NASA/CR- *1998-* **208213**

FINAL REPORT

GLAS SPACECRAFT POINTING STUDY
NASA GODDARD SPACE FLIGHT CENTER

NAG5-6146

PRINCIPAL INVESTIGATOR:

DR. GEORGE H. BORN

9/1/97 – 8/31/98

GLAS Pointing Study

To: Joe Dezio, Darell Zimbelman, Bob Schutz, Jay Zwalley, Scott Mitchell

From: Kenn Gold, Michael Ondrey, Dan Kubitschek, Penina Axelrad, Attila Komjathy, George Born
        Colorado Center for Astrodynamics Research, University of Colorado- Boulder

Date: 05/12/98

Re: GLAS Attitude Control Algorithm

Introduction

Science requirements for the GLAS mission demand that the laser altimeter be pointed to within 50 m of the location of the previous repeat ground track. The satellite will be flown in a repeat orbit of 182 days. Operationally, the required pointing information will be determined on the ground using the nominal ground track, to which pointing is desired, and the current/propagated orbit of the satellite as inputs to the roll computation algorithm developed by CCAR. The roll profile will be used to generate a set of fit coefficients which can be uploaded on a daily basis and used by the on-board attitude control system. In addition, an algorithm has been developed for computation of the associated command quaternions which will be necessary when pointing at targets of opportunity.

It may be desirable in the future to perform the roll calculation in an autonomous real-time mode on-board the spacecraft. GPS can provide near real-time tracking of the satellite, and the nominal ground track can be stored in the on-board computer. It will be necessary to choose the spacing of this nominal ground track to meet storage requirements in the on-board environment. Several methods for generating the roll profile from a sparse reference ground track are presented.

Development and Testing of the Roll Algorithm

Orbit and Ground Track Used for Simulation

Development of the algorithm required the generation of reference and perturbed orbits and ground tracks. The calculation of the roll profile begins by defining a nominal or reference satellite epoch-state vector in the ECI coordinate system. A second state vector is defined with a 1 Km offset from the first at the equator (accomplished by adjusting the RAAN of the second vector), and with an inclination perturbation of 0.03 degrees. This inclination change, discussed later in this memo, is comparable to that expected for GLAS due to Lunar/Solar perturbations. Both of these initial epoch-state vectors are then integrated to produce orbit arcs with a high precision orbit propagator. The Jet Propulsion Laboratory's Orbit Integration module (from Gipsy-Oasis II) was used in the study. The ground track (relative to the ellipsoid) is calculated for each of the orbits by first converting the Cartesian state vector to Geodetic Latitude and Longitude, and zeroing the height component. These values are then converted back into Cartesian X,Y,Z coordinates to represent the respective ground track points, and the pointing algorithm is applied.

(Note: For the most accurate roll calculation and repeat of ground track pointing, it will be necessary to include topographic map information into the CCAR algorithm. This is important

over the polar caps where elevation is around 3000 m above the ellipsoid. The algorithm modification involves determining the offset between the ellipsoid and local topography. This offset will then be added into the zeroed height component before re-converting to Cartesian ground track points)

Roll Algorithm

The algorithm is defined as follows:

1) Load a file containing the following elements:

- Set of reference XYZ ECI ground track coordinates.

$$\times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times$$

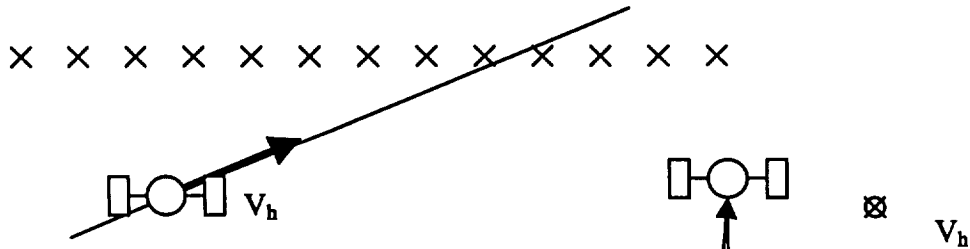- Set of propagated XYZ ECI satellite position and velocity.

- Set of geodetic nadir points corresponding to the propagated satellite position.

2) Compute the velocity component of the satellite, V_h, that is parallel to the local geodetic horizontal.

$$\vec{V}_h = (\vec{n} \times \vec{V}) \times \vec{n} \qquad (1)$$

Top View

Rear View

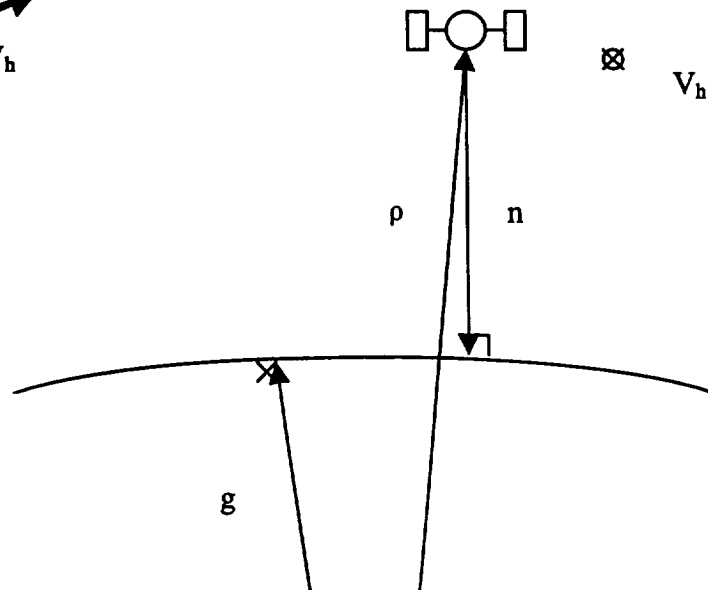Figure 1. Description of orbit geometry

3) At each epoch in the satellite's orbit, sweep along a specified ground track interval, computing the pointing vector, c_i.

$$\vec{c}_i = \vec{g}_i - \vec{\rho} \qquad (2)$$

Define a quantity called pointer whose value will be zero whenever c is normal to V_h.

$$pointer = \left(\frac{\vec{c}_i \bullet \vec{V}_h}{|\vec{c}_i| |\vec{V}_h|} \right) \qquad (3)$$
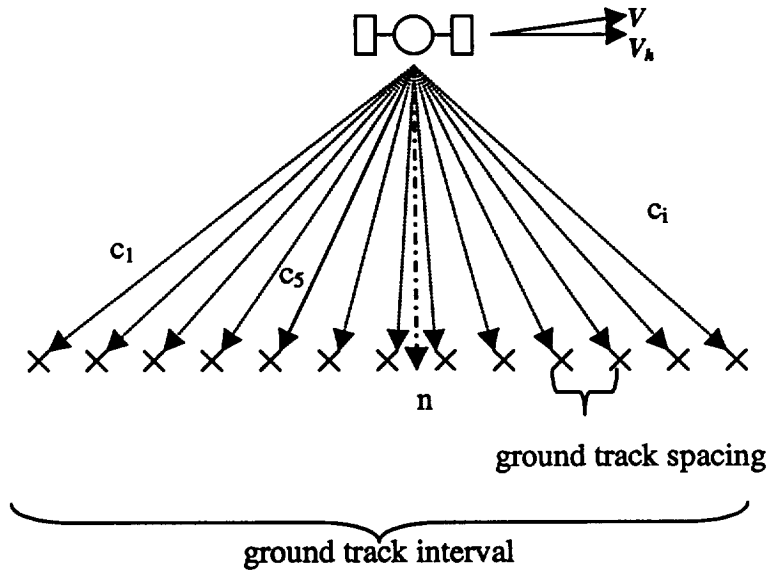

ground track spacing

ground track interval

Figure 2. Side view of satellite/ groundtrack geometry

4) Interpolate the XYZ ground track coordinates within the ground track interval for a *pointer* value of zero (the roll angle represents a rotation about the local horizontal velocity direction vector). This interpolated ground track location is called g^*, and its resultant pointing vector called c^*.
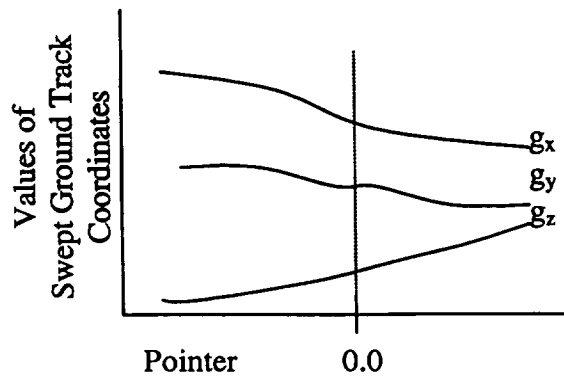

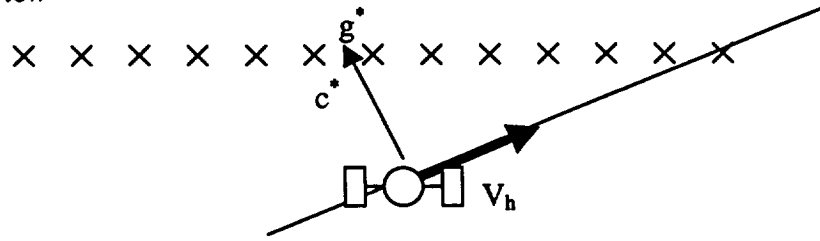Figure 3. Ground Track coordinate interpolation

Top View

Figure 4. Top view of pointing vector, c*

5) Compute the roll angle, β, measured with respect to the nadir direction vector of the satellite.

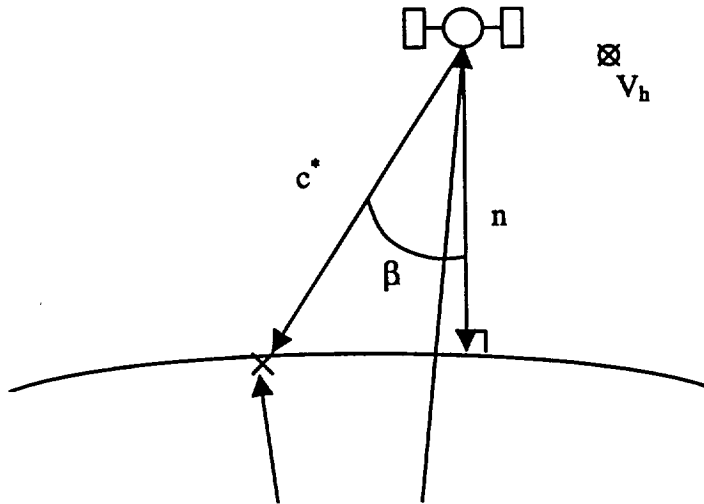$$\beta = \cos^{-1}\left(\frac{\vec{n} \bullet \vec{c}^{\,*}}{|\vec{n}||\vec{c}^{\,*}|}\right) \qquad (4)$$

Figure 5. Definition of positive roll angle.

6) The sign of the roll angle (see Fig. 5) is determined by taking the dot product between the "horizontal" angular momentum, H, and the pointing vector as follows:

$$\vec{H} = \vec{n} \times \vec{V}_h \qquad (5)$$

$$product = -\left(\vec{c}^{\,*} \bullet \vec{H}\right) \qquad (6)$$

$$\beta_{new} = sign(product) \cdot \beta \qquad (7)$$

Simulated Roll Profiles

Figure 6 shows the pointing profile that would result with the orbit perturbed 1 Km to the west of the nominal orbit at the equator. This represents the maximum expected deviation due to atmospheric drag in ground track control. A gravity field of degree and order 50x50 is used throughout the orbit integration. No Luni-solar effects are included. Latitude of the nominal ground track is also shown for reference. Additional effects that must be included in the roll

4

profile simulation are the effects of Luni-Solar perturbations over the the 182 day epoch difference between repeat tracks. As shown in figure 7, a long term orbit prediction reveals inclination changes as large as 0.03 degrees peak to peak. This inclination perturbation is simulated by decreasing the initial epoch state inclination of the second orbit by 0.03 degrees. This results in a nearly 2 km separation between the reference and current ground track at high latitudes. Figure 8 shows the resulting roll profiles if these effects are included along with the 1 km equatorial offset case. The geodetic latitude of the reference orbit is also shown in figure 8.
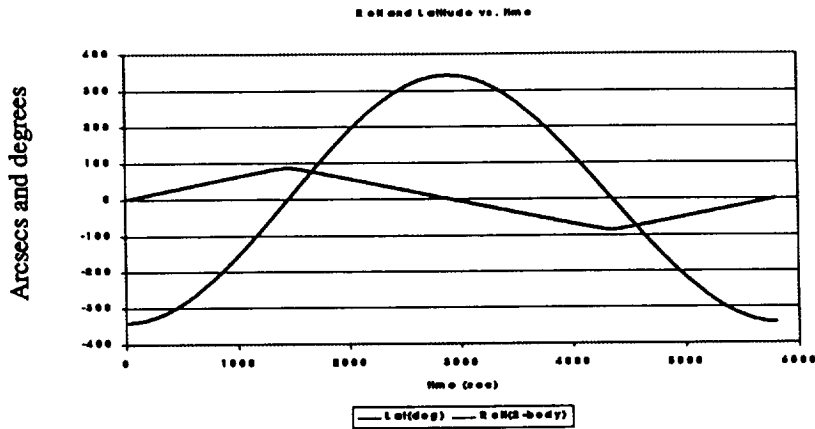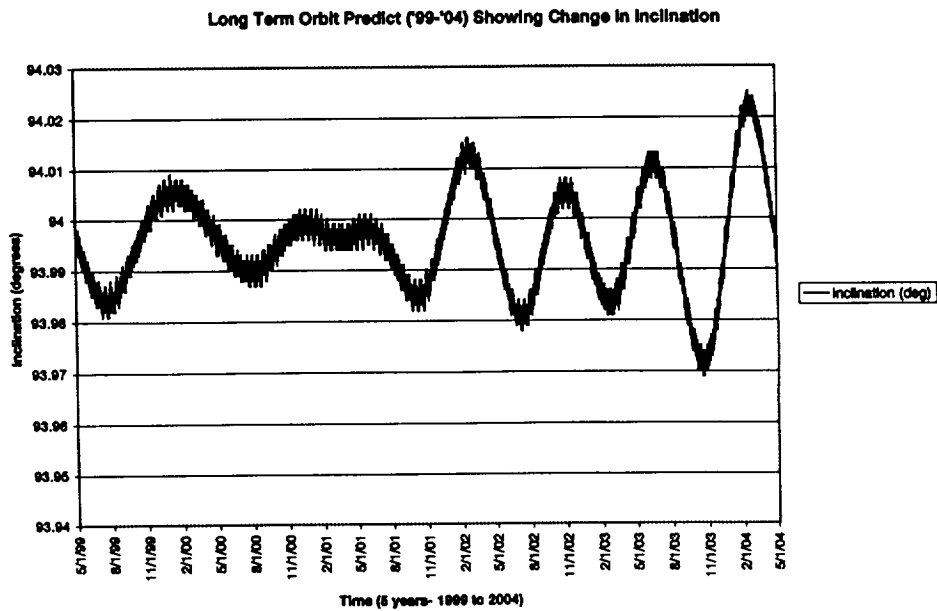

Figure 6 Roll Profile with 1 km equatorial offset

Figure 7. Expected Inclination Change for GLAS over Five Years
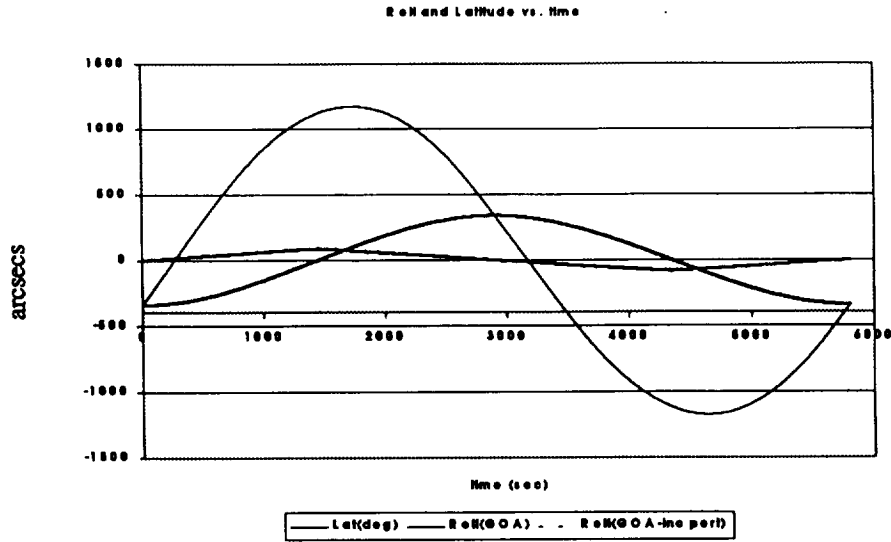
Roll and Latitude vs. time

Figure 8. Effects of inclination perturbation on Roll profile.

Figure 9. shows the difference between roll profiles due to a 182 day epoch change. This is the effect on the ground track due soley to the location of the sun and moon relative to the satellite orbit being different after 182 days and does not account for the inclination perturbation. The profile generated with these conditions (equator offset, inclination perturbation, epoch difference) at 1 second intervals became the truth model for later real-time roll algorithm simulations.
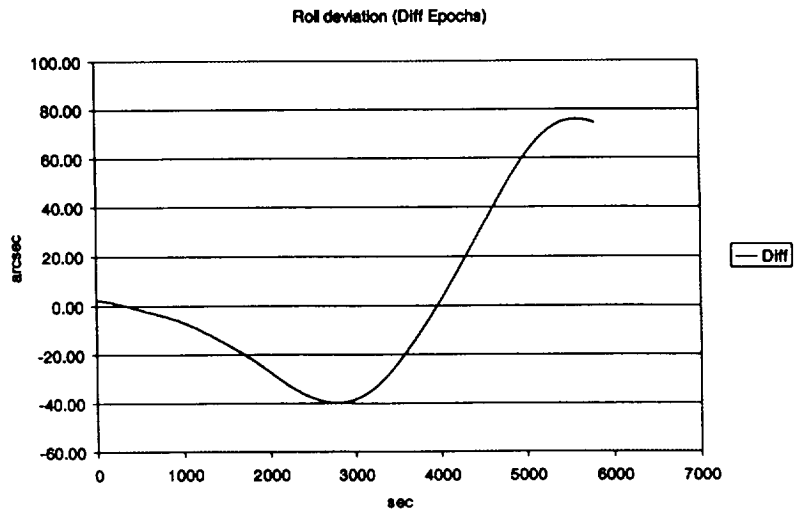

Roll deviation (Diff Epochs)

Figure 9. Differences in Roll induced by differing epochs.

GLAS Attitude Control

Polynomial Fit to the Roll Profile

For purposes of attitude control, computing and uploading the 1/second roll angle information used thus far is not practical. Instead, the roll values should be defined as a continuous function of time rather than a discrete data set. The objective is to minimize the number of daily uploaded commands to the spacecraft. Given that the spacecraft's orbit can be propagated accurately enough for an entire day, it would be advantageous to upload only one set of polynomial coefficients which define a continuous, 24 hour roll profile. This would also minimize the amount of data to be uploaded to the spacecraft. Results have shown that a roll accuracy of better than 2 arcsecs RMS is possible over a period of 4 orbits with a 25th order polynomial fit (see Table 1).

Table 1. RMS and Maximum Deviation Resulting from Polynomial Fitting

Polynomial Order	Number of Orbits	RMS of fit *arcseconds*	Maximum Deviation *arcseconds*
9	1	0.3	0.6
	2	10.7	21.0
13	2	0.7	1.5
	3	7.1	12.0
	4	124.3	220.0
14	2	0.4	1.2
	3	3.3	6.0
	4	72.0	115.0
22	3	0.9	2.6
	4	2.9	7.0
23	3	0.8	2.1
	4	2.1	5.0
24	3	0.8	2.1
	4	2.2	4.5
25	3	0.8	2.1
	4	2.0	5.3

Table 1 indicates that one set of polynomial coefficients cannot be used to describe the roll angle over the 24 hour period to the required accuracy of 5 arcseconds maximum deviation. However, polynomial fits to 20 successive 1.2 hour periods have been accomplished and used to reconstruct the roll angle over the entire 24 hour period with an accuracy of better than 5 arcsecs in maximum deviation. Thus, 20 sets of 7th order polynomial coefficients, with an associated start time tag, can be determined and uploaded once per day. Figure 10 shows the true and reconstructed roll profiles over a 24 hour period for a 1 Km equatorial offset. Figure 11 shows the difference between the true roll angle and the roll angle which was reconstructed using 20 sets of 7th order polynomial coefficients. Figure 12 shows the true and reconstructed roll profiles over

degree inclination offset. Figure 13 shows the difference between the true roll angle and the reconstructed roll angle.
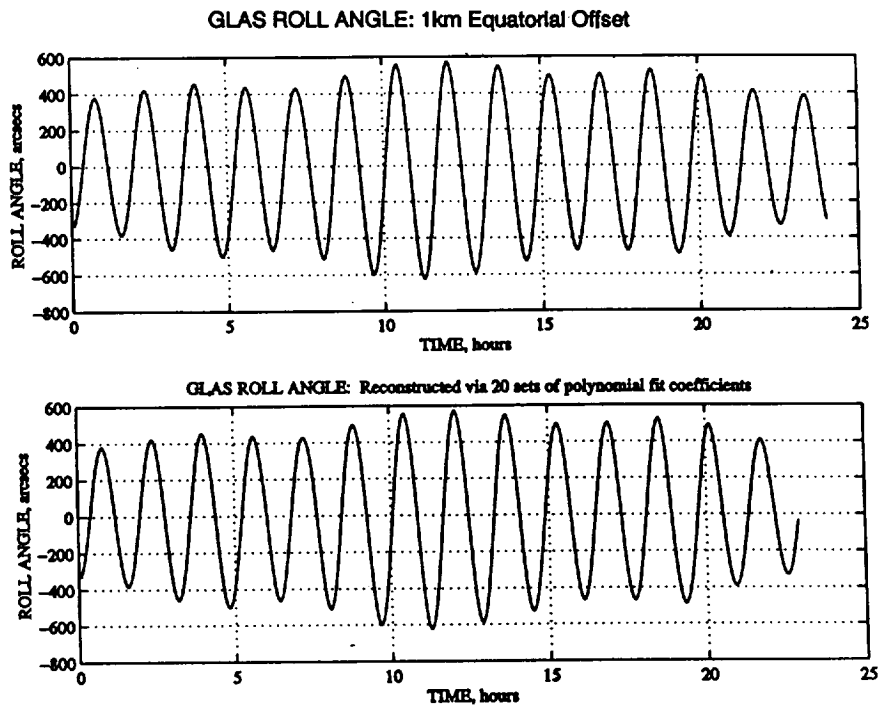
GLAS ROLL ANGLE: 1km Equatorial Offset

GLAS ROLL ANGLE: Reconstructed via 20 sets of polynomial fit coefficients

Figure 10. Original and reconstructed GLAS roll angle over a 24 hour period.
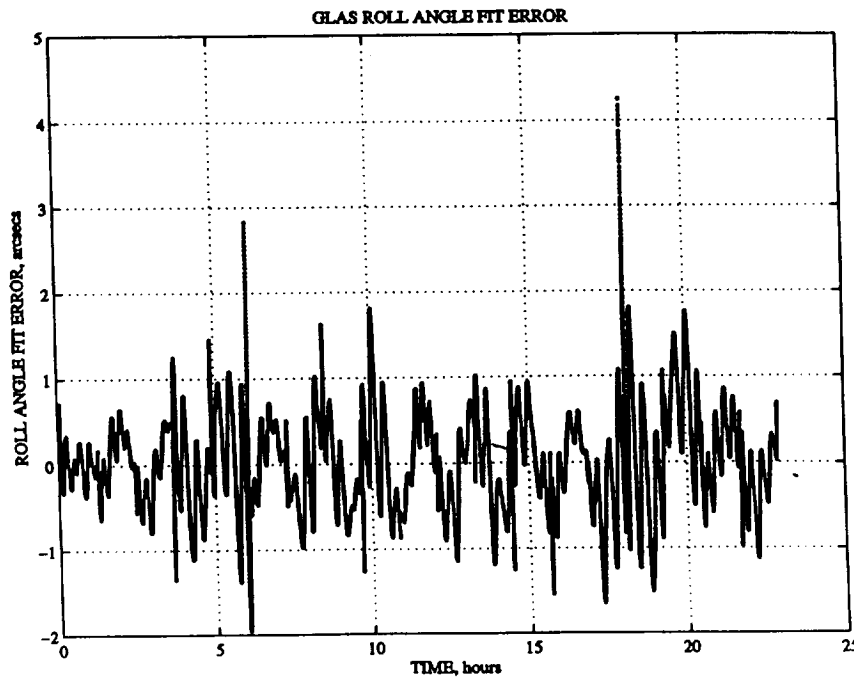
GLAS ROLL ANGLE FIT ERROR

Figure 11. Difference between original and reconstructed roll angle over a 24 hour period.
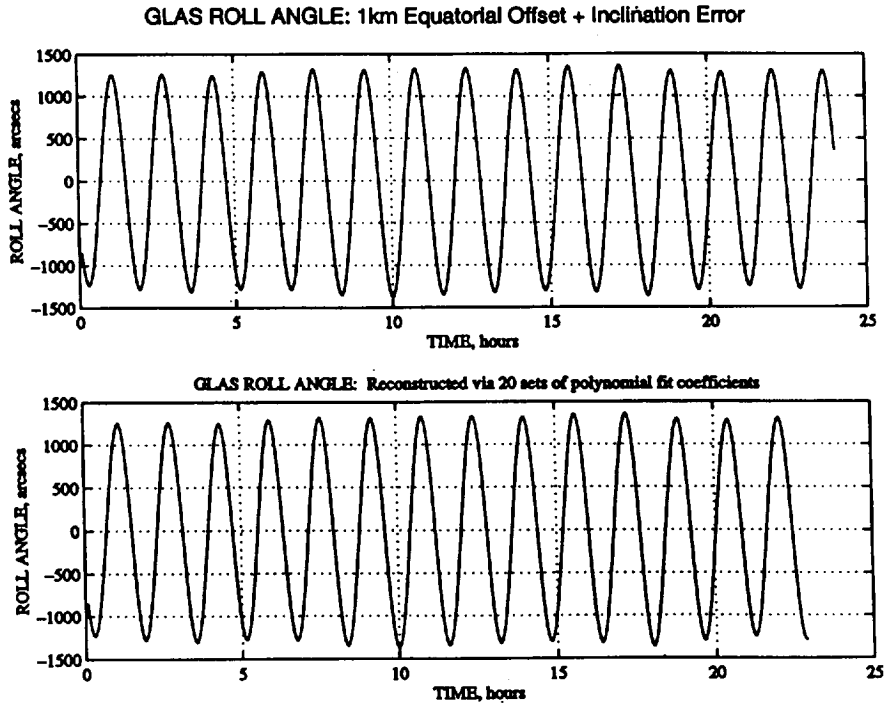
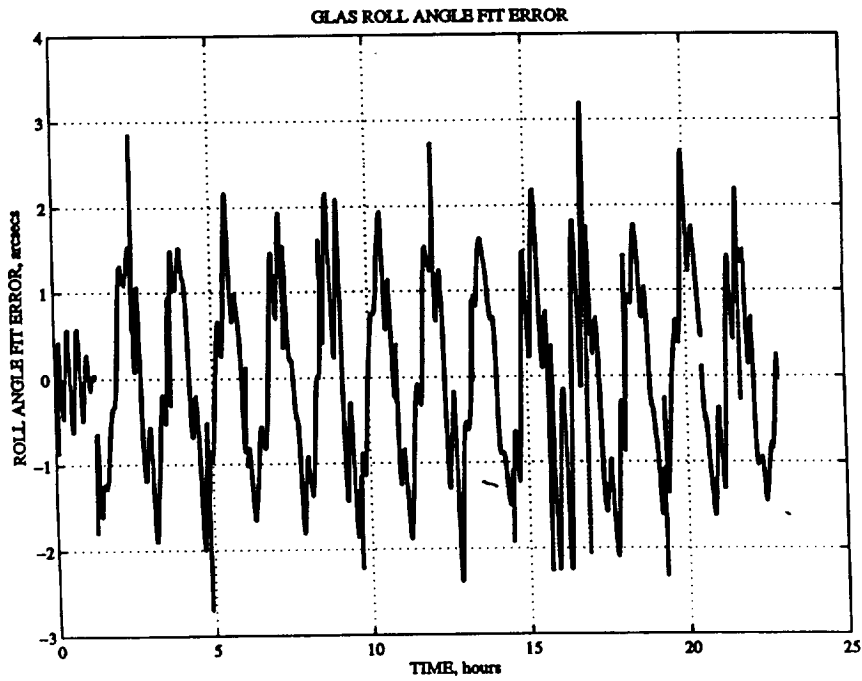**Figure 12. Original and reconstructed GLAS roll angle over a 24 hour period.**

Figure 13. Difference between original and reconstructed roll angle over a 24 hour period.

Pointing at Targets of Opportunity

When pointing at targets of opportunity, a simple roll about the local geodetic horizontal direction vector may not be adequate. In this case, a set of command quaternions can be determined and uploaded to the on-board attitude control system.

A spacecraft attitude control system requires a commanded attitude in the form of a quaternion relating the reference ECI frame to the spacecraft body-fixed frame. The previous algorithm defines roll angle relative to the orbit frame. This section defines an additional transformation to create the required commanded quaternion.

The command quaternions can be computed at any given time assuming the following predicted data are available relative to the ECI frame of reference:

- Argument of latitude for the predicted satellite orbit
- $\beta' \equiv$ angle between the orbit plane and the sun line direction vector
- Satellite position and velocity
- Sub-satellite ground track coordinates
- Target track coordinates or target coordinates

These command quaternions are defined as a function of roll angle, β, during the nominal attitude pointing mode and can be computed independently once a target of opportunity has been identified. Based on our understanding of nominal spacecraft attitude maneuvers for solar cell illumination, the following is an algorithm for computing the command quaternions during the nominal attitude pointing mode for each β' case:

1) Define the orbit frame of reference to be the local geodetic vertical/local geodetic horizontal (LVLH) frame

2) Compute the quaternion which represents the transformation from the ECI frame of reference to the orbit frame of reference,

$$\bar{q}_{0 \leftarrow ECI}$$

3) Compute the roll angle, β, about the local geodetic horizontal direction vector using the GLAS roll algorithm described earlier in this memo.

4) Compute the quaternion which represents the transformation from the orbit frame of reference to the s/c body fixed frame of reference

Case: $-32° < \beta' < 0°$, $\Psi_{yaw} = 180°$

$$\bar{q}_{s/c \leftarrow 0} = \left\{ \begin{array}{c} -\sin(\beta/2) \\ 0 \\ 0 \\ \cos(\beta/2) \end{array} \right\}$$

Case: $32° > \beta' > 0°$, $\Psi_{yaw} = 0°$

$$\overline{q}_{s/c \leftarrow 0} = \begin{Bmatrix} \sin(\beta/2) \\ 0 \\ 0 \\ \cos(\beta/2) \end{Bmatrix}$$

Case: $\beta' \geq 32°$, $\Psi_{yaw} = -90°$

$$\overline{q}_{s/c \leftarrow 0} = \begin{Bmatrix} 0 \\ \sin(\beta/2) \\ 0 \\ \cos(\beta/2) \end{Bmatrix} \otimes \begin{Bmatrix} 0 \\ 0 \\ -\sin(\pi/4) \\ \cos(\pi/4) \end{Bmatrix}$$

Case: $\beta' \leq -32°$, $\Psi_{yaw} = 90°$

$$\overline{q}_{s/c \leftarrow 0} = \begin{Bmatrix} 0 \\ -\sin(\beta/2) \\ 0 \\ \cos(\beta/2) \end{Bmatrix} \otimes \begin{Bmatrix} 0 \\ 0 \\ \sin(\pi/4) \\ \cos(\pi/4) \end{Bmatrix}$$

5) Compute the command quaternion which represents the orientation of the s/c body fixed frame of reference relative to the ECI frame

$$\overline{q}_{cmd} = \overline{q}_{s/c \leftarrow 0} \otimes \overline{q}_{0 \leftarrow ECI}$$

Roll Profile Calculation in the On-board Environment

Interpolation Problem

In this section we examine the problem of implementing a roll algorithm on-board the satellite. The proposed procedure is to store the reference ground track on-board and to compute the roll angle using a spacecraft ephemeris either computed on-board or uploaded from the ground. We do not examine the effects of ephemeris errors here, only errors introduced by the roll algorithm.

The roll profile (including node and inclination effects) illustrated in Figure 8 is used as a reference profile to examine the effects of sparse reference ground track points on the roll algorithm. The algorithm was originally implemented in MATLAB, with an option to use a sparse nominal ground track instead of the 1 sec interval version. A reduction in ground track spacing to 10 sec, in order to reduce on-board storage shows generally good agreement at every point except at the crossover between the two ground tracks, where the roll angle approaches zero. Figure 14 shows the difference between the reference (ground track points at 1 sec spacing)

and the sparse ground track roll profile for one of the crossings. The MATLAB interpolator does not seem to be accurate enough to produce the desired results.
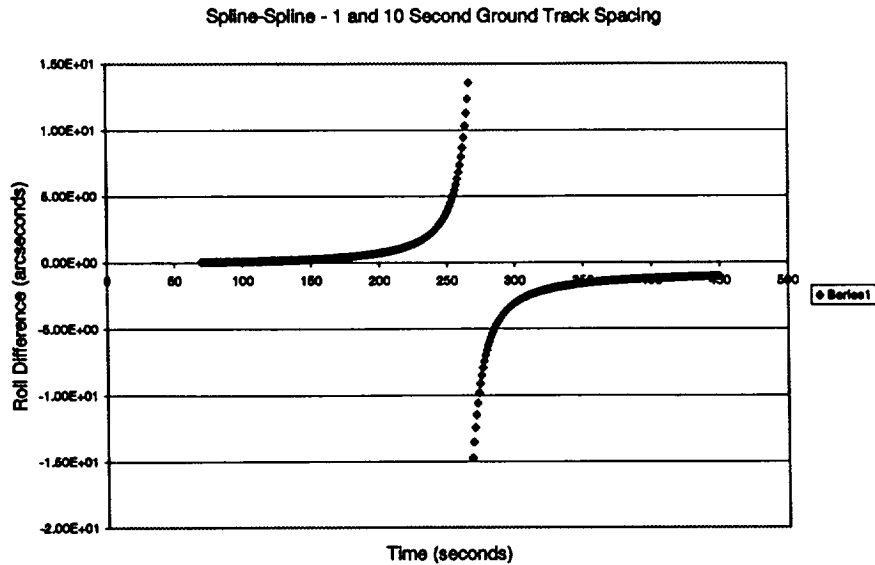
Spline-Spline - 1 and 10 Second Ground Track Spacing

Figure 14. Difference between Reference and Sparse Ground track Roll profiles

Basically the problem is that as roll approaches zero, the roll rate has its maximum value. With a sparse ground track, there are not enough points to pick up this rate increase, so the calculated roll begins to lag behind the reference. Hence, small changes or errors are magnified as the roll angle approaches zero.

The relevant information is that even at the worst case for the roll problem, the time of cross over can be determined to better than 1 second. Thus, the independent variable can be identified exactly when crossover occurs, (0 roll should result). The entire behavior from when the calculated roll deviates from the reference by 1 arcsec until it returns to being within 1 second covers a time span of about 100 seconds. The reference roll curve is linear over such a short span. Hence, one may avoid the problem illustrated in Fig. 14 by interpolating through zero roll angle at a crossover point. Several other solutions to this problem were examined, and are described next.

FORTRAN Implementation

The previous discussion is based on third order quadratic interpolation for determining the ground track point from which to compute β. In order to improve accuracy, use of a more sophisticated interpolater was examined. The MATLAB code was ported to FORTRAN 77 to make use of other math libraries available, and to increase the speed of calculations so multiple cases could be analyzed. The interpolator used in this version of the code comes from Jet Propulsion Laboratories MATH 77 library. The algorithm calculates a least squares polynomial fit to the data points, and can fit polynomials up to 15th order. The JPL interpolation algorithm results in improved results with ground track spacings of 10 sec. The feature associated with a crossover is no longer evident as shown in Figure 15, which used a 3rd order fit. At sparse ground track

12

spacing of 1 minute however, the feature re-appears at crossover, and the overall accuracy is on the order of several arc-seconds.
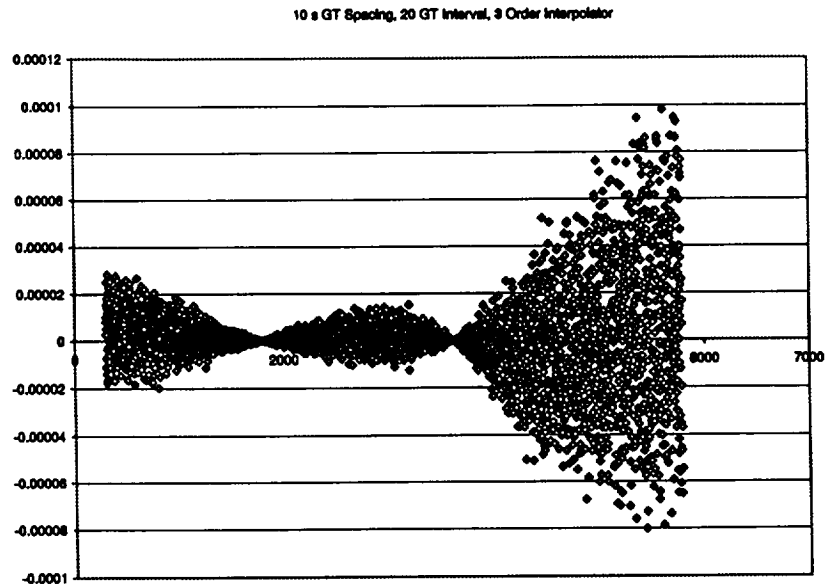
Figure 15. Difference between Reference and 10 Second Sparse Ground track Using FORTRAN Program

Double Interpolation with Geodetic Latitude

It was observed that accuracy of the interpolater was improved at cross over points if a reference ground track point was available near the crossover point. Hence, the algorithm was modified such that a ground track point would be available near the crossover. This was expected to aid in the interpolation by providing a reference point when the roll rate was near maximum. The Fortran implementation of the roll algorithm was augmented such that two iteration steps are now used. The previous implementation involved the calculation of a pointer for sparse ground track points near the nadir point of the current satellite location. These pointer values, with their corresponding truth ground track points were then passed to the interpolation routine (once each for X,Y, and Z) and the ground track coordinate corresponding to a pointer value of zero was determined. This was then used in the calculation of the roll angle. The new addition still preserves this structure with the addition of an iteration step for each coordinate. The geodetic latitude of the satellites current sub-nadir point is used as the independent variable to interpolate the sparse ground track points (in each coordinate) so that a ground track point near the crossover point is determined. This point then generates a pointer that is passed to the interpolator as before.

The results seem promising with this addition. The asymptotic appearance in the comparisons between 1 sec spaced ground tracks, and sparse ground track cases as shown in Fig. 14 is no longer evident. The meaningful result from the use of latitude as the independent variable is that the roll profile near crossover is as accurate as in any other part of the orbit. Currently however, this accuracy degrades to about 3 arcsec max (~9m) when the ground track points are spaced at 1 min intervals. This level of error is greater than should be expected. Also, discontinuities are observed at extreme latitudes where the latitude is slowly changing.

13

Double Interpolation with Argument of Latitude

In order to correct the high latitude errors that occur with interpolation with geodetic latitude, the algorithm was modified to interpolate based on argument of latitude as the independent variable. The argument of latitude values for both the reference orbit, and the current orbit are calculated and are indexed with the corresponding ground track points for the two orbits. The sparse reference ground track points are interpolated to the argument of latitude of the current satellite location. This scenario eliminates the high latitude problem and works well as shown in Figure 16 for sparse ground track spacing of 60 seconds. A 3rd order fit was used in creating this figure. The first crossover, which occurs near 300 sec is not included in the fit, but results in the higher than expected feature at the start of the curve. The discontinuity at crossover is again evident in this analysis, but the magnitude is greatly reduced.
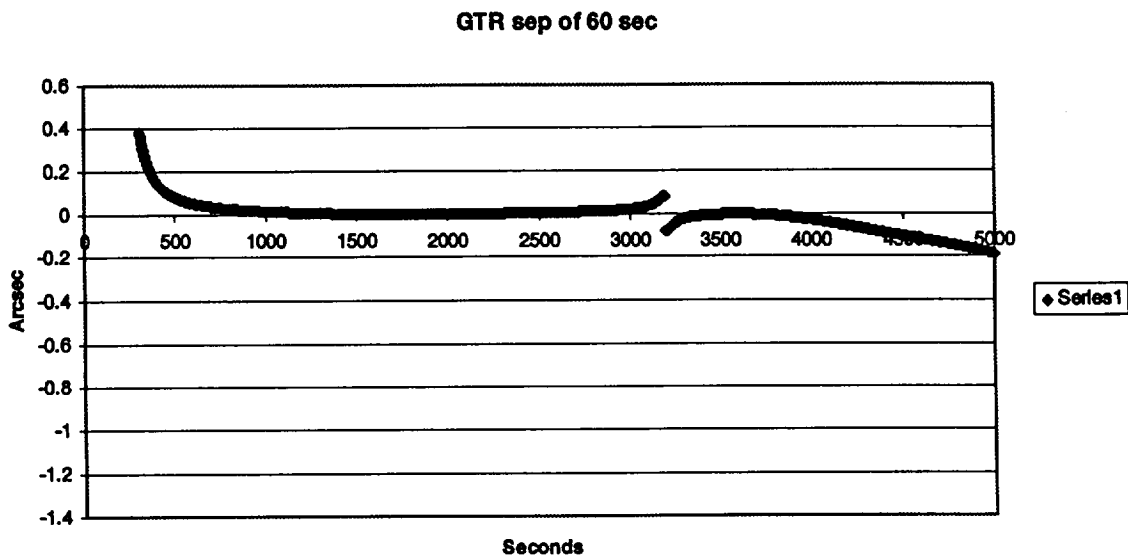
GTR sep of 60 sec

Figure 16 Differences with addition of argument of latitude interpolation

Linear-Fit across Crossover region

As previously discussed, analysis of the discontinuities that occur at crossover lead to a potentially simple change to the real-time roll algorithm. This solution will not work however, if the current and reference orbits are very close. However, for other cases, this method does not involve improved interpolators or double interpolation. An "if" statement can be included to record the independent variable value (either time or arg. of latitude) when roll reaches 50 arcsecs (or other value indicative of crossover approach) , with a negative roll rate. The sign of the calculated roll is then monitored for a change in direction, at which point the independent variable is again recorded. The calculated roll values at these two points can then be interpolated to determine the time/argLat for which Beta should be 0. Then a linear fit from roll = 50 arcsecs to roll = 0 is performed, and the process reversed on the other side of the cross over up to a roll of − 50 arcsecs. The standard algorithm is then used for subsequent calculations.

Close In Solutions

Solutions with no separation between current and nominal ground track, and those with separations at the equator of 10 m, and 100 m perform accurately for the single and double interpolation versions of the algorithm. These cases worked in the MATLAB formulation as well as the augmented Fortran implementation described above, and provide results with errors of only a small fraction of an arcsec. The rate of roll is not as severe for these cases, so the algorithm seems fully capable of handling them.

Summary

The CCAR Roll Algorithm has been tested under a variety of simulation profiles, and performs as expected. Once the roll profile is determined on the ground from propagated orbits, a fit can be performed. The coeficients of this fit, along with the command quaternion set can then be uploaded to the satellite attitude control system.

It has been shown that the roll profile algorithm developed by CCAR can be modified and extended for use in an on-board environment. The major requirement is to reduce the storage requirements for the nominal pointing ground track in the on-board computer. If pointing ground track coordinates are stored at a frequency of 1/minute (Cartesian X,Y,Z) it is possible to calculate roll to within 1 arcsec using a double interpolation scheme with argument of latitude as the independent variable.

The pointing algorithm and various mathematical support functions can be provided in a FORTRAN 77 and MATLAB implementation. The interpolation library used with this version is owned by JPL, and only binary/combined libraries are available to CCAR. The fidelity of the interpolator used should be closely examined when the code is ported to the operational computer system to avoid problems similar to those experienced by CCAR in the development.

The algorithm presented here was tested under the assumption that the ground tracks and pointing sub-satellite laser illumination were on the reference ellipsoid. Since the ice sheets at the poles are approximately 3 kilometers thick in certain areas, including elevation above the ellipsoid in the algorithm should be considered. Topographic maps containing this offset can be used to account for this deviation, as described in this memo. It should be noted that the problem of computing the roll angle at crossover is only a problem for an on-board algorithm. On the ground, the algorithm can use reference ground track information at high enough frequency to avoid this problem.

Future Work

It would be useful to perform several additional studies. Use of a more sophisticated scheme for determining the coefficients for the roll profile should be examined. For example, a sinusoidal fit to the roll profile using several frequencies and amplitudes could significantly reduce the number of coefficients which must be uploaded to the spacecraft. Also, the accuracy of the predicted orbit should be assessed and the resulting impact on the roll accuracy quantified. The use of an on-board scheme for computing the roll angle needs additional refinement and error analysis. The attitude control system for the Ball spacecraft should be examined to see how best to integrate a desired roll profile into the system. Finally, the inclusion of topographic information for the polar caps needs study.