thermistor_ref3_polyn_coeffs   = A(3), B(3), C(3), D(3)
thermistor_ref4_polyn_coeffs   = A(4), B(4), C(4), D(4)
thermistor_nsrc1_polyn_coeffs = A(5), B(5), C(5), D(5)
thermistor_nsrc2_polyn_coeffs = A(6), B(6), C(6), D(6)
thermistor_fh1_polyn_coeffs    = A(7), B(7), C(7), D(7)
thermistor_fh2_polyn_coeffs    = A(8), B(8), C(8), D(8)
thermistor_wg11_polyn_coeffs = A(9), B(9), C(9), D(9)
```

thermistor_wg12_polyn_coeffs = A(10), B(10), C(10), D(10)

thermistor_wg21_polyn_coeffs = A(11), B(11), C(11), D(11)

thermistor_wg22_polyn_coeffs = A(12), B(12), C(12), D(12)

thermistor_wg31_polyn_coeffs = A(13), B(13), C(13), D(13)

thermistor_wg32_polyn_coeffs = A(14), B(14), C(14), D(14)

thermistor_wg41_polyn_coeffs = A(15), B(15), C(15), D(15)

thermistor_wg42_polyn_coeffs = A(16), B(16), C(16), D(16)

thermistor_ref1_temp_min_thres = Txmin(1)

thermistor_ref2_temp_min_thres = Txmin(2)

thermistor_ref3_temp_min_thres = Txmin(3)

thermistor_ref4_temp_min_thres = Txmin(4)

thermistor_nsrc1_temp_min_thres = Txmin(5)

thermistor_nsrc2_temp_min_thres = Txmin(6)

thermistor_fh1_temp_min_thres = Txmin(7)

thermistor_fh2_temp_min_thres = Txmin(8)

thermistor_wg11_temp_min_thres = Txmin(9)

thermistor_wg12_temp_min_thres = Txmin(10)

thermistor_wg21_temp_min_thres = Txmin(11)

thermistor_wg22_temp_min_thres = Txmin(12)

thermistor_wg31_temp_min_thres = Txmin(13)

thermistor_wg32_temp_min_thres = Txmin(14)

thermistor_wg41_temp_min_thres = Txmin(15)

thermistor_wg42_temp_min_thres = Txmin(16)

thermistor_ref1_temp_max_thres = Txmax(1)

thermistor_ref2_temp_max_thres = Txmax(2)

thermistor_ref3_temp_max_thres = Txmax(3)

thermistor_ref4_temp_max_thres = Txmax(4)

thermistor_nsrc1_temp_max_thres = Txmax(5)

thermistor_nsrc2_temp_max_thres = Txmax(6) thermistor_fh1_temp_max_thres = Txmax(7)

thermistor_fh2_temp_max_thres = Txmax(8)

thermistor_wg11_temp_max_thres = Txmax(9)

thermistor_wg12_temp_max_thres = Txmax(10)

thermistor_wg21_temp_max_thres = Txmax(11)

thermistor_wg22_temp_max_thres = Txmax(12)

thermistor_wg31_temp_max_thres = Txmax(13)

thermistor_wg32_temp_max_thres = Txmax(14)

thermistor_wg41_temp_max_thres = Txmax(15)

thermistor_wg42_temp_max_thres = Txmax(16)

ch1_noise_diode1_temp_cor_coefs = KTN(0,1,1), KTN(1,1,1), KTN(2,1,1), KTN(3,1,1)

ch1_noise_diode2_temp_cor_coefs = KTN(0,1,2), KTN(1,1,2), KTN(2,1,2), KTN(3,1,2)

ch1_noise_diode3_temp_cor_coefs = KTN(0,1,3), KTN(1,1,3), KTN(2,1,3), KTN(3,1,3)

ch2_noise_diode1_temp_cor_coefs = KTN(0,2,1), KTN(1,2,1), KTN(2,2,1), KTN(3,2,1)

ch2_noise_diode2_temp_cor_coefs = KTN(0,2,2), KTN(1,2,2), KTN(2,2,2), KTN(3,2,2)

ch2_noise_diode3_temp_cor_coefs = KTN(0,2,3), KTN(1,2,3), KTN(2,2,3), KTN(3,2,3)

ch3_noise_diode1_temp_cor_coefs = KTN(0,3,1), KTN(1,3,1), KTN(2,3,1), KTN(3,3,1)

ch3_noise_diode2_temp_cor_coefs = KTN(0,3,2), KTN(1,3,2), KTN(2,3,2), KTN(3,3,2)

ch3_noise_diode3_temp_cor_coefs = KTN(0,3,3), KTN(1,3,3), KTN(2,3,3), KTN(3,3,3)

ch4_noise_diode1_temp_cor_coefs = KTN(0,4,1), KTN(1,4,1), KTN(2,4,1), KTN(3,4,1)

ch4_noise_diode2_temp_cor_coefs = KTN(0,4,2), KTN(1,4,2), KTN(2,4,2), KTN(3,4,2)

ch4_noise_diode3_temp_cor_coefs = KTN(0,4,3), KTN(1,4,3), KTN(2,4,3), KTN(3,4,3)

feedhorn_calib_sensor1_kf = KF1(1), KF1(2), KF1(3), KF1(4)

feedhorn_calib_sensor2_kf = KF2(1), KF2(2), KF2(3), KF2(3)

path_loss_coefficients = L(1), L(2), L(3), L(4)

ref_load_calib_coeff_kr = KR(1), KR(2), KR(3), KR(4)

waveguide_calib_sensor1_kw = KW1(1), KW1(2), KW1(3), KW1(4)

waveguide_calib_sensor2_kw = KW2(1), KW2(2), KW2(3), KW2(4)

Note that the name of this file follows SSALTO conventions and is named as follows: JJ1_CHD_AXVJPL_yyyymmdd_hhmmss_yyyymmdd_hhmmss_yyyymmdd_hhmmss
The three sets of yyyymmdd_hhmmss strings in the name of the file are date strings that provide dates in year (yyyy), month (mm), day (dd), hour (hh), minute (mm), second (ss) format, with the first date string indicating the date that the file was created, the second date string indicating the start date that the file is valid, and the third date string indicating the end date that the file is valid. If the end date is unknown, or if the file is assumed to be a static file with no end date, then the third date string is set to 00000000_000000. Usually the name of the file will also be provided as the first line of the header section.

The data section contains lines in a “keyword = value” format. If the keyword on a line refers to an array, then all the values for that array follow on the same line, and each value (except the last one on each line) is delimited by a comma. The thermistor data are always in mux address order.

The keywords that appear in the data file are the actual variable names that are present in the JMR Level 1.0 software. In **Table 1** the values are substituted by the variable names that are used in the algorithm definition and specifications described in this document. These variable names are used below to provide a description of each parameter in the JMR level 1.0 static auxiliary data file and the unit of the parameter. Parameters that are dimensionless are indicated to have the unit (/).

- Maximum duration allowed between successive packets. DTPKGAP nominally has a value of 10 seconds to accommodate any slight variations in the packet interval while still being able to detect gaps between packets.

  – DTPKGAP (sec)

- Frequency of signal that is used to measure time differential between JMR measurements and JMR time stamps. This signal has a nominal frequency of 50000 Hz.

  – CNTFRE (Hz)

- Semi major axis of the reference ellipsoid. Presently the value of this parameter is 6378136.3 m.

  – SM_Axis (m)

- Flattening coefficient of the reference ellipsoid. Presently the value of this parameter is 1/298.257.

  – Flattening (/)

- Absolute value of maximum time difference between once-per-second radiometer count measurements, and thermistor data that can be used to compute antenna temperatures. DT_TEMP has a nominal value of 32 seconds.

  – DT_TEMP (sec)

- Absolute value of time difference between once-per-second radiometer count measurements and calibration sequence data that can be used to compute antenna temperatures.

  – DT_CAL1 (sec)

- Default value of radiometer counts. Nominally, the JMR instrument defines the default count to have a value of zero.
  - DEFCNT (/)
- Minimum tolerance on counts to be used for equations with denominators that are in counts. This value should be slightly larger than the peak to peak noise in counts.
  - MINTOLCNT (/)
- Waveguide number, WGNUM, to be used in the antenna temperature computation for mode 1 operation. Note that WGNUM can only have a value of 1 or 2 to indicate that waveguide 1 or 2, respectively, is to be used.
  - WGNUM (/)
- Radiometer count renormalization constant, K_NORM.
  - K_NORM (/)
- Noise source number, NSRCNUM, to be used to correct the noise diode temperatures. NSRCNUM takes on either the value 1 if NSRC1 is to be used, or the value 2 if NSRC2 is to be used.
  - NSRCNUM (/)
- Low and high thermistor calibration resistances, RLO(i) and RHI(i) respectively, for mux address i ($i \in [1:2]$)
  - RLO[1:2] (ohms)
  - RHI[1:2] (ohms)
- Calibration coefficients to compute temperatures from the resistance of each thermistor, m ($m \in [1:16]$). Note that the index m corresponds to thermistor counts in the specific sequence REFi with $i \in [1:4]$, NSRC1, NSRC2, FH1, FH2, WG11, WG12, WG21, WG22, WG31, WG32, WG41, WG42, which is the mux address order of the thermistor data (see **Figure 3**).
  - A[1:16] (Kelvin)
  - B[1:16] (Kelvin/ohm)
  - C[1:16] (Kelvin/ohm^2)
  - D[1:16] (Kelvin/ohm^3)
- Minimum and maximum thermistor temperature threshold limits, Txmin(m) and Txmax(m) respectively, for thermistor m ($m \in [1:16]$). Note that the index m corresponds to thermistor counts in the specific sequence REFi with $i \in [1:4]$, NSRC1, NSRC2, FH1, FH2, WG11, WG12, WG21, WG22, WG31, WG32, WG41, WG42, which is the mux address order of the thermistor data (see **Figure 3**).

- Txmin[1:16] (Kelvin)
- Txmax[1:16] (Kelvin)

- Noise diode temperature correction coefficients, KTN(n,i,j) for a third order correction polynomial. The index n indicates the order of the coefficient (n ∈[0:3]), the index i indicates the channel number (i ∈ [1:4]), and the index j indicates the noise diode (j ∈ [1:3]).
  - KTN[0][1:4][1:3] (Kelvin)
  - KTN[1][1:4][1:3] (/)
  - KTN[2][1:4}[1:3] (Kelvin^{-1})
  - KTN[3][1:4][1:3] (Kelvin^{-2})
- Feedhorn calibration coefficient for sensors 1 and 2, KF1(i) and KF2(i), respectively, and channel i, (i ∈ [1:4]).
  - KF1[1:4] (/)
  - KF2[1:4] (/)
- Path loss constant L(i) for channel i (i ∈ [1:4]). This is needed for mode 1 operation of the JMR.
  - L[1:4] (/)
- Reference load calibration coefficient KR(i) for channel i, (i ∈ [1:4])
  - KR[1:4] (/)
- Waveguide calibration coefficient for sensors 1 and 2, KW1(i) and KW2(i) respectively, and channel i (i ∈ [1:4])
  - KW1[1:4] (/)
  - KW2[1:4] (/)

The Appendix of this document provides a sample of the JMR level 1.0 static auxiliary data file.

2.2 OUTPUT DATA

The JMR level 1.0 data processing algorithms output one set of level 1.0 parameters that are structured in exactly the same sequence as the input PLTM packets. Refer to RD2 for more information about the content of the JMR Level 1.0 output data.

3. TELEMETRY EXTRACTION ALGORITHMS

The goal of the telemetry extraction algorithms is to extract raw JMR measurements and engineering data from JMR PLTM packets into a form that can be used by the JMR science data algorithms.

3.1. PROCESSING OVERVIEW

3.1.1. BRIEF DESCRIPTION

A JMR PLTM packet file is made available to the JMR Level 1.0 (L1) processor by the real-time subsystem, JTCCS. The JMR L1 processor telemetry extraction algorithms then process data in the same sequence as received. The received JMR PLTM packets are nominally expected to have no overlap between packets and to be chronologically ordered. This is verified by the telemetry extraction algorithms, which check for sequential duplicate packets or an unordered sequence of packets and exclude these anomalous packets from any further science data processing. The telemetry extraction algorithms also check for gaps in the telemetry data by using the relevant platform time tag information in the JMR PLTM packets. The telemetry extraction algorithms assemble sets of thermistor data and associated engineering data from the raw data contained in the individual telemetry packets. These data are then made available to the science algorithms for additional processing.

3.1.2 FORMAT OF JMR SOURCE PACKETS

WARNING: This section is provided in this document only for convenience. Information in Figures 1 and 2 below repeats information provided in Version 2.1 of AD7, dated July 12, 1999. Information provided in the most recent version of AD7 always supercedes information provided in this section and should be referenced for the most recent definition of the JMR PLTM packets.

The main characteristics of the JMR telemetry packets (also known as source packets) are as follows:

- The JMR instrument generates a source packet once every 8 seconds
- The first 48 bits of each packet consist of a standard header composed of a packet identifier, a sequence counter, and packet data length as specified in AD3.
- Each source packet contains 8 once per second measurements from the JMR.
- The multiplexed thermistor and engineering data span multiple JMR source packets (2 packets for thermistor data, 4 packets for engineering data).
- Each parameter is coded on a 16-bit word.
- For radiometer counts, all bits are significant.
- For thermistor and engineering counts, only bits 2^0 to 2^{11} are significant.
- The input JMR source packets are nominally expected to be time-ordered and non-redundant, but this is verified within the JMR L1 processor. Sequential duplicate packets and out of sequence packets are excluded from the science data processing.

	1553B TRANSMIT															
WD#	SUBADDRESS		------->													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	PAC ID	R3111	TEMP2	R4224	S2133	TIME4	S3246	N1255	S4359	N2368	ENG2	N4172	R2181	RF11	RF45	RF79
2	SEQC	N3112	R1121	N4225	R2234	TIME4	R3347	S1256	CMD	S2369	TEMP1	S4173	N2182	RF12	RF46	RF81
3	LEN	S3113	N1122	S4226	N2235	TIME4	N3348	R1357	STAT51	R3161	TEMP2	R4274	S2183	RF13	RF47	RF82
4	TIME1	R3214	S1123	R4327	S2236	CNT	S3349	N1358	STAT2	N3162	R1171	N4275	R2284	RF14	RF48	RF83
5	TIME1	N3215	R1224	N4328	R2337	ENG1	R4141	S1359	CDCNT	S3163	N1172	S4276	N2285	RF15	RF49	RF84
6	TIME1	S3216	N1225	S4329	N2338	ENG2	N4142	R2151	TIME6	R3264	S1173	R4377	S2286	RF16	RF51	RF85
7	TIME1	R3317	S1226	CMD	S2339	TEMP1	S4143	N2152	TIME6	N3265	R1274	N4378	R2387	RF17	RF52	RF86
8	TIME1	N3318	R1327	STAT21	R3131	TEMP2	R4244	S2153	TIME6	S3266	N1275	S4379	N2388	RF18	RF53	RF87
9	CNT	S3319	N1328	STAT2	N3132	R1141	N4245	R2254	TIME6	R3367	S1276	CMD	S2389	RF19	RF54	RF88
10	ENG1	R4111	S1329	MDIV1	S3133	N1142	S4246	N2255	TIME6	N3368	R1377	STAT71	R3181	RF21	RF55	RF89
11	ENG2	N4112	R2121	TIME3	R3234	S1143	R4347	S2256	CNT	S3369	N1378	STAT2	N3182	RF22	RF56	SP
12	TEMP1	S4113	N2122	TIME3	N3235	R1244	N4348	R2357	ENG1	R4161	S1379	RVF11	S3183	RF23	RF57	SRCNT
13	TEMP2	R4214	S2123	TIME3	S3236	N1245	S4349	N2358	ENG2	N4162	R2171	TIME8	R3284	RF24	RF58	ITIM1
14	R1111	N4215	R2224	TIME3	R3337	S1246	CMD	S2359	TEMP1	S4163	N2172	TIME8	N3285	RF25	RF59	ITIM2
15	N1112	S4216	N2225	TIME3	N3338	R1347	STAT41	R3151	TEMP2	R4264	S2173	TIME8	S3286	RF26	RF61	ITIM3
16	S1113	R4317	S2226	CNT	S3339	N1348	STA2D	N3152	R1161	N4265	R2274	TIME8	R3387	RF27	RF62	ITIM4
17	R1214	N4318	R2327	ENG1	R4131	S1349	SYCNT	S3153	N1162	S4266	N2275	TIME8	N3388	RF28	RF63	ITIM5
18	N1215	S4319	N2328	ENG2	N4132	R2141	TIME5	R3254	S1163	R4367	S2276	CNT	S3389	RF29	RF64	ITIM6
19	S1216	CMD	S2329	TEMP1	S4133	N2142	TIME5	N3255	R1264	N4368	R2377	ENG1	R4181	RF31	RF65	ITIM7
20	R1317	STAT11	R3121	TEMP2	R4234	S2143	TIME5	S3256	N1265	S4369	N2378	ENG2	N4182	RF32	RF66	ITIM8
21	N1318	STAT2	N3122	R1131	N4235	R2244	TIME5	R3357	S1266	CMD	S2379	TEMP1	S4183	RF33	RF67	CMSA3
22	S1319	UPLIND	S3123	N1132	S4236	N2245	TIME5	N3358	R1367	STAT61	R3171	TEMP2	R4284	RF34	RF68	SP
23	R2111	TIME2	R3224	S1133	R4337	S2246	CNT	S3359	N1368	STAT2	N3172	R1181	N4285	RF35	RF69	SP
24	N2112	TIME2	N3225	R1234	N4338	R2347	ENG1	R4151	S1369	SP	S3173	N1182	S4286	RF36	RF71	SP
25	S2113	TIME2	S3226	N1235	S4339	N2348	ENG2	N4152	R2161	TIME7	R3274	S1183	R4387	RF37	RF72	SP
26	R2214	TIME2	R3327	S1236	CMD	S2349	TEMP1	S4153	N2162	TIME7	N3275	R1284	N4388	RF38	RF73	SP
27	N2215	TIME2	N3328	R1337	STAT31	R3141	TEMP2	R4254	S2163	TIME7	S3276	N1285	S4389	RF39	RF74	SP
28	S2216	CNT	S3329	N1338	STA2C	N3142	R1151	N4255	R2264	TIME7	R3377	S1286	CMD	RF41	RF75	EXBTS
29	R2317	ENG1	R4121	S1339	BUFNO	S3143	N1152	S4256	N2265	TIME7	N3378	R1387	STAT81	RF42	RF76	SELEC
30	N2318	ENG2	N4122	R2131	TIME4	R3244	S1153	R4357	S2266	CNT	S3379	N1388	STAT2	RF43	RF77	SFREV
31	S2319	TEMP1	S4123	N2132	TIME4	N3245	R1254	N4358	R2367	ENG1	R4171	S1389	REF1	RF44	RF78	MA5
32	ECOR1	ECOR2	ECOR3	ECOR4	ECOR5	ECOR6	ECOR7	ECOR8	ECOR9	ECOR10	ECOR11	ECOR12	ECOR13	ECOR14	ECOR15	ECOR16

Figure 1. Matrix Definition of JMR 1553B Source Packets

Figure 1 provides the matrix that defines the content of the 1553B JMR source packets. This matrix is stored in column major form in the JMR source packets and is therefore extracted column by column. The parameters in this matrix are defined with index i denoting channel number ($i \in [1:4]$); j denoting noise diode number ($j \in [1:3]$); k denoting measurement number in packet ($k \in [1:8]$); and l denoting the counter type ($l \in [1:9]$).

Following is a description of each parameter in the JMR 1553B source packets. **Figure 2** provides a bitwise representation of the first three parameters in the JMR 1553B packets, PAC ID, SEQC, and LEN.

		JMR 1553B PACKET MODE 1 AND MODE 2, 10-25-99 (sofdesP.XLS)															
BIT	----	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PACKET ID																	
(PAC ID)		1	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0
		(8D80H DATA SYSTEM A)															
(PAC ID)		1	0	0	0	1	1	0	1	1	1	0	0	0	0	0	0
		(8DC0H DATA SYSTEM B)															
PACKET																	
SEQUENCE																	
CONTROL																	
(SEQC)		1	1	C	C	C	C	C	C	C	C	C	C	C	C	C	C
PACKET																	
LENGTH																	
(LEN)		0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	1
1017 DECIMAL)																	
C - COUNT																	

Figure 2. Bitwise Representation of PAC ID, SEQC and LEN (see AD3)

PAC ID = Packet ID (Constant at (36224 (8D80H) Bd. A, 36288 (8DC0H) Bd. B)

SEQC = Sequence count, the packet number transferred to the spacecraft, starts at 49153 and increments by 1 per packet to 65535, then repeats at 49153

LEN = Packet length, constant at 1017 (3F9H)

TIMEk = 1553B input time for FPGA second k. (Format provided in RD3)

CNT = Number of counts from a 50000 Hz signal from the start of the data taking sequence to the start of the 1PPS. CNT = COUNTER_1PPS - COUNTER_FPGA
The counts will be in the range of 0 to 50150.
A value of 65535 indicates no 1PPS arrived for that FPGA second.
Any other value will indicate an error.

ENGn = Voltage word in counts ($n \in [1:2]$). See Figure 3 for definition.

TEMPn= Temperature word in counts ($n \in [1:2]$). See Figure 3 for definition.

Rijkl = Radiometer mode 1 or mode 2 reference load counts.

Nijkl = Radiometer mode 1 or mode 2 signal counts with noise diode j ON.

Sijkl = Radiometer mode 2 signal counts with noise diode j OFF.These counts are set to 0 in mode 1.

CMD = Command word. Bit assignments are as follows:

- BIT 15 - 0 = Noise diode test on, 1 = Normal
- BIT 14 - 0 = Normal, 1 = Use backup 1PPS
- BIT 13 - 0 = Normal, 1 = Enable Altimeter 2 to inhibit
- BIT 12 - 0 = Normal, 1 = Enable Altimeter 1 to inhibit
- BIT 11 - 0 = Mode 1, 1 = Mode 2 (Bit 11 is data mode)
- BIT 10 - 0 = Normal, 1 = Dicke locked into signal

BIT 9 - 0 = Normal, 1 = Dicke locked into reference
BIT 8 - 0 = Normal, 1 = Force calibration
BIT 7 - 0 = Normal, 1 = Radiometer off
BIT 6 - 0 = Normal, 1 = Disable RF overload protection
BIT 5 - 0 = Noise diode 3 off, 1 = Noise diode 3 on
BIT 4 - 0 = Noise diode 2 off, 1 = Noise diode 2 on
BIT 3 - 0 = Noise diode 1 off, 1 = Noise diode 1 on
BIT 2 - C2, Calibration interval as shown below
BIT 1 - C1
BIT 0 - C0
Every 16*2(C0 + 2*C1 + 4*C2) lines will contain two calibration lines.

STATk1 = Status word 1 for FPGA second k. Bits 2^0 to 2^{10} are mux address bits and bits 2^{11} to 2^{15} are instrument status bits. Mux address bits 0 to 4 are used to address the temperature and voltage muxes (see Figure 3). Mux addresses 5 to 10 are used internally to the FPGA circuits. Bits 2^{11} to 2^{15} are as follows:
BIT 15 - Undefined
BIT 14 - During Mode 1, 0 = data acquisition, 1 = calibration sequence. Undefined during Mode 2.
BIT 13 - Data system ID, 0 = Data system 0
BIT 12 - 1 indicates that channel 4 signal has overflowed.
BIT 11 - 1 indicates that channel 1 signal has overflowed.

STAT2 = Status word 2, bits 2^0 to 2^1 are instrument status bits, and bits 2^2 to 2^{15} are spare status bits that are nominally 0. For bit 2^0, a 1 indicates altimeter pulses at the altimeter 1 input to the FPGA circuits. A 0 indicates no pulses. For bit 2^1, a 1 indicates altimeter pulses at the altimeter 2 input to the FPGA circuits. A 0 indicates no pulses.

STA2C = Status word 2, bits 2^0 to 2^1 are instrument status bits which have the same function as bits 2^0 to 2^1 of STAT2. Bits 2^2 to 2^{15} are memory test results. If (STAT2C AND 0xFFFC) equals 0xC0C0 memory test pass, 0x2000 to 0x3FFF failed address.

STA2D = Status word 2, bits 2^0 to 2^1 are instrument status bits which have the same function as bits 2^0 to 2^1 of STAT2.

SP = Spare words. Nominally 0.

ECOR1 to 16 = Error correction word. By starting at 0xA0 then 'exclusive or' with each of the column words (WD#1 to 31) should equal the ECOR word for that column.

RVF11 = Raw data counter 1 of VF1 of second 8. Same number as packet location R1181.

REF1 = Reference data counter 1 of second 8. Same number as packet location RF81.

MDIV1 = Diagnostic constant = (14003*55000)/14353 = 53658. Any other number shows controller error.

UPLIND = Upload indicator 0xF0FX (where X is error code) or last address of upload, with X as follows:
X = 0, first upload word not 0X6363
X = 1, checksum error
X = 2, word count error
X = 4, upload start address out of range
X = 8, upload end address out of range
The last address is the number of words in the upload multiplied by 2 then added to the start address of the upload.

BUFNO = Space data buffer number, subaddress is as follows:
BUFNO = 1, Subaddress = 1
BUFNO = 2, Subaddress = 2
BUFNO = 4, Subaddress = 3
BUFNO = 8, Subaddress = 4
BUFNO = 16, Subaddress = 5

BUFNO = 32, Subaddress = 6
BUFNO = 64, Subaddress = 7
BUFNO = 128, Subaddress = 8
All other values of BUFNO will be an error

SYCNT = Synchronize with data word mode code counter. Each time the bus controller sends a synchronize with data word command to JMR then this counter increments. Nominally this number equals (SEQC - 49153).

CDCNT = Total 1553 commands counter. Each time the 1553 bus controller sends a 1553 command to the JMR this counter is incremented. In normal operations for an eight second packet there are 32 vector word mode code commands, 8 receive time words commands, 16 transmit 32 word commands and 1 synchronization with data word command for a total of 57 1553 commands. Therefore in normal operations this number will increment by 57 per packet.

RFkl = Reference data counter counts for FPGA data second k and counter type l. Nominal values are 53625 for Mode 2 with no altimeter pulse blanking, variable ± 20% about 53625 with altimeter blanking, 14901 for mode 1 with no altimeter blanking, variable ± 20% about 14901 with altimeter blanking. In mode 1 during calibration cycle 53625 with no altimeter blanking, variable ± 20% about 53625 with altimeter blanking.

ITIMk = Counter_FPGA counts at start of data taking sequence for FPGA second k. Nominally 50000 difference from the previous FPGA second. This counter only counts up to 65535 then resets to 0 and starts counting again.

SRCNT = Serial data packet counter. Starts at 0 and increments by 1 per packet up to 1206. If serial reset is sent prior to 1206 then counter is reset to 0 and starts counting again. If serial stop is sent then it remains at the last counter value.

SFREV = Software revision number where bits B15, B14 are 00 for breadboard, 01 for engineering model, 10 for flight model. BITS B13, B12 are 00 for board A, 01 for board B, etc. BIT B11 is 0 for prom code, 1 for uploaded code. BITS B0 to B10 are the revision number.

MA5 = 0xA5A5 for mem bit test at this location (42405 decimal)

EXBTS = Extbits register monitor. 80C51 registers.

SELEC = Selector register 8 MS bits, Genbits register 8 LS bits. 80C51 for registers.

CMSA3 = Total 16 bit FPGA and secondary commands. Each time A FGPA or secondary command is sent then this counter is incremented.

Note that the JMR measures each of the three counts Rijkl, Nijkl, and Sijkl once every 330 ms, or equivalently three times for every once per second JMR measurement. Each of these three 330 ms measurements is performed with a different noise diode. As such, the index l is redundant and can be determined from the noise diode index j and the symbols R, N and S such that:

- $l = (j-1)\times 3 + 1$ for reference load counts (Rijkl)
- $l = (j-1)\times 3 + 2$ for signal counts with noise diode on (Nijkl)
- $l = (j-1)\times 3 + 3$ for signal counts with noise diode off (Sijkl).

The index l is ignored for the parameters Rijkl, Nijkl and Sijkl in the science data algorithm specifications. Note also that with the index k associated with each of the 8 once per second measurements in a packet is also ignored in the science data algorithm specifications. This is due to the convention that each algorithm pertains to the computations performed on a single once per second measurement.

JMR MUX ADDRESS VS THERMISTOR & ENGINEERING LOCATIONS

MUX ADDR (5 LSB)		TEMP Word		ENG Word	
BIN	**DEC**	**TEMP1**	**TEMP2**	**ENG1**	**ENG2**
00000	0	TCAL-LO1	TCAL-LO2	ECAL-LO1	ECAL-LO2
00001	1	TCAL-HI1	TCAL-HI2	ECAL-HI1	ECAL-HI2
00010	2	CH11	CH12	CH1 +6.5	CH2 +6.5
00011	3	CH21	CH22	CH1 -6.5	CH2 -6.5
00100	4	CH31	CH32	CH1 +5.2V	CH2 +5.2V
00101	5	CH41	CH42	CH1 +10V	CH2 +10V
00110	6	PWR1	PWR2	CH1 -10V	CH2 -10V
00111	7	REF1	REF2	CH1 +15V	CH2 +15V
01000	8	REF3	REF4	CH1 +5V	CH2 +5V
01001	9	NSRC1	NSRC2	CH3 +6.5	CH4 +6.5
01010	10	FH1	FH2	CH3 -6.5V	CH4 -6.5V
01011	11	WG11	WG12	CH3 +5.2V	CH4 +5.2V
01100	12	WG21	WG22	CH3 +10V	CH4 +10V
01101	13	WG31	WG32	CH3 -10V	CH4 -10V
01110	14	WG41	WG42	CH3 +15V	CH4 +15V
01111	15	DATA1	DATA2	CH3 +5V	CH4 +5V
10000	16	TCAL-LO1	TCAL-LO2	INTA +15V	INTB +15V
10001	17	TCAL-HI1	TCAL-HI2	INTA -15V	INTB -15V
10010	18	CH11	CH12	INTA +5V	INTB +5V
10011	19	CH21	CH22	ANA1 +12V	ANA2 +12V
10100	20	CH31	CH32	ANA1 -12V	ANA2 -12V
10101	21	CH41	CH42	ANA1 +5V	ANA2 +5V
10110	22	PWR1	PWR2	GND	GND
10111	23	REF1	REF2	+5 REF1	+5 REF2
11000	24	REF3	REF4	SPARE	SPARE
11001	25	NSRC1	NSRC2	SPARE	SPARE
11010	26	FH1	FH2	SPARE	SPARE
11011	27	WG11	WG12	SPARE	SPARE
11100	28	WG21	WG22	SPARE	SPARE
11101	29	WG31	WG32	SPARE	SPARE
11110	30	WG41	WG42	SPARE	SPARE
11111	31	DATA1	DATA2	SPARE	SPARE

TCAL-LOi = TMUX calibration count, mux i
TCAL-HIi = TMUX calibration count, mux i
CHij = Radiometer channel i, sensor j
PWRi = Power Supply, sensor i
REFi = Reference load, channel i
NSRCi = Noise source, sensor i
FHi = Feedhorn, sensor i
DATAi = Data system, sensor i

ECAL-LOi = EMUX calibration count, mux i
ECAL-HIi = EMUX calibration count, mux i
INTA = Digital electonics bus A
INTB = Digital electonics bus B
ANA = TEMP & ENG MUX voltage

Figure 3. JMR Mux Address vs. Thermistor and Engineering Locations

The command word, CMD, is used to determine the mode of operation of the JMR. In mode 1 operation bit 11 of CMD is equal to 0, and in mode 2 operation bit 11 of CMD is equal to 1. The status word STATk1 is also relevant to mode 1 operation. Bit 14 of STATk1 is set to 0 when the JMR is in mode 1 data acquisition operation, and is set to 1 when the JMR is in mode 1 calibration sequence. Bit 0 of STATk1 is set to 0 when the JMR is in the first second of the 2 second mode 1 calibration sequence, and is set to 1 when the JMR is in the second second of the 2 second mode 1 calibration sequence. During the first second of the mode 1 calibration sequence the first three words of radiometer data for each channel provide DC amp offset data for that channel, and the other words for that second are to be considered as undefined. During the following second of the mode 1 calibration sequence all words provide relevant radiometer calibration data.

The thermistor and engineering data, specified by the words TEMP1, TEMP2, ENG1, and ENG2 in the matrix description defined in **Figure 1,** span multiple packets. In particular, the thermistor data span 2 packets, and the engineering data span 4 packets. Only the thermistor data are relevant to the JMR level 1.0 data processing. The engineering data will not be used in any of the JMR level 1.0 algorithms and will only be used by the JMR instrument team at JPL. **Figure 3** provides the correspondence between the parameters specified by words TEMP1, TEMP2, ENG1, and ENG2 and the specific JMR mux addresses specified by the 5 least significant bits of the word STATk1.

The physical temperatures and calibration references provided in the thermistor words TEMP1 and TEMP2 are assumed to be stable over the 16-second interval spanned by 2 packets, and even as long as 32 seconds (4 packets). The thermistor and calibration counts provided in words TEMP1 and TEMP2 that span 16 seconds of data (2 packets) are considered to form a single set of thermistor and calibration counts. This complete set of thermistor and calibration counts is updated only once every 16 seconds (2 packets) when a completely new set becomes available. If a situation arises in which there is an incomplete set of thermistor data, then that whole set of thermistor data is ignored. Computations that require the thermistor and calibration counts use the set of thermistor or calibration data that is closest in time to the once per second measurement, with the restriction that the set of thermistor or calibration data is within a specified window that is centered at the once per second measurement. This half width of this window is DT_TEMP for thermistor data, and DT_CAL1 for calibration data (see **Table 1**).

3.1.3. LIST OF FUNCTIONS

A list of the functions of the telemetry extraction algorithms is given in **Figure 4**.

FUNCTION
JMR_PKCHK_01 - To check for anomalies in the packet sequence
JMR_PKEXT_01 - To extract raw JMR data from JMR PLTM packets
JMR_PKGAP_01 - To check for gaps in the telemetry

Figure 4: Functions of the Extraction processing

3.2. FUNCTIONS

A detailed description of the functions of the telemetry extraction algorithms is given in this section.

JMR_PKCHK_01 - TO CHECK FOR ANOMALIES IN THE PACKET SEQUENCE
DEFINITION, ACCURACY AND SPECIFICATION

Prepared by:	S. Desai (JPL) T. Antczak (JPL)	
Approved by:	P. Vincent (CNES) B. Haines (JPL)	

Document Ref: SMM-ST-M1-EA-11577-CN		Issue: 3	Update: 0

Algorithm change record	creation	date:	Issue:	Update:
	CCM			

FUNCTION

This function monitors the sequence of JMR PLTM packets to detect duplicate sequential packets or regressions in the sequence of packets.

ALGORITHM DEFINITION

Input Data

- Product data:
  - Current JMR PLTM packet
  - First JMR PLTM packet in data set, or previous non-duplicate or in-sequence JMR PLTM packet
- Computed data: None
- Dynamic auxiliary data: None
- Static auxiliary data: None

Output Data

- Packet status flag

Mathematical Statement

Each JMR packet provides a packet sequence count, SEQC. This counter increments by a value of 1 for each sequential packet by circling through values between 49153 and 65535, inclusive. This algorithm uses this packet sequence count to check for duplicate packets, or regressions in the packet sequence count.

If the current packet and the previous valid packet are found to have identical values of SEQC then the current packet is ignored in all further science data processing. If the current packet and the previous valid packet are found to have regressing values of SEQC then the current packet is ignored in further science data processing.

If the current packet and the previous valid packet are found to have identical values of SEQC, then the data management algorithms compare the two packets to determine whether or not the two packets are identical to each other.

The data management algorithms maintain a count of each of the following three cases and issue separate warning messages to identify the packets that are being excluded from further data processing:

- Packets that are identical to each other.
- Packets that are not identical to each other, but which have identical values of SEQC.
- Packets that have values of SEQC that are out of sequence.

ALGORITHM SPECIFICATION

Input Data

• Packet sequence count of current packet	SEQC1	(\)
• Packet sequence count of previous packet with PKCHK_STATUS=0	SEQC2	(\)

Output Data

• Packet status flag[(1)]	PKCHK_STATUS	(/)
• Execution status		

Processing

Check for duplicate or regressing packet sequence counts:

- Initialize status flag:
  - PKCHK_STATUS = 0
- Check for packets with identical packet sequence count:
  - If (SEQC1 - SEQC2 = 0) then
    - PKCHK_STATUS = 1
- Check for regressing packet sequence counts:
  - If ((SEQC1 - SEQC2 >= -7) and (SEQC1 - SEQC2 <= -1)) OR ((SEQC1 - SEQC2) >= 16376) and (SEQC1 - SEQC2) <= 16382)) then
    - PKCHK_STATUS = 2

The execution status is set to “Good” if all input parameters are found and all output parameters computed, and set to “Bad” otherwise.

Accuracy

None

---

(1) PKCHK_STATUS has a value of 0 if the value SEQC of the current packet appears to be in sequence with the previous valid packet, a value of 1 if the two packets have identical values of SEQC, and a value of 2 if the two packets have out of sequence values of SEQC.

Comments

The JMR instrument has a circular memory buffer that can store up to 8 packets. As long as the memory buffer has not been read and reset, the JMR will first begin to store up to 8 sequential packets. If after 8 packets (64 seconds of data) have been stored and the JMR memory buffer has not been read and reset, the JMR instrument will continue to store packets by circularly overwriting the memory buffer.

In the nominal scenario, the JMR instrument is expected to be interrogated for available packets once every 250 milliseconds. Therefore, in the nominal scenario only 1 packet of JMR data should be stored by the JMR instrument at any time because its memory buffer will nominally be read and reset no more than 250 milliseconds after a packet becomes available.

However, in the scenario where more than one packet is stored by the JMR instrument, the memory buffer will be read in reverse order of the memory buffer, first reading the last packet in the memory buffer and last reading the first packet in the memory buffer. Such a case would make the downloaded packet sequence appear to be in regressing order. In such a case the difference between two sequential packets should be between -1 and -7, if the packet sequence has not looped through its maximum value of 65535, and between 16376 and 16382 when the packet sequence count has looped through its maximum value of 65535.

References

None

JMR_PKEXT_01 - TO EXTRACT RAW JMR DATA FROM JMR PLTM PACKETS

DEFINITION, ACCURACY AND SPECIFICATION

Prepared by:	T. Antczak (JPL) T. Pavlovitch (JPL)	
Approved by:	P. Vincent (CNES) B. Haines (JPL)	

Document Ref: SMM-ST-M1-EA-11577-CN		Issue: 3	Update: 0

Algorithm change record	creation	date:	Issue:	Update:
	CCM			

FUNCTION

This function extracts the raw JMR science and engineering data from the JMR PLTM packets into a form that can be used by the JMR science data algorithms

ALGORITHM DEFINITION

Input Data

- Product data:
  - JMR PLTM packets
- Computed data: None
- Dynamic auxiliary data: None
- Static auxiliary data: None

Output Data

- JMR thermistor data and associated engineering data.

Mathematical Statement

This function generates JMR thermistor data and associated engineering data from JMR PLTM packets. The definition of the JMR PLTM packets is provided in **Figure 1**.

ALGORITHM SPECIFICATION

Input Data

- JMR packets

Output Data

- JMR parameters defined in Section 3.1.2 with the exception of the packet header fields (PAC ID, LEN)
- Execution status

Processing

Read JMR PLTM packet from the input JMR PLTM packet file.

Verify that packet header fields (PAC ID, LEN) contain the fixed values specified in **Figure 2** (see also AD3).

For each of the 8 measurements contained in the packet, transfer all JMR parameters after the packet header, as specified in Section 3.1.2, into integer arrays in local storage.

The execution status in set to “Good” if the entire JMR PLTM packet is successfully read from the PLTM file and all parameters are successfully extracted into local storage. Set to “Bad” otherwise.

Accuracy

None

Comments

None

References

None

JMR_PKGAP_01 - TO CHECK FOR GAPS IN THE TELEMETRY

DEFINITION, ACCURACY AND SPECIFICATION

Prepared by:	J. Stum (CNES) T. Pavlovitch (JPL)	
Approved by:	P. Vincent (CNES) B. Haines (JPL)	

Document Ref: SMM-ST-M1-EA-11577-CN		Issue: 3	Update: 0

Algorithm change record	creation	date:	Issue:	Update:
	CCM			

FUNCTION

This function monitors the GPS UTC time or on-board time from the JMR PLTM packets to detect gaps in the telemetry.

ALGORITHM DEFINITION

Input Data

- Product data:
  - Current JMR PLTM packet
  - Previous JMR PLTM packet
- Computed data: None
- Dynamic auxiliary data: None
- Static auxiliary data:
  - The maximum allowable time gap between successive telemetry packets, DTPKGAP (see **Table 1**)

Output Data

- Continuity check status flag

Mathematical Statement

The difference between the GPS UTC time or on-board time on the previous packet from that on the current packet is compared to a specified allowable duration gap. If the time difference is larger than the allowable duration gap then a continuity fault is detected and the function returns a significant status flag of value 1; otherwise, the function returns a status flag value of 0.

ALGORITHM SPECIFICATION

Input Data

• Time tag of current packet	T1	(sec)
• Time tag of previous packet	T2	(sec)
• Allowable duration gap between successive packets	DTPKGAP	(sec)

Output Data

- Continuity status flag[(1)] PKGAP_STATUS (/)
- Execution status

Processing

If the time difference between the time tag on the current packet and that on the previous packet is larger than DTPKGAP, then the continuity status flag is set to 1; otherwise, there is no gap between the packets and the continuity status flag is set to 0.

- If (T1 - T2) > DTPKGAP then
  - PKGAP_STATUS = 1
- otherwise
  - PKGAP_STATUS = 0

The execution status is set to “Good” if all input parameters are found and all output parameters computed, and set to “Bad” otherwise.

Accuracy

None

Comments

None

References

None

---

(1) PKGAP_STATUS has a value of 1 if a gap in the telemetry is detected and 0 if there is no gap in the telemetry.

4. JMR PROCESSING

4.1. PROCESSING OVERVIEW

4.1.1 BRIEF DESCRIPTION

A brief overview of the main functions of the JMR processing is given in this section. A more detailed description is provided in section 4.2. The JMR can operate in one of two modes, mode 1 or mode 2, and different algorithms to compute antenna temperatures are required for each mode. Mode 2 is the nominal operating mode of the JMR.

- The binary on-board times of each JMR source packet are converted into TAI time.
- The latitude and the longitude corresponding to the middle of each 1-s averaged JMR measurement are computed.
- The validity of elementary radiometer measurements is determined.
- The active 23.8-GHz channel (either channel 2 or channel 3, or both) is determined.
- The radiometer counts Rijkl, Nijkl and Sijkl are renormalized (to account for blanking pulses sent by the JASON altimeter), using the reference data counts RFkl.
- The thermistor counts specified in the words TEMP1 and TEMP2 are converted into path physical temperatures, by using information from high and low calibration resistors, along with the known resistance values of these calibration resistors.
- The thermistor temperatures collected over 2 JMR source packets are assigned to each 1-s JMR measurement.
- The 3 noise diode temperatures in each channel are corrected for the effects of the physical temperatures of the noise source.
- If the JMR operation mode is mode 1, the zero offsets and the system noise temperatures are computed from the 2-second calibration sequence.
- If the JMR operation mode is mode 1, the zero offsets and the system noise temperatures are assigned to each 1-s JMR measurement.
- The individual (330-ms) antenna temperatures are computed, according to the JMR mode (mode 1 or mode 2).
- The antenna temperatures are averaged over 1 s.

Note that the JMR 1.0 product (RD2) provides a JMR mode flag that indicates which mode the instrument is operating. This flag is not specifically generated in any of the following algorithms, but is defined in the data management and control algorithms. The JMR mode flag has the following three values:

- JMR Mode Flag = 0 when JMR is in Mode 1 Data Acquisition Sequence
- JMR Mode Flag = 1 when JMR is in Mode 2

- JMR Mode Flag = 2 when JMR is in Mode 1 Calibration Sequence

4.1.2. LIST OF FUNCTIONS

A list of the functions of the JMR processing is given in **Figure 5.**

FUNCTION
JMR_TME_01 - To determine TAI time of the JMR measurements
SMM_EVERETT - To compute latitude and longitude of the JMR measurements
JMR_VALRAD_01 - To check the validity of the elementary radiometer measurements
JMR_ACTCH_01 - To determine the active 23.8 GHz Channel
JMR_NRMCNT_01 - To renormalize the radiometer counts
JMR_THERM_01 - To convert the thermistor counts into temperatures
JMR_GETTRM_01 - To assign the thermistor temperatures to the once per second JMR measurements
JMR_NDIODE_01 - To correct the noise diode temperatures
JMR_ZEROOFF_01 - To compute the zero offsets for mode 1
JMR_SYSTEMP_01 - To compute the system noise temperatures for mode 1
JMR_ASCAL_01 - To assign the zero offsets and the system noise temperatures to the once per second JMR measurements for mode 1
JMR_TAMODE1_01 - To compute the 330 ms antenna temperatures in mode 1
JMR_TAMODE2_01 - To compute the 330 ms antenna temperatures in mode 2
JMR_TAAVG_01 - To average the antenna temperatures

Figure 5: Functions of the JMR processing

4.2. FUNCTIONS

A detailed description of the functions of the telemetry extraction algorithms is given in this section.

JMR_TME_01 - TO DETERMINE TAI TIME OF THE JMR MEASUREMENTS
DEFINITION, ACCURACY AND SPECIFICATION

Prepared by:	J. Stum (CNES) S. Desai (JPL) A. Kitiyakara (JPL) T. Pavlovitch (JPL)	
Approved by:	P. Vincent (CNES) B. Haines (JPL)	

Document Ref: SMM-ST-M1-EA-11577-CN		Issue: 3	Update: 0

Algorithm change record	creation	date:	Issue:	Update:
	CCM			

FUNCTION

This function determines the TAI time of the JMR measurements from the time tags provided on the JMR telemetry packets.

ALGORITHM DEFINITION

Input Data

- Product data:
  - For a particular once per second measurement denoted by index k
    - * The 5 words TIMEk(n), n=0 to 4
    - * The number of counts, CNT, from a 50 KHz signal
- Computed data: None
- Dynamic auxiliary data: None
- Static auxiliary data:
  - Frequency of signal from which to compute time differential with start time of measurement, CNTFRE
  - File with UTC/TAI time differences (see AD5).
  - SSALTO reference date

Output Data

- TAI time for each measurement
- Flag indicating if time tag originates from GPS UTC time or from On-Board time

Mathematical Statement

The PROTEUS platform on the JASON-1 spacecraft provides time stamps on the JMR packets either in the form of GPS UTC time from the PROTEUS platform GPS, or in the form of On-Board time from the quartz oscillator that is associated with the PROTEUS Data Handling Unit. The algorithm to extract time tags from the JMR packets is identical for both GPS UTC time and On-Board time. Nominally, time is provided as GPS UTC time, unless the spacecraft is in safehold mode in which case On-Board time is provided. Note that the time stamps on the JMR packets already account for leap seconds. In the case of GPS UTC time, the PROTEUS platform provides UTC time by adding UTC leap seconds to the GPS time provided by the on-board GPS receivers. Note that GPS time is exactly equal to UTC time on January 6, 1980, 0 hours.

The definition of the time words is provided in RD3. The first time word, TIMEk(0), specifies the week number in bits 0 to 11, where week 0 is January 6-12, 1980. Bit 12 provides a quality flag for the GPS UTC time by indicating if fewer than four GPS satellites were in visibility when the GPS UTC time was generated. The flag provided in bit 12 is ignored in this algorithm. Bits 13 and 14 are set to 0 for the JMR instrument. Bit 15 indicates that the time stamp is in the form of GPS UTC time when bit 15 = 0, and in the form of On-Board time when bit 15 = 1. The reference data is January 6, 1980, 0 hours.

The second and third time words, TIMEk(1) and TIMEk(2), combine to form a 32-bit integer that specifies the number of seconds in the week, where TIMEk(1) is the most significant word. TIMEk(3) and TIMEk(4) combine to form a 32-bit integer that specifies the number of 2^{-32} seconds in the second, where TIMEk(3) is the most significant word. The time stamp on each one second measurement that is defined by TIMEk[0:4] differs from the actual start time of each of the once per second measurements. This time differential is provided as the number of counts, CNT, of a 50 KHz signal that begins counting from the start of each of the once per second JMR measurements and ends when the time stamp TIMEk[0:4] is placed on each measurement. The UTC time tag for the middle of the once per second measurement is computed by adding 0.5 s to the UTC time tags defined by TIMEk[0:4] and CNT. The final UTC time that corresponds to the middle of the once per second measurement is then converted into TAI time by adding a (TAI-UTC) time difference. The TAI time is then converted into seconds elapsed since the SSALTO reference date. The SSALTO reference date is January 1, 1950 00:00:00.000 (hh:mm:ss).

ALGORITHM SPECIFICATION

Input Data

• Time stamp of once per second JMR measurement	TIME[0:4]	
• Number of counts from a signal of frequency CNTFRE	CNT	(/)
• Frequency of signal to compute time differential	CNTFRE	(Hz)
• SSALTO reference date with respect to January 6, 1980, 0 hours	TREFSSALTO	(sec)
• Number of entries in table of (TAI-UTC) time differences	NTAIMUTC	(/)
• UTC times in table of (TAI-UTC) time differences	UTCTAB[0:NTAIMUTC-1]	(sec)
• Time differences between TAI time and UTC time	DTAIUTC[0:NTAIMUTC-1]	(sec)

Output Data

• Time in TAI time with respect to SSALTO reference date	TSSALTO	(sec)
• Flag indicating origin of time tag(1)	TIMETYPE	(/)
• Execution status		

NOTE: The times UTCTAB[0:NTAIMUTC-1] must span the UTC times of the PLTM packets to be processed otherwise the execution status will be returned as "Bad" even though TSSALTO is still computed.

Processing

(1) TIMETYPE has a value of 0 if the time tag originates from GPS UTC time, and has a value of 1 if the time tag originates from On-Board time.

- TIMETYPE is set to value of bit 15 of TIME(0).
- W is set to value determined by bits 0 to 11 of TIME(0)
- The UTC time of the middle of the once per second measurement in seconds with respect to January 6, 1980, 0 hours is defined by TUTC80.

$$\begin{aligned}\mathrm{TUTC80} &= 0.5 + 604800 \times \mathrm{W} + [65536 \times \mathrm{TIME(1)} + \mathrm{TIME(2)}] \\ &+ [65536 \times \mathrm{TIME(3)} + \mathrm{TIME(4)}] \times 2^{-32} - [\mathrm{CNT/CNTFRE}]\end{aligned} \quad (1)$$

- Determine the index N of the table of (TAI-UTC) time differences to apply to TUTC80
  - If TUTC80 ≥ UTCTAB[NTAIMUTC-1] then
    - * N = NTAIMUTC-1
    - * Execution status set to “Bad”, but continue with computation of TTAI80 and TSSALTO
  - Else if TUTC80 < UTCTAB[0] then
    - * N = 0
    - * Execution status set to “Bad”, but continue with computation of TTAI80 and TSSALTO
  - Otherwise
    - * Determine the index N where UTCTAB[N] ≤ TUTC80 < UTCTAB[N+1]
    - * Execution status set to “Good”, and continue with computation of TTAI80 and TSSALTO
- Compute the TAI time with respect to January 6, 1980, 0 hours, TTAI80

$$\mathrm{TTAI80} = \mathrm{TUTC80} + \mathrm{DTAIUTC[N]} \quad (2)$$

- Compute the time with respect to the SSALTO reference date.

$$\mathrm{TSSALTO} = \mathrm{TTAI80} - \mathrm{TREFSSALTO} \quad (3)$$

<u>Accuracy</u>

None

<u>Comments</u>

Note that the value of CNT will be verified to be in the range from 0 to 50150 by the data management and control algorithms. If it is out of range, then a value of CNT = 0 is input to this algorithm and a warning is issued that an out of range value of CNT was found.

References

None

SMM_EVERETT - TO COMPUTE LATITUDE AND LONGITUDE OF THE JMR MEASUREMENTS

DEFINITION, ACCURACY AND SPECIFICATION

Prepared by:	J. Stum (CNES)	
Approved by:	P. Vincent (CNES) B. Haines (JPL)	

Document Ref: SMM-ST-M1-EA-11577-CN		Issue: 3	Update: 0

Algorithm change record	creation	date:	Issue:	Update:
	CCM			

FUNCTION

This function computes the location (latitude and longitude) of the JMR measurement using the time tag that corresponds to the middle of the once per second measurement.

ALGORITHM DEFINITION

Input Data

- Product data: None
- Computed data:
  - From "JMR_TME_01 - To determine TAI time of the JMR measurements"
    - * Time tag corresponding to the middle of the 1-s averaged JMR measurement
- Dynamic auxiliary data:
  - DORIS Navigator orbit data covering the time span of the input product. i.e. at regular time steps:
    - * Position of the satellite in a terrestrial reference frame: $\vec{P} = (\overline{P_X}, \overline{P_Y}, \overline{P_Z})$
- Static auxiliary data:
  - Processing parameters for the orbit interpolation
  - Processing parameters for the determination of the latitude
  - Desired accuracy for the latitude
- Universal constants:
  - Flattening coefficient of the reference ellipsoid
  - Semi major axis of the reference ellipsoid

Output Data

- Location, i.e. latitude and longitude of the 1-s measurement

Mathematical Statement

- The latitude and the longitude corresponding to an input JMR time-tag are computed as follows:
  - N (typically N=8) position vectors are selected from the input orbit file (N/2 before and N/2 after the JMR time tag). These vectors are interpolated to the JMR time tag corresponding to the middle of the once per second measurement using Everett's formula (Abramowitz, 1965).
  - The interpolated position $\vec{P} = (\overline{P_X}, \overline{P_Y}, \overline{P_Z})$ of the satellite is then projected onto the reference ellipsoid to provide the latitude and longitude (e.g., see Guinn, 1990)

ALGORITHM SPECIFICATION

WARNING: The selection, possible change of frame, and interpolation of the orbit position and velocity vectors to the JMR time tag is considered to be a "data management" algorithm. If the interpolated orbit position and velocity vectors cannot be computed then the latitude and longitude will be set to their default values, but science processing of the JMR data will continue for that data point. The default values for latitude and longitude are the maximum values available to these parameters as defined on the JMR Level 1.0 data product (RD2). These values are 2147483647 microdegrees for latitude, and 4294967295 microdegrees for longitude.

Input Data

- Position of the satellite (interpolated)
  - X coordinate PSx (m)
  - Y coordinate PSy (m)
  - Z coordinate PSz (m)
- Characteristics of the reference ellipsoid:
  - Semi major axis SM_Axis (m)
  - Flattening Flattening (/)
- Thresholds for the iterative process:
  - Desired accuracy for the latitude Acc_Lat (degrees)

Output Data

- Latitude Lat (degrees)
- Longitude Lon (degrees [0, 360[)
- Execution status

Processing

- First, the square of the eccentricity (Ecc2) is computed:

$$\mathrm{Ecc2} = \mathrm{Flattening} \times (2 - \mathrm{Flattening}) \quad (1)$$

- The longitude (Lon ∈ [0, 360[degrees) is computed as follows where the Arctg function is defined in the interval]-π/2. π2[:[1]

$$\text{Lon} = 0 \qquad \text{if PSy = 0 and PSx > 0} \qquad (2)$$

$$\text{Lon} = 90 - \frac{180}{\pi} \times \text{Arctg}\left(\frac{\text{PSx}}{\text{PSy}}\right) \qquad \text{if PSy > 0} \qquad (3)$$

$$\text{Lon} = 180 \qquad \text{if PSy = 0 and PSx < 0} \qquad (4)$$

$$\text{Lon} = 270 - \frac{180}{\pi} \times \text{Arctg}\left(\frac{\text{PSx}}{\text{PSy}}\right) \qquad \text{if PSy < 0} \qquad (5)$$

- The latitude (Lat ∈ [-90, 90] degrees) is computed using the following iterative process:

  – Initialization of the iterative process:

    * The geocentric distance of the satellite (GCdistS) is computed by:

$$\text{GCdistS} = \sqrt{\text{PSx}^2 + \text{PSy}^2 + \text{PSz}^2} \qquad (6)$$

    * The declination (Declin ∈ [-90,90] degrees) of the satellite position is computed by:

$$\text{Declin} = \frac{180}{\pi} \times \text{Arctg}\left(\frac{\text{PSz}}{\sqrt{\text{PSx}^2 + \text{PSy}^2}}\right) \qquad (7)$$

    * The geocentric latitude (Gclat ∈ [-90,90] degrees), the geodetic latitude (Gdlat ∈ [-90,90] degrees) and the geodetic altitude of the satellite (GDaltS) are initialized by:

$$\text{GClat}_0 = \text{Declin} \qquad (8)$$

$$\text{GDlat}_0 = \frac{180}{\pi} \times \text{Arctg}\left(\frac{\text{tg}\left(\frac{\pi}{180} \times \text{GClat}_0\right)}{(1 - \text{Flattening})^2}\right) \qquad (9)$$

$$\text{GDaltS}_0 = \text{GCdistS} \qquad (10)$$

---

[1] The case PSx = PSy = 0 should not occur.

– The geodetic altitude of the satellite, the geocentric distance of the nadir point (GCdistN), and the geocentric and geodetic latitudes of the nadir point are then computed iteratively (loop index i starting from 1 by equations (11) to (19).

* Computation of the geocentric distance of the nadir point:

$$\mathrm{GCdistN}_i = \frac{\mathrm{SM\_Axis} \times (1 - \mathrm{Flattening})}{\sqrt{1 - \mathrm{Ecc2} \times \cos^2\left(\frac{\pi}{180} \times \mathrm{GClat}_{i-1}\right)}} \qquad (11)$$

* Computation of the topocentric aspect angle (TAangle)

$$\mathrm{TAangle}_i = \mathrm{GDlat}_{i-1} - \mathrm{GClat}_{i-1} \qquad (12)$$

* Computation of the geodetic altitude of the satellite:

$$\mathrm{GDaltS}_i = \sqrt{\mathrm{GSdistS}^2 - \mathrm{GCdistN}_i^2 \times \sin^2\left(\frac{\pi}{180} \times \mathrm{TAangle}_i\right)} - \mathrm{GCdistN}_i \times \cos^2\left(\frac{\pi}{180} \times \mathrm{TAangle}_i\right) \qquad (13)$$

* Computation of geocentric aspect angle:

$$\mathrm{GAangle}_i = \frac{180}{\pi} \times \arcsin\left[\frac{\mathrm{GDaltS}_i}{\mathrm{GCdistS}} \times \sin\left(\frac{\pi}{180} \times \mathrm{TAangle}_i\right)\right] \qquad (14)$$

* Computation of the geocentric latitude:

$$\mathrm{GClat}_i = \mathrm{Declin} - \mathrm{GAangle}_i \qquad (15)$$

* Computation of the geodetic latitude

$$\mathrm{GDlat}_i = \frac{180}{\pi} \times \mathrm{Arctg}\left[\frac{\mathrm{tg}\left(\frac{\pi}{180} \times \mathrm{GClat}_i\right)}{(1 - \mathrm{Flattening})^2}\right] \qquad (16)$$

* Test on the precision of the estimates:

If $\left|\mathrm{GDlat}_i - \mathrm{GDlat}_{i-1}\right| \le \mathrm{Acc\_Lat}$ then

Lat = Gdlat_i, the iteration loop is exited, and the algorithm returns an execution status of "Good"

else return to equation (11)

Accuracy

- The error due to Everett's interpolation method is smaller than 1mm if the number N or orbit points taken into account is large enough (typically N = 8, i.e. 4 points before and 4 points after the JMR time tag).

Comments

"tg" represents the tangent function, while "Arctg" represents the inverse tangent function.

References

- Abramowitz M. And Stegun I.A., Handbook of Mathematical Functions, Dover Publication Inc. N.Y., 1965
- Guinn, J.R., Definition of Reference Earth Ellipsoid for TOPEX/POSEIDON, JPL Interoffice Memorandum, 314.5-1409, 15 February 1990.

JMR_VALRAD_01 - TO CHECK THE VALIDITY OF THE ELEMENTARY RADIOMETER MEASUREMENTS

DEFINITION, ACCURACY AND SPECIFICATION

Prepared by:	J. Stum (CNES) S. Desai (JPL) A. Kitiyakara (JPL) T. Pavlovitch (JPL)	
Approved by:	P. Vincent (CNES) B. Haines (JPL)	

Document Ref: SMM-ST-M1-EA-11577-CN		Issue: 3	Update: 0

Algorithm change record	creation	date:	Issue:	Update:
	CCM			

FUNCTION

This function determines 36 validity flags for each of the 36 radiometer count measurements provided with each once per second JMR measurement.

ALGORITHM DEFINITION

Input Data

- Product data:
  - For a particular once per second measurement denoted by index k
    - 36 Radiometer counts R_{ijkl}, N_{ijkl} , and S_{ijkl} ($i \in [1:4]$, $j \in [1:3]$)
- Computed data: None
- Dynamic auxiliary data: None
- Static auxiliary data:
  - Default value of radiometer counts, DEFCNT (see **Table 1**).

Output Data

- 36 validity flags associated with each of the 36 unnormalized radiometer count measurements in a once per second JMR measurement

Mathematical Statement

A validity flag is assigned to each of the 36 radiometer count measurements that are provided with a once per second measurement by flagging those radiometer counts with values set to the default value, DEFCNT. This algorithm assumes that any missing words or invalid measurements have radiometer count values set to the default value, DEFCNT. The default value, DEFCNT, should nominally be zero.

ALGORITHM SPECIFICATION

Input Data

- For the once per second JMR measurements for channel $i \in [1:4]$, noise diode $j \in [1:3]$:
  - 12 unnormalized reference load counts R[1:4][1:3] (/)
  - 12 unnormalized signal counts with noise diode on N[1:4][1:3] (/)
  - 12 unnormalized signal counts with noise diode off S[1:4][1:3] (/)
- Default value for radiometer counts DEFCNT (/)

Output Data

- 12 validity flags for unnormalized reference load counts[(1)] VFUR[1:4][1:3] (/)
- 12 validity flags for unnormalized signal counts with noise diode on[(1)] VFUN[1:4][1:3] (/)
- 12 validity flags for unnormalized signal counts with noise diode off[(1)] VFUS[1:4][1:3] (/)
- Execution status

Processing

- For each channel i ∈ [1:4], and noise diode j ∈ [1:3], set all flags to "invalid":
  - VFUR(i,j) = 1
  - VFUN(i,j) = 1
  - VFUS(i,j) = 1
- For each channel i ∈ [1:4], and noise diode j ∈ [1:3], check that radiometer counts do not have default value.
  - If R(i,j) ≠ DEFCNT then VFUR(i,j) = 0
  - If N(i,j) ≠ DEFCNT then VFUN(i,j) = 0
  - If S(i,j) ≠ DEFCNT then VFUS(i,j) = 0
- If all 36 of the radiometer counts have the default value
  - Set execution status to "Bad"
- Otherwise,
  - Set execution status to "Good"

Accuracy

None

Comments

None

References

None

---

(1) The validity flags VFUR(i,j), VFUN(i,j), and VFUS(i,j) are set to a value of 1 if the corresponding unnormalized radiometer count has the default value, DEFCNT, and set to 0 otherwise.

JMR_ACTCH_01 - TO DETERMINE THE ACTIVE 23.8 GHZ CHANNEL
DEFINITION, ACCURACY AND SPECIFICATION

Prepared by:	J. Stum (CNES) S. Desai (JPL) A. Kitiyakara (JPL) T. Pavlovitch (JPL)	
Approved by:	P. Vincent (CNES) B. Haines (JPL)	

Document Ref: SMM-ST-M1-EA-11577-CN		Issue: 3	Update: 0

Algorithm change record	creation	date:	Issue:	Update:
	CCM			

FUNCTION

This function determines which of the two 23.8 GHz channels is in operation for each once per second JMR measurement.

ALGORITHM DEFINITION

Input Data

- Product data:
  - Command word, CMD
  - Status word 1, STATk1
- Computed data:
  - From "JMR_VALRAD_01 - To check the validity of the elementary radiometer measurements"
    - * Validity flags for unnormalized radiometer counts from channels 2 and 3
- Dynamic auxiliary data: None
- Static auxiliary data: None

Output Data

- Flag indicating the active 23.8 GHz channel

Mathematical Statement

Channels 2 and 3 of the JMR are both 23.8 GHz channels. Channel 3 is the nominal 23.8 GHz channel and channel 2 is the redundant 23.8 GHz channel. The nominal configuration of the JMR instrument has only one of channels 2 or 3 in operation at any given time. However, it is possible for the JMR instrument to have both 23.8 GHz channels active at the same time. Both channels would be commanded to be active simultaneously only for diagnostic and calibration evaluation purposes, and this should be a rare occurrence if at all. In order to allow for all possible configurations of the JMR instrument, all subsequent algorithms allow for the unusual configuration of both 23.8 GHz channels being simultaneously active by performing computations on both 23.8 GHz channels if both appear to be active. However, in the nominal configuration where only one 23.8 GHz channel is active at any given time, computations are only performed on the active 23.8 GHz channel. In the special case where both 23.8 GHz channels appear to be active, those applications that expect a single antenna temperature measurement at the 23.8 GHz frequency should ignore the antenna temperature from Channel 2, and should use the antenna temperature from Channel 3 as the nominal antenna temperature measured at the 23.8 GHz frequency. When the JMR is operating in mode 1 data acquisition (bit 11 of CMD = 0, bit 14 of STATk1 = 0) the radiometer counts associated with the bins for radiometer counts from the antenna with noise diode off, S_{ijkl}, are always zero valued. As such a 23.8 GHz channel is determined to be inactive in mode 1 data acquisition operation when all 6 radiometer counts R_{ijkl} and N_{ijkl} with $j \in [1:3]$ are set to the default value, DEFCNT. In all other cases, namely in mode 2 operation (bit 11 of CMD = 1) and the mode 1 calibration sequence (bit 11 of CMD = 0, bit 14 of STATk1 = 1) a 23.8 GHz channel is determined to be inactive when all nine radiometer counts, R_{ijkl}, N_{ijkl} and S_{ijkl}, are set to the default value, DEFCNT. The default value should nominally have a value of zero.

ALGORITHM SPECIFICATION

Input Data

- For the once per second JMR measurements for channel i ∈ [1:4], noise diode j ∈ [1:3]:
  - Validity flag for unnormalized reference load counts(1) VFUR[1:4][1:3] (/)
  - Validity flag for unnormalized signal counts with noise diode on(1) VFUN[1:4][1:3] (/)
  - Validity flag for unnormalized signal counts with noise diode off(1) VFUS[1:4][1:3] (/)
  - Command word CMD (/)
  - Status word 1 STAT1 (/)

Output Data

- Flag denoting active 23.8 GHz channel(2) ACT238 (/)
- Execution status

Processing

- Initialize active channel flag to indeterminate value.
  - ACT238 = 0
- For each of channels 2 and 3, i = 2, and i = 3
  - Set a flag ACTCH(i-1) = 0
  - For each j ∈ [1:3]
    * If VFUR(i,j) = 0 then ACTCH(i-1) = 1
    * If VFUN(i,j) = 0 then ACTCH(i-1) = 1

---

(1) The validity flags VFUR(i,j), VFUN(i,j), and VFUS(i,j) are set to a value of 1 if the corresponding unnormalized radiometer count has the default value, DEFCNT, and set to 0 otherwise

(2) ACT238 has a value of 0 if both 23.8 GHz channels (channels 2 and 3) are inactive, 2 if channel 2 is the only active 23.8 GHz channel, 3 if channel 3 is the only active 23.8 GHz channel, and 5 if channels 2 and 3 are both active.

* If in mode 2 [bit 11 of CMD = 1] OR in mode 1 calibration sequence [bit 11 of CMD = 0 AND bit 14 of STAT1 = 1] then

  If VFUS(i,j) = 0 then ACTCH(i-1) = 1

- If ACTCH(1) = ACTCH(2) = 1 then both 23.8 GHz channels appear to be active, then
  - ACT238 = 5
  - Set execution status to "Good"
- else if ACTCH(1) = 1 and ACTCH(2) = 0, then
  - ACT238 = 2
  - Set execution status to “Good”
- else if ACTCH(2) = 1, and ACTCH(1) = 0, then
  - ACT238 = 3
  - Set execution status to “Good”
- Else both channels appear to be inactive (ACTCH(1) = ACTCH(2) = 0), then
  - Set execution status to "Bad"

Accuracy

None

Comments

None

References

None

JMR_NRMCNT_01 - TO RENORMALIZE THE RADIOMETER COUNTS
DEFINITION, ACCURACY AND SPECIFICATION

Prepared by:	J. Stum (CNES) S. Desai (JPL) A. Kitiyakara (JPL) T. Pavlovitch (JPL)	
Approved by:	P. Vincent (CNES) B. Haines (JPL)	

Document Ref: SMM-ST-M1-EA-11577-CN		Issue: 3	Update: 0

Algorithm change record	creation	date:	Issue:	Update:
	CCM			

FUNCTION

This function renormalizes the radiometer counts to account for altimeter blanking pulses

ALGORITHM DEFINITION

Input Data

- Product data:
  - For a particular once per second measurement denoted by index k
    - Unnormalized radiometer counts R_{ijkl}, N_{ijkl} , and S_{ijkl} ($i \in [1:4]$, $j \in [1:3]$)
    - Reference data counter counts RF_{kl}
    - Command word, CMD
    - Status word 1, STATk1
    - Status word 2, STAT2, STAT2C, or STAT2D
- Computed data:
  - From “JMR_VALRAD_01 - To check the validity of the elementary radiometer measurements”
    - Validity flags for unnormalized radiometer counts
  - From “JMR_ACTCH_01 - To determine the active 23.8 GHz Channel”
    - Flag indicating the active 23.8GHz channel
- Dynamic auxiliary data: None
- Static auxiliary data:
  - Minimum tolerance for division by counts, MINTOLCNT (see **Table 1**)
  - Normalization constant, K_NORM (see **Table 1**.)
  - Default value of radiometer counts, DEFCNT (see **Table 1**).

Output Data

- 36 normalized counts, RN_{ij}, NN_{ij}, and SN_{ij} for each channel $i \in [1:4]$, and noise diode $j \in [1:3]$.
- 36 validity flags for normalized radiometer counts.

Mathematical Statement

The JASON altimeter generates altimeter-blanking pulses that are used to disable the radiometer data counters during altimeter transmission. This prevents altimeter noise from introducing errors in the radiometer measurements. As a consequence of the altimeter blanking pulses, the actual counting period from one counter gate period to the next varies slightly. More specifically, this is due to the variable overlap between the "settling time" pulses generated by the JMR timing circuits and the nominal altimeter blanking pulses. To correct for this variable counting time, a reference counter has been implemented which counts a 2 MHz fixed clock signal and is gated in parallel with the data counters. The reference count is then used to renormalize the corresponding radiometer measurement.

When the JMR is operating in mode 1 data acquisition (bit 11 of CMD = 0 AND bit 14 of STATk1 = 0), and altimeter blanking pulses are not present, the reference counts will overflow the integer size that is allocated to the reference counts in the telemetry packets. In such cases, the correct value of the reference counts is computed by adding 65536 to the reference count value that is extracted from the telemetry packets. Altimeter blanking pulses are present and enabled when (bit 12 of CMD = 1 and bit 0 of STAT2 = 1), or when (bit 13 of CMD = 1 and bit 1 of STAT2 = 1). It should be noted that the status word STAT2 actually refers to the status words STAT2, STAT2C or STAT2D from the 8-second telemetry packets. Bits 0 and 1 of the status words STAT2, STAT2C and STAT2D are instrument status bits with exactly the same function. The status word STAT2 is provided in seconds 1, 2, 5, 6, 7, and 8 of the 8 second telemetry packet, while the status word STAT2C is provided in place of STAT2 in second 3, and the status word STAT2D is provided in place of STAT2 in second 4.

Each radiometer measurement must be renormalized using the corresponding reference count. Given the nominal 0.11 second accumulation per measurement, each measurement is normalized as follows:

$$X_{corrected} = X_{uncorrected} \times \frac{K\_NORM}{RF} \qquad (1)$$

where K_NORM is a constant = (0.11 second x 2MHz)/4 = 55000, $X_{uncorrected}$ represents the uncorrected counts R_{ijkl}, N_{ijkl}, and S_{ijkl}, $X_{corrected}$ represents the respective renormalized counts, RN_{ij}, NN_{ij}, and SN_{ij}, and RF is the corresponding reference data count.

Note that this algorithm must check for zero valued reference data counts. If the reference data count is zero then the normalized radiometer counts are set to a default value and flagged as "bad" values.

ALGORITHM SPECIFICATION

Input Data

- For the once per second JMR measurements for channel i ∈ [1:4], noise diode j ∈ [1:3]:
  - Unnormalized reference load counts R[1:4][1:3] (/)
  - Unnormalized signal counts with noise diode on N[1:4][1:3] (/)

- Unnormalized signal counts with noise diode off S[1:4][1:3] (/)
- 12 validity flags for unnormalized reference load counts[1] VFUR[1:4][1:3] (/)
- 12 validity flags for unnormalized signal counts with noise diode on[1] VFUN[1:4][1:3] (/)
- 12 validity flags for unnormalized signal counts with noise diode off[1] VFUS[1:4][1:3] (/)
- Flag denoting active 23.8 GHz channel[2] ACT238 (/)

• For the once per second JMR measurements
  - Reference data counts RF[1:9] (/)
  - Command word CMD (/)
  - Status word 1 STAT1 (/)
  - Status word 2[3] STAT2 (/)

• Tolerance for division by counts MINTOLCNT (/)

• Normalization constant K_NORM (/)

• Default value for radiometer counts DEFCNT (/)

Output Data

• For the once per second JMR measurements for channel i ∈ [1:4], noise diode j ∈ [1:3]:
  - Normalized reference load counts RN[1:4][1:3] (/)
  - Normalized signal counts with noise diode on NN[1:4][1:3] (/)
  - Normalized signal counts with noise diode off SN[1:4][1:3] (/)

---

(1) The validity flags VFUR(i,j), VFUN(i,j), and VFUS(i,j) are set to a value of 1 if the corresponding unnormalized radiometer count has the default value, DEFCNT, and set to 0 otherwise

(2) ACT238 has a value of 0 if both 23.8 GHz channels (channels 2 and 3) are inactive, 2 if channel 2 is the only active 23.8 GHz channel, 3 if channel 3 is the only active 23.8 GHz channel, and 5 if channels 2 and 3 are both active.

(3) STAT2 refers to status words STAT2, STAT2C or STAT2D from the 8 second telemetry packets, since STAT2 is provided in seconds 1, 2, 5, 6, 7, and 8 of the 8 second telemetry packet, while STAT2C is provided in place of STAT2 in second 3, and STAT2D is provided in place of STAT2 in second 4.

- Validity flags for the normalized radiometer counts for each channel i ∈ [1:4], and noise diode j ∈ [1:3]:
  - 12 validity flags for normalized reference load counts[4] VFR[1:4][1:3] (/)
  - 12 validity flags for normalized signal counts with noise diode on[4] VFN[1:4][1:3] (/)
  - 12 validity flags for normalized signal counts with noise diode off[4] VFS[1:4][1:3] (/)
- Execution status

<u>Processing</u>

- Initialize execution status to “good”.
- If in mode 1 data acquisition sequence [bit 11 of CMD = 0 AND bit 14 of STAT1 = 0] then check for overflow condition of reference counts, and correct for this overflow:
  - Initialize altimeter blanking pulses flag to indicate altimeter-blanking pulses not present.
    - * ALTBLANK = 0
  - Check for altimeter blanking pulses present and enabled:
    - * If (bit 12 of CMD = 1) AND (bit 0 of STAT2 = 1) then ALTBLANK = 1
    - * If (bit 13 of CMD = 1) AND (bit 1 of STAT2 = 1) then ALTBLANK = 1
  - If ALTBLANK = 0, then for each l ∈ [1:9]:
    - * RF[l] = RF[l] + 65536
- For each channel i ∈ [1:4], and each noise diode j ∈ [1:3]
  - Initialize normalized counts to default value.
    - * RN(i,j) = NN(i,j) = SN(i,j) = DEFCNT
  - Initialize validity flags of normalized counts to “invalid”:
    - * VFR(i,j) = VFN(i,j) = VFS(i,j) = 1

---

[4] The validity flags VFR(i,j), VFN(i,j), and VFS(i,j) are set to a value of 1 if the corresponding normalized radiometer count has the default value, DEFCNT, and set to 0 otherwise.

- Only normalize counts for active channels: if i = 1, or i = 2 AND ACT238 = 2 or 5, or i = 3 AND ACT238 = 2 or 5, or i = 4 then

  * For valid radiometer reference load counts, where l = (j-1)×3 + 1, if VFUR(i,j) = 0 then

    If RF(l) > MINTOLCNT then set flag to "valid" and normalize counts:

    $$\mathrm{VFR(i, j) = 0} \quad (1)$$

    $$\mathrm{RN(i, j) = R(i, j) \times \frac{K\_NORM}{RF(l)}} \quad (2)$$

    Otherwise, set execution status to "bad".

  * For valid signal counts with noise diode on, where l =(j-1)×3 + 2, if VFUN(i,j) = 0 then

    If RF(l) > MINTOLCNT then set flag to "valid" and normalize counts:

    $$\mathrm{VFN(i, j) = 0} \quad (3)$$

    $$\mathrm{NN(i, j) = N(i, j) \times \frac{K\_NORM}{RF(l)}} \quad (4)$$

    Otherwise set execution status to "bad".

  * For valid signal counts with noise diode off, where l = (j-1)×3 + 3, if VFUS(i,j) = 0 then

    If RF(l) > MINTOLCNT then set flag to "valid" and normalize counts:

    $$\mathrm{VFS(i, j) = 0} \quad (5)$$

    $$\mathrm{SN(i, j) = S(i, j) \times \frac{K\_NORM}{RF(l)}} \quad (6)$$

    Otherwise set execution status to "bad".

Accuracy

None

Comments

None

References

- Kitiyakara, A., Addendum to Memo *JASON Microwave Radiometer (JMR)* T_A *Retrieval* dated April 6, 1998, JPL Interoffice Memorandum, April 17, 1998.

JMR_THERM_01 - TO CONVERT THE THERMISTOR COUNTS INTO TEMPERATURES

DEFINITION, ACCURACY AND SPECIFICATION

Prepared by:	J. Stum (CNES) S. Desai (JPL) A. Kitiyakara (JPL) T. Pavlovitch (JPL)	
Approved by:	P. Vincent (CNES) B. Haines (JPL)	

Document Ref: SMM-ST-M1-EA-11577-CN		Issue: 3	Update: 0

Algorithm change record	creation	date:	Issue:	Update:
	CCM			

FUNCTION

This function converts the thermistor counts into temperatures in Kelvin

ALGORITHM DEFINITION

Input Data

- Product data:
  - Low and high calibration counts for sensor 1, TCALLO(1) and TCALHI(1)
  - Low and high calibration counts for sensor 2, TCALLO(2) and TCALHI(2)
  - Feedhorn thermistor counts for sensors 1 and 2, FH1 and FH2, respectively
  - Noise source counts for sensors 1 and 2, NSRC1 and NSRC2, respectively
  - For each of the four channels $i \in [1:4]$
    - Waveguide thermistor counts for sensors 1 and 2, WGi1 and WGi2, respectively
    - Reference load thermistor counts, REFi
- Computed data: None
- Dynamic auxiliary data: None
- Static auxiliary data:
  - Minimum tolerance for division by counts, MINTOLCNT (see **Table 1**)
  - Low and high calibration resistance values for sensor 1, RLO(1) and RHI(1) (see **Table 1**)
  - Low and high calibration resistance values for sensor 2, RLO(2) and RHI(2) (see **Table 1**)
  - For each of 16 thermistors, four calibration coefficients to compute temperatures from resistances, A(m), B(m), C(m), and D(m), $m \in [1:16]$ (see **Table 1**)
  - For each of 16 thermistors, maximum and minimum threshold limits on thermistor temperatures, Txmax(m), Txmin(m), $m \in [1:16]$ (see **Table 1**)

Output Data

- Feedhorn thermistor temperatures for sensors 1 and 2
- Noise source temperatures for sensors 1 and 2

- For each of the four channels i ∈ [1:4]
  - Waveguide thermistor temperatures for sensors 1 and 2
  - Reference load thermistor temperatures
- 16 temperature quality flags indicating if any of thermistor temperature thresholds have been exceeded

Mathematical Statement

The Jason temperature sensors consist of YS1 thermistors, each connected in parallel with a 5.23kΩ resistor. A constant current source (≈ 1 mA) is switched across each thermistor-resistor pair, and then across each of two calibration resistors (LO and HI) with the resulting voltages converted to counts by a voltage-to-frequency converter and a gated frequency counter. The calibration resistors provide an accurate measure of the unknown resistance independent of current drift or voltage offset, dependent only on the calibration resistances. The conversion of the counts into temperatures is a two step process, first converting the counts into resistances (Ohms), then converting the resistances into temperatures (Kelvin).

In the first step, the 16 resistances R(m) with m ∈[1:16], are computed from:

$$R(m) = RLO(n) + \frac{RHI(n) - RLO(n)}{TCALHI(n) - TCALLO(n)} \times (X(m) - TCALLO(n)) \qquad (1)$$

where X(m) are the thermistor counts, REFi with i ∈ [1:4], NSRC1, NSRC2, FH1, FH2, WG11, WG12, WG21, WG22, WG31, WG32, WG41, and WG42, that are respectively assigned index m ∈ [1:16]. In equation (1) the index n denotes the sensor number and the relationship between index n and m can be inferred from **Figure 3**, where:

- n = 1 when m is odd, for REF1, REF3, NSRC1, FH1, and WGi1 for i ∈ [1:4]
- n = 2 when m is even for REF2, REF4, NSRC2, FH2, and WGi2 for i ∈ [1:4]

In the second step, the resistances are converted to temperatures. The JMR thermistors are wired with a fixed parallel resistor which allows the thermistor temperatures (in Kelvin), T(m) with m ∈ [1:16], to be related to the thermistor resistances (in Ohms), R(m) with m ∈ [1:16], by the following relationship.

$$T(m) = A(m) + B(m)R(m) + C(m)R^2(m) + D(m)R^3(m) \qquad (2)$$

Each of the 16 temperatures should then lie within their respective nominal thresholds:

$$Txmin(m) \le T(m) \le Txmax(m) \qquad (3)$$

A temperature quality flag is set to indicate if the respective threshold limit has been exceeded.

Note that the 16 thermistor counts that are converted into temperatures by this algorithm are derived from 2 JMR data packets and span 16 seconds of once per second data measurements. These 16 thermistor measurements should always be considered as a single set of thermistor measurements. The set of 16 thermistor temperatures and quality flags will be assigned a time tag that corresponds to the time tag of the last second of thermistor data that is used to compose the set of 16 thermistor measurements, and a table in time of the thermistor temperatures and associated quality flags will be generated. Assigning the time tag to the thermistor temperatures and quality flags and generation of the table of thermistor data will be considered to be part of “data management and control algorithms”. Note that the continuity status flag PKGAP_STATUS generated by “JMR_PKGAP_01 - To check for gaps in the telemetry” is used when accumulating the sets of thermistor data over two consecutive data packets, by indicating that gaps in the sets of thermistor data have occurred, or by indicating that a complete set of thermistor data could not be accumulated over the 16 seconds spanned by 2 data packets.

ALGORITHM SPECIFICATION

Input Data

• Low calibration counts for sensors 1 and 2	TCALLO[1:2]	(/)
• High calibration counts for sensors 1 and 2	TCALHI[1:2]	(/)
• Thermistor counts	X[1:16]	(/)
• Tolerance for division by counts	MINTOLCNT	(/)
• Low calibration resistances	RLO[1:2]	(Ohms)
• High calibration resistances	RHI[1:2]	(Ohms)
• Calibration coefficients	A[1:16]	(Kelvin)
• Calibration coefficients	B[1:16]	(Kelvin/Ω)
• Calibration coefficients	C[1:16]	(Kelvin/Ω^2)
• Calibration coefficients	D[1:16]	(Kelvin/Ω^3)
• Minimum threshold for 16 thermistor temperatures	Txmin[1:16]	(Kelvin)
• Maximum threshold for 16 thermistor temperatures	Txmax[1:16]	(Kelvin)

Output Data

• Thermistor temperatures	T[1:16]	(Kelvin)

- Thermistor temperature quality flag[(1)] TQUAL (/)
- Execution status

Note: The thermistor counts, X[1:16], calibration coefficients, A[1:16], B[1:16], C[1:16], and D[1:16], thermistor temperature thresholds, Txmin[1:16] and Txmax[1:16], thermistor temperatures, T[1:16], and each bit (from the least significant to the most significant) of the 16 bit thermistor temperature quality flag, TQUAL, are all identically ordered in the sequence that corresponds to REFi with i ∈ [1:4], NSRC1, NSRC2, FH1, FH2, WG11, WG12, WG21, WG22, WG31, WG32, WG41, WG42 (see Figure 3).

<u>Processing</u>

- Initialize 16 bit thermistor temperature quality flag to “bad” value, and thermistor temperatures to 0
  - Set all T[1:16] = 0
  - TQUAL = 32768
- Check that the low and high calibration counts are not equal to each other:
  - If TCALHI(1) - TCALLO(1) ≤ MINTOLCNT or TCALHI(2) - TCALLO(2) ≤ MINTOLCNT then
    - Set execution status to “Bad” and exit algorithm
- For each thermistor m ∈ [1:16]:
  - Define the index n as follows:
    - n = 1 when m is odd for REF1, REF3, NSRC1, FH1, and WGi1 for i ∈ [1:4]
    - n = 2 when m is even for REF2, REF4, NSRC2, FH2, and WGi2 for i ∈ [1:4]
  - Compute the thermistor resistances:
    - $$R(m) = RLO(n) + \frac{RHI(n) - RLO(n)}{TCALHI(n) - TCALLO(n)} \times (X(m) - TCALLO(n)) \quad (1)$$
  - Compute the thermistor temperatures:
    - $$T(m) = A(m) + B(m)R(m) + C(m)R^2(m) + D(m)R^3(m) \quad (2)$$

---

(1) Bit m of the 16 bit integer TQUAL is set to 0 if the temperature of thermistor (m+1) is within its respective threshold, and 1 otherwise. Therefore, the 16 bit integer TQUAL has a value of 0 when all 16 thermistor temperatures are within their respective thresholds.

- Check that the thermistor temperature thresholds are satisfied:
  - * If $\text{Txmin}(m) \le T(m) \le \text{Txmax}(m)$ then set bit (m-1) of TQUAL to 0
- Set execution status to "Good"

Accuracy

None

Comments

None

References

• Kitiyakara, A., JASON Thermistor Calibration, JPL Interoffice Memorandum, December 10, 1997.

JMR_GETTRM_01 - TO ASSIGN THE THERMISTOR TEMPERATURES TO THE ONCE PER SECOND JMR MEASUREMENTS

DEFINITION, ACCURACY AND SPECIFICATION

Prepared by:	J. Stum (CNES) S. Desai (JPL) A. Kitiyakara (JPL) T. Pavlovitch (JPL)	
Approved by:	P. Vincent (CNES) B. Haines (JPL)	

Document Ref: SMM-ST-M1-EA-11577-CN		Issue: 3	Update: 0

Algorithm change record	creation	date:	Issue:	Update:
	CCM			

FUNCTION

This function assigns the thermistor temperatures to a once per second JMR measurement.

ALGORITHM DEFINITION

Input Data

- Product data: None
- Computed data:
  - From "JMR_TME_01 - To determine TAI time of the JMR measurements"
    - Time tags corresponding to the middle of the 1-s averaged JMR measurement
  - Table in time of 16 thermistor temperatures and their respective thermistor temperature quality flags
- Dynamic auxiliary data: None
- Static auxiliary data:
  - Time difference threshold DT_TEMP (see **Table 1**)

Output Data

- 16 thermistor temperatures and respective quality flags that are assigned to the once per second JMR measurement
- Flag indicating if valid set of thermistor data assigned to once per second data, TH_FLAG

Mathematical Statement

The algorithm "JMR_THERM_01 - To convert the thermistor counts into temperatures" computes thermistor temperatures whenever a complete set of thermistor data becomes available from 2 packets (16 seconds) of JMR radiometer count measurements. A time-ordered table of the thermistor temperatures and associated quality flags is then generated by "data management and control" algorithms, where the time tag assigned to the set of thermistor data corresponds to the last second of the 16 seconds of data that compose the set of the thermistor data. The resulting table is queried to determine the set of thermistor temperatures and associated quality flags to assign to a once per second JMR measurement. The thermistor data assigned to the once per second measurement is that which is closest in time to the once per second measurement, and which satisfies the following criteria:

- The absolute value of the time difference between the time tag on the thermistor data and the once per second measurement must be less than DT_TEMP seconds.
- All 16 associated thermistor temperature quality flags must indicate that no threshold limits were exceeded by the thermistor temperatures.

ALGORITHM SPECIFICATION

Input Data

• Table of time tags assigned to each set of 16 thermistor temperatures	TTHERM[0:NTHERM-1]	(sec)
• Table of 16 thermistor temperatures	T[0:NTHERM-1][1:16]	(Kelvin)
• Table of thermistor temperature quality flags(1)	TQUAL[0:NTHERM-1]	(/)
• Time tag of once per second JMR measurement	TSSALTO	(sec)
• Time difference threshold	DT_TEMP	(sec)

Output Data

• 16 thermistor temperatures assigned to JMR measurement	TH[1:16]	(Kelvin)
• Assigned thermistor data flag(2)	TH_FLAG	(/)
• Execution status		

Note: The thermistor temperatures, T[0:NTHERM-1][1:16], each bit (from least significant to most significant) of the 16 bit thermistor temperature quality flag, TQUAL, and the assigned thermistor temperatures, TH[1:16], are all identically ordered in the sequence that corresponds to REFi with i ∈ [1:4], NSRC1, NSRC2, FH1, FH2, WG11, WG12, WG21, WG22, WG31, WG32, WG41, WG42 (see Figure 3).

Processing

- Initialize assigned thermistor data flag to indicate that no set of thermistor data was assigned to the once per second measurement.
  - Set TH_FLAG = 1
  - Set all TH[1:16] = 0
- Loop through the table of thermistor data to save the indices of those data that have time tags within the time window with half width DT_TEMP of the once per second measurement, in the order of closest to farthest.

---

(1) Bit m of the 16 bit integer TQUAL is set to 0 if the temperature of thermistor (m+1) is within its respective threshold, and 1 otherwise. Therefore, the 16 bit integer TQUAL has a value of 0 when all 16 thermistor temperatures are within their respective thresholds.

(2) The flag TH_FLAG is set to a value of 0 if a valid set of thermistor data was assigned to the once per second measurement, and set to a value of 1 otherwise.

- Initialize a counter kmax = 0
- For each n = 0 to NTHERM-1
  - * If |TTHERM(n) - TSSALTO| ≤ DT_TEMP then

    If kmax = 0 then save first index and time difference

    ns(kmax) = n

    dt(kmax) = |TTHERM(n) - TSSALTO|

    else If |TTHERM(n) - TSSALTO| > dt(kmax-1) then insert index to end of list of saved indices

    ns(kmax) = n

    dt(kmax) = |TTHERM(n) - TSSALTO|

    Otherwise insert index into list of increasing time differences

    For each k = 0 to kmax-1

    If |TTHERM(n) - TSSALTO| < dt(k) then

    kk = k

    Exit loop in k

    For each k = (kmax-1) to kk (decreasing order)

    ns(k+1) = ns(k)

    dt(k+1) = dt(k)

    ns(kk) = n

    dt(kk) = |TTHERM(n) - TSSALTO|

    kmax = kmax + 1

- If kmax = 0, then no thermistor temperature data available within the time window
  - Set execution status to “Bad”
- Otherwise find first set of valid thermistor data
  - For n = 0 to (kmax-1)

* If TQUAL(ns(n)) = 0

  Assign the thermistor temperatures as TH[1:16] = T[ns(n)][1:16]

  Set flag to indicate thermistor data assigned, TH_FLAG = 0

  Set execution status to “Good” and exit algorithm

- If TH_FLAG = 1
  - Set execution status to “Bad”

Accuracy

None

Comments

None

References

None

JMR_NDIODE_01 - TO CORRECT THE NOISE DIODE TEMPERATURES
DEFINITION, ACCURACY AND SPECIFICATION

Prepared by:	J. Stum (CNES) S. Desai (JPL) A. Kitiyakara (JPL) T. Pavlovitch (JPL)	
Approved by:	P. Vincent (CNES) B. Haines (JPL)	

Document Ref: SMM-ST-M1-EA-11577-CN		Issue: 3	Update: 0

Algorithm change record	creation	date:	Issue:	Update:
	CCM			

FUNCTION

This function corrects the pre-launch values of the noise diode temperatures for the physical temperature of the noise source.

ALGORITHM DEFINITION

Input Data

- Product data: None
- Computed data:
  - From "JMR_GETTRM_01 - To assign the thermistor temperatures to the once per second JMR measurements"
    - Noise source 1 and 2 thermistor temperatures, TNSRC1 and TNSRC2, assigned to the once per second JMR measurement, where TNSRC1 = TH(5) and TNSRC2 = TH(6).
    - Flag indicating if valid set of thermistor data assigned to once per second data, TH_FLAG
- Dynamic auxiliary data: None
- Static auxiliary data:
  - Noise source number to use to correct the noise diode, NSRCNUM (see **Table 1**)
  - Calibration coefficients to correct noise diode temperatures, KTN[0:3][1:4][1:3]. These calibration coefficients represent a third order polynomial ($n \in [0:3]$), for each channel $i \in [1:4]$, and each noise diode $j \in [1:3]$.

Output Data

- Corrected noise diode temperatures, TN_COR[1:4][1:3], for each channel $i \in [1:4]$, and noise diode $j \in [1:3]$.

Mathematical Statement

The JMR has a single noise source, which is monitored by thermistors NSRC1 and NSRC2. This noise source essentially generates three noise diode signals. Ground calibration of the JMR instrument provides pre-launch values of the noise diode temperatures. However, the noise diode temperatures change slightly with changes in the physical temperatures of the noise source and the noise diode temperatures must be corrected to take into account the physical temperatures of the thermistors NSRC1 and NSRC2. Each noise diode therefore requires a set of calibration coefficients for each of the four radiometer channels.

It should be noted that the counts NSRC1 and NSRC2 essentially measure the same temperature and tests have revealed that the temperatures of NSRC1 and NSRC2 track each other to within 0.1 Kelvin. Therefore, in this particular algorithm the temperature of only one of the two noise source thermistor temperatures is used to compute the corrected noise diode temperature. The parameter NSRCNUM provided in the JMR static auxiliary data file (see **Table 1**) specifies which noise source temperature to use to correct the noise diode temperature. Nominally this will be noise source NSRC1. Corrected noise diode temperatures are computed as follows:

$$\mathrm{TN\_COR}(i,j) = \mathrm{KTN}(0,i,j) + \mathrm{KTN}(1,i,j) \times \mathrm{TNSRC} + \mathrm{KTN}(2,i,j) \times \mathrm{TNSRC}^2 + \mathrm{KTN}(3,i,j) \times \mathrm{TNSRC}^3 \qquad (1)$$

where TNSRC is the temperature of the noise source thermistor that is specified by NSRCNUM, and is also the temperature that is assigned to the particular once per second JMR measurement. If no thermistor data is assigned to a once per second measurement then the corrected noise diode temperature, TN_COR(i,j), is simply set to the coefficient KTN(0,i,j).

If one of the noise diodes on the JMR instrument fails, then the JMR instrument can be commanded (from the ground) to substitute the failed noise diode with one of the three remaining noise diodes. In such a scenario, the calibration coefficients KTN that correspond to the failed noise diode would have to be changed, with those calibration coefficients that correspond to the failed noise diode being replaced with the calibration coefficients that correspond to the noise diode that was commanded to substitute the failed noise diode. Such a change in the noise diode configuration on the JMR instrument would necessarily require a new JMR static auxiliary file that corresponds to the time interval where the noise diode substitution occurred.

ALGORITHM SPECIFICATION

Input Data

- Thermistor temperature of noise source 1 (NSRC1) — TNSRC1 — (Kelvin)
- Thermistor temperature of noise source 2 (NSRC2) — TNSRC2 — (Kelvin)
- Assigned thermistor data flag[(1)] — TH_FLAG — (/)
- Flag indicating which noise source temperature to use[(2)] — NSRCNUM — (/)
- Calibration coefficients (order 0) — KTN[0][1:4][1:3] — (Kelvin)
- Calibration coefficients (order 1) — KTN[1][1:4][1:3] — (/)

---

(1) The flag TH_FLAG is set to a value of 0 if a valid set of thermistor data was assigned to the once per second measurement, and set to a value of 1 otherwise.

(2) NSRCNUM has a value 1 if the temperature of NSRC1 is used to correct the noise diode temperature, and a value of 2 is the temperature of NSRC2 is used to correct the noise diode temperature.

- Calibration coefficients (order 2) KTN[2][1:4][1:3] ($Kelvin^{-1}$)
- Calibration coefficients (order 3) KTN[3][1:4][1:3] ($Kelvin^{-2}$)

Output Data

- Corrected values of noise diode temperatures TN_COR[1:4][1:3] (Kelvin)
- Execution status

NOTE: TNSRC1 and TNSRC2 are output from "JMR_GETTRM_01 - To assign the thermistor temperatures to the once per second JMR measurements" as TH(5) and TH(6), respectively.

Processing

- Choose the noise source thermistor temperature to correct the noise diode temperature
  - If NSRCNUM = 2 then TNSRC = TNSRC2
  - otherwise TNSRC = TNSRC1
- For each channel i ∈ [1:4], and noise diode j ∈ [1:3], compute the corrected noise diode temperatures from:
  - If TH_FLAG = 1 (no valid thermistor data were assigned to once per second data) then
    - TN_COR(i,j) = KTN(0,i,j)
    - Set execution status to "bad"
  - otherwise (use noise source temperature from assigned set of thermistor data):
    - $$\mathrm{TN\_COR}(i,j) = \mathrm{KTN}(0,i,j) + \mathrm{KTN}(1,i,j) \times \mathrm{TNSRC} + \mathrm{KTN}(2,i,j) \times \mathrm{TNSRC}^2 + \mathrm{KTN}(3,i,j) \times \mathrm{TNSRC}^3 \quad (1)$$
    - Set execution status to "good"

Accuracy

None

Comments

None

References

None

JMR_ZEROOFF_01 - TO COMPUTE THE ZERO OFFSETS FOR MODE 1
DEFINITION, ACCURACY AND SPECIFICATION

Prepared by:	J. Stum (CNES) S. Desai (JPL) A. Kitiyakara (JPL) T. Pavlovitch (JPL)	
Approved by:	P. Vincent (CNES) B. Haines (JPL)	

Document Ref: SMM-ST-M1-EA-11577-CN		Issue: 3	Update: 0

Algorithm change record	creation	date:	Issue:	Update:
	CCM			

FUNCTION

This function computes the zero offsets from the first line of the mode 1 calibration sequence.

ALGORITHM DEFINITION

Input Data

- Product data:
  - For a particular once per second measurement denoted by index k
    - Command word, CMD
    - Status word 1, STATk1
- Computed data for a once per second measurement
  - From "JMR_VALRAD_01 - To check the validity of the elementary radiometer measurements"
    - 36 validity flags associated with each of the 36 normalized radiometer count measurements in a once per second JMR measurement
  - From "JMR_NRMCNT_01 - To renormalize the radiometer counts"
    - 36 normalized counts, RN_{ij}, NN_{ij}, and SN_{ij} for each channel $i \in [1:4]$, and noise diode $j \in [1:3]$
- Dynamic auxiliary data: None
- Static auxiliary data: None

Output Data

- Zero offsets, Z_i for each of four channels ($i \in [1:4]$)
- Associated quality flags of the zero offsets

Mathematical Statement

The JMR is operating in mode 1 when bit 11 of the command word CMD is equal to 0. Mode 1 operation of the JMR requires a calibration measurement that lasts 2 seconds. The first second of the 2-second calibration sequence is used to generate a zero offset measurement for each channel and the subsequent second is used to generate a system noise temperature measurement for each noise diode on each channel. Both the zero offset and the system noise temperature are typically stable. Therefore, it is sufficient to measure them approximately once every 10 minutes, and the 2-second calibration sequences are spaced at approximately 10-minute intervals. Note that the antenna temperature measurements are interrupted in Mode 1 operation whenever the 2-second calibration sequence occurs.

When the JMR is in mode 1 operation (bit 11 of CMD = 0), then bit 14 of the word STATk1 has a value of 0 when the JMR is in a data acquisition mode, and a value of 1 when the JMR is in a calibration sequence mode. Furthermore, when the JMR is in a mode 1 calibration sequence (bit 11 of CMD = 0, AND bit 14 of STATk1 = 1), then bit 0 of STATk1 is equal to 0 when the JMR is in the first second of the 2-second calibration sequence and is generating the zero offset measurement, and bit 0 of STATk1 is equal to 1 when the JMR is in the second of the 2-second calibration sequence and is generating the system noise temperatures.

The zero offset is calculated from the first line of the calibration sequence. The zero offset is dependent on the channel but independent of the noise diode. The zero offset measurements lie in the source packet bins that are usually assigned to the radiometer count measurements from the first noise diode, R_{i1}, N_{i1}, and S_{i1}. Data bins usually assigned to R_{ij}, N_{ij}, and S_{ij} with $j \neq 1$ are simply ignored in the first second of the mode 1 calibration sequence. The counts in the data bins R_{i1}, N_{i1}, and S_{i1} must be renormalized with their respective reference count values before computing a single zero offset measurement. Note that the three zero offset measurements provided in the data bins R_{i1}, N_{i1}, and S_{i1} are simply consecutive measurements of the same quantity. These three renormalized measurements of the zero offset are averaged to provide a single zero offset measurement. Z_i, for each channel i ($i \in [1:4]$):

$$Z_i = \frac{RN_{i1} + NN_{i1} + SN_{i1}}{3} \quad (1)$$

where RN_{i1}, NN_{i1}, and SN_{i1} are the renormalized measurements of the zero offsets. This function also returns validity flags for the zero offsets, where an invalid zero offset is computed when any one of the three zero offsets specified by R_{i1}, N_{i1}, and S_{i1} is invalid.

Note that the 4 zero offset measurements computed by this algorithm are generated by the JMR only once approximately every 10 minutes. Implicit in the definition of the mode 1 calibration sequence is the fact that the single second of data corresponding to a zero offset measurement is immediately followed by a single second of data that provides the system noise temperature measurements. System noise temperatures are computed in “JMR_SYSTEMP_01 - To compute the system noise temperatures for mode 1”. As such, computation of the zero offsets should be immediately followed by the computation of the system noise temperatures. These two measurements along with their respective quality flags, form a single set of mode 1 calibration data. This set of mode 1 calibration data is then assigned a time tag that corresponds to the time tag of the system noise temperature measurements, and a table of the mode 1 calibration data is generated. Assigning the time tag to the mode 1 calibration data and the generation of the tables of mode 1 calibration data will be considered to be part of “data management and control algorithms”.

ALGORITHM SPECIFICATION

Input Data

- Command word CMD (/)
- Status word 1 STAT1 (/)

- 12 validity flags for normalized reference load counts(1) VFR[1:4][1:3] (/)
- 12 validity flags for normalized signal counts with noise diode on(1) VFN[1:4][1:3] (/)
- 12 validity flags for normalized signal counts with noise diode off(1) VFS[1:4][1:3] (/)
- Normalized reference load counts RN[1:4][1:3] (/)
- Normalized signal counts with noise diode on NN[1:4][1:3] (/)
- Normalized signal counts with noise diode off SN[1:4][1:3] (/)

Output Data

- 4 zero offsets Z[1:4] (/)
- 4 zero offset quality flags(2) Z_FLAG[1:4] (/)
- Execution status

Processing

- Initialize zero offset values and flags to "bad" values
  - Set all Z[1:4] = 0
  - Set all Z_FLAG[1:4] = 1
- Verify that the once per second data corresponds to the zero offset measurement (line 1) of the JMR mode 1 calibration sequence
  - If bit 11 of CMD = 0, AND bit 14 of status word STAT1 = 1, AND bit 0 of status word STAT1 = 0 then
    - * For each channel i ∈ [1:4]

      If VFR(i,1) = 0, AND VFN(i,1) = 0, AND VFS(i,1) = 0, then compute zero offset and set quality flag to "good"

---

(1) The validity flags VFR(i,j), VFN(i,j), and VFS(i,j) are set to a value of 1 if the corresponding normalized radiometer count has the default value, DEFCNT, and set to 0 otherwise.

(2) The zero offset quality flags Z_FLAG[1:4] are set to a value of 0 if the zero offsets are considered to be valid, and 1 if they are invalid.

$$Z(i) = \frac{RN(i,1) + NN(i,1) + SN(i,1)}{3} \quad (1)$$

Z_FLAG(i) = 0

- If Z_FLAG(i) = 1 for all channels i, then
  - Set execution status to “Bad”
- Otherwise
  - Set execution status to “Good”

Accuracy

None

Comments

None

References

- Kitiyakara, A., JMR Mode 1 TA retrieval, JPL Interoffice Memorandum, October 14, 1998.

JMR_SYSTEMP_01 - TO COMPUTE THE SYSTEM NOISE TEMPERATURES FOR MODE 1

DEFINITION, ACCURACY AND SPECIFICATION

Prepared by:	J. Stum (CNES) S. Desai (JPL) A. Kitiyakara (JPL) T. Pavlovitch (JPL)	
Approved by:	P. Vincent (CNES) B. Haines (JPL)	

Document Ref: SMM-ST-M1-EA-11577-CN		Issue: 3	Update: 0

Algorithm change record	creation	date:	Issue:	Update:
	CCM			

FUNCTION

This function computes the system noise temperatures from the second line of the mode 1 calibration sequence.

ALGORITHM DEFINITION

Input Data

- Product data:
  - For a particular once per second measurement denoted by index k
    - Command word, CMD
    - Status word 1, STATk1
- Computed data:
  - From "JMR_VALRAD_01 - To check the validity of the elementary radiometer measurements"
    - 36 validity flags associated with each of the 36 normalized radiometer count measurements in a once per second JMR measurement
  - From "JMR_NRMCNT_01 - To renormalize the radiometer counts"
    - 36 normalized counts, RN_{ij}, NN_{ij}, and SN_{ij} for each channel $i \in [1:4]$, and noise diode $j \in [1:3]$
  - From "JMR_GETTRM_01 - To assign the thermistor temperatures to the once per second JMR measurements"
    - Reference load thermistor temperatures, $TREF_i$ ($i \in [1:4]$) that are assigned to the once per second JMR measurement, where TREF(i) = TH(i), with $i \in [1:4]$
    - Flag indicating if valid set of thermistor data assigned to once per second data, TH_FLAG
  - From "JMR_NDIODE_01 - To correct the noise diode temperatures"
    - Corrected noise diode temperatures, TN_COR[1:4][1:3], for each channel $i \in [1:4]$, and noise diode $j \in [1:3]$.
  - From "JMR_ZEROOFF_01 - To compute the zero offsets for mode 1"
    - Zero offsets, Z_i for each of four channels ($i \in [1:4]$)
    - Associated quality flags of the zero offsets
- Dynamic auxiliary data: None

- Static auxiliary data:
  - Minimum tolerance for division by counts, MINTOLCNT (see **Table 1**)
  - Path loss coefficients, L_i for each channel i (i ∈ [1:4]) (see **Table 1**)

Output Data

- System noise temperatures, TS_{ij} for each of four channels (i ∈ [1:4]) and three noise diodes (j ∈ [1:3])
- Associated quality flags of the system noise temperatures

Mathematical Statement

Refer to the mathematical statement of “JMR_ZEROOFF_01 - To compute the zero offsets for mode 1” for more details about the JMR mode 1 calibration sequence, and for a description of how a table of mode 1 calibration data is generated.

The second line of the calibration sequence contains the R_{ij}, N_{ij}, and S_{ij} counts that are used to calculate the system noise temperature for each noise diode in each channel. Here again, the counts must be renormalized with their respective reference count values before generating the system noise temperature TS_{ij} for each channel i (i ∈ [1:4]) and noise diode j (j ∈ [1:3]) as follows:

$$TS_{ij} = \frac{TN\_COR_{ij}}{L_i} \bullet \frac{RN_{ij} - Z_i}{NN_{ij} - SN_{ij}} - T\_REF_i \qquad (1)$$

where TN_COR[1:4][1:3] are the corrected noise diode temperatures, T_REF[1:4] are the reference load thermistor temperatures, and L[1:4] are path loss constants specified in the instrument characterization file. In this case RN[1:4][1:3], NN[1:4][1:3], and SN[1:4][1:3] are the renormalized reference load counts, antenna signal counts with noise diode on, and antenna signal counts with noise diode off, respectively.

The function also returns validity flags for the system noise temperatures, where an invalid system noise temperature is indicated when any one of the radiometer counts R_{ij}, N_{ij}, and S_{ij} is invalid, or if thermistor data could not be assigned to the calibration data.

ALGORITHM SPECIFICATION

Input Data

• Command word	CMD	(/)
• Status word 1	STAT1	(/)

• 12 validity flags for normalized reference load counts[(1)]	VFR[1:4][1:3]	(/)
• 12 validity flags for normalized signal counts with noise diode on[(1)]	VFN[1:4][1:3]	(/)
• 12 validity flags for normalized signal counts with noise diode off[(1)]	VFS[1:4][1:3]	(/)
• Normalized reference load counts	RN[1:4][1:3]	(/)
• Normalized signal counts with noise diode on	NN[1:4][1:3]	(/)
• Normalized signal counts with noise diode off	SN[1:4][1:3]	(/)
• Reference load thermistor temperatures assigned to measurement	T_REF[1:4]	(Kelvin)
• Assigned thermistor data flag[(2)]	TH_FLAG	(/)
• Corrected values of noise diode temperatures	TN_COR[1:4][1:3]	(Kelvin)
• 4 zero offsets	Z[1:4]	(/)
• 4 zero offset quality flags[(3)]	Z_FLAG[1:4]	(/)
• Tolerance for division by counts	MINTOLCNT	(/)
• Path loss coefficients	L[1:4]	(/)

Output Data

• 12 system noise temperatures	TS[1:4][1:3]	(Kelvin)
• 12 associated quality flags[(4)]	TS_FLAG[1:4][1:3]	(/)
• Execution status		

---

(1) The validity flags VFR(i,j), VFN(i,j), and VFS(i,j) are set to a value of 1 if the corresponding normalized radiometer count has the default value, DEFCNT, and set to 0 otherwise.

(2) The flag TH_FLAG is set to a value of 0 if a valid set of thermistor data was assigned to the once per second measurement, and set to a value of 1 otherwise.

(3) The zero offset quality flags Z_FLAG[1:4] are set to a value of 0 if the zero offsets are considered to be valid, and 1 if they are invalid.

(4) The system noise temperature quality flags, TS_FLAG[1:4][1:3] are set to a value of 0 if the system noise temperatures are considered to be valid, and 1 if they are invalid.

NOTE: TREF[1:4] are the output as TH(i), with i $\in$ [1:4] from “JMR_GETTRM_01 - To assign the thermistor temperatures to the once per second JMR measurements”.

Processing

- Initialize the system noise temperatures and quality flags to their “bad” values.
  - Set all TS[1:4][1:3] = 0
  - Set all TS_FLAG[1:4][1:3] = 1
- Verify that the once per second data correspond to line 2 of the JMR mode 1 calibration sequence, and that valid thermistor data have been assigned to the calibration sequence.
  - If bit 11 of CMD = 0, AND bit 14 of status word STAT1 = 1, AND bit 0 of status word STAT1 = 1, AND TH_FLAG = 0, then
    - * For each channel i $\in$ [1:4]

      If Z_FLAG(i) = 0, AND L[i] $\neq$ 0 then

      For each noise diode j $\in$ [1:3]

      If VFR(i,j) = 0, AND VFN(i,j) = 0, AND VFS(i,j) = 0, NN(i,j) > 0, AND (NN(i,j) - SN(i,j)) > MINTOLCNT then

$$TS(i,j) = \frac{TN\_COR(i,j)}{L(i)} \bullet \frac{RN(i,j) - Z(i)}{NN(i,j) - SN(i,j)} - T\_REF(i) \quad (1)$$

      Set TS_FLAG(i,j) = 0

- If all TS_FLAG(i,j) = 1 (i $\in$ [1:4], j $\in$ [1:3]) then
  - Set execution status to “Bad”
- Otherwise
  - Set execution status to “Good”

Accuracy

None

Comments

None

References

- Kitiyakara, A., JMR Mode 1 TA retrieval, JPL Interoffice Memorandum, October 14, 1998.

JMR_ASCAL_01 - TO ASSIGN THE ZERO OFFSETS AND THE SYSTEM NOISE TEMPERATURES TO THE ONCE PER SECOND JMR MEASUREMENTS FOR MODE 1

DEFINITION, ACCURACY AND SPECIFICATION

Prepared by:	J. Stum (CNES) S. Desai (JPL) A. Kitiyakara (JPL) T. Pavlovitch (JPL)	
Approved by:	P. Vincent (CNES) B. Haines (JPL)	

Document Ref: SMM-ST-M1-EA-11577-CN		Issue: 3	Update: 0

Algorithm change record	creation	date:	Issue:	Update:
	CCM			

FUNCTION

This function assigns the zero offsets and the system noise temperatures to each once per second JMR measurement.

ALGORITHM DEFINITION

Input Data

- Product data:
  - For a particular once per second measurement denoted by index k
    - * Command word, CMD
    - * Status word 1, STATk1
- Computed data:
  - From “JMR_TME_01 - To determine TAI time of the JMR measurements”
    - * Time tags corresponding to the middle of the 1-s averaged JMR measurement
  - From “JMR_ACTCH_01 - To determine the active 23.8 GHz Channel”
    - * Flag indicating the active 23.8GHz channel
  - Table in time of 4 zero offsets and 12 system noise temperatures and their respective quality flags
- Dynamic auxiliary data: None
- Static auxiliary data:
  - Time difference threshold DT_CAL1 (see **Table 1**)

Output Data

- 4 zero offsets and 12 system noise temperatures and associated quality flags that are assigned to the once per second JMR measurement
- Flag indicating if valid set of mode 1 calibration data assigned to once per second data, CAL1_FLAG

Mathematical Statement

The algorithms "JMR_ZEROOFF_01 - To compute the zero offsets for mode 1" and "JMR_SYSTEMP_01 - To compute the system noise temperatures for mode 1" compute the mode 1 calibration data (zero offsets and system noise temperatures) approximately once every 10 minutes. A time tag is assigned to the mode 1 calibration data and a time-ordered table of these data is then generated by "data management and control algorithms". This algorithm queries the time ordered table to determine the set of mode 1 calibration data to assign to a once per second JMR measurement. The set of calibration data assigned to the once per second measurement is that which is closest in time to the once per second measurement, and which satisfies the following criteria:

- The absolute value of the time difference between the time tag on the calibration data and the once per second measurement must be less than DT_CAL1 seconds.
- All associated calibration data quality flags must indicate that the calibration data are valid.

ALGORITHM SPECIFICATION

Input Data

• Command word	CMD	(/)
• Status word 1	STAT1	(/)
• Flag denoting active 23.8 GHz channel[(1)]	ACT238	(/)
• Table of time tags assigned to the mode 1 calibration data	TCAL[0:NCAL-1]	(sec)
• Table of 4 zero offsets	Z[0:NCAL-1][1:4]	(/)
• Table of 4 zero offset quality flags[(2)]	Z_FLAG[0:NCAL-1][1:4]	(/)
• Table of 12 system noise temperatures	TS[0:NCAL-1][1:4][1:3]	(Kelvin)
• Table of 12 system noise temperature quality flags[(3)]	TS_FLAG[0:NCAL-1][1:4][1:3]	(/)
• Time tag of once per second JMR measurement	TSSALTO	(sec)

(1) ACT238 has a value of 0 if both 23.8 GHz channels (channels 2 and 3) are inactive, 2 if channel 2 is the only active 23.8 GHz channel, 3 if channel 3 is the only active 23.8 GHz channel, and 5 if channels 2 and 3 are both active.

(2) The zero offset quality flags Z_FLAG[1:4] are set to a value of 0 if the zero offsets are considered to be valid, and 1 if they are invalid.

(3) The system noise temperature quality flags, TS_FLAG[1:4][1:3] are set to a value of 0 if the system noise temperatures are considered to be valid, and 1 if they are invalid.

- Time difference threshold DT_CAL1 (sec)

Output Data

- 4 zero offsets assigned to JMR measurement ZA[1:4] (/)
- 12 system noise temperatures assigned to JMR measurement TSA[1:4][1:3] (/)
- Assigned mode 1 calibration data flag[4] CAL1_FLAG (/)
- Execution status

Processing

- Initialize assigned mode 1 calibration data flag to indicate that no set of mode 1 calibration data was assigned to the once per second measurement.
  - Set CAL1_FLAG = 1
  - Set all ZA[1:4] = 0
  - Set all TSA[1:4][1:3] = 0
  - Initialize a counter kmax = 0
- Confirm that the JMR is in mode 1 data acquisition mode
  - If bit 11 of CMD = 0, AND bit 14 of STAT1 = 0
    - * Loop through the table of calibration data to save the indices of those data that have time tags within the time window with half width DT_CAL1 of the once per second measurement, in the order of closest to farthest.
    - * For each n = 0 to NCAL-1

      If $|TCAL(n) - TSSALTO| \leq DT\_CAL1$ then

      If kmax = 0 then save first index and time difference

      ns(kmax) = n

      dt(kmax) = $|TCAL(n) - TSSALTO|$

---

[4] The flag CAL1_FLAG is set to a value of 0 if a valid set of mode 1 calibration data was assigned to the once per second measurement, and set to a value of 1 otherwise.

else If |TCAL(n) - TSSALTO| > dt(kmax-1) then insert index to end of list of saved indices

ns(kmax) = n

dt(kmax) = |TCAL(n) - TSSALTO|

Otherwise insert index into list of increasing time differences

For each k = 0 to kmax-1

If |TCAL(n) - TSSALTO| < dt(k) then

kk = k

Exit loop in k

For each k = (kmax-1) to kk (decreasing order)

ns(k+1) = ns(k)

dt(k+1) = dt(k)

ns(kk) = n

dt(kk) = |TCAL(n) - TSSALTO|

kmax = kmax + 1

- If kmax = 0, then no calibration data available within the time window
  - Set execution status to "Bad", exit algorithm
- Otherwise find first set of valid calibration data
  - For n = 0 to (kmax-1)
    - * First check that all zero offsets and system noise temperatures for the active channels are valid

For each channel i $\in$ [1:4]

If i = 2 or 3, AND ACT238 $\neq$ i or 5, then

Skip to next channel, i

If Z_FLAG(ns(n),i) = 1 then

Exit loop in i, and skip to next n

For each noise diode j ∈ [1:3]

If TS_FLAG(ns(n),i,j) = 1 then

Exit loop in i, and skip to next n

* Assign set that has all valid zero offsets and system noise temperatures for the active channels

CAL1_FLAG = 0

For each channel i ∈ [1:4]

If i = 2 or 3, AND ACT238 ≠ i or 5, then

Skip to next channel, i

ZA(i) = Z(ns(n), i)

For each noise diode j ∈ [1:3]

TSA(i,j) = TS(ns(n),i,j)

* Set execution status to “Good”, and exit algorithm

- If CAL1_FLAG = 1 then
  - Set execution status to “Bad”

Accuracy

None

Comments

None

References

None

JMR_TAMODE1_01 - TO COMPUTE THE 330 MS ANTENNA TEMPERATURES IN MODE 1

DEFINITION, ACCURACY AND SPECIFICATION

Prepared by:	J. Stum (CNES) S. Desai (JPL) A. Kitiyakara (JPL) T. Pavlovitch (JPL)	
Approved by:	P. Vincent (CNES) B. Haines (JPL)	

Document Ref: SMM-ST-M1-EA-11577-CN		Issue: 3	Update: 0

Algorithm change record	creation	date:	Issue:	Update:
	CCM			

FUNCTION

This function computes the individual 330 ms antenna temperatures when the JMR mode is mode 1.

ALGORITHM DEFINITION

Input Data

- Product data:
  - For a particular once per second measurement denoted by index k
    - Command word, CMD
    - Status word 1, STATk1
- Computed data:
  - From "JMR_VALRAD_01 - To check the validity of the elementary radiometer measurements"
    - 24 validity flags associated with each of the 24 normalized radiometer count measurements in a once per second JMR measurement
  - From "JMR_ACTCH_01 - To determine the active 23.8 GHz Channel"
    - Flag indicating the active 23.8GHz channel
  - From "JMR_NRMCNT_01 - To renormalize the radiometer counts"
    - 24 normalized counts, RN_{ij}, and NN_{ij} for each channel $i \in [1:4]$, and noise diode $j \in [1:3]$
  - From "JMR_GETTRM_01 - To assign the thermistor temperatures to the once per second JMR measurements"
    - 16 thermistor temperatures that are assigned to the once per second JMR measurement
    - Flag indicating if valid set of thermistor data assigned to once per second data, TH_FLAG
  - From "JMR_NDIODE_01 - To correct the noise diode temperatures"
    - Corrected noise diode temperatures, TN_COR[1:4][1:3], for each channel $i \in [1:4]$, and noise diode $j \in [1:3]$.
  - From "JMR_ASCAL_01 - To assign the zero offsets and the system noise temperatures to the once per second JMR measurements for mode 1"
    - 4 zero offsets that are assigned to the once per second JMR measurement

  * 12 system noise temperatures that are assigned to the once per second JMR measurement

  * Flag indicating if valid set of mode 1 calibration data assigned to once per second data, CAL1_FLAG

- Dynamic auxiliary data: None
- Static auxiliary data:
  - Minimum tolerance for division by counts, MINTOLCNT (see **Table 1**)
  - Path loss coefficients, L[1:4] (see **Table 1**)
  - Waveguide number, WGNUM (see **Table 1**)

Output Data

- 12 individual 330 ms antenna temperatures (for each of four channels and each of three noise diodes)
- 12 associated antenna temperature quality flags

Mathematical Statement

When the JMR is in mode 1 data acquisition, zero offsets, system noise temperatures, and thermistor temperatures are required to compute antenna temperatures. Three individual 330 ms antenna temperature measurements are performed in each channel to provide a single once per second antenna temperature measurement. Each of the 330 ms measurements is derived by a Dicke cycle that consists of a 5ms reference load measurement, and a 5 ms antenna measurement with noise diode on. These are individually accumulated into 165 ms measurements each on the reference load and on the antenna with noise diode on, which together provide a single 330 ms antenna temperature measurements in each channel. The three 330 ms antenna temperature measurements are on three different noise diodes, and these three 330 ms measurements are always averaged to provide a single once per second antenna temperature measurement. Combining the three 330ms measurements into a single once per second antenna temperature serves to reduce the uncorrelated noise in the measurement by a factor of $(3)^{1/2}$. Note that in mode 1 operation no measurements are performed on the antenna with noise diode off and the data bins S_{ij} are always equal to zero.

The antenna temperature TA(i,j) in each active channel i ($i \in [1:4]$), and for each noise diode j ($j \in [1:3]$) is computed as follows:

$$TA(i,j) = \frac{NN_{ij} - RN_{ij}}{RN_{ij} - Z_i} \bullet L(i) \bullet \left[T\_REF(i) + TSA(i,j)\right] + L(i) \bullet \left[T\_REF(i) - T\_WG(i)\right] - TN\_COR(i,j) + T\_WG(i) \qquad (1)$$

where RN_{ij} and NN_{ij} are the renormalized radiometer counts from the reference load and the antenna with noise diode on, respectively, L(i) is the path loss coefficient (see **Table 1**), TN_COR(i,j) is the corrected noise diode temperature, T_REF(i) and T_WG(i) represent channel dependent thermistor temperatures of the reference load and waveguide. The particular waveguide thermistor to be used in mode 1 is determined by the input parameter WGNUM that is set to 1 for waveguide 1, and 2 for waveguide 2. The zero offsets ZA(i) and system noise temperatures TSA(i,j) should have been generated from a mode 1 calibration sequence and already assigned to the particular once per second JMR measurement. The path loss coefficients L(i) are determined from thermal vacuum tests of the instrument prior to launch.

ALGORITHM SPECIFICATION

Input Data

• Command word	CMD	(/)
• Status word 1	STAT1	(/)
• 12 validity flags for normalized reference load counts(1)	VFR[1:4][1:3]	(/)
• 12 validity flags for normalized signal counts with noise diode on(1)	VFN[1:4][1:3]	(/)
• Flag denoting active 23.8 GHz channel(2)	ACT238	(/)
• Normalized reference load counts	RN[1:4][1:3]	(/)
• Normalized signal counts with noise diode on	NN[1:4][1:3]	(/)
• 16 thermistor temperatures assigned to JMR measurement	TH[1:16]	(Kelvin)
• Assigned thermistor data flag(3)	TH_FLAG	(/)
• Corrected values of noise diode temperatures	TN_COR[1:4][1:3]	(Kelvin)
• 4 zero offsets assigned to JMR measurement	ZA[1:4]	(/)
• 12 system noise temperatures assigned to JMR measurement	TSA[1:4][1:3]	(/)

---

(1) The validity flags VFR(i,j), and VFN(i,j) are set to a value of 1 if the corresponding normalized radiometer count has the default value, DEFCNT, and set to 0 otherwise.

(2) ACT238 has a value of 0 if both 23.8 GHz channels (channels 2 and 3) are inactive, 2 if channel 2 is the only active 23.8 GHz channel, 3 if channel 3 is the only active 23.8 GHz channel, and 5 if channels 2 and 3 are both active.

(3) The flag TH_FLAG is set to a value of 0 if a valid set of thermistor data was assigned to the once per second measurement, and set to a value of 1 otherwise.

- Assigned mode 1 calibration data flag(4) CAL1_FLAG (/)
- Tolerance for division by counts MINTOLCNT (/)
- Path loss coefficients L[1:4] (/)
- Waveguide number WGNUM (/)

Output Data

- Antenna temperatures for each channel and noise diode TA[1:4][1:3] (Kelvin)
- Antenna temperature quality flags(5) TA_FLAG[1:4][1:3] (/)
- Execution status

Note: The assigned thermistor temperatures, TH[1:16], are all identically ordered in the sequence that corresponds to REFi with i ∈ [1:4], NSRC1, NSRC2, FH1, FH2, WG11, WG12, WG21, WG22, WG31, WG32, WG41, and WG42 (see Figure 3).

Processing

- Initialize the antenna temperatures and associated quality flags to “bad” values.
  - Set all TA[1:4][1:3] = 0
  - Set all TA_FLAG[1:4][1:3] = 1
- Confirm that valid thermistor data and calibration data have been assigned to the once per second measurement, and that the instrument is in mode 1 data acquisition
  - If TH_FLAG = 1, OR CAL1_FLAG = 1, OR bit 11 of CMD ≠ 0, OR bit 14 of STAT1 ≠ 0 then
    - * Set execution status to “Bad” and exit algorithm
- For each channel i ∈ [1:4]
  - Check for inactive 23.8 GHz channel: if i = 2 or i = 3, AND ACT238 ≠ i or 5 then

(4) The flag CAL1_FLAG is set to a value of 0 if a valid set of mode 1 calibration data was assigned to the once per second measurement, and set to a value of 1 otherwise.

(5) The antenna temperature quality flags TA_FLAG[1:4][1:3] are set to a value of 0 for valid antenna temperatures and 1 for invalid antenna temperatures.

* Go to next channel, i

– Define the waveguide to use for the path loss using the input parameter WGNUM

* If WGNUM = 1 then NWG = 7

* else if WGNUM = 2 then NWG = 8

– Compute the loss term

* $$TL(i) = L(i) \bullet [TH(i) - TH((2 \times i) + NWG)] \quad (1)$$

– For each noise diode, j ∈ [1:3]

* If VFR(i,j) = 0 AND VFN(i,j) = 0 (all radiometer counts are valid) then

* If RN(i,j) ≤ 0 OR (RN(i,j) - Z(i)) ≤ MINTOLCNT then skip to next noise diode j

$$TA(i, j) = \frac{NN(i, j) - RN(i, j)}{RN(i, j) - Z(i)} \bullet L(i) \bullet [TH(i) + TSA(i, j)] + TL(i) - TN\_COR(i, j) + TH((2 \times i) + NWG) \quad (2)$$

Check that computed antenna temperature is positive

If TA(i,j) > 0 then set TA_FLAG(i,j) = 0

- Set execution status to “good”

Accuracy

None

Comments

From personal communication from A. Kitiyakara on August 23, 1999, the normalization by the factor (RN(i,j) - Z(i)) for the computation of antenna temperatures only requires Z(i) to be scaled by 1.5 if raw (unnormalized) counts are used. When normalized counts are used the scale factor is no longer needed.

References

- Kitiyakara, A., JMR Mode 1 TA retrieval, JPL Interoffice Memorandum, October 14, 1998.

JMR_TAMODE2_01 - TO COMPUTE THE 330 MS ANTENNA TEMPERATURES IN MODE 2

DEFINITION, ACCURACY AND SPECIFICATION

Prepared by:	J. Stum (CNES) S. Desai (JPL) A. Kitiyakara (JPL) T. Pavlovitch (JPL)	
Approved by:	P. Vincent (CNES) B. Haines (JPL)	

Document Ref: SMM-ST-M1-EA-11577-CN		Issue: 3	Update: 0

Algorithm change record	creation	date:	Issue:	Update:
	CCM			

FUNCTION

This function computes the individual 330 ms antenna temperatures when the JMR is operating in mode 2.

ALGORITHM DEFINITION

Input Data

- Product data:
  - For a once per second measurement
    - * The command word, CMD
- Computed data:
  - From “JMR_VALRAD_01 - To check the validity of the elementary radiometer measurements”
    - * 36 validity flags associated with each of the 36 normalized radiometer count measurements in a once per second JMR measurement
  - From “JMR_ACTCH_01 - To determine the active 23.8 GHz Channel”
    - * Flag indicating the active 23.8GHz channel
  - From “JMR_NRMCNT_01 - To renormalize the radiometer counts”
    - * 36 normalized counts, RN_{ij}, NN_{ij}, and SN_{ij} for each channel $i \in [1:4]$, and noise diode $j \in [1:3]$
  - From “JMR_GETTRM_01 - To assign the thermistor temperatures to the once per second JMR measurements”
    - * 16 thermistor temperatures that are assigned to the once per second JMR measurement
    - * Flag indicating if valid set of thermistor data assigned to once per second data, TH_FLAG
  - From “JMR_NDIODE_01 - To correct the noise diode temperatures”
    - * Corrected noise diode temperatures, TN_COR[1:4][1:3], for each channel $i \in [1:4]$, and noise diode $j \in [1:3]$.
- Dynamic auxiliary data: None
- Static auxiliary data:
  - Minimum tolerance for division by counts, MINTOLCNT (see **Table 1**)

- 4 reference load calibration coefficients, KR(i) for each channel i, ($i \in [1:4]$) (see **Table 1**)
- 4 waveguide calibration coefficients for each of sensors 1 and 2, KW1(i) and KW2(i) respectively, and each channel i ($i \in [1:4]$) (see **Table 1**)
- 4 feedhorn calibration coefficients for each of sensors 1 and 2, KF1(i) and KF2(i), respectively, and each channel i ($i \in [1:4]$) (see **Table 1**)

Output Data

- 12 individual 330 ms antenna temperatures (for each of four channels and each of three noise diodes)
- 12 associated antenna temperature quality flags

Mathematical Statement

The JMR is operating in mode 2 when bit 11 of the CMD word is equal to 1. Three individual 330 ms antenna temperature measurements are performed in each channel to provide a single once per second antenna temperature measurement. Each of the 330 ms measurements is derived by a Dicke cycle that consists of a 5ms reference load measurement, a 5 ms antenna measurement with noise diode on, and a 5 ms antenna temperature measurement with noise diode off. These are individually accumulated into 110 ms measurements each on the reference load, on the antenna with noise diode on, and on the antenna with noise diode off, which together provide a single 330 ms antenna temperature measurement in each channel. The three 330 ms antenna temperature measurements are on three difference noise diodes, and are always averaged to provide a single once per second antenna temperature measurement. Combining the three 330 ms measurements into a single once per second antenna temperature measurement serves to reduce the uncorrelated noise in the measurement by a factor of $(3)^{1/2}$.

The antenna temperature TA(i,j) in each active channel i ($i \in [1:4]$), and for each noise diode j ($j \in [1:3]$) is computed as follows:

$$TA(i,j) = TN\_COR(i,j) \times \frac{SN_{ij} - RN_{ij}}{NN_{ij} - SN_{ij}} + TLR(i) - TLWG(i) - TLFH(i) \quad (1)$$

where TN_COR(i,j) is the corrected noise diode temperature, and RN_{ij}, NN_{ij}, and SN_{ij} are the renormalized radiometer counts from the reference load, the antenna with noise diode on, and the antenna with noise diode off, respectively. TLR(i) is the channel dependent reference load loss term, TLWG(i) is the channel dependent waveguide loss term, and TLFH(i) is the channel dependent feedhorn loss term.

$$\begin{aligned} TLR(i) &= KR(i) \times T\_REF(i) \\ TLWG(i) &= KW1(i) \times T\_WG1(i) + KW2(i) \times T\_WG2(i) \\ TLFH(i) &= KF1(i) \times T\_FH1 + KF2(i) \times T\_FH2 \end{aligned} \quad (2)$$

Here, T_REF(i), T_WG1(i), and T_WG2(i) represent the channel dependent thermistor temperatures of the reference load, waveguide for sensor 1, and waveguide for sensor 2, respectively, and T_FH1 and T_FH2 are the feedhorn thermistor temperatures for sensors 1 and 2. The respective calibration coefficients KR(i), KW1(i), KW2(i), KF1(i), and KF2(i), are determined from thermal vacuum tests of the instrument prior to launch.

ALGORITHM SPECIFICATION

Input Data

• Command word	CMD	(/)
• 12 validity flags for normalized reference load counts(1)	VFR[1:4][1:3]	(/)
• 12 validity flags for normalized signal counts with noise diode on(1)	VFN[1:4][1:3]	(/)
• 12 validity flags for normalized signal counts with noise diode off(1)	VFS[1:4][1:3]	(/)
• Flag denoting active 23.8 GHz channel(2)	ACT238	(/)
• Normalized reference load counts	RN[1:4][1:3]	(/)
• Normalized signal counts with noise diode on	NN[1:4][1:3]	(/)
• Normalized signal counts with noise diode off	SN[1:4][1:3]	(/)
• 16 thermistor temperatures assigned to JMR measurement	TH[1:16]	(Kelvin)
• Assigned thermistor data flag(3)	TH_FLAG	(/)
• Corrected values of noise diode temperatures	TN_COR[1:4][1:3]	(Kelvin)
• Tolerance for division by counts	MINTOLCNT	(/)
• Reference load calibration coefficients	KR[1:4]	(/)
• Waveguide calibration coefficients for sensor 1	KW1[1:4]	(/)
• Waveguide calibration coefficients for sensor 2	KW2[1:4]	(/)
• Feedhorn calibration coefficient for sensor 1	KF1[1:4]	(/)

---

(1) The validity flags VFR(i,j), VFN(i,j), and VFS(i,j) are set to a value of 1 if the corresponding normalized radiometer count has the default value, DEFCNT, and set to 0 otherwise.

(2) ACT238 has a value of 0 if both 23.8 GHz channels (channels 2 and 3) are inactive, 2 if channel 2 is the only active 23.8 GHz channel, 3 if channel 3 is the only active 23.8 GHz channel, and 5 if channels 2 and 3 are both active.

(3) The flag TH_FLAG is set to a value of 0 if a valid set of thermistor data was assigned to the once per second measurement, and set to a value of 1 otherwise.

- Feedhorn calibration coefficient for sensor 2 KF2[1:4] (/)

Output Data

- Antenna temperatures for each channel and noise diode TA[1:4][1:3] (Kelvin)
- Antenna temperature quality flags[4] TA_FLAG[1:4][1:3] (/)
- Execution status

Note: The assigned thermistor temperatures, TH[1:16], are all identically ordered in the sequence that corresponds to REFi with i ∈ [1:4], NSRC1, NSRC2, FH1, FH2, WG11, WG12, WG21, WG22, WG31, WG32, WG41, and WG42 (see Figure 3).

Processing

- Initialize the antenna temperatures and associated quality flags to "bad" values.
  - Set all TA[1:4][1:3] = 0
  - Set all TA_FLAG[1:4][1:3] = 1
- Confirm that valid thermistor data have been assigned to the once per second measurement, and that the instrument is in mode w
  - If TH_FLAG = 1, OR bit 11 of CMD ≠ 1 then
    - * Set execution status to "Bad" and exit algorithm
- For each channel i ∈ [1:4]
  - Check for inactive 23.8 GHz channel: if i = 2 or i = 3, AND ACT238 ≠ i or 5 then
    - * Go to next channel, i
  - Compute the feedhorn loss term, TLFH(i), the reference load loss term, TLR(i), and the waveguide loss term TLWG(i)
    - * $TLFH(i) = KF1(i) \times TH(7) + KF2(i) \times TH(8)$ (1)
    - * $TLR(i) = KR(i) \times TH(i)$ (2)

---

[4] The antenna temperature quality flags TA_FLAG[1:4][1:3] are set to a value of 0 for valid antenna temperatures and 1 for invalid antenna temperatures.

* $\text{TLWG(i)} = \text{KW1(i)} \times \text{TH}((2 \times \text{i}) + 7) + \text{KW2(i)} \times \text{TH}((2 \times \text{i}) + 8)$ (3)

– For each noise diode j ∈ [1:3]

* If VFR(i,j) = 0, AND VFN(i,j) = 0, AND VFS(i,j) = 0 (all radiometer counts valid) then

* If NN(i,j) ≤ 0 OR (NN(i,j) - SN(i,j)) ≤ MINTOLCNT then skip to next noise diode, j

$$\text{TA(i, j)} = \text{TN\_COR(i, j)} \times \frac{\text{SN(i, j)} - \text{RN(i, j)}}{\text{NN(i, j)} - \text{SN(i, j)}} + \text{TLR(i)} - \text{TLWG(i)} - \text{TLFH(i)} \quad (4)$$

Check that computed antenna temperature is positive

If TA(i,j) > 0 then set TA_FLAG(i,j) = 0

- Set execution status to “good”

Accuracy

None

Comments

From personal communication with A. Kitiyakara on August 20, 1999, the feedhorn calibration coefficients are actually channel dependent, and therefore so is the feedhorn loss term.

References

- Kitiyakara, A., JASON Microwave Radiometer (JMR) T_A Retrieval, JPL Interoffice Memorandum, April 6, 1998, revised August 8, 1998.
- Kitiyakara, A., Addendum to Memo *JASON Microwave Radiometer (JMR) T_A Retrieval* dated April 6, 1998, JPL Interoffice Memorandum, April 17, 1998, revised August 8, 1998.

JMR_TAAVG_01 - TO AVERAGE THE ANTENNA TEMPERATURES
DEFINITION, ACCURACY AND SPECIFICATION

Prepared by:	J. Stum (CNES) S. Desai (JPL) A. Kitiyakara (JPL) T. Pavlovitch (JPL)	
Approved by:	P. Vincent (CNES) B. Haines (JPL)	

Document Ref: SMM-ST-M1-EA-11577-CN		Issue: 3	Update: 0

Algorithm change record	creation	date:	Issue:	Update:
	CCM			

FUNCTION

This function averages the three individual (330-ms) antenna temperatures in each channel for a once per second antenna temperature measurement.

ALGORITHM DEFINITION

Input Data

- Product data: None
- Computed data:
  - From “JMR_TAMODE1_01 - To compute the 330 ms antenna temperatures in mode 1” or “JMR_TAMODE2_01 - To compute the 330 ms antenna temperatures in mode 2”
    - 12 individual 330 ms antenna temperatures (for each of four channels and each of three noise diodes)
    - 12 associated antenna temperature quality flags
- Dynamic auxiliary data: None
- Static auxiliary data: None

Output Data

- For each once per second JMR measurement
  - 4 averaged antenna temperatures (one for each of four channels)
  - 4 associated antenna temperature quality flags (one for each of four channels)
  - Number of 330 ms measurements used to compute once per second measurement

Mathematical Statement

For each channel, the averaged antenna temperatures are derived by calculating the mean of the three individual 330 ms antenna temperatures that are measured from three noise diodes. The associated antenna temperature quality flag for each channel is set to 0 when all three 330ms antenna temperatures have valid measurements, and is set to 1 when any one of the three 330ms antenna temperatures is indicated to be invalid.

ALGORITHM SPECIFICATION

Input Data

- Antenna temperatures for each channel and noise diode TA[1:4][1:3] (Kelvin)

- Antenna temperature quality flags[1] TA_FLAG[1:4][1:3] (/)

Output Data

- Average once per second antenna temperature in each channel TA_AVG[1:4] (Kelvin)
- Quality flag for average once per second antenna temperature[2] TAAVG_FLAG[1:4] (/)
- Number of points used to compute average antenna temperatures NAVG[1:4] (/)
- Execution status

Processing

- For each channel i ∈ [1:4]
  - Set TA_AVG(i) = 0
  - Set NAVG(i) = 0 (number of measurement in average)
  - Set TAAVG_FLAG(i) = 1
  - For each noise diode j ∈ [1:3]
    - If TA_FLAG(i,j) = 0 then
      - TA_AVG(i) = TA_AVG(i) + TA(i,j)
      - NAVG(i) = NAVG(i) + 1
  - If NAVG(i) > 0 then
    - $\mathrm{TA\_AVG(i)} = \frac{\mathrm{TA\_AVG(i)}}{\mathrm{NAVG(i)}}$
    - Set TAAVG_FLAG(i) = 0
- If TAAVG_FLAG(i) = 1 for all i ∈ [1:4]
  - Set execution status to “bad”
- Otherwise
  - Set execution status to “good”

Accuracy

---

(1) The antenna temperature quality flags TA_FLAG[1:4][1:3] are set to a value of 0 for valid antenna temperatures and 1 for invalid antenna temperatures.

(2) The antenna temperature quality flags TAAVG_FLAG[1:4] for the averaged antenna temperatures are set to a value of 0 for valid antenna temperatures, and 1 for invalid antenna temperatures.

None

Comments

None

References

- Kitiyakara, A., JASON Microwave Radiometer (JMR) T_A Retrieval, JPL Interoffice Memorandum, April 6, 1998, revised August 8, 1998

DISTRIBUTION LIST

INTERNAL:

P. ESCUDIER	DGA/T/ED/AL/D	PB 2002	
A. AURIOL	DGA/T/ED/AL/DO	"	
M. COSTES		"	
F. PARISOT	DGA/T/ED/AL/TO		
P. HOZE			
J. PERBOS			
M. BLUMBERGER			
G. ZAOUCHE			
S. COUTIN-FAYE	DGA/T/ED/AL/MA	"	
J. NOUBEL		"	
P. VINCENT			
B. DELATTE			
B. BONHOURE			
N. PICOT			
G. TAVERNIER			
R. QUINTARD	DGA/T/ED/SG/CP	"	
P. SNINI		"	
P. GILLEN			
G. BOUTONNET	DGA/T/TI/PS	BP 1501	
M. H. GOMEZ		BP 211	

EXTERNAL:

M. DESTOUESSE	CLS		