Ball

MEMORANDUM

Date: May 15, 1996

To: Ron Mullerschoen (JPL)

From: Scott Mitchell

Subject: GFO Yaw Behavior for Gipsy/Oasis II

This memo describes and interprets the attached data, which describe the GFO attitude control laws and their implementation into the Gipsy/Oasis II code. A Fortran subroutine which performs some but not all of these functions has been written and tested, and can serve as a starting point for writing the Gipsy subroutine.

Pages 11-7 - 11-12 from the CDR package explain the various reference frames. As shown on p. 11-8, the spacecraft flies pitched up 10 degrees (Xb is 10 degrees above the local horizontal plane). This is an artifact of fitting into a Pegasus fairing. The body Z axis (Zb) is 10 degrees from the altimeter boresight (Zc or nadir). In point mode (the operating mode), the cross product of the altimeter boresight (Zc) with the solar panel normal defines Yc (pp. 11-8 & 9). The attitude is controlled so that Yc is aligned with the cross product of the sun direction and geodetic nadir. The yaw angle is defined as the angle about Zc or the altimeter boresight, and is zero if Xb is in the velocity direction. To optimize pointing of the canted of the solar panel, there is a -6.1928 degrees offset to all yaw values; i.e., at beta = 90 degrees, the yaw angle is constant at -6.1928 degrees.

The spacecraft will follow these attitude laws exactly, except at low beta angles. There are two exceptions at low beta angles: (1) If beta is less than about 3 degrees, the sun location is varied from the actual, to produce superior yaw behavior. (2) If beta is less than about 15 degrees, the actual yaw angle will lag behind the desired yaw by a predictable amount. This lag is predictable because yaw motion is constrained by acceleration and rate limits.

Based on the Gipsy subroutine topex_yaw, I (with help from the GFO attitude engineer) wrote the similar subroutine gfo_yaw. gfo_yaw uses the same input and produces the same type of output as topex_yaw. A copy of this subroutine is attached, and I also will E-Mail you a copy. This subroutine incorporates the attitude laws, including the sun location modification for beta less that 3 degrees, but not the yaw latency at beta less than 15 degrees. Copies of plots made during testing are attached. Since the GPS antenna is mounted nearly directly over the center of mass, yaw errors of modest size will not affect the CM-antenna offset significantly. Yaw errors will, however, contribute to POD errors due to "phase wind-up," as pointed out by you and Willy when we were at JPL for the Gipsy class.

GFO attitude engineer Doug Wiemer generated plots of yaw error for several low beta angles. These plots are contained in an attached memo. Yaw error occurs only near the subsolar and anti-subsolar points. The error or latency is of opposite sign at these two locations. The error is not symmetric about the maximum, so two curve fits (function of orbit angle omega) are used at each beta angle. A multi-dimensional curve fit (orbit angle and beta angle) was thought to be too complicated, and not attempted.

At a given (positive) beta angle, the actual yaw angle will be (all angles in degrees):

(yaw angle calculated by current version of gfo_yaw) + ΔY

where:

$$\Delta Y = \begin{pmatrix} \Delta Y_1(\omega) & \text{if } 80 < \omega < 92 \\ \Delta Y_2(\omega) & \text{if } 92 < \omega < 104 \\ -\Delta Y_1(\omega) & \text{if } 260 < \omega < 272 \\ -\Delta Y_2(\omega) & \text{if } 272 < \omega < 284 \end{pmatrix}$$

where:

ω = orbit angle (as defined in both topex_yaw and gfo_yaw)
$\Delta Y_1(\omega)$ = yaw error best fit equation for $80 < \omega < 92$
$\Delta Y_2(\omega)$ = yaw error best fit equation for $92 < \omega < 104$

The yaw error best fit equations are shown in the lower plots on the 2nd through 11th pages of Wiemer's memo. If the orbit angle is in the range of 260 to 284, (ω-180) must be used in the best fit equations.

For negative beta angles (as defined in both topex_yaw and gfo_yaw), ΔY would be the negative of that calculated from the best fit equations.

I suggest that the subroutine calculate the yaw angle latency (using the best fit equations) for a few beta angles (say 3, 5, 10, and maybe 15 degrees), and interpolate between these based on the actual beta angle. An outline of the possible calculations follows.

(must explicitly declare all new variables)
(to be inserted into gfo_yaw near the end, just before "if(j.eq.1) then")

```
c Calculate yaw angle with yaw latency at low beta angles

c Test to see if this can be skipped

      if ( beta .gt. 15.0 .or. beta .lt. -15.0 .or.
     .   omega .lt. 80. .or.
     .   (omega .gt.104. and. omega .lt. 260.) .or.
     .   omega .gt. 284. ) go to 15

c Calculate yaw angle latency at the actual omega angle, for several beta angles

c Delta yaw at beta = 3 degrees

      if ( omega .gt. 80. .and. omega .le. 92. ) then
       delta_yaw_3 = (curve fit 80-92 Y on Fig. 6 of Wiemer memo, using actual ω)
      elseif ( omega .gt. 92. and. omega .lt. 104. ) then
       delta_yaw_3 = (curve fit 92-104 Y on Fig. 6 of Wiemer memo, using actual ω)
      elseif ( omega .gt. 260. .and. omega .le. 272. ) then
       delta_yaw_3 = - (curve fit 80-92 Y on Fig. 6 of Wiemer memo, using ω–180)
      elseif (omega .gt. 272. .and. omega .lt. 284. ) then
       delta_yaw_3 = - (curve fit 92-104 Y on Fig. 6 of Wiemer memo, using ω–180)

c Delta yaw at beta = 5 degrees

                  (similar code to above, using plots on Fig. 12 of Wiemer memo
                  to get value of delta_yaw_5)

c Delta yaw at beta = 10 degrees

                  (similar code to above, using plots on Fig. 27 of Wiemer memo
                  to get value of delta_yaw_10)

c Delta yaw at beta = 15 degrees

                  (similar code to above, using plots on Fig. 30 of Wiemer memo
                  to get value of delta_yaw_15)

c Interpolate based on actual beta value to get delta yaw

                (interpolate between delta_yaw_3, delta_yaw_5, delta_yaw_10, delta_yaw_15,
                depending on actual beta angle)

c Invert if beta is negative

      if ( beta .lt. o. ) delta_yaw = -delta_yaw

c Calculate actual yaw value including yaw latency

      psi1 = psi1 + delta_yaw

 15 continue
```

By the way, sometime Steve Nerem of UT/CSR would like a copy of the final gfo_yaw subroutine, as they also want to do POD for GFO.

If you have any questions, call me at 303-939-4386, or Doug Wiemer at 303-702-5530.

cc: Willy Bertiger (JPL) (w/o attachments)
Glenn Mapes (w/o attachments)
Brian Fike (w/o attachments)
Doug Wiemer (w attachments)