NASA/TM–2019-219037 ICESat-2-ALG-PROC-0675

ATLAS Flight Science Receiver Algorithms

Version 4.0

Jan McGarry, Claudia Carabajal, John Degnan, Stephen Holland, Anthony Mallama, Stephen Palm, Ann Rackley Reese, Randall Ricklefs, and Jack Saba

September 2019

NASA/TM–2019-219037 ICESat-2-ALG-PROC-0675

ATLAS Flight Science Receiver Algorithms

Version 4.0

Jan McGarry
Goddard Space Flight Center, Greenbelt, MD

Claudia Carabajal
Hexagon US Federal, Lanham, MD

John Degnan
Hexagon US Federal, retired, Lanham, MD

Stephen Holland
Hexagon US Federal, Lanham, MD

Anthony Mallama
Emergent Space Technologies, retired, Laurel, MD

Stephen Palm
SSAI, Lanham, MD

Ann Rackley Reese
KBR, Greenbelt, MD

Randall Ricklefs
Hexagon US Federal, Lanham, MD

Jack Saba
SSAI, Lanham, MD

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, MD 20771

September 2019

Trade names and trademarks are used in this report for identification only. Their usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

Level of Review: This material has been technically reviewed by technical management.

Available from

NASA STI Program
Mail Stop 148
NASA's Langley Research Center
Hampton, VA 23681-2199

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
703-605-6000

Available in electronic form at https://www.sti.nasa.gov and https://ntrs.nasa.gov

ATLAS Flight Science Receiver Algorithms

Version 4.0

ICESat-2-ALG-PROC-0675

September 2019

Jan McGarry (NASA/GSFC: Jan.McGarry@nasa.gov)
Claudia Carabajal (Hexagon US Federal[1])
John Degnan (Hexagon US Federal[1], retired)
Stephen Holland (Hexagon US Federal[1])
Anthony Mallama (Emergent Space Technologies, retired)
Stephen Palm (SSAI)
Ann Rackley Reese (KBR[2])
Randall Ricklefs (Hexagon US Federal[1])
Jack Saba (SSAI)

Earth Sciences Division
NASA / Goddard Space Flight Center

Development of the DEM/DRM/SRM by
Lori Magruder, Holly Wallis,
Tim Urban, and Bob Schutz,
University of Texas, Center for Space Research

Website: icesat-2.gsfc.nasa.gov

[1] Formerly Sigma Space
[2] Formerly SGT

Acknowledgements

The authors would like to thank all of the members of the ATLAS Instrument Team, the ICESat-2 Mission Team, the Project Science Office, the Precision Orbit Determination Team, the ICESat-2 teams at the University of Texas at Austin, and the team at Orbital ATK for all of their support during the design, development and testing of the ATLAS Receiver Algorithms. In particular we would like to acknowledge the major contributions of the following people:

At the Goddard Space Flight Center:
ATLAS Instrument: John Cavanaugh, Philip Luers, Anthony Martino, Carol Lilly, Peter Gonzales
Project Science Office: Thomas Neumann
Precision Orbit Determination Team: Scott Luthcke
Flight Software: Joseph-Paul Swinski, Steven Slegel, Krishnan Narayanan, Alexander "Sandy" Calder
Bench Checkout Equipment: Peter Liiva, Henock Legesse
Instrument Support Facility: Peggy Jester, James Golder, Jairo Santana

At the University of Texas at Austin:
Surface Reference Mask Development: Tim Urban, Joseph Shih, Yidnek Tibebu, Afsheen Vaid
Digital Elevation Model and Digital Relief Map Development: Lori Magruder, Holly Wallace Leigh, Bob Schutz[3]

[3] Deceased 2015

Table of Contents

0.1 Introduction

This is the basis document for the ATLAS Receiver Algorithms. ATLAS is the single instrument on the ICESat-2 mission. The Receiver (Rx) Algorithms select the signal location in real-time and instruct the hardware to telemeter a vertical band of received time-tags about this signal location. The algorithms are implemented in the Flight Software (FSW) and in the hardware of the Photon Counting Electronics (PCE) cards. The sole purpose of the algorithms is to reduce the telemetry data volume to fit within the downlink constraint while maximizing the probability of downlinking surface signal. All versions of the Receiver Algorithms from version 2.8 onward incorporate our best knowledge of the ATLAS hardware at this time. It presents algorithms that, based on current Simulator testing, satisfy all of the requirements for the Rx Algorithms. Testing during ATLAS Instrument Integration and Testing with the hardware and software implementation of the Rx Algorithms revealed characteristics and responses that required changes to the Rx Algorithms and this document, which are captured in the change record below.

All relevant documentation for the Receiver Algorithms can be found on the ICESat-2 Technical Data Management System (TDMS) under the ATLAS Algorithms subsystem.

The launch version of the FSW (version 3.1.0) is based upon version 3.7c of this document.

The latest version of this document can be found on TDMS under ICESat-2-ALG-PROC-0675.

Other related documents, which are currently on the ICESat-2 TDMS, are:

- The ATLAS Coordinate Systems Descriptions Report, written by Marc Saltzman, is on TDMS under ICESat-2-SYS-RPT-0591.

- A description of the Simulator is given in the “Guide to the ATLAS Simulator for Users and Programmers” which is on TDMS under ICESat-2-ALG-TN-0577.

- The document describing the Receiver Algorithms testing is the “ATLAS Receiver Algorithms Test Plan,” which explains the tests, including the purpose of each, and how to set each up. This is on TDMS under ICESat-2-ALG-PLAN-0310. The results of the tests described in this document are detailed in the “ATLAS Receiver Algorithms Simulator Test Results” document. This is on TDMS under ICESat-2-ALG-RPT-0659.

- The latest onboard databases (DEM, DRM, SRM) and their related documentation are on TDMS under ICESat-2-ALG-TN-0362.

- The latest Receiver Algorithm Parameters are on TDMS under ICESat-2-ALG-SPEC-0255.

- The Rx Algorithm Parameter Definition Document is on TDMS under ICESat-2-ALG-TN-0876.

0.2 Changes since previous documents

Changes from 3.7d to 4.0

1. Document numbers updated to reflect what is on TDMS
2. Company names updated for authors
3. Best attempt of unifying tense used throughout doc
4. Multiple rewordings completed to make documented clearer
5. Exact PCE delay values and explanation of values used by Rx Algs given in section 4.15
6. Text in 5.3 regarding the calculation of the number of software bins in a histogram corrected based on bin numbering starting at 0.
7. Padding table updated with accurate values.
8. Rx Alg Knobs Decision flowchart (Fig 7.1) updated to accurately reflect the knobs files AND what is allowed by the Algs
9. The DRM-140 relief is no longer an option for *Nosig_Relief2_Strong/Weak* in the Algorithm Parameter Knobs files. This resolves the mismatch between the Simulator and the FSW that was caused by the lack of clarity in the Receiver Algorithms Document. The Simulator only allows the DRM-700 or DEM to be selected for relief while in the timer 2 state, where as the FSW allows either the DRM-140, DRM-700 or DEM. This document has been updated to clarify only the DRM-700 and DEM can be selected for the *Nosig_Relief2_Strong/Weak* parameter.

Changes from 3.7c to 3.7d

1. Changed launch date to reflect actual launch.
2. Added a comment in Section 5.3 that the software binsize must be =< 64 hardware bins.
3. Added a note in Section 7.6 that the calculated background is very inaccurate when the entire range window is used for the calculation.
4. Switched all of the parameters given in Appendix F from v5 to v6 (launch version).
5. Corrected knob parameter names in Appendix G from “S” to “s” to match Flight Software. The parameters affected were Nosig_Relief, Nosig_Timer, and Nosig_Scale.

Changes from 3.6b to 3.7

1. Changes were made to sections 3.2, 4.2, 4.6 and 4.8 describing the ATLAS coordinate system used. The term Instrument Coordinate System (ICS) has been replaced with terms that match those used in Marc Saltzman’s “ATLAS Coordinate Systems Description Report”.
2. A few paragraphs have been added to Appendix E (Error Analysis) to explain the laser vector errors and the errors introduced by the coordinate systems being used and the Beam Steering Mechanism.
3. Figure 1.2 has been changed to show the current orientation of the laser spot pattern on the Earth’s surface and the orientation with respect to the ATLAS_CSYS (ATLAS MRF) X-Y axes.
4. There has been another leap second since v3.6b of this document and so the leap second value has been updated in section 4.3.
5. A note has been added to 4.17 to capture the fact that the Altimetric Range Window must be within the Atmospheric window.

6. The Parameters in Appendix F have been updated to the current version (v5.0) which will be installed on ATLAS in August 2017.

Changes from 3.6a to 3.6b

Addition of section 7.10 describing the modifications needed in the ATLAS FSW to send all fires down from all three of the PCEs.

Changes from 3.5g to 3.6a

1. Corrected the wording on what values Nsf could have (several places).
2. Corrected reference to the number of steps in process in section 5.4.
3. Clarified handling of the SuperFrame subwindow end in section 5.4.

Changes from 3.5c to 3.5g

1. In Appendix G (Parameter Type Definitions) the parameters TEPstart (strong, weak) were changed from UINT32 to INT32 to allow TEPstart to be negative.
2. Wording changes to clarify meaning in section 7.3 and 7.4.
3. Added an additional check to see if the TEP_TLMB_start is negative in section 7.5. This is now needed because the TEPstart can now be negative.
4. Updated parameters files in Appendix F to latest versions (v3.0 on MIS).
5. Spot numbers are listed with tracks (in Figure 1.2).

Changes from 3.4g to 3.5c

1. In section 3.5 and 3.13 the reference number of the parameter files is given for MIS.
2. In section 5.4 an additional item was added to the superframe procedure when Nsf=2.
3. Sections 7.1, 7.3 and 7.4 were revised to clarify the various options and their relationships. A flowchart was added (as Appendix J) to help in this clarification.
4. Appendix E was updated to reflect the latest information.
5. The Parameter Files in Appendix F were updated to those released with FSW v2.1.
6. A Parameter Type Definition list was inserted as Appendix G (in front of the spacecraft data formats).
7. Correction of a typo in section 7.3.
8. Replaced TBDs in section 9.0 and some wording updates.

Changes from 3.4e to 3.4g

1. Modified wording in section 7.3 to clarify coastline relief, padding and scaling.
2. Modified wording in section 7.4 to clarify relief, padding and scaling used for NoSig_Time1 and NoSig_Timer2 when coastline bit is on.
3. Modified section 5.4 to add need to adjust the subwindow start by the PCE delay before comparing to the start of the range window.
4. Modified section 7.4 to add need to adjust the signal location by the PCE delay before comparing to the range window start and end.

Changes from 3.3 to 3.4e

1- Clarified the exclusion about bins with max count that are too close to the front or end of the histogram (section 6.3).
2- Removed the last sentence in the paragraph in section 5.4 that starts with “The subwindow width is calculated from the DRM-700 relief”, which indicated to move the subwindow entirely within the range window. Moving the subwindow is not required.
3- Added a step #7 to section 5.4 to ensure that the signal location falls within the range window.
4- Added a clarification to section 6.3 in case there were two bins with the maximum count.
5- Modified Appendix D to reflect the current state of the sigma multiplier table and which histogram to use for the number of bins.
6- Corrected wording in section 6.2 on which 400 shot histogram bins only had data from a single 200 shot histogram.
7- Changed wording in Intro to clarify statement on our knowledge of ATLAS hardware.
8- Added verbage in section 7.3 to explicitly specify the need for doubling the padding in this section.
9- Clarified in section 7.7 that if the telemetry bands are not combined due to the max restriction that the chopping of bins does not occur until after all combining is done.
10- Clarified in section 5.4 that the subwindow start can be negative with respect to the RWS for MF#3.
11- Change to section 7.4 reflecting the suggestion by the FSW team about switching immediately to state 2 (NoSig_Timer2) when the previous signal was outside the current range window.
12- Now explicitly states that the subwindow must be forced to be non-negative with respect to the lowest Range Window start in section 5.4.

Changes from v3.2 to v3.3

1- Additional clarification to the NoSig timer state (section 7.4).
2- Change to the calculation of SigLoc in section 5.3 to add 0.5 hwbins or 1 clock cycle.

Changes from v3.1 to v3.2

1- In section 5.3 the Algorithms say that if the maximum bin is beyond the last full hardware bin, then the last full software bin should be chosen as the max bin. The Flight Software is not able to do this. We have left the Algorithm as stated, but have written an exception for the FSW.
2- The threshold should be calculated using the threshold function. This has now been explicitly stated in section 5.3.
3- The TEP parameters to be used to define the TEP_NOT region should be converted to start and end and then rounded to the nearest hardware bin boundary. Now stated in section 5.3.
4- Section 5.3 now states that, in the Bnoise calculation, the max bin used should be the one AFTER the TEP_NOT rejections.
5- The calculation of the threshold in section 5.3 has “ceiling” added.
6- In section 7.4 the timer state handling has been clarified, plus other clarifications.
7- The downlink band overlap was clarified in section 7.7.

Changes from v3.0 to v3.1

1- In section 7.4 the Telemetry Band Offset usage was clarified for both the NoSig_Timer1 and NoSig_Timer2 cases.

Changes from v2.8g to v3.0

1- Due to the leap second being introduced June 30, 2015, GPS-UT1 (aka LS) will become 17 seconds (section 4.3) so this value should now be used as the default.
2- In section 5.4 verbage was added to clarify that the range window being discussed refers to the range window of MF #3 out of the 5 in the Super Frame.
3- In section 7.4 a sentence has been added to clarify which signal location to use if both the primary and tertiary exist.

Changes from v2.8d to v2.8g

1- Modified the paragraph related to relief for coastline data (section 7.4).

Changes from v2.8d to v2.8f

1- Added some information on the TEP to section 3.10.
2- Corrected the way the PCE delays are being added to the range window (section 4.15). This involves reordering some of the calculations from what was in v2.8d of the document.
3- Changed the conversion of Jrw and Mrw from RWS and RWC in sections 4.16 and 4.17 to use the "ceiling" function instead of "integer" (which is the same as the FORTRAN "floor").
4- Added note to clip the telemetry band if any part falls outside the range window (section 7.9).
5- Changed the handling of the signal location for the PCE delay in the NoSig_Timer2 case (section 7.4).
6- Changed the handling of the TEP telemetry region for the PCE delay (section 7.5).
7- Updated the Algorithm Parameter files in Appendix F.

Changes from v2.8c to v2.8d

1- Cosmetic change to insert a line break between iTEP_flag=0 and next line on page 51.
2- Changed NoSig_Relief1 to NoSig_Scaling in the equation at the top of page 50.

Changes since v2.7

1- An algorithm to handle the "PCE delay" hardware issue has been added (see sections 4.15 and section 7.9).
2- A warning has been added to section 4.14 to limit the cosine of the off-nadir angle to +/- 1.0 before taking the arccosine.
3- The algorithm to find the secondary signal has been corrected (see section 5.3). This was an error in the previous version of the Rx Algorithm that was found during Simulator testing.
4- All flowcharts have been updated to ensure they reflect the latest version of the Algorithms.
5- The Baseline Knob values have been modified slightly by the Project Scientists and this is now reflected in section 7.1.
6- The wording for the No Signal case in section 7.4 has been clarified.

7- The Algorithm was changed to switch from the NoSig_Timer1 parameters to the NoSig_Timer2 parameters when the surface type changed (in addition to when the NoSig_Timer1 expires). This was an issue discovered during Simulator testing that required correction. See section 7.4.
8- In section 7.6 the derivation of the pTLMB now correctly references the Range Window and not the DEM.
9- Section 7.7 now specifies that the telemetry bands should not be combined if the combined length would be greater than the telemetry band maximum.
10- Section 8 was revised to clarify some of the information.
11- The Flight Parameters Appendix F was updated to contain the latest parameter files.
12- The actual Algorithm Parameter names are now listed in this document.

Changes since v2.6

1- Some clarifications were added to section 3.
2- The value for the speed of light had a typo in the lowest digit which was corrected.
3- In section 4 it is clarified that, while the atmospheric histogram has a maximum length of approximately 14 km, that range window can be less.
4- In section 4.8 the range-rate calculation is now shown as using up to 10 range values.
5- It is clarified in section 4.10 that the FSW will normally be doing the extrapolation calculations for the range window conversion out to 1.5 seconds.
6- In section 4.11 an explanation is given on what to do in the case of a DEM error where either there is no lower tier tile when there should be, or when the lowest tier's range window width is greater than the maximum.
7- A calculation for the solar zenith angle has been added to the Rx Algorithms in section 4.13, and this value is used to determine day or night, which in turn is used to index into the limits on the window widths.
8- Section 4.16 adds calculations to the Rx Algorithm to limit the separation between strong and weak spot altimetric range windows, addressing a need by the PCE hardware recently determined.
9- Clarifications were added to section 5 to explain what is happening in the hardware.
10- Section 5.2 now talks about the 50 shot summation of events in the altimetric range window which is performed in the PCE hardware.
11- In section 5.3 the Rx Algorithm was modified to allow restriction of the TEP region from the signal processing and to explain the ramifications of this.
12- Specific names were given to the coarsened histograms. The one coming from the hardware is called the overlapping histogram, while the subset of the coarsened histogram that gets used for signal processing is called the selected histogram.
13- Section 5.3 now also provides an explicit method for calculating the number of software bins in the selected histogram.
14- Section 5.4 now indicates to move the SF subwindow if it sticks out beyond either end of the range window.
15- In section 7, an explanation is given for handling telemetry when no signal has been found from the start of Science Mode.

16- Section 7.4 now uses the range window start and stop, rather than the DEM min and max, for the telemetry band.
17- There is an explicit statement now added to section 7.5 indicating when TEP should be sent. The TEP calculation was also changed to replace RWS and RWW with Jrw and Nrw.
18- Section 7.7 has been rewritten to make it easier to understand.
19- The TEP table in section 7.8 has been updated.
20- The section on limiting the data to what the SpaceWire can handle has been removed since the Rx Algorithm does not have enough information to make this limitation in an effective manner. In addition, the algorithm for doing this is too time consuming for the FSW to implement.

1.0 ATLAS Instrument and ICESat-2 Mission

1.1 Mission science objectives

- Quantifying polar ice sheet contributions to current and recent sea-level change and the linkages to climate conditions.
- Quantifying regional signatures of ice sheet changes to assess mechanisms driving those changes and improve predictive ice sheet models.
- Estimating sea ice thickness to examine ice/ocean/atmosphere exchanges of energy, mass and moisture.
- Measuring vegetation canopy height as a basis for estimating large-scale biomass and biomass change.
- Enhancing the utility of other Earth observation systems through supporting measurements.

1.2 Key mission parameters

Orbit: 500km, 91-day repeat, with monthly sub-cycles
Orbital inclination: 92°
Maximum off nadir pointing during data collection: 5°
Laser Rep. Rate: 10 kHz
Number of beams: 6, with unequal energy
Along-track spot separation: 70 cm
Wavelength: 532 nm
Launch and lifetime: September 15, 2018 with 3-year mission lifetime

1.3 Overview of the instrument

The ATLAS instrument is a single-photon-sensitive lidar system with a laser repetition rate of 10 kHz (see ATLAS block diagram in Figure 1.1). The laser is capable of transmitting 250 to 900 microJoules per pulse (tunable on-orbit) at 532nm. The beam divergence is approximately 25 microradians (FWHM) giving a spot diameter on the Earth surface of about 12 meters. The receiver field of view diameter is about 3 times that of the laser spot.

The transmitted laser pulse is split into six beamlets by a diffractive optical element (DOE). The beamlets are separated by 5-7 milliradians in angle which gives ~ 2.5 to 3.5 km separation of the spots on the surface. The six spots are grouped into three mostly identical "tracks" consisting of 2 spots each (Figure 1.2). The laser energy is divided unequally between the spots in a track. One contains nominally four times the transmit laser energy of the other. The spot with the higher energy is called "strong" and the lower energy one is called "weak". Because the spacecraft flies about half the time oriented in one direction and half the time in the opposite orientation, the strong spot sometimes precedes the weak and at other times it trails.

The altitude of the spacecraft, combined with the laser pulse repetition frequency (PRF), implies that there are about 33 laser pulses in flight at any given time. At a PRF of 10 kHz, each laser pulse travels approximately 30 km (15 km round-trip) before the next laser pulse is fired. For clouds with altitudes at or greater than 15 km, ground returns may overlap with cloud returns from the subsequent pulse, potentially confusing the onboard Receiver Algorithms (Figure 1.3). This will most likely not be much of an issue, except for vegetated areas, since the cloud returns being diffuse are generally broader than the surface returns.

The photons from each spot are collected by the ATLAS optical system and delivered to the detector associated with that spot. Each detector face is divided into multiple channels. The strong beams have sixteen detector channels, and the weak beams have four. The signal and noise for each spot are spread randomly over all of the detector channels. This is done to reduce the effect of receiver dead time and also reduce the effect of first photon bias.

The laser beam pulsewidth is expected to be about 1.5 ns. Since the dead-time of the receiver subsystem is about 3 ns for each channel, a photon arriving at a given channel less than 3 ns behind the previous is not recorded.

The Photon Counting Electronics (PCE), a part of the Main Electronics Box (MEB), measure the laser fire and photon receive times (or photon time tags) and also perform much of the low level calculations for the Receiver Algorithms. These include separating received events into histogram bins and some low-level processing. This work must be done in the PCE hardware because of the high laser PRF and the speed at which processing is required.

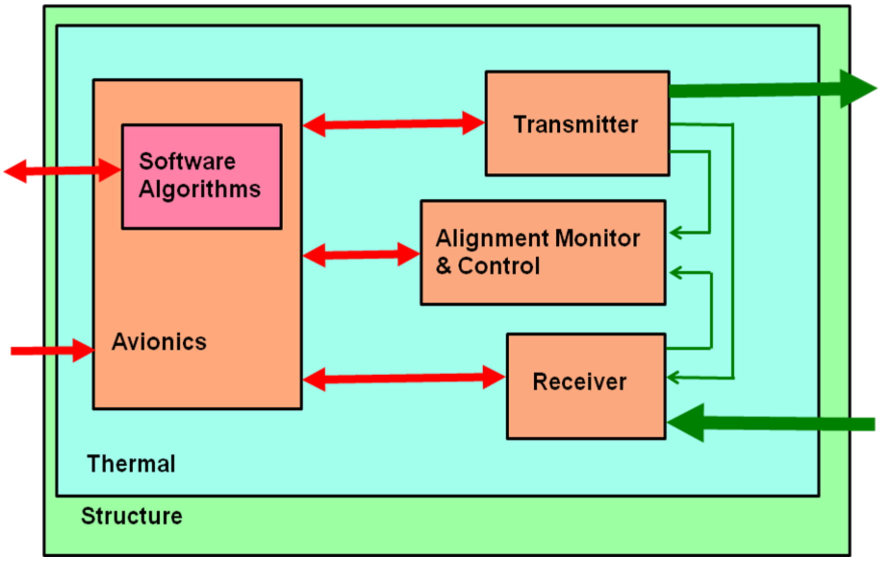


Figure 1.1: High Level Functional View of ATLAS Instrument (figure courtesy of Anthony Martino, code 615, NASA/GSFC)

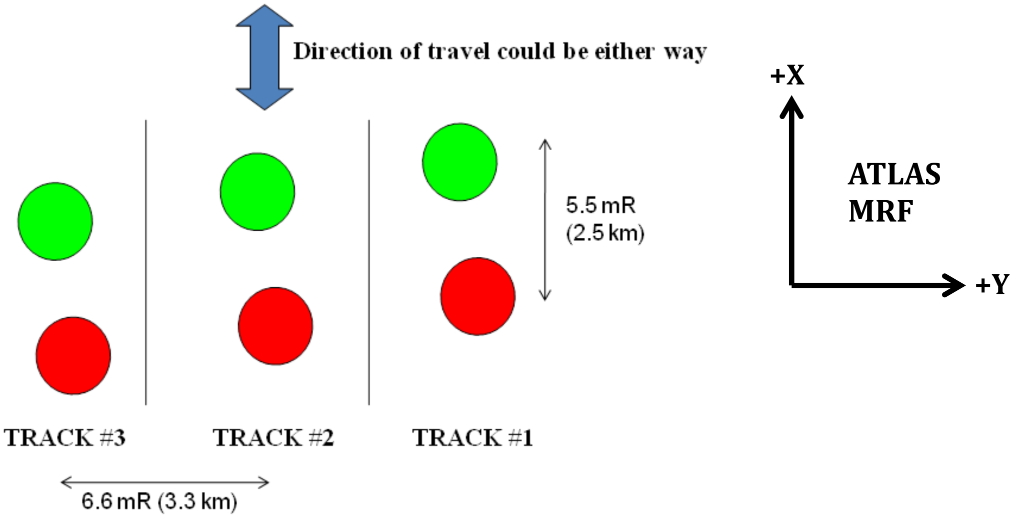


Figure 1.2: Laser spot pattern on the Earth. Spot 1 (strong: RED) and 2 (weak: GREEN) are in Track 1, Spot 3 (strong) and 4 (weak) are in Track 2, and spot 5 (strong) and 6 (weak) are in Track 3.

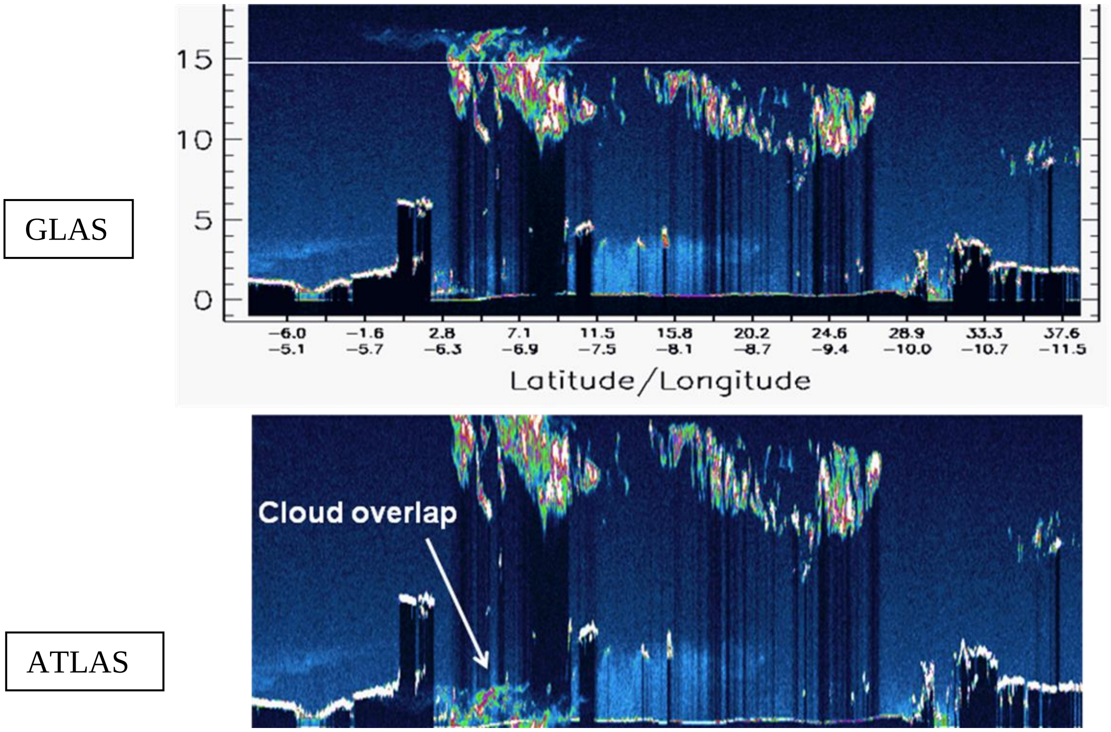


Figure 1.3: Cloud overlap example. Top picture shows nighttime clouds as GLAS measured them. Bottom shows the same atmosphere as measured by ATLAS. Both are at night. Clouds above 15 km in GLAS plot show up at bottom of ATLAS plot.

2.0 Purpose, Products, Requirements and Constraints

The ATLAS Flight Science Receiver Algorithms (FSRA) has one purpose: to reduce the science telemetry bandwidth to fit within the downlink constraint (577.4 Gb/day) while maximizing the probability of selecting and downlinking returns from the surface. Without the Algorithms, the expected daylight high noise rates, coupled with the expected surface return rates, could give a data volume >> 15 times the currently specified bandwidth limitation.

The Algorithms reduce the telemetry volume by performing signal processing on the received events to determine the surface location in vertical space, and by using an onboard digital relief map (DRM) of the Earth to determine the expected height range of the surface data. These two pieces of information are then used to select the time tags to be downlinked to the ground. The remaining time tags are discarded.

In addition to the signal timing information, an atmospheric histogram is generated over a 14 km vertical window. The atmospheric histogram from each strong spot, built from the data for 400 consecutive shots, is telemetered to the ground.

The Receiver Algorithms are run simultaneously on three identical processors (one processor on each of the three Photon Counting Electronic boards). Each board contains the electronics for the strong and weak laser spots for one track. Although there are actually six ATLAS spots, this document will refer to only two: one strong and one weak. The Receiver Algorithms described here are identical for all three tracks.

3.0 Overview of the Algorithms

The ATLAS Flight Science Receiver Algorithms have a heritage going back to the Mars Observer Laser Altimeter (MOLA) in the 1980s where a receiver algorithm was implemented in the Flight Software to optimize the data collected. Most recently the Mercury Laser Altimeter (MLA) and the Lunar Observer Laser Altimeter (LOLA) Receiver Algorithms were developed by teams that included members of the ATLAS Algorithms Team. The ATLAS Algorithms share many of the fundamental characteristics of the former Algorithms including the goals of maximizing the probability of detection, minimizing the probability of false alarm, and performing signal processing over a period of time called a frame.

The Geoscience Laser Altimeter System (GLAS) Flight Science Receiver Algorithms also provide a heritage for ATLAS, although many of the Algorithms on GLAS were related to the detector digitized waveform which has no analogy on ATLAS. ATLAS, however, continues the GLAS practice of flying an onboard Digital Elevation Model (DEM) for use in determining where to set the Range Window.

On GLAS and MLA the spacecraft provided real-time orbital information to the onboard Algorithms. For MLA this was an estimated range to the surface, used to set the Range Window. On GLAS, similarly to ATLAS, this was a rectangular position of the spacecraft relative to the center of the Earth.

MOLA was an instrument on the Mars Global Surveyor (http://mars.jpl.nasa.gov/mgs/), MLA is an instrument on MESSENGER (http://messenger.jhuapl.edu/), and GLAS was the instrument on the original ICESat mission (http://icesat.gsfc.nasa.gov/).

3.1 Spacecraft position and pointing

The spacecraft position, velocity, pointing (in the form of a quaternion), and rotational rates (relative to the spacecraft body) are given to the Flight Software once per second. All pieces of information are less than 500 milliseconds stale when they are transmitted from the spacecraft to the Flight Software. The spacecraft position and associated velocity arrive in a packet together. The quaternion and body rates also arrive together and at a potentially different time than the position and velocity information. Each packet has an associated timestamp representing the time of the data given in the spacecraft GPS time.

3.2 Laser beamlets

A diffractive optical element (DOE) is used to break the laser beam into 6 beamlets. Each has a different off-nadir angle. Associated with each laser beamlet is a fixed vector defining its direction in the ATLAS coordinate system (ATLAS_CSYS). It is assumed for purposes of this document (and Algorithms) that all six beamlets have the same coordinate system origin at the origin of the ATLAS_CSYS. Each strong spot has an associated weak spot that travels along a close path. These spot pairings are called tracks.

3.3 Laser spot location on the Earth ellipsoid

The location of each laser spot on the Earth ellipsoid is calculated in real time from the spacecraft position, velocity and pointing information, along with the fixed unit vector representing the laser transmit direction in the instrument coordinate system.

The Flight Software requires time to process this information and send it to the hardware, so the spot locations must be extrapolated up to 1.5 seconds into the future. Each laser vector is rotated from the instrument body coordinate system to Earth Centered Earth Fixed (ECEF) and then translated to the surface of the Earth ellipsoid.

The latitude and longitude of the laser spots on the Earth ellipsoid, and a distance (range) from the origin of the spacecraft coordinate system, are then calculated.

3.4 Digital Elevation Model (DEM), Digital Relief Maps (DRM), and the Surface Reference Mask (SRM)

The Digital Elevation Model is an onboard database of tiles containing minimum and maximum heights relative to the WGS84 ellipsoid for each tile. The index into the DEM is a function of latitude and longitude. See Appendix A for more information.

The Digital Relief Maps are onboard databases containing the maximum relief over a specific ground track distance for each tile. As with the DEM the index into the DRM is based on latitude and longitude. There are two DRMs, one corresponding to a ground track distance of 140m and one corresponding to a ground track distance of 700m. See Appendix B for more information.

The Surface Reference Mask is a database of information on the surface type and associated characteristics. It is referenced by latitude and longitude as the DEM and DRM are. The four surface types are ocean, land, sea ice, and land ice (ice sheets, glaciers, ice fields, ice caps, floating glacier tongues and ice shelves all fall into the last category). The SRM also specifies whether vegetation is present in this tile, and if the tile contains a coastline. There are four additional bits in the SRM that are left as spares for future use.

3.5 Receiver Algorithm Parameters

The Receiver Algorithms use several parameters that can be modified on-orbit. Their values are specified in a set of parameter files. These files are used by the Receiver Algorithms Team during testing and then given to the Flight Software Team for use on-orbit. There is a separate set of files for each of the three PCE boards. These files consist of: (1) extrapolation and range parameters (PPR), (2) signal and telemetry parameters (ST), (3) parameters related to adjusting what is in the telemetry downlink for nominal operations (Knobs), (4) parameters setting the telemetry downlink for ocean scans and targets of opportunity (Alternate Knobs), and (5) the sigma multiplier tables which are used by the signal processing algorithm. These files can be found on TDMS (**ICESat-2-ALG-SPEC-0255)**.

3.6 Range Window

The distance to the laser spot on the Earth ellipsoid, combined with the DEM minimum and maximum heights (converted to range space), provides a starting and ending estimated range to the surface. Margin, based on surface type, is added to the start and end range to compensate for uncertainties in the DEM and the range calculation. The final products of these calculations are the altimetry Range Window Start (RWS, aka Jrw), the altimetry Range Window Width (RWW, aka Nrw) and the start of the atmospheric (cloud) Range Window (RWC, aka Mrw). The altimetry Range Window Width cannot exceed 6 km due to hardware limitations. The atmospheric Range Window Width (height) is normally 14 km.

Note that the Range Window Start represents the range closest to the spacecraft where surface echoes are expected, and the end of the Range Window (RWS+RWW) is the location farthest from the spacecraft where surface echoes are expected.

The RWS, RWW and RWC are provided to the hardware in units of clock cycles (where 1 clock cycle = ~10 ns) from the fire event. This information is used by the hardware to generate histograms of the incoming events. Receive time tags are only generated for events that fall within the altimetric Range Window. All events that occur outside of the Range Window are lost.

The delay of the fire from the closest clock cycle is NOT taken into account by the hardware so there is some smearing of the data within the histogram bins at the 10 ns (1.5 m) level.

3.7 Range Window histograms

In each Photon Counting Electronics (PCE) board Field Programmable Gate Arrays (FPGAs) generate histograms from the incoming photon events across the entire Range Window. For the altimetry histogram the Range Window Width (RWW) can be as large as 6 km, with the lower limit set by the margin added to the Range Window to compensate for uncertainties (nominally +/- 250 m). The hardware histogram bin size is 2 clock cycles (~20 ns or ~3 m) and the integration time is 200 shots (0.02 seconds at 10 kHz, during which the spacecraft moves ~140 m); this period (0.02s) is called a Major Frame (MF). For the atmospheric histogram the Range Window width is nominally ~14 km (93.4 µs), the histogram bin size is 20 clock cycles (~200 ns or ~30 m) and for the hardware, the integration time is also 200 shots. As part of the histogram processing the Flight Software directs the hardware to combine two consecutive 200 shot atmospheric histograms to produce a 400 shot histogram (0.04 seconds) every 0.02 seconds.

The starts of the altimetric and atmospheric Range Windows are not necessarily the same and the starts of each of the six spots' Range Windows are generally not the same, given that the location of the spots on the Earth surface are not identical.

At the end of the integration time (0.02 seconds), the hardware informs the software that the histograms are ready. For each spot the atmospheric histogram is formed by combining data from two consecutive major frames. Every other combined histogram from the strong spots is telemetered to the ground (i.e., one every 0.04 seconds). The weak spot histograms are not telemetered. Both the strong and the weak spot atmospheric histograms are available for the signal processing software along with the altimetric histogram. Altimetric histograms are only sent to the ground for diagnostic purposes and are not telemetered during normal data collection.

The altimetric histogram is used as indicated below in the onboard signal processing.

3.8 Major frame signal processing

A 200 shot period of time is called a Major Frame (MF). When the altimetric histogram is ready for processing, the hardware provides the Flight Software with some preliminary calculations, including the number of events across the entire histogram, the indices of the three bins with the most counts, and the number of counts in these bins. An uploadable parameter value based on surface type indicates whether to pick a software bin size from a table or to use the DRM value. The software bin size is given to the hardware, which combines the 3 m bins, converting the original histogram into a histogram with wider (coarser) bins whose width is a function of surface type. This is done to increase the probability of signal detection by matching the bin size with the expected response from the surface. This should enhance the ability to find weak narrow surface responses from low relief, as well as to find the broader responses from trees and high relief regions.

The event rate and the number of bins in the chosen histogram are used by the software to estimate the mean noise count per bin and derive an "optimum" threshold which yields a high probability of surface detection while limiting the number of false alarms. The number of bins in the histogram is used to select a scale factor from a “Sigma Multiplier Table” that is used in the threshold calculation to keep the false alarms at a rate of < 10%. See section 5 for details on these calculations. This threshold and bin size are provided to the hardware for further calculations. The FPGA provides very fast low level histogram routines and allows the software to queue up calculations, without having to wait for them to finish.

At the end of the Major Frame processing, the Flight Software decides whether signal is found or not. If signal is found, the bins where the signal is located is determined.

The calculations using the atmospheric histogram are strictly for the purpose of determining if clouds are present and are too thick for surface signal to get through. The calculations are performed for both the strong and weak spots individually even though only the strong spot atmospheric histogram is sent to the ground. Details of how these calculations are used are explained in section 7.

3.9 Multiple frame signal processing

Five consecutive Major Frames (MF) (labeled 1,2,3,4,5) are combined into a Super Frame (SF). The SF is used to support the determination of signal location in MF #3. The SF calculation is considered to have successfully found signal if the signal location was found in Nsf of the five MFs. In the current Algorithms Nsf = 3. The SF signal location is calculated by linear interpolation using the two MFs with signal that are closest to MF #3, but NOT using the center MF itself.

If signal is not found in either the Major Frame or the Super Frame, then signal locations are extrapolated from previous locations where the signal was found. Depending upon the Algorithm knobs this either continues for a limited period of time and then stops, or continues indefinitely until signal is again found.

3.10 Selecting what to telemeter

When signal is found in the Major Frame, relief from the DRM-140 tile corresponding to the spot's latitude and longitude is used to determine the telemetry band (in the vertical or height dimension). For frames with an interpolated or extrapolated signal location, the DRM-700 is used to determine the relief. The relief is scaled upward based upon a table look-up scale factor and then padded to compensate for the relief uncertainty. In the current Algorithms, the scale factor is 2 for land and land ice, implying that land and ice relief is doubled relative to the DRM value. A surface-type dependent maximum limit is imposed on the telemetry band. The band is nominally centered over the signal location. Margins to the band and an offset to the centering are selected from a table lookup based upon surface type.

The hardware allows for two bands to be selected for telemetry. The Algorithms use two telemetry bands for a number of circumstances, including when a Major Frame signal bin is outside the vertical location of the other signal bins in the Super Frame. A second telemetry band is also used when there is a strong secondary histogram peak that lies outside the primary telemetry band. See section 7 for details.

In addition to receiving return signals from the surface, the ATLAS instrument also sends a small portion of the transmitted pulse back into the detector for calibration. Each of these events is called the Transmitter Echo Pulse (TEP). The energy in the TEP is expected to be low enough that the Algorithms will not detect it. The TEP sometimes falls within the Range Window and sometimes not; when it does, the region containing the TEP is telemetered if indicated by the Algorithm Parameters.

Once a telemetry band or bands are selected, the starting hardware bin number and number of bins are converted to clock cycles and given to the hardware. The hardware then sends down time tags for all events that fall within these histogram bins.

3.11 Comment on units and calculations

All units in time (nanoseconds) are two-way or roundtrip and are exact. All units given in distance (meters) are one-way and in this document are an approximate conversion from the time.

All calculations given below are double precision floating point unless specified by either ***integer***[X] which implies truncate X, ***round***[X] which implies round X to closest integer, or ***ceiling***[X] which implies select the least integer that is greater than or equal to X. Double precision floating point values are assumed to have 52 or more significant bits (15.6 decimal digits). Integers are assumed to be 32 bit.

All tables and parameters that relate to the signal processing and telemetry have separate names and allow for different parameter values for each of the six spots. The parameters are grouped into three files, one for each track. See Appendix F.

3.12 Physical constants

The following physical constants are to be used in the implementation of the FSRA:

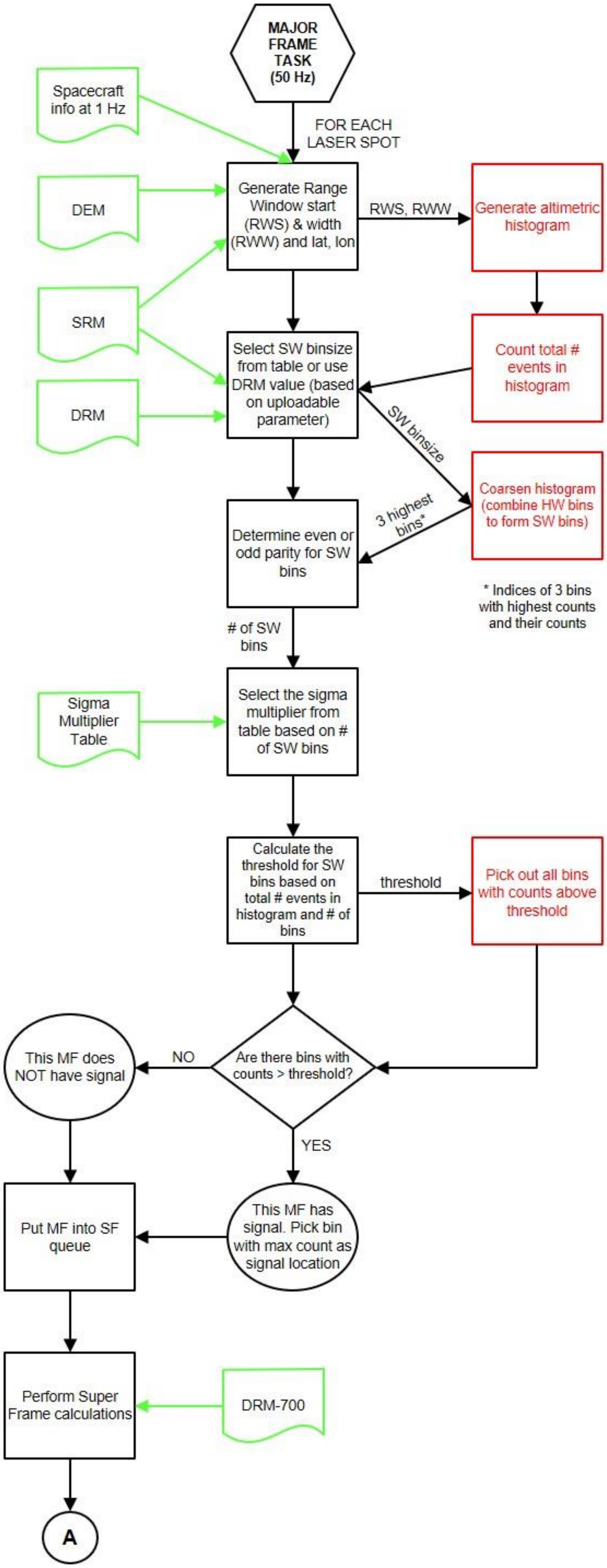
Speed of light: 299792458 m/s
Radius of the Earth (WGS-84): 6378137 m
Flattening of the Earth (WGS-84): 1/298.257224

3.13 Caveats

Algorithm Parameter values shown are for example only and may be different in the Algorithm Parameter files. Please check the latest files for the latest values. Files can be found on TDMS (**ICESat-2-ALG-SPEC-0255)**, or obtained from Jan McGarry (Jan.McGarry@nasa.gov) or Ann Rackley Reese (Ann.L.Rackley@nasa.gov). Values in Section 3.12 and Section 4's equations, however, are the final values and should be used as given in these sections.

3.14 FSRA flowchart

A simplified flow of Algorithms is shown with no Telemetry Knob selections and no use of the Palm Algorithm to determine if clouds prevent surface echoes from getting to the instrument. The red boxes are functions performed by the hardware. Green boxes are data inputs to software.


MAJOR FRAME TASK (50 Hz)
FOR EACH LASER SPOT
Spacecraft info at 1 Hz
DEM
SRM
DRM
Generate Range Window start (RWS) & width (RWW) and lat, lon
RWS, RWW
Generate altimetric histogram
Count total # events in histogram
Select SW binsize from table or use DRM value (based on uploadable parameter)
SW binsize
Coarsen histogram (combine HW bins to form SW bins)
3 highest bins*
Determine even or odd parity for SW bins
* Indices of 3 bins with highest counts and their counts
# of SW bins
Sigma Multiplier Table
Select the sigma multiplier from table based on # of SW bins
Calculate the threshold for SW bins based on total # events in histogram and # of bins
threshold
Pick out all bins with counts above threshold
Are there bins with counts > threshold?
NO
This MF does NOT have signal
YES
This MF has signal. Pick bin with max count as signal location
Put MF into SF queue
Perform Super Frame calculations
DRM-700
A

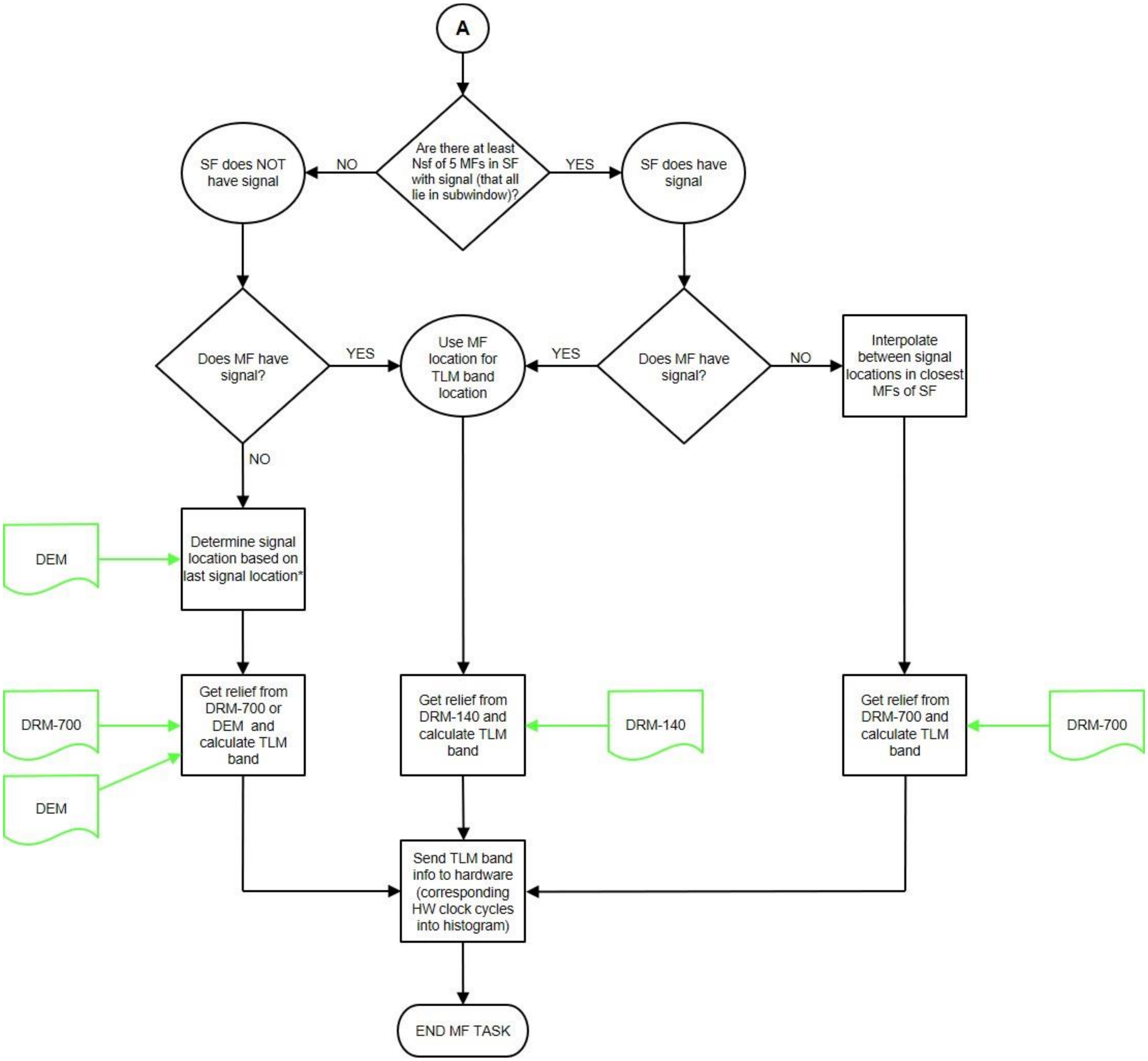

A
Are there at least Nsf of 5 MFs in SF with signal (that all lie in subwindow)?
NO
YES
SF does NOT have signal
SF does have signal
Does MF have signal?
YES
Use MF location for TLM band location
YES
Does MF have signal?
NO
Interpolate between signal locations in closest MFs of SF
NO
DEM
Determine signal location based on last signal location*
DRM-700
DEM
Get relief from DRM-700 or DEM and calculate TLM band
Get relief from DRM-140 and calculate TLM band
DRM-140
Get relief from DRM-700 and calculate TLM band
DRM-700
Send TLM band info to hardware (corresponding HW clock cycles into histogram)
END MF TASK

Figure 3.1. Simplified Receiver Algorithm Flowchart, showing an overview of the decisions for signal processing and telemetry downlink.

Note (*): If no previous signal, use DEM for telemetry band location.

4.0 Calculating Range Window Start and Width

4.1 What is a Range Window and why do we need it?

A temporal window is needed for the altimetry signal processing to reduce the false alarm rate. The Range Window provides this function. The software calculates the Range Window information based on the spacecraft location and pointing information, the fixed laser vectors in the instrument coordinate system, the Earth reference ellipsoid information, and an onboard Digital Elevation Model (DEM) of the minimum and maximum heights for each latitude/longitude tile on the Earth.

The software produces 12 Range Windows for each Major Frame. There is one altimetry Range Window and one atmospheric Range Window for each of the six spots. Each Range Window can potentially have different start and stop times. The width of the altimetric Range Window (stop minus start) is between 500 meters and 6 km. The maximum width of the Range Window allowed by the hardware is 6 km. The minimum width is determined by the accuracies of the DEM and the onboard range calculations (see appendix E). The Range Window width parameters (*PPR:Range_Window_Width_Max_Strong/Weak* and *Range_Window_Width_Min_Strong/Weak*) must be set to follow the Range Window width constraints and can also be used to further limit the width of the Range Window.

The atmospheric Range Window width (*PPR:Atm14km10ns_Strong/Weak*) is approximately 14 km and its start is calculated using the altimetry Range Window end and an offset (*PPR:Atmos_Range_Window_Offset_Strong/Weak*). Section 4.17 details this calculation.

The PCE hardware only saves the time of arrival of those received events that occur within the altimetric Range Windows for each spot.

4.2 Spacecraft supplied information

The spacecraft supplies position and velocity as well as pointing and rate information. The spacecraft position is in rectangular coordinates (X,Y,Z) in units of meters referenced to the Earth Centered Inertial (ECI) coordinate system (J2000.0). The pointing comes in the form of a quaternion, consisting of four elements, which transforms the position vector from the ECI reference frame to the ATLAS Coordinate System (ATLAS_CSYS) also known as the ATLAS Master Reference Frame (MRF). See Saltzman's ATLAS Coordinate Systems Descriptions Report (ICESat-2-SYS-RPT-0591). For the purposes of this document (and Algorithms) the origin of the ATLAS MRF and the spacecraft center of mass are the same. All positional and velocity information are provided at 1 Hz with less than 500 ms latency. The ATLAS MRF system used is essentially the reference system associated with the Laser Reference System (LRS). The ATLAS MRF Z-axis lies along the Observatory Coordinate System Z-axis (which is along the optical axis of the telescope). The X-Y axes of the ATLAS MRF are clocked -0.024 degrees from the Observatory Coordinate System (OBS_CSYS). This angular difference between these two coordinate systems is small enough to be ignored for the onboard Rx Algorithms as it results in an error for the ground spot location of < 20 meters. The rates are the angular velocities about the 3 ATLAS axes.

The Earth reference ellipsoid used by the onboard Algorithms and used in developing the DEM is WGS-84. Only the radius of the Earth at the equator (ae) and the flattening (flat) are needed by the Algorithms to define the WGS-84 ellipsoid. The values used by the onboard Rx Algorithms are given in section 3.11 (Physical Constants).

The spacecraft supplied information is extrapolated ahead, up to 1.5 seconds in the future, to the time that the Flight Software and electronics need it.

The following definitions hold throughout section 4:

XI, YI, ZI is the spacecraft supplied position vector in Earth Centered Inertial (ECI) to the center of the ATLAS instrument coordinate system at time TP. XI(TP) implies the value of XI at time “TP”. Similarly for YI(TP) and ZI(TP).

XIdot, YIdot, ZIdot is the spacecraft supplied velocity vector in ECI of the XI, YI, ZI spacecraft position. Similar to the position notation, XIdot(TP) is the X axis velocity at time “TP”.

q1, q2, q3, q4 are the four parts to the spacecraft supplied quaternion which represent the rotation of the spacecraft position from ECI to the ATLAS Master Reference Frame (MRF). The quaternion information is at time TQ (which will be less than +/- 500 ms different from TP).

dt is the FSW required extrapolation time.

The following define the reference frames used in the Rx Algorithms calculations, and the names for each of the parameters:

ECI is the Earth Centered Inertial reference frame (J2000.0).
ECEF is the Earth Centered Earth Fixed (ie rotating with earth) reference frame.
ECEF_hat is the coordinate system with axes aligned with ECEF but with origin at the ATLAS MRF origin.

Both ECEF and ECEF_hat are at time “TQ”, the time of the quaternion.

(xls,yls,zls) is the unit vector along the laser direction in ATLAS MRF. This is known and is fixed and is given by the Algorithm Parameter (*PPR:Beam_Pointing_Strong/Weak*).
(xlr,ylr,zlr) is the laser (unit) vector in the ECEF_hat coordinate system.

(XI,YI,ZI) is the vector from Earth center to the origin of the ATLAS MRF in ECI.
(XR,YR,ZR) is the vector from center of Earth to the origin of the ATLAS MRF in ECEF.

(xle,yle,zle) is the laser spot location on the Earth ellipsoid in ECEF.

PP is the rotation matrix from ECI to ECEF (about the ECI Z axis through mean sidereal time).
R is the range to the surface from the ATLAS MRF origin along the laser unit vector.

4.3 Calculating the time

Time is calculated from three separate places: (1) the position and velocity message from the spacecraft contains an embedded time which represents the time the information is valid (TP); (2) the quaternion message includes a similar embedded time (TQ); and (3) the "time at the tone" message contains the time of the last 1 Hz message (T0) and is used by the software to keep track of the current time. The formats of all of the three time messages are the same so the conversion is the same for all.

Conversion of the filtered time of the spacecraft position/velocity is straightforward. Call the integer seconds iTP. Call the fractions of seconds mTP. Then TP = float(iTP) + float(mTP)/10^7.

Do a similar calculation for the attitude/rate time: TQ = float(iTQ) + float(mTQ)/10^7.

The spacecraft time should be very close to the elapsed GPS time from 6 Jan 1980 00:00:00, and so no conversion between spacecraft time and GPS time is needed.

Note that the time on the spacecraft is GPS time which differs from UTC by the number of leap seconds since GPS was established. All calculations related to the Rx Algorithms are done in GPS time with the exception of the Sidereal Time calculation, which corrects the GPS time with the leap seconds.

The Mean Sidereal Time should be calculated at TQ. To do this requires the calculation of Julian Date (JD) when the time is given in date/time form of year (Y) and day of year (D):

$$\mathrm{NLY} = \textit{integer}[\ (\mathrm{Y}-2001)/4\ ] + 1$$
$$\mathrm{JD} = (\mathrm{Y}-2000)\cdot 365 + \mathrm{NLY} + \mathrm{D} + 2451543.5$$

However, with time coming from the spacecraft in seconds past the start of 6 January 1980, the onboard calculation is:

$$\mathrm{JD} = 2444244.5 + \mathrm{TQ}/86400.0$$

where 2444244.5 is the Julian Date corresponding to 6 January 1980 at 00:00:00 (GPS time).

Then the Mean Sidereal Time **(GMST)** calculation is:

$$\mathrm{XD} = \mathrm{JD} - 2451545.0 - \mathrm{LS}/86400.0 = ((\mathrm{TQ} - \mathrm{LS})/86400.0) - 7300.5$$

$$\mathrm{GMST} = 18.697374558 + 24.06570982441908\cdot \mathrm{XD}$$

(equation from the USNO Approximate Sidereal Time website: https://aa.usno.navy.mil/faq/docs/GAST.php, last accessed on July 30, 2019) where LS is the difference between GPS and UT1 and has both an integer and fractional part. From July 1st 2015 to December 31st 2016, LS = 17.0. As of January 1st 2017, LS = 18.0. There needs to be a command to upload the value of LS approximately once per month.

The Mean Sidereal Time (GMST) must be limited to be a valid time of day (0 to < 24 hours). This can be done taking GMST modulo 24:

```
GMST = mod(GMST,24.0)
```

The Mean Sidereal Time expressed as an angle in radians is then:

```
Θ = (GMST·15)·π/180
```

4.4 Conversion of spacecraft position, velocity, and quaternion from s/c messages

The spacecraft provides position, velocity, pointing and pointing rate information in two separate packets at 1 Hz to the Flight Software. The format of these messages is shown in Appendix G along with the conversions to units required by the FSRA and used in this document.

4.5 Extrapolation of position

Use the velocities to perform a linear extrapolation to get the spacecraft position to the same time as the quaternion:

```
XI(TQ) = XI(TP+ΔT) = XI(TP) + ΔT·XIdot(TP),  and similarly for YI,ZI.
```

where "∆T" is extrapolation time (may be -0.5 to +0.5 seconds) and represents the difference between the spacecraft positional time and the quaternion time:

```
ΔT = TQ – TP
```

The spacecraft position is also needed at time TQ+dt:

```
XI(TQ+dt) = XI(TP) + (ΔT+dt)·XIdot(TP),  and similarly for YI,ZI.
```

where dt = 0 to 1.5 seconds.

4.6 Spacecraft pointing

The spacecraft supplies four pieces of angular information (the Quaternion): q1,q2,q3,q4 representing the rotation at time TQ. Perform the following to calculate the rotation matrix from ECI to ATLAS MRF at time TQ. These equations were courtesy of Charles Webb, code DK000, NASA/HQ.

```
q11 = q1 · q1
q22 = q2 · q2
q33 = q3 · q3
```

```
q44 = q4 · q4

q12 = q1 · q2
q13 = q1 · q3
q14 = q1 · q4

q23 = q2 · q3
q24 = q2 · q4
q34 = q3 · q4

AA(1,1) = q11 - q22 - q33 + q44
AA(1,2) = 2.0·(q12 + q34)
AA(1,3) = 2.0·(q13 - q24)

AA(2,1) = 2.0·(q12 - q34)
AA(2,2) = -q11 + q22 - q33 + q44
AA(2,3) = 2.0·(q23 + q14)

AA(3,1) = 2.0·(q13 + q24)
AA(3,2) = 2.0·(q23 - q14)
AA(3,3) = -q11 - q22 + q33 + q44

AAT is the transpose (also the inverse) of AA.
AAT provides ATLAS MRF to ECI rotation.

AAT(i,j) = AA(j,i)
```

4.7 Calculate the precession from J2000 to True of Date

The following equations are from the Explanatory Supplement to the Astronomical Almanac, ed by P. Kenneth Seidelmann, U.S. Naval Observatory, Washington, D.C., University Science Books, 1992, Section 3.21 Precession, pp 99 – 104.

$$T = (JD - 2451545.0) / 36525.0$$

Following values are in units of radians

$$\zeta_A = (2306.2181 \cdot T + 0.30188 \cdot T^2 + 0.017998 \cdot T^3)/s2r$$

$$z_A = (2306.2181 \cdot T + 1.09468 \cdot T^2 + 0.018203 \cdot T^3)/s2r$$

$$\theta_A = (2004.3109 \cdot T - 0.42665 \cdot T^2 - 0.041833 \cdot T^3)/s2r$$

where s2r = 206264.8062470964 (converts arcseconds to radians).

Generate the precession rotation matrix:

$$BB(1,1) = \cos(z_A) \cdot \cos(\theta_A) \cdot \cos(\zeta_A) - \sin(z_A) \cdot \sin(\zeta_A)$$
$$BB(1,2) = -\cos(z_A) \cdot \cos(\theta_A) \cdot \sin(\zeta_A) - \sin(z_A) \cdot \cos(\zeta_A)$$
$$BB(1,3) = -\cos(z_A) \cdot \sin(\theta_A)$$
$$BB(2,1) = \sin(z_A) \cdot \cos(\theta_A) \cdot \cos(\zeta_A) + \cos(z_A) \cdot \sin(\zeta_A)$$

$BB(2,2) = -\sin(z_A)\cdot\cos(\theta_A)\cdot\sin(\zeta_A) + \cos(z_A)\cdot\cos(\zeta_A)$
$BB(2,3) = -\sin(z_A)\cdot\sin(\theta_A)$
$BB(3,1) = \sin(\theta_A)\cdot\cos(\zeta_A)$
$BB(3,2) = -\sin(\theta_A)\cdot\sin(\zeta_A)$
$BB(3,3) = \cos(\theta_A)$

$$BB(i,j) = \begin{matrix} BB(1,1) & BB(1,2) & BB(1,3) \\ BB(2,1) & BB(2,2) & BB(2,3) \\ BB(3,1) & BB(3,2) & BB(3,3) \end{matrix}$$

4.8 Geolocate the laser spots on the Earth ellipsoid at time TQ

Rotate laser unit vector from ATLAS MRF to ECI to ECEF_hat:

$$(xlr,ylr,zlr) = PP\cdot BB\cdot AAT\cdot(xls,yls,zls)$$

Rotate spacecraft position to ECEF:

$$(XR,YR,ZR) = PP\cdot BB\cdot(XI,YI,ZI)$$

Defining the matrix PP as PP(i,j):

$$\text{with } PP(i,j) = \begin{matrix} PP(1,1) & PP(1,2) & PP(1,3) \\ PP(2,1) & PP(2,2) & PP(2,3) \\ PP(3,1) & PP(3,2) & PP(3,3) \end{matrix}$$

And:
$PP(1,1) = \cos\Theta$
$PP(1,2) = \sin\Theta$
$PP(1,3) = 0$
$PP(2,1) =-\sin\Theta$
$PP(2,2) = \cos\Theta$
$PP(2,3) = 0$
$PP(3,1) = 0$
$PP(3,2) = 0$
$PP(3,3) = 1$

Θ = mean sidereal time (GMST)in radians at time TQ

Then:
$XR = PP(1,1)\cdot XI' + PP(1,2)\cdot YI' + PP(1,3)\cdot ZI'$
$YR = PP(2,1)\cdot XI' + PP(2,2)\cdot YI' + PP(2,3)\cdot ZI'$
$ZR = PP(3,1)\cdot XI' + PP(3,2)\cdot YI' + PP(3,3)\cdot ZI'$

Where:
$XI' = BB(1,1)\cdot XI + BB(1,2)\cdot YI + BB(1,3)\cdot ZI$
$YI' = BB(2,1)\cdot XI + BB(2,2)\cdot YI + BB(2,3)\cdot ZI$
$ZI' = BB(3,1)\cdot XI + BB(3,2)\cdot YI + BB(3,3)\cdot ZI$

Setting $DD = PP\cdot DD'$, with $DD'= BB\cdot AAT$,

```
DD′(i,j) = BB(i,1)·AAT(1,j)+BB(i,2)·AAT(2,j)+BB(i,3)·AAT(3,j)

DD(i,j) = PP(i,1)·DD′(1,j)+PP(i,2)·DD′(2,j)+PP(i,3)·DD′(3,j)
```

And then:

```
xlr =  DD(1,1)·xls + DD(1,2)·yls + DD(1,3)·zls
ylr =  DD(2,1)·xls + DD(2,2)·yls + DD(2,3)·zls
zlr =  DD(3,1)·xls + DD(3,2)·yls + DD(3,3)·zls
```

Find the distance along the laser vector from the spacecraft to the Earth ellipsoid (this is the range, R):

```
ap = ae · (1.0 – flat)
alphax = ap · ap · (xlr · xlr + ylr · ylr) + ae · ae · zlr · zlr
betax = 2.0 · ap · ap · (XR · xlr + YR · ylr) + 2.0 · ae · ae · ZR · zlr
gammax = ap · ap · (XR · XR + YR · YR) + ae · ae · (ZR · ZR – ap · ap)
```

where “ae” is the Earth radius at the equator and “flat” is the Earth flattening. Both are Algorithm Parameters.

Then:

```
  dscr = betax · betax - 4.0 · alphax · gammax
  Error condition if dscr =< 0
```

Calculate:

```
  tau1 = (-betax + sqrt(dscr)) / 2.0 / alphax
  tau2 = (-betax - sqrt(dscr)) / 2.0 / alphax
```

Finally, select intersection distance:

```
  if (tau1 < tau2) then
    R = tau1
  else
    R = tau2
  end if
```

The calculation above gives the range “R” at time TQ (courtesy of Charles Webb, code DK000, NASA/HQ). An error condition exists if R < 0.

Put the laser spot on the Earth ellipsoid:

```
(xle,yle,zle) = (XR,YR,ZR) + R·(xlr,ylr,zlr)
```

which becomes:

```
xle = XR + R·xlr
yle = YR + R·ylr
zle = ZR + R·zlr
```

Save the range to the ellipsoid (R) for the next second’s calculation. Calculate the range velocity from up to 10 second’s worth of ranges, where “Nr” is an Algorithm parameter (*PPR:Range_Rate_Average_Points_Strong/Weak*) between 2 and 10:

```
Rsum = Σ( (R($TQ_i$)-R($TQ_{i-1}$))/( $TQ_i$ - $TQ_{i-1}$) ) for i=2,…,Nr

Rdot = Rsum/(Nr-1)
```

While the time difference between TQ in one second and TQ in the previous is most likely equal to one second, it is best not to make that assumption. Care must be taken with the velocity calculation should the time roll over.

4.9 Calculate the velocity of the spot on the Earth ellipsoid

Using the last Nv vectors (xle, yle, zle), estimate the velocity of the laser spot's movement on the Earth ellipsoid. Here Nv is an Algorithm Parameter (*PPR:Velocity_Average_Points_Strong/Weak*):

```
Xsum = Σ((xle(i)-xle(i-1))/(TQ(i)-TQ(i-1)) for i=2,…,Nv
Ysum = Σ((yle(i)-yle(i-1))/(TQ(i)-TQ(i-1)) for i=2,…,Nv
Zsum = Σ((zle(i)-zle(i-1))/(TQ(i)-TQ(i-1)] for i=2,…,Nv

Xdot = Xsum/(Nv-1)
Ydot = Ysum/(Nv-1)
Zdot = Zsum/(Nv-1)
```

Nv (*PPR:Velocity_Average_Points_Strong/Weak*) must always satisfy 2=< Nv =< 10. Care must be taken with the velocity calculation should the time roll over.

4.10 Extrapolate the laser spot's location on the Earth ellipsoid

Extrapolate the laser spot's location up to 1.5 seconds ahead (whatever is needed for the hardware):

```
xle(TQ+dt) = xle(TQ) + dt·Xdot
yle(TQ+dt) = yle(TQ) + dt·Ydot
zle(TQ+dt) = zle(TQ) + dt·Zdot
```

where dt = 0 … 1.5 seconds.

4.11 Determine the geodetic latitude (lat) and east longitude (lon) and choose height and surface type information

```
xleP = xle(TQ+dt)
yleP = yle(TQ+dt)
zleP = zle(TQ+dt)
```

From (xleP,yleP,zleP) calculate latitude and longitude:

```
rle = sqrt(xleP^2 + yleP^2 + zleP^2)
```

```
lon = atan2(yleP,xleP)

tan(latc) = zle/sqrt(xleP^2 + yleP^2)
```
where “latc” is the geocentric latitude.

```
lat = atan( tan(latc)/(1-flat)^2 ).
```

Both the latitude (lat) and the longitude (lon) are needed in degrees. Generally, the atan and atan2 functions return angles in radians. To convert from radians to degrees, multiply the angle in radians by (180/π). The longitude must satisfy: 0 =< lon < 360.

Use the calculated latitude and longitude to pick the correct tile in the 1° x 1° DEM. If the max-min height is > 5.5 km, then go to the next tier down (0.25° x 0.25° tile). If the max-min height in that is > 5.5 km, go to the 0.05° x 0.05° DEM. This height difference may still be greater than 5.5 km, but these Hmin and Hmax values are used in the following calculations regardless. Should a lower tier not exist when expected, an error flag is set, and the Hmin and Hmax of the last available tier are used. See Appendix A for more details. Note that the 5.5 km value is an Algorithm Parameter (*PPR:DEM_Delta_Limit_Strong/Weak*). There are six versions of this parameter, one for each of the six spots, however, all six values MUST BE SET TO THE SAME VALUE as they are determined by hardware limitations. This is a requirement on any future parameter changes.

Use latitude and longitude to reference the correct SRM tile and pick up the surface type (Appendix C).

4.12 Calculate the corresponding range at each extrapolated laser spot

Use the calculated Rdot from section 4.8 to calculate the extrapolated range to the ellipsoid:

```
R(TQ+dt) = R(TQ) + dt·Rdot.
```

4.13 Calculate the solar zenith angle

Use the value of GMST, **Θ,** and XD as given in section 4.3 above.

Calculate the following, in double precision (equations are from the USNO website: http://aa.usno.navy.mil/faq/docs/SunApprox.php, last accessed on July 30, 2019).

Values are initially calculated in degrees but then converted to radians.

```
g = 357.529 + 0.98560028·XD
g = mod(g,360)
$g_r$ = g· π/180
q = 280.459 + 0.98564736·XD
q = mod(q,360)
L = q + 1.915·sin($g_r$) + 0.020·sin(2·$g_r$)
L = mod(L,360)
```

```
    $L_r = L \cdot \pi/180$
    $e = 23.439 - 0.00000036 \cdot XD$
    e = mod(e,360)
    $e_r = e \cdot \pi/180$
    $sindec = \sin(e_r) \cdot \sin(L_r)$
    $cosdec = \sqrt{}\ (1 - sindec \cdot sindec)$
    $ras = atan2(\ \cos(e_r) \cdot \sin(L_r),\ \cos(L_r)\ )$
    $if\ (ras < 0)\ ras\ {+}{=}\ 2 \cdot \pi$
```

Use the atan2 function as it will give you the right quadrant.

Perform a subset of the conversion from Right Ascension (ras) and Declination (dec) to local azimuth and elevation (AZ, EL) where there is no translation to the surface of the earth. For our accuracy needs this is acceptable. All calculations here are done in radians. Latitude (lat) and Longitude (lon) are from section 4.11 **BEFORE** they are converted to degrees and **BEFORE** they are truncated to be used as database indices. The following calculation is done for each spot, although if that is not possible, it can be done for spot #3 only and used for all of the other spots.

```
    $lha = lon + \boldsymbol{\Theta} - ras$
    $if(\ lha < 0\ )\ lha\ {+}{=}\ 2 \cdot \pi$
    $lha = mod(lha,\ 2 \cdot \pi)$
    $cosZ = \sin(lat) \cdot sindec + \cos(lat) \cdot \cos(lha)\ ) \cdot cosdec$
```

cosZ is the cosine of the solar zenith angle. If cosZ < 0.14 (*PPR:Cosine_Solar_Zenith_Angle_Cutoff*) then for purposes of calculating the Range Window, the Algorithms assume it is night. If cosZ >= 0.14 then it is day.

Send both the value of cosZ and the flag indicating day or night to the PCE.

This calculation of Day/Night is independent of the calculation in section 7.6 where the background rate is used to make this determination.

4.14 Calculate the start and stop of the altimetric Range Window

The range (R(TQ+dt), calculated in section 4.10 above) must be modified by the min and max heights taken from the DEM. The actual angle of incidence on the ellipsoid must be used to convert the DEM min and max heights which are normal to the ellipsoid into min and max range modifiers.

Calculate the off-nadir angle of the laser vector, **β**, from the dot product of the two vectors at the current time:

$$\mathrm{Cos}(\beta) = -(XR,YR,ZR)\cdot(xlr,ylr,zlr) \;/\; (\; ||(XR,YR,ZR)|| \cdot ||(xlr,ylr,zlr|| \;)$$

where ||(x,y,z)|| is the magnitude of the vector (x,y,z)

and is = $\sqrt{(x^2 + y^2 + z^2)}$.

And where

$$\begin{aligned} xlr &= xle - XR \\ ylr &= yle - YR \\ zlr &= zle - ZR \end{aligned}$$

and $(XR,YR,ZR)\cdot(xlr,ylr,zlr) = XR\cdot xlr + YR\cdot ylr + ZR\cdot zlr$

Here all values are assumed to be at time TQ.

To get the extrapolated off-nadir angle at time TQ+dt, save the old value of cos(**β**), called cos**β**(TQ_{i-1}), and use it and the currently calculated cos(**β**) to calculate the velocity of this quantity:

$$\cos\beta dot = [\cos\beta(TQ_i) - \cos\beta(TQ_{i-1})]/(\, TQ_i - TQ_{i-1})$$

The extrapolated **cosβ** at time TQ+dT is then calculated from:

$$\cos\beta(TQ+dt) = \cos\beta(TQ) + dt\cdot\cos\beta dot$$

The error in using **β** instead of the actual angle with respect to the nadir line is approximately equal to the difference between geodetic and geocentric latitude and with a maximum difference of about 0.2 degrees which for the onboard Rx Algorithms requirements is small enough to neglect for this calculation. Note, however, that because of the approximate nature of this calculation, the value of **cosβ** must be limited to values between -1.0 and +1.0 before taking the arccosine.

The laser vector intersects the terrain at a slightly different latitude and longitude for nonzero heights. The maximum height above the ellipsoid and its effect on latitude/longitude must be taken into account for the tile overlap in the DEM generation.

As in section 4.11, the top tier (1° x 1°) of the DEM is read and the maximum (Hmax) and minimum heights (Hmin) for the tile corresponding to (lat,lon) are obtained. If the difference of these heights is greater than 5.5 km (*PPR:DEM_Delta_Limit_Strong/Weak*), then the next tier (0.25° x 0.25°) of the

DEM is read. If the difference between these heights is greater than 5.5 km, then the lowest tier (0.05° x 0.05°) is read. This height difference may still be greater than 5.5 km, but these Hmin and Hmax values are used in the following calculations regardless. Should a lower tier not exist when expected, an error flag is set, and the Hmin and Hmax of the last available tier is used.

The minimum and maximum range values are then calculated from:

```
      Rmin = R – Hmax(lat,lon)/cos(β)
      Rmax = R – Hmin(lat,lon)/cos(β)
```

where R is the range calculated in section 4.10 above, and is assumed to be at time TQ+dt.

Rmin and Rmax are then converted to integer units of clock cycles (cc). Rmin is rounded down and Rmax is rounded up. Use the value given in section 3.11 for speed of light (c), and the parameter representing the number of nanoseconds in a clock cycle (*PPR:Clock_Cycles_in_ns,* MUST BE EQUAL TO *ST: Clock_Cycles_in_ns*) to convert Rmin, Rmax from meters to clock cycles:

```
      Rmin(cc) = integer[(2·Rmin(m)/c)·1.E9/clock_cycles_in_ns]
      Rmax(cc) = integer[(2·Rmax(m)/c)·1.E9/clock_cycles_in_ns]+1
```

For the altimetric Range Window, the surface type is used to select the offset, Roffset (*PPR:Range_Window_Offset_Strong/Weak*). This value is given in clock cycles. A padding value, Rpadding (*PPR:Range_Window_DEM_Margin_Strong/Weak*), that is used to compensate for uncertainties in the entire Range Window calculation chain modifies both the RWS and the RWW. This value is in clock cycles, nominally a value corresponding to 250 m, and is a function of both spot number and surface type.

```
      RWS = Rmin(cc) + Roffset - Rpadding
      RWW = Rmax(cc) – Rmin(cc) + 2·Rpadding
```

4.15 Compensate for the different hardware delays between spots

Due to delays in the PCE hardware between the Range Window start and the start of the histogram, the events in the first part of the Range Window do not show up in the histogram. In addition, the end of the histogram is beyond the end of the Range Window (making the effective histogram area the full width of the original Range Window). The bins at the end of the Range Window contain nonzero values that are most likely representative of the background rate, but the time-tags for these last few histogram bins are not be available. This situation is shown graphically in Figure 4.1.

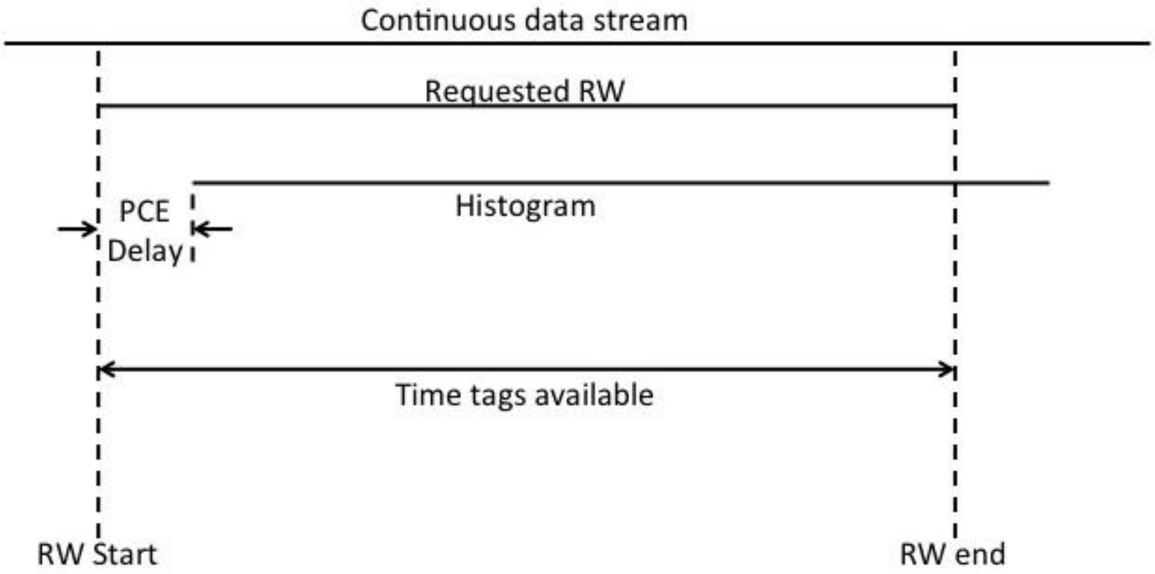


Figure 4.1: PCE Delay

The delays are estimated to be 44 ns for the altimetric strong, 63 ns for the altimetric weak, 34 ns for the atmospheric strong, and 33 ns for the atmospheric weak. Since the Range Window calculations are done in clock cycles the delay values are rounded up to 5 cs for the altimetric strong, 7 cs for the altimetric weak, and 4 cs for the atmospheric strong and weak. By being conservative and rounding up the Range Window starts earlier and ends later ensuring the requested data is captured in the histogram.

For the strong spot, the Range Window start is decreased by the Algorithm Parameter representing the PCE delay amount (*PPR:RW_AltimHist_PCE_Delay_Strong*), and the Range Window width is increased by the corresponding amount. This causes the Range Window to open early enough that the full histogram is populated with data that corresponds to the unaltered Range Window. A similar calculation is performed for the weak spot using the weak spot Algorithm Parameter (*PPR:RW_AltimHist_PCE_Delay_Weak*).

```
RWS = RWS – RW_AltimHist_PCE_Delay_Strong

RWW = RWW + RW_AltimHist_PCE_Delay_Strong
```

The Algorithms calculations that use the altimetric histogram are performed the same as if there were no PCE delays from this point onward until the determination of the hardware bins to send down in telemetry. The PCE delays are applied differently to the telemetry bands than described here for the Range Windows and are rounded to the nearest clock cycle to ensure the closest transformation to the time-of-flight non-PCE delayed environment. These PCE delay values are defined in the ST Algorithm Parameter files. Please see section 7.9 for details on modifying the telemetry bins to correct for the PCE delay.

4.16 Limit the Range Window start and width

The width of the Range Window is checked against minimum and maximum limits. The minimum and maximum limits, used below as Wmax and Wmin (*PPR:RangeWindow_Width_Max_Strong/Weak and PPR:RangeWindow_Width_Min_Strong/Weak*), are Algorithm Parameters that are dependent upon day/night (see section 4.13) and surface type and are given in units of clock cycles. Both the RWS and the RWW are modified if the limits are exceeded.

```
If (RWW > Wmax) then:
      RWS = RWS + (RWW-Wmax)/2
      RWW = Wmax

If (RWW < Wmin) then:
      RWS = RWS + (RWW-Wmin)/2
      RWW = Wmin
```

The width of the Range Window is forced to be an integer multiple of the hardware histogram bin size (two clock cycles) by performing the following:

```
Nrw = integer[(RWW+1)/2]·2
```

and Nrw is given to the hardware as the width of the altimetric Range Window.

To accommodate the hardware, the Range Window movement must be limited. The Range Window start cannot *decrease* from one Major Frame to the next by more than 500 m. This is to keep the current Range Window from potentially encroaching on a previous one. It is the software's task to prevent this from happening. The parameter representing the maximum decrease in the Range Window, **rw_decrease_limit**, is an Algorithm parameter (*PPR:Range_Decrease_Limit_Strong/Weak*) and is given in units of positive clock cycles.

```
If ( (RWS(frame N) - RWS(frame N+1)  ) > rw_decrease_limit )  then
      RWS(frame N+1) = RWS(frame N) – rw_decrease_limit.
```

Range window movements (for RWS) must be made in hardware bin increments. The following calculation ensures that the start and end of the Range Window is always on a hardware bin boundary:

```
Jrw = ceiling[RWS/2]·2
```

and Jrw is given to the hardware as the start of the altimetric Range Window.

4.17 Calculate the start of the atmospheric Range Window

The atmospheric Range Window start (RWC) calculations use the above calculated altimetric Range Window values. An offset based upon the surface type (*PPR:Atmos_Range_Window_Offset_Strong/Weak*) is used to offset the start of RWC from RWS:

RWC = RWS + RWW – 9340 + Roffatm(cc), where

Roffatm(cc) is an Algorithm parameter given in units of clock cycles.
The width of the atmospheric window is 14 km which is 9340 clock cycles (93400 ns).

The atmospheric Range Window must start on a 200 ns boundary, so the following calculation must be performed:

Mrw = ***ceiling***[**RWC/20**]·**20**, which is the start of the atmospheric histogram.

As with the altimetric Range Window, the atmospheric Range Window start cannot *decrease* from one Major Frame to the next by more than 500 m. This Range Window movement is limited similarly.

Additionally, the Altimetric Range Window must lie completely within the Atmospheric Window.

4.18 Limiting the altimetric Range Windows between weak & strong

The difference between the start and end of the weak/strong altimetric Range Windows (for the same track) must be no more than 13.2 km (*PPR: Max_Sep_WeakStrong*), given in clock cycles. If the difference is greater, the end of the trailing Range Window must be chopped.

See pseudo-code below.

```
IF  ( Jrw(strong) =< Jrw(weak) )  THEN

      IF ( (Jrw(weak)+Nrw(weak) - Jrw(strong)) > Max_Sep_WeakStrong ) THEN
            Nrw(weak) = Max_Sep_WeakStrong – ( Jrw(weak) - Jrw(strong) )
            IF(Nrw(weak)=<Wmin) THEN
                  Nrw(weak)=Wmin
                  Jrw(weak) = Jrw(strong) + Max_Sep_WeakStrong – Nrw (weak)
            ENDIF
      ENDIF

ELSE

      IF ( (Jrw(strong)+Nrw(strong) - Jrw(weak)) > Max_Sep_WeakStrong ) THEN
            Nrw(strong) = Max_Sep_WeakStrong – ( Jrw(strong) - Jrw(weak) )
            IF(Nrw(strong)=<Wmin) THEN
                  Nrw(strong)=Wmin
                  Jrw(strong) = Jrw(weak) + Max_Sep_WeakStrong – Nrw (strong)
            ENDIF
      ENDIF

ENDIF
```

*** NOTE that the **Max_Sep_WeakStrong parameter needs to be a multiple of 2 clock cycles (integer multiple of HW bins).**

5.0 Signal Processing of Altimetry Data

5.1 Theoretical probability of success and false alarm

The signal processing must find a segment of the surface echoes in each 700 m ground track (~0.1 seconds) whenever possible. See Appendix D for probability analysis. In particular for most of the ATLAS Design Cases the signal processing must find a segment of the surface with a probability of >=90% while keeping the probability of a false alarm to <=10%.

Signal processing is performed by generating histograms in height (vertical) space. Noise is randomly distributed across the Range Window while signal events are highly correlated in time (range). Terrain roughness and slope tend to spread the return pulse. Histograms that contain returns from sloped surfaces will have a larger signal spread over longer time spans. To keep the spread within acceptable limits, a frame time of 200 shots (=0.02 seconds ≈ 140 m along track) was chosen. For all of the strong and many of the weak spot Design Cases, 200 shots provides a probability of ACQ >= 90% while keeping the false alarm rate =< 10%. The 200 shot time period is called a Major Frame (MF).

The highest probability of acquisition is achieved when all of the signal falls in one histogram bin. To increase the probability that most of the signal will fall in one bin, the Algorithms look at overlapping bins with bin widths varying with the surface type (see section 5.3). This technique, however, doubles the number of histogram bins, which requires raising the bin count threshold to ensure the false alarm rate stays at or below 10%.

To further increase the probability of acquisition while also reducing the probability of false alarm, five (5) consecutive Major Frames are combined into a Super Frame (SF). This effectively increases the integration time without increasing the signal spread.

The initial (hardware) histogram for every Major Frame is generated with an integration time of 200 shots and a bin width of two clock cycles (~20 ns or ~3 meters). This bin width was chosen because it represented the smallest reasonable width for the hardware. This small bin size provides a high resolution and allows accurate calculation of the signal location.

The stable onboard oscillator defines the length of a clock cycle. This time is close to, but not exactly equal to, 10 ns and may change slightly during the course of the mission. The representation of the clock period in nanoseconds is an Algorithm Parameter (*ST:Clock_Cycles_in_ns,* MUST BE EQUAL TO *PPR:Clock_Cycles_in_ns*).

5.2 Calculations done in the hardware

Initial histograms are formed in an FPGA on the Photon Counting Electronics (PCE) board. The software provides the Range Window start time (RWS) and width (RWW) to the hardware. These are the values calculated in section 4.0 from the spacecraft position and pointing, the laser beam vectors, and the Digital Elevation Model.

The start time and width of the Range Window are given to the hardware in units of clock cycles. The maximum width allowed by the hardware is 4000 clock cycles (=2000 hardware bins or ≈ 6 km).

The hardware generates a histogram of all incoming events that fall within the Range Window with the following restrictions:

(1) the hardware can only handle a sustained event rate of up to 12 MHz per spot,
(2) there is a hardware dead-time of 3 ns per channel,
(3) the histogrammer can only count a maximum of two events every clock cycle per fire per channel,
(4) a small percentage (~5%) of the time-tags are double counted.

Restriction #3 implies that while there may be many time tags in a hardware bin period, at most 16 events for a weak spot or 64 events for a strong spot are counted in the hardware histogram bins.

Each strong spot histogram bin is represented by a 14 bit unsigned integer which allows for counts up to 16384 per bin, and each weak spot histogram bin is 12 bits which allows for counts up to 4096 per bin. A 12 MHz background rate gives an average per bin count of 48 over 200 shots. Adding an expected signal of about 1 photoelectron per shot gives the signal bin a maximum value of about 248 counts over 200 shots. Adding an expected signal of about 9 photoelectrons (land ice in ideal conditions) to the 12 MHz background gives a maximum value of about 1848 counts over 200 shots.

The hardware also produces a sum of all events in the altimetric Range Window over 50 shots. There are four of these sums per major frame. These sums are telemetered to the ground but are not used by the FSW.

Changes to the Range Window start and width are not permitted by the hardware during the 200 shot integration time.

5.3 Major Frame calculations (performed in FSW)

The software performs the signal processing on a histogram with a bin size that is larger than the 3 meter hardware bin. Bin widths larger than 3 m are used in an attempt to capture all of the signal in a single bin thus achieving the highest signal to noise ratio. Each surface type has an option (*ST:DRM_for_SW_Bin_Size_Strong/Weak*) to select the bin size from a table (*ST:Bin_Size_Strong/Weak*) or use the unpadded DRM-140 relief. An example of this table is given below. If the option is to use the table then the bin size in the rightmost column is used. If the option is DRM, then the unpadded relief from the DRM-140 is used to establish the bin size. Although the DRM-140 can be used as the bin size for any surface it is expected that the DRM-140 will be used only for the land surface type.

The DRM-140 relief (R) is given in meters. This value must be converted to clock cycles:

```
R(cc) = integer[(2·R(m)/c)·1.E9/clock_period_in_ns]
```

where "c" is the speed of light (given in section 3.11). Because the software bin size must be an even multiple of hardware bins, where each hardware bin is two clock cycles, the software bin size in clock cycles from the DRM-140 unpadded relief must be integer divided by 4 and then multiplied by 4. A maximum and minimum bin size (*ST:sw_bin_size_lower_limit_strong/weak* and *ST:sw_bin_size_upper_limit_strong/weak*) are used to limit the size of the bins regardless of how they are generated. These min/max limits are surface-type dependent. Note, however, that the maximum software bin size can be no larger than 64 hardware bins (hardware constraint).

Surface Type	Option	Bin Size in clock cycles (in meters)
Ocean	Table	8 (12m)
Land	DRM	-----
Sea Ice	Table	8 (12m)
Land Ice	Table	16 (24m)

If the Land Option were "Table", the software bin size for the land would nominally be 48 m.

The request to generate a "coarser" histogram (with wider bins) is presented to the hardware. Coarser histograms (called software histograms) are produced by adding ~3 meter hardware bins to make the larger bin sizes (software bin sizes). The resultant bins are overlapped by one half the software bin size to increase the chance that all signal will fall in a single bin. For example, to make 12 meter bins, hardware bins 0, 1, 2, and 3 are added, and then the summation is shifted by 2 bins and the next 12 meter bin consists of hardware bins 2, 3, 4 and 5 (Figure 5.2). Software bin sizes should always be an even number of hardware bins (this is a requirement on the setting of the Algorithm Parameters). Partial software bins (ones that do not have the full complement of hardware bins) should not be used and should not be counted. FSW is not able to exclude partial bins, but the Simulator excludes them.

The software receives a list of the three software bins with the highest values (along with their values). If there are multiple bins with one of the three highest values, the hardware preferentially returns the indices of the bins that are latest in time first. If the hardware says the maximum bin is beyond the last full software bin, then the Algorithms choose the last full software bin as the max bin. (NOTE: the FSW is not able to do this, so they keep the maximum bin as indicated by the hardware regardless of whether it is full or partial). The full histogram (as delivered by the hardware after combining hardware bins into software bins) is referred to as the **overlapping histogram**.

The software searches for a secondary signal. Any secondary signal must meet two requirements: it must be very large, and it must not be too close to the primary signal. The latter restriction is to make sure it is indeed a separate signal. See Figure 5.1.

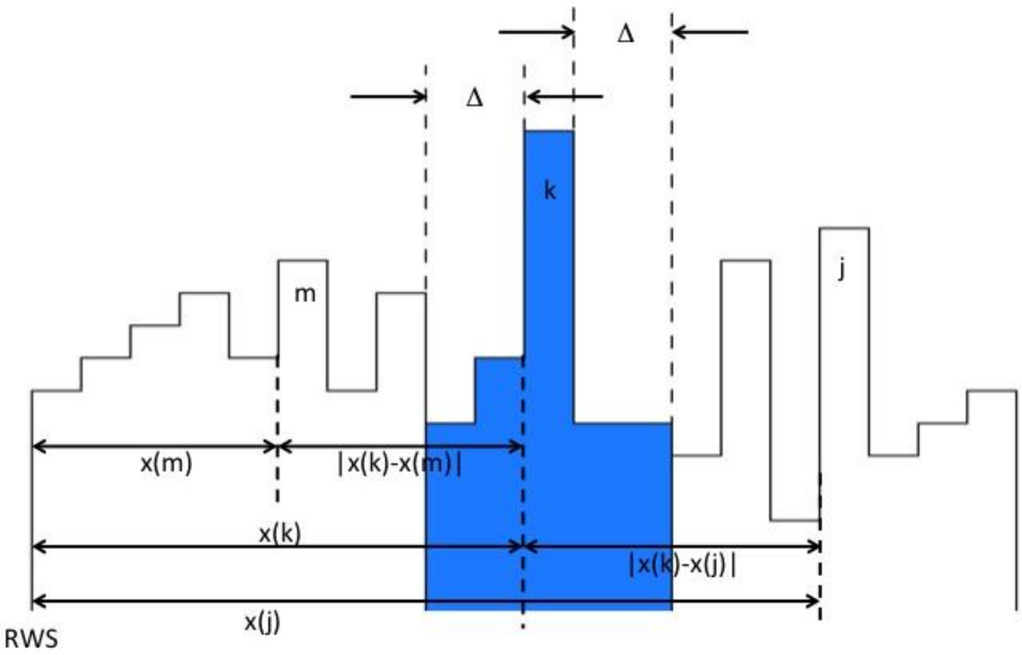


Figure 5.1. Secondary signal detection. The figure shows the software histogram. Primary signal was found in software bin k; the left edge of this bin is at x(k) clock cycles after RWS. Δ is an input parameter defining the exclusion zone around the primary peak. See text for an explanation of j, x(j), k, x(k), m, and x(m). The width of one software bin is a multiple of 2 clock cycles and is surface-type dependent.

The software bin with the highest count is compared to the software bin with the second (and possibly third) highest count. The result of the calculation below is the location of the secondary signal in the software histogram.

```
"k" represents the index to the bin with the highest count in the overlapping
histogram,

"j" represents the index to the bin with the second highest count in the
overlapping histogram,

"m" represents the index to the bin with the third highest count in the
overlapping histogram

x(k) is the delay (in clock cycles) from the start of the Range Window to the
start of software bin k,

x(j) is the delay (in clock cycles) from the start of the Range Window to the
start of software bin j,

x(m) is the delay (in clock cycles) from the start of the Range Window to the
start of software bin m

x(k) = int[k/2] * swbinsize + mod(k,2)*swbinsize/2
x(j) = int[j/2] * swbinsize + mod(j,2)*swbinsize/2
x(m) = int[m/2] * swbinsize + mod(m,2)*swbinsize/2

where swbinsize is the software bin size in clock cycles (surface-type
dependent, the software bin index starts at 0, and "N" is an Algorithm
Parameter specifying the exclusion zone about the primary signal location
(ST:Min_Secondary_SWbin_Separation, in # of software bins).

IF ( abs(x(k) - x(j)) =< N*swbinsize ) THEN
   Discard histogram bin "j" and pick up histogram bin "m"
   IF ( abs(x(k) - x(m)) =< N*swbinsize ) THEN
      Discard histogram bin "m"; there is no secondary signal
   ELSE
      Use histogram bin "m" as the secondary signal location,
      and exit these calculations
   ENDIF
ELSE
   Use histogram bin "j" as the secondary signal location, and exit these
calculations
ENDIF
```

If the TEP_NOT flag (*Knobs:TEP_not_strong/weak*) is set to TRUE, the Transmitter Echo Pulse (TEP) region is excluded from signal processing. This is done by using the TEP Algorithm Parameters that define the start and width of the TEP region in clock cycles (*ST:TEPstart_strong/weak* and *ST:TEPwidth_strong/weak*). The start and end of the TEP region are rounded to a hardware bin boundary. If any part of the software bin with the highest value falls in the TEP region, it is discarded. If any part of the software bin with the next highest count also falls in the TEP region it should also be discarded. If both are discarded, the third highest bin is checked. If any part of this software bin falls within the TEP region it is also discarded. If all three highest software bins are within the TEP region this implies that the MF does not have signal.

If the highest software bin falls within the TEP region (and TEP_NOT is TRUE) and the software bin with the second highest count does not fall in the TEP region, then the second highest count bin is used to check for primary signal. If both the highest and second highest software bins fall within the TEP region, but the third highest does not, then the third highest software bin is used to check for primary signal. This is the only time that the third highest software bin is used by the Algorithms.

If the TEP_NOT flag is set to FALSE then the TEP region is treated no differently than any other region in the Range Window.

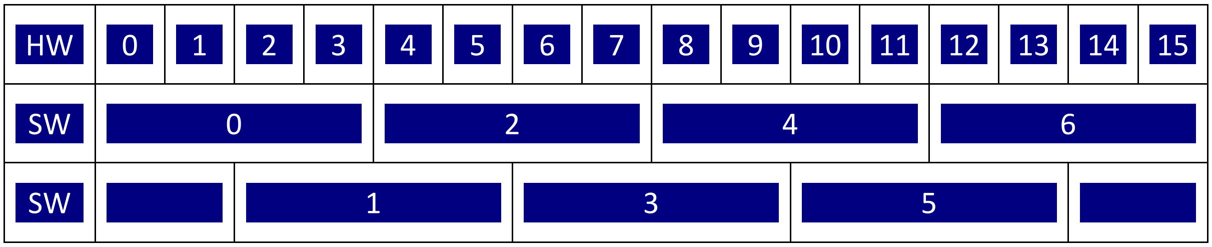


Figure 5.2: Alignment of 3m hardware bins with 12m software bins.

Calculate the number of software bins (nswbin) for the histogram as follows, using integers for all numbers and performing all calculations using integer arithmetic:

1. Count the number of full bins in the overlapping histogram
2. Determine the bin with the highest count, and determine its parity:
    a. If the parity is odd, the result is one-half the number of bins in the overlapping histogram
    b. If the parity is even there are two additional cases:
        i. When the number of bins in the overlapping histogram is even the result is one-half the number of bins in that histogram
        ii. When the number of bins in the overlapping histogram is odd, the result is one more than half the number of bins in that histogram.

The software calculates a threshold for the selected histogram signal processing from the total event count in the hardware histogram (minus the signal count):

The noise is first calculated (in floating point) from:

$$B_{noise} = \frac{C_{hist} - C_{maxbin}}{(\frac{N_{hwbins}}{N_{hw2sw}} - 1)}$$

where:

C*hist* is the total event count in the 3m hardware histogram,

C*maxbin* is the event count in the software histogram bin with the maximum count (after any TEP_NOT rejections),

N*hwbins* is the number of hardware bins in the histogram,

N*hw2sw* is the number of hardware bins per software (coarsened) bin,

B*noise* is then the average noise per software bin.

As an example, see Figure 5.2. Take the total event count in the hardware histogram to be 100. There are 16 hardware bins in this example and 7 software bins with the number of hardware bins per software bins equal to 4. Assuming software bin #3 has the maximum count which is 50. Then the equation above gives:

B*noise* = (100 - 50)/((16/4) – 1) = 50/3 = 16.7.

If there had been 18 hardware histogram bins there would have been 8 software histogram bins, assuming the same number of hardware bins per software bin. Using the same number of total events in the hardware histogram as in the example above and the same number of events in bin #3,

B*noise* = (100 - 50)/((18/4) – 1) = 50/3.5 = 14.3.

Note that this calculation is performed in floating point and not as integer arithmetic.

The multiplier of the standard deviation is then determined. We want the prob(FA) <= pfa_fixed (nominally 0.1 or 10% probability of false alarm) for the entire selected histogram. A table lookup, using the number of software bins in the selected histogram (nswbin) to index into the sigma_multiplier table, is used to get a value called σ_{scale}. Search for the number of bins in the 2nd column of the table that is equal to "nswbin." If there is none equal to "nswbin" then use the next entry greater than "nswbin." Pick the σ_{scale} corresponding to this (see Appendix D for an explanation of how σ_{scale} is calculated).

There is a separate table of σ_{scale} for each of the four surface types. The value of σ_{scale} is limited to a minimum of 2.0 and a maximum of 6.0 with steps of approximately 0.01.

The threshold (τ) is then calculated from:

$$\tau = ceiling[B_{noise} + \ \sigma_{scale} \cdot \ \sqrt{B_{noise}}]$$

where $\sqrt{B_{noise}}$ is the standard deviation of Bnoise for a Poisson distribution. A minimum value of the threshold (*ST:Min_Counts_For_Signal_Strong/Weak*) is set from the Algorithm Parameters (with the nominal min = 10).

If there is at least one software bin with a count at or above the threshold, the Major Frame is marked as having signal. The bin with the highest count above the threshold is called the primary bin and is marked as the signal location.

If more than one bin has a count above the threshold that has survived the ***Min_Secondary_SWbin_Separation*** test above and the TEP test, then the bin with the second highest count is examined. This bin may have the same count as the primary bin, but is presented by the hardware as the second bin. For two or more bins with the same count, the last or latest bin in the histogram with this maximum count is chosen by the hardware as the primary.

A test is made on the secondary bin to ensure that it has a very high probability of being signal:

$$N_\sigma = \frac{bin(k) - B_{noise}}{\sqrt{B_{noise}}}$$

where:
bin(k) is the count in software histogram bin k,
B_{noise} is noise in this software histogram as defined above.

If $N_\sigma > N_{high}$ (an Algorithm Parameter, nominally 5.0, *ST:Sigma_For_Significance_Strong/Weak*) then this bin is assumed to also have signal and this becomes the secondary signal location.

The primary signal location is the bin that has the maximum count. A more accurate location is calculated by picking up the hardware bins that correspond to the software (ie coarsened) signal bin and one software bin on either side. A centroid calculation is then performed on these hardware bins to come up with a signal location, sigloc, that is accurate to within +/- 20 ns. The value sigloc is carried to the telemetry processing in clock cycles from the start of the Range Window.

$$\text{sigloc} = \frac{\sum_{k=-n}^{2n-1}[(i+k)\cdot(\Delta bin(i+k)]}{\sum_{k=-n}^{2n-1}(\Delta bin(i+k))} + 0.5$$

where:
n = number of 3m hardware histogram bins in a software (coarsened) bin,
i = j·n/2,
j = coarsened or software signal bin # (from either the even or odd histogram),
bin(k) = count in hardware bin #k
Δbin(k) = max(0, (bin(k) – (Bnoise/ N_{hw2sw})))
Histogram bins start counting from 0

Using Figure 5.2 as an example, and assuming that we are using the odd histogram with software bin #3 having the maximum count, then n=4, j=3, i=3·2=6, and the summations go from (i+k)=2 to (i+k)=13.

The exception to which hardware bins go into the centroid calculation, is when the software signal bin is either the first or the last bin in the histogram. In this case the centroid calculation is performed using only those hardware bins corresponding to the software signal bin itself.

For any secondary bin that passes the N_{high} test given above, a more accurate signal location is also calculated using the centroid method given above. This information is then also carried to the telemetry processing.

5.4 Super Frame calculations (performed in FSW)

A Super Frame (SF) consists of 5 consecutive Major Frames (MFs). The Super Frame is used to determine the signal location in those Major Frames that have not been initially identified as containing signal, and is used to extend the downlink band in those Major Frames where the signal location lies outside of the region with the rest of the Super Frame's signals.

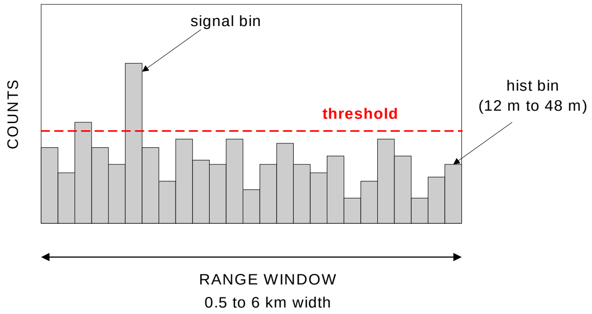


Figure 5.3: Major Frame Software Histogram

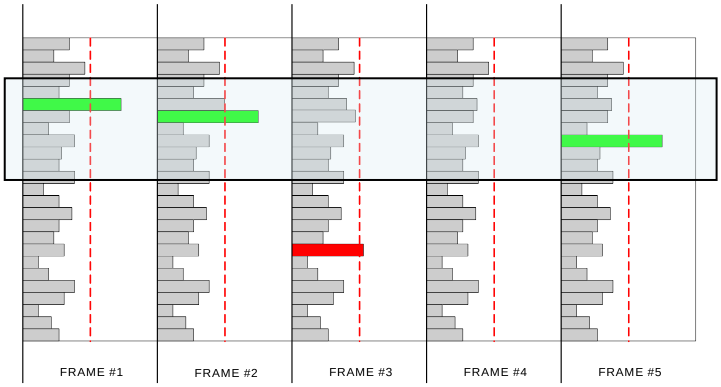


Figure 5.4: Super Frame Histograms with Subwindow. (Green bins are signal. Red bin is false alarm).

Every time a Major Frame signal detection process is completed, the MF is added to the current Super Frame and the earliest Major Frame is dropped from the Super Frame.

The Super Frame is considered to have signal if at least Nsf (*ST:Nsf_Strong/Weak,* nominally 3) of the 5 Major Frames have signal and these signal bins all lie within the subwindow. Nsf must be greater than one and less than or equal to five.

The subwindow width is calculated from the DRM-700 relief. The DRM-700 relief corresponding to the middle Major Frame (frame #3 of 5) is selected and converted to clock cycles. The relief is scaled based upon values selected from a table dependent upon surface type (*ST:DRM_Scaling_Strong/Weak*), and then padded (*ST:Padding_700_Strong/Weak*). The scaling and padding apply additional width to compensate for uncertainties in the relief and in the physical location of the signal. The padding is selected from a table and is a function of both the relief and surface type. The resulting subwindow is checked against a min and max limit (*ST:subwindow_min_strong/weak* and *ST:subwindow_max_strong/weak*) before being used. These min/max limits are surface-type dependent parameters and are distinct from those parameters used to limit the telemetry bands. Note that the start of the subwindow can be negative with respect to MF #3's Range Window start.

Examples of scaling and padding parameter tables with latest set of values as of v6 of the Algorithm Parameter files (Appendix F):

Surface Type	Scale
Ocean	1x
Land	2x
Sea Ice	1x
Land Ice	2x

	--------------------- Padding	-------------------------		
Relief(R)	Ocean	Land	Sea Ice	Land Ice
R=< 189m	+/- 15 m	+/- 24 m	+/- 15 m	+/- 24 m
189m < R =< 567m	+/- 15 m	+/- 140 m	+/- 140 m	+/- 140 m
567m < R =<1323m	+/- 15 m	+/- 210 m	+/- 210 m	+/- 210 m
1323m < R	+/- 15 m	+/- 510 m	+/- 510 m	+/- 510 m

The calculations required to determine if the Super Frame has signal are given below in steps 1 through 8. All of the calculations are assumed to be in clock cycles. Nsf is the number of MFs needed to contain signal in order to declare that the SF has signal. If the number of MFs in the SF with signal is less than Nsf, it is assumed that the SF has no signal and calculations 1-8 are skipped.

1- The offsets between the start of each of the 5 Range Windows in this SF must be determined. Find the minimum value of the Range Window Start (Jrw) from the five major frames and call this Jrw0. For each MF, calculate the offset between the start of the Range Window and Jrw0: Jrw_offset(i)=Jrw(i)-Jrw0.

   Example (all numbers in clock cycles):
   Jrw = 337666, 337666, 337666, 337666, 337664
   Jrw0 = 337664
   Jrw_offset = 2, 2, 2, 2, 0

2- Put the signal location for all of the MFs with signal into an array, along with the MF number (values of "i" from 1 to 5). Correct the signal locations calculated in step #1 above by adding the offsets:
   SigLocCor(i)=SigLoc(i)+Jrw_offset(i).

   Example:
   SigLocCor = 195.8, 195.4, 191.8, 199.8, 199.4 (i=1,2,…5)

3- Sort the array from lowest SigLocCor(i) value to highest, where the lowest SigLocCor(i) value is assigned index k=1. Keep the frame number (i=1 to 5) correctly associated with its SigLocCor(i) location during the sort.

   Example:
   SigLocCor_ordered = (3,191.8), (2,195.4), (1,195.8), (5,199.4), (4,199.8) (k=1,2,…5)
   where the arrays are (i,sigloccor)

4- Compute the difference between the sorted corrected signal location values for k=Nsf and k=1. Call it Diff(1).

   Example: Diff(1)=sigloccor(k=3) – sigloccor(k=1) = 195.8 – 191.8 = 4.0

5- If there are more MFs in the sorted array, calculate the difference between the corrected signal location values at k=Nsf+1 with k=2. Call it Diff(2).

Example: Diff(2) = sigloccor(k=4) – sigloccor(k=2) = 199.4 – 195.4 = 4.0

6- If there are more MFs in the sorted array, then calculate the difference between the signal location k=Nsf+2 with k=3. Call it Diff(3).

Example: Diff(3) = sigloccor(k=5) – sigloccor(k=3) = 199.8 – 195.8 = 4.0

7- Continue until you have reached the end of the sorted array.

Example: we have reached the end.

8- Pick Q such that Diff(Q) = min(Diff(k)) over all "k". If this difference is less than the subwindow width, then the SF has signal. If Diff(k) = min(Diff(k)) for more than one value of k, select the lowest value of k to use.

Example: Diff(1)=Diff(2)=Diff(3) ➔ Q = 1. If Diff(1) < subwindow_width, then SF has signal.

If the SF is determined to have signal in step #8 above, the location of the subwindow is then determined by centering the subwindow about the midpoint of the two signal locations corresponding to the Major Frames that went into the calculation of Diff(Q). If the subwindow start is negative with respect to the minimum Range Window start, then the subwindow start is chopped at the start of the minimum Range Window. (Note that to do the comparison of the subwindow start with the Range Window, the start of the subwindow must be temporarily adjusted by the corresponding PCE delay before the comparison is made because the subwindow is in histogram space but the Range Window is not). Similarly if the subwindow end is beyond the end of the maximum Range Window end it is chopped to end at the end of the maximum Range Window end. (And similar to the adjustment to the subwindow start, the subwindow end must be temporarily adjusted by the corresponding PCE delay before comparing to the maximum Range Window end).

If signal is found in the current MF (MF #3), the location of this signal is the primary (and possibly secondary) signal location. If primary signal is found in the SF and either (a) the MF does not have signal or (b) the MF primary signal location is outside the SF subwindow, then SF signal location is determined using data for the two Major Frames that are closest to MF #3 and have signal *that is located within the subwindow*. If possible, MFs with signal on either side of MF #3 are used. The signal location of the current frame is then a weighted linear interpolation from these two chosen MFs with signal.

The specific method for determining the signal location for a MF with no signal in the subwindow is given in steps 1-7 below. Note that in all of these calculations the SigLocCor values are used.

1- If there is signal within the subwindow in frames 2 and 4 average their signal locations:
SigLocCor(3)=[SigLocCor(2)+SigLocCor(4)]/2.

2- Otherwise, if there is signal within the subwindow in frames 1 and 4 take the weighted average:
SigLocCor(3)=SigLocCor(1)/3 + 2*SigLocCor(4)/3.

3- Otherwise, if there is signal within the subwindow in frames 2 and 5 take the weighted average:

SigLocCor(3)=2*SigLocCor(2)/3 + SigLocCor(5)/3.

4- For Nsf=2, if there are only 2 MFs with signal within the subwindow, both on the same side of MF #3, then calculate:
SigLocCor(3)=2*SigLocCor(4)/3 + SigLocCor(5)/3, or
SigLocCor(3)=SigLocCor(1)/3 + 2*SigLocCor(2)/3.

5- For Nsf=2, if there is signal in frames 1 and 5 ONLY, then average their signal locations:
SigLocCor(3)=[SigLocCor(1)+SigLocCor(5)]/2.

6- Remove the offset from the calculated signal location: SigLoc(3) = SigLoc(3) – Jrw_offset(3).

7- Verify that the calculated signal location, SigLoc(3), falls within the Range Window for MF#3. If it does not then discard it. If it does then send it on to the telemetry processing.

For MFs which either do not have signal, or whose signal lies outside the SF subwindow, the calculated SF signal location becomes the tertiary signal location for this MF and it is carried to the telemetry processing along with the primary and possibly secondary signal location information. A Super Frame signal can exist even if there is no tertiary signal location. This can occur if the signal location as calculated above lies outside the Range Window for MF#3.

6.0 Atmospheric Processing

6.1 Atmospheric histogram calculations in hardware

The hardware calculates two atmospheric histograms per track: one from the strong spot data and one from the weak spot data. The histogram data is collected over a 200 shot interval (the same as for the altimetric histogram) but using a bin size of ~30 meters. The vertical span of the histogram is 93.4 µs (~ 14 km). The nominal offset of the window from the minimum DEM height is zero.

At the end of the 200 shot interval, the atmospheric histograms are given to the Flight Software.

6.2 Atmospheric histogram calculations in software

The Flight Software directs the hardware to add the histograms from two consecutive Major Frames to generate a 400 shot histogram. The calculation is done every 200 shots, for both the strong and the weak spots. Both the strong and weak spot histograms are used by the Flight Software to determine the cloud thickness. Although a 400 shot histogram is calculated at every major frame, only every other atmospheric histogram from the strong spot is downlinked. The weak spot histograms are discarded after use.

If the start of the atmospheric histogram (Mrw) changes from one major frame (200 shots) to the next, one of the 200 shot histograms must be shifted before it can be added to the other. Changes to the histogram start are always done in 200 ns (~30 m) increments so that the histogram bins can always be aligned. The atmospheric histogram with bins closest to the surface is retained, while the bins highest in the atmosphere, that do not overlap between histograms, are discarded.

For example, if the first 200 shot atmospheric histogram has a range delay (Mrw1) of 3.0 milliseconds and the second 200 shot atmospheric histogram has a range delay (Mrw2) of 3.0004 milliseconds, the first 2 bins of histogram #1 are dropped. Bins from histogram #1 are added to bins from histogram #2 in the following way:

hist_combined(k) = hist1(k+2) + hist2(k)

The resulting histogram is 93.4 µs in length. The bottom (last) two bins of the resulting histogram only contain information from histogram #2. The offset between the start of the two atmospheric histograms is sent down in the telemetry.

6.3 Calculations to determine if clouds are too thick to get surface returns.

If the clouds are too thick, surface returns are very unlikely. An algorithm (the Palm Algorithm) to determine if the clouds are too thick was generated by Steve Palm. This algorithm uses the 400 shot atmospheric histogram. The calculation is performed separately for the weak and strong spots. This algorithm performs best during Day, so it will only be used when the noise rate is greater than some minimum value, which is an Algorithm Parameter (*Knobs:Day_Night_Threshold_Strong/Weak*).

a- Compute the mean (μ) and the standard deviation (σ) of the atmospheric histogram:

   i- Sum all of the histogram bins, excluding certain bins, depending upon CXbin (the Algorithm Parameter *ST:Cloud_Bins_Exclude_Strong/Weak*). If CXbin = 0, exclude the bin with the maximum count from this sum. If CXbin = M > 0, then exclude from this sum the bin with the maximum count and M bins on either side of that maximum bin. If CXbin < 0, then do not exclude any bins from this sum.

   If the bin with the maximum count lies too close to the ends of the histogram (either the start or the end) then some of the bins that would have normally gotten excluded fall outside the histogram. In this case only use those histogram bins that fall within the histogram. There is no other adjustment that needs to be made. So for example, if CXbin=2 (where bin #1 is the first histogram bin), and the bin with the maximum count is bin #2 of the histogram, then exclude bins 1 through 4 of the atmospheric histogram.

   ii- Divide this sum by the number of histogram bins (N) used. This is the mean (μ).
   iii- Compute the standard deviation (σ) as the square root of the mean.

b- Compute a cloud threshold T:

$$T = \mu + F * \sigma$$

   where F is an Algorithm parameter (*ST:Cloud_Scale_Factor_Strong/Weak*).

c- Request from the hardware the bin numbers with counts above this threshold. Sum up the counts from all of these bins, excluding bins above bin number “LBin” (*ST: LBin_Strong/Weak*). Excluding bins above LBin will exclude bins below ground. Call this sum S.

   If there are two peaks in the atmospheric histogram with the same maximum count, then select the bin that is closest to the ground (ie has the largest range) similar to what is done for the altimetric histogram maximum bin selection.

d- If $S > S_C$ (the opaque cloud threshold, *ST:Cloud_Threshold_Strong/Weak*), the clouds are too thick to determine the surface.

Based on simulations that use ICESat-1 data, the values chosen for the parameters used in the Palm (Cloud) Algorithm eliminate about half of the cloud returns (about 20% of the total shots), and eliminate < 0.8% of shots where there might be a chance of detecting the ground.

How the information from this calculation is used is explained in the Section 7.

7.0 Selecting What to Telemeter

7.1 Choosing what gets telemetered

To allow for changes in the telemetry while on orbit, there are Rx Algorithms Knobs that provide the capability to change the conditions under which data is downlinked. The flowchart given in Figure 7.1 describes the flow of the software decisions using these knobs (which are Algorithm Parameters). The baseline telemetry decisions for the Rx Algorithms were developed in coordination with the Project Scientists and are described here:

> <u>OCEAN:</u> Telemeter STRONG spot (Day or Night) where Major Frame shows signal. No data is downlinked for the WEAK spot under any condition, and none for the STRONG if the Major Frame does not show signal.
>
> <u>LAND:</u> Telemeter any data where the Major Frame shows signal (STRONG, WEAK, NIGHT, DAY). If Major Frame does not show signal, but the Super Frame shows signal then telemeter data (STRONG, WEAK, NIGHT, DAY). If the Major Frame and Super Frame both do not show signal, then:
>
>> \- if STRONG DAY and the Palm Algorithm indicates thick clouds, then do NOT telemeter data,
>> \- if STRONG DAY and the Palm Algorithm does not indicate thick clouds, do telemeter data,
>> \- if STRONG NIGHT, do telemeter data,
>> \- if WEAK DAY or NIGHT, do NOT telemeter.
>
> <u>SEA ICE and LAND ICE</u>: Telemeter any data where the Major Frame or Super Frame shows signal (STRONG, WEAK, NIGHT, DAY). If the Major Frame and Super Frame both do not show signal (STRONG, WEAK), then:
>
>> \- if Night, telemeter data.
>> \- if Day and Palm algorithm indicates thick clouds, do NOT telemeter data.
>> \- if Day and Palm algorithm does not indicate thick clouds, telemeter data.

In January 2014 the Project Scientists modified the above baseline slightly for the Land Ice surface type. Now for the Land Ice Strong spot, the telemetry band is always sent down for both Day and Night, and the Palm (Cloud) Algorithm is never called.

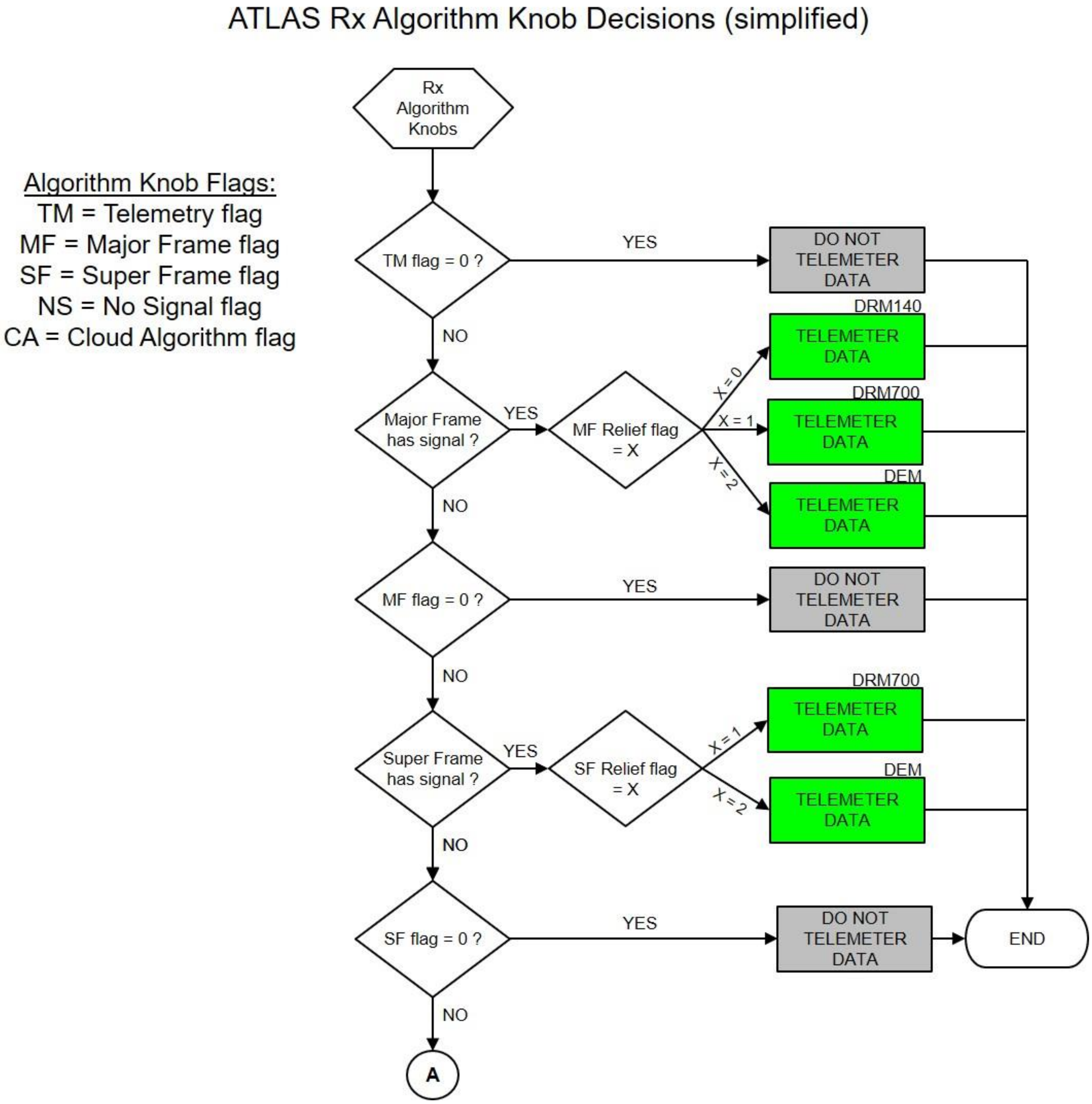


Figure 7.1a: Rx Algorithms Knobs for telemetry flow. There are 8 separate flowcharts, each corresponding to one of the four surface types and either a weak or strong spot.

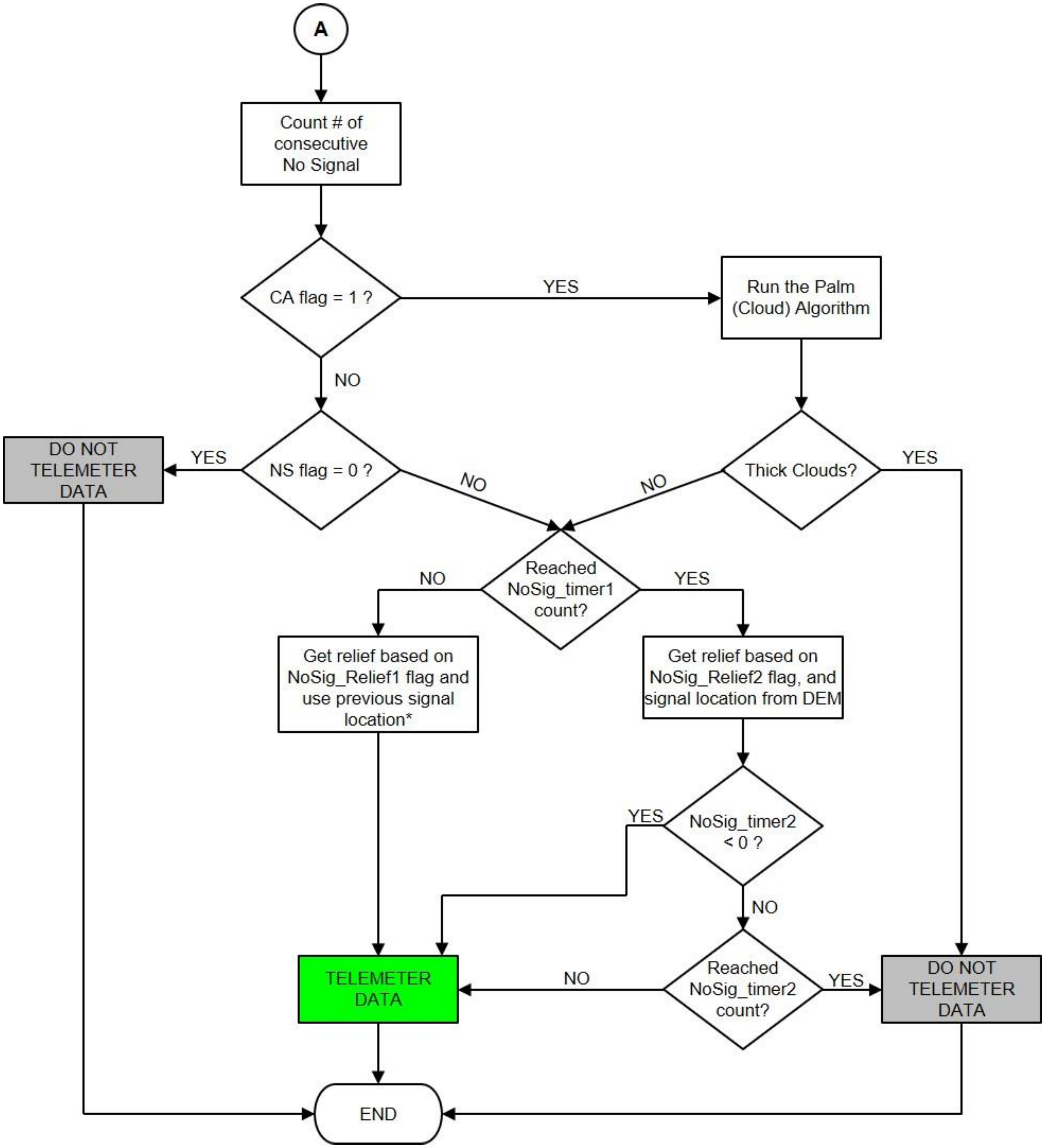

A
Count # of consecutive No Signal
CA flag = 1 ?
YES
Run the Palm (Cloud) Algorithm
NO
DO NOT TELEMETER DATA
YES
NS flag = 0 ?
NO
NO
Thick Clouds?
YES
Reached NoSig_timer1 count?
NO
YES
Get relief based on NoSig_Relief1 flag and use previous signal location*
Get relief based on NoSig_Relief2 flag, and signal location from DEM
YES
NoSig_timer2 < 0 ?
NO
TELEMETER DATA
NO
Reached NoSig_timer2 count?
YES
DO NOT TELEMETER DATA
END

Figure 7.1b: Rx Algorithms Knobs for telemetry flow. This is the No Signal section of the flowchart. There are 8 separate flowcharts, each corresponding to one of the four surface types and either a weak or strong spot. Note that this is a simplification of the actual No Signal part of the Rx Algorithm and doesn't reflect what happens when the surface type changes.
If the coastline relief flag (*ST:Coastline_Relief_Flag_Strong/Weak*) is TRUE, the coastline bit in the SRM is ON and the latitude test passes [ie latitude >= +60 deg (*ST:Coastline_Relief_North_Strong/Weak*) or latitude =< -60 deg (*ST:Coastline_Relief_South_Strong/Weak*)], then the relief is calculated from the maximum of the coastline relief parameter value (*ST:coastline_relief_Strong/Weak*) and the DRM-700 relief.
Note (*): if no previous signal, use DEM for telemetry band location. See Appendix J for a detailed flow of the telemetry logic. If there are discrepancies between Appendix J and either this figure or the text in section 7.4, the flowcharts in Appendix J take precedence.

7.2 An alternate knobs table for calibration and special interest targets

There are two Knob parameter files resident on each PCE board computer at all times. Only one, however, is in use at any given time. One of the files is in use for most of the orbit, while the other, the alternate knobs, is strictly for calibration (ocean scans) and targets of opportunity (TOOs). The switch between the two parameter files must be done by command during Science Mode.

7.3 Calculating the telemetry band for the primary signal location

The telemetry band width is normally calculated from the relief. Depending upon the Algorithm Knob settings, the relief comes from the DRM-140, DRM-700, DEM or is set based on the coastline bit in the selected SRM tile and the latitude. Under nominal Algorithm Knob settings, if the coastline relief flag (*ST:Coastline_Relief_Flag_Strong/Weak*) is FALSE, and if the coastline bit is OFF (=0) or the coastline bit is ON (=1) but the latitude test fails, then the DRM-140 is used for relief when signal is found in the Major Frame, and the DRM-700 is used for relief when signal is found only in the Super Frame:

R(MF) = DRM-140-relief and R(SF) = DRM-700-relief

If the coastline relief flag (*ST:Coastline_Relief_Flag_Strong/Weak*) is TRUE, the coastline bit in the SRM is ON and the latitude test passes [ie latitude >= +60 deg (*ST:Coastline_Relief_North_Strong/Weak*) or latitude =< -60 deg (*ST:Coastline_Relief_South_Strong/Weak*)], then the relief is calculated from the maximum of the coastline relief parameter value (*ST:coastline_relief_Strong/Weak*, nominally 200 m) and the DRM-700 relief:

R = max(coastline-relief, DRM-700-relief).

This method of finding the max of the two reliefs is only used if the Algorithm Knob settings indicates to use the DRM-140 or DRM-700 for relief. If the Algorithm Knob settings indicate to use the DEM, then the Range Window width (Nrw) is used to set the relief, regardless of the coastline settings.

If the relief comes from the DRM or the coastline relief parameter, then the relief is first scaled. For non-coastline areas the scaling *(ST: DRM_Scaling_Strong/Weak)* is based on the surface type. The scaling comes from a table lookup (shown below). For coastline where the latitude test is true, the scaling factor is a separate coastline parameter (*ST:coastline_scaling_strong/weak*), nominally 2x, and it is not based on surface type. If the relief comes from the DEM, then no scaling is applied.

Margin (*ST:Padding_140_Strong/Weak* and *ST:Padding_700_Strong/Weak*) must be added to the relief to compensate for the relief inaccuracy. For the DRM-140 and DRM-700, the padding is chosen from a table based upon both the relief value and the surface type (same table as in Section 5.4). There are separate tables for the DRM-140 and DRM-700. If the relief comes from the coastline relief parameter, then the DRM-700 padding is used. Padding (margin) is doubled and then added to the scaled relief to make the telemetry band width. This band width is converted to clock cycles and an upper limit is imposed. If the band width is larger than the maximum band limit (*ST:Band_Hi_Limit_Strong_Strong/Weak*), then an equal number of clock cycles is removed from both ends of the band to ensure its width is at or under the maximum band limit.

When using the DEM relief, the Range Window width is used (Nrw) and it is not be scaled or padded.

The tables and values listed here are Algorithm Parameters. Values given are representative only.

Relief Scaling:

Surface Type	Scale
Ocean	1x
Land	2x
Sea Ice	1x
Land Ice	2x

A Telemetry Band Offset Table (*ST:Offset_140_Strong/Weak* and *ST:Offset_700_Strong/Weak*) contains offsets for draping the telemetry band over the signal. This table is based upon the surface type and is given in clock cycles. An offset of zero implies that the center of the telemetry band is placed over the signal location. A positive offset implies the center of the telemetry band is placed further down in the histogram (longer ranges). Likewise, a negative offset implies the center of the telemetry band is placed higher in the histogram (shorter ranges). There are separate Telemetry Band Offset tables for the DRM-140 and the DRM-700.

If the relief comes from the coastline relief parameter, then the telemetry band offset is based on the DRM-700 table. No offset is used if the relief comes from the DEM.

Telemetry Band Offset:

Surface Type	Offset
Ocean	0
Land	0
Sea Ice	0
Land Ice	0

The hardware histogram bins selected to be telemetered are calculated from:

TLM_bandwidth_hwbins = ***integer*** [TLM_bandwidth(clockcycles) / 2] + 1
TLM_start_hwbin = Signal_location_hwbin – ***integer***[TLM_bandwith_hwbins/2]
+ TLM_band_offset
TLM_stop_hwbin = TLM_start_hwbin + TLM_bandwidth_hwbins – 1

7.4 What to do when neither the Major Frame nor the Super Frame have signal

If the Major Frame does not show signal and the Super Frame does not show signal then the last signal location is used for a period of time (*Knobs:Nosig_Timer1_Strong/Weak*), assuming that the knobs are set to still telemeter data, and assuming that the last signal location is within the current MF #3's Range Window. The value of the last signal location (relative to the fire and not relative to the start of the Range Window) must be used, as the Range Window start may move. NoSig_Timer1 must be >=0 (this is state 1). Note that if both the primary and tertiary signals are present in the last MF, then the tertiary signal location is used as that is less likely to be a false alarm. If the last signal location is outside of the current MF #3's Range Window then the Algorithms switch immediately to state 2, and use NoSig_Timer2. See the paragraph below on state 2. (Note that to do the comparison of the last signal location with the Range Window, the signal location must be temporarily adjusted by the corresponding PCE delay before the comparison is made because the signal is in histogram space, but the Range Window is not).

While in no-signal state 1, the relief (R) is calculated as described in section 7.3, except that the base for the relief (DRM-140, DRM-700 or DEM) is specified by the no-signal state 1 relief Algorithm Parameter (*Knobs:Nosig_Relief1_Strong/Weak*).

Normally, a scaling factor is applied to the relief to calculate the telemetry band (see bullets below).

However, if the DEM is used for the telemetry band width, then the telemetry band width is the Range Window Width or the maximum allowed telemetry band width (*ST:Band_Hi_Limit_Strong/Weak*), whichever is smaller, and the telemetry band is not scaled.

If the DEM is NOT used for the telemetry band width, the scaled relief is calculated as follows:

- If the coastline relief flag (*ST:Coastline_Relief_Flag_Strong/Weak*) is TRUE, the coastline bit in the SRM is ON and the latitude is poleward of the critical latitude (*ST:Coastline_Relief_North/South_Strong/Weak*), then the scaling factor is the coastline scaling Algorithm Parameter (*ST:coastline_scaling_strong/weak*). The scaled relief (RS) is then given by:
  RS(strong) = coastline_scaling_strong · R or RS(weak) = coastline_scaling_weak · R,
  where R is the relief calculated in section 7.3.
- Otherwise, the scaling factor is the no signal scaling Algorithm Parameter (*Knobs:Nosig_Scale1_Strong/Weak*). The scaled relief (RS) is then given by:
  RS(strong) = Nosig_Scale1_Strong · R or RS(weak) = Nosig_Scale1_Weak · R,
  where R is the relief calculated in section 7.3.

If the DEM (Nrw) is used for the relief, then no padding is performed. Otherwise, padding is based on what is used to calculate the relief and the relief value itself. The padding (*ST:Padding_140_Strong/Weak* and *ST:Padding_700_Strong/Weak*) is doubled and then added to the scaled relief. If the relief comes from the coastline relief parameter, then the DRM-700 padding is used.

In the case where the Range Window changes so all or part of the telemetry band lies outside the new Range Window, the telemetry band is shifted the minimum amount required to put it entirely in the Range Window.

If there has been no signal found since the start of Science Mode, and the knob settings allow telemetry to be sent for the no signal case, then the Rx Algorithms act as if the NoSig_Timer1 period of time has ended and transition immediately into the actions described in the paragraph below. Note that when the Algorithms first start there is insufficient data to do a SF analysis until five MFs have accumulated, so the first possible SF signal is in MF 3. If neither MF nor SF signal is found in this MF, the Algorithms shift immediately to the timer2 state, regardless of other considerations.

After the NoSig_Timer1 period has expired (counted down to zero from a positive value) or if the surface type changes while the NoSig_Timer1 is still positive or if the old signal location falls outside of the Range Window, and if signal has still not been found, the Algorithms transition to the NoSig_Timer2 (*Knobs:Nosig_Timer2_Strong/Weak*) case (state 2), where the Range Window start and width is used to determine the signal location. The signal location is calculated from:

sigloc = Jrw + scale • (Nrw).

Nominally "scale" (*Knobs:NoSig_DEM_fraction_Strong/Weak*) is 0.5 which results in using the midpoint of the DEM as the signal location. The relief (R) is calculated as described in section 7.3, except that the base for the relief (DRM-700 or DEM) is specified by the no-signal state 2 relief Algorithm Parameter (*Knobs:Nosig_Relief2_Strong/Weak*). The DRM-140 relief is NOT an option for the no-signal state 2 relief.

Similar to the no-signal state1, a scaling factor is applied to the relief to calculate the telemetry band unless the DEM is used for the telemetry band width. When the DEM is selected the telemetry band width is the Range Window Width or the maximum allowed telemetry band width (*ST:Band_Hi_Limit_Strong/Weak*), whichever is smaller, and the telemetry band is not scaled. Unlike the no-signal state 1, the no-signal state 2 scaling Algorithm Parameter (*Knobs:Nosig_Scale2_Strong/Weak*) is the only scaling factor used to calculate the scaled relief, even when the coastline relief is used.

RS(strong) = Nosig_Scale2_Strong · R or RS(weak) = Nosig_Scale2_Weak · R,
where R is the relief calculated in section 7.3.

If the DEM (Nrw) is used for the relief, then no padding is performed. If the DRM-700 or the coastline relief is used, then the DRM-700 padding (*ST:Padding_700_Strong/Weak*) is doubled and added to the scaled relief.

Again, if the NoSig_Relief2 parameter indicates that the DEM is used for relief (which is the nominal case) then there is no scaling and no padding.

Because the signal location (sigloc) for the NoSig_Timer2 case is calculated from an absolute position relative to the start of the Range Window, and not from the altimetric histogram, its location must be artificially modified by the PCE delay to put it in the same reference system as the altimetric histogram. This will allow it to be correctly positioned when all of the telemetry band histogram bin information is adjusted for the PCE delay *(ST:RW_AltimHist_PCE_Delay_Strong)* prior to being sent to the hardware.

```
    sigloc = sigloc – RW_AltimHist_PCE_Delay_Strong
```

A similar calculation is done for the weak spot using its PCE delay (*ST:RW_AltimHist_PCE_Delay_Weak*).

After the NoSig_Timer2 (*Knobs:Nosig_Timer2_Strong/Weak*) has expired with no signal having been found, then no more telemetry is sent until signal is again found. At this point the system transitions from state 2 to state 3. The exception to this rule is if NoSig_Timer2 is negative. The telemetry continues to be sent, using the NoSig_Relief2 and NoSig_Scale2 parameters to select the relief and scaling, until signal is found again.

The Telemetry Band Offset explained in section 7.3 above shall be applied for the NoSig_Timer1 case but NOT for the NoSig_Timer2 case.

If the Algorithms are in a no-signal state (state 1, state 2, or state 3), they remain in a no-signal state until signal is detected. The following applies whenever the surface type changes while in a no-signal state:

- If the instrument is in state 1 and the surface type changes, the Algorithms switch to state 2 if N <= NT2 or NT2 < 0, or to state 3 if N > NT2 and NT2 >= 0.
- If the instrument is in state 2 and the surface type changes, the Algorithms remain in state 2 if N <= NT2 or NT2 < 0, or switch to state 3 immediately if N > NT2 and NT2 >= 0.
- If the instrument is in state 3 and the surface type changes, the Algorithms remain in state 3 if N > NT2 and NT2 >= 0, or switch to state 2 if N <= NT2 or NT2 < 0.

where N = the number of contiguous MFs that the algorithm has been in a no-signal state, and NT2 = NoSig_Timer2 for the new surface type.

The period of time and scaling parameters are dependent upon surface type and day/night and are given in number of Major Frames. The scaling parameters must be greater than or equal to 1. NoSig_Timer1 must be >= 0. NoSig_Timer2 can be any number >= -1. See the flowchart in Figure 7.1 for a graphical overview of this logic. See Appendix J for a detailed flow of the telemetry logic. If there are discrepancies between Appendix J and either this text or Figure 7.1, the flowcharts in Appendix J take precedence.

7.5 Transmitter Echo Pulse (TEP)

The transmitter echo pulse (TEP) is a received pulse from the laser transmitted pulse that travels through part of the ATLAS system, never leaving the instrument. It is used to calibrate the instrument and arrives approximately 30 nanoseconds after the laser fires. Signal from the TEP is measured by the PCE hardware if it falls within a Range Window just like any received event. The expected return rate is low enough so that the Rx Algorithms signal processing cannot see it. If the return rate gets high enough for the Rx Algorithms to see, it will adversely interfere with the surface echo finding.

During Science Mode, if the TEP falls within the Range Window, the events in the TEP region are telemetered to the ground, provided that either (a) the TEP flag (an Algorithm Parameter, *Knobs:TEPflag_strong/weak*) is on and indicates it should be telemetered, or (b) the downlink telemetry band #2 is not otherwise being used, or (c) the TEP telemetry band was combined with another telemetry band that is being downlinked. If either (a) or (b) is true, then the TEP is telemetered even if the other Knobs Parameters indicate that no data should be sent. See section 7.8 for more detail on the priorities and how to select the data for the telemetry bands. Unless it is combined with another signal region, the TEP is normally sent in the downlink telemetry band two.

Determining if the TEP is within the Range Window is described by the pseudo-code below. This code sets the iTEP_flag to 1 if at least half of the TEP is in the Range Window:

```
TEPsigloc = TEPstart + 0.5*TEPwidth
Q1 = mod(Jrw, FireInterval)
Q2 = mod(Jrw+Nrw, FireInterval)
iTEP_flag=0
if(Q1.le.Q2) then
   if(TEPsigloc.ge.Q1.and.TEPsigloc.le.Q2) iTEP_flag=1
else
   if(TEPsigloc.ge.Q1.or.TEPsigloc.le.Q2) iTEP_flag=1
endif
```

where TEPstart and TEPwidth are Rx Algorithm Parameters (*ST:TEPstart_strong/weak* and *ST:TEPwidth_strong/weak*), Jrw is the Range Window start as defined previously, and Nrw is the Range Window width. FireInterval is 10000 hardware clock cycles and represents the interval between laser fires.

If at least half of the TEP falls within the Range Window, then the following pseudo-code picks up the start and the end of the telemetry band to send down.

```
if(iTEPflag.eq.1) then

   TEP_TLMB_start = TEPstart - Q1
   if(Q1.le.Q2) then
      if(TEP_TLMB_start.lt.0) TEP_TLMB_start = 0  ! Assumes you start at index zero
   else
      if(TEP_TLMB_start.lt.0) TEP_TLMB_start = TEP_TLMB_start + FireInterval
      if(TEP_TLMB_start.lt.0) TEP_TLMB_start=0
   endif
   TEP_TLMB_end = (TEPstart+TEPwidth) - Q1
   if(TEP_TLMB_end.lt.0) TEP_TLMB_end = TEP_TLMB_end + FireInterval
   if(TEP_TLMB_end.gt.Nrw) TEP_TLMB_end = Nrw

endif
```

Because the TEP is calculated from an absolute position relative to the start of the Range Window, and not from the altimetric histogram, its location must be artificially modified by the PCE delay to put it in the same reference system as the altimetric histogram. This will allow it to be correctly positioned when all of the telemetry band histogram bin information is adjusted for the PCE delay *(ST:RW_AltimHist_PCE_Delay_Strong)* prior to being sent to the hardware.

```
TEP_TLMB_start = TEP_TLMB_start - RW_AltimHist_PCE_Delay_Strong

TEP_TLMB_end = TEP_TLMB_end - RW_AltimHist_PCE_Delay_Strong
```

A similar calculation is done for the weak spot using its PCE delay (*ST:RW_AltimHist_PCE_Delay_Weak*).

7.6 Determining if it is day or night

The background noise rate is calculated from the noise in the Range Window. The events in the telemetry band are assumed to be signal and are eliminated from the calculations if possible. The DRM-700 is first scaled and padded based on surface type and relief category. Call this width pTLMB. The value of pTLMB is limited by the Algorithm Parameter for the telemetry band maximum (*ST:Band_Hi_Limit_Strong/Weak*) and is slid so that it resides completely within the Range Window. If pTLMB is greater than some fraction of the Range Window (*ST:background_bins_fraction_strong/weak),* nominally 50% of the Range Window, then the noise is calculated from the entire Range Window. This is also how the noise is calculated for the no signal case. When there is signal in the Range Window but the entire Range Window information is used to calculate the background, the calculation tends to be very inaccurate. Fortunately, this situation occurs operationally < 10% of the time.

If pTLMB is less than 50% of the Range Window width and if signal has been found in the MF, then pTLMB is used to eliminate any signal from the histogram. This band is centered about the primary MF signal location. The calculated noise rate in Hz for the MF, $N_{\mathbf{rate}}$, is then calculated from:

$$N_{\mathbf{rate}} = (N_{\mathbf{total}} - N_{\mathbf{tlm}}) / [\ 200 \cdot (\mathrm{Nrw} - \mathrm{pTLMB}) \cdot \mathrm{clock\_cycles\_in\_ns} \cdot 0.000000001\ ]$$

where:
$N_{\mathbf{total}}$ is the total number of events in the MF hardware altimetric histogram,
$N_{\mathbf{tlm}}$ is the number of events in the preliminary telemetry band,
Nrw is the width of the Range Window in clock cycles (from section 4.13),
pTLMB is the width of the preliminary telemetry band calculated from the DRM-700,
in clock cycles,
clock_cycles_in_ns is the value of the clock cycles in ns (nominally 10).

If signal has not been found in the MF or pTLMB is too large, then all of the events in the histogram are used to calculate the background noise rate:

$$N_{\mathbf{rate}} = N_{\mathbf{total}} / [\ 200 \cdot \mathrm{Nrw} \cdot \mathrm{clock\_cycles\_in\_ns} \cdot 0.000000001\ ]$$

This value is then compared against a surface-type dependent Algorithm Parameter. If $N_{\mathbf{rate}}$ > Day_Night_Threshold (*Knobs:Day_Night_Threshold_Strong/Weak*) then it is Day. Otherwise it is Night.

This calculation is based on the background rate and is used by the PCE to determine Day or Night. It is independent of the calculation of the solar zenith angle performed in section 4 which is used only by the SBC to determine day or night for the selection of the Range Window minimum and maximum values.

7.7 Calculating the telemetry band for the secondary and tertiary signal locations

If there is a secondary or tertiary signal, or if the TEP lies in the Range Window, the appropriate telemetry band from section 7.3 (7.5 for the TEP) is draped over the signal location(s), and the band is converted to clock cycles and limited by the telemetry maximum limit (as was done for the primary signal telemetry band). The exception to this is that there is no DRM-140 knob option for the tertiary signal telemetry band width (*Knobs: DRMDEM_SFSig_Strong/Weak*).

Overlapping or abutting telemetry bands are combined: the software picks the start and end of the combination, and the combined signal retains the name of the higher-level portion (e.g., secondary signal + tertiary signal will be called secondary). Once two bands are combined, the original bands are ignored. The order in which telemetry bands are combined is:

- Primary and secondary signal are combined to generate a new primary signal band
- Primary and tertiary signal are combined to generate a new primary signal band
- Primary and TEP are combined to generate a new primary signal band
- Secondary and tertiary signal are combined to generate a new secondary signal band (unless tertiary has priority, in which case the combined band is the new tertiary band)
- Secondary and TEP are combined to generate a new secondary signal band (unless TEP has priority in which case the combined band is the new TEP band)
- Tertiary and TEP are combined to generate a new tertiary signal band (unless TEP has priority in which case the combined band is the new TEP band)

The combining process is iterated until none of the remaining bands overlap or abut. The start and end bin numbers of each of the final signal telemetry bands are sent to section 7.8. All telemetry bands must be checked against the upper telemetry band limit (*ST:Band_Hi_Limit_Strong/Weak*) prior to combining. If the combined telemetry band is greater than the telemetry band limit, the bands are kept separated and NOT combined. If the two bands overlap but are not combined, the overlapping segment is removed from the second telemetry band. This chopping of overlapping bins does not take place until all combining of bands has been completed. Once combined the telemetry bands are not checked against any limit.

7.8 Telemetry priority and selecting what data to telemeter

Only two downlink telemetry bands can be sent by the hardware, but there can be up to four potential data sets vying for these two slots: (a) primary, (b) secondary, (c) tertiary, and (d) TEP. The normal priority is the order listed, with (a) being the highest priority and (d) being the lowest. There are two flags, however, that can change the order of data going into downlink telemetry band #2 (note that downlink telemetry band #1 always contains the primary signal if it exists). When the tertiary signal flag (which is an Algorithm Parameter given for each spot, *Knobs:tertiary_signal_flag_strong*/*weak*) is set (=1), this implies the tertiary signal has priority over the secondary signal for telemetry in downlink band #2. When this flag is off, the secondary signal band has priority over the tertiary. When the TEP flag (*Knobs:TEPflag_strong/weak*) is set (=1), this implies the TEP signal is sent in downlink telemetry band #2 when the TEP is within the Range Window, regardless of whether there is a secondary or tertiary signal. If the TEP flag is off, the TEP region is still telemetered when there is no secondary or tertiary signal.

A decision table for filling the two telemetry bands is shown below.

Table 7.1 Deciding what is telemetered in the two possible telemetry bands.

Case	MF Pri	Sec exists	SF	Pri in subW	Ter exists	Ter flag	Signal loc(s)
0							See note
1	X						Pri
2	X	X					Pri + Sec
3	X		X	Y			Pri
4	X		X	N	X		Pri + Ter
5	X	X	X	Y			Pri + Sec
6	X	X	X	N	X	0	Pri + Sec
7	X	X	X	N	X	1	Pri + Ter
8			X		X		Ter

Here Pri = Primary, Sec = Secondary, and Ter = Tertiary.

Tertiary signal flag values (*Knobs:tertiary_signal_flag_strong/weak*):
0 - use secondary if it exists otherwise use tertiary
1 - use tertiary if it exists otherwise use secondary

TEP flag values (*Knobs:TEPflag_strong/weak*):
0 – the cases to telemeter TEP as downlink telemetry band #2 are #0, 1,3 and 8
1 – telemeter TEP in every case as downlink telemetry band #2 regardless of other flags or band

If there is no signal but the TEP is in the Range Window, the TEP is sent down, regardless of how any other flags are set.

Notes:
(1) When no signal is found, the rules pertaining to the last known signal location apply. See section 7.4.
(2) In the table above case 1 is equivalent to case 3, and case 2 is equivalent to case 5.

7.9 Telling the hardware what to telemeter

If any downlink telemetry band extends beyond the histogram on either side, that downlink telemetry band is slid to ensure that the full extent of the band is used and that the band remains inside the histogram. For example, if there are “M” histogram bins and “N” bins in the downlink telemetry band,

and the calculated location for the telemetry band goes from K to M+7, then the bins selected to be telemetered are bins K-7 to M. Similarly, if the calculated location for the telemetry indicates bins -7 to 20, then the bins selected to be telemetered are bins 1 to 28. If the calculated downlink telemetry band is larger than the histogram, then the bins sent to the hardware for telemetry are the entire histogram, from 1 to M.

The starting bin of each downlink telemetry band and the number of contiguous bins following are converted to clock cycles past the start of the Range Window. At this point they must be modified by the PCE delay. For the strong spot, the starting and ending clock cycles for the telemetry must be increased by RW_AltimHist_PCE_Delay_Strong, and similarly for the weak spot (RW_AltimHist_PCE_Delay_Weak). See Figure 4.15.

Once the PCE delay correction has been made, any part of the telemetry band that falls outside of the Range Window is discarded and not given to the hardware to send.

There are no modifications to the atmospheric histogram for the PCE delay.

Once the PCE delay conversions are made the Algorithms move on to checking if the fire time tags are being telemetered for this PCE.

7.10 Checking if all laser fire time tags are telemetered

After the telemetry bands for both spots in a PCE have been generated, and if the AlwaysSendFires flag is set to ON, then if it is determined that no telemetry bands are to be sent for this MF, then the software shall send to the hardware a telemetry band of size 2 hardware bins with starting location 0 (i.e. start of the Range Window), and with a software channel mask where all of the channels are disabled (turned off). This should cause the hardware to send all of the fire time tags in the telemetry but none of the receive time tags.

Both the telemetry band size (2 hardware bins) and the starting bin of the telemetry band (bin # 0) shall be global variables that are easy to modify with a software patch should the need arise. It is not currently anticipated, however, that any change will need to be made to these variables.

A command to turn this procedure ON (which sets AlwaysSendFires ON) and one to turn it OFF (which sets AlwaysSend Fires OFF) must be available so that procedure can be turned OFF should the data volume impact be too great. The launch default is to have this procedure set to ON.

After performing the above to ensure all laser fire time tags are sent if AlwaysSendFires is ON, the hardware histogram bins are then given to the hardware, which is responsible for telemetering the corresponding time tags. As mentioned above, there may be up to two separate downlink telemetry bands per spot per Major Frame. Each can be up to 1.5 km in width (*ST:Band_Hi_Limit_Strong/Weak,* based upon hardware limitations).

8.0 On-orbit Operations and Algorithms Knobs

Because there exists the possibility that the amount of science data selected by the Rx Algorithms is larger than the physical capability of the downlink, the Flight Software will have various “Knobs” that can be used to dial down (or up) the telemetry volume.

8.1 Descriptions of knobs and relevant parameters

Except where noted, knobs come in sixteen flavors: one day and one night for each of the four surface types and for the Strong and the Weak spots. Parameters may have less flexibility. Most Algorithm parameters are set to maximize the chance that the downlinked data will include the returned signal photons. These parameters should not be used for adjusting data volume and are not included in this section. Figures 7.1 a and b show the decision flow through the knobs. The knobs are described in Table 8.1.

Table 8.1. Knob descriptions

Knob	Description
Tm_*	0: Do not telemetry data under any conditions. 1: Telemeter data if signal found in MF. Otherwise, check the TM_NoMFSig_* knob.
Tm_NoMFSig_*	0: Do not telemeter data if signal was not found in the MF. 1: Look for signal in the SF.
Tm_NoSFSig_*	0: Do not telemeter data if signal was not found in either the MF or the SF. 1: Check the Cloud_Alg_* knob.
Cloud_Alg_*	0: Do not run the cloud algorithm. Check the Tm_NoSig_* knob. 1: Run the cloud algorithm. Telemeter data unless the cloud algorithm determines that clouds are too thick for a ground return to be seen.
Tm_NoSig_*	This knob is checked if signal was not found in either the MF or the SF and the Cloud_Alg_* knob says the cloud algorithm should not be run. 0: Do not telemeter data. 1: Telemeter data based on the NoSig_* parameters.
DRMDEM_MFSig_*	Specifies the source for determining the telemetry band relief when signal is found in the MF. 0: Use DRM140. 1: Use DRM700. 2: Use the Range Window.
DRMDEM_SFSig_*	Specifies the source for determining the telemetry band relief when signal is found in the SF. 1: Use DRM700. 2: Use the Range Window.
SWChDisable_*	Bit mask specifying which channels are to be used for each spot.
No-signal knobs	There are three different knobs for each of the two states.

Knob	**Description**
Nosig_Timer1_*	The instrument remains in timer1 state for this number of consecutive MFs with no signal. It then switches to timer2 state.
Nosig_Relief1_*	Specifies the source for determining the telemetry band relief when in timer1 state. 0: Use DRM140. 1: Use DRM700. 2: Use the Range Window.
Nosig_Scale1_*	Scaling factor to be applied to the relief when in timer1 state.
Nosig_Timer2_*	The instrument remains in timer2 state until signal is not found for this number of consecutive MFs. It then stops telemetering data until signal is found. If Nosig_Timer2_* < 0, the instrument remains in timer2 state until signal is found.
Nosig_Relief2_*	Specifies the source for determining the telemetry band relief when in timer2 state. 1: Use DRM700. 2: Use the Range Window.
Nosig_Scale2_*	Scaling factor to be applied to the relief when in timer2 state.
NoSig_DEM_fraction_*	Factor used to compute the center of the telemetry band in the timer2 state.
TEP_Not_*	This knob specifies whether the region around the TEP location is to be excluded when determining if signal is present. 0: Include the TEP region when determining if signal is present. 1: Exclude the TEP region when determining if signal is present.
TEPflag_*	This knob specifies the TEP signal priority relative to the secondary or tertiary signal. 0: TEP does not have priority. 1: TEP has priority.
Tertiary_Signal_Flag_*	This knob specifies the relative priority of the secondary and tertiary signal. 0: Secondary has priority over tertiary. 1: Tertiary has priority over secondary.

8.2. Impact of Knobs and Algorithm Parameters on data

Table 8.2. Impact of algorithm "knobs" and parameters

Knob	Impact on science	Impact on data volume
TM flag	Obvious	Obvious
MF, SF, Cloud alg, and No signal flags	If a flag is set so data are not transmitted, there is no chance of finding signal on the ground.	Depending on why the onboard Algorithms failed to find signal, the impact on data volume could be relatively small or quite large (e.g., very noisy data).
Telemetry band limits (TBL)	If the maximum TBL is decreased, the probability that the downlinking data contains the signal is decreased. However, if the Algorithms are working correctly, the returned signal will be near the center of the downlinked band, and such adjustment would only impact the science in extremely rough regions, or if the TBL is decreased significantly.	The volume of the downlinked data changes approximately linearly with the downlink band.
Use DEM rather than DRM-700 or DRM-140 **or** Use DRM-700 rather than DRM-140	This will increase the probability that the downlinking data contains the signal.	In most parts of the world, the DEM range is substantially larger than the DRM-700 range. This would have a significant impact on the instantaneous data rate. If used for a substantial period of time, it could increase the data volume beyond the limit.
Day/Night Bkg Rate Discriminator	If the discriminator is raised, a larger fraction of the orbit is treated as night, allowing more data to be sent down and increasing the probability that the downlinked data contain the signal.	The twilight period for the spacecraft (the period that can be affected by the discriminator) typically lasts about 30 seconds, twice an orbit, with an upper limit of about six minutes. Shifting the discriminator so the entire period is defined as Day could increase the time it takes for the onboard algorithm to process data because it turns on the cloud filter algorithm.
DRM-140 and DRM-700 padding	Extra padding increases the probability that the downlinked data contain the signal. This is probably more important when signal is not found in the MF.	Data volume increases approximately linearly with padding.
Use Cloud filter algorithm, adjust algorithm parameters	If the algorithm is working properly, this will probably have little impact on science. Initial tests indicate < 0.8% of high optical depth spots will have detectable ground signal with the default values of N and S.	Turning the cloud filter algorithm OFF could increase data volume by 20% with the current values of the filter parameters. If these are changed, the change in data volume will also change. Increasing the parameters **F** or **Sc** will reduce the data volume, but these parameters should be set to improve the algorithm results, not to adjust the data volume.
Disable channels	Disabling channels will weaken the signal and reduce the chance of separating signal from noise, but it may allow data from a larger elevation band to be used for downlink, by reducing the total number of photons. Because	The data volume varies linearly with the total number of channels that are enabled in either hardware or software. The number of photons detected by the weak and strong channels should be approximately the same given the balance between beam strength and number of channels per spot.

Knob	Impact on science	Impact on data volume
	each strong beam is split into 16 channels, while the weak beams are split into 4 channels each, disabling channels from the strong beam will have less of an impact on most science as long as the signal is sufficiently strong. If the signal is too weak to be detected by the weak beam, then disabling weak beam channels will impact the science less.	
DRM scaling	Increasing the DRM band increases the probability of including the signal in the downlinked data.	Increasing the DRM band increases the data volume approximately linearly.

If the data volume is lower than expected, what changes will have the greatest potential positive impact on the science?

- Use the DEM rather than DRM-700 for no signal case over the ice sheets and sea ice, then over land areas.
- Use DEM-700 rather than DRM-140 when signal is detected in the MF.
- Increase the padding around the downlink band especially in high relief areas (use DRM700 rather than DRM140?)
- Increase the DRM scaling.
- Downlink weak spot for vegetated area even if no signal found.
- Adjust the TLBs to increase the maximum downlink band. This could improve the chance of capturing ground signal in rough areas or areas where the topography is poorly known.

If the data volume is higher than expected, what changes will cause the least loss of science data?

- Reduce the number of channels sent down for the shot.
- Do not send down the weak spots over land
- Always use DRM-140 over land even if the signal was lost.
- Decrease the DRM scaling
- Do not send down land data unless signal was found
- Use DRM-700, not DEM, for no signal case over ice sheets (this is currently the default)
- Adjust the TLBs to decrease the maximum downlink band. This could reduce the chance of capturing ground signal in rough areas or areas where the topography is poorly known.
- Send down only calibration data over the oceans.

9.0 Error Reponses

Spacecraft data drops out

The Algorithms are able to extrapolate accurately for about 3 seconds. The accuracy of this extrapolation will degrade and after 3 seconds of no spacecraft data (no position and/or no pointing) the telemetry will stop.

This problem is not under control of the Receiver Algorithm. Ground system and/or Spacecraft personnel should determine the cause of the error and get the spacecraft data flowing back to ATLAS. If this is not possible, then timed commands to set the Range Window over those areas of the Earth that are most important should be given to the spacecraft to command ATLAS for fixed Range Window starts and widths.

No spacecraft data

If no spacecraft data is ever received the Algorithms will not work. No science data can be collected.

This problem is not under control of the Receiver Algorithm. See the action suggested above.

Bad spacecraft data

Spacecraft data that causes the denominator in the conversion calculations to be zero is considered bad spacecraft data for the Rx Algorithms even though it will happen as it represents the spacecraft pointing the instrument away from the Earth. This situation should be monitored by the FSW and treated as if there were no spacecraft data.

This problem is not under control of the Receiver Algorithm. Spacecraft personnel need to figure out why this is happening and fix it if not expected. If this isn't possible, see the response to "spacecraft data drops out".

Loss of laser fire

With no laser fires no science data can be collected.

This problem is not under control of the Receiver Algorithm. ATLAS laser subsystem working with ground system personnel need to determine how to get the laser working or switch to the alternate laser. There is no other fix for this.

Laser fire rate different than 10 kHz

Small differences from 10 kHz will never be noticed by the onboard Algorithms. Larger laser fire rate errors will cause errors in the Algorithms calculations, as the design is based on 200 shots occurring in 0.02 seconds. There is no onboard response for this. This will require ground intervention to change parameters.

The Rx Algorithm team personnel should be contacted and made aware of any situation where the laser fire rate differs from 10 kHz by more than 100 Hz. It may be possible to change some of the parameters to compensate, however, the best situation is if the laser can be forced somehow to fire closer to 10 kHz (and this may not be possible once the instrument is on orbit).

Bad channel in the detector data

If it is determined from the ground that a channel in the detector is bad (either high noise or no noise), it will be turned off from the ground. This will affect some of the Rx Algorithm calculations and some of the parameter will need to be modified or knobs turned off.

Turning off the hardware channels will reduce the total signal and noise rates, but should not affect the signal to noise ratio. It will impact the Palm cloud algorithm and the daylight decision parameter. It may be necessary to change the cloud algorithm parameters by command from the ground; if too many channels go bad, it may not be possible to use the algorithm at all to determine if thick clouds are present. If this situation occurs, then ground intervention will be needed to prevent the Palm algorithm from being run. Details on changing the knobs to handle this case will be explained in a document (Changing the Rx Algorithm Knobs while in Orbit) to be produced by the Rx Algorithms team in the coming year.

One other parameter that may need to be changed is the minimum signal count. However, reducing this parameter needs to be done with care as it can cause the TEP signal to start being picked up by the Rx Algorithms as surface echoes.

Corrupt or bad onboard databases

Science data collection will be difficult with bad onboard databases. The appropriate response is to reload database(s) from ground. If the problem is the DEM, it would be possible to fix the Range Window at a given delay and at the maximum width. This would ensure that some science data would be collected. If the problem is the SRM, it is possible to upload a fixed surface type parameter. If the problem is the DRM, then uploading a fixed relief value for the DRM-140 and one for the DRM-700 would allow science data to continue being collected, however, this could have an adverse effect on the daily data volume.

No histograms from the PCE hardware

No science data can be collected.

The cause of the error will have to be determined. If it cannot be fixed from the ground, then the ground will have to send up timed commands with fixed telemetry band locations for the hardware to send down since no signal processing can occur.

Extremely high noise rate

The Receiver Algorithms can handle noise rates greater than 12 MHz for short periods of time (under some conditions) and still function properly. If the very high noise rate happens often then there will be loss of data since this will cause the spacewire data rate limit to be exceeded and the Algorithms will automatically reduce the telemetry bands to ensure the data that has the highest probability of being surface echoes gets downlinked. If the high noise rate continues, the ground will need to intervene to turn off detector channels. More directly, noise rates > 12 MHz will reduce the probability of finding signal, decrease Prob(ACQ) and increase Prob(FA).

Extremely low return signal levels

Low return signal levels can be handled by the Algorithms when the noise rate is low. As the noise rate increases the return signal strength that can be found by the Algorithms is limited to higher and higher levels. Continued low signal strength levels will probably require increasing the laser transmit energy from the ground. If this isn't possible, the Algorithms will still be able to find low signal level returns at night. Depending on the severity of the problem, the Algorithms may or may not be able to detect signal through relatively thin clouds at night. Depending on the severity of the problem, the ground may need to go to timed commands with fixed telemetry band locations for the hardware to send down.

Off-pointing greater than 5 degrees

The Algorithms can successfully handle an off-nadir pointing angle up to 5 degrees. Testing beyond 5 degrees has not yet occurred, however, there should be a graceful degradation in the ability of the Algorithms to determine the laser spot location on the Earth surface and to calculate the range to the surface. Prior to off-pointing greater than 5 degrees, some Algorithm parameters may need to be changed to optimize the science data collection.

If off-nadir angles greater than 5 degrees are required, contact the Rx Algorithms team for a review and possible update of the parameters. Please do this with sufficient time for the Rx Algorithms team to analyze the situation.

Higher data volume than expected

If the data volume is higher than expected, especially if it is higher than the daily downlink can handle, there are a number of "knobs" that can be "turned" to decrease the data volume (see section 8). Care must be taken in determine which knobs to use and how the knobs are combined, as many knobs cause science data collection to be reduced.

The Rx Algorithms team will produce a document (Receiver Algorithms User's Guide) that will indicate what knobs to change for this case and several other potential cases.

Appendix A: DEM (Digital Elevation Model)

The development of the DEM was completed by Lori Magruder and Holly Wallis from the University of Texas, Center for Space Research.

The DEM consists of a 3 tiered set of information. Each tier consists of a minimum height data set and a maximum height data set. The upper tier consists of a 1 deg x 1 deg resolution grid of tiles referenced by the latitude and longitude of the south-west corner.

For any tile in this grid whose maximum minus minimum height difference is greater than 5.5 km, a middle tier of 4x4=16 tiles, each 0.25 x 0.25 deg, will be generated. For any tile in this middle tier that has a max minus min height difference greater than 5.5 km, a lower tier of 5x5=25 tiles, each 0.05 x 0.05 deg, will be generated. If the height differences in any of the lower tier tiles are greater than 5.5 km, the information will be left unchanged and appropriate actions will be taken in the onboard software to deal with this situation.

The reference system used in the generation of the DEM must match that used in the generation of the onboard Range Window.

The accuracy of the minimum and maximum heights is currently expected to be less than 100 meters (3 sigma) and in most cases much less than that. We are carrying +/- 150 meters as the error budget for the difference between DEM max minus min height.

To accommodate both the spacecraft ground track speed and the accuracy with which the onboard algorithm can compute the latitude/longitude of the laser spots, a 2 km overlap is required for every edge of each tile in every tier.

For a more detailed description of the DEM see the DEM/DRM/SRM Documentation which is included with the databases (ICESat-2-ALG-TN-0362) on the ICESat-2 Technical Data Management System (TDMS): https://ipdtdms.gsfc.nasa.gov

Appendix B: DRMs (Digital Relief Maps)

The development of the DRM was completed by Lori Magruder and Holly Wallis from the University of Texas, Center for Space Research.

The DRM consists of two single tiered files with 0.25 x 0.25 deg tiles. The two DRMs are DRM-140 and DRM-700. The value in each DRM-140 tile represents the maximum difference in terrain height over a horizontal distance of 140 meters, over the entire tile. This is the distance traveled by each laser spot during a Major Frame (0.02 seconds). Similarly the value in each tile of the DRM-700 represents the maximum difference in terrain height over a horizontal distance of 700 meters, over the entire tile. The 700m distance represents the ground travel of each laser spot over 5 Major Frames (0.1 seconds).

The height differences calculated for the DRM include vegetation heights where vegetation is present and manmade structures.

The DRM contains the calculated maximum height difference over the tile for the given horizontal distance. This value will be a number >=0. If the calculated maximum height difference is zero, then zero should be the value in the DRM for that tile. This value should not be limited to a minimum value. This value must, however, be limited by the maximum value possible within the byte; this upper limit is 4347 m, larger than any expected real relief on a scale of 140 m or 700 m. When using the DRM, the Flight Software will limit the minimum and maximum values of relief used based on terrain type, background noise and other factors. So just as with the DEM, the DRM tile values must not be limited.

The accuracy of the DRM greatly affects the Telemetry Data Volume, as it impacts the padding that must be added to the DRM relief to make the telemetry band. Current assessment of accuracies possible are given in the DEM/DRM/SRM Statement of Work for UT/CSR.

The padding added by the FSW to the relief will be a combination of the DRM accuracy and the vertical accuracy of the signal location. The latter is expected to be 3 meters. The FSW will limit the DRM to no less than the minimum value after the padding has been applied. The minimum value will be a function of the terrain type. The maximum value limit will also be imposed after padding, and will also be a function of terrain type, but will also depend upon the background noise rate.

As with the DEM, a 2 km overlap is required for every edge of each tile in the DRMs.

For a more detailed description of the DEM see the DEM/DRM/SRM Documentation which is included with the databases (ICESat-2-ALG-TN-0362) on the ICESat-2 Technical Data Management System (TDMS): https://ipdtdms.gsfc.nasa.gov

Appendix C: SRM (Surface Reference Mask)

The development of the SRM was completed by Tim Urban et al. from the University of Texas, Center for Space Research.

The surface reference mask contains information on the surface type in tiles that are 0.25 x 0.25 deg. There are four surface types similar to GLAS: ocean(0), land(1), sea ice(2), and ice sheet(3). In addition there are six other bits of information for each tile: whether the tile has land with vegetation, whether there is coastline present in the tile, and four spare bits for future use. The vegetation bit is currently not used by the Receiver Algorithms. The use of the coastline bit is described in section 7.

The following is the order of priority of the surface types:

1- if any ice sheet is present,
2- else if sea ice present,
3- else if land present,
4- else ocean.

As with the DEM and DRMs, a 2 km overlap is required for each tile in the SRM.

For a more detailed description of the DEM see the DEM/DRM/SRM Documentation which is included with the databases (ICESat-2-ALG-TN-0362) on the ICESat-2 Technical Data Management System (TDMS): https://ipdtdms.gsfc.nasa.gov

Appendix D: Theoretical Signal Processing Probability Analysis

D.1 Probability Calculations

The requirement is for the Algorithms to correctly find a surface segment in 700 meter ground tracks for most of the ATLAS Design Cases with a probability of >= 90% and a false alarm rate of =<10%. For more information on the Design Cases see the ATLAS Instrument Predicted Performance document on TDMS (ICESat-2-ATSYS-TN-0028).

The probability of correctly finding surface echoes in a Major Frame is called the Probability of Acquisition or Prob(ACQ) and is given by the following.

The noise standard deviation (σ) and the threshold (X) are given by the following where R is the expected noise photoelectrons (pes) after F fires:

$$\sigma = \sqrt{R}$$

$$X = R + 3\sigma$$

The probability that one noise bin will have X or less photoelectrons is:

$$P_{XL} = e^{-R} \sum_m \frac{R^m}{m!}$$

where m = 0,…,X

The probability that the signal bin will have X or more photoelectrons after F fires is:

$$R_s = R \ + \ Fn_{pe}$$

$$P_{XG} = 1 - e^{-R_g} \sum_i \frac{R_g^i}{i!}$$

where i = 0,…, X-1,
and where n_{pe} is the expected number of signal photoelectrons returning per fire.

Then the probability of acquisition is given by the following, where n_b = number of bins in the Range Window histogram:

$$P_{ACQ} = (P_{XL})^{n_b-1}P_{XQ}$$

The probability of mistaking noise for surface echoes in a MF is called the Probability of False Alarm or Prob(FA) and is given by the following.

The probability that one noise bin will have X or greater noise photoelectrons is:

$$P_{XX} = 1 - e^{-R}\sum_i \frac{R^i}{i!}$$

where i = 0,…,X-1.

Then the probability that at least one bin in the histogram will have X or greater noise pes is:

$$P_{IA} = n_b(P_{XL})^{n_b-1}P_{XX}.$$

Looking at 5 frames combined as a Super Frame allows further rejection of false alarms while allowing a wider vertical movement in the returns. The Probability of Acquisition and False Alarm for the SF is given in the following where pacq and pfa are the Prob(ACQ) and Prob(FA) for the Major Frame.

$$\mathrm{P_{ACQ}} = \sum\nolimits_{\mathrm{i}}\left(\frac{\mathrm{M!}}{\mathrm{(M-i)!i!}}\right)\left(\mathrm{p_{acq}}\right)^{\mathrm{i}}\left(1-\mathrm{p_{acq}}\right)^{\mathrm{M-i}}$$

where i = N,…,M. Here M = 5 Major Frames and N is the number of MFs that must have signal in order for the SF to have signal. In the current Algorithms N = 3.

$$\mathrm{P_{FA}} = \sum\nolimits_{\mathrm{i}}\left(\frac{\mathrm{M!}}{\mathrm{(M-i)!i!}}\right)(\mathrm{p_{fa}})^{\mathrm{i}}(1-\mathrm{p_{fa}})^{\mathrm{M-i}}$$

where i = N,…,M, M = 5 and N = 3.

If either the MF or SF Prob(ACQ)>=90% then the Algorithms are deemed to be successful. See the spreadsheet of Design Cases in Tables D.1 (weak spots) and D.2 (strong spots).

D.2 Theory behind the sigma multiplier table entries

To maintain a 10% probability of false alarm over the large differences in Range Window widths, the Algorithms must take into account the width of the Range Window to calculate a scale factor needed for the standard deviation when computing the threshold. The results of this calculation are stored in the "sigma_multiplier" file. The Flight Software reads the table of data from this file and uses it to look-up the correct scale factor. The calculations used to produce the data in this file are as follows.

R = average noise in the software bin (in a MF)

nb = total number of "software" bins in the overlapping histogram

The threshold, X, for the Algorithm is calculated as:

$$\sigma = \sqrt{R}$$

$$X = R + s \cdot \sigma$$

Where "s" is chosen to satisfy the equation:

$$0.05 \quad = nb \cdot \mathrm{erfc}\,(s / \sqrt{2})$$

Where "s" represents the standard deviation scale factor, and "erfc" is the complementary error function. Here "0.1" is the desired Prob(FA) across the entire Range Window, but we are using "nb" from the selected software histogram, while the full software histogram in use is the overlapping histogram which has about twice the number of bins. Using a sigma multiplier that gives a 5% probability of false alarm for the selected software histogram will ensure that the Prob(FA) =< 0.1 for whichever software histogram is selected.

The calculation of "s" from "nb" is then:

$$s = \mathrm{erfc}^{-1}(0.05/nb) \cdot \sqrt{2}$$

To generate the sigma multiplier table, "s" is calculated from various values of "nb" where the intervals between values of "nb" are chosen to give close enough spacing to allow for accurate control to keep the probability of false alarm at or below 0.1 while minimizing the size of the table. The sigma multiplier file contains a table of "nb" values each with its corresponding "s" value. See section 5.3 for how this table is used. In section 5.3 the value of "nb" is called "nswbin" and the value of "s" is called "$\boldsymbol{\sigma}_{\mathrm{scale}}$".

Table D.1: Weak spot theoretical Prob(ACQ) calculations for the Design Cases. All were done with a Range Window width of 6 km (except for ocean where Range Window width was taken as 1 km). Green cells with the letters "MF" imply that the Prob(ACQ) was achieved in the Major Frame. Yellow cells with the letters "SF" indicate that the Prob(ACQ) was achieved in the Super Frame. A red cell implies that the Prob(ACQ) was not achieved. White cells are not conditions that need to be examined. All of the required Design Cases pass.

	WEAK Case #	Signal pes/shot	Noise MHz	Binsize 12m	Binsize 18m	Binsize 24m	Binsize 30m	Binsize 36m	Binsize 42m	Binsize 48m	REQMT ?
Icesheet	**1a**	2.15	0.50	MF	MF	MF	MF	MF	MF	MF	Yes
	1b	3.00	0.50	MF	MF	MF	MF	MF	MF	MF	Yes
	2a	2.04	6.19	MF	MF	MF	MF	MF	MF	MF	Yes
	2b	1.36	6.21	MF	MF	MF	MF	MF	MF	MF	Yes
	2c	0.43	6.00	MF	MF	MF	MF	MF	SF	SF	Yes
	3a	1.01	0.50	MF	MF	MF	MF	MF	MF	MF	Yes
	3b	0.68	0.50	MF	MF	MF	MF	MF	MF	MF	Yes
	3c	0.15	0.50	MF	MF	MF	MF	MF	MF	MF	Yes
	4a	0.68	4.69	MF	MF	MF	MF	MF	MF	MF	Yes
Sea Ice	**5a**	1.51	0.50	MF	MF	MF	X	X	X	X	Yes
	5b	0.26	0.50	MF	MF	MF	X	X	X	X	Yes
	5c	2.21	0.50	MF	MF	MF	X	X	X	X	Yes
	6a	0.60	5.73	MF	MF	MF	X	X	X	X	Yes
	6b	0.06	2.92	F	F	F	X	X	X	X	No
	6c	5.67	1.77	MF	MF	MF	X	X	X	X	Yes
Land - vegetation -min cover	**7a**	0.20	3.37	F	F	F	F	F	F	F	No
	7b	0.20	3.37	F	F	F	F	F	F	F	No
	7c	0.20	3.37	F	F	F	F	F	F	F	No
	8a	0.73	2.01	MF	MF	MF	MF	MF	MF	MF	No
	8b	0.73	2.01	MF	MF	MF	MF	MF	MF	MF	No
	8c	0.73	2.01	MF	MF	MF	MF	MF	MF	MF	No
	9a	0.73	2.00	MF	MF	MF	MF	MF	MF	MF	No
	9b	0.73	2.00	MF	MF	MF	MF	MF	MF	MF	No
	9c	0.73	2.00	MF	MF	MF	MF	MF	MF	MF	No
Land - vegetation - max cover	**7a**	0.15	2.73	F	F	F	F	F	F	F	No
	7b	0.15	2.73	F	F	F	F	F	F	F	No
	7c	0.15	2.73	F	F	F	F	F	F	F	No
	8a	0.44	1.21	MF	MF	MF	MF	MF	MF	MF	No
	8b	0.44	1.21	MF	MF	MF	MF	MF	MF	MF	No
	8c	0.44	1.21	MF	MF	MF	MF	MF	MF	MF	No
	9a	0.32	1.47	MF	MF	MF	MF	MF	MF	MF	No
	9b	0.32	1.47	MF	MF	MF	MF	MF	MF	MF	No
	9c	0.32	1.47	MF	MF	MF	MF	MF	MF	MF	No
Ocean	**10a**	0.49	2.68	MF	MF	MF	X	X	X	X	No
	10b	0.21	1.94	MF	MF	SF	X	X	X	X	No
	10c	0.12	1.71	F	X	X	X	X	X	X	No

Table D.2: Strong spot theoretical Prob(ACQ) calculations for the Design Cases. All were done with a Range Window width of 6 km (except for ocean where Range Window width was taken as 1 km). The colors of the cells are defined as in Table D.1 above. All of the required Design Cases pass.

	STRONG	Signal	Noise	Binsize	Binsize	Binsize	Binsize	Binsize	Binsize	Binsize	
	Case#	pes/shot	MHz	12m	18m	24m	30m	36m	42m	48m	REQMT?
Icesheet	**1a**	8.60	0.50	MF	MF	MF	MF	MF	MF	MF	Yes
	1b	12.00	0.50	MF	MF	MF	MF	MF	MF	MF	Yes
	2a	8.16	6.19	MF	MF	MF	MF	MF	MF	MF	Yes
	2b	5.44	6.21	MF	MF	MF	MF	MF	MF	MF	Yes
	2c	1.72	6.00	MF	MF	MF	MF	MF	MF	MF	Yes
	3a	4.04	0.50	MF	MF	MF	MF	MF	MF	MF	Yes
	3b	2.72	0.50	MF	MF	MF	MF	MF	MF	MF	Yes
	3c	0.60	0.50	MF	MF	MF	MF	MF	MF	MF	Yes
	4a	2.72	4.69	MF	MF	MF	MF	MF	MF	MF	Yes
Sea Ice	**5a**	6.04	0.50	MF	MF	MF	X	X	X	X	Yes
	5b	1.03	0.50	MF	MF	MF	X	X	X	X	Yes
	5c	8.84	0.50	MF	MF	MF	X	X	X	X	Yes
	6a	2.40	5.73	MF	MF	MF	X	X	X	X	Yes
	6b	0.23	2.92	MF	SF	SF	X	X	X	X	Yes
	6c	22.68	1.77	MF	MF	MF	X	X	X	X	Yes
Land - vegetation -min cover	**7a**	0.78	3.37	MF	MF	MF	MF	MF	MF	MF	No
	7b	0.78	3.37	MF	MF	MF	MF	MF	MF	MF	No
	7c	0.78	3.37	MF	MF	MF	MF	MF	MF	MF	No
	8a	2.92	2.01	MF	MF	MF	MF	MF	MF	MF	No
	8b	2.92	2.01	MF	MF	MF	MF	MF	MF	MF	No
	8c	2.92	2.01	MF	MF	MF	MF	MF	MF	MF	No
	9a	2.92	2.00	MF	MF	MF	MF	MF	MF	MF	No
	9b	2.92	2.00	MF	MF	MF	MF	MF	MF	MF	No
	9c	2.92	2.00	MF	MF	MF	MF	MF	MF	MF	No
Land - vegetation - max cover	**7a**	0.62	2.73	MF	MF	MF	MF	MF	MF	MF	No
	7b	0.62	2.73	MF	MF	MF	MF	MF	MF	MF	No
	7c	0.62	2.73	MF	MF	MF	MF	MF	MF	MF	No
	8a	1.78	1.21	MF	MF	MF	MF	MF	MF	MF	No
	8b	1.78	1.21	MF	MF	MF	MF	MF	MF	MF	No
	8c	1.78	1.21	MF	MF	MF	MF	MF	MF	MF	No
	9a	1.30	1.47	MF	MF	MF	MF	MF	MF	MF	No
	9b	1.30	1.47	MF	MF	MF	MF	MF	MF	MF	No
	9c	1.30	1.47	MF	MF	MF	MF	MF	MF	MF	No
Ocean	**10a**	1.95	2.68	MF	MF	MF	X	X	X	X	Yes
	10b	0.84	1.94	MF	MF	MF	X	X	X	X	Yes
	10c	0.49	1.71	MF	MF	MF	X	X	X	X	Yes

Appendix E: Spacecraft Position & Pointing and Database Accuracy

Introduction

There are multiple constraints on the ATLAS Flight Science Receiver Algorithms (FSRA). The maximum daily data volume limit is 577.4 Gb/day for the science data. The data corresponding to the required Design Cases must all be correctly telemetered to the ground successfully >= 90% of the time. The space available in the Flight EEPROM space for the FSRA databases is < 3 MB. The PCE hardware can only handle a maximum Range Window of 6 km. The instantaneous data rate cannot exceed 80 Mb/s. The Algorithms must be able to handle a noise rate up to 12 MHz and must contend with clouds that could potentially overlap the surface echoes. Lastly the Algorithms must perform successfully anywhere from 0 to 5 degrees off-nadir. Because the laser beams are already off-pointed, this implies a maximum off-nadir pointing of 5.5 degrees.

To satisfy the Algorithms requirements while living within the constraints, it was decided that we would make use of the spacecraft onboard knowledge of position and attitude, and use this information to generate a limited search region (called the Range Window) based upon the knowledge of the spacecraft location and orientation, and an onboard database of the Earth heights.

The ATLAS Flight Software (FSW) will receive separate position and attitude messages at 1 Hz from the spacecraft. The information in the messages can be up to 500 ms stale when it arrives. The AFSW will take up to 1 second to calculate the Range Window information needed for the following second. This calculation must cover the entire second. To ensure that the calculated data on the SBC lines up with the signal processing software on the PCE the total extrapolation time must be 3 seconds.

Spacecraft information accuracies

The spacecraft position and attitude information is converted by the FSRA into a range to the Earth ellipsoid and location on the Earth (latitude and longitude). The latitude and longitude information is used to index into a database of heights, which in turn, is used to convert the range to the ellipsoid to a range to the surface of the Earth (see section 4 of the ATLAS Flight Science Receiver Algorithms Document). The accuracy of this conversion (and extrapolation) will impact the width of the Range Window used to search for the surface echoes. Larger Range Windows mean lower signal to noise ratios and imply lower percentages of success in finding the signal.

The requirements on the information provided onboard by the spacecraft are:

Position: 30 m (1 σ, each axis),
Velocity: 0.5 m/s (1 σ, each axis),
Pitch and Roll angles: each 150 μrad (1 σ),
Yaw angle: 1 mrad
Pitch and Roll rates: each 50 μrad/s

Yaw angle: 333 μrad/s

Combining the axes and using 3σ values to ensure a 99% coverage gives the following accuracies:

Position: 156 m
Velocity: 2.6 m/s
Pointing (assuming 5.5 deg off-nadir when calculating yaw): 710 μrad
Rates (assuming 5.5 deg off-nadir when calculating yaw): 240 μrad/s
Pointing (assuming 2.5 deg off-nadir when calculating yaw): 650 μrad
Rates (assuming 2.5 deg off-nadir when calculating yaw): 220 μrad/s

Errors in the time stamps from the spacecraft will be < 100 ms (expected < 50 ms). This is insignificant for precession and sidereal time calculations. Surface velocity for this orbit is ~7 km/s implying a < 700 meter error in surface location caused by a < 100 ms timing error. Maximum range change over a second for a nominal orbit (non-slewing) is 25 meters.

A source of error related to time, however, is the use of mean sidereal time rather than apparent sidereal time. This error can be as much 1.15 seconds of time. This equates to a surface movement of 525 meters which must be factored into the error calculations for surface location but can be ignored for range.

Errors after extrapolation

Linear extrapolation will be used to generate position and attitude information at times in the future when it is needed by the FSW. Higher order extrapolation will not be used for two reasons: (1) accelerations are not available and would have to be calculated, and (2) linear extrapolation is less prone to large errors due to fitting data with noise.
The error in the range due strictly to the position/velocity errors and to ignoring acceleration:

$$\Delta R = 156\text{ m} + (2.6\text{m/s})\cdot\Delta t + (\alpha_R/2)\cdot\Delta t^2.$$

The range acceleration (α_R) for ICESat-2 orbits is negligible except during slews where it can be approximately 5 m/s^2 (all due to angular motion). Taking $\Delta t = 3.0$ seconds gives a range error of ~165 m due to position/velocity errors. Combined acceleration in ECI rectangular coordinates is about 10 m/s^2 for a nominal orbit. The error in the spot positioning on the ground from the position and velocity errors, plus unaccounted for acceleration is:

$$\Delta X = 156\text{ m} + (2.6\text{m/s})\cdot\Delta t + (\alpha_P/2)\cdot\Delta t^2$$

This results in a spot position error after 3 seconds of ~ 210 m.

The combined angular error due to attitude/rate errors and ignoring angular acceleration:

$$\Delta\theta = 710\ \mu\text{rad} + (240\ \mu\text{rad/s})\cdot\Delta t + (\alpha_\theta/2)\cdot\Delta t^2.$$

The angular acceleration for off-nadir slews is constrained to be < 180 $\mu\text{rad/s}^2$. Extrapolating 3.0 seconds ahead gives an angular error of 2.24 mrad.

Total error budget for the Range Window

Assuming 3.0 seconds extrapolation time, 5.5 deg off-nadir angle, and all of the above constraints and accuracies:

Range error due to extrapolated position:	< 165 m nominal
Range error due to extrapolated pointing:	< 110 m
Range (to ellipsoid) calculation error:	< 10 m
Range due to timing error:	< 3 m
DEM accuracy:	< 150 m

RSS of these numbers gives 249 m (180 $\mu\text{rad/s}^2$ angular acceleration).

Total error budget for the geolocation error

Assuming 3.0 seconds extrapolation time, 5.5 deg off-nadir angle, and all of the above constraints and accuracies:

Error due to positional extrapolation:	< 210 m
Error due to extrapolated angle:	< 1242 m
Calculation error:	< 100 m
Timing error:	< 700 m
Error from use of mean sidereal time:	< 525 m
Error due to 9 km mountain in the way:	< 867 m

RSS of these numbers gives 1765 m. Every frame must lie entirely in the same lat/lon tile, so to the above numbers the S/C motion in the frame (200 shots ➔ 140 m) must be added, giving the need for overlap in the DEM tiles of >= 1905 m.

Note that there has been no error assigned to the laser vectors but that about 100 meters of the error budget is in reserve. From a 500 km orbit 100 meters is about 200 microradians (or a little less if we are 5 degrees off-nadir).

To compensate for the movement of the telescope relative to the laser beams, the Beam Steering Mechanism (BSM) is used to steer the laser beams. The BSM can move the beams up to 4 mrad. The onboard Rx Algorithms do not know anything about how the BSM has steered the laser beams. Until the laser vectors in the parameter files are updated (by uplinking a new file), the Rx Algorithms continue to calculate the laser beam intersection with the ground based on the old laser beam angular orientation with respect to the ATLAS MRF. If the BSM steers the laser too far away from where the Rx Algorithms think they are, the latitude/longitude used to access the DEM and DRM tiles in the onboard databases will not be the right one, resulting in errors in the Range Window location.

The major movement between transmit (laser beams) and receive (telescope) paths is expected to occur during launch. One of the first activities after launch will be to collect ocean calibration data in order to update the laser vectors. Fortunately the ocean surface is the most forgiving surface type in terms of having the right latitude/longitude tile, so there should not be an issue with being able to collect this science data. In addition the Rx Algorithm performance degrades slowly with increasing angular error, so we are optimistic that a majority of the science data can be collected even right after launch.

The Instrument Scientist believes the BSM will change only tens of microradians during normal operations based on the changes seen during TVAC. If this is the case, the laser vectors may never have to updated after the initial post-launch update, or updates may be needed only once every few months. However, if the BSM changes more than 100 microradians, the laser vectors should be updated by the ICESat-2 Precision Orbit Determination (POD) team using ocean calibration data and these new vectors should be uploaded as a Rx Algorithms Parameter file update.

Onboard databases and data volume

The onboard database size is constrained by the available EEPROM space. The data volume telemetered to the ground has been estimated to be very close to the 577.4 Gb/day limit.

Because of the constraints on the maximum hardware Range Window width and the size in EEPROM of the onboard databases, it is not possible to widen the Range Window beyond 5.5 km. In addition 5.5 km covers the maximum height difference for land ice. Reducing this number implies that we will not be able to produce a Range Window that correctly covers all land ice. With +/- 250 m padding to cover the accuracy in the Range Window location, the 5.5 km Range Window becomes the maximum 6 km that the hardware can handle, so an increase in the range error must result in a decrease in the maximum height difference allowable.

The requirement on the DEM accuracy has been set at 100 m. There are many areas of the Earth where this accuracy limit can be reached, however, there are other areas where existing information is sparse and where this error is not achievable. We are therefore carrying 1.5 times the original database accuracy requirement to compensate for this as yet not completely known accuracy.

To compensate for errors in positioning the 6 spots on the Earth, the lat/lon tiles will overlap by 2 km (ie each tile contains information from 2 km into all of the other tiles that touch it). Increasing this overlap will increase the daily data volume since the relief per tile, for many tiles, will increase. We examined the difference in data volume between a 1 km overlap and a 2 km overlap and while it did go up, the increase was minimal. This does not imply anything, however, about increasing the overlap beyond 2 km.

Conclusions

We have combined the worst case errors from all sources, which is a very conservative approach, however, the requirements currently are for the Receiver Algorithms to work should all worst case conditions occur together. In addition there remain unknowns in the calculations. Included in these are the accuracy of the UT/CSR generated onboard DEM.

The calculations in this document show that angular accelerations of =< 180 μrad/s^2 give a Range Window error of <= 250 m. Angular accelerations that are greater than this give errors above the 250 m limit. Because of the remaining unknowns it seems prudent to limit the region where we guarantee that the Receiver Algorithms will work.   A maximum acceleration of 180 μrad/s^2 would work according to the calculations, but even that value is pushing the Algorithms ability under worst case conditions.

Appendix F: Flight Parameters

The FSRA parameters are stored in 12 files. The information is divided between parameters relating to the Prediction, Pointing, and Range (PPR), the Signal and Telemetry (ST), and two sets of Knob Tables. For each of these groups of information there are 3 files – one for each track (ie PCE board). In each file the information is given for the strong and the weak spots where a differentiation between spots is required.

The Knob parameter file comes in two flavors: (a) the nominal knob parameters which are used for most of the orbit and are the default knob parameters, and (b) the calibration and special interest knob parameters, which are used for ocean scans, targets of opportunity (TOOs) and other things. Both are resident on all PCE board computers at the same time but only one is in use at any given time. The switch between the nominal Knob parameter file and the calibration Knob parameter file should be done with a single command upload and this switch must be done while in Science Mode.

The FSRA Simulator uses the information in all of these parameters files and this information is also passed to the Flight Software team for use onboard. This helps prevent any errors in passing the data between the two teams.

One of each of the three nominal file types (PPR, ST, Knobs) is given below. In all three cases the file for Track #1 is shown. Tracks #2 and #3 have the same format, but have some different values. The latest set of all 12 Parameter Files is currently stored and continually updated on TDMS (**ICESat-2-ALG-SPEC-0255)**. For further information please contact Jan McGarry (Jan.McGarry@nasa.gov) or Ann Rackley Reese (Ann.L.Rackley@nasa.gov).

F.1 Algorithm Signal/Telemetry (ST) Parameters (for Track 1): v6 (launch) Parameters

```
! ATLAS flight software algorithm parameters for signal and telemetry (ST 1)

&alg_parms_st_input

! ATLAS flight software algorithm parameters for signal and telemetry

! name                           assignment       units        notes

Version_ST                       =  '0000000029'  ! n/a        sequential number
Version_Date                     =  '2017-04-30'  ! YYYY-MM-DD
Track                            =  1             ! n/a        track number
Clock_Cycles_in_ns               =  10.0D0        ! ns         duration of a clock cycle

Nsf_Strong                       =  3             ! frames     N of 'N of M', number of frames where
signal must be found in a super-frame
Nsf_Weak                         =  3             ! frames     "  "  "  " "     "    "     "

Msf_Strong                       =  5             ! frames     M of 'N of M', number of frames in super-
frame analysis
Msf_Weak                         =  5             ! frames     "  "  "  " "     "    "     "

Bin_Size_Strong(0)               =  8             ! cs*        Size of software bins for water
Bin_Size_Strong(1)               = 32             ! cs*          "       "       "    "  land
Bin_Size_Strong(2)               =  8             ! cs*          "       "       "    "  sea ice
Bin_Size_Strong(3)               = 16             ! cs*          "       "       "    "  land ice
Bin_Size_Weak(0)                 =  8             ! cs*        Size of software bins for water
Bin_Size_Weak(1)                 = 32             ! cs*          "       "       "    "  land
Bin_Size_Weak(2)                 =  8             ! cs*          "       "       "    "  sea ice
Bin_Size_Weak(3)                 = 16             ! cs*          "       "       "    "  land ice

Sigma_For_Significance_Strong    =  5.0D0         ! std. dev.  Standard deviations above noise required
for second maximum signal detection
Sigma_For_Significance_Weak      =  5.0D0         ! std. dev.      "         "       "     "      "

Min_Secondary_SWbin_Separation   =  2.0D0         ! SW bins    Minimum number of SW bins in comb hist
needed between primary and next highest peak to be secondary signal (multiplier)

Min_Counts_For_Signal_Strong     = 10             ! counts     Minimum value of threshold for signal
finding
Min_Counts_For_Signal_Weak       = 10             ! counts          "      "     "    "       "

Cloud_Scale_Factor_Strong        =    3.D0        ! n/a        Scale factor for sqrt(background_ctr).
Used in calculation of cloud threshold
Cloud_Scale_Factor_Weak          =    3.D0        ! n/a          "      "    "    "       "      "

Cloud_Bins_Exclude_Strong        =    1           ! bins       Number of bins on either side of Max to
be excluded in background calculation
Cloud_Bins_Exclude_Weak          =    1           ! bins          "   "   "   "    "     "   "   "  "  "

Cloud_Threshold_Strong           =  600           ! counts     Threshold for determining if clouds are
too think
Cloud_Threshold_Weak             =  600           ! counts          "     "    "      "

Atm_Histogram_Width_Strong       =  9340          ! cs*        Range of the atmospheric histogram
Atm_Histogram_Width_Weak         =  9340          ! cs*          "   "   "       "          "

Cloud_Bin_Time_Strong            =  20            ! cs*        Light-time corresponding to atmospheric
bin size
Cloud_Bin_Time_Weak              =  20            ! cs*          "    "         "       "       "

Lbin_Strong                      =  334           ! bins       Lbin for atmopsheric histogram
processing.  Exclude bins above LBin when summing counts of bins above threshold. 1-based.
Lbin_Weak                        =  334           ! bins         "   "        "          "         "

Padding_140_Strong(1,0)          =  10            ! cs*        amount of Padding to add for DRM 140,
interval 1, water
```

```
Padding_140_Strong(1,1)          =  16            ! cs*          "         "        "       "   " ,
"    1, land
Padding_140_Strong(1,2)          =  10            ! cs*          "         "        "       "   " ,
"    1, sea ice
Padding_140_Strong(1,3)          =  16            ! cs*          "         "        "       "   " ,
"    1, land ice
Padding_140_Strong(2,0)          =  10            ! cs*          "         "        "       "   " ,
interval 2, water
Padding_140_Strong(2,1)          =  93            ! cs*          "         "        "       "   " ,
"    2, land
Padding_140_Strong(2,2)          =  93            ! cs*          "         "        "       "   " ,
"    2, sea ice
Padding_140_Strong(2,3)          =  93            ! cs*          "         "        "       "   " ,
"    2, land ice
Padding_140_Strong(3,0)          =  10            ! cs*          "         "        "       "   " ,
interval 3, water
Padding_140_Strong(3,1)          = 140            ! cs*          "         "        "       "   " ,
"    3, land
Padding_140_Strong(3,2)          = 140            ! cs*          "         "        "       "   " ,
"    3, sea ice
Padding_140_Strong(3,3)          = 140            ! cs*          "         "        "       "   " ,
"    3, land ice
Padding_140_Strong(4,0)          =  10            ! cs*          "         "        "       "   " ,
interval 4, water
Padding_140_Strong(4,1)          = 340            ! cs*          "         "        "       "   " ,
"    4, land
Padding_140_Strong(4,2)          = 340            ! cs*          "         "        "       "   " ,
"    4, sea ice
Padding_140_Strong(4,3)          = 340            ! cs*          "         "        "       "   " ,
"    4, land ice
Padding_140_Weak(1,0)            =  10            ! cs*       amount of Padding to add for DRM 140,
interval 1, water
Padding_140_Weak(1,1)            =  16            ! cs*          "         "        "       "   " ,
"    1, land
Padding_140_Weak(1,2)            =  10            ! cs*          "         "        "       "   " ,
"    1, sea ice
Padding_140_Weak(1,3)            =  16            ! cs*          "         "        "       "   " ,
"    1, land ice
Padding_140_Weak(2,0)            =  10            ! cs*          "         "        "       "   " ,
interval 2, water
Padding_140_Weak(2,1)            =  93            ! cs*          "         "        "       "   " ,
"    2, land
Padding_140_Weak(2,2)            =  93            ! cs*          "         "        "       "   " ,
"    2, sea ice
Padding_140_Weak(2,3)            =  93            ! cs*          "         "        "       "   " ,
"    2, land ice
Padding_140_Weak(3,0)            =  10            ! cs*          "         "        "       "   " ,
interval 3, water
Padding_140_Weak(3,1)            = 140            ! cs*          "         "        "       "   " ,
"    3, land
Padding_140_Weak(3,2)            = 140            ! cs*          "         "        "       "   " ,
"    3, sea ice
Padding_140_Weak(3,3)            = 140            ! cs*          "         "        "       "   " ,
"    3, land ice
Padding_140_Weak(4,0)            =  10            ! cs*          "         "        "       "   " ,
interval 4, water
Padding_140_Weak(4,1)            = 340            ! cs*          "         "        "       "   " ,
"    4, land
Padding_140_Weak(4,2)            = 340            ! cs*          "         "        "       "   " ,
"    4, sea ice
Padding_140_Weak(4,3)            = 340            ! cs*          "         "        "       "   " ,
"    4, land ice

Padding_700_Strong(1,0)          =  10            ! cs*       amount of Padding to add for DRM 700,
interval 1, water
Padding_700_Strong(1,1)          =  16            ! cs*          "         "        "       "   " ,
"    1, land
Padding_700_Strong(1,2)          =  10            ! cs*          "         "        "       "   " ,
"    1, sea ice
Padding_700_Strong(1,3)          =  16            ! cs*          "         "        "       "   " ,
"    1, land ice
```

```
Padding_700_Strong(2,0)           =  10            ! cs*            "         "        "       "   " ,
interval 2, water
Padding_700_Strong(2,1)           =  93            ! cs*            "         "        "       "   " ,
"    2, land
Padding_700_Strong(2,2)           =  93            ! cs*            "         "        "       "   " ,
"    2, sea ice
Padding_700_Strong(2,3)           =  93            ! cs*            "         "        "       "   " ,
"    2, land ice
Padding_700_Strong(3,0)           =  10            ! cs*            "         "        "       "   " ,
interval 3, water
Padding_700_Strong(3,1)           = 140            ! cs*            "         "        "       "   " ,
"    3, land
Padding_700_Strong(3,2)           = 140            ! cs*            "         "        "       "   " ,
"    3, sea ice
Padding_700_Strong(3,3)           = 140            ! cs*            "         "        "       "   " ,
"    3, land ice
Padding_700_Strong(4,0)           =  10            ! cs*            "         "        "       "   " ,
interval 4, water
Padding_700_Strong(4,1)           = 340            ! cs*            "         "        "       "   " ,
"    4, land
Padding_700_Strong(4,2)           = 340            ! cs*            "         "        "       "   " ,
"    4, sea ice
Padding_700_Strong(4,3)           = 340            ! cs*            "         "        "       "   " ,
"    4, land ice
Padding_700_Weak(1,0)             =  10            ! cs*         amount of Padding to add for DRM 700,
interval 1, water
Padding_700_Weak(1,1)             =  16            ! cs*            "         "        "       "   " ,
"    1, land
Padding_700_Weak(1,2)             =  10            ! cs*            "         "        "       "   " ,
"    1, sea ice
Padding_700_Weak(1,3)             =  16            ! cs*            "         "        "       "   " ,
"    1, land ice
Padding_700_Weak(2,0)             =  10            ! cs*            "         "        "       "   " ,
interval 2, water
Padding_700_Weak(2,1)             =  93            ! cs*            "         "        "       "   " ,
"    2, land
Padding_700_Weak(2,2)             =  93            ! cs*            "         "        "       "   " ,
"    2, sea ice
Padding_700_Weak(2,3)             =  93            ! cs*            "         "        "       "   " ,
"    2, land ice
Padding_700_Weak(3,0)             =  10            ! cs*            "         "        "       "   " ,
interval 3, water
Padding_700_Weak(3,1)             = 140            ! cs*            "         "        "       "   " ,
"    3, land
Padding_700_Weak(3,2)             = 140            ! cs*            "         "        "       "   " ,
"    3, sea ice
Padding_700_Weak(3,3)             = 140            ! cs*            "         "        "       "   " ,
"    3, land ice
Padding_700_Weak(4,0)             =  10            ! cs*            "         "        "       "   " ,
interval 4, water
Padding_700_Weak(4,1)             = 340            ! cs*            "         "        "       "   " ,
"    4, land
Padding_700_Weak(4,2)             = 340            ! cs*            "         "        "       "   " ,
"    4, sea ice
Padding_700_Weak(4,3)             = 340            ! cs*            "         "        "       "   " ,
"    4, land ice

Padding_140_Step_Strong(1)        =   126          ! cs*         DRM140 size interval limit for selecting
the Padding_140 element
Padding_140_Step_Strong(2)        =   378          ! cs*            "     "     "       "    "      "
"        "        "
Padding_140_Step_Strong(3)        =   882          ! cs*            "     "     "       "    "      "
"        "        "
Padding_140_Step_Weak(1)          =   126          ! cs*         DRM140 size interval limit for selecting
the Padding_140 element
Padding_140_Step_Weak(2)          =   378          ! cs*            "     "     "       "    "      "
"        "        "
Padding_140_Step_Weak(3)          =   882          ! cs*            "     "     "       "    "      "
"        "        "
```

```
Padding_700_Step_Strong(1)       =   126          ! cs*        DRM700 size interval limit for selecting
the Padding_700 element
Padding_700_Step_Strong(2)       =   378          ! cs*           "     "     "       "    "      "
"        "        "
Padding_700_Step_Strong(3)       =   882          ! cs*           "     "     "       "    "      "
"        "        "
Padding_700_Step_Weak(1)         =   126          ! cs*        DRM700 size interval limit for selecting
the Padding_700 element
Padding_700_Step_Weak(2)         =   378          ! cs*           "     "     "       "    "      "
"        "        "
Padding_700_Step_Weak(3)         =   882          ! cs*           "     "     "       "    "      "
"        "        "

DRM_Scaling_Strong(0)            =  1.D0          ! n/a        Scale factor to multiply DRM for water
DRM_Scaling_Strong(1)            =  2.D0          ! n/a          "     "     "     "     "   "  land
DRM_Scaling_Strong(2)            =  1.D0          ! n/a          "     "     "     "     "   "  sea ice
DRM_Scaling_Strong(3)            =  2.D0          ! n/a          "     "     "     "     "   "  land ice
DRM_Scaling_Weak(0)              =  1.D0          ! n/a        Scale factor to multiply DRM for water
DRM_Scaling_Weak(1)              =  2.D0          ! n/a          "     "     "     "     "   "  land
DRM_Scaling_Weak(2)              =  1.D0          ! n/a          "     "     "     "     "   "  sea ice
DRM_Scaling_Weak(3)              =  2.D0          ! n/a          "     "     "     "     "   "  land ice

Offset_140_Strong(0)             =  0             ! cs*        DRM140 Offset for water
Offset_140_Strong(1)             =  0             ! cs*          "      "     "  land
Offset_140_Strong(2)             =  0             ! cs*          "      "     "  sea ice
Offset_140_Strong(3)             =  0             ! cs*          "      "     "  land ice
Offset_140_Weak(0)               =  0             ! cs*        DRM140 Offset for water
Offset_140_Weak(1)               =  0             ! cs*          "      "     "  land
Offset_140_Weak(2)               =  0             ! cs*          "      "     "  sea ice
Offset_140_Weak(3)               =  0             ! cs*          "      "     "  land ice

Offset_700_Strong(0)             =  0             ! cs*        DRM700 Offset for water
Offset_700_Strong(1)             =  0             ! cs*          "      "     "  land
Offset_700_Strong(2)             =  0             ! cs*          "      "     "  sea ice
Offset_700_Strong(3)             =  0             ! cs*          "      "     "  land ice
Offset_700_Weak(0)               =  0             ! cs*        DRM700 Offset for water
Offset_700_Weak(1)               =  0             ! cs*          "      "     "  land
Offset_700_Weak(2)               =  0             ! cs*          "      "     "  sea ice
Offset_700_Weak(3)               =  0             ! cs*          "      "     "  land ice

Max_Data_Rate_Threshold_Strong   =  10500         ! events     Maximum data rate threshold for telemetry
(CURRENTLY NOT BEING USED)
Max_Data_Rate_Threshold_Weak     =  10500         ! events        "      "    "      "      "      "

Band_Hi_Limit_Strong(0)          =  1022          ! cs*        Maximum bandwidth to downlink for water
Band_Hi_Limit_Strong(1)          =  1022          ! cs*           "        "           "      "  land
Band_Hi_Limit_Strong(2)          =  1022          ! cs*           "        "           "      "  sea ice
Band_Hi_Limit_Strong(3)          =  1022          ! cs*           "        "           "      "  land
ice
Band_Hi_Limit_Weak(0)            =  1022          ! cs*        Maximum bandwidth to downlink for water
Band_Hi_Limit_Weak(1)            =  1022          ! cs*           "        "           "      "  land
Band_Hi_Limit_Weak(2)            =  1022          ! cs*           "        "           "      "  sea ice
Band_Hi_Limit_Weak(3)            =  1022          ! cs*           "        "           "      "  land
ice

Tm_Atmos_Strong                  =  TRUE          ! T-F        Telemeter atmosphere data
Tm_Atmos_Weak                    =  FALSE         ! T-f            "          "      "

DRM_for_SW_Bin_Size_Strong(0)    =  FALSE         ! T-F        Use DRM for software bin size, else use
table for water
DRM_for_SW_Bin_Size_Strong(1)    =  FALSE         ! T-F         "   "   "     "      "    "    "    "
"    "  land
DRM_for_SW_Bin_Size_Strong(2)    =  FALSE         ! T-F         "   "   "     "      "    "    "    "
"    "  sea ice
DRM_for_SW_Bin_Size_Strong(3)    =  FALSE         ! T-F         "   "   "     "      "    "    "    "
"    "  land ice
DRM_for_SW_Bin_Size_Weak(0)      =  FALSE         ! T-F         "   "   "     "      "    "    "    "
"    "  water
DRM_for_SW_Bin_Size_Weak(1)      =  FALSE         ! T-F         "   "   "     "      "    "    "    "
"    "  land
```

```
DRM_for_SW_Bin_Size_Weak(2)      =  FALSE         ! T-F         "   "   "     "      "    "    "    "
"    "  sea ice
DRM_for_SW_Bin_Size_Weak(3)      =  FALSE         ! T-F         "   "   "     "      "    "    "    "
"    "  land ice

Coastline_Relief_Flag_Strong(0)  =  TRUE          ! T-F        Flag to do Coastline relief latitude test
for water
Coastline_Relief_Flag_Strong(1)  =  TRUE          ! T-F         "   "  "      "       "       "      "
"  land
Coastline_Relief_Flag_Strong(2)  =  TRUE          ! T-F         "   "  "      "       "       "      "
"  sea ice
Coastline_Relief_Flag_Strong(3)  =  TRUE          ! T-F         "   "  "      "       "       "      "
"  land ice
Coastline_Relief_Flag_Weak(0)    =  TRUE          ! T-F         "   "  "      "       "       "      "
"  water
Coastline_Relief_Flag_Weak(1)    =  TRUE          ! T-F         "   "  "      "       "       "      "
"  land
Coastline_Relief_Flag_Weak(2)    =  TRUE          ! T-F         "   "  "      "       "       "      "
"  sea ice
Coastline_Relief_Flag_Weak(3)    =  TRUE          ! T-F         "   "  "      "       "       "      "
"  land ice

Coastline_Relief_North_Strong(0) =   60.D0        ! degrees    Southern limit of northern coastline
relief for water
Coastline_Relief_North_Strong(1) =   60.D0        ! degrees       "      "    "    "         "       "
"  land
Coastline_Relief_North_Strong(2) =   60.D0        ! degrees       "      "    "    "         "       "
"  sea ice
Coastline_Relief_North_Strong(3) =   60.D0        ! degrees       "      "    "    "         "       "
"  land ice
Coastline_Relief_North_Weak(0)   =   60.D0        ! degrees       "      "    "    "         "       "
"  water
Coastline_Relief_North_Weak(1)   =   60.D0        ! degrees       "      "    "    "         "       "
"  land
Coastline_Relief_North_Weak(2)   =   60.D0        ! degrees       "      "    "    "         "       "
"  sea ice
Coastline_Relief_North_Weak(3)   =   60.D0        ! degrees       "      "    "    "         "       "
"  land ice

Coastline_Relief_South_Strong(0) =  -60.D0        ! degrees    Northern limit of southern coastline
relief for water
Coastline_Relief_South_Strong(1) =  -60.D0        ! degrees       "      "    "    "         "       "
"  land
Coastline_Relief_South_Strong(2) =  -60.D0        ! degrees       "      "    "    "         "       "
"  sea ice
Coastline_Relief_South_Strong(3) =  -60.D0        ! degrees       "      "    "    "         "       "
"  land ice
Coastline_Relief_South_Weak(0)   =  -60.D0        ! degrees       "      "    "    "         "       "
"  water
Coastline_Relief_South_Weak(1)   =  -60.D0        ! degrees       "      "    "    "         "       "
"  land
Coastline_Relief_South_Weak(2)   =  -60.D0        ! degrees       "      "    "    "         "       "
"  sea ice
Coastline_Relief_South_Weak(3)   =  -60.D0        ! degrees       "      "    "    "         "       "
"  land ice

! Limit of background bin fraction in day/night determination
background_bins_fraction_strong  = 0.5D0       ! fraction        use the whole histogram if the DRM700
spans more than this fraction
background_bins_fraction_weak    = 0.5D0       !     "            "   "    "       "      "  "     "
"    "    "    "       "

TEPstart_strong                  = -6          ! cs   TEP telemetry band start from fire  (TEP delay in
Hw_Parms file)
TEPstart_weak                    = 0           ! cs   " "
TEPwidth_strong                  = 18          ! cs   TEP telemetry band width
TEPwidth_weak                    = 0           ! cs   " "

subwindow_min_strong(0)          = 8           ! cs   SF subwindow minimum, water
subwindow_min_weak(0)            = 8           ! cs   SF subwindow minimum, water
subwindow_min_strong(1)          = 8           ! cs   SF subwindow minimum, land
```

```
subwindow_min_weak(1)            = 8            ! cs   SF subwindow minimum, land
subwindow_min_strong(2)          = 8            ! cs   SF subwindow minimum, sea ice
subwindow_min_weak(2)            = 8            ! cs   SF subwindow minimum, sea ice
subwindow_min_strong(3)          = 8            ! cs   SF subwindow minimum, land ice
subwindow_min_weak(3)            = 8            ! cs   SF subwindow minimum, land ice
subwindow_max_strong(0)          = 700          ! cs   SF subwindow maximum, water
subwindow_max_weak(0)            = 700          ! cs   SF subwindow maximum, water
subwindow_max_strong(1)          = 700          ! cs   SF subwindow maximum, land
subwindow_max_weak(1)            = 700          ! cs   SF subwindow maximum, land
subwindow_max_strong(2)          = 700          ! cs   SF subwindow maximum, sea ice
subwindow_max_weak(2)            = 700          ! cs   SF subwindow maximum, sea ice
subwindow_max_strong(3)          = 700          ! cs   SF subwindow maximum, land ice
subwindow_max_weak(3)            = 700          ! cs   SF subwindow maximum, land ice

coastline_relief_strong          = 200          ! m    minimum coastline relief for telemetry
coastline_relief_weak            = 200          ! m    " "
coastline_scaling_strong         = 2.D0         ! n/a  coastline relief scaling for telemetry
coastline_scaling_weak           = 2.D0         ! n/a  " "

sw_bin_size_lower_limit_strong(0) =      4       ! cs              Minimum software binsize for water
sw_bin_size_lower_limit_strong(1) =      4       ! cs                 "       "        "     "  land
sw_bin_size_lower_limit_strong(2) =      4       ! cs                 "       "        "     "  sea ice
sw_bin_size_lower_limit_strong(3) =      4       ! cs                 "       "        "     "  land ice
sw_bin_size_lower_limit_weak(0)   =      4       ! cs                 "       "        "     "  water
sw_bin_size_lower_limit_weak(1)   =      4       ! cs                 "       "        "     "  land
sw_bin_size_lower_limit_weak(2)   =      4       ! cs                 "       "        "     "  sea ice
sw_bin_size_lower_limit_weak(3)   =      4       ! cs                 "       "        "     "  land ice
sw_bin_size_upper_limit_strong(0) =     48       ! cs              Maximum software binsize for water
sw_bin_size_upper_limit_strong(1) =     80       ! cs                 "       "        "     "  land
sw_bin_size_upper_limit_strong(2) =     48       ! cs                 "       "        "     "  sea ice
sw_bin_size_upper_limit_strong(3) =     48       ! cs                 "       "        "     "  land ice
sw_bin_size_upper_limit_weak(0)   =     48       ! cs                 "       "        "     "  water
sw_bin_size_upper_limit_weak(1)   =     80       ! cs                 "       "        "     "  land
sw_bin_size_upper_limit_weak(2)   =     48       ! cs                 "       "        "     "  sea ice
sw_bin_size_upper_limit_weak(3)   =     48       ! cs                 "       "        "     "  land ice

RW_AltimHist_PCE_Delay_Strong     =      4       ! cs        Margin to move start of the range window
and add to width to compensate for PCE delay
RW_AltimHist_PCE_Delay_Weak       =      6       ! cs          "    "   "     "   "   "    "     "     "
"  "    "   "       "      "   "    "

RW_AtmHist_PCE_Delay              =      3       ! cs        Margin to move start of the Atm range
window and add to width to compensate for PCE delay

! * cs   =  clock cycles

/
```

F.2 Algorithm Position, Pointing and Range (PPR) Parameters (for Track 1): v6 (launch) Parameters

```
! ATLAS flight software algorithm parameters for position, pointing and ranges (PPR 1)

&alg_parms_ppr_input

! ATLAS flight software algorithm parameters for position, pointing and ranges

! name                                 assignment        units        notes

Version_PPR                       = '0000000027'  ! n/a        Sequential number
Version_Date                      = '2017-07-11'  ! YYYY-MM-DD
Track                             = 1             ! n/a        Track number
Clock_Cycles_in_ns                = 10.0D0        ! ns         duration of a clock cycle

DEM_Delta_Limit_Strong            = 5500          ! m          Value of range width to move to next DEM
tier
DEM_Delta_Limit_Weak              = 5500          ! m            "    "   "     "   "   "   "   "    "

Range_Window_Width_Min_Strong(0,0) = 334          ! cs*        Minimum allowed range window width - day,
water
Range_Window_Width_Min_Strong(1,0) = 334          ! cs*           "       "      "      "     "   -
night, water
Range_Window_Width_Min_Strong(0,1) = 334          ! cs*           "       "      "      "     "   - day,
land
Range_Window_Width_Min_Strong(1,1) = 334          ! cs*           "       "      "      "     "   -
night, land
Range_Window_Width_Min_Strong(0,2) = 334          ! cs*           "       "      "      "     "   - day,
sea ice
Range_Window_Width_Min_Strong(1,2) = 3340         ! cs*           "       "      "      "     "   -
night, sea ice
Range_Window_Width_Min_Strong(0,3) = 334          ! cs*           "       "      "      "     "   - day,
land ice
Range_Window_Width_Min_Strong(1,3) = 3340         ! cs*           "       "      "      "     "   -
night, land ice
Range_Window_Width_Min_Weak(0,0)   = 334          ! cs*           "       "      "      "     "   - day,
water
Range_Window_Width_Min_Weak(1,0)   = 334          ! cs*           "       "      "      "     "   -
night, water
Range_Window_Width_Min_Weak(0,1)   = 334          ! cs*           "       "      "      "     "   - day,
land
Range_Window_Width_Min_Weak(1,1)   = 334          ! cs*           "       "      "      "     "   -
night, land
Range_Window_Width_Min_Weak(0,2)   = 334          ! cs*           "       "      "      "     "   - day,
sea ice
Range_Window_Width_Min_Weak(1,2)   = 3340         ! cs*           "       "      "      "     "   -
night, sea ice
Range_Window_Width_Min_Weak(0,3)   = 334          ! cs*           "       "      "      "     "   - day,
land ice
Range_Window_Width_Min_Weak(1,3)   = 3340         ! cs*           "       "      "      "     "   -
night, land ice

Range_Window_Width_Max_Strong(0,0) = 4000         ! cs*        Maximum allowed range window width - day,
water
Range_Window_Width_Max_Strong(1,0) = 4000         ! cs*           "       "      "      "     "   -
night, water
Range_Window_Width_Max_Strong(0,1) = 4000         ! cs*           "       "      "      "     "   - day,
land
Range_Window_Width_Max_Strong(1,1) = 4000         ! cs*           "       "      "      "     "   -
night, land
Range_Window_Width_Max_Strong(0,2) = 4000         ! cs*           "       "      "      "     "   - day,
sea ice
Range_Window_Width_Max_Strong(1,2) = 4000         ! cs*           "       "      "      "     "   -
night, sea ice
Range_Window_Width_Max_Strong(0,3) = 4000         ! cs*           "       "      "      "     "   - day,
land ice
Range_Window_Width_Max_Strong(1,3) = 4000         ! cs*           "       "      "      "     "   -
night, land ice
```

```
Range_Window_Width_Max_Weak(0,0)   = 4000         ! cs*          "       "      "      "     "   - day,
water
Range_Window_Width_Max_Weak(1,0)   = 4000         ! cs*          "       "      "      "     "   -
night, water
Range_Window_Width_Max_Weak(0,1)   = 4000         ! cs*          "       "      "      "     "   - day,
land
Range_Window_Width_Max_Weak(1,1)   = 4000         ! cs*          "       "      "      "     "   -
night, land
Range_Window_Width_Max_Weak(0,2)   = 4000         ! cs*          "       "      "      "     "   - day,
sea ice
Range_Window_Width_Max_Weak(1,2)   = 4000         ! cs*          "       "      "      "     "   -
night, sea ice
Range_Window_Width_Max_Weak(0,3)   = 4000         ! cs*          "       "      "      "     "   - day,
land ice
Range_Window_Width_Max_Weak(1,3)   = 4000         ! cs*          "       "      "      "     "   -
night, land ice

Atmos_Range_Window_Offset_Strong  = 0             ! cs*       Range window offset for the atmsopheric
histogram
Atmos_Range_Window_Offset_Weak    = 0             ! cs*         "      "     "     "   "       "
"

Range_Window_Offset_Strong(0)     = 0             ! cs*       Range offset (distance below the
estimated surface) for water
Range_Window_Offset_Strong(1)     = 0             ! cs*         "     "        "       "    "     "
"      "  land
Range_Window_Offset_Strong(2)     = 0             ! cs*         "     "        "       "    "     "
"      "  sea ice
Range_Window_Offset_Strong(3)     = 0             ! cs*         "     "        "       "    "     "
"      "  ice sheet
Range_Window_Offset_Weak(0)       = 0             ! cs*       Range offset (distance below the
estimated surface) for water
Range_Window_Offset_Weak(1)       = 0             ! cs*         "     "        "       "    "     "
"      "  land
Range_Window_Offset_Weak(2)       = 0             ! cs*         "     "        "       "    "     "
"      "  sea ice
Range_Window_Offset_Weak(3)       = 0             ! cs*         "     "        "       "    "     "
"      "  ice sheet

Range_Decrease_Limit_Strong       = 320           ! cs*       Limitation on range window start time
decrease between successive frames
Range_Decrease_Limit_Weak         = 320           ! cs*           "     "    "     "      "     "

Range_Window_DEM_Margin_Strong(0) = 167           ! cs*       Margin to be added to delta DEM to give
range window width for water
Range_Window_DEM_Margin_Strong(1) = 167           ! cs*          "   "   "   "   "    "    "   "   "
"      "     "    "  land
Range_Window_DEM_Margin_Strong(2) = 167           ! cs*          "   "   "   "   "    "    "   "   "
"      "     "    "  sea ice
Range_Window_DEM_Margin_Strong(3) = 167           ! cs*          "   "   "   "   "    "    "   "   "
"      "     "    "  ice sheet
Range_Window_DEM_Margin_Weak(0)   = 167           ! cs*       Margin to be added to delta DEM to give
range window width for water
Range_Window_DEM_Margin_Weak(1)   = 167           ! cs*          "   "   "   "   "    "    "   "   "
"      "     "    "  land
Range_Window_DEM_Margin_Weak(2)   = 167           ! cs*          "   "   "   "   "    "    "   "   "
"      "     "    "  sea ice
Range_Window_DEM_Margin_Weak(3)   = 167           ! cs*          "   "   "   "   "    "    "   "   "
"      "     "    "  ice sheet

Atm14km10ns_Strong                = 9340          ! cs*       Clock cycles per 14 km atmosphere segment
Atm14km10ns_Weak                  = 9340          ! cs*         "     "     "  "  "      "        "

Velocity_Average_Points_Strong    = 10            ! n/a       Number of velocities to average for
extrapolation
Velocity_Average_Points_Weak      = 10            ! n/a          "    "     "      "     "     "

Range_Rate_Average_Points_Strong  = 2             ! n/a       Number of ranges to use in range rate
computation
Range_Rate_Average_Points_Weak    = 2             ! n/a          "    "    "    "  "   "  "      "
```

```
Beam_Pointing_Strong(1)           = -0.002545050D0 ! n/a        Unit vectors of the beams
Beam_Pointing_Strong(2)           =  0.006675060D0 ! n/a          "     "    "   "    "
Beam_Pointing_Strong(3)           =  0.999974000D0 ! n/a          "     "    "   "    "
Beam_Pointing_Weak(1)             =  0.002444363D0 ! n/a          "     "    "   "    "
Beam_Pointing_Weak(2)             =  0.006654247D0 ! n/a          "     "    "   "    "
Beam_Pointing_Weak(3)             =  0.999974873D0 ! n/a          "     "    "   "    "

Cosine_Solar_Zenith_Angle_Cutoff_Strong = 0.14D0  ! n/a        Minimum value for daylight in checking
range window max/min limits
Cosine_Solar_Zenith_Angle_Cutoff_Weak   = 0.14D0  ! n/a           "      "    "     "     "     "

Max_Sep_WeakStrong                = 8820          ! cs*        Max separation between start of one range
window and end of other in a track

RW_AltimHist_PCE_Delay_Strong     = 5             ! cs         Margin to move start of the range window
and add to width to compensate for PCE delay
RW_AltimHist_PCE_Delay_Weak       = 7             ! cs           "    "   "     "   "   "    "     "

RW_AtmHist_PCE_Delay              = 4             ! cs         Margin to move start of the Atm range
window and add to width to compensate for PCE delay

! * cs = clock cycles
/
```

F.3 Algorithm Knobs Parameters (for Track 1): v6 (launch) Parameters

```
! ATLAS flight software algorithm knobs 1

&alg_parms_knobs_input

! ATLAS flight software algorithm knobs
! WARNING: For NOGO states, lower level flags set to PASS.
!
Version_Knobs = '0000000024'  ! sequential number
Version_Date  = '2016-01-25'  ! YYYY-MM-DD
Track         = 1             ! track number
!
! Index descriptions (2,4) = (0=day, 1=night)
!                            (0=water, 1=land, 2=sea ice, 3=land ice)

! Knobs for telemetering data.
! 0 = do not telemeter under any conditions
! 1 = IF MF signal is found, do telemeter data.
!     If MF signal is not found, check the *NoMFSig* flag.
Tm_Strong(0,0) = 1   ! day,   water
Tm_Strong(1,0) = 1   ! night, water
Tm_Strong(0,1) = 1   ! day,   land
Tm_Strong(1,1) = 1   ! night, land
Tm_Strong(0,2) = 1   ! day,   sea ice
Tm_Strong(1,2) = 1   ! night, sea ice
Tm_Strong(0,3) = 1   ! day,   land ice
Tm_Strong(1,3) = 1   ! night, land ice
Tm_Weak(0,0)   = 0   ! day,   water
Tm_Weak(1,0)   = 0   ! night, water
Tm_Weak(0,1)   = 1   ! day,   land
Tm_Weak(1,1)   = 1   ! night, land
Tm_Weak(0,2)   = 1   ! day,   sea ice
Tm_Weak(1,2)   = 1   ! night, sea ice
Tm_Weak(0,3)   = 1   ! day,   land ice
Tm_Weak(1,3)   = 1   ! night, land ice

! Knobs for telemetering data if MF signal is NOT found and *Tm*=1
! This flag is only checked if no MF signal was found
! 0 = do not telemeter data
! 1 = if SF signal is found, do telemeter data
!     if SF signal is not found, go to next test.
Tm_NoMFSig_Strong(0,0)  = 0   ! day,   water
Tm_NoMFSig_Strong(1,0)  = 0   ! night, water
Tm_NoMFSig_Strong(0,1)  = 1   ! day,   land
Tm_NoMFSig_Strong(1,1)  = 1   ! night, land
Tm_NoMFSig_Strong(0,2)  = 1   ! day,   sea ice
Tm_NoMFSig_Strong(1,2)  = 1   ! night, sea ice
Tm_NoMFSig_Strong(0,3)  = 1   ! day,   land ice
Tm_NoMFSig_Strong(1,3)  = 1   ! night, land ice
Tm_NoMFSig_Weak(0,0)    = 1   ! day,   water -- not reached
Tm_NoMFSig_Weak(1,0)    = 1   ! night, water -- not reached
Tm_NoMFSig_Weak(0,1)    = 1   ! day,   land
Tm_NoMFSig_Weak(1,1)    = 1   ! night, land
Tm_NoMFSig_Weak(0,2)    = 1   ! day,   sea ice
Tm_NoMFSig_Weak(1,2)    = 1   ! night, sea ice
Tm_NoMFSig_Weak(0,3)    = 1   ! day,   land ice
Tm_NoMFSig_Weak(1,3)    = 1   ! night, land ice

! Knobs for telemetering data if SF signal is not found and *Tm*=1 and *NoMF*=1
! This flag is only checked if signal was not found in either the MF or the SF.
! 0 = do not telemeter data
! 1 = if SF signal is not found, go on to Cloud Algorithm flag.
Tm_NoSFSig_Strong(0,0)  = 1   ! day,   water -- not reached
Tm_NoSFSig_Strong(1,0)  = 1   ! night, water -- not reached
Tm_NoSFSig_Strong(0,1)  = 1   ! day,   land
Tm_NoSFSig_Strong(1,1)  = 1   ! night, land
Tm_NoSFSig_Strong(0,2)  = 1   ! day,   sea ice
Tm_NoSFSig_Strong(1,2)  = 1   ! night, sea ice
Tm_NoSFSig_Strong(0,3)  = 1   ! day,   land ice
Tm_NoSFSig_Strong(1,3)  = 1   ! night, land ice
Tm_NoSFSig_Weak(0,0)    = 1   ! day,   water -- not reached
Tm_NoSFSig_Weak(1,0)    = 1   ! night, water -- not reached
Tm_NoSFSig_Weak(0,1)    = 0   ! day,   land
Tm_NoSFSig_Weak(1,1)    = 0   ! night, land
Tm_NoSFSig_Weak(0,2)    = 1   ! day,   sea ice
Tm_NoSFSig_Weak(1,2)    = 1   ! night, sea ice
Tm_NoSFSig_Weak(0,3)    = 1   ! day,   land ice
Tm_NoSFSig_Weak(1,3)    = 1   ! night, land ice

! Knobs for using cloud algorithm to determine if data are to be sent down if no
! signal is found.
! 0 = do not run cloud algorithm. Check the *NoSig* flag.
! 1 = run cloud algorithm. If it ==> thick clouds, do not telemeter
!     Otherwise do telemeter data based on the NoSig parms (timer, relief, scale)
Cloud_Alg_Strong(0,0)      = 1   ! day,   water -- not reached
Cloud_Alg_Strong(1,0)      = 0   ! night, water -- not reached
Cloud_Alg_Strong(0,1)      = 1   ! day,   land
Cloud_Alg_Strong(1,1)      = 0   ! night, land
Cloud_Alg_Strong(0,2)      = 1   ! day,   sea ice
Cloud_Alg_Strong(1,2)      = 0   ! night, sea ice
Cloud_Alg_Strong(0,3)      = 0   ! day,   land ice
Cloud_Alg_Strong(1,3)      = 0   ! night, land ice
Cloud_Alg_Weak(0,0)        = 1   ! day,   water -- not reached
Cloud_Alg_Weak(1,0)        = 0   ! night, water -- not reached
Cloud_Alg_Weak(0,1)        = 1   ! day,   land -- not reached
Cloud_Alg_Weak(1,1)        = 0   ! night, land -- not reached
Cloud_Alg_Weak(0,2)        = 1   ! day,   sea ice
Cloud_Alg_Weak(1,2)        = 0   ! night, sea ice
Cloud_Alg_Weak(0,3)        = 1   ! day,   land ice
Cloud_Alg_Weak(1,3)        = 0   ! night, land ice

! Knobs for telemetering data if no signal is found and the logic says to run
! this test.
! 0 = do not telemeter data
! 1 = do telemeter data based on the NoSig parms (timer, relief, scale)
Tm_NoSig_Strong(0,0) = 1   ! day,   water -- not reached
Tm_NoSig_Strong(1,0) = 1   ! night, water -- not reached
Tm_NoSig_Strong(0,1) = 1   ! day,   land -- Cloud Alg path taken
Tm_NoSig_Strong(1,1) = 1   ! night, land
Tm_NoSig_Strong(0,2) = 1   ! day,   sea ice -- Cloud Alg path taken
Tm_NoSig_Strong(1,2) = 1   ! night, sea ice
Tm_NoSig_Strong(0,3) = 1   ! day,   land ice
Tm_NoSig_Strong(1,3) = 1   ! night, land ice
Tm_NoSig_Weak(0,0)   = 1   ! day,   water -- not reached
Tm_NoSig_Weak(1,0)   = 1   ! night, water -- not reached
Tm_NoSig_Weak(0,1)   = 1   ! day,   land -- not reached
Tm_NoSig_Weak(1,1)   = 1   ! night, land -- not reached
Tm_NoSig_Weak(0,2)   = 1   ! day,   sea ice -- Cloud Alg path taken
Tm_NoSig_Weak(1,2)   = 1   ! night, sea ice
Tm_NoSig_Weak(0,3)   = 1   ! day,   land ice -- Cloud Alg path taken
Tm_NoSig_Weak(1,3)   = 1   ! night, land ice

! Knobs for relief of telemetry band if major frame signal is found
! Values of relief are 0 = DRM140, 1 = DRM700, 2 = DEM (actually, Range Window in this context)
DRMDEM_MFSig_Strong(0,0) = 0   ! day,   water
DRMDEM_MFSig_Strong(1,0) = 0   ! night, water
DRMDEM_MFSig_Strong(0,1) = 0   ! day,   land
DRMDEM_MFSig_Strong(1,1) = 0   ! night, land
DRMDEM_MFSig_Strong(0,2) = 0   ! day,   sea ice
DRMDEM_MFSig_Strong(1,2) = 0   ! night, sea ice
DRMDEM_MFSig_Strong(0,3) = 0   ! day,   land ice
DRMDEM_MFSig_Strong(1,3) = 0   ! night, land ice
DRMDEM_MFSig_Weak(0,0)   = 0   ! day,   water -- not reached
DRMDEM_MFSig_Weak(1,0)   = 0   ! night, water -- not reached
DRMDEM_MFSig_Weak(0,1)   = 0   ! day,   land
DRMDEM_MFSig_Weak(1,1)   = 0   ! night, land
DRMDEM_MFSig_Weak(0,2)   = 0   ! day,   sea ice
DRMDEM_MFSig_Weak(1,2)   = 0   ! night, sea ice
DRMDEM_MFSig_Weak(0,3)   = 0   ! day,   land ice
DRMDEM_MFSig_Weak(1,3)   = 0   ! night, land ice

! Knobs for relief of telemetry band if super frame signal is found
! Values of relief are 1 = DRM700, 2 = DEM (actually, Range Window in this context)
DRMDEM_SFSig_Strong(0,0) = 1   ! day,   water -- not reached
DRMDEM_SFSig_Strong(1,0) = 1   ! night, water -- not reached
DRMDEM_SFSig_Strong(0,1) = 1   ! day,   land
DRMDEM_SFSig_Strong(1,1) = 1   ! night, land
DRMDEM_SFSig_Strong(0,2) = 1   ! day,   sea ice
DRMDEM_SFSig_Strong(1,2) = 1   ! night, sea ice
DRMDEM_SFSig_Strong(0,3) = 1   ! day,   land ice
DRMDEM_SFSig_Strong(1,3) = 1   ! night, land ice
DRMDEM_SFSig_Weak(0,0)   = 1   ! day,   water -- not reached
DRMDEM_SFSig_Weak(1,0)   = 1   ! night, water -- not reached
DRMDEM_SFSig_Weak(0,1)   = 1   ! day,   land
DRMDEM_SFSig_Weak(1,1)   = 1   ! night, land
DRMDEM_SFSig_Weak(0,2)   = 1   ! day,   sea ice
DRMDEM_SFSig_Weak(1,2)   = 1   ! night, sea ice
DRMDEM_SFSig_Weak(0,3)   = 1   ! day,   land ice
DRMDEM_SFSig_Weak(1,3)   = 1   ! night, land ice

! Threshold to determine if it is day or night
! If computed background noise is greater than this threshold, then it's day
Day_Night_Threshold_Strong(0)  =  0.8D6  ! counts, water
Day_Night_Threshold_Strong(1)  =  0.8D6  ! counts, land
Day_Night_Threshold_Strong(2)  =  0.8D6  ! counts, sea ice
Day_Night_Threshold_Strong(3)  =  0.8D6  ! counts, land ice
Day_Night_Threshold_Weak(0)    =  0.8D6  ! counts, water
Day_Night_Threshold_Weak(1)    =  0.8D6  ! counts, land
Day_Night_Threshold_Weak(2)    =  0.8D6  ! counts, sea ice
Day_Night_Threshold_Weak(3)    =  0.8D6  ! counts, land ice

! Software channel disable flags.
! If a SW channel is disabled, data collected by that channel is not telemetered
SWChDisable_Water_Day_Strong    =       983040  ! == 0x000F0000 (disable weak chans 16-19)
SWChDisable_Water_Night_Strong  =       983040  ! == 0x000F0000 " "
SWChDisable_Land_Day_Strong     =       983040  ! == 0x000F0000 " "
SWChDisable_Land_Night_Strong   =       983040  ! == 0x000F0000 " "
SWChDisable_SeaIce_Day_Strong   =       983040  ! == 0x000F0000 " "
SWChDisable_SeaIce_Night_Strong =       983040  ! == 0x000F0000 " "
SWChDisable_Ice_Day_Strong      =       983040  ! == 0x000F0000 " "
SWChDisable_Ice_Night_Strong    =       983040  ! == 0x000F0000 " "
SWChDisable_Water_Day_Weak      =       65535   ! == 0x0000FFFF (disable strong chans 0-15)
SWChDisable_Water_Night_Weak    =       65535   ! == 0x0000FFFF " "
SWChDisable_Land_Day_Weak       =       65535   ! == 0x0000FFFF " "
SWChDisable_Land_Night_Weak     =       65535   ! == 0x0000FFFF " "
SWChDisable_SeaIce_Day_Weak     =       65535   ! == 0x0000FFFF " "
SWChDisable_SeaIce_Night_Weak   =       65535   ! == 0x0000FFFF " "
SWChDisable_Ice_Day_Weak        =       65535   ! == 0x0000FFFF " "
SWChDisable_Ice_Night_Weak      =       65535   ! == 0x0000FFFF " "

! Nosig_Relief during timer 1:
! Values of relief are 0 = DRM140, 1 = DRM700, 2 = DEM (actually, Range Window in this context)
Nosig_Relief1_Strong(0,0) = 1   ! day,   water -- not reached
Nosig_Relief1_Strong(1,0) = 1   ! night, water -- not reached
Nosig_Relief1_Strong(0,1) = 1   ! day,   land
Nosig_Relief1_Strong(1,1) = 1   ! night, land
Nosig_Relief1_Strong(0,2) = 1   ! day,   sea ice
Nosig_Relief1_Strong(1,2) = 1   ! night, sea ice
Nosig_Relief1_Strong(0,3) = 1   ! day,   land ice
Nosig_Relief1_Strong(1,3) = 1   ! night, land ice
Nosig_Relief1_Weak(0,0) = 1     ! day,   water -- not reached
Nosig_Relief1_Weak(1,0) = 1     ! night, water -- not reached
Nosig_Relief1_Weak(0,1) = 1     ! day,   land -- not reached
Nosig_Relief1_Weak(1,1) = 1     ! night, land -- not reached
Nosig_Relief1_Weak(0,2) = 1     ! day,   sea ice
Nosig_Relief1_Weak(1,2) = 1     ! night, sea ice
Nosig_Relief1_Weak(0,3) = 1     ! day,   land ice
Nosig_Relief1_Weak(1,3) = 1     ! night, land ice

! Nosig_Relief during timer 2:
```

```
! Values of relief are 1 = DRM700, 2 = DEM (actually, Range Window in this context)
Nosig_Relief2_Strong(0,0) = 2   ! day,   water -- not reached
Nosig_Relief2_Strong(1,0) = 2   ! night, water -- not reached
Nosig_Relief2_Strong(0,1) = 2   ! day,   land
Nosig_Relief2_Strong(1,1) = 2   ! night, land
Nosig_Relief2_Strong(0,2) = 2   ! day,   sea ice
Nosig_Relief2_Strong(1,2) = 2   ! night, sea ice
Nosig_Relief2_Strong(0,3) = 2   ! day,   land ice
Nosig_Relief2_Strong(1,3) = 2   ! night, land ice
Nosig_Relief2_Weak(0,0) = 2     ! day,   water -- not reached
Nosig_Relief2_Weak(1,0) = 2     ! night, water -- not reached
Nosig_Relief2_Weak(0,1) = 2     ! day,   land -- not reached
Nosig_Relief2_Weak(1,1) = 2     ! night, land -- not reached
Nosig_Relief2_Weak(0,2) = 2     ! day,   sea ice
Nosig_Relief2_Weak(1,2) = 2     ! night, sea ice
Nosig_Relief2_Weak(0,3) = 2     ! day,   land ice
Nosig_Relief2_Weak(1,3) = 2     ! night, land ice

! Number of major frames with no signal to telemeter
! Timer = -1 means to never stop sending data
Nosig_Timer1_Strong(0,0) =  25   ! day,   water -- not reached
Nosig_Timer1_Strong(1,0) =  25   ! night, water -- not reached
Nosig_Timer1_Strong(0,1) =  50   ! day,   land
Nosig_Timer1_Strong(1,1) =  50   ! night, land
Nosig_Timer1_Strong(0,2) =  25   ! day,   sea ice
Nosig_Timer1_Strong(1,2) =  25   ! night, sea ice
Nosig_Timer1_Strong(0,3) =  50   ! day,   land ice
Nosig_Timer1_Strong(1,3) =  50   ! night, land ice
Nosig_Timer1_Weak(0,0) =  25     ! day,   water -- not reached
Nosig_Timer1_Weak(1,0) =  25     ! night, water -- not reached
Nosig_Timer1_Weak(0,1) =  50     ! day,   land -- not reached
Nosig_Timer1_Weak(1,1) =  50     ! night, land -- not reached
Nosig_Timer1_Weak(0,2) =  25     ! day,   sea ice
Nosig_Timer1_Weak(1,2) =  25     ! night, sea ice
Nosig_Timer1_Weak(0,3) =  50     ! day,   land ice
Nosig_Timer1_Weak(1,3) =  50     ! night, land ice

! Number of major frames with no signal to telemeter after timer 1 has passed
! Timer = -1 means to never stop sending data
Nosig_Timer2_Strong(0,0) =  -1   ! day,   water -- not reached
Nosig_Timer2_Strong(1,0) =  -1   ! night, water -- not reached
Nosig_Timer2_Strong(0,1) = 100   ! day,   land
Nosig_Timer2_Strong(1,1) = 100   ! night, land
Nosig_Timer2_Strong(0,2) =  -1   ! day,   sea ice
Nosig_Timer2_Strong(1,2) =  -1   ! night, sea ice
Nosig_Timer2_Strong(0,3) =  -1   ! day,   land ice
Nosig_Timer2_Strong(1,3) =  -1   ! night, land ice
Nosig_Timer2_Weak(0,0) =  1000   ! day,   water -- not reached
Nosig_Timer2_Weak(1,0) =  1000   ! night, water -- not reached
Nosig_Timer2_Weak(0,1) =   100   ! day,   land -- not reached
Nosig_Timer2_Weak(1,1) =   100   ! night, land -- not reached
Nosig_Timer2_Weak(0,2) =    -1   ! day,   sea ice
Nosig_Timer2_Weak(1,2) =    -1   ! night, sea ice
Nosig_Timer2_Weak(0,3) =    -1   ! day,   land ice
Nosig_Timer2_Weak(1,3) =    -1   ! night, land ice

! Nosig_Scale:
! Scale factor to be applied to relief during timer 1
Nosig_Scale1_Strong(0,0) = 2.0D0   ! day,   water -- not reached
Nosig_Scale1_Strong(1,0) = 2.0D0   ! night, water -- not reached
Nosig_Scale1_Strong(0,1) = 2.0D0   ! day,   land
Nosig_Scale1_Strong(1,1) = 2.0D0   ! night, land
Nosig_Scale1_Strong(0,2) = 2.0D0   ! day,   sea ice
Nosig_Scale1_Strong(1,2) = 2.0D0   ! night, sea ice
Nosig_Scale1_Strong(0,3) = 2.0D0   ! day,   land ice
Nosig_Scale1_Strong(1,3) = 2.0D0   ! night, land ice
Nosig_Scale1_Weak(0,0) = 2.0D0     ! day,   water -- not reached
Nosig_Scale1_Weak(1,0) = 2.0D0     ! night, water -- not reached
Nosig_Scale1_Weak(0,1) = 2.0D0     ! day,   land -- not reached
Nosig_Scale1_Weak(1,1) = 2.0D0     ! night, land -- not reached
Nosig_Scale1_Weak(0,2) = 2.0D0     ! day,   sea ice
```

```
Nosig_Scale1_Weak(1,2) = 2.0D0     ! night, sea ice
Nosig_Scale1_Weak(0,3) = 2.0D0     ! day,   land ice
Nosig_Scale1_Weak(1,3) = 2.0D0     ! night, land ice

! Nosig_Scale:
! Scale factor to be applied to relief during timer 2
Nosig_Scale2_Strong(0,0) = 1.0D0   ! day,   water -- not reached
Nosig_Scale2_Strong(1,0) = 1.0D0   ! night, water -- not reached
Nosig_Scale2_Strong(0,1) = 1.0D0   ! day,   land
Nosig_Scale2_Strong(1,1) = 1.0D0   ! night, land
Nosig_Scale2_Strong(0,2) = 1.0D0   ! day,   sea ice
Nosig_Scale2_Strong(1,2) = 1.0D0   ! night, sea ice
Nosig_Scale2_Strong(0,3) = 1.0D0   ! day,   land ice
Nosig_Scale2_Strong(1,3) = 1.0D0   ! night, land ice
Nosig_Scale2_Weak(0,0) = 1.0D0     ! day,   water -- not reached
Nosig_Scale2_Weak(1,0) = 1.0D0     ! night, water -- not reached
Nosig_Scale2_Weak(0,1) = 1.0D0     ! day,   land -- not reached
Nosig_Scale2_Weak(1,1) = 1.0D0     ! night, land -- not reached
Nosig_Scale2_Weak(0,2) = 1.0D0     ! day,   sea ice
Nosig_Scale2_Weak(1,2) = 1.0D0     ! night, sea ice
Nosig_Scale2_Weak(0,3) = 1.0D0     ! day,   land ice
Nosig_Scale2_Weak(1,3) = 1.0D0     ! night, land ice

! Telemetry band location for Nosig_Timer2 cases
NoSig_DEM_fraction_Strong(0) = 0.5D0 ! water -- not reached
NoSig_DEM_fraction_Strong(1) = 0.5D0 ! land
NoSig_DEM_fraction_Strong(2) = 0.5D0 ! sea ice
NoSig_DEM_fraction_Strong(3) = 0.5D0 ! land ice
NoSig_DEM_fraction_Weak(0)   = 0.5D0 ! water -- not reached
NoSig_DEM_fraction_Weak(1)   = 0.5D0 ! land -- not reached
NoSig_DEM_fraction_Weak(2)   = 0.5D0 ! sea ice
NoSig_DEM_fraction_Weak(3)   = 0.5D0 ! land ice

! Flag to reject TEP region as signal
TEP_not_strong = 1   ! reject TEP region as signal (1 = reject, 0 = do nothing)
TEP_not_weak   = 0   ! " "

! Tertiary signal flag for telemetry priority
tertiary_signal_flag_strong(0) = 0   ! flag values '1' gives preference to tertiary over secondary, '0'
is vice-versa
tertiary_signal_flag_strong(1) = 0   ! " "
tertiary_signal_flag_strong(2) = 0   ! " "
tertiary_signal_flag_strong(3) = 0   ! " "
tertiary_signal_flag_weak(0)   = 0   ! " "
tertiary_signal_flag_weak(1)   = 0   ! " "
tertiary_signal_flag_weak(2)   = 0   ! " "
tertiary_signal_flag_weak(3)   = 0   ! " "

! TEP flag for telemetry
TEPflag_strong(0) = 0   ! flag values '0' means telemeter TEP as secondary band for TM cases 0, 1, 3 and
8
TEPflag_strong(1) = 0   ! " "
TEPflag_strong(2) = 0   ! " "
TEPflag_strong(3) = 0   ! " "
TEPflag_weak(0)   = 0   ! " "
TEPflag_weak(1)   = 0   ! " "
TEPflag_weak(2)   = 0   ! " "
TEPflag_weak(3)   = 0   ! " "
                        ! '1' means telemeter TEP as secondary band for all cases

/
```

Appendix G: Parameter Type Definitions

The following tables are from the ATLAS Flight Software Team.

SIGNAL/TELEMETERY (ST) PARAMETER FILE

Variable Name	Data Type
Version_ST (64 characters max)	String
Version_Date (64 characters max)	String
Track	UINT32
Clock_Cycles_in_ns	DOUBLE
Nsf_Strong	UINT32
Nsf_Weak	UINT32
Bin_Size_Strong(x),x=0,1,2,3	UINT32
Bin_Size_Weak(x),x=0,1,2,3	UINT32
Sigma_For_Significance_Strong	DOUBLE
Sigma_For_Significance_Weak	DOUBLE
Min_Secondary_SWbin_Separation	DOUBLE
Min_Counts_For_Signal_Strong	UINT32
Min_Counts_For_Signal_Weak	UINT32
Cloud_Scale_Factor_Strong	DOUBLE
Cloud_Scale_Factor_Weak	DOUBLE
Cloud_Bins_Exclude_Strong	INT32
Cloud_Bins_Exclude_Weak	INT32
Cloud_Threshold_Strong	UINT32
Cloud_Threshold_Weak	UINT32
Atm_Histogram_Width_Strong	UINT32
Atm_Histogram_Width_Weak	UINT32
Lbin_Strong	UINT32
Lbin_Weak	UINT32
Padding_140_Strong(x,y),x=1,2,3,4, y=0,1,2,3	UINT32
Padding_140_Weak(x,y),x=1,2,3,4, y=0,1,2,3	UINT32
Padding_700_Strong(x,y),x=1,2,3,4, y=0,1,2,3	UINT32
Padding_700_Weak(x,y),x=1,2,3,4, y=0,1,2,3	UINT32
Padding_140_Step_Strong(x),x=1,2,3	UINT32
Padding_140_Step_Weak(x),x=1,2,3	UINT32
Padding_700_Step_Strong(x),x=1,2,3	UINT32
Padding_700_Step_Weak(x),x=1,2,3	UINT32
DRM_Scaling_Strong(x),x=0,1,2,3	DOUBLE
DRM_Scaling_Weak(x),x=0,1,2,3	DOUBLE
Offset_140_Strong(x),x=0,1,2,3	INT32

Variable Name	Data Type
Offset_140_Weak(x),x=0,1,2,3	INT32
Offset_700_Strong(x),x=0,1,2,3	INT32
Offset_700_Weak(x),x=0,1,2,3	INT32
Max_Data_Rate_Threshold_Strong	UINT32
Max_Data_Rate_Threshold_Weak	UINT32
Band_Hi_Limit_Strong(x),x=0,1,2,3	UINT32
Band_Hi_Limit_Weak(x),x=0,1,2,3	UINT32
DRM_for_SW_Bin_Size_Strong(x),x=0,1,2,3	BOOLEAN
DRM_for_SW_Bin_Size_Weak(x),x=0,1,2,3	BOOLEAN
Coastline_Relief_Flag_Strong(x),x=0,1,2,3	BOOLEAN
Coastline_Relief_Flag_Weak(x),x=0,1,2,3	BOOLEAN
Coastline_Relief_North_Strong(x),x=0,1,2,3	DOUBLE
Coastline_Reief_North_Weak(x),x=0,1,2,3	DOUBLE
Coastline_Relief_South_Strong(x),x=0,1,2,3	DOUBLE
Coastline_Relief_South_Weak(x),x=0,1,2,3	DOUBLE
Background_bins_fraction_strong	DOUBLE
Background_bins_fraction_weak	DOUBLE
TEPstart_strong	INT32
TEPstart_weak	INT32
TEPwidth_strong	UINT32
TEPwidth_weak	UINT32
Subwindow_min_strong(x),x=0,1,2,3	UINT32
Subwindow_min_weak(x),x=0,1,2,3	UINT32
Subwindow_max_strong(x),x=0,1,2,3	UINT32
Subwindow_max_weak(x),x=0,1,2,3	UINT32
Coastline_relief_strong	UINT32
Coastline_relief_weak	UINT32
Coastline_scaling_strong	DOUBLE
Coastline_scaling_weak	DOUBLE
SW_bin_size_lower_limit_strong(x),x=0,1,2,3	UINT32
SW_bin_size_lower_limit_weak(x),x=0,1,2,3	UINT32
SW_bin_size_upper_limit_strong(x),x=0,1,2,3	UINT32
SW_bin_size_upper_limit_weak(x),x=0,1,2,3	UINT32
RW_AltimHist_PCE_Delay_Strong	UINT32
RW_AltimHist_PCE_Delay_Weak	UINT32
RW_AtmHist_PCE_Delay	UINT32

KNOBS PARAMETER FILE

Variable Name	Data Type
Version_Knobs (64 characters max)	String
Version_Date (64 characters max)	String
Track	UINT32
Tm_Strong(x,y),x=0,1; y=0,1,2,3	UINT32
Tm_Weak(x,y),x=0,1; y=0,1,2,3	UINT32
Tm_NoMFSig_Strong(x,y), x=0,1; y=0,1,2,3	UINT32
TM_NoMFSig_Weak(x,y), x=0,1; y=0,1,2,3	UINT32
Tm_NoSFSig_Strong(x,y), x=0,1; y=0,1,2,3	UINT32
TM_NoSFSig_Weak(x,y), x=0,1; y=0,1,2,3	UINT32
Cloud_Alg_Strong(x,y), x=0,1; y=0,1,2,3	UINT32
Cloud_Alg_Weak(x,y), x=0,1; y=0,1,2,3	UINT32
Tm_NoSig_Strong(x,y), x=0,1; y=0,1,2,3	UINT32
Tm_NoSig_Weak(x,y), x=0,1; y=0,1,2,3	UINT32
DRMDEM_MFSig_Strong(x,y), x=0,1; y=0,1,2,3	UINT32
DRMDEM_MFSig_Weak(x,y), x=0,1; y=0,1,2,3	UINT32
DRMDEM_SFSig_Strong(x,y), x=0,1; y=0,1,2,3	UINT32
DRMDEM_SFSig_Weak(x,y), x=0,1; y=0,1,2,3	UINT32
Day_Night_Threshold_Strong(x), x=0,1,2,3	DOUBLE
Day_Night_Threshold_Weak(x), x=0,1,2,3	DOUBLE
SWChDisable_Water_Day_Strong	UINT32
SWChDisable_Water_Night_Strong	UINT32
SWChDisable_Land_Day_Strong	UINT32
SWChDisable_Land_Night_Strong	UINT32
SWChDisable_SeaIce_Day_Strong	UINT32
SWChDisable_SeaIce_Night_Strong	UINT32
SWChDisable_Ice_Day_Strong	UINT32
SWChDisable_Ice_Night_Strong	UINT32
SWChDisable_Water_Day_Weak	UINT32
SWChDisable_Water_Night_Weak	UINT32
SWChDisable_Land_Day_Weak	UINT32
SWChDisable_Land_Night_Weak	UINT32
SWChDisable_SeaIce_Day_Weak	UINT32
SWChDisable_SeaIce_Night_Weak	UINT32
SWChDisable_Ice_Day_Weak	UINT32
SWChDisable_Ice_Night_Weak	UINT32
Nosig_Relief1_Strong(x,y), x=0,1; y=0,1,2,3	UINT32

Variable Name	Data Type
Nosig_Relief1_Weak(x,y), x=0,1 y=0,1,2,3	UINT32
Nosig_Relief2_Strong(x,y), x=0,1; y=0,1,2,3	UINT32
Nosig_Relief2_Weak(x,y), x=0,1 y=0,1,2,3	UINT32
Nosig_Timer1_Strong(x,y), x=0,1; y=0,1,2,3	INT32
Nosig_Timer1_Weak(x,y), x=0,1; y=0,1,2,3	INT32
Nosig_Timer2_Strong(x,y), x=0,1; y=0,1,2,3	INT32
Nosig_Timer2_Weak(x,y), x=0,1; y=0,1,2,3	INT32
Nosig_Scale1_Strong(x,y), x=0,1; y=0,1,2,3	DOUBLE
Nosig_Scale1_Weak(x,y), x=0,1; y=0,1,2,3	DOUBLE
Nosig_Scale2_Strong(x,y), x=0,1; y=0,1,2,3	DOUBLE
Nosig_Scale2_Weak(x,y), x=0,1; y=0,1,2,3	DOUBLE
NoSig_DEM_Fraction_Strong(x), x=0,1,2,3	DOUBLE
NoSig_DEM_Fraction_Weak(x), x=0,1,2,3	DOUBLE
TEP_NOT_Strong	UINT32
TEP_NOT_Weak	UINT32
Tertiary_signal_flag_strong(x), x=0,1,2,3	UINT32
Tertiary_signal_flag_weak(x), x=0,1,2,3	UINT32
TEPflag_Strong(x), x=0,1,2,3	UINT32
TEPflag_Weak(x), x=0,1,2,3	UINT32

POSITION, POINTING AND RANGE (PPR) PARAMETER FILE

Variable Name	Data Type
Version_PPR (64 characters max)	String
Version_Date (64 characters max)	String
Track	UINT32
Clock_Cycles_in_ns	DOUBLE
DEM_Delta_Limit_strong	UINT32
DEM_Delta_Limit_weak	UINT32
Range_Window_Width_Min_Strong(x,y),x=0,1; y=0,1,2,3	UINT32
Range_Window_Width_Min_Weak(x,y),x=0,1; y=0,1,2,3	UINT32
Range_Window_Width_Max_Strong(x,y),x=0,1; y=0,1,2,3	UINT32
Range_Window_Width_Max_Weak(x,y),x=0,1; y=0,1,2,3	UINT32
Atmos_Range_Window_Offset_Strong	INT32
Atmos_Range_Window_Offset_Weak	INT32
Range_Window_Offset_Strong(x),x=0,1,2,3	INT32
Range_Window_Offset_Weak(x),x=0,1,2,3	INT32
Range_Decrease_Limit_Strong	UINT32
Range_Decrease_Limit_Weak	UINT32
Range_Window_DEM_Margin_Strong(x),x=0,1,2,3	UINT32
Range_Window_DEM_Margin_Weak(x),x=0,1,2,3	UINT32
Atm14km10ns_Strong	UINT32
Atm14km10ns_Weak	UINT32
Velocity_Average_Points_Strong	UINT32
Velocity_Average_Points_Weak	UINT32
Range_Rate_Average_Points_Strong	UINT32
Range_Rate_Average_Points_Weak	UINT32
Beam_Pointing_Strong(x),x=1,2,3	DOUBLE
Beam_Pointing_Weak(x),x=1,2,3	DOUBLE
Cosine_Solar_Zenith_Angle_Cutoff_Strong	DOUBLE
Cosine_Solar_Zenith_Angle_Cutoff_Weak	DOUBLE
Max_Sep_WeakStrong	UINT32
RW_AltimHist_PCE_Delay_Strong	UINT32
RW_AltimHist_PCE_Delay_Weak	UINT32
RW_AtmHist_PCE_Delay	UINT32

Appendix H: Spacecraft Position and Pointing Messages

The spacecraft sends the Flight Software packets of information once a second. The information used by the Flight Science Receiver Algorithms is contained in three messages: (1) Time of Day, (2) Position & Velocity, and (3) Attitude & Rate.

The function "float" below is intended to represent the conversion of an integer to double precision floating point.

<u>Time</u>

Conversion of the filtered spacecraft time from the spacecraft message is the same for all three types of spacecraft messages. This is true for both TP and TQ given in section 4.3. T represents the elapsed seconds since 6-January 1980 at 00:00:00 UTC.

Call the integer seconds iT. Call the fractions of seconds mT. Then:

$$T = \text{float}(iT) + \text{float}(mT)/10^7.$$

<u>Position & Velocity</u>

Call the 32 bit integer representation of the X-position "XI" in the spacecraft message iX. Then XI in meters is:

$$XI = \text{float}(iX) \cdot 0.01$$

Use the same conversion for "YI" and "ZI". Call the 32 bit integer representing the X-velocity "XIdot" in the spacecraft message iVx. Then XIdot in meters/seconds is:

$$XIdot = \text{float}(iVx) \cdot 0.0001$$

Use the same conversion for "YIdot" and "ZIdot".

<u>Attitude</u>

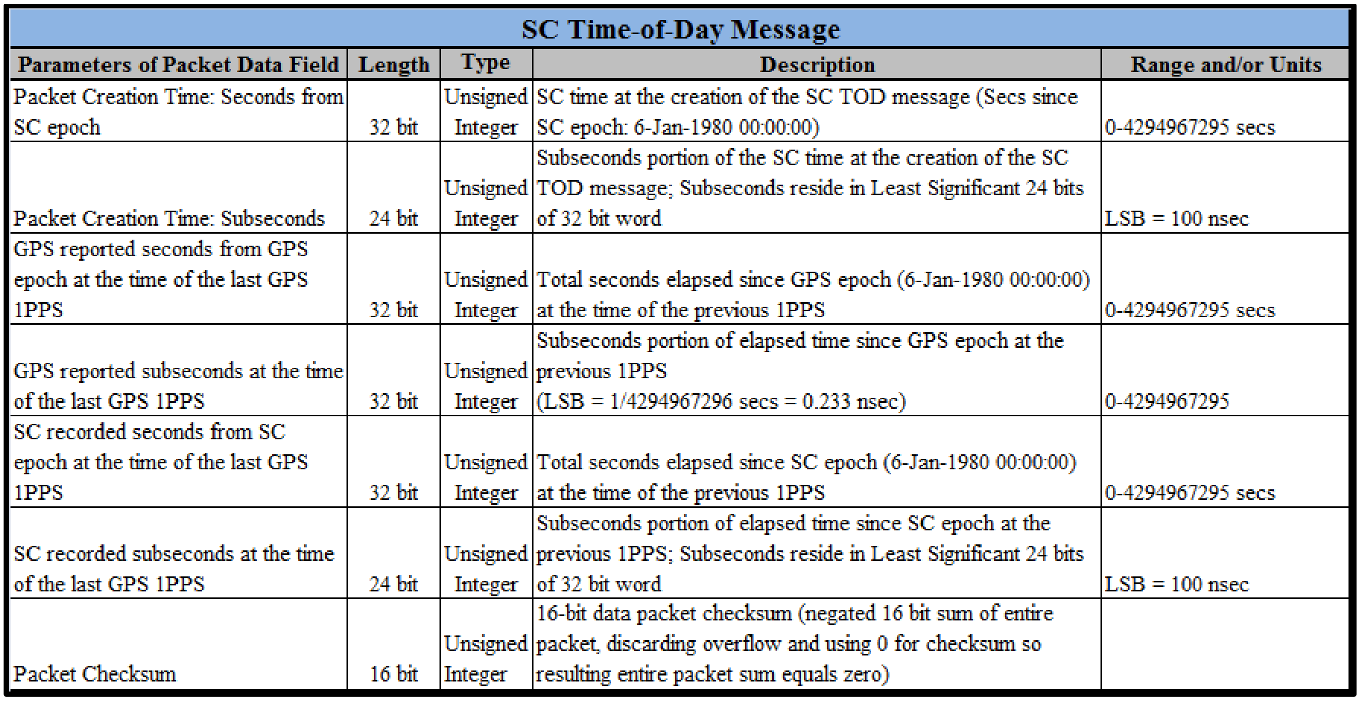
The conversion for the SC Inertial to LRS quaterion "i" (where i=1,2,3, or 4) is:

$$q(i) = \text{float}(iQ) \cdot 0.000000001$$

Where iQ is the 32 bit integer from the spacecraft attitude message.

The following tables (H.1, H.2 and H.3) were from Orbital ATK (now Northrop Grumman Corporation).

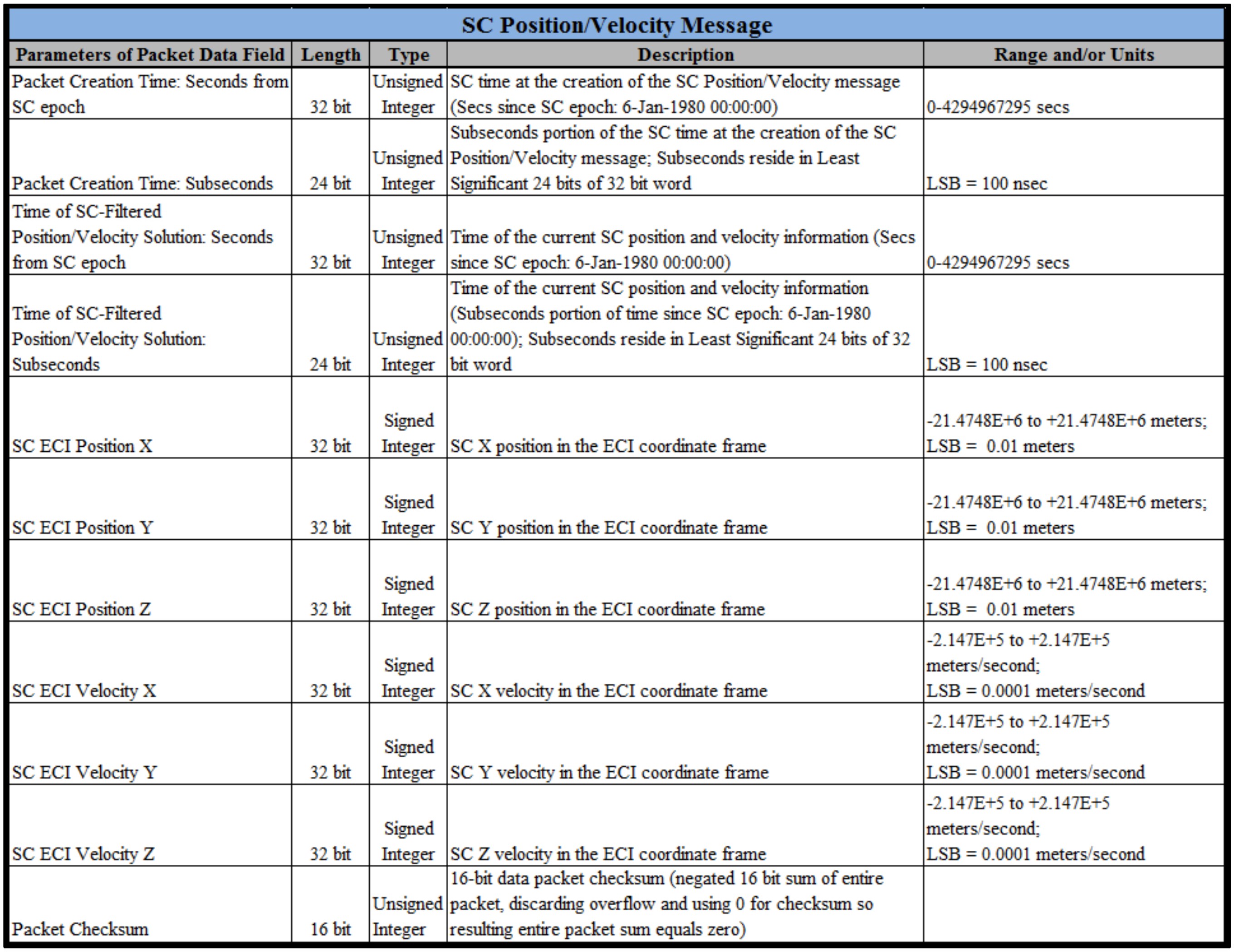
H.1 Spacecraft Time of Day Message



SC Time-of-Day Message

Parameters of Packet Data Field	Length	Type	Description	Range and/or Units
Packet Creation Time: Seconds from SC epoch	32 bit	Unsigned Integer	SC time at the creation of the SC TOD message (Secs since SC epoch: 6-Jan-1980 00:00:00)	0-4294967295 secs
Packet Creation Time: Subseconds	24 bit	Unsigned Integer	Subseconds portion of the SC time at the creation of the SC TOD message; Subseconds reside in Least Significant 24 bits of 32 bit word	LSB = 100 nsec
GPS reported seconds from GPS epoch at the time of the last GPS 1PPS	32 bit	Unsigned Integer	Total seconds elapsed since GPS epoch (6-Jan-1980 00:00:00) at the time of the previous 1PPS	0-4294967295 secs
GPS reported subseconds at the time of the last GPS 1PPS	32 bit	Unsigned Integer	Subseconds portion of elapsed time since GPS epoch at the previous 1PPS (LSB = 1/4294967296 secs = 0.233 nsec)	0-4294967295
SC recorded seconds from SC epoch at the time of the last GPS 1PPS	32 bit	Unsigned Integer	Total seconds elapsed since SC epoch (6-Jan-1980 00:00:00) at the time of the previous 1PPS	0-4294967295 secs
SC recorded subseconds at the time of the last GPS 1PPS	24 bit	Unsigned Integer	Subseconds portion of elapsed time since SC epoch at the previous 1PPS; Subseconds reside in Least Significant 24 bits of 32 bit word	LSB = 100 nsec
Packet Checksum	16 bit	Unsigned Integer	16-bit data packet checksum (negated 16 bit sum of entire packet, discarding overflow and using 0 for checksum so resulting entire packet sum equals zero)	

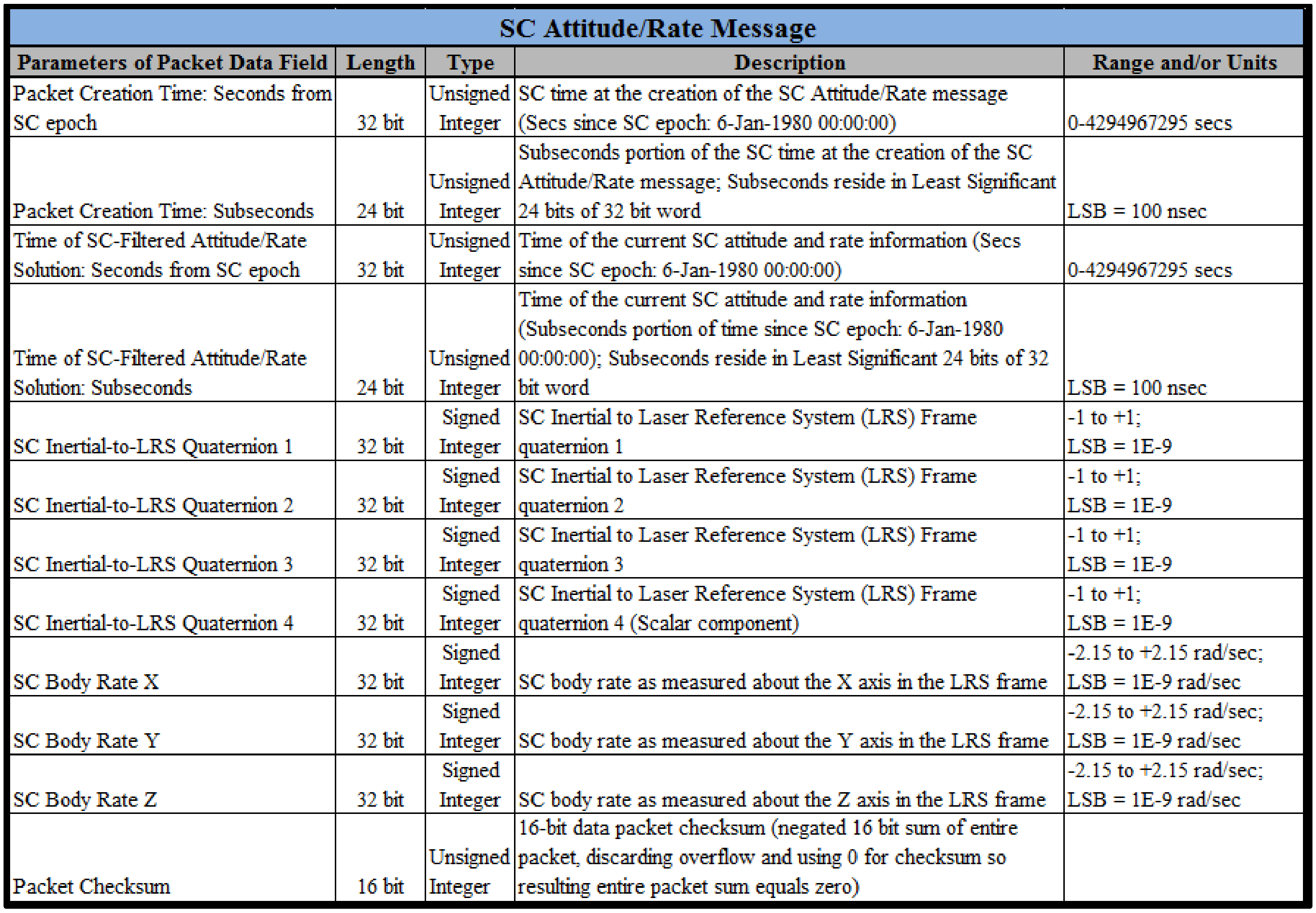
H.2 Spacecraft Position & Velocity Message



SC Position/Velocity Message

Parameters of Packet Data Field	Length	Type	Description	Range and/or Units
Packet Creation Time: Seconds from SC epoch	32 bit	Unsigned Integer	SC time at the creation of the SC Position/Velocity message (Secs since SC epoch: 6-Jan-1980 00:00:00)	0-4294967295 secs
Packet Creation Time: Subseconds	24 bit	Unsigned Integer	Subseconds portion of the SC time at the creation of the SC Position/Velocity message; Subseconds reside in Least Significant 24 bits of 32 bit word	LSB = 100 nsec
Time of SC-Filtered Position/Velocity Solution: Seconds from SC epoch	32 bit	Unsigned Integer	Time of the current SC position and velocity information (Secs since SC epoch: 6-Jan-1980 00:00:00)	0-4294967295 secs
Time of SC-Filtered Position/Velocity Solution: Subseconds	24 bit	Unsigned Integer	Time of the current SC position and velocity information (Subseconds portion of time since SC epoch: 6-Jan-1980 00:00:00); Subseconds reside in Least Significant 24 bits of 32 bit word	LSB = 100 nsec
SC ECI Position X	32 bit	Signed Integer	SC X position in the ECI coordinate frame	-21.4748E+6 to +21.4748E+6 meters; LSB = 0.01 meters
SC ECI Position Y	32 bit	Signed Integer	SC Y position in the ECI coordinate frame	-21.4748E+6 to +21.4748E+6 meters; LSB = 0.01 meters
SC ECI Position Z	32 bit	Signed Integer	SC Z position in the ECI coordinate frame	-21.4748E+6 to +21.4748E+6 meters; LSB = 0.01 meters
SC ECI Velocity X	32 bit	Signed Integer	SC X velocity in the ECI coordinate frame	-2.147E+5 to +2.147E+5 meters/second; LSB = 0.0001 meters/second
SC ECI Velocity Y	32 bit	Signed Integer	SC Y velocity in the ECI coordinate frame	-2.147E+5 to +2.147E+5 meters/second; LSB = 0.0001 meters/second
SC ECI Velocity Z	32 bit	Signed Integer	SC Z velocity in the ECI coordinate frame	-2.147E+5 to +2.147E+5 meters/second; LSB = 0.0001 meters/second
Packet Checksum	16 bit	Unsigned Integer	16-bit data packet checksum (negated 16 bit sum of entire packet, discarding overflow and using 0 for checksum so resulting entire packet sum equals zero)	

H.3 Spacecraft Attitude & Rate Message

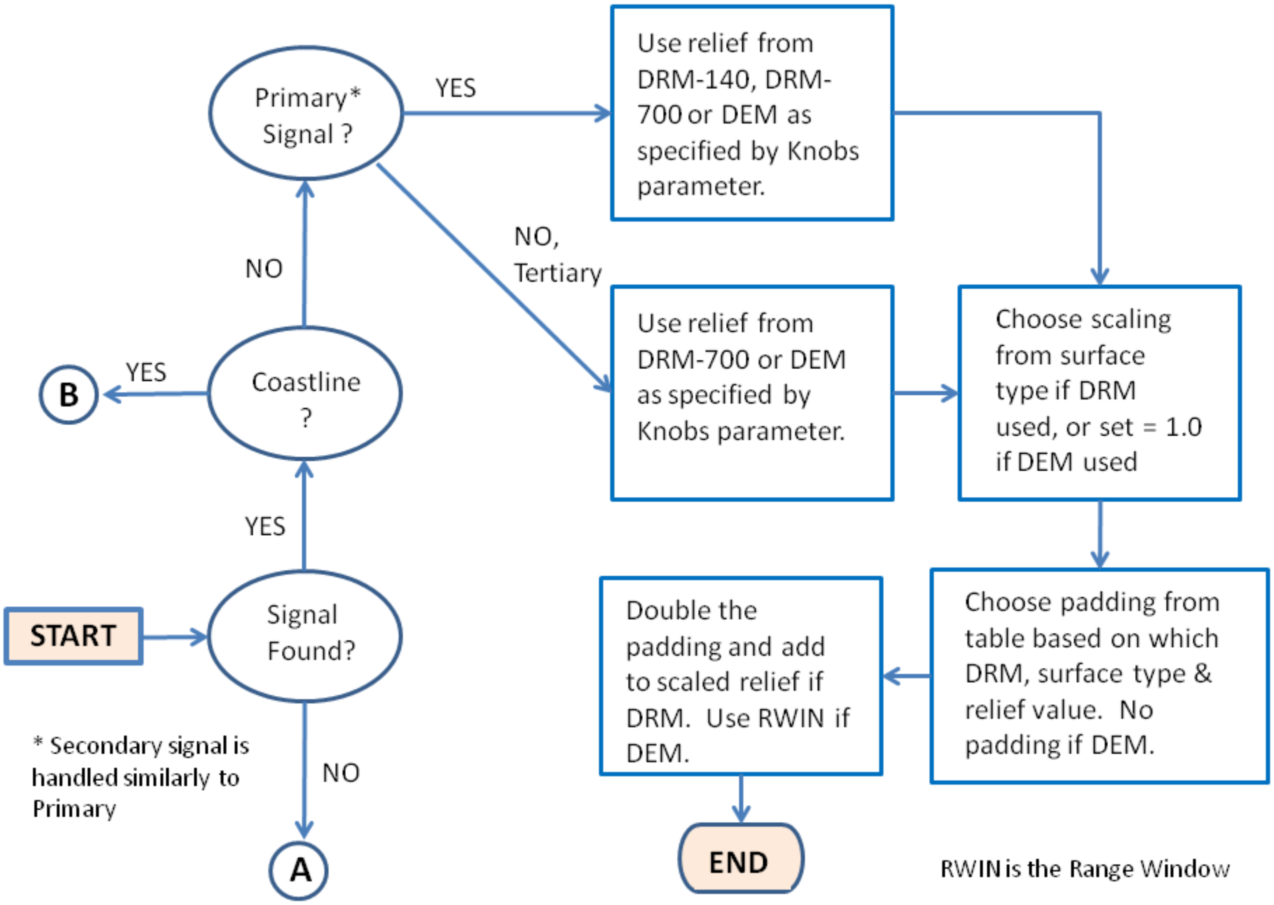


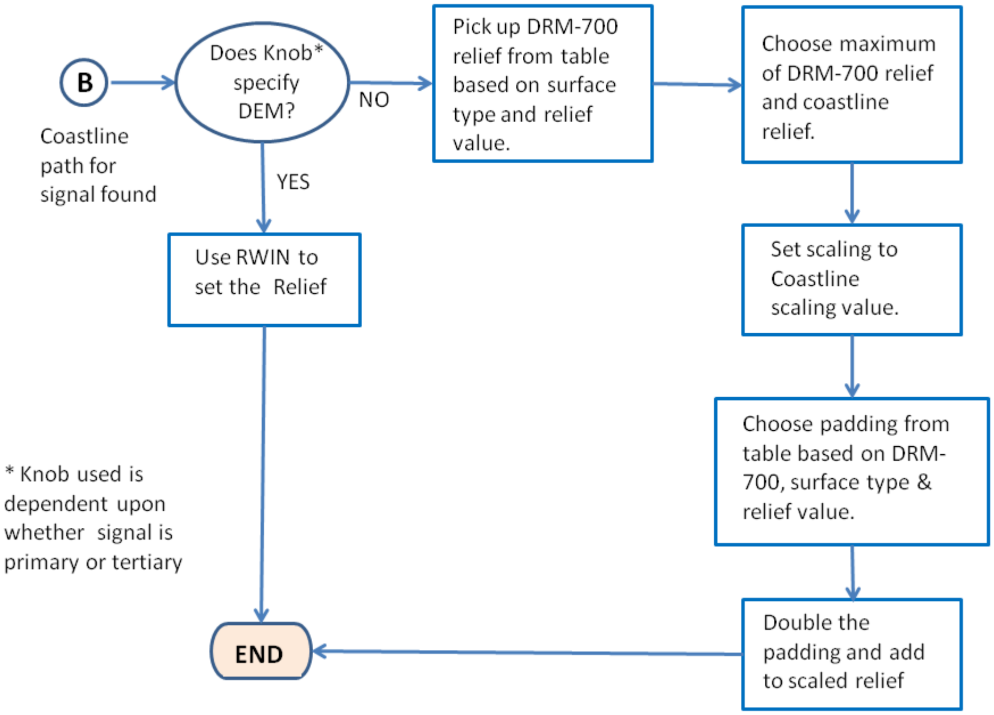
SC Attitude/Rate Message

Parameters of Packet Data Field	Length	Type	Description	Range and/or Units
Packet Creation Time: Seconds from SC epoch	32 bit	Unsigned Integer	SC time at the creation of the SC Attitude/Rate message (Secs since SC epoch: 6-Jan-1980 00:00:00)	0-4294967295 secs
Packet Creation Time: Subseconds	24 bit	Unsigned Integer	Subseconds portion of the SC time at the creation of the SC Attitude/Rate message; Subseconds reside in Least Significant 24 bits of 32 bit word	LSB = 100 nsec
Time of SC-Filtered Attitude/Rate Solution: Seconds from SC epoch	32 bit	Unsigned Integer	Time of the current SC attitude and rate information (Secs since SC epoch: 6-Jan-1980 00:00:00)	0-4294967295 secs
Time of SC-Filtered Attitude/Rate Solution: Subseconds	24 bit	Unsigned Integer	Time of the current SC attitude and rate information (Subseconds portion of time since SC epoch: 6-Jan-1980 00:00:00); Subseconds reside in Least Significant 24 bits of 32 bit word	LSB = 100 nsec
SC Inertial-to-LRS Quaternion 1	32 bit	Signed Integer	SC Inertial to Laser Reference System (LRS) Frame quaternion 1	-1 to +1; LSB = 1E-9
SC Inertial-to-LRS Quaternion 2	32 bit	Signed Integer	SC Inertial to Laser Reference System (LRS) Frame quaternion 2	-1 to +1; LSB = 1E-9
SC Inertial-to-LRS Quaternion 3	32 bit	Signed Integer	SC Inertial to Laser Reference System (LRS) Frame quaternion 3	-1 to +1; LSB = 1E-9
SC Inertial-to-LRS Quaternion 4	32 bit	Signed Integer	SC Inertial to Laser Reference System (LRS) Frame quaternion 4 (Scalar component)	-1 to +1; LSB = 1E-9
SC Body Rate X	32 bit	Signed Integer	SC body rate as measured about the X axis in the LRS frame	-2.15 to +2.15 rad/sec; LSB = 1E-9 rad/sec
SC Body Rate Y	32 bit	Signed Integer	SC body rate as measured about the Y axis in the LRS frame	-2.15 to +2.15 rad/sec; LSB = 1E-9 rad/sec
SC Body Rate Z	32 bit	Signed Integer	SC body rate as measured about the Z axis in the LRS frame	-2.15 to +2.15 rad/sec; LSB = 1E-9 rad/sec
Packet Checksum	16 bit	Unsigned Integer	16-bit data packet checksum (negated 16 bit sum of entire packet, discarding overflow and using 0 for checksum so resulting entire packet sum equals zero)	

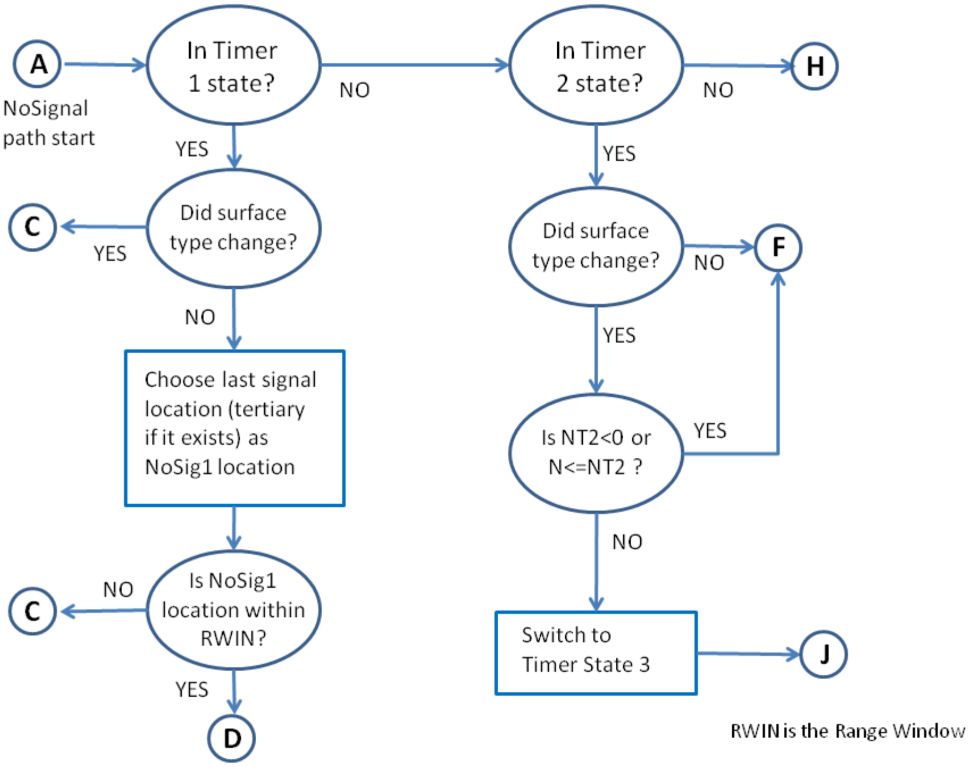
Appendix J: Telemetry Logic Flowcharts

The following pages define the Rx Algorithm's logical flow for the telemetry selection. These flowcharts support the description in section 7 of this document. If there is any disagreement between these flowcharts and the text in section 7, these flowcharts should take precedence.

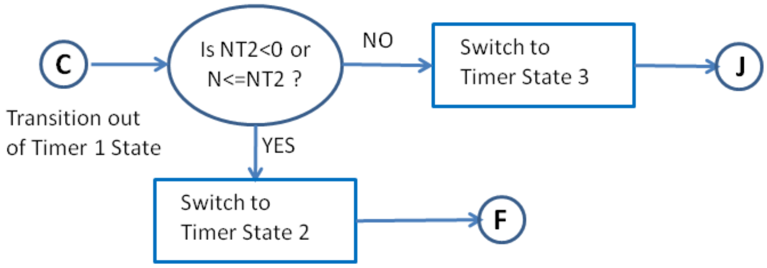

START
Signal Found?
YES
NO
A
Coastline ?
YES
B
NO
Primary* Signal ?
YES
NO, Tertiary
Use relief from DRM-140, DRM-700 or DEM as specified by Knobs parameter.
Use relief from DRM-700 or DEM as specified by Knobs parameter.
Choose scaling from surface type if DRM used, or set = 1.0 if DEM used
Choose padding from table based on which DRM, surface type & relief value. No padding if DEM.
Double the padding and add to scaled relief if DRM. Use RWIN if DEM.
END
* Secondary signal is handled similarly to Primary
RWIN is the Range Window

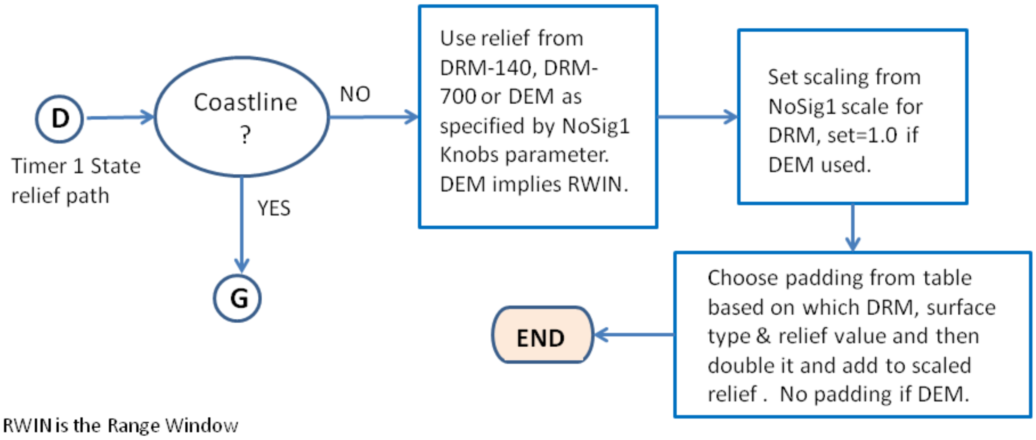

B
Coastline path for signal found
Does Knob* specify DEM?
NO
YES
Pick up DRM-700 relief from table based on surface type and relief value.
Choose maximum of DRM-700 relief and coastline relief.
Set scaling to Coastline scaling value.
Choose padding from table based on DRM-700, surface type & relief value.
Double the padding and add to scaled relief
Use RWIN to set the Relief
END

* Knob used is dependent upon whether signal is primary or tertiary

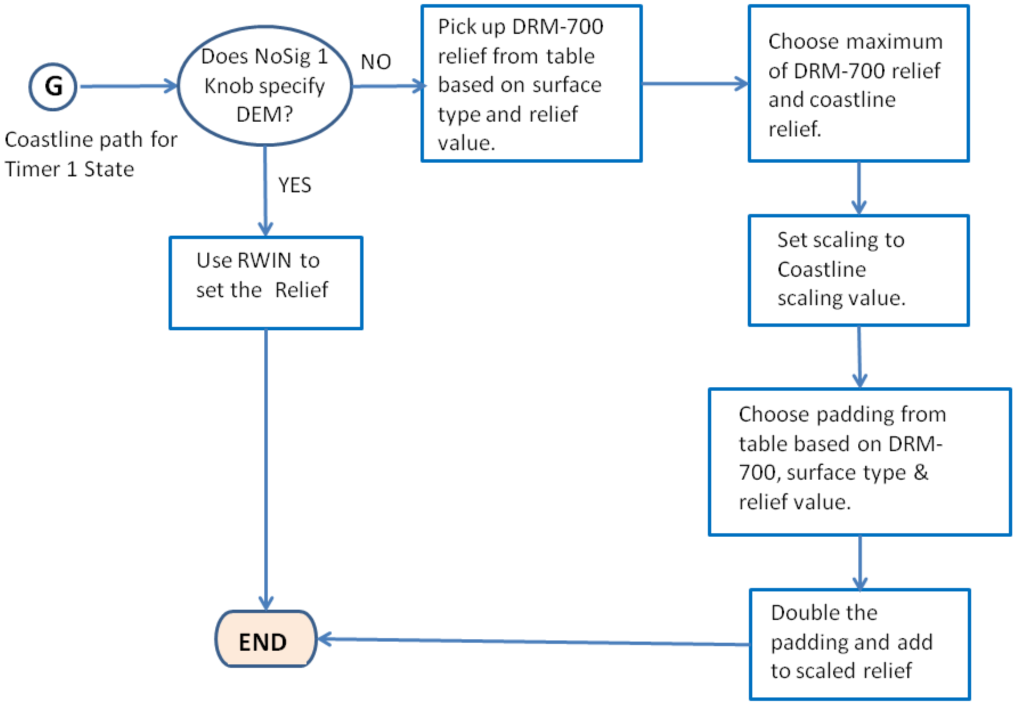


RWIN is the Range Window

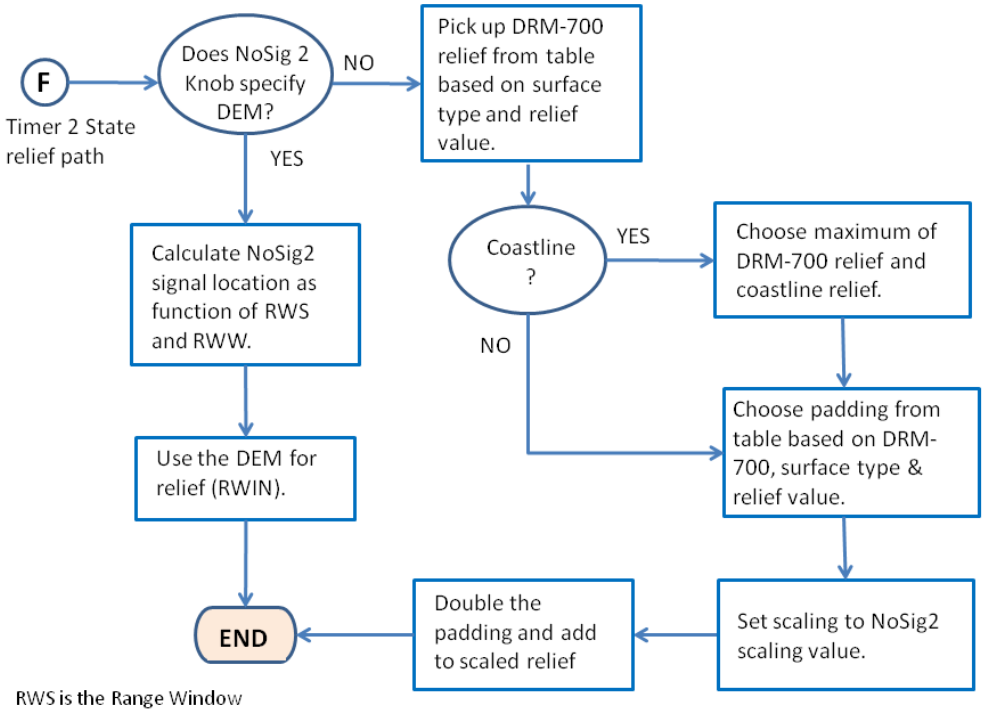



D
Timer 1 State relief path
Coastline ?
NO
Use relief from DRM-140, DRM-700 or DEM as specified by NoSig1 Knobs parameter. DEM implies RWIN.
Set scaling from NoSig1 scale for DRM, set=1.0 if DEM used.
YES
G
Choose padding from table based on which DRM, surface type & relief value and then double it and add to scaled relief . No padding if DEM.
END

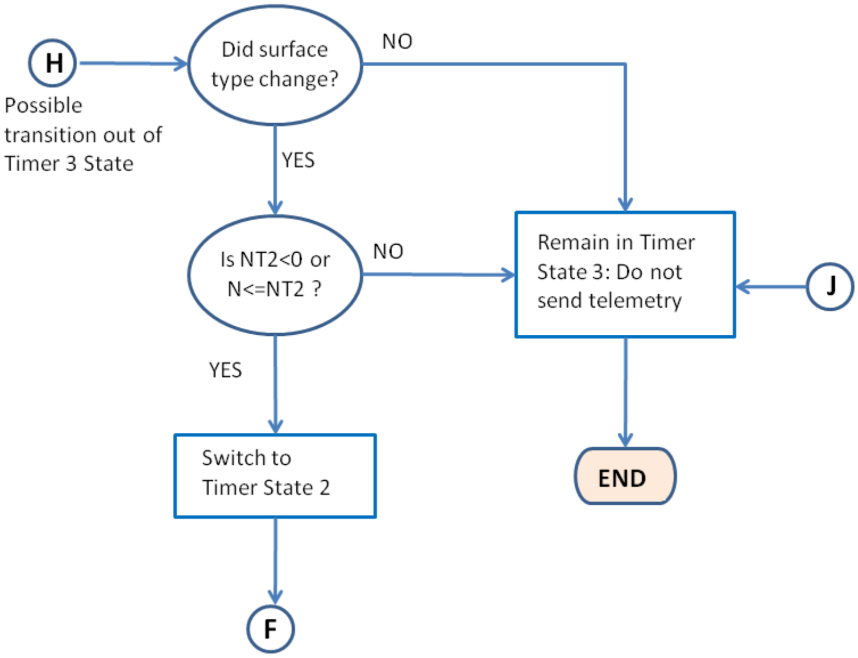
RWIN is the Range Window


G
Coastline path for Timer 1 State
Does NoSig 1 Knob specify DEM?
NO
Pick up DRM-700 relief from table based on surface type and relief value.
Choose maximum of DRM-700 relief and coastline relief.
YES
Use RWIN to set the Relief
Set scaling to Coastline scaling value.
Choose padding from table based on DRM-700, surface type & relief value.
Double the padding and add to scaled relief
END

RWIN is the Range Window


F
Timer 2 State relief path
Does NoSig 2 Knob specify DEM?
NO
Pick up DRM-700 relief from table based on surface type and relief value.
YES
Calculate NoSig2 signal location as function of RWS and RWW.
Coastline ?
YES
Choose maximum of DRM-700 relief and coastline relief.
NO
Choose padding from table based on DRM-700, surface type & relief value.
Use the DEM for relief (RWIN).
END
Double the padding and add to scaled relief
Set scaling to NoSig2 scaling value.

RWS is the Range Window Start and RWW is the Range Window Width.


H
Possible transition out of Timer 3 State
Did surface type change?
NO
YES
Is NT2<0 or N<=NT2 ?
NO
Remain in Timer State 3: Do not send telemetry
J
YES
Switch to Timer State 2
END
F

Glossary of Acronyms and Terms

Acronym or term	**Definition**
ACQ	Acquisition
AFSRA	ATLAS Flight Science Receiver Algorithms
ATLAS	Advanced Topographic Laser Altimetry System
BSM	Beam Steering Mechanism
DEM	Digital Elevation Model
DOE	Diffractive Optical Element, the device used to split the laser beam into six beamlets
DRM	Digital Relief Map
DRM-140/DRM-700	140-m and 700-m resolution DRMs
ECEF	Earth-Centered-Earth-Fixed coordinate system
ECEF_hat	Coordinate system aligned with the Earth-Centered-Earth-Fixed coordinate system but with origin at the orgin of the ATLAS coordinate system
ECI	Earth-Centered Inertial coordinate system
FPGA	Field Programmable Gate Array, a fast hardware processor
FSW	Flight SoftWare
FWHM	Full Width at Half Maximum
GLAS	Geoscience Laser Altimetry System
GMST	Greenwich Mean Siderial Time
GPS	Global Positioning System
ICESat	Ice, Cloud, and land SATellite
ICESat-2	ICESat follow-on mission
JD	Julian Day
LOLA	Lunar Orbiter Laser Altimeter
LRS	Laser Reference System
MEB	Main Electronics Box
MF	Major Frame
MLA	Mercury Laser Altimeter
MOLA	Mars Orbiter Laser Altimeter
MRF	ATLAS Master Reference Frame (also known as ATLAS_CSYS)
PCE	Photon Counting Electronics board, where some of the onboard processing will be done
POD	Precision Orbit Determination
PPR	Position, Pointing and Range parameter file
PRF	Pulse Repetition Frequency
Rx	Receiver
SBC	Single Board Computer
SF	Super Frame
SGT	Stinger Gaffarian Technologies, Incorporated
SRM	Surface Reference Mask
SSAI	Science, Systems, and Applications, Incorporated
ST	Signal/Telemetry parameter file
TBL	Telemetry Band Limit
TLB, TLMB	Telemetry Band
TM, TLM	Telemetry
UTC	Coordinated Universal Time
WGS-84	World Geodetic System 1984 ellipsoid