FOS GIMP Correction Option #3: Interpolating Polynomial

Glenn Foley
Flight Software Systems Branch

June 21, 1991

Review of Onboard Options

o pure ATC commands containing X and Y corrections
+ no NSSC-I flight software/no RTCS index
- requires maximum stored command memory

o NSSC-I AP which unpacks table-loaded compressed X and Y corrections and sends to FOS via RTCS
+ simple onboard logic

o NSSC-I AP which evaluates interpolating polynomials to obtain X and Y corrections and sends to FOS via RTCS
+ minimizes stored command memory usage

Interpolating Polynomial Approach.

o coefficients to X and Y interpolating polynomials would be periodically uplinked to the NSSC-I

o NSSC-I AP would evaluate the interpolating polynomials (by Horner's method) to generate X and Y deflections and send them to the FOS via RTCS

o 3rd order polys would require new coefficients every 30 minutes;
5rd order polys would require new coefficients every 50 minutes;
7rd order polys would require new coefficients every 90 minutes

o with correction LSB = 1 deflection step DAC setting, 8 bit output could specify correction range of 256 steps (-128 to +127 steps, as interpreted by the FOS string)

Example of 3rd Order Inter. Poly.

$$Xcorr = a_3t^3 + a_2t^2 + a_1t + a_0$$

$$Ycorr = b_3t^3 + b_2t^2 + b_1t + b_0$$

where:

t = time in units TBD (seconds or 40 second "ticks", for example)

Xcorr, Ycorr = the X and Y corrections in units of deflection steps, with a range of 8 bits (0 -> 255, or -128 -> +127, as interpreted by the FOS string)

a_3, b_3 = cubic coeff in units of (deflection steps) / (unit time)^3

a_2, b_2 = square coeff in units of (deflection steps) / (unit time)^2

a_1, b_1 = linear coeff in units of (deflection steps) / (unit time)

a_0, b_0 = constant coeff in units of (deflection steps)

Final Stored Command Memory Analysis (1 of 2)

①

NSSC-I AP: Interp. Poly. (assumes single precision coeffs, stored contig. in NSSC-I memory)

3rd order: 4x + 4y coeff + 1 wd zero time

= 9 wds + 4 wds (memory load stored cmd overhead)

= 13 wds every 30 min, or ~ **39 words ATC/orbit**

5th order: 6x + 6y coeff + 1 wd zero time

= 13 wds + 4 wds = 17 wds every 50 min, or

$\frac{17 \text{ wds}}{50 \text{ min}} \cdot \frac{90 \text{ min}}{\text{orbit}} \simeq$ **30.6 words ATC/orbit**

7th order: 8x + 8y coeff + 1 wd zero time

= (16 + 4) wds + (1 + 4) wds = 25 wds every 90 min,

or ~ **25 words ATC/orbit**

FLIGHT SOFTWARE SYSTEMS BRANCH

Final Stored Command Memory Analysis (2 of 2)

② Pure ATC Corrections

3 wd xcorr + 3 wd ycorr every 40 sec

$$\frac{6 \text{ wd ATC}}{40 \text{ sec}} \cdot \frac{60 \text{ sec}}{\text{min}} \cdot \frac{90 \text{ min}}{\text{orbit}} \cong \boxed{810 \text{ words ATC/orbit}}$$

③ NSSC-I AP: Correction Table (16 packed x4y corrections)

$$\frac{(16+4) \text{ ATC wds}}{\text{table}} \cdot \frac{1 \text{ table}}{16 \text{ corrections}} \cdot \frac{1 \text{ correction}}{40 \text{ secs}} \cdot \frac{60 \text{ secs}}{\text{min}} \cdot \frac{90 \text{ min}}{\text{orbit}} \cong \boxed{168.75 \text{ words ATC/orbit}}$$

FLIGHT
SOFTWARE
SYSTEMS
BRANCH

Questions

o which way to proceed (interp. poly. most efficient in terms of stored command memory usage)

Any approaches w/ 4.0 baseline

o how to handle error conditions

- result overflow (>8 bits significance)
- intermediate result overflow (possible, depending on units selected)
- how to handle expiration of time domain