### TIRS Line-of-Sight Model Creation

#### Background/Introduction

The LOS model creation algorithm gathers the ancillary data and calibration parameters required to support geometric processing of the input TIRS image data set; validates the image time codes; extracts, validates, and preprocesses the TIRS SSM telemetry contained in the ancillary data stream; extracts the corresponding ephemeris and attitude data from the ancillary data stream; performs the necessary coordinate transformations; and stores the results in a geometric model structure for subsequent use by other geometric algorithms. The TIRS LOS model creation algorithm is derived from the OLI model creation algorithm. Its implementation will be very similar to the corresponding OLI application and will draw on the same spacecraft model, math, and utility libraries. Note that the ephemeris and attitude preprocessing logic common to both sensors is performed by the ancillary data preprocessing algorithm (6.1.4) to isolate the bulk of the geometric processing logic from the details of the incoming ancillary data stream. New attitude data processing logic has also been added to separate the high- and low-frequency attitude effects to allow the image resampling process to better correct for jitter at frequencies above the original 10 Hz algorithm design limit without requiring an unreasonably dense resampling grid.

#### Dependencies

The TIRS LOS Model Creation algorithm assumes that the Ancillary Data Preprocessing algorithm (6.1.4) has been executed to accomplish the following:

* Validated ephemeris data for the full imaging interval have been generated
* Validated attitude data for the full imaging interval have been generated
* The ancillary data have been scaled to engineering units

The Ancillary Data Preprocessing algorithm will generate preprocessed smoothed and cleaned ephemeris and attitude data streams. This provides a standard validated input for subsequent LOS model generation.

#### Inputs

The TIRS LOS Model Creation algorithm uses the inputs listed in the following table. Note that some of these “inputs” are implementation conveniences (e.g., using an ODL parameter file to convey the values of and pointers to the input data; including data set IDs to provide unique identifiers for data trending).

|  |
| --- |
| **Algorithm Inputs** |
| ODL File (implementation) |
| Acquisition Type (Earth, Lunar, Stellar) (optional, defaults to Earth) |
| CPF File Name |
| Preprocessed Ancillary Data Input File Name |
| L0R/L1R File Name |
| WRS Path/Row (stored in model and used for trending) |
| Trending On/Off Switch |
| Geometric work order common characterization ID (for trending) |
| Work Order ID (for trending) |
| Optional Precision Model Input Parameters (see note 6) |
| Input Precision Model Reference Time (optional) |
| Input Precision Ephemeris Correction Order (optional) |
| Input Precision X Correction Parameters (optional) |
| Input Precision Y Correction Parameters (optional) |
| Input Precision Z Correction Parameters (optional) |
| Input Precision Attitude Correction Order (optional) |
| Input Precision Roll Correction Parameters (optional) |
| Input Precision Pitch Correction Parameters (optional) |
| Input Precision Yaw Correction Parameters (optional) |
| CPF Contents |
| WGS84 Earth ellipsoid parameters |
| Earth orientation parameters (UT1UTC, pole wander, leap seconds) (see note 1) |
| Earth rotation velocity |
| Speed of light |
| TIRS to ACS reference alignment matrix/quaternion |
| Spacecraft center of mass (CM) to TIRS offset in ACS reference frame (new) |
| High frequency attitude data cutoff frequency (Hz) |
| Scene select mirror calibration parameters (new) (see Table 6‑28 below) |
| Focal plane model parameters (Legendre coefficients) |
| TIRS detector delay table (including whole pixel deselect offsets) (see note 10) |
| Nominal L0R fill (per band/SCA) |
| Nominal TIRS frame time nominal\_frame\_time (14.2857143 msec) |
| Nominal TIRS integration time |
| Image time code outlier thresholds delta\_time\_tolerance (DTIME\_TOL) and time\_outlier\_tolerance (OUTLIER\_TOL) (see note 3) |
| SSM encoder outlier threshold (see note 7) |
| Preprocessed Ancillary Data Contents |
| Attitude Data |
| Attitude data UTC epoch: Year, Day of Year, Seconds of Day |
| Time from epoch (one per sample, nominally 50 Hz) |
| ECI quaternion (vector: q1, q2, q3, scalar: q4) (one per sample) |
| ECEF quaternion (one per sample) |
| Body rate estimate (roll, pitch, yaw rate) (one per sample) |
| Roll, pitch, yaw estimate (one per sample) |
| Ephemeris Data |
| Ephemeris data UTC epoch: Year, Day of Year, Seconds of Day |
| Time from epoch (one per sample, nominally 1 Hz) |
| ECI position estimate (X, Y, Z) (one set per sample) |
| ECI velocity estimate (Vx, Vy, Vz) (one set per sample) |
| ECEF position estimate (X, Y, Z) (one set per sample) |
| ECEF velocity estimate (Vx, Vy, Vz) (one set per sample) |
| L0R/L1R Data Contents |
| Image Time Codes (one per line) |
| Integration Time |
| Scene Select Mirror Telemetry Packets (from Ancillary Data, see Table 6‑28 below) |
| TIRS Ancillary Data Time Code (one per 1 Hz frame) |
| Mirror Encoder Readout (24 bits, in counts) (20 samples per 1 Hz frame) (see note 9) |
| Detector Alignment Fill Table (see note 2) |

#### Outputs

|  |
| --- |
| TIRS LOS Model (additional detail is provided in Table 6‑28 below) |
| WGS84 Earth ellipsoid parameters |
| Earth Orientation Parameters (for current day) from CPF |
| Earth rotation velocity |
| Speed of light |
| TIRS to ACS reference alignment matrix/quaternion |
| Spacecraft center of mass to TIRS offset in ACS reference frame |
| SSM model parameters (Telescope alignment matrix and preprocessed SSM angles) |
| Focal plane model parameters (Legendre coefs) |
| Detector delay table (including whole pixel deselect offsets) |
| Nominal detector alignment fill table (from CPF) |
| L0R detector alignment fill table (from L0R) |
| ECI J2000 spacecraft ephemeris model (original and corrected) |
| ECEF spacecraft ephemeris model (original and corrected) |
| Spacecraft attitude model (time, roll, pitch, yaw) (orig and corr) (see note 4) |
| High frequency attitude perturbations (roll, pitch, yaw) per image line (jitter table) |
| Image time codes (see note 5) (in seconds) |
| Integration Time (in seconds) |
| Sample Time (in seconds) |
| WRS Path/Row |
| Model Trending Data |
| WRS Path/Row |
| Acquisition Date/Time |
| Geometric work order common characterization ID |
| Work Order ID |
| Image start UTC time (year, day of year, seconds of day) |
| Computed image frame time (in seconds) |
| Number of image lines |
| Number of out of limit image time codes |
| Number of out of limit SSM time codes |
| Number of out of limit SSM encoder measurements |

#### Options

Trending On/Off Switch

Optional precision model input parameters can be used to force model corrections.

#### Prototype Code

Input to the executable is an ODL file; output is a HDF4 formatted TIRS model file.

The prototype code was compiled with the following options when creating the test data files:

-g -Wall –O2 -march=nocona -m32 –mfpmath=sse –msse2

The following text is a brief description of the main set of modules used within the prototype with each module listed along with a very short description. It should be noted that not all library modules are referenced in the explanations below. The modules within the main create directory of the prototype are discussed and any library modules that were determined to be important to the explanation of either results, input parameters, or output parameters. Note that the modules in the main “create” directory are the same as the corresponding OLI routines. The TIRS-specific differences reside in the library routines.

model\_create – Main procedure that retrieves the input parameters and invokes the model generation and model output logic.

getpar – Retrieves the user-provided ODL parameters.

oli\_zero\_model – Library routine that initializes the model structure. Adapted from the corresponding OLI routine. Note that since many of the geometric model routines involve substantial reuse of heritage OLI code, many have not been renamed. This made it possible to leave the model creation driver routines unchanged.

get\_path\_row\_l0ra - Designed to retrieve the WRS path and row numbers from the L0R data. In the baseline algorithm these are ODL input parameters but they should ultimately be extracted from the Level 0R data directly. This unit is a placeholder for the time being.

oli\_run\_model – Library routine that loads the CPF, L0R, and preprocessed ancillary data into the model structure.

oli\_get\_cpf – Library routine that reads the CPF. Largely reuse from the corresponding OLI CPF input routine with new TIRS SCENE\_SELECT\_MIRROR group added.

tirs\_get\_scene\_select\_mirror – Library routine that reads the new TIRS SCENE\_SELECT\_MIRROR calibration parameter group from the CPF.

oli\_get\_model\_sensor\_params – Library routine that loads the sensor section of the model structure using data from the CPF and the L0R frame header. Adapted from corresponding OLI routine.

oli\_get\_model\_image\_params – Library routine that loads the image section of the model structure using data from the CPF, the L0R line headers, and the L0R detector offset fields. This unit also validates the image line time codes. Adapted from corresponding OLI routine.

tirs\_get\_ssm\_from\_l0r – Library routine that reads and preprocesses the TIRS SSM telemetry extracted from the L0R data, and loads it into the TIRS geometric model.

tirs\_align\_ssm\_data – Library routine that aligns the TIRS SSM encoder telemetry samples with the 1 Hz ancillary data frames, so that there are 20 complete samples per frame.

tirs\_check\_ssm\_data – Library routine that quality checks the TIRS SSM telemetry time codes and encoder angle data.

tirs\_smooth\_ssm\_data – Library routine that applies a moving window smoother to the TIRS SSM encoder data.

oli\_get\_model\_earth\_params – Library routine that loads the Earth model parameters from the CPF. Reuse of OLI routine.

oli\_get\_ancillary\_pre – Library routine that loads the attitude and ephemeris model sections using data from the preprocessed ancillary data file. Reuse of OLI routine.

oli\_build\_jitter\_table – Library routine that splits the attitude data from the ancillary data into low- and high-frequency streams, interpolates the high frequency data to match the TIRS line times, stores this per image line high frequency attitude data in the jitter table structure, and replaces the original combined attitude data stream with the low-frequency stream.

remez – Library routine that uses the Remez exchange algorithm to synthesize the weights (taps) of a low pass finite impulse response digital filter based on input filter size and cutoff frequency parameters. GNU Public License code written by Jake Janovetz, formerly of UIUC, which is available online at his site: http://www.janovetz.com/jake/

and more specifically:

http://www.janovetz.com/jake/remez/remez-19980711.zip

l8\_correct\_attitude – Library routine that applies the user-input precision model attitude corrections (if any).

l8\_convert\_ephem – Library routine that applies the user-input precision model ephemeris corrections (if any).

oli\_put\_model – Library routine that writes the TIRS model structure to the output HDF model file. Adapted from corresponding OLI routine.

#### Procedure

The TIRS LOS model is stored as a structure and is created from information contained in the Level 0R or Level 1R image data, the CPF, and the Ancillary data. The model is subsequently used along with the CPF to create a resampling grid. Data present in the model structure includes satellite position, velocity, and attitude, LOS angles, timing references, scene select mirror position, precision correction information (if any), and the software version. The TIRS LOS model is also used in several characterization and calibration routines for mapping input line/sample locations to geographic latitude/longitude.

The TIRS LOS model may be thought of in two parts, an instrument model that provides a line-of-sight vector for each TIRS detector (and, hence, each image line/sample), and a spacecraft model that provides spacecraft ephemeris (position and velocity) and attitude as a function of time. These models are linked by the image time stamps that allow each Level 0R or Level 1R image sample to be associated with a time of observation. The spacecraft portion of the model is common to the OLI LOS model.

*Instrument Model*

Figure 6‑59 shows the arrangement of the bands and SCAs on the TIRS focal plane. The model treats every band of every SCA independently. This is done by defining a set of 3rd order Legendre polynomials (see maturity note #2) for each band of each SCA. Unlike the OLI, the TIRS detectors are arranged in a two-dimensional array with two rows of that array being downlinked for each spectral band. One of the downlinked rows is primary and the other, redundant row is only used to replace bad pixels in the primary row. The TIRS LOS Legendre polynomials represent a theoretical “nominal” set of detectors that are best-fit to the primary row of detectors. This approach treats any replaced detectors as though they were aligned with the primary detectors for purposes of sensor LOS generation. This is a simplification of the OLI approach, which also must account for even/odd detector stagger. In the TIRS case, this stagger is effectively set to zero. This approach explicitly models any offsets caused by detector replacement, and the subpixel deviations of each detector from its nominal (Legendre best fit) location, for correction during image resampling. This leads to three detector types: nominal, actual, and exact. A nominal detector is calculated from the Legendre polynomials. An actual detector corrects the nominal detector location for the nominal (whole pixel) pixel select offsets. Like the OLI, since individual detectors may be deselected/replaced, these offsets are detector dependent. An exact detector has the actual correction applied but also includes the specific individual (subpixel) detector offsets. The Legendre polynomials and a table of detector offset values are stored in the CPF.

There is a slight angular difference between the line of sight vectors or angles associated with the primary and any replaced detectors. If the nominal LOS, generated using the 3rd order Legendre model, is nominal, the look angles for the actual and exact detectors are as follows:

x\_actual = x\_nominal + round(detector\_shift\_x) \* IFOV

y\_actual = y\_nominal + round(detector\_shift\_y) \* IFOV

x\_exact = x\_nominal + detector\_shift\_x \* IFOV

y\_exact = y\_nominal + detector\_shift\_y \* IFOV

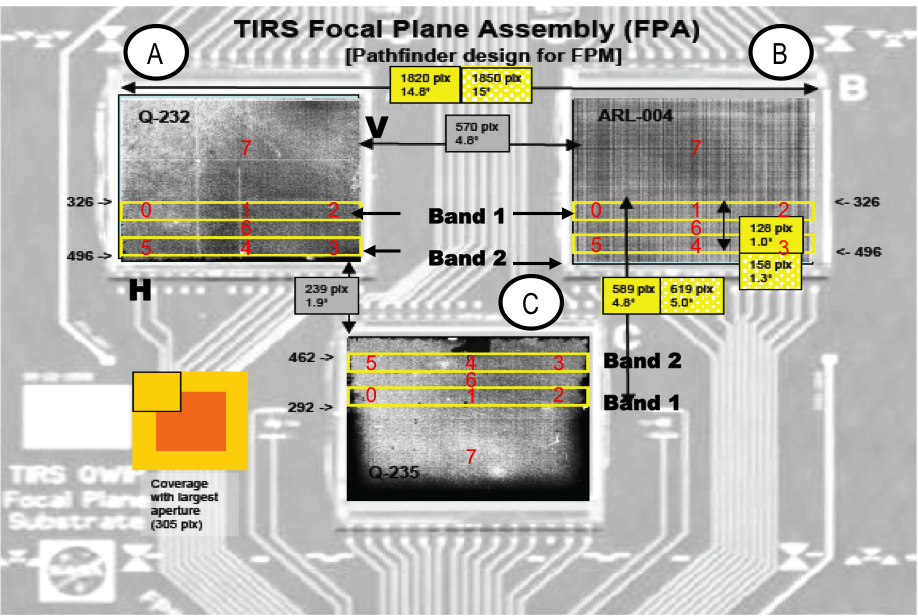


Figure 6‑59. TIRS Focal Plane Layout

The detector\_shift\_x and detector\_shift\_y values are the detector-specific offsets stored in the CPF detector delay tables. These offsets include both the whole-pixel deselect/replacement offsets and the fractional-pixel detector placement effects, and must be rounded to extract the integer portion. Note that the integer portion of the detector\_shift\_y value is always zero since the deselect effects are applicable only in the X direction. Also note that the integer portion of the detector\_shift\_x values will also all be zero in the event that no primary detectors are bad.

The nominal LOS is used in most line-of-sight projection applications. The actual LOS is used in conjunction with the actual image time (see below) to model the errors introduced by trading time (sample delay) for space (detector offset) for purposes of correcting the nominal LOS model. The exact LOS is generally used only for data simulation and other analytical purposes rather than in the geometric correction model, as the subpixel portion of the detector delay is applied directly in the image resampler rather than being included in the LOS model.

*Sample Timing*

The TIRS provides a time stamp with each image line collected. These time stamps make it possible to relate the image samples (pixels) to the corresponding spacecraft ephemeris and attitude data. The TIRS time stamps are contained in a line header that precedes each image line. The TIRS line header contents are shown in Figure 6‑60. Several items in this figure are worthy of particular note. First, the time stamp associated with a data frame is recorded at the beginning of the detector integration period. Second, the line header includes the integration time(TIRS does not use a separate frame header) and identifies the detector rows selected for downlink for each band and each SCA. This includes the dark band, which is not included in the TIRS geometric model. The line header fields other than the time code should be static within an imaging interval.



Figure 6‑60. TIRS Line Header Contents

Note that having the time code define the start of detector integration is different than the OLI where the time code represents the end of integration. This has the effect of making the integration time correction a positive adjustment to the pixel time for TIRS rather than a negative adjustment, as is the case for the OLI.

Also note that the TIRS line header is composed of 12-bit data words, like the other TIRS “pixels,” where the most significant 4 bits are all zero. The assumption here is that Level 0 processing will treat the line header words the same way that it handles TIRS pixel data and repackage the 12-bit fields into 16-bit data words. If instead, Level 0 processing strips off the extra 4 bits from the line header fields, then the line header preprocessing step mentioned below, will not be necessary.

One further complication to the problem of assigning times to image samples is the fact that the Level 0R/1R input imagery may include fill pixels inserted to achieve nominal primary and replaced detector alignment. This fill insertion allows the geometrically unprocessed 0R/1R imagery to be viewed as a spatially contiguous image without detector misalignments. The amount of detector alignment fill present will be indicated in the L0R/L1R image data (this is the purpose of the detector alignment fill table input noted above) so that the association of image samples with their corresponding time stamps can be adjusted accordingly. The assumption here is that, like OLI data, image fill will not be used to achieve nominal SCA or band alignment for TIRS data.

Due to the potential for deselected/replaced detectors, the nominal and actual times associated with a given pixel may not be the same. The actual time reflects the time that the current detector was actually sampled whereas the nominal time reflects the time at which the idealized detector represented by the TIRS LOS model would have been sampled.

If the current position within the image is given as a line and sample location, the two different “types” of times for TIRS pixels are calculated by:

l0r\_fill\_pixels = round(detector\_shift\_x) + nominal\_fill\_pixels

time\_index = line\_number - l0r\_fill\_pixels

if ( time\_index < 0 ) time\_index = 0

if (time\_index > (num\_time\_stamps - 1)) time\_index = num\_time\_stamps - 1

actual\_time = line\_time\_stamp[time\_index] + integration\_time/2

+ (line\_number - l0r\_fill\_pixels - time\_index) \* TIRS\_sample\_time

nominal\_time = actual\_time + round(detector\_shift\_x) \* TIRS\_sample\_time

where:

* line\_number is the zero-referenced TIRS image line number (N).
* nominal\_fill\_pixels is the amount of detector alignment fill to be inserted at the beginning of pixel columns that correspond to nominal detectors; that is, those detectors with a delay value of zero that are the basis for the Legendre polynomial LOS model. This value comes from the CPF and will be zero if there are no bad detectors to replace.
* l0r\_fill\_pixels is the total amount of detector alignment fill to be inserted at the beginning of the pixel column associated with the current detector. It includes both the nominal\_fill\_pixels and the detector-specific delay fill required to align deselected/replaced detectors.
* num\_time\_stamps is the total number of time codes (image lines) in the image. It is tested to ensure that time\_index, the line\_time\_stamp index, does not go out of bounds.
* detector\_shift\_x is the amount of detector offset for the current detector from the TIRS LOS model detector delay table. It is rounded to the nearest integer pixel because time offsets can only occur in whole line increments.

The detector\_shift\_x parameter is the detector-specific along-track offset as recorded in the CPF and subsequently stored in the LOS model detector delay table. It is rounded to the nearest integer so as to include the effects of even/odd detector stagger and detector deselect, but not the detector-specific subpixel offsets. The L0R/L1R data can be accessed by SCA making the association of sample number with detector index more straightforward. Note that, like OLI, the TIRS Level 0R data organizes the image samples from all 3 TIRS SCAs so that the samples are numbered left-to-right for all SCAs. This convention is also followed in the CPF detector offset tables. There are 640 samples per SCA for each spectral band.

Note that when fill is used to align replaced detectors the spatial difference between the nominal and actual look vectors is approximately compensated by the time difference between tnominal and tactual.

*TIRS Scene Select Mirror Model*

The TIRS SSM redirects the lines-of-sight from the TIRS telescope, which is oriented with its optical axis nominally in the +X direction, toward either: 1) the nadir Earth view; or 2) the space view port; or 3) the internal black body. It is the Earth view case that is of interest to the geometric processing models. Figure 6‑61 shows the SSM and TIRS telescope in relationship to the TIRS coordinate system, in which the telescope optical axis defines the +X axis. The TIRS coordinate system is nominally aligned with the spacecraft coordinate system (+X toward the direction of flight, +Z toward nadir, +Y completing a right-handed coordinate system).

The SSM angle, , is defined as the angle between the SSM normal and the SSM axis of rotation. This angle is nominally /4 radians (45 degrees). The SSM angle is a parameter of the SSM system that would be stored in the CPF.

Define a scene select mirror coordinate system, nominally parallel to the TIRS coordinate system, with the +X axis parallel to the SSM axis of rotation (**X**ssm), the +Y axis in the direction of the cross product of the mirror normal vector **nssm** and **X**ssm, and the +Z axis completing a right handed coordinate system.

The mirror axis of rotation and mirror normal are as follows:

**X**ssm = **n**ssm =

To include the effect of SSM rotation about its axis, rotate **n**ssm about **X**ssm by an angle ( – 0) where 0 is the SSM encoder angle at the nominal nadir pointing angle and  is the actual SSM encoder angle reported in the TIRS ancillary data. The mirror normal as a function of  is:

**n**ssm() =

The nominal nadir pointing angle 0 would be a SSM calibration parameter stored in the CPF. The measured encoder angle as a function of time, t, would be reported in the 1 Hz TIRS ancillary data stream, with 20 samples provided in each 1 Hz TIRS ancillary data packet. Any time delay between the actual encoder sample time(s) and the corresponding ancillary data packet time code would have to be accounted for, so this time delay, t, is included as a parameter in the CPF.



Figure 6‑61. Scene Select Mirror Line-of-Sight Redirection

In the TIRS telescope coordinate system, the LOS emerging from the telescope, **l**tele, is:

**l**tele =

where: XT is the across-track LOS angle (from the Legendre polynomial model)

AT is the along-track LOS angle (from the Legendre polynomial model)

To account for misalignments between the SSM and the TIRS telescope, rotate **l**tele about the X axis by an angle r, about the Y axis by an angle p, and about the Z axis by an angle y:



Where:



The telescope misalignment angles, r, p, and y, are calibration parameters stored in the CPF.

The LOS vector is reflected off the SSM by multiplying it by the matrix **P**():

**P**() = **I** – 2 **n**SSM() **n**TSSM()

And the resulting reflected LOS is:

**l**TIRS() = **P**() **l**SSM = [**I** – 2 **n**SSM() **n**TSSM()] **M**(r,p,y) **l**tele

Define =  – 0, and  = /4 +, where  is the departure from the ideal mirror angle.

The corresponding reflection matrix, **P**, becomes:



For an ideal SSM,  = 0, so the ideal reflection matrix, **P**0(), becomes:

**P**0() =

Which for nadir pointing (=0) reduces to:

**P**0() =

For a perfectly aligned SSM, r = p = y = 0, so:

**l**TIRS(0) = **P**0() **l**tele =

Note that this matches the nadir-pointing LOS formulation used for the OLI. To minimize the differences between the standard nadir-pointing (OLI) LOS model and the SSM reflected (TIRS) LOS model, we formulate the SSM effect as a rotation applied to a nadir-pointing LOS vector as follows:

Noting that: **P**0() **P**0() = **I**

**l**TIRS() = [**I** – 2 **n**SSM() **n**TSSM()] **P**0() **P**0() **M**(r,p,y) **P**0() **P**0() **l**tele

**l**TIRS() = [**P**0(,) **P**0(0)] [**P**0() **M**(r,p,y) **P**0()] **l**TIRS(0)

The TIRS LOS, **l**TIRS(), can thus be written as the ideal nadir-pointed LOS , **l**TIRS(0), rotated by the telescope alignment matrix, **M’**(r,p,y) and reflected by the SSM matrix, **P’**(,):

**l**TIRS() = **P’**(,) **M’**(r,p,y)**l**TIRS(0)

where:





For an ideal SSM, the product of the reflection and alignment rotation matrices, **P’**0(,0) **M’**(0,0,0), reduces to:

**M**0() = **P’**0(,0) **M’**(0,0,0) = , and **M**0(0) = **I**

Noting that , , r, p, and y are all close to zero, it can be shown that the  mirror angle offset is approximately equivalent to a pitch misalignment of 2. Using this approximation we can write the TIRS LOS transformation equation as the product of a reflection matrix, that is a function of only the mirror rotation angle , and a static telescope alignment matrix:

**l**TIRS() ≈ **M**0() **M’**(r,p+2,y)**l**TIRS(0)

In practice, the SSM reflection matrix **M**0() will be close to **I** so it will be difficult to distinguish telescope misalignments, modeled by **M’**, from TIRS instrument misalignments, and corrections to any prelaunch telescope alignment parameters will be absorbed by the TIRS alignment angles estimated on-orbit. For this reason we do not anticipate performing on-orbit calibration for the SSM parameters. The TIRS SSM calibration parameters are summarized in Table 6‑28. Note that since the primary (side A) and redundant (side B) SSM encoders are not perfectly aligned, the encoder values that correspond to nadir pointing will not be exactly the same. Thus, the nominal nadir encoder angle, 0, will be equal to either A or B depending on the mirror side/encoder in use. The mirror side in use will be indicated in the TIRS ancillary data.

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Symbol** | **Nominal Value** | **Source** |
| Mirror Angle |  | /4 radians | Fixed constant |
| Mirror Angle Deviation |  | 0 | Prelaunch characterization |
| Nadir Pointing Angle / Encoder Origin (Side A) | A | 0 | Defined value – establishes reference for TIRS alignment calibration |
| Nadir Pointing Angle / Encoder Origin (Side B) | **B** | 0 | Defined value – establishes reference for TIRS alignment calibration |
| Telescope Roll Offset | r | 0 | Prelaunch characterization |
| Telescope Pitch Offset | p | 0 | Prelaunch characterization |
| Telescope Yaw Offset | y | 0 | Prelaunch characterization |
| Encoder Time Offset | t | 0 | Prelaunch characterization |

Table 6‑28. TIRS Scene Select Mirror Model Calibration Parameters

*Spacecraft Model*

The spacecraft ephemeris and attitude models are constructed from the input preprocessed ancillary data by extracting the ancillary data that span the current image. Both ECI and ECEF versions of the ephemeris data are retained in the model structure to avoid the need to repeatedly invoke the ECI/ECEF coordinate system conversion. The ALIAS heritage roll-pitch-yaw representation of the attitude model is retained in the model structure though a quaternion representation may be used in a future algorithm revision (see note 4).

**Prepare TIRS LOS Model Sub-Algorithm**

This function gathers the information from the preprocessed ancillary data and the Level 0R/1R TIRS image and ancillary data needed to process model data and run the TIRS LOS model. Though it has the same overall purpose and function as the heritage OLI oli\_run\_model unit, new logic is required to handle the TIRS SSM telemetry information. The spacecraft (preprocessed ancillary data) sections are the same as the OLI model.

The main steps are as follows:

1. Load the image time codes and convert to seconds since spacecraft epoch.
2. Determine the image time window.
3. Validate/smooth the image time codes.
4. Extract the integration time from the Level 0R/1R image line header data.
5. Extract and preprocess the SSM telemetry from the Level 0R ancillary data.
6. Extract the associated ephemeris and attitude data from the preprocessed ancillary data stream.
7. Preprocess the input attitude data into a low-frequency stream, used for basic geometric modeling, and a high-frequency stream, used as a fine correction in the image resampler. This preprocessing was added to improve the ability of the geometric correction system to compensate for jitter disturbance frequencies above 10 Hz.

The input preprocessed ancillary data are stored in an HDF file. The attitude and ephemeris ancillary data streams each have an epoch time identifying the UTC date/time reference. Within these data streams, each attitude or ephemeris observation in the HDF file has a corresponding time offset relative to the epoch. This incoming ancillary data stream spans the entire imaging interval containing the image data represented in the Level 0R/1R input data. In creating the model we identify and extract the ancillary data sequence required to process the current image data.

The input Level 0R/1R image data are also packaged in HDF files that include the image samples for each band and SCA and the time codes assigned to each image line by the TIRS instrument. As shown in Figure 6‑60 above and Figure 6‑62 below, these spacecraft time codes are provided by the TIRS in CCSDS T-Field format which includes days since epoch (16-bit integer), milliseconds of day (32-bit integer) and microseconds of millisecond (16-bit integer) fields:

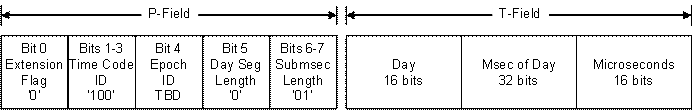


Figure 6‑62. TIRS Time Code Format

The baseline algorithm assumes that Level 0 processing will preserve these time codes in their original form, as days, milliseconds, and microseconds since the spacecraft epoch. Since they are derived from the spacecraft clock, the image time codes will be based on the same epoch used by the ancillary data (e.g., TAI seconds from J2000). Like all fields in the TIRS line header, the time codes are packaged in 12 bit fields with only the low order 8 bits containing valid data (see Figure 6‑62). Any initial line header preprocessing steps necessary to extract the 8 valid data bits from each line header data word are assumed to have been performed as part of Level 0Rp generation.

*Process Image Time Codes*

The image time codes are loaded from the input HDF Level 0R/1R data set. Deselected/replaced detector alignment fill will be inserted into the Level 0R/1R imagery as described above, if necessary, so the image lines each contain samples collected at times that may be offset from the time specified by the corresponding time code. The relationship between these time codes, the TIRS integration time, and the pixel center times has already been described above. The assumption here is that the L0Rp data will contain one time code per image line, excluding any fill lines, or a nominal 2071 time codes per scene. The image files themselves may be up to 30 lines longer to accommodate the redundant row deselect/replacement detector-alignment fill pixels. Simulated scenes may also be longer to provide the additional scene-to-scene overlap needed to support interval stitching.

1. Convert the time code to seconds from spacecraft epoch:

Line\_time = TC\_Day\*86400 + TC\_MSec/1000 + TC\_Micro/1e6

Note that an IEEE 754 double precision (64-bit) number with a 52-bit fraction should provide sufficient precision to represent time differences from 01JAN2000 to 01JAN2050 with microsecond accuracy (1.6e15 microseconds < 2^51).

1. Validate and correct the image time codes as follows:
   1. Loop through the time codes from 1 to N-1, where N is the number of image data frames/time codes, and test the difference between the current and previous time codes against the nominal frame time from the CPF using the #define tolerance DTIME\_TOL. The first of two consecutive time codes that are within the tolerance is the first valid time code. The DTIME\_TOL value is a constant in the prototype code but it would be better for it to be a CPF parameter.
   2. Initialize the TIRS clock model by setting the least squares variables to zero: A00 = A01 = A11 = L0 = L1 = 0
      1. Since the normal equation matrix, A, is symmetric, A10 = A01 so it is not computed separately.
      2. Add the first valid time code observation by adding 1 to A00. This is all that is required since, by definition, the index difference and time difference (see below) are zero at the first valid point.
   3. For each subsequent time code:
      1. Compare the time code difference to a larger outlier tolerance (OUTLIER\_TOL) chosen to bound the possible drift in the TIRS clock relative to the spacecraft clock (currently set to 50 microsec). The OUTLIER\_TOL value is a constant in the prototype code but it would be better for it to be a CPF parameter.
      2. If the time code difference is within the outlier range, add the current time to a least squares linear TIRS clock model:
         1. num = current index – first valid index
         2. time = current time – first valid time
         3. Accumulate:
            1. Valid point count: A00 += 1
            2. Index difference: A01 += num
            3. Squared index diff: A11 += num\*num
            4. Time difference: L0 += time
            5. Time diff\*index diff: L1 += num\*time
   4. Once all time codes have been analyzed, solve for the linear TIRS clock model parameters:
      1. determinant = A00\*A11 – A01\*A01
      2. If abs(determinant) <= 0.0 return an error
      3. Offset = first valid time + (A11\*L0 – A01\*L1) / determinant
      4. Rate = (A00\*L1 – A01\*L0) / determinant
   5. Use the correction model to replace bad time codes:
      1. For each time code:
         1. Calculate the corresponding model time as:

Mtime = Offset + (current index – first valid index) \* Rate

* + - 1. Calculate the actual time – model time difference.

Diff = abs( time code – Mtime )

* + - 1. Test the difference against DTIME\_TOL
      2. If the difference exceeds DTIME\_TOL, replace the current time code with the model value, Mtime
  1. If no valid time codes were found, return an error.
  2. Calculate the average observed frame time, delta\_time, by subtracting the first valid/corrected time code from the last valid/corrected time code and dividing by the number of time codes minus one.
  3. Store delta\_time in the model.

1. Compute the image start time: image\_start = line\_time[0]
2. Ensure that the ancillary data ephemeris covers the image:
   1. Convert the ephemeris epoch to TAI seconds from spacecraft epoch:
      1. Load the leap second table from the CPF.
      2. Search the leap second table for the last entry that is not later than the ephemeris epoch.
      3. Add the total leap second count for that entry to the UTC date/time to yield TAI date/time.
      4. Subtract the spacecraft TAI epoch to compute ephem\_start in TAI seconds since the spacecraft epoch.
   2. Check the beginning of the ephemeris interval against the image start time (which is also TAI seconds since the spacecraft epoch):

If (image\_start – ephem\_start) < 4 seconds

Then report error “Ephemeris data does not cover the image” and exit

* 1. Check the end of the ephemeris interval against the image stop time:

ephem\_stop = ephem\_start + ephem\_time[M-1] where M is the number of ephemeris data entries.

image\_stop = image\_start + delta\_time\*(N-1)

If (ephem\_stop – image\_stop) < 4 seconds

Then report error “Ephemeris data does not cover the image” and exit

* 1. Note that the 4 second overlap threshold would be a good thing to put in a #define statement as suggested below.

1. Repeat step 4 using the ancillary attitude data in place of the ephemeris data.
2. Compute the image start UTC epoch by converting image\_start to UTC as described under “Convert Spacecraft Time Code to UTC” in the Ancillary Data Preprocessing ADD (6.1.4). This epoch will be stored as: Year, Day of Year, Seconds of Day.
3. Make sure the epoch is consistent with the ancillary data:
   1. If image\_year > ephem\_year or image\_day > ephem\_day

Then image\_year = ephem\_year

image\_day = ephem\_day

image\_seconds = image\_seconds + 86400

This ensures that all computations for a given imaging interval are based on the same day and, hence, on the same UT1UTC, pole wander, and leap second corrections.

1. Subtract the image start time from the line time codes so that the times are seconds from image start.
2. Store the image start UTC epoch (image\_year, image\_day, image\_seconds) and the image offset times in the model structure.
3. Report/trend the results of the time code processing including:
   1. WRS Path/Row (input parameters)
   2. Image UTC epoch (year, day, seconds of day)
   3. L0R ID (input parameter)
   4. Work order ID (input parameter)
   5. Computed frame time (delta\_time)
   6. Number of replaced time codes (bad\_image\_time\_count)

Extract the integration time field from the Level 0R/1R image line header data. There will be one value for each image line. Convert the integration time from the first valid TIRS line header record to units of seconds and store in the model structure.

*Extract and Process SSM Data*

To populate the SSM model portion of the TIRS geometric model it is necessary to construct two elements: the SSM alignment matrix **M’**(r,p+2,y) which is a function of the SSM alignment parameters from the CPF; and a sequence of SSM pointing angles and associated times. The SSM pointing angles are computed using the SSM telemetry data contained in the Level 0R ancillary data. The TIRS ancillary data group is provided every second. The relevant contents of the TIRS ancillary data group are shown in Table 6‑29. Each 1 Hz group contains a time code and twenty-one 24-bit resolution SSM encoder samples.

|  |  |  |
| --- | --- | --- |
| **Field** | **Size** | **Contents** |
| Day | 16 bits | Days since spacecraft epoch |
| Milliseconds | 32 bits | Milliseconds of day |
| Microseconds | 16 bits | Microseconds of millisecond |
| SSM Position 1-21 | 24 bits | 24-bit resolution SSM encoder readout |

Table 6‑29. TIRS Scene Select Mirror Ancillary Data

A twenty-first sample is included, even though the encoder is sampled at 20 Hz, to ensure that encoder samples do not accumulate in the ancillary data output buffer. This means that the twenty-first sample will likely contain zeros in both the high order and low order words. Due to variations in the encoder data generation and ancillary buffer output timing, the high- and low-order encoder data words can become misaligned, and one or both of the data words in sample #21 will be non-zero. This condition must be detected and corrected by the SSM data processing logic.

Retrieve the SSM alignment angle and mirror angle deviation parameters from the CPF and construct the SSM alignment matrix, **M’**, using the equations above. Store the resulting alignment matrix in the SSM model. Determine which mirror control electronics (MCE) side (A or B) is active by performing a majority vote on the MCE bits (bits 0 and 1) of the elec\_enabled\_flags status word. If the number of status words with bit 1 set exceeds the number with bit 0 set, make the SSM reference angle equal to the CPF side B mirror nadir angle. Otherwise, use the side A value.

Quality check the entire TIRS SSM telemetry set as follows:

1. Validate the SSM telemetry time codes:
   1. Find the first valid time code as the first time code for which the time difference between it and the following time code is the nominal 1.0 second sampling interval, to within a pre-defined tolerance.
   2. Use the valid time code to correct all previous time codes using the nominal sampling interval.
   3. Use the valid time code to check all subsequent time codes using the nominal sampling interval. Any time codes failing the sampling interval tolerance are corrected using the previous valid sample time and the nominal sampling interval.
2. Due to the asynchronous SSM telemetry generation and ancillary data assembly processes, the SSM encoder samples will sometimes be improperly aligned with the 1 Hz ancillary data frames. The encoder telemetry generation logic can run either faster or slower than nominal, leading to variations in the number of samples accumulated in the output buffer between 1 second buffer read operations. Due to these variations in the encoder sample timing, any given ancillary data frame may contain from 19 to 21 encoder samples. The TIRS ancillary data assembly logic was modified to read (and output) 21 samples for each frame to ensure that any extra encoder samples do not accumulate in the encoder telemetry buffer. Furthermore, the upper 16 bits and lower 16 bits of each encoder value are buffered separately, so a given frame may have partial encoder samples (only upper 16 or only lower 16). The following logic is designed to align the upper and lower data words of the SSM telemetry encoder samples in each ancillary data record:
   1. If encoder sample 20 is zero and encoder sample 21 is zero
      1. If this is the last ancillary record, set sample 20 equal to sample 19
      2. Otherwise set sample 20 equal to sample 1 from the next record.
   2. If the high 16-bits of sample 20 are zero:
      1. If the high 16-bits of sample 21 are non-zero, move the bits from sample 21 to sample 20.
      2. Otherwise, move the high 16-bits from the first sample in the next record to sample 20, and move all the high order words in the next record up one sample (setting the 21st sample to zero). If the current record is the last record, copy sample 19 into sample 20.
   3. If the low 16-bits of sample 20 are zero:
      1. If the low 16-bits of sample 21 are non-zero, move the bits from sample 21 to sample 20.
      2. Otherwise, move the low 16-bits from the first sample in the next record to sample 20, and move all the low order words in the next record up one sample. If the current record is the last record, copy sample 19 into sample 20.
   4. If sample 21 is equal to zero, go to the next record.
   5. If only the high 16-bits of sample 21 are zero, and this is not the last record, move all the high order words in the next record up one sample.
   6. If only the low 16-bits of sample 21 are zero, and this is not the last record, move all the low order words in the next record up one sample.
3. Extract the first 20 samples in each record for subsequent processing. Note that this method discards extra samples and fills missing samples by duplicating either the next or the previous sample.
4. Validate the 24-bit SSM telemetry encoder readings:
   1. Find the first valid encoder reading as the first reading for which the difference between it and the SSM reference angle (nominal nadir pointing angle for the current MCE side, determined above) is less than the quality tolerance specified in the CPF.
   2. Use the valid angle to replace all previous 24-bit encoder readings.
   3. Use the valid angle to check and, if necessary, replace all subsequent 24-bit encoder readings, by comparing each value to the previous, valid/corrected, value.
5. Although an anomaly with the SSM encoder that occasionally rendered the upper 14 bits of the read out invalid, was observed during subsystem level test, this behavior has not been seen in the integrated TIRS instrument. Due to the low probability of this being a problem, no special logic has been added to handle this case. Instead, standard outlier detection and correction logic will be relied upon to correct this problem if it occurs.

The quality-checked TIRS ancillary data groups are examined to find the range of samples that correspond to the TIRS image.

1. Scan through the TIRS ancillary data, and convert each time code to seconds from spacecraft epoch.
2. Find the last ancillary data record with a time code that is earlier than the TIRS image start time. This is the first TIRS ancillary data packet to extract.
3. Find the first ancillary data record with a time code that is later than the TIRS image stop time. This is the last TIRS ancillary data packet to extract.
4. Extract the SSM (mirror control electronics or MCE) telemetry fields and time codes for the ancillary data records covering the TIRS image.

For each extracted SSM telemetry group:

1. Convert the TIRS ancillary data time code to seconds from spacecraft epoch and find the difference between the ancillary data time and the TIRS image start time.
2. Add the SSM encoder time offset (from the CPF) to the sample time so that it represents the time of the first SSM encoder sample.
3. Compute the sample times for the remaining samples by adding increments of 0.05 seconds to the previous sample time.
4. Load the SSM encoder samples into signed 32-bit integer variables for subsequent manipulation.
5. Scale the encoder counts to radians and subtract the SSM reference angle (set as described above):
   1. Angle = 2\*pi\*Sample\_Value/0x01000000 – 0
   2. If Angle > pi Then Angle -= 2\*pi
   3. If Angle < -pi Then Angle += 2\*pi

Note: These angles should all be close to zero for Earth viewing.

1. Add the 20 computed times (from image start) and the 20 SSM angles to the SSM model.

Smooth the SSM angles as follows:

1. For each SSM angle:
   1. Find the previous two and next two original data points, or the closest four points if two cannot be found before and after.
   2. Compute the average of the current point and the four closest points.
   3. The smoothed value is the mean of the current point value and the 4 closest points.
2. Store the smoothed sequence of SSM angles in the SSM model.

The baseline TIRS geometric algorithms assume no significant temperature dependence in either the SSM or in TIRS alignment. We do assume that the temperature telemetry present in the TIRS ancillary data will be recorded in the trending database (for radiometric purposes) so that pointing temperature sensitivities could be studied on-orbit (see note #8).

*Extract Ancillary Ephemeris and Attitude Data*

The subset of ancillary ephemeris and attitude data needed to span the image data are extracted from the input preprocessed ancillary data stream and stored in the model structure. Extra ancillary data, nominally 4 seconds, is required before and after the image start/stop times to ensure model continuity from scene to scene within an imaging interval. This ancillary data overlap time parameter could be stored in a #define statement as it would not be expected to change once established.

The ephemeris data extraction/subsetting procedure is as follows:

1. Compute the time offset from the ephemeris epoch time to the desired ephemeris start time for this image. Note that since the image epoch has been adjusted to fall in the same day as the ephemeris epoch this can be done using the seconds of day fields only.

ephem\_start = image\_seconds – ancillary\_overlap – ephem\_seconds

Noting that image\_seconds and ephem\_seconds are the seconds of day fields from the image and ephemeris epoch times, respectively.

1. Loop through the ephemeris sample times to find the last entry that does not exceed ephem\_start. This is the ephemeris start index (eph\_start\_index).
2. Compute the time offset from the ephemeris epoch time to the desired ephemeris stop time for this image.

ephem\_stop = image\_seconds + line\_time[N-1] + ancillary\_overlap – ephem\_seconds

N is the number of image lines, and N-1 is the index of the last image line time.

1. Loop through the ephemeris sample times to find the first entry that exceeds ephem\_stop. This is the ephemeris stop index (eph\_stop\_index).
2. Compute a new ephemeris UTC epoch for this image:

imgeph\_year = ephem\_year

imgeph\_day = ephem\_day

imgeph\_seconds = ephem\_seconds + ephem\_samp\_time[eph\_start\_index]

1. Load the ECI and ECEF ephemeris samples from eph\_start\_index to eph\_stop\_index (inclusive) into the model structure, adjusting the sample times so that they are offset from the UTC epoch computed in step 5.

The attitude data extraction/subsetting procedure is as follows:

1. Compute the time offset from the attitude epoch time to the desired attitude start time for this image. Note that since the image epoch has been adjusted to fall in the same day as the ancillary data (ephemeris and attitude) epochs this can be done using the seconds of day fields only.

att\_start = image\_seconds – ancillary\_overlap – att\_seconds

Noting that image\_seconds and att\_seconds are the seconds of day fields from the image and attitude epoch times, respectively.

1. Loop through the attitude sample times to find the last entry that does not exceed att\_start. This is the attitude start index (att\_start\_index).
2. Compute the time offset from the attitude epoch time to the desired attitude stop time for this image.

att\_stop = image\_seconds + line\_time[N-1] + ancillary\_overlap – att\_seconds

1. Loop through the attitude sample times to find the first entry that exceeds att\_stop. This is the attitude stop index (att\_stop\_index).
2. Compute a new attitude UTC epoch for this image:

imgatt\_year = att\_year

imgatt\_day = att\_day

imgatt\_seconds = att\_seconds + att\_samp\_time[att\_start\_index]

1. For Earth-view acquisitions, load the roll-pitch-yaw samples from att\_start\_index to att\_stop\_index (inclusive) into the model structure, adjusting the sample times so that they are offset from the UTC epoch computed in step 5.
2. For lunar/stellar acquisitions, convert the ECI quaternion samples from att\_start\_index to att\_stop\_index (inclusive) to ECI roll-pitch-yaw values, as described below, and store the computed roll-pitch-yaw values in the model structure, adjusting the sample times so that they are offset from the UTC epoch computed in step 5.

*Converting ECI Quaternions to Roll-Pitch-Yaw*

For lunar and stellar acquisitions, the ECI attitude representation is stored in the model structure. In the baseline model, this is done by converting the ECI quaternions to roll-pitch-yaw values relative to the ECI axes. This is one of the motivations for considering a transition to using a quaternion attitude representation in the model in the future.

The ECI quaternions are converted to roll-pitch-yaw values as follows:

1. Compute the rotation matrix corresponding to the ECI quaternion values:

**M**ACS2ECI =



1. Compute the corresponding ACS to ECI roll-pitch-yaw values:



Note that in implementing these calculations it is important to use the ATAN2 rather than the ATAN arctangent implementation in order to retain the correct quadrants for the Euler angles. This is not a concern in Earth-view imagery where the angles are always small, but becomes an issue for these lunar/stellar ACS to ECI angles.

1. Store the ECI roll-pitch-yaw values in the model attitude data table.

At the completion of this sub-algorithm the model structure contains the image frame time stamps, the multispectral and panchromatic sample and integration times, the ancillary ephemeris data, in both ECI and ECEF representations, covering the image, and the ancillary attitude data covering the image.

*Jitter Correction Data Preprocessing*

Jitter correction preprocessing operates on the roll-pitch-yaw attitude data stream extracted from the spacecraft ancillary data to separate the low frequency spacecraft pointing effects from the higher frequency jitter disturbances. The low frequency pointing model is used for line-of-sight projection and other geolocation processing while the high frequency jitter effects are applied as per-line corrections during image resampling. To implement this frequency separation in the line-of-sight model the original attitude sequence is passed through a low pass filter with a cutoff frequency defined as a parameter in the CPF. This cutoff frequency will nominally be in the 1 Hz to 10 Hz range. The value ultimately selected for this cutoff frequency will depend upon the actual disturbance profile observed in the spacecraft attitude data. The high frequency data stream should be limited in magnitude to subpixel (ideally sub-half-pixel) effects, but the lower the cutoff frequency can be, the sparser (and smaller) the TIRS resampling grid can be made in the line (time) dimension.

The low pass filtered version of the attitude sequence is differenced with the original data to construct the complementary high pass data sequence. The high pass sequence is then interpolated at the TIRS image line times to provide a table containing high frequency roll-pitch-yaw corrections for each image line. This jitter table is stored in the TIRS line-of-sight model. The original attitude sequence in the line-of-sight model is replaced with the low pass filtered sequence to avoid double counting the high frequency effects. This process is depicted in Figure 6‑63.



Figure 6‑63. Jitter Correction Table Generation Data Flow

The jitter table construction processing sequence is as follows:

1. Extract a copy of the original attitude data sequence from the TIRS line-of-sight model.
2. Retrieve the low pass filter cutoff frequency from the CPF.
3. Design a low pass filter with the desired cutoff frequency and apply it to the attitude data.
   1. Use the cutoff frequency and attitude data sampling time to compute the size of the desired filter as follows:
      1. Compute the normalized cutoff frequency (the ratio of the cutoff frequency to the attitude data sampling frequency):

n\_cutoff = cutoff\_frequency / attitude\_sample\_frequency

Note that this is the same as:

n\_cutoff = cutoff\_frequency \* attitude\_sample\_time

* + 1. Compute the number of samples per cycle at the cutoff frequency:

Nsamp = 1 / n\_cutoff

* + 1. Multiply the number of samples per cycle by 3 and add 1 to yield the desired filter size:

FSize = 3\*Nsamp + 1

* + 1. If this results in an even filter size, add one:

If ( FSize modulo 2 == 0 ) FSize = FSize + 1

* 1. Use the Remez exchange algorithm to design the filter and generate the filter weights. The standard Parks-McClellan finite impulse response (FIR) digital filter design method uses the Remez exchange algorithm (ref. Theory and Application of Digital Signal Processing, Rabiner and Gold, Prentice-Hall, 1975). A C implementation of this algorithm called remez.c, authored by Jake Janovetz at the University of Illinois, is available under the GNU Public License. This implementation specifies the desired (low pass, in this case) filter response using the following parameters:
     1. Filter size (number of taps) – FSize computed in item a. above.
     2. Number of frequency bands to use – 2, one pass band (low frequency) and one stop band (high frequency).
     3. Band frequency bounds – 0 to the normalized cutoff frequency (n\_cutoff) for the pass band and 1.5\*n\_cutoff to 0.5 (normalized Nyquist frequency) for the stop band.
     4. Desired band gains – 1 for pass band (low) and 0 for stop band (high).
     5. Band weights (how tightly to constrain the actual filter response to the design filter response in each band) – 1 for pass band and 10 for stop band.
     6. Filter type – BANDPASS (the low pass filter is a special case of the more general BANDPASS filter type supported by the remez algorithm.
  2. Make sure the synthesized filter is normalized (weights sum to 1) by adding the filter tap values and dividing each tap by the total.

sum = Σ h[i] where h[i] are the FSize filter taps.

h’[i] = h[i] / sum for i = 1 to FSize.

* 1. Convolve the filter with the roll-pitch-yaw attitude data one axis at a time:

half\_size = FSize / 2

for index = 0 to num\_rpy – 1

low\_roll[index] = low\_pitch[index] = low\_yaw[index] = 0

for ii = -half\_size to half\_size

if ( index + ii < 0 ) j = -index – ii

else if ( index + ii < num\_rpy ) j = index + ii

else j = 2\*num\_rpy – index - ii – 1

low\_roll[index] += roll[j]\*h[ii+half\_size]

low\_pitch[index] += pitch[j]\*h[ii+half\_size]

low\_yaw[index] += yaw[j]\*h[ii+half\_size]

1. Subtract the low pass filtered sequences from the original sequences to extract the high frequency portion of the data, and transfer any residual bias (non-zero mean value) from the imaging portion of the high frequency sequence to the low frequency sequence:

roll\_bias = pitch\_bias = yaw\_bias = 0

att\_pts = 0

for index = 0 to nrpy–1

high\_roll[index] = roll[index] – low\_roll[index]

high\_pitch[index] = pitch[index] – low\_pitch[index]

high\_yaw[index] = yaw[index] – low\_yaw[index]

if ( image\_start\_time < attitude\_time[index] < image\_stop\_time )

roll\_bias += high\_roll[index]

pitch\_bias += high\_pitch[index]

yaw\_bias += high\_yaw[index]

att\_pts += 1

roll\_bias = roll\_bias / att\_pts

pitch\_bias = pitch\_bias / att\_pts

yaw\_bias = yaw\_bias / att\_pts

for index = 0 to nrpy-1

high\_roll[index] -= roll\_bias

low\_roll[index] += roll\_bias

high\_pitch[index] -= pitch\_bias

low\_pitch[index] += pitch\_bias

high\_yaw[index] -= yaw\_bias

low\_yaw[index] += yaw\_bias

1. Interpolate the high frequency sequence values at the TIRS line sampling times to create the model jitter table:

For each TIRS image line = 0 to number of lines:

Compute the line sampling time as:

index = line

line\_time = line\_time\_stamp[index]

+ integration\_time/2

Convert to time from attitude epoch:

line\_time += image\_epoch – attitude \_epoch

Interpolate high frequency roll-pitch-yaw values at this time using four point Lagrange interpolation:

Compute starting index for interpolation:

index = floor(line\_time / attitude\_sample\_time) – 1

Compute the fractional sample offset to the line time:

w = line\_time / attitude\_sample\_time – index – 1

Compute the Lagrange weights:

w1 = -w \* (w – 1) \* (w – 2) / 6

w2 = (w + 1) \* (w – 1) \* (w – 2) / 2

w3 = -w \* (w + 1) \* (w – 2) / 2

w4 = (w + 1) \* w \* (w – 1) / 6

Interpolate:

roll = high\_roll[index]\*w1 + high\_roll[index+1]\*w2

+ high\_roll[index+2]\*w3 + high\_roll[index+3]\*w4

pitch = high\_pitch[index]\*w1 + high\_pitch[index+1]\*w2

+ high\_pitch[index+2]\*w3 + high\_pitch[index+3]\*w4

yaw = high\_yaw[index]\*w1 + high\_yaw[index+1]\*w2

+ high\_yaw[index+2]\*w3 + high\_yaw[index+3]\*w4

1. Replace the original model attitude data sequence with the low pass filtered attitude data sequence.

Note that if TIRS and OLI processing is combined, the attitude filtering and high-pass/low-pass separation logic should be common, but the two sensors would still require their own jitter tables since these tables are based on the image line times, which are different for TIRS and OLI.

**Process LOS Model Sub-Algorithm**

This function loads the LOS Legendre polynomial coefficients and other model components from the CPF, and performs additional processing on the attitude and ephemeris information in the LOS model structure. It invokes the following sub-algorithms.

**Read CPF Model Parameters Sub-Algorithm**

This function loads model components from the CPF. In the heritage ALIAS implementation some of these model components either did not exist (e.g., instrument offset from spacecraft center of mass) or were used for image resampling but not LOS model computations (e.g., detector offset table) and so, were not included in the model. These are included in the TIRS model to make it self-contained for purposes of line-of-sight computations.

CPF parameters loaded into the geometric model include the following:

1. Earth orientation parameters – the UT1UTC and pole wander (x,y) parameters for the current day are stored in the model to avoid the necessity of repeatedly looking them up in the CPF.
2. TIRS offset from spacecraft center of mass – a 3-vector that captures the small offset, in spacecraft body coordinates, between the TIRS instrument, where images are captured, and the spacecraft center of mass, the position of which is reported in the ancillary ephemeris data, making it possible to translate the ephemeris data to the TIRS. Technically, this would be the vector from the spacecraft center of mass to the center of the TIRS entrance pupil. Note that this formulation assumes that the spacecraft on-board GPS data processing includes the GPS to spacecraft center of mass (CM) offset and that the spacecraft is, in fact, reporting CM positions not GPS antenna positions. If the ephemeris represents the GPS antenna location then we would need to know the spacecraft CM to GPS antenna offset as well.
3. TIRS to ACS alignment matrix – a 3-by-3 matrix that captures the relative orientation of the TIRS coordinate system to the ACS coordinate system, making it possible to rotate the TIRS instrument-space line-of-sight vectors into the ACS reference system. In the heritage ALIAS system this was actually represented in the CPF by an ACS to instrument rotation matrix, which was inverted for each LOS model invocation. Whichever convention is used in the CPF, the LOS model should store the TIRS-to-ACS rotation matrix.
4. TIRS line-of-sight Legendre polynomials – a set of 8 coefficients (4 along-track and 4 across-track) for each band on each SCA. Each set of 4 forms a 3rd order Legendre polynomial that is used to evaluate a nominal LOS angle (along- or across-track) for the detectors in that band on that SCA. This differs from the heritage implementation, which used a 2nd order model (see the Read LOS Vectors Sub-Algorithm description below).
5. TIRS detector delay table – a table consisting of two values (along- and across-track) per detector reflecting the offset of each actual detector from its nominal location (as modeled by the 3rd order Legendre polynomials – see below). In the heritage ALIAS implementation these were small subpixel offsets that were applied in the image resampling procedure. With the TIRS, this table will also contain any offsets due to detector deselect/replacement (i.e., the operational use of a detector from one of the redundant rows). This table is needed in those LOS projection algorithms that utilize either actual (whole pixel offsets) or exact (full subpixel offsets) detector locations.

**Read LOS Vectors Sub-Algorithm**

This function retrieves the line of sight vectors from the CPF. The line of sight vectors are stored as sets of 3rd order Legendre polynomial coefficients. There is a unique set of 8 coefficients for each band of each SCA, 4 for the along-track polynomial and 4 for the across-track polynomial. These values are read from the CPF and stored in the LOS model. The polynomials are used to compute along- and across-track viewing angles for each nominal detector.

**Initialize the Precision Model Sub-Algorithm**

This function initializes the precision LOS correction model parameters. If the optional precision model input parameters are provided, those values are used. In the normal case, those parameters are absent and the correction model is initialized as follows:

Set the precision correction reference time to the center of the scene:

t\_ref = line\_time[N/2] where: N is the number of time codes in the image

Set the ephemeris correction model order to zero: eph\_order = 0

Set both ephemeris X correction parameters to zero:

x\_corr[0] = 0.0, x\_corr[1] = 0.0

Set both ephemeris Y correction parameters to zero:

y\_corr[0] = 0.0, y\_corr[1] = 0.0

Set both ephemeris Z correction parameters to zero:

z\_corr[0] = 0.0, z\_corr[1] = 0.0

Set the attitude correction model order to zero: att\_order = 0

Set all three attitude roll correction parameters to zero:

roll\_corr[0] = 0.0, roll\_corr[1] = 0.0, roll\_corr[2] = 0.0

Set all three attitude pitch correction parameters to zero:

pitch\_corr[0] = 0.0, pitch\_corr[1] = 0.0, pitch\_corr[2] = 0.0

Set all three attitude yaw correction parameters to zero:

yaw\_corr[0] = 0.0, yaw\_corr[1] = 0.0, yaw\_corr[2] = 0.0

Note that these parameters are used to compute the corrected ephemeris and attitude data sequences, which are also stored in the model. The parameters themselves are included in the model primarily to document the magnitude of the corrections applied and to facilitate more advanced uses of the model creation logic. For example, it is sometimes useful to be able to force a particular model bias (e.g., a roll angle) into a model that is to be used for data simulation (see note 6). Therefore, though not strictly necessary for operational data processing, these parameters aid in anomaly resolution, data simulation, and algorithm development. In normal operations, these initial correction parameters are all zero and the "corrected" attitude and ephemeris data sequences are identical to the "original" attitude and ephemeris data prior to the execution of the LOS model correction algorithm. Subsequent algorithms (e.g., LOS projection) operate on the corrected data.

**Correct Attitude Sub-Algorithm**

This function applies the ACS/body space attitude corrections computed by the LOS/precision correction procedure to the attitude data sequence. It outputs a parallel table of roll-pitch-yaw values with the precision corrections applied. In the model creation context the precision corrections are zero so the two sets of attitude data are identical. Though applying the precision corrections to construct the corrected attitude sequence could be said to be overkill for model creation (since the corrections are nominally zero at this point) this capability is required for LOS model correction and is used here to support the use of the model creation algorithm for data simulation and anomaly resolution as it makes it possible to force initial biases into the model. This sub-algorithm will also be used by the LOS/precision correction algorithm to create the precision model. Note that the formulation is somewhat different for Earth-view scenes (Acquisition Type = Earth) than it is for lunar and stellar observations.

*Earth Scenes*

For Earth-view scenes the sequence of transformations required to convert a line-of-sight in the TIRS instrument coordinate system, generated using the Legendre polynomials, is:

**x**ECEF = **M**ORB2ECEF **M**ACS2ORB **M**Precision **M**TIRS2ACS **M**SSM() **M**Tele2SSM **x**TIRS

where: **x**TIRS is the Legendre-derived instrument LOS vector

**M**Tele2SSMis the TIRS telescope alignment matrix described above

**M**SSM() is the SSM reflection matrix, described above, which is a function of SSM angle

**M**TIRS2ACS is the TIRS to ACS alignment matrix from the CPF

**M**Precision is the correction to the attitude data computed by the LOS/precision correction procedure

**M**ACS2ORB is the spacecraft attitude (roll-pitch-yaw)

**M**ORB2ECEF is the orbital to ECEF transformation computed using the ECEF ephemeris

**x**ECEF is the LOS vector in ECEF coordinates

Since TIRS will occasionally be viewing off-nadir and it is more natural to model attitude errors in the ACS/body coordinate system than in the orbital coordinate system, the order of the **M**ACS2ORB and **M**Precision rotations have been reversed for L8/9 as compared to the heritage Landsat/EO-1 implementation. The impact is minimal in the model and LOS projection but becomes more important for the LOS/precision correction algorithm.

This new sub-algorithm pre-computes the **M**ACS2ORB **M**Precision combination and stores the corresponding corrected roll-pitch-yaw attitude sequence in the model structure. This approach has several advantages:

1. It streamlines the application of the model for LOS projection by removing the step of explicitly applying the precision correction.
2. It allows for the use of a more complex correction model in the future since the application of the model is limited to this unit. Note that the Earth-view attitude correction model consists of the following model parameters:

Precision reference time: t\_ref in seconds from the image epoch (at the center of the image time window)

Attitude model order: att\_order = 1

Roll bias and rate corrections: roll\_corr[] = roll\_bias, roll\_rate

Pitch bias and rate corrections: pitch\_corr[] = pitch\_bias, pitch\_rate

Yaw bias and rate corrections: yaw\_corr[] = yaw\_bias, yaw\_rate

This model is dealt with in more detail in the line-of-sight correction algorithm description.

1. Retaining both the original and corrected attitude sequences in the model make the model self-contained and will make it unnecessary for the LOS/precision correction algorithm to access the preprocessed ancillary data.

The disadvantage is that it doubles the size of the attitude data in the model structure.

The construction of the corrected attitude sequence proceeds as follows:

For each point in the attitude sequence j = 0 to K-1:

1. Compute the rotation matrix corresponding to the jth roll-pitch-yaw values:

**M**ACS2ORB =



1. Compute the precision correction at the time (t\_att = att\_seconds + att\_time[j]) corresponding to the attitude sample:
2. 
3. 
4. 

Note that only the seconds of day fields are needed for the attitude and image epochs as they are constrained to be based on the same year and day.

1. Compute the rotation matrix corresponding to roll\_correction (r), pitch\_correction (p), and yaw\_correction (y) (**M**Precision) using the same equations presented in step 1 above.
2. Compute the composite rotation matrix: **M** = **M**ACS2ORB **M**Precision
3. Compute the composite roll-pitch-yaw values:



1. Store the composite roll’-pitch’-yaw’ values in the jth row of the corrected attitude data table.

*Lunar and Stellar Scenes*

Though there is no TIRS requirement for lunar or stellar data processing, this capability is retained to maintain compatibility with the OLI geometric model. For celestial (lunar or stellar) observations the sequence of transformations required to convert a line-of-sight in the TIRS instrument coordinate system, generated using the Legendre polynomials, is:

**x**ECI = **M**ACS2ECI **M**Precision **M**TIRS2ACS **M**SSM() **M**Tele2SSM **x**TIRS

where: **x**TIRS is the Legendre-derived instrument LOS vector

**M**Tele2SSMis the TIRS telescope alignment matrix described above

**M**SSM() is the SSM reflection matrix, described above, which is a function of SSM angle

**M**TIRS2ACS is the TIRS to ACS alignment matrix from the CPF

**M**Precision is the correction to the attitude data computed by the LOS/precision correction procedure

**M**ACS2ECI is the spacecraft attitude in the ECI frame derived from the ECI quaternions in the preprocessed ancillary data

**x**ECI is the LOS vector in ECI coordinates

The advantage of modeling the precision attitude corrections in ACS rather than orbital coordinates becomes apparent here, since the orbital frame is not used in the lunar case.

This sub-algorithm pre-computes the **M**ACS2ECI **M**Precision combination and stores the corresponding corrected attitude sequence (as roll-pitch-yaw values relative to ECI) in the model structure. Another difference between the Earth-view and lunar/stellar models is in the formulation of the precision model. The lunar attitude correction model adds an acceleration term to the Earth-view correction model parameters:

Precision reference time: t\_ref in seconds from the image epoch (nominally near the center of the image time window)

Attitude correction model order: att\_order = 2

Roll bias, rate, and acceleration corrections: roll\_corr[] = roll\_bias, roll\_rate, roll\_acceleration

Pitch bias, rate, and acceleration corrections: pitch\_corr[] = pitch\_bias, pitch\_rate, pitch\_acceleration

Yaw bias, rate, and acceleration corrections: yaw\_corr[] = yaw\_bias, yaw\_rate, yaw\_acceleration

Due to the different orders of the Earth-view and lunar correction models, this model is stored as an array in the model structure along with a field defining the model order. The precision model is dealt with in more detail in the line-of-sight correction algorithm description.

The processing steps to construct the corrected attitude sequence is the same for lunar/stellar acquisitions, although the interpretation of the roll-pitch-yaw values is slightly different, and proceeds as follows:

For each point in the attitude sequence j = 0 to K-1:

1. Compute the rotation matrix corresponding to the jth ECI roll-pitch-yaw values:

**M**ACS2ECI =



1. Compute the precision correction at the time (t\_att = att\_seconds + att\_time[j]) corresponding to the attitude sample:
2. 
3. 
4. 

Note that only the seconds of day fields are needed for the attitude and image epochs as they are constrained to be based on the same year and day.

1. Compute the rotation matrix corresponding to roll\_correction (r), pitch\_correction (p), and yaw\_correction (y):

**M**Precision =



1. Compute the composite rotation matrix: **M** = **M**ACS2ECI **M**Precision
2. Compute the composite ACS to ECI roll-pitch-yaw values:



Note that in implementing these calculations it is important to use the ATAN2 rather than the ATAN arctangent implementation in order to retain the correct quadrants for the Euler angles. This is not a concern in Earth-view imagery where the angles are always small, but becomes an issue for these lunar/stellar ACS to ECI angles.

1. Store the composite roll’-pitch’-yaw’ values in the jth row of the corrected attitude data table.

**Correct Ephemeris Sub-Algorithm**

The heritage ALIAS function converts the ephemeris information (position and velocity) from the Earth-Centered Inertial (ECI J2000) system to the ECEF system and applies the ephemeris corrections computed in the LOS/precision correction procedure to both ephemeris sets. Since both ECI and ECEF representations of the ephemeris are now provided by the ancillary data preprocessing algorithm (6.1.4), the first portion of the heritage algorithm is no longer necessary (or could be reused in the ancillary data preprocessing algorithm). Though applying the precision corrections to construct the corrected ephemeris sequence could be said to be overkill for model creation (since the corrections are nominally zero at this point) this capability is required for LOS model correction and is used here to support the use of the model creation algorithm for data simulation and anomaly resolution as it makes it possible to force initial biases into the model. This sub-algorithm will also be used by the LOS/precision correction algorithm to create the precision model.

The precision correction parameters are stored in the LOS model in the spacecraft orbital coordinate system as three position (x\_bias, y\_bias, z\_bias) corrections and three velocity (x\_rate, y\_rate, z\_rate) corrections that, like the attitude corrections, are relative to t\_ref. These values must be converted to the ECEF and ECI coordinate systems. Once the precision correction is determined in the ECEF/ECI coordinate system, the ECEF/ECI ephemeris values can be updated with the precision parameters.

Loop on LOS model ephemeris points j = 0 to N-1

Compute the precision correction:

Calculate delta time for precision correction:

dtime = ephem\_seconds + ephem\_time[j] – t\_ref – image\_seconds

Calculate the change in X, Y, Z due to precision correction. Corrections are in terms of spacecraft orbital coordinates.

dx orb = model precision x\_corr[0] + model precision x\_corr[1] \* dtime

dy orb = model precision y\_corr[0] + model precision y\_corr[1] \* dtime

dz orb = model precision z\_corr[0] + model precision z\_corr[1] \* dtime

where:

model precision x\_corr[0] = precision (orbital) update to X position

model precision y\_corr[0] = precision (orbital) update to Y position

model precision z\_corr[0] = precision (orbital) update to Z position

model precision x\_corr[1] = precision (orbital) update to X velocity

model precision y\_corr[1] = precision (orbital) update to Y velocity

model precision z\_corr[1] = precision (orbital) update to Z velocity

Construct precision position and velocity “delta” vectors.





Calculate the orbit to ECF transformation [ORB2ECEF] using ECEF ephemeris (See the ancillary data preprocessing ADD (6.1.4) for this procedure).

Transform precision “delta” vectors to ECEF.



Adjust ECEF ephemeris by the appropriate “delta” precision vector and store the new ephemeris in the model. These ephemeris points will be used when transforming an input line/sample to an output projection line/sample.



where:

All parameters are 3x1 vectors

ephemeris ecef values are the interpolated one-second ephemeris values in ECEF coordinates

Calculate the orbit to ECI transformation [ORB2ECI] using ECI ephemeris.

Transform precision “delta” vectors to ECI.



Adjust ECI ephemeris by the appropriate “delta” precision vector and store the new ephemeris in the model. These ephemeris points will be used with lunar/stellar observations.



where:

All parameters are 3x1 vectors

ephemeris eci values are the interpolated one-second ECI ephemeris

**Move Satellite Sub-Algorithm**

This function computes the satellite position and velocity at a delta time from the ephemeris reference time using Lagrange interpolation. This is a utility sub-algorithm that accesses the model ephemeris data to provide the TIRS position and velocity at any specified time. Since the model ephemeris arrays are inputs to this sub-algorithm it will work with either the ECI or ECEF ephemeris data.

Table 6‑30 below summarizes the contents of the TIRS LOS model structure. The estimated size of this structure is approximately 1.5 megabytes.

|  |
| --- |
| **LOS Model Structure Contents** |
| Satellite Number (8) |
| Format Version Number (for documentation and backward compatibility) |
| WRS Path |
| WRS Row (may be fractional) |
| Acquisition Type (Earth, Lunar, Stellar) |
| Earth Orientation Parameters |
| UT1UTC Correction (in seconds) |
| Pole Wander X Correction (in arc seconds) |
| Pole Wander Y Correction (in arc seconds) |
| Image Model |
| Number of image lines |
| Image UTC epoch: image\_year, image\_day, image\_seconds |
| For each line: frame time offset (in seconds) from image epoch |
| For each line: roll, pitch, yaw high frequency jitter correction (in radians) |
| Nominal alignment fill table (from CPF) one value per band per SCA (in pixels) |
| Detector alignment fill table (from L0R/L1R) one value per detector (in pixels) |
| Sensor Model |
| TIRS to ACS reference alignment matrix [3x3] |
| Spacecraft center of mass to TIRS offset in ACS reference frame [3x1] in meters |
| Integration Time in seconds |
| Computed Sample Time in seconds |
| Number of SCAs (3) |
| Number of Bands (4) |
| Along-Track IFOV in radians |
| Across-Track IFOVs (MS and pan) in radians |
| Number of Detectors per SCA in each Band (4x1 array) |
| Focal plane model parameters (Legendre coefs) [NSCAxNBANDx2x4] (in radians) |
| Detector delay table [NSCAxNBANDx2xNDET] (in pixels) |
| Scene Select Mirror Model |
| Telescope to SSM alignment matrix [3x3] |
| Number of SSM encoder angles |
| Time from image epoch (one per sample, nominally 20 Hz) (in seconds) |
| SSM angle (one per sample) (in radians) |
| Ephemeris Model |
| Scene ephemeris data UTC epoch: imgeph\_year, imgeph\_day, imgeph\_seconds |
| Number of ephemeris samples |
| Time from epoch (one per sample, nominally 1 Hz) (in seconds) |
| Original ECI position estimate (X, Y, Z) (one set per sample) (in meters) |
| Original ECI velocity estimate (Vx, Vy, Vz) (one set per sample) (in meters/sec) |
| Original ECEF position estimate (X, Y, Z) (one set per sample) (in meters) |
| Original ECEF velocity estimate (Vx, Vy, Vz) (one set per sample) (in meters/sec) |
| Corrected ECI position estimate (X, Y, Z) (one set per sample) (in meters) |
| Corrected ECI velocity estimate (Vx, Vy, Vz) (one set per sample) (in meters/sec) |
| Corrected ECEF position estimate (X, Y, Z) (one set per sample) (in meters) |
| Corrected ECEF velocity estimate (Vx, Vy, Vz) (one set per sample) (in meters/sec) |
| Attitude Model |
| Scene attitude data UTC epoch: imgatt\_year, imgatt\_day, imgatt\_seconds |
| Number of attitude samples |
| Time from epoch (one per sample, nominally 50 Hz) (in seconds) |
| Original Roll, pitch, yaw estimate (one per sample) (in radians) |
| Corrected Roll, pitch, yaw estimate (one per sample) (in radians) |
| Precision Correction Model |
| Precision reference time (t\_ref) seconds from image epoch |
| Ephemeris correction order: eph\_order (0 none, 1 for Earth-view and lunar/stellar) |
| X correction model: x\_bias, x\_rate (meters, meters/sec) |
| Y correction model: y\_bias, y\_rate (meters, meters/sec) |
| Z correction model: z\_bias, z\_rate (meters, meters/sec) |
| Attitude correction order: att\_order (0 none, 1 for Earth, 2 for lunar/stellar) |
| Roll correction model: roll\_bias, roll\_rate, roll\_acc (rad, rad/sec, rad/sec2) |
| Pitch correction model: pitch\_bias, pitch\_rate, pitch\_acc (rad, rad/sec, rad/sec2) |
| Yaw correction model: yaw\_bias, yaw\_rate, yaw\_acc (rad, rad/sec, rad/sec2) |

Table 6‑30. TIRS LOS Model Structure Contents

Note that in the precision correction model only the correction model array elements up to att\_order are valid. For example, for Earth-view scenes att\_order = 1 and roll\_corr[0] = roll\_bias, roll\_corr[1] = roll\_rate and roll\_corr[2] is not used.

#### Maturity

Though much of the OLI model creation algorithm was reusable for TIRS there are several areas where changes were necessary:

1. The TIRS SSM telemetry is extracted from the ancillary data stream, quality checked, and smoothed. The SSM model is a new component that has been added to the TIRS LOS model. Known issues with the performance of the SSM encoder may necessitate the development of additional correction logic to account for errors in the high-order encoder bits (see also note #7 below). This is not included in the current baseline.
2. Analysis of the TIRS optical model has shown that the 2nd order Legendre polynomial model used to generate OLI lines-of-sight will not be adequate for TIRS. This is due primarily to the larger field of view of the TIRS SCAs. Since each SCA covers a larger portion of the instrument’s field of view, it is subject to more of the optical distortion variations that occur across the field of view. Initial analysis indicates that a 3rd order Legendre model will capture the nominal TIRS detector lines of sight for each band and each SCA with sufficient fidelity. The TIRS detector line-of-sight model has been updated accordingly.
3. Some features of the OLI instrument (e.g., odd-even detector offset) are not relevant for TIRS but are retained, with appropriate calibration parameters set to zero, to maintain commonality across the models.
4. Temperature measurements at the TIRS mount points and in the scene select mirror mechanism may allow the inclusion of temperature dependent effects in the TIRS to ACS (and TIRS to OLI) alignment and/or the SSM model. This is not included in the baseline model but we do assume that the relevant temperature telemetry is trended (see note #8 below).

#### Notes

Some additional background assumptions and notes include the following:

1. The static precession, nutation, and sidereal time parameters needed to convert Earth-Centered Inertial J2000 to/from WGS84 Earth-Fixed are built into the software rather than being provided as input data. The dynamic terms (UT1UTC correction, polar wander) are provided in the CPF. For Landsat 8, the CPF was expanded to include a leap second table to allow for converting spacecraft TAI-reference time codes to UTC. This TAI to UTC time conversion and the ECI/ECEF conversion algorithm are discussed in the ancillary data preprocessing algorithm description document (6.1.4).
2. While it seems to be generally agreed that the TIRS Level 0R/Level 1R data will not use fill pixels to nominally align bands and/or SCAs, it may include fill pixels to achieve nominal detector-to-detector alignment in the case where bad detectors are replaced from the redundant detector row. Since this is an existing capability in the heritage OLI logic, the L0R/L1R detector alignment fill table input identified in the input table will be retained as the mechanism for the Level 0R/1R data to identify the number of any fill pixels used. In practice, it may be preferable to keep the TIRS L0R in strict time order (with no fill) even if dead detector replacement is performed.
3. The "thresholds and limits" parameters, stored either in system tables or the database for L7 and ALI, will be included in the CPF for Landsat 8/9. This will make date specific changes, e.g., due to a change in the nominal orbit during early- or late-mission operations, easier to manage.
4. The current algorithm baseline is to use the heritage attitude model roll-pitch-yaw representation. This could be updated in a future revision to use a quaternion representation. This is the motivation for including both quaternion and roll-pitch-yaw representations of the attitude data sequence in the output from the ancillary data preprocessing algorithm (6.1.4).
5. This algorithm includes a simple image time code validation/smoother function to fix errors and/or smooth out quantization effects in the downlinked time codes. This may not be necessary or it may need to be more elaborate depending on the reliability of the TIRS time codes.
6. The baseline algorithm prototype implementation allows the precision correction model parameters to be provided as optional input parameters. This would not be used for operational data processing and these parameters would not ordinarily be provided, with their values defaulting to those set in the Initialize the Precision Model sub-algorithm. Having such an R&D capability to force model corrections at model creation time can prove useful in applications such as data simulation and anomaly resolution.
7. The reliability of the SSM encoder telemetry is unknown. As a contingency, the baseline model includes a quality check and smoothing logic to allow for SSM encoder data preprocessing. The quality check is based on a threshold check that ensures sample-to-sample consistency. The threshold is a CPF parameter. A check for consistency with a nominal value may also be required.
8. The baseline model does not include any temperature dependent effects in either the SSM or in the overall TIRS alignment. The temperature telemetry provided in the TIRS ancillary data would provide a means for investigating any such dependencies on orbit. This algorithm assumes that the TIRS temperature telemetry will be collected and trended by the radiometric processing algorithms and would therefore be available, if needed, for future implementation of temperature-based calibration adjustments.
9. The SSM encoder position is provided at a 20 Hz sampling rate even though the SSM telemetry packets are generated at 1 Hz. Each packet contains 20 samples, with each sample representing one 24-bit encoder read out.
10. The TIRS detector deselect mechanism and detector offset geometry differ from the OLI versions but the same correction logic can be applied. In the OLI case, adjacent redundant detectors are switched on in place of the defective primary detectors causing the active detector location to be shifted in the along-track direction. This shift is in addition to the normal even/odd detector offset. For TIRS, there is no even/odd offset and instead of having individual redundant detectors that must be switched on individually, an entire redundant row of detectors is downlinked for each band. Detector replacement is performed in Level 0 processing where the samples from defective primary detectors are swapped with the samples from the corresponding detector in the redundant row. The net result is that TIRS detectors will have an integer detector offset of either 0, for detectors from the primary row, or whatever the line offset happens to be between the primary and redundant detector rows, for detectors that are replaced/deselected. Since the selection of primary and redundant rows will be made separately for each band on each SCA, these offsets can vary from SCA-to-SCA but will be constant within a given band on a given SCA.