W. W. Hansen Experimental Physics Laboratory
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305 - 4085

Gravity Probe B Relativity Mission

GB-P ITF Timing Test Analysis: ECU Science Signals Time Tagging

S0539 Rev A

July 9, 2003

This document addresses the following T003 requirement:

1) 16.6.7: Experimental Control Unit Science Signals Time Tagging

Prepared by: Paul Shestople — Date: July 9 2003

Approved by: Rich Whelan — Date: July 9, 2003

Approved by: Mac Keiser — Date: July 9, 2003

ITAR Assessment Performed ____ for Tom Langenstein 7/9/03 — ITAR Control Req'd? ☐ Yes ☒ No

Revision History

Rev Level	Comments/notes	Date	Revised By
-	First release of this document	April 15, 2003	Paul Shestople
A	Added Payload Magnetometer Section Added Error bar plot on page 4	July 9, 2003	Paul Shestople

Table of Contents:

1.0 Scope

This document describes the analysis done to verify T003, 16.6.7: Experimental Control Unit Science Signals Time Tagging, which states:

> *The sampling and time tagging of the Experiment Control Unit temperature and magnetometer signals shall be done to within less than 10 ms relative to the 16 f_o rollover counter.*

Conclusions are provided.

This document ☒ Does ☐ Does not provide formal verification of GP-B requirements.
This document ☐ Does ☒ Does not include constraints and restrictions for the Payload.

2.0 Data

Data for analysis of this verification was taken during the Part B Timing Tests, done at Lockheed Palo Alto Integrated Test Facility on March 3, 2003, March 17, 2003 and April 7, 2003. For more information about the ITF part B timing tests, please see S0697.

The test setup injects a sine wave into the ECU. The input sine wave has well understood properties. An output sine wave can be seen in the ECU telemetry; the phase of the output sine wave differs from the phase of the input sine wave by some amount. By comparing the phase of the output sine wave to the phase of the input sine wave, the data latency can be measured. By comparing multiple sets of data, a consistent sampling time emerges.

The ECU has three sampling tables used to collect telemetry. Five mnemonics were arbitrarily chosen from the three tables to be representative of all the mnemonics. The results of the testing are shown in table 1. A sample derivation is presented in Appendix A. Two matlab routines were used to determine the phase of the output sine wave, and are shown and described in appendix B and C.

Table 1: ECU Time Delays in Milliseconds

Period		15						30					
Mnemonic \ Date		3/3/03	Delta	3/17/03	Delta	4/7/03	Delta	3/3/03	Delta	3/17/03	Delta	4/7/03	Delta
DE_VACG_P9A_SIG	FFT (ms)	N/A	N/A	3.649	0.033	3.427	-0.112	3.464	-0.368	3.243	-0.103	3.211	-1.690
	LS (ms)	N/A	N/A	3.601	0.900	3.430	0.857	3.628	0.907	3.303	0.826	3.427	0.857
TE_QBS_a__GT10P	FFT (ms)	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	LS (ms)	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
TE_TelD_1_GT12Q	FFT (ms)	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	LS (ms)	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
TE_PrSLn_aST19P	FFT (ms)	5.180	-0.011	4.955	-0.132	5.010	-0.069	4.950	0.404	4.934	-0.013	4.714	-0.372
	LS (ms)	5.090	1.273	5.036	1.259	5.007	1.252	4.876	1.219	4.645	1.161	4.883	1.221
TE_Win_3_aST25P	FFT (ms)	5.091	0.080	4.955	-0.460	5.055	-0.014	5.131	0.200	4.633	0.016	5.013	-0.151
	LS (ms)	5.055	1.264	5.067	1.267	5.021	1.255	4.945	1.236	4.645	1.161	4.972	1.243
Period		90						180					
Mnemonic \ Date		3/3/03	Delta	3/17/03	Delta	4/7/03	Delta	3/3/03	Delta	3/17/03	Delta	4/7/03	Delta
DE_VACG_P9A_SIG	FFT (ms)	3.518	1.127	2.588	0.444	3.208	-0.908	2.745	-0.129	2.500	0.571	2.613	-0.067
	LS (ms)	3.494	0.874	2.852	0.713	3.319	0.830	2.852	0.713	2.408	0.602	2.586	0.647
TE_QBS_a__GT10P	FFT (ms)	N/A	N/A	N/A	N/A	N/A	N/A	0.411	2.056	0.483	-0.990	0.271	-0.540
	LS (ms)	N/A	N/A	N/A	N/A	N/A	N/A	0.197	0.049	0.224	0.056	0.242	0.060
TE_TelD_1_GT12Q	FFT (ms)	N/A	N/A	N/A	N/A	N/A	N/A	1.500	2.441	2.864	-0.452	1.416	-1.557
	LS (ms)	N/A	N/A	N/A	N/A	N/A	N/A	2.055	0.514	2.577	0.644	1.862	0.466
TE_PrSLn_aST19P	FFT (ms)	4.432	-1.057	4.870	0.664	5.157	0.379	4.326	-0.152	4.727	0.238	4.656	0.843
	LS (ms)	4.759	1.190	5.348	1.337	4.773	1.193	4.467	1.117	4.822	1.206	4.605	1.151
TE_Win_3_aST25P	FFT (ms)	5.329	1.335	6.145	-1.080	4.409	-2.635	4.436	-0.002	4.634	-0.072	4.393	0.764
	LS (ms)	4.634	1.158	3.769	0.942	5.216	1.304	4.368	1.092	4.228	1.057	4.775	1.194

Because of the sampling frequency of ECU sampling table 1 with respect to the input Sine Wave, aliasing precluded calculation of time delays for the TE_QBS_a__GT10P and TE_TelD_1_GT12Q mnemonics for periods less than 180 seconds. FFT Uncertainties were calculated by taking the difference between the plus and minus lobes. LS uncertainties are a flat 25% of the delay value.

An immediate observation concerns the time delay of the GRT mnemonics TE_QBS_a__GT10P and TE_TelD_1_GT12Q. Testing revealed a time latency on the order of one half second for each of these monitors. The delay is due to double-pole filters in series with the GRT thermometry. The filters were fabricated with 1% components; the time delays measured above are self-consistent. From the table it is clear that time tagging of the monitors can be determined to within less than 10 milliseconds relative to the 16 f_o rollover counter in all cases.

Figure 1 below displays graphically the FFT calculated time delays with the associated uncertainty. From the plot it is seen that the values, even with uncertainties, do not exceed the specified 10 ms requirement.

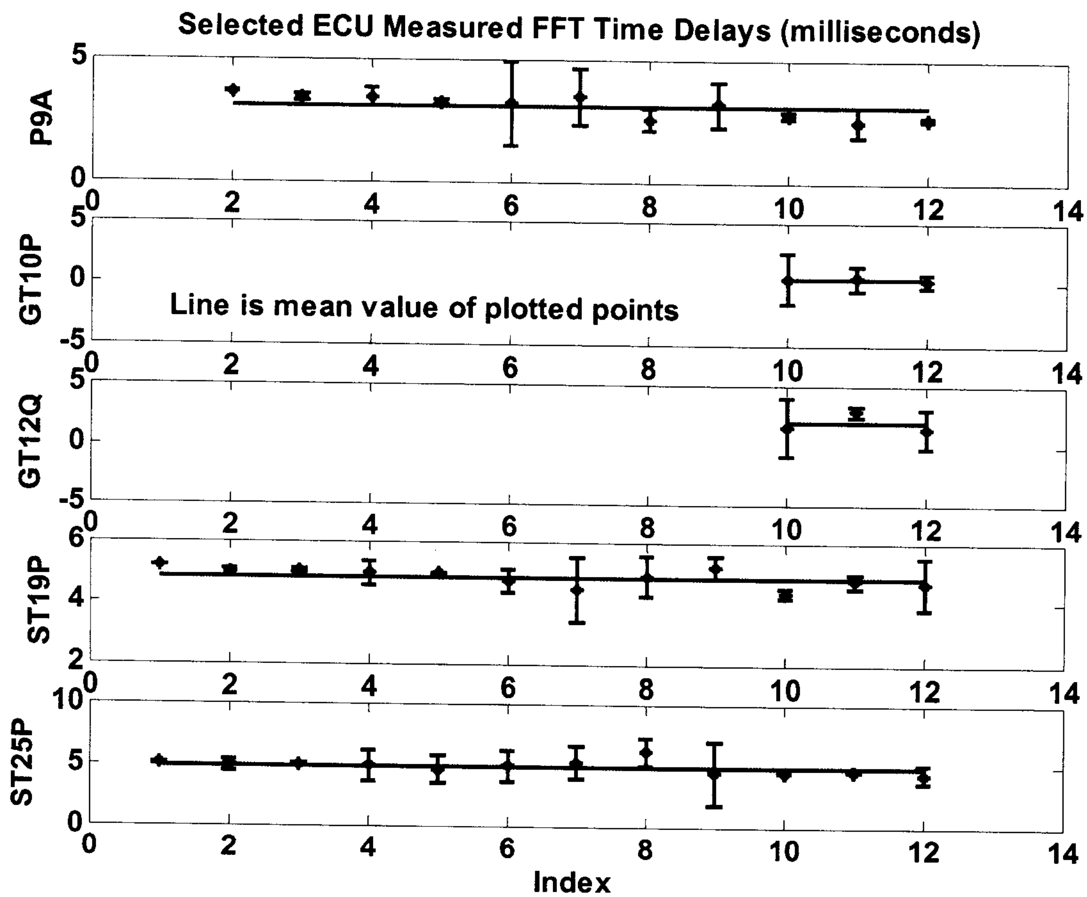

Figure 1: Measured FFT time delays (in milliseconds) of selected ECU mnemonics, including experimental uncertainties. The flat line is the mean value of the plotted values.

3.0 Payload Magnetometer

Four magnetometers, mounted around the dewar, are used by GP-B to quantify how the magnetic fields vary with time. These four magnetometers are referred to as the "payload magnetometers" to differentiate them from the ATC's navigational magnetometers.

The magnetometers and electronics circuit design were procured through a Stanford subcontract with Applied Physical Systems and Iufer & Associates. The ECU houses the electronic boards for the payload magnetometer: 2 magnetometers on ECU A-side, 2 magnetometers on ECU B-side. The magnetometers are powered on when the ECU is on. There are no on/off commands. (ECU User's Manual, LMMS/P480138D, 04/30/2001)

Magnetometer outputs are multiplexed into two Analog / Digital converters. Therefore, only two axes may be read at a time, one through ADC 3A and one through ADC 3B. The ADC conversion is 10 V = 15 bits or 3277 bits / volt. The ADC's are read into telemetry at a 10 Hz rate. All magnetometer readings occur within 4 msec of the 10 Hz ATC strobe. (ECU User's Manual, LMMS/P480138D, 04/30/2001)

The magnetometer electronic boards contain 0.88Hz single pole anti-aliasing analog filters. The magnetometers are run in DC mode, but the time delay due to the vehicle roll rate (.333 rpm) is 179.4 milliseconds. The RC components of the filter are 1%, thus the actual time delay may vary from the predicted value by as much as 2% or 3.5 milliseconds, well below the specified 10 millisecond requirement.

4.0 Conclusion

The ITF Timing Tests have proven an invaluable contribution to the overall understanding of the GPB timing system. For the ECU mnemonics, the test data provided direct verification of time tagging as specified by T003, 16.6.7: Experimental Control Unit Science Signals Time Tagging. It is therefore concluded that the sampling and time tagging of the Experiment Control Unit temperature and magnetometer signals can be done to within less than 10 ms relative to the 16 f_o rollover counter. The T003, 16.6.7: Experimental Control Unit Science Signals Time Tagging requirement is verified.

5.0 Appendix A: Sample Derivation

For the March 3, 2003 timing test, at a frequency of 184.5 seconds, the TE_Win_3_aST25P monitor had the following values:

	Vehicle Time Stamp	PE_VehTime_LSB2	PE_VehTimeFrac2	Science Vehicle Time
Initial	48153566.1	135	205	48155783.80
Final	48161566.1	27	205	48170523.80

Where the Science vehicle time is calculated as follows

$$SVT = floor\left(\frac{VehicleTimeStamp}{256}\right) * 256 + PE\_VehTime\_LSB2 + round\left(\frac{PE\_VehTimeFrac2}{25.6}\right)/10$$

Calculate the midpoint: Midpoint = (48155783.80+ 48170523.80)/2 = 48163153.80

Phase Analysis returns the following results:

Measured Phase at $Midpoint_{LS\_Phase}$ = -90.6910 degrees

Measured Phase at $Midpoint_{FFT\_Phase}$ = -90.6912 degrees

Since the sine wave generator's zero crossing output is input into the PPS input of the SRE, the SQ_PPSv16F_Time mnemonic captures the value of the 31 bit rollover counter when the zero crossing pulse occurs. We can therefore calculate the time offset from the nearest 10 Hz time. For the vehicle time of 48163158.5 the following telemetry is seen:

Mnemonic	SQ_SciVehTimByt	SQ_SciVehTime8	SQ_PPSv16F_Time	SQ_Sci10HzTime
Value	48163158 (S)	0.5 (10^{th} of seconds)	227678534 counts	228869765 counts

(SQ_PPSv16F_Time - SQ_Sci10HzTime = 227678534 - 228869765 = -1191231 counts

Convert this value to a time by dividing by$16f_0$ = -1191231 / 16368000 = -0.07277804252199 seconds

The precise zero crossing time is therefore SQ_SciVehTimByt + SQ_SciVehTime8 + Time Offset

48163158 + 0.5 - 0.07277804252199 = 48163158.42722195 seconds

The midpoint – the zero crossing is 48163153.80 – 48163158.42722195 = -4.62722195 seconds

Finally, the time delay is calculated as follows:

$$Delay = \left(\left(\left(\left(\left(\frac{MP - ZC}{Period}\right) * 360\right) - 90\right) - Phase\right) / 360\right) * Period\right) * 1000$$

Where 90 is subtracted because the Sine Wave is actually a Cosine.

$$Delay = \left(\left(\left(\left(\left(\frac{-4.62722195}{184.5}\right) * 360\right) - 90\right) - -90.6910207\right) / 360\right) * 184.5\right) * 1000 = 4.436\ mS$$

Similarly

$$Delay = \left(\left(\left(\left(\left(\frac{-4.62722195}{184.5}\right) * 360\right) - 90\right) - 90.6911526\right) / 360\right) * 184.5\right) * 1000 = 4.368\ mS$$

Conclusion: Time delay values can be calculated with a high degree of precision.

6.0 Appendix A: Least Square Phase Analysis Code

```
% LS Phase Analysis
% This routine determines the phase of an input sine wave by Least Squares Method.

close all;
dataset = input('Enter dataset to be used: ');
Sig = dataset;
stSig = input('Input Dataset Plot Name: ','s');
disp([stSig])

Period = input('Time Period of Data Set (1 = 15.3750, 2 = 30.7500, 3 = 92.2501, 4 = 184.5002): ');
  if Period == 1; Tp  =  15.3750; end
  if Period == 2; Tp  =  30.7500; end
  if Period == 3; Tp  =  92.2501; end
  if Period == 4; Tp  = 184.5002; end

Sampling = input('Enter Sampling Rate (1 = .1, 2 = .02, 3 = .05): ')
  if Sampling == 1; fs = 0.1;  end
  if Sampling == 2; fs = 0.02; end
  if Sampling == 3; fs = 0.05; end

N = length(Sig);
mode   = 1;                                   % mode: 0-reference; 1-real data.
Ts     = 1/fs;
phase  = 24*pi/180;

Na1   = 1;  Na2   = N; t_ref = (N-1)/2*Ts; dt    = Ts*[0:Na2-Na1]' - t_ref; Ta    = Ts*(Na2-Na1);

if ( mode==0 );   a      = cos( 2*pi/Tp*dt + phase);
else;  a      = Sig;
end

H   = [ cos(2*pi/Tp*dt) -sin(2*pi/Tp*dt) dt/Ta ones(size(dt)) ];
Y   = a(Na1:Na2);
X   = inv(H'*H)*H'*Y;
err = Y-H*X;

ampli_est = sqrt(X(1)*X(1)+X(2)*X(2));
phase_est = atan2(X(2),X(1))*180/pi;
if ( mode == 0 ); phase_err = phase_est - phase*180/pi; end

stphase_est = num2str(phase_est,9); stampli_est = num2str(ampli_est,9);
```

```
disp('*******************************');
disp('Least Square Method');
disp(['Signal = ',stSig, ', Sampling Rate = ', num2str(fs), ', Period = ', num2str(Tp)]);
disp(['phase_est = ', stphase_est]);
disp('*******************************');

figure(1)
subplot(4,1,1),plot(a); ylabel([stSig]); title('Least Square Fitting');
subplot(4,1,2),plot(Y); ylabel('Data for LS')
subplot(4,1,3),plot(err); ylabel('Residuals')
subplot(4,1,4); axis([0,100,0,100]); axis off
text(30,70,sprintf(['Estimated Amplitude: ',stampli_est]));
text(30,30,sprintf(['Estimated Phase: ',stphase_est]));
figure(1); set(gcf,'PaperOrientation','landscape'); set(gcf,'PaperPosition',[0.25 0.25 10.5 8]);
print(figure(1));
SAVEAS(figure(time), stSig,'fig')

return;
```

7.0 Appendix C: FFT Phase Analysis Code

```
% FFT Phase Analysis
% This routine determines the phase of an input sine wave by FFT.

close all;
dataset = input('Enter dataset to be used: ');
Sig = dataset;
stSig = input('Input Dataset Plot Name: ','s');
disp([stSig])

Period = input('Time Period of Data Set (1 = 15.3750, 2 = 30.7500, 3 = 92.2501, 4 = 184.5002): ');
   if Period == 1; Tp  =  15.3750; end
   if Period == 2; Tp  =  30.7500; end
   if Period == 3; Tp  =  92.2501; end
   if Period == 4; Tp  = 184.5002; end

Sampling = input('Enter Sampling Rate (1 = .1, 2 = .02, 3 = .05): ')
   if Sampling == 1; fs = 0.1;  end
   if Sampling == 2; fs = 0.02; end
   if Sampling == 3; fs = 0.05; end

N  = length(Sig);

mode   = 1;                                   % mode: 0-reference; 1-real data.
i    = [0:N-1]';  Ts    = 1/fs;  phase  = 24*pi/180;

if ( mode==0 );  a     = cos( 2*pi/Tp*Ts*i + phase);
else  a     = Sig;
end

a0    = a - mean(a);  wn    = 4;  w     = sin(pi*(i+0.5)/N).^wn;  s     = a0.*w;

f  = fft(s);
f_est    = 1/Tp;
ix_est   = round(N*f_est/fs) + 1;
amp2_est = real(f(ix_est))^2 + imag(f(ix_est))^2;
for (j=ix_est-10:ix_est+10);
  amp2 = real(f(j))^2 + imag(f(j))^2;
  if ( amp2>amp2_est);
    amp2_est = amp2;
    ix_est   = j;
  end
end
```

```
ix = ix_est;
im = ix - 1;
ip = ix + 1;
rm = abs(f(im))/abs(f(ix));
rp = abs(f(ip))/abs(f(ix));

if ( wn == 2 );
  xpp = (2*rp-1)/(1+rp);
  xpm = (1-2*rm)/(1+rm);
end
if ( wn == 4 );
  xpp = (3*rp-2)/(1+rp);
  xpm = (2-3*rm)/(1+rm);
end

xp     = (xpp+xpm)/2;
freq   = ( ix - 1 + xp )*fs/N;
est_p  = 1/freq;
if ( mode == 0 );
  est_err = est_p - Tp ;
end

signx     = (-1)^(ix-1);
signm      = -signx;
signp     = -signx;
phasem     = atan2( signm*imag(f(im)), signm*real(f(im)))*180/pi - (im-1)*180/N;
phasex     = atan2( signx*imag(f(ix)), signx*real(f(ix)))*180/pi - (ix-1)*180/N;
phasep     = atan2( signp*imag(f(ip)), signp*real(f(ip)))*180/pi - (ip-1)*180/N;
phase_avg  = ( phasem + phasex + phasep )/3;

if ( mode == 0 )
        phase_ref    = mod( (N-1)/2*Ts/Tp*360, 360 ) + phase*180/pi;
        phasem_err   = phasem - phase_ref;
        phasex_err   = phasex - phase_ref;
        phasep_err   = phasep - phase_ref;
  phase_avgerr = phase_avg - phase_ref;
  disp('phase_ref = ', num2str(phase_ref,9));
        disp('phasem_err = ', num2str(phasem_err,9));
  disp('phasex_err = ', num2str(phasex_err,9));
  disp('phasep_err = ', num2str(phasep_err,9));
  disp('phase_avgerr = ', num2str(phase_avgerr,9));
end
```

```
stphasem    = num2str(phasem,9);
stphasex    = num2str(phasex,9);
stphasep    = num2str(phasep,9);
stphase_avg = num2str(phase_avg,9);

disp('*******************');
disp('FFT Method');
disp(['Signal = ',stSig, ', Sampling Rate = ', num2str(fs), ', Period = ', num2str(Tp)]);
disp(['Center Lobe Phase = ',stphasex, ', Average Phase = ',stphase_avg, ', Minus Phase = ',stphasem,',
Pos Phase = ',stphasep]);
disp('*******************');


figure(1)
subplot(5,1,1), plot(a);; ylabel([stSig]); title('FFT Square Fitting:  Phase_-Analy1.m');
subplot(5,1,2); axis([0,100,0,100]); axis off;
text(30,85,sprintf(['Phase M: ',stphasem]));
text(30,60,sprintf(['Phase X: ',stphasex]));
text(30,35,sprintf(['Phase P: ',stphasep]));
text(30,10,sprintf(['Average Phase: ',stphase_avg]));
subplot(5,1,3), plot(abs(f));  ylabel('Absolute FFT')
subplot(5,1,4), plot(real(f)); ylabel('Real FFT')
subplot(5,1,5), plot(imag(f)); ylabel('Imaginary FFT')
figure(1); set(gcf,'PaperOrientation','landscape'); set(gcf,'PaperPosition',[0.25 0.25 10.5 8]);
print(figure(1));
SAVEAS(figure(time), stSig,'fig');

return
```