Preliminary PASS requirements for FOS GIMP correction

Date: 6 June 1991

As outlined during the meetings of May 9th and 23rd, there is a request to the ground system for support of a correction calculation for the FOS geomagnetic field deflection response. The photocathode image/spectrum appears to move around in a cyclical fashion. This apparent motion is really caused by the interaction between the FOS internal detector magnetic field and the geomagnetic field. The absolute detector deflection (x,y) coordinates of any given physical point on the photocathode varies with time. That is, the deflection coordinate system is not fixed. The actual image location in detector coordinates at any given time depends on 1) where the telescope is within the geomagnetic field, 2) how the telescope is oriented in this field, and 3) the observation's deflection pattern parameters.

We have discussed many potential implementation scenarios. In this memo I will present a preliminary set of PASS requirements for two solutions which still remain viable at this time.

There will be a patch to the FOS onboard software to inject x and y deflection offsets before any deflection change is done. These deflection offsets must be provided from the ground periodically during each FOS exposure so that the deflection pattern remains stationary with respect to the photocathode image space. (We will be compensating for the magnetic field changes which occur throughout the exposure duration.) So, from our knowledge of the geomagnetic field, the spacecraft ephemeris and the exposure's deflection pattern we need to compute the delta-x and delta-y motions of the deflection coordinate system as a function of time.

Proposal #1: Upload delta-x,delta-y offsets

In this scheme, the SMS would provide information about the deflection pattern and computation start time. PASS would calculate delta-x and delta-y deflection offsets for the start time plus 15 additional times spaced every 40 seconds. The actual computation algorithms have not yet been supplied. Crudely, they will look as follows:

```
delta-x = function(geomagnetic field,s/c pointing,defl parameters)
delta-y = function(geomagnetic field,s/c pointing,defl parameters)
```

The SMS input might look like:

```
FOSGIMP, _DETECTOR(RED), _PARAM1(17), _PARAM2(54),
       , _COMPSTART((ORB,4000,EASCNCR,07H12M00.000S))
       , TIME = (ORB,4000,EASCNCR, 07H11M00.000S)
```

The PASS output would be a 16-word memory load command which contains the offsets for a 640 second period. The starting memory location is TBD.

location		
delta-x1	delta-y1	offsets at compstart
delta-x2	delta-y2	offsets at compstart + 40s
delta-x3	delta-y3	offsets at compstart + 80s
delta-x4	delta-y4	offsets at compstart + 120s
:	:	
:	:	
delta-x16	delta-y16	offsets at compstart + 600s

There would be a new NSSC-1 application processor which unpacks this table and sends the deflection offset commands to the FOS at the appropriate times.

Proposal #2: Upload polynomial fit coefficients

In this scheme, the SMS would provide information about the deflection pattern, the computation start time, and the computation duration time. PASS would calculate delta-x and delta-y deflection offsets to cover the whole computation duration time. Another offset (x,y) set would be calculated every TBD seconds. The actual computation algorithms have not yet been supplied. (Note: It might turn out that the computation duration time will be fixed and therefore would not be passed in from the SMS.) Crudely, the computation algorithms will look as follows:

```
delta-x = function(geomagnetic field,s/c pointing,defl parameters)
delta-y = function(geomagnetic field,s/c pointing,defl parameters)
```

The SMS input might look like:

```
FOSGIMP, _DETECTOR(RED), _PARAM1(17), _PARAM2(54),
       , _COMPSTART((ORB,4000,EASCNCR,07H12M00.000S))
       , _COMPDUR(1200S)
       , TIME = (ORB,4000,EASCNCR, 07H11M00.000S)
```

After computing all these offsets, PASS should do a TDB-order polynomial fit to the x-offset data and another fit to the y-offset data. The PASS output would be a TBD-word memory load command which contains the polynomial coefficients. The starting memory location is TBD.

```
delta-x = A + Bt + C t**2 + D t**3 + E t**4 + ...
delta-y = Z + Yt + X t**2 + W t**3 + V t**4 + ...
```

location
coefficient A
coefficient B
coefficient C
coefficient D
coefficient E
:
:
coefficient Z
coefficient Y
coefficient X
coefficient W
coefficient V
:
:

There would be a new NSSC-1 application processor which unpacks this table, computes the polynomials every 40 seconds, and sends the offset commands to the FOS.