IBEX — 2002

SAP REPORT

F. Mencaraglia

3/17/03

Operating conditions

Before entering in the details it is interesting to have a look at the operating conditions of the SAP. All results described below have been obtained at float altitude; temperature of the system has always been within the specs as we can see in figure 1 inclinometer and shaft encoder temperature have always been higher than the threshold for heater going on.

From the results we notice that the temperature has several spikes. A short section may be seen in figure 2 below.

Note that the spike width is only 1 datum as seen in figures 3 and 4. Temperature data are taken with a sample rate of ~ 0.5 sec; it seems unlikely that temperature changes of 2 ÷ 3 degrees in such a short time, so what we are seeing is probably a problem in the data bus. Similar effects have not been seen for other SAP housekeepings (but something similar is seen for the output port temperature, figure 6) so no further analysis on their origin has been done.

A final check on the consistency of the temperatures has been done by comparing the inclinometer and shaft encoder temperatures (see figure 5). The gap in the distribution is due to a telemetry loss which is also visible in figures 1 and 2.

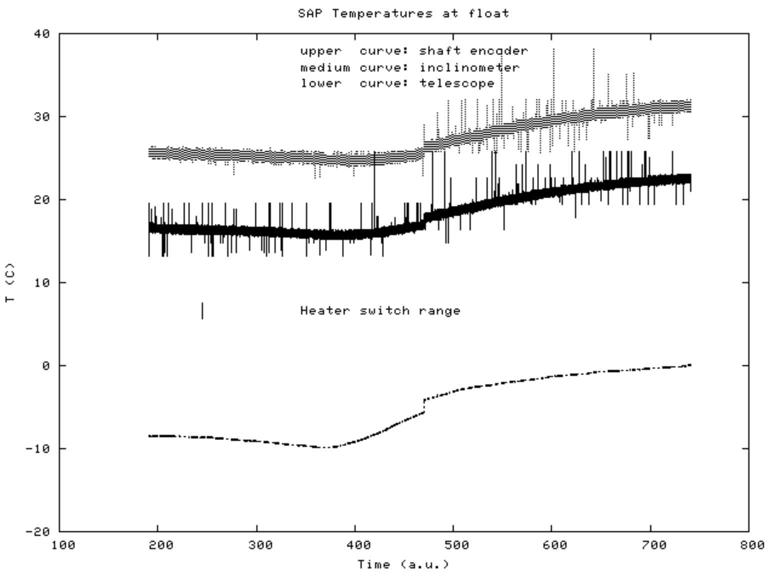


Illustration 1 Temperature of SAP components

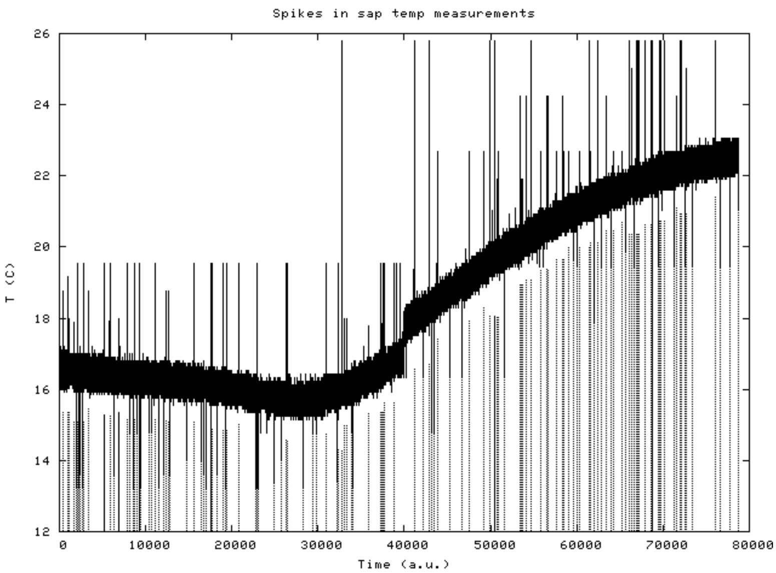


Illustration 2 Spikes in SAP temperature

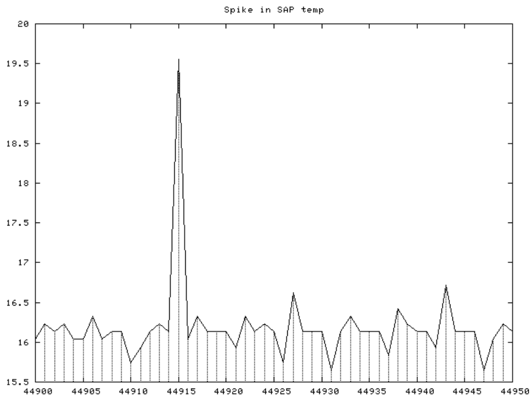


Illustration 3 Example of temp. spike

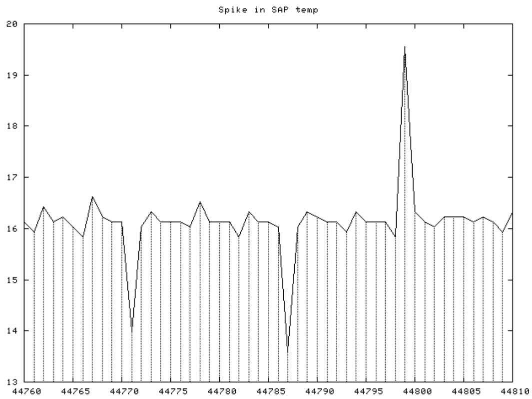


Illustration 4 Example of temp. spike

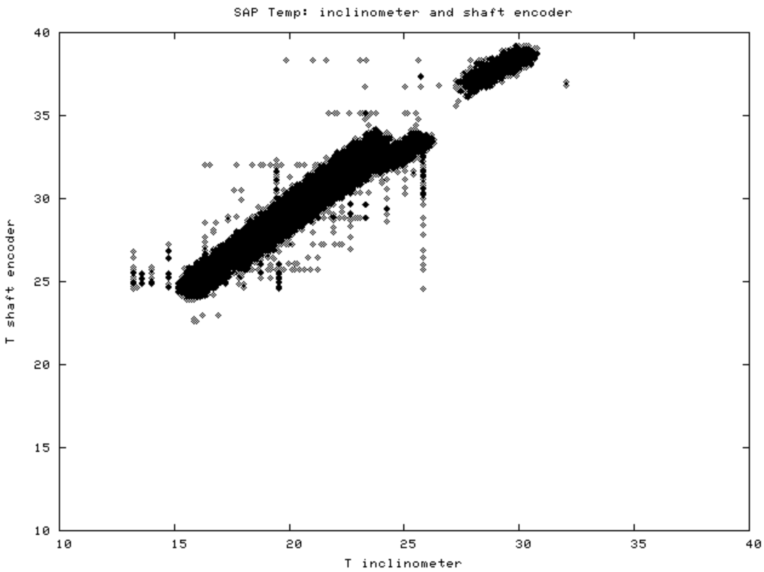


Illustration 5 Comparison of Inclinometer and shaft encoder temperatures

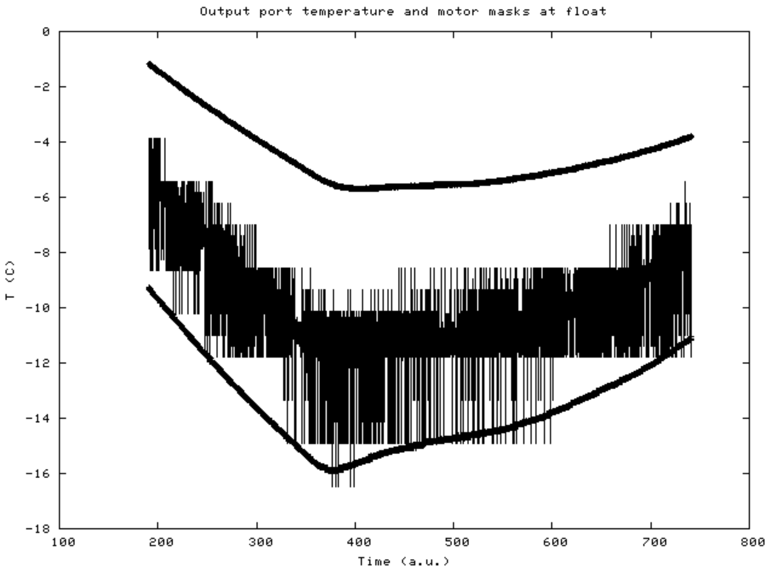


Illustration 6 Spikes in the output port temperature

FOREWORD

The SAP is a system designed to provide an absolute earth oriented (that is referred to gravity vector) pointing. Nominal accuracy is ~0.02^{o} for deviations up to 3^{o} from vertical.
The SAP is divided in two parts, the *fixed* part which is fixed to the gondola, and the *moving* part which in normal flight conditions should keep the horizontal. The moving part can be rotated with respect to the fixed one by mean of a brushless motor **SAP Torquer Motor** and the relative position between the two elements is sensed by a **Linear Variable Differential Transformer**. Mounted on the movable part we find a rate integrating **Gyroscope** which senses the angular displacement with respect to the preset stabilization axis, and an **Inclinometer**[1] which senses the vector sum of the gravity and any lateral acceleration and a **Shaft Angle Encoder**[2] which measures the angle between the actual limb scan direction and a reference direction.
Both inclinometer and gyroscope are required for the stabilization because:

- the inclinometer by itself, when the moving part is locked to horizontal, senses the lateral acceleration of the oscillation providing a false 'correction' signal
- the gyroscope by itself, since it provides an absolute (star oriented) reference, will give origin to a false correction signal when the balloon moves over the earth (correction value depending on the latitude and on the pointing direction[3]).

From the analysis done it results that:

1. The so called manual (libretto istruzioni) does not contain many suggestion on how the proper working of the intrument is to be checked; in particular no suggestion on how the different sap signals are related one to the other is given
2. Conversion of telemetry to engineering values has been done using the file **CS.saf2k2**[i] (prepared by Pascale fom the hskop.h file prepared by Menca)[ii]; conversion factors are taken in part from the electronics drawing, in part from information in the manual, i part from tests done when the SAP arrived in Florence
3. The digitization of analogue channels seems to be insufficient. In many cases the signal is mostly one bit noise as shown in figures 7, 8, 9, 10 [4].

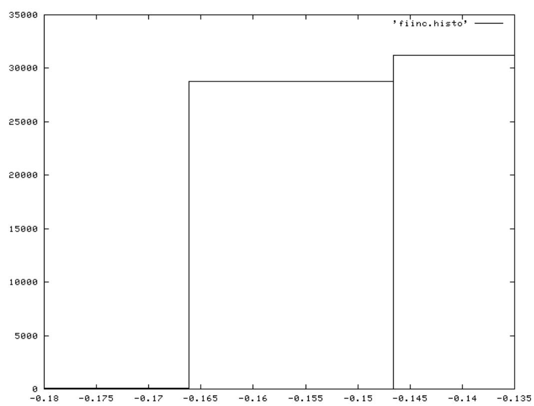


Illustration 7 Filtered inclinometer histogram

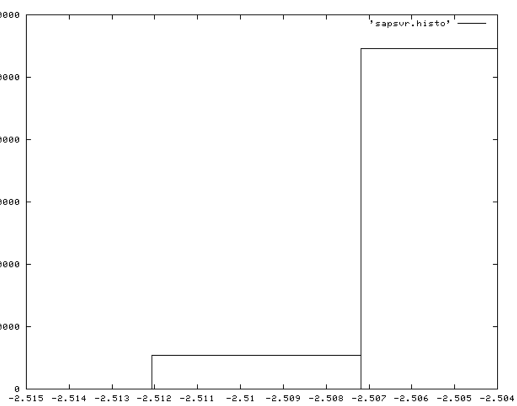


Illustration 8 SAP servo histogram

1 See in directory BALLOON_INST/SAP the data sheet (LSOP)
2 See in directory BALLOON_INST/SAP the data sheet (HMT25)
3 As explained below in our case this effect is not present since the observation direction is towards North
4 See script histo for the creation of histograms

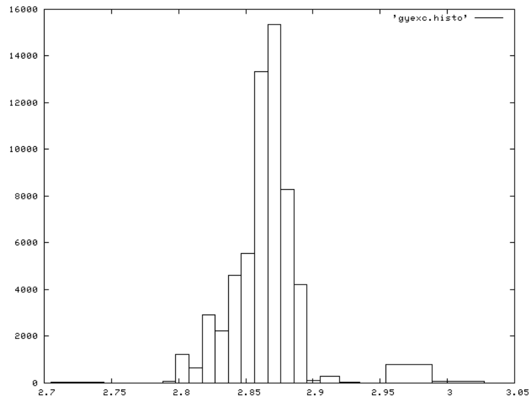


Illustration 9 Gyro excitation histogram

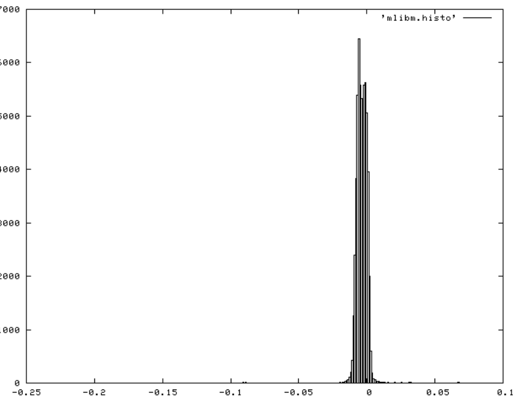


Illustration 10 Mlibm histogram

IMPORTANT NOTICE

All results presented here have been obtained using the data decoded a long time ago. Since then I have lost any idea on how the 'stripout' program works, so no other data can be

SHAFT ANGLE ENCODER

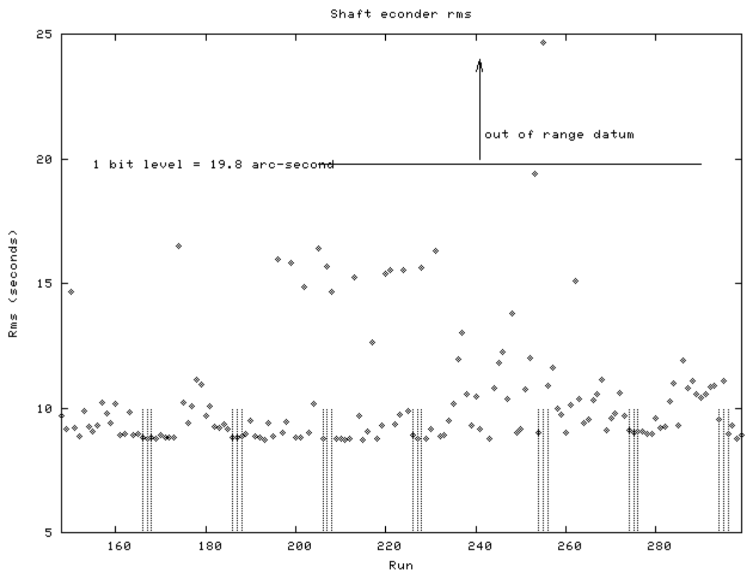


Illustration 11 Shaft Encoder - RMS values

Mounted on the movable part the **Shaft Angle Encoder**[5] provides the angular reference for pointing with respect to the movable part of the SAP. In figure 11I have shown the rms of the shaft encoder readings for the different runs[6]. Note that, with the exception of two runs, the rms is below the one bit level[7], that is we do not expect any problem in pointing from the shaft angle encoder. Vertical lines denote runs with limb direction above horizontal.

A small rms however does not imply that values are consistently small. As a check I have evaluated the range of shaft encoder values for each run (range = uppermost value – lowermost value); result is shown in figure 12 .

Note that there are runs wich have large oscillations; it may be shown that this effect comes from extremity of the run when the limb scan mirror moves from one position to the other and a residual of oscillation is present when the measurement has started (figure 13)

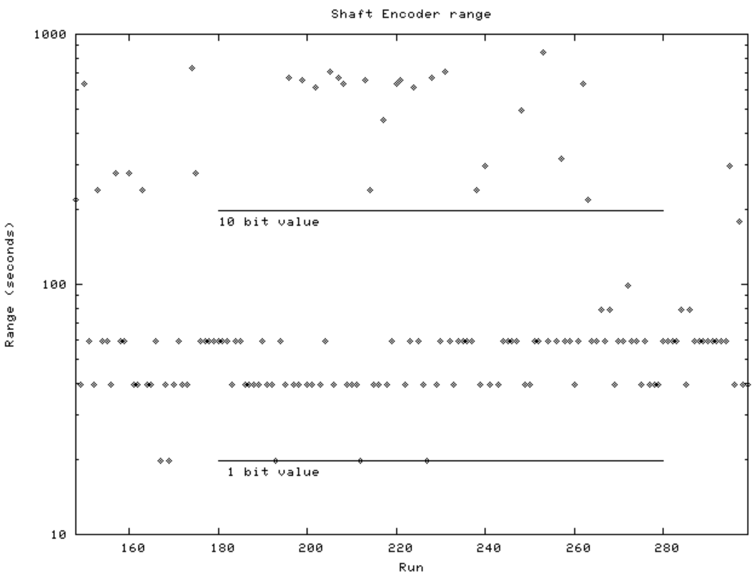
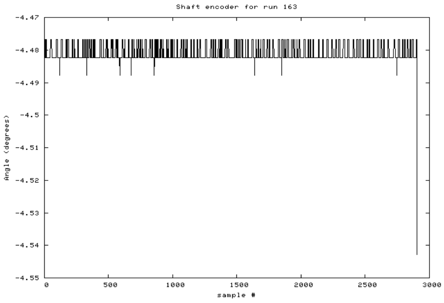


Illustration 12 Shaft Encoder - Reading Range

5 The Shaft Angle Encoder reads angle with a resolution of $\Delta=360*3600/65536=19.77''$

6 For this and similar evaluations see the script sap_rms

7 Quantization noise expected is $\sim\Delta/\sqrt{12} = 5.7''$



Coarse Inclinometers (alias Lucas Inclinometers)

Let us now have a look on how the gondola platform oscillates; we will for this use the coarse inclinometers[8] monted on the instrument baseplane[9]. The two inclinometers senses the oscillations of the axis parallel to the interferometer movement and of the axis normal to it (along the line of view). Results are shown in figure 14[10].

Note the behaviour of inclinometer 1 which clearly shows how the gondola oscillates because of the change in center of mass due to the movement of the interferometer mirror. Note the stability of the other inclinometer[11].

8 See accuDAS20.pdf in BALLOON/INST/SAP for the data sheet; please note that also bubble air inclinometer senses acceleration (see 00NTMJS.pdf)

9 Nominal resolution is $40^{o}/4096*3600=35" \sim 0.01^{o}$ (see the file hskop.h); note however that from close examination of the plot the quantization level seems to be higher ($.05^{o}$). There is here something in the conversion which is unclear. In particular for the inclinometer #2 (which is the one related to the imb scan direction) the results imply shift in pointing which can increase from 1.2' to 6'.

10 Times have been evaluate using the frame frequency as a standard (see file 2002.include)

11 Really it seems that there is a change at time ~205; it would be interesting to check if this corresponds to a ballast drop.

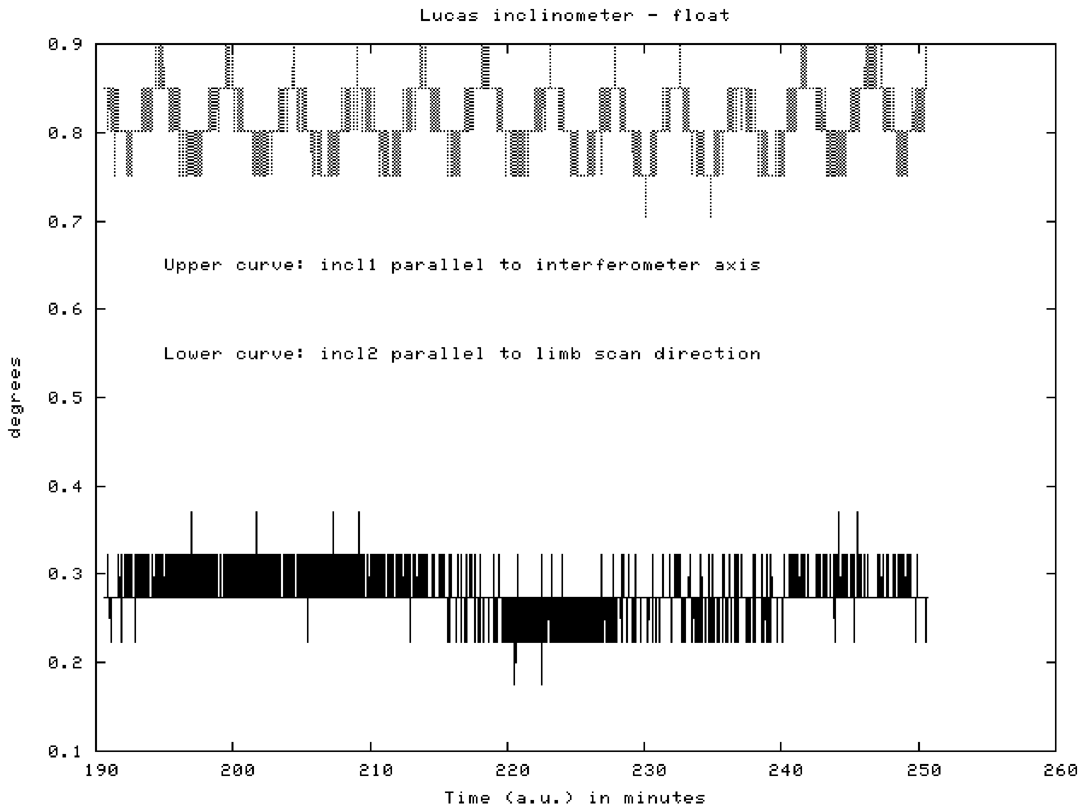


Illustration 14 Coarse Inclinometer - behaviour vs time

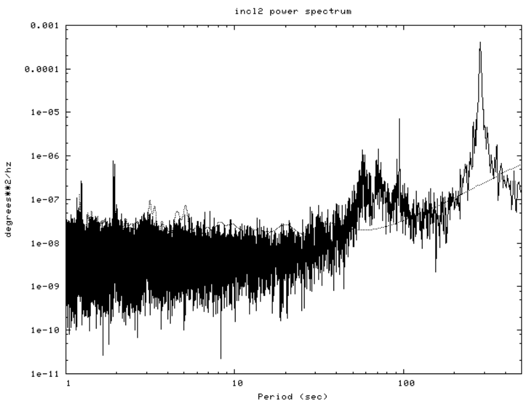


Illustration 15 Coarse inclinometer [1] - Power spectrum

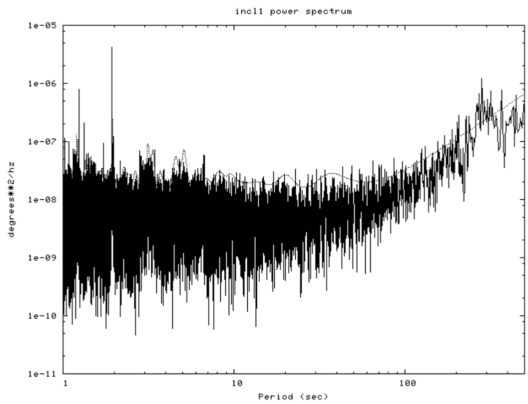


Illustration 16 Coarse inclinometer [2] - Power spectrum

It is interesting to note that inclinometer 1 has a peak for period of ~300 sec (figure 16)[12] which is the time required for two runs (forward + reverse); such a period is not seen for the other inclinometer (figure 15) and this probably means that there is no component of the movement along the transverse axis. Note also that

12 In this figure as well in the following ones the power spectrum has been evaluated using the standard FFT techniques; also reported in the plots (as dotted line) the power spectrum evaluated using maximum spectral entropy which is smoother but apparently sometimes misses some feature.

1. both the inclinometers show a resonance at about 2 seconds[13] (which is in agreement with oscillations of the gondola with respect to pivot, equivalent pendulum length ~1 m)
2. that there is apparently a small broad oscillation with period 50-90 seconds of the instrument optical axis (which does not appear on the other inclinometer). Its origin is not understood but, since it does not interfere with the pointing, we will not discuss it any more[14])
3. no oscillation with periods of ~20 seconds is seen on either inclinometer; this means that the devices do not sense the oscillation due to the gondola with respect to balloon (equivalent pendulum length ~100 m)[15]

It is interesting to note that the time and frequency behaviour of the inclinometer along the instrument axis is similar, but not equal to the one of the current drawn by motor (see figures 18 and 19). This probably means that the difference between the forward / reverse directions is *built* inside the instrument (main peak at low frequency) but there is also a small effect and do not depend on the small angle

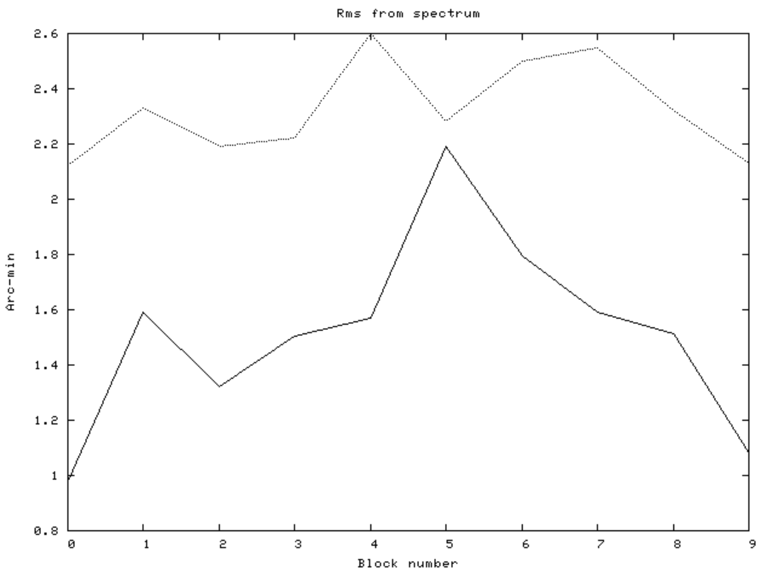


Illustration 17 Rms of inclinometers for different periods

Splitting the data in shorter section (8192 data each, about 1 hour) the rms for the different section is shown in figure 17. Note that rms for the inclinometer along instrument axis is greater than the one along the limb scan direction, but is more constant. The inclinometer along limb scan direction has a peak about 5 hours after float has been reached, and it would be intersting to compare this with the trajectory to unserstand is there is any relation with the change of N/S direction (southmost point reached).

13 Near the threshold which is (-3dB) at 4 sec . (0.25 hz)

14 Long period oscillations have also been described by SAO; it has been suggested that they are of atmospheric origin (lee waves or buoyancy oscillations); in our case the second hypothesis may be discarded (expected resonance is at about 250 seconds). If we need more information we could slit the data in several segments and check if there is a temporal dependence. See also in the text the considerations on figure 17

15 The Lucas inclinometer acts as an air bubble inclinometer; is this consistent with what we observe? According to what I have found the bubble inclinometer should act as an accelerometer.

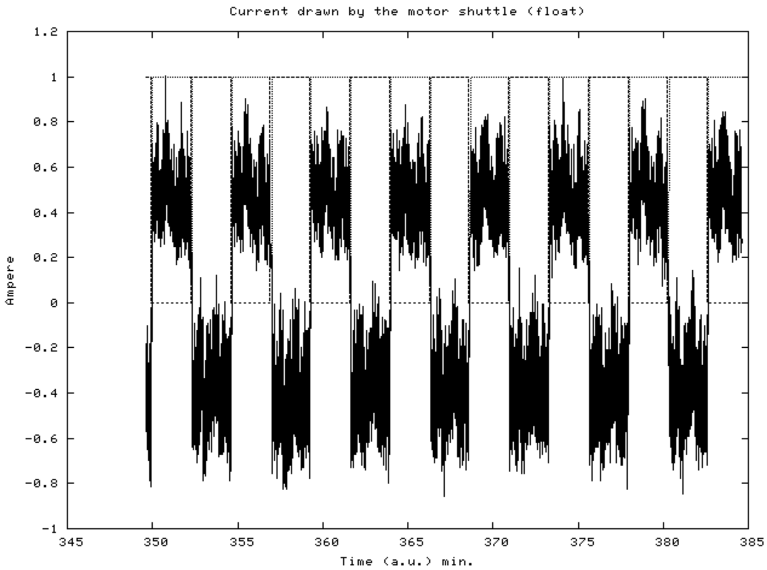


Illustration 18 Current drawn by interferometer motor

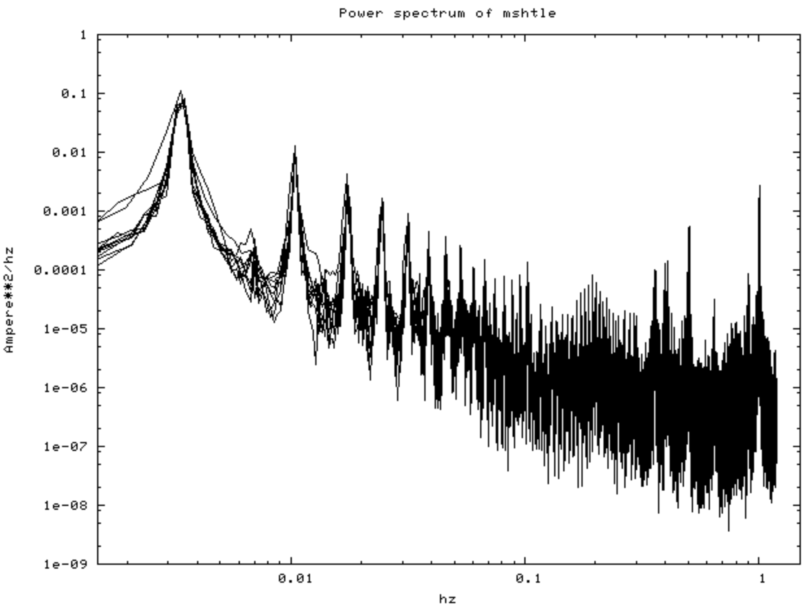


Illustration 19 Power spectrum of current drawn by motor

SAP Inclinometer

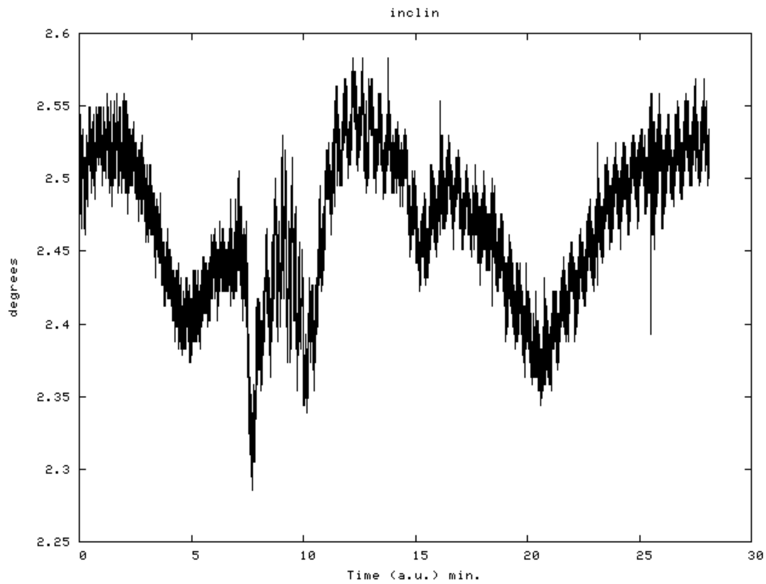


Illustration 20 SAP inclinometer - behaviour vs time

In figure 20 I show an excerpt (25 minutes) of data from the SAP inclinometer taken during float

Note that the oscillations are within 0.1^o from the vertical[16] that is well within the specs (3^o) for proper working.

The power spectrum of the inclinometer signal is shown in figure 21. We can clearly see the oscillation at about 20-25 seconds (pendulum with respect to balloon, see also note 3 in Lucas inclinometer) and the oscillation at 2-2.5 seconds (pendulum with respect to pivot, also seen in Lucas inclinometer). There is also an oscillation with period 8 seconds whose origin I do not understand, but which is probably related to some joint in the balloon train[17],[18]. There may also be (but in this case it is very small) a contribution at the double run (forward + reverse).

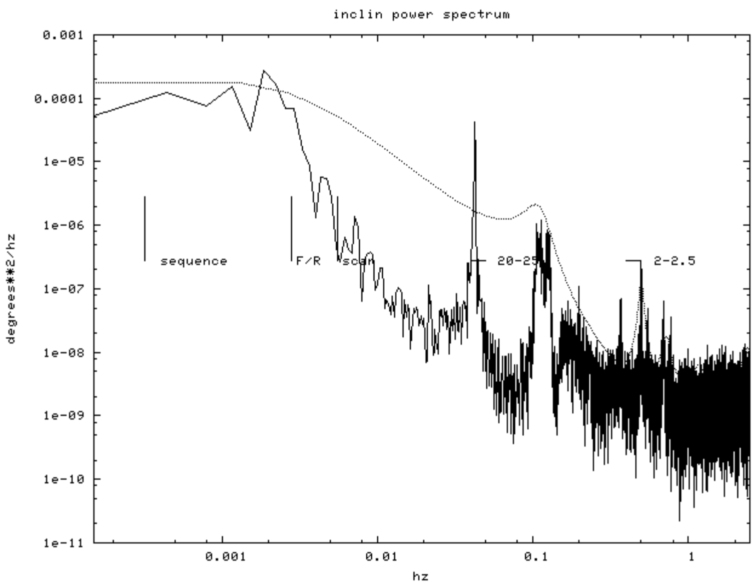


Illustration 21 SAP Inclinometer - Power spectrum

Here I have a problem: how to find out from the spectrum the contribution of the different modes to the oscillations (should this be important). The obvious (and tedious) procedure should probably be to isolate the part of spectrum we are interested in and then make an inverse FFT. For simplicity I think that a raw procedure may be adopted taking advantage of the fact that the square integral is constant; so, for instance, the oscillation at 0.5 hz has a content of $\sim 2.10^{-7} * 0.1 = 2.10^{-8}$ $degree^2$; the corresponding oscillation should therefore be 1.4 10^{-4} degrees ~ 0.5" ; the oscillation at .05 hz has a total content of ~ $10^{-4} * 0.1 = 10^{-5}$ which correspond to 0.003 degrees ~ 0.2'. These crude estimates however seems to be in disagreement with what I see on the plots (see figure 22).

16 The offset level (2.5^o) is the residual of an incomplete conversion from telemetry stream

17 I have asked Andrea if he has the drawing of the balloon train

18 It is not clear to me why this band is seen using the SAP inclinometer and not using the Lucas one. Maybe this is due to the different mechanisms og the two devices?

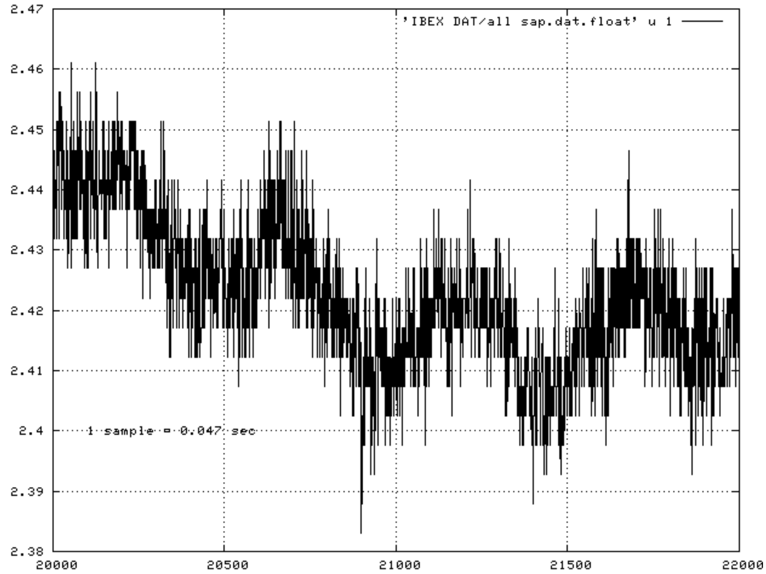


Illustration 22 SAP Inclinometer - short section

The oscillation at 500 samples (23 sec) seems to span over $\mp 0.01^o$ and the one at high frequency seem to span over $\mp 0.005^o$, that is a factor 2-3 greater than the crude estimate. Anyway, even assumig the higher values of 0.01 degrees for the oscillations, the filtering effect of SAP should leave a residual motion for SAP less than 0.001 degrees (4 ÷ 5").

There are two final considerations to make about the inclinometer; as said above the inclinometer senses also lateral accelerations, that is, even in absence of oscillations a non null signal may be produced because of changes in balloon speed along N-S direction (which is our pointing direction). In order to evaluate the possible effects of acceleration I have extracted from the gps data the information relative to the trajectory[19] (see figure 23)

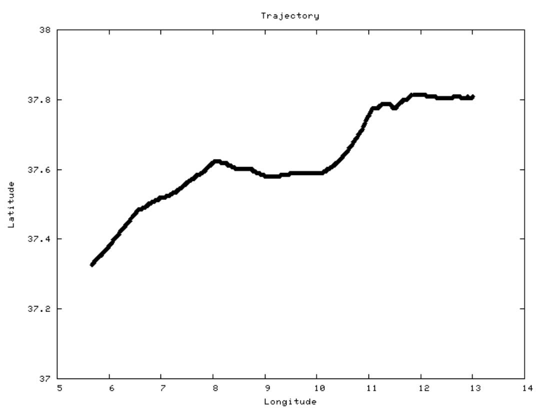


Illustration 23 Balloon Trajectory

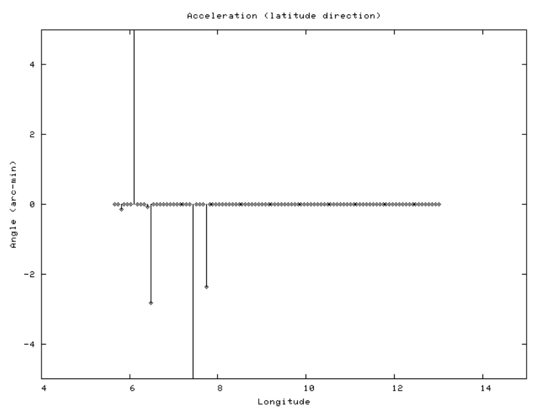


Illustration 24 Balloon N-S acceleration

and then , by differentiating twice I have evaluated the acceleration along latitude (figure 24)
The acceleration is expressed as angular deviation from the local vertical expressed in arc minutes[iii]. Note that with a few exceptions which are probaby outliers (I have not checked by hand all the gps data) the effect of lateral acceleration is well below the arc-min (probably a few arc-sec) and it may be therefore be disregarded. This is an advantage of a summer flight when high altitude winds are more stable than during turnaround. As a comparison during turnaround values as high as 0.7^o for period of hours have been suggested corresponding to accelerations of 0.11 m/sec^2 and change of speed of 400 m/sec over one hour. From the knowledge we have of the wind this was not our case.
Another source of problem may be the rotation of the gondola around its vertical axis. Calling ω the rotational speed the error with respect to true vertical will be given by φ[20] ~ $\omega^2 R/g$; as-

19 See files speed.awk and accel.awk
20 Angle in radiants

suming $R \sim 1m$, $g = 9.81\ m/sec^2$ → $\omega \sim \sqrt{(g\ \varphi)}$ → $T > 118/\sqrt{\phi}$[21]. So accuracy with 1' requires rotation with period larger than 120 seconds. Azimuth pointing system has kept the N-S direction within 1° for all the flight so no problem is expected. Remember that amimuth pointing has affects on the horizontal (see above, pag. 5) because of the movement over earth surface[22]. Pointing towards geographical North eliminates this effect, and since we point towards magnetic North a small effect (time dependent because of the variation in magnetic declination) is expected[23]; the order of magnitude of the error is 1÷3 arc-min.

A more rough but still interesting point of view may be the one obtained by analysis of the zero crossing signal (a pulse which is true whenever the inclinometer passes through the vertical. Results for different time periods are shown in figure 25 (for more clarity they are shown shifted in frequency). Same results also shown in figures 26 and 27). There is clearly a main oscillation at about 0.5 hz, the strength of which changes during the flight.

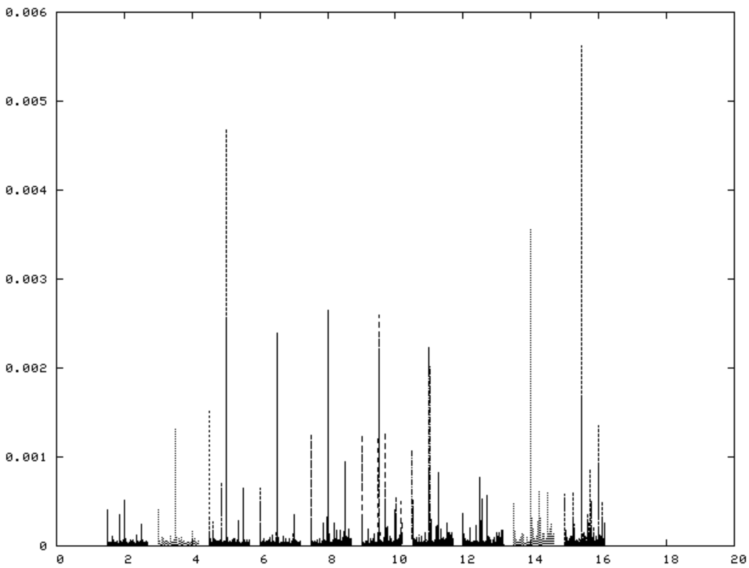
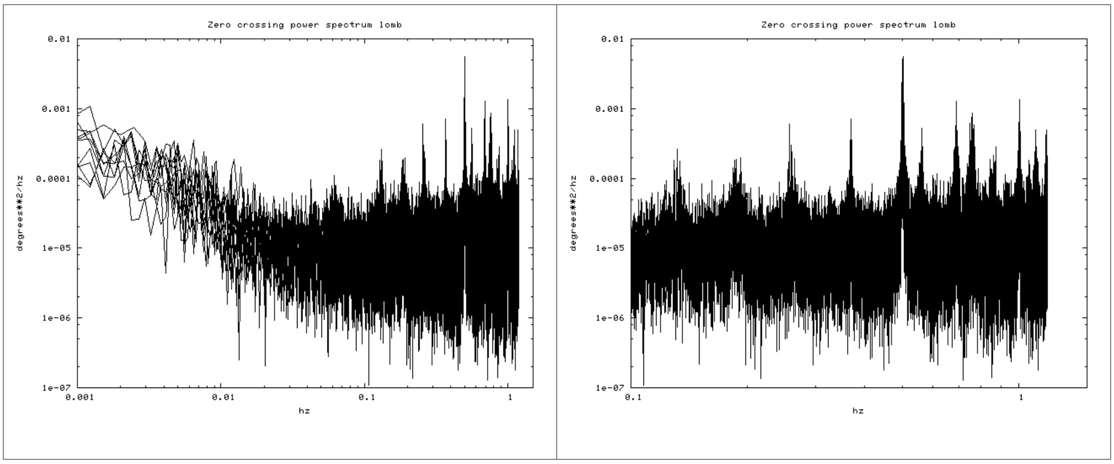


Illustration 25 Power spectra of vertical signal



21 Angle in minutes
22 See also BALLOON_INST/correzione_terra
23 See in BALLOON_INST/documenti lancio an programs in Lavoro/model geophysics

LVDT

In normal conditions the LVDT is out of the control chain, so it acts only as a sensor of the displacement between the movable and the fixed part of SAP. In the simplified case of simple pendulum oscillation the LVDT signal will therefore closely mimic the inclinometer signal.

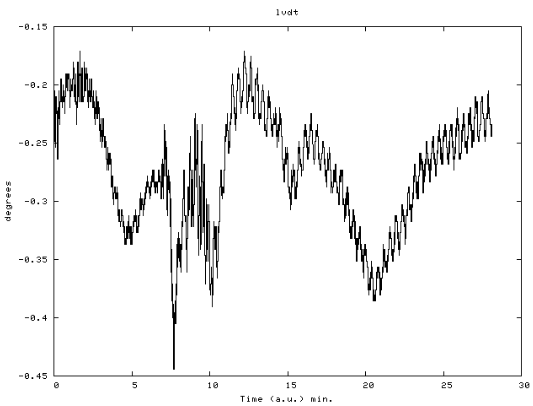


Illustration 26 SAP LVDT - behaviour vs time

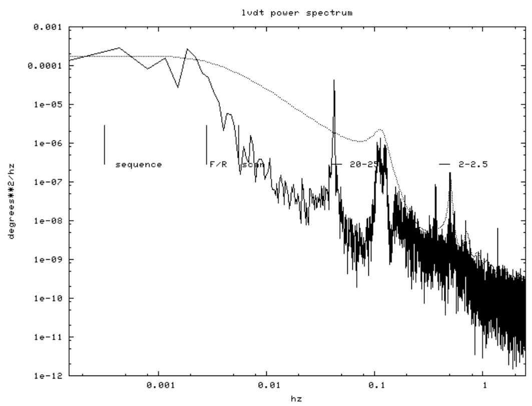


Illustration 27 SAP LVDT - Power spectrum

This is clearly apparent from figures 26 and 27. A better way of seeing the correlation between inclinometer and LVDT is shown in figure 28 where the LVDT values are plotted as a funcion of inclinometer readings.

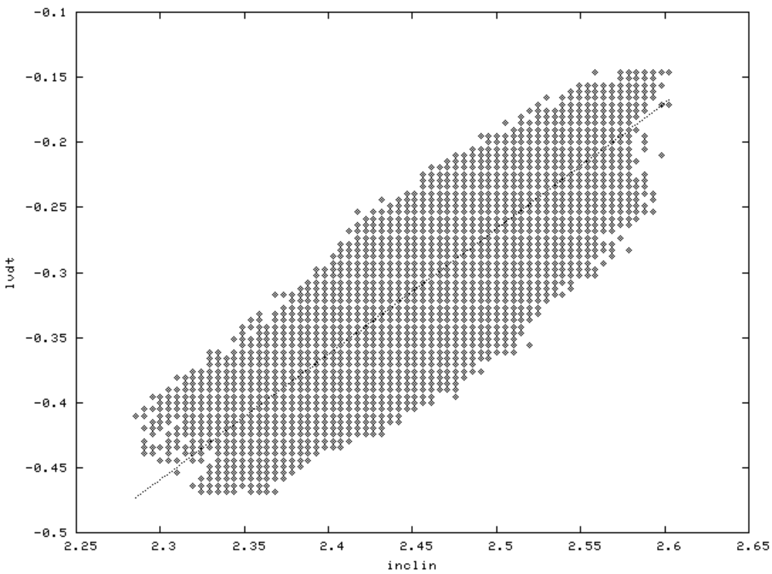


Illustration 28 LVDT vs Inclinometer readings

The linear best fit (using all data recorded during float) is

LVDT = (0.964 +/- 0.001) * Inclinometer – (2.675 +/- 0.002)

with a residual χ-square of 0.0009.

A similar result may be found if we compare the values averaged over single runs (figure 29). The two best fits have been found by fitting inclinometer vs lvdt and lvdt vs inclinometer; the error in dependent variable has been used as a weight.

variance of residuals (reduced chisquare) = WSSR/ndf : 5.45743

Final set of parameters		Asymptotic Standard Error	
=======================		=========================	
a	= 1.06102	+/- 0.03572	(3.367%)
b	= -2.92804	+/- 0.08796	(3.004%)

variance of residuals (reduced chisquare) = WSSR/ndf : 1.98106

Final set of parameters   Asymptotic Standard Error

```
=======================          =========================
A             = 0.711949          +/- 0.02037      (2.862%)
B             = 2.68151           +/- 0.00641      (0.2391%)
```

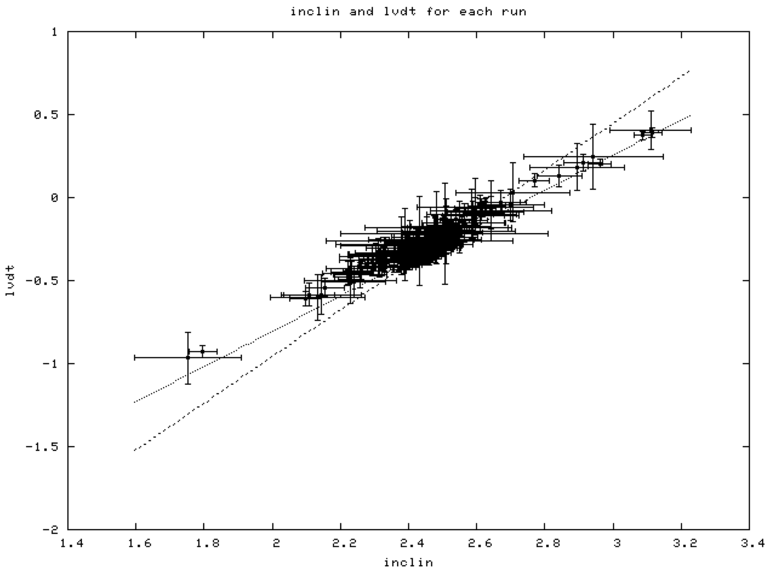


Illustration 29 LVDT vs inclinometer - averaged over runs

```
=======================          =========================

a             = 1.06102            +/- 0.03572      (3.367%)
b             = -2.92804           +/- 0.08796      (3.004%)

variance of residuals (reduced chisquare) = WSSR/ndf : 1.98106

Final set of parameters          Asymptotic Standard Error
=======================          =========================

A             = 0.711949          +/- 0.02037      (2.862%)
B             = 2.68151           +/- 0.00641      (0.2391%)
```

Tilt

There is no special purpose device for measuring the tilt[24] apart for the Lucas inclinometer (see section Coarse Inclinometers (alias Lucas Inclinometers), pag 8). Since the LVDT measures (when the SAP is in normal mode) the sway and tilt, while the inclinometer measures the sway we may try to evaluate the tilt comparing the two data sets.

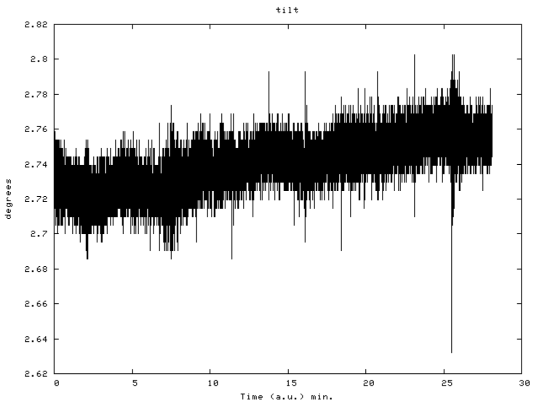


Illustration 30 Tilt - behaviour vs time

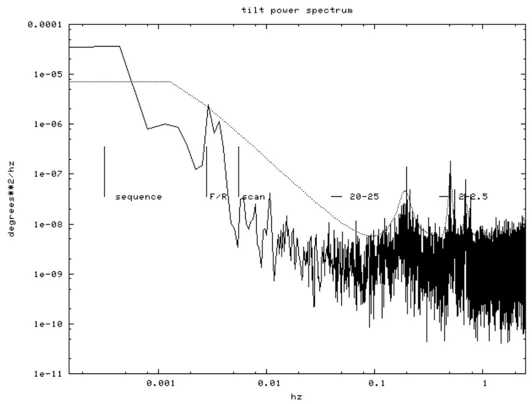


Illustration 31 Tilt - Power spectrum

Results are shown as a function of time in figure 30, and as power spectrum in figure 31. It is interesting to nore that the tilt has a slow change (~1' in about 30 minutes). Such a change was not seen from the Lucas inclinometers (figure 14) because their resolution is not enough[25]. It is interesting to note that the power spectrum of tilt has also a component corresponding to the the double run (forward/reverse) period[26]; this suggests that there is some small unaligment left between the stabilization axis and the mechanical layout of the instrument (carriage movement is not normal with respect to stabilization axis). Note also the small residuals at ~2 and ~8 hz which are there because the acceleration sensed by the inclinometer corrects only the larger sway of the 'long' pendulum.

24 The only tilt we are interested in is the one along the limb scan axis.

25 Again: fron the specs the resolution would be enough, but there are problems with temperature drifts and non linearities. See also note 9

26 With some good willingness some indication of this may also be seen in figure 14

Filtered Inclinometer

The signal from the inclinometer is passed through a low pass filter (f < 0.0032 hz); resulting signal is used to correct for slow changes due either to gyro drift, from initial offsets from horizontal, from movements above earth surface.

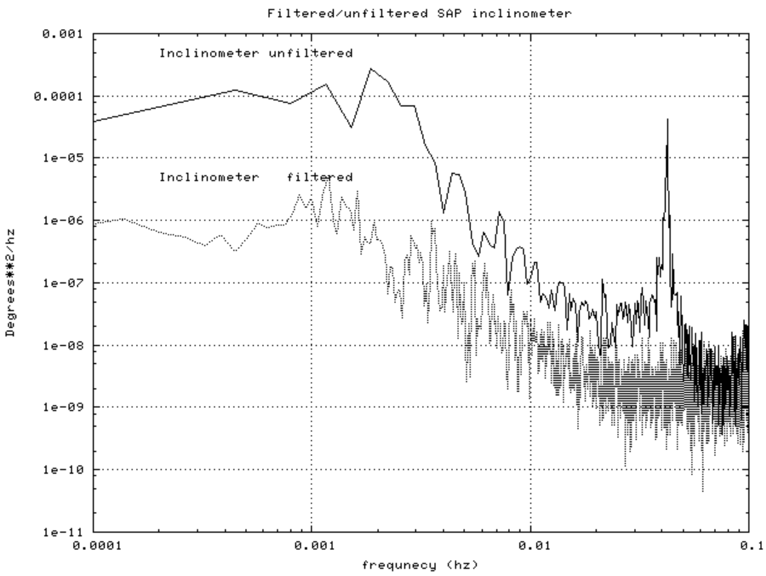


Illustration 32 Filtered/unfiltered inclimometer comparison

A comparison of the the filtered and unfiltered signals is shown in figure 32. Note that, as expected, the resonance at ~0.05 hz (20 sec, balloon pendulum) is damped by a factor ~ 10^{-3} more than expected from the manual. The filtered inclinometer as a function of time is shown in figure 33. Note that it is practically constant (+/-1 bit).

As a further check I have numerically integrated the inclinometer (integration was done using a simple running average kernel); the running average kernel covered a span of 20 minutes (more or less in agreement with the 1000 sec. time constant of the filter). The computed averaged signal has been compared with the filtered signal (no quantization introduced for simplicity) and results are shown in figure 34; it is clear that the two sets of data are in good agreement.

According to SAP manual the average inclinometer (which I think should be the filtered inclinometer) should not be larger that the value read by the gyro pickoff. The comparison has been done (see figure 33); apparently the two values are more or less similar but the filtered inclinometer is mostly one bit noise. So, i think, we may consider the requirement fulfilled.

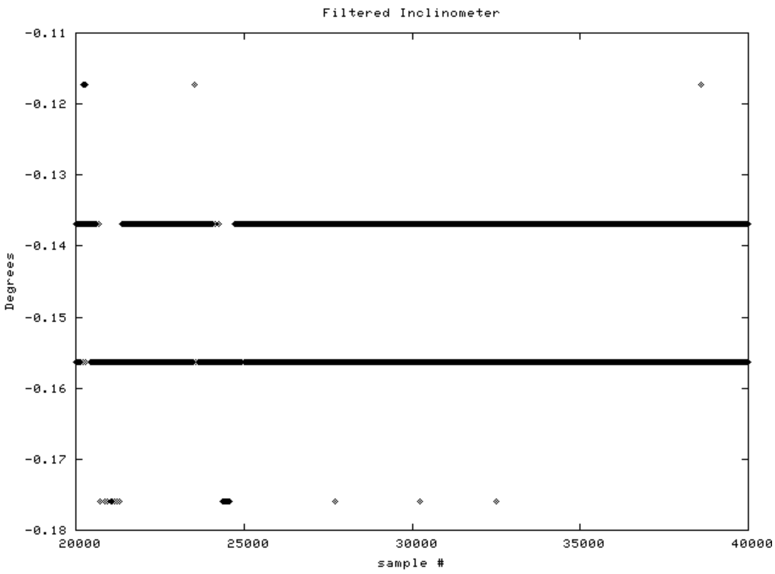


Illustration 33 Filtered Inclinometer vs time

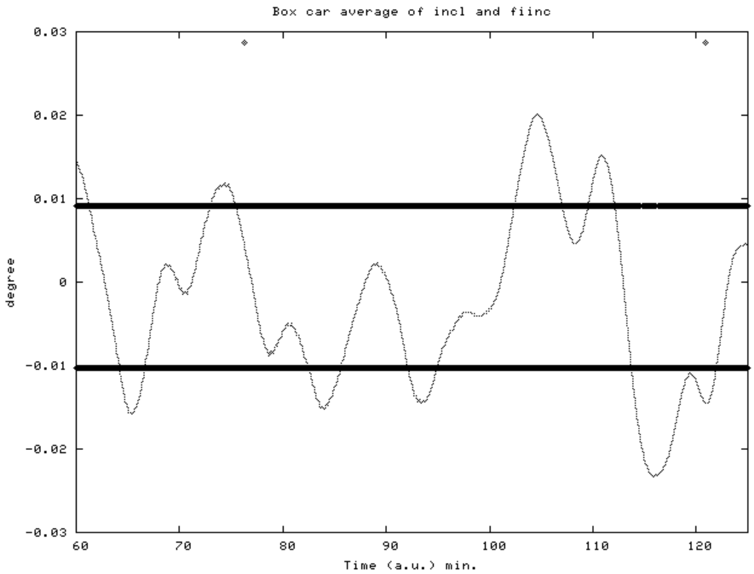


Illustration 34 Filtered Inclinometer and integrated inclinometer

Other SAP values

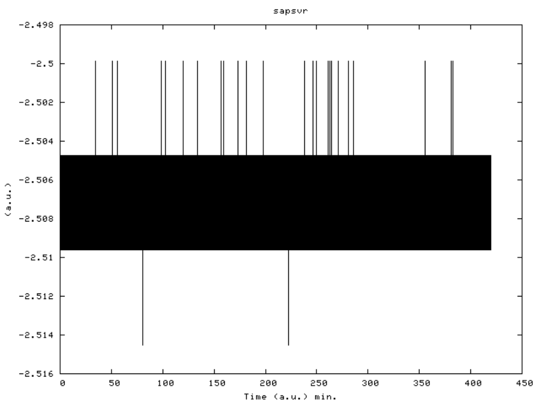


Illustration 35 SAP servo vs time

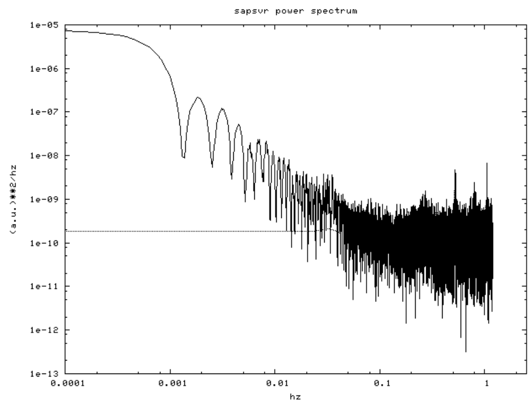


Illustration 36 SAP servo **-** *Power spectrum*

Not all other housekeepings may be handled (and understood) easily ; let us for instance consider the sap servo motor current (Table SAP-2 pag 7 of the SAP manual)[27]. Plotting its value as a function of time (see figure 35) it is evident that the variation are within the quantization limit and that that the spectrum looks like a white noise passed through a low pass filter.

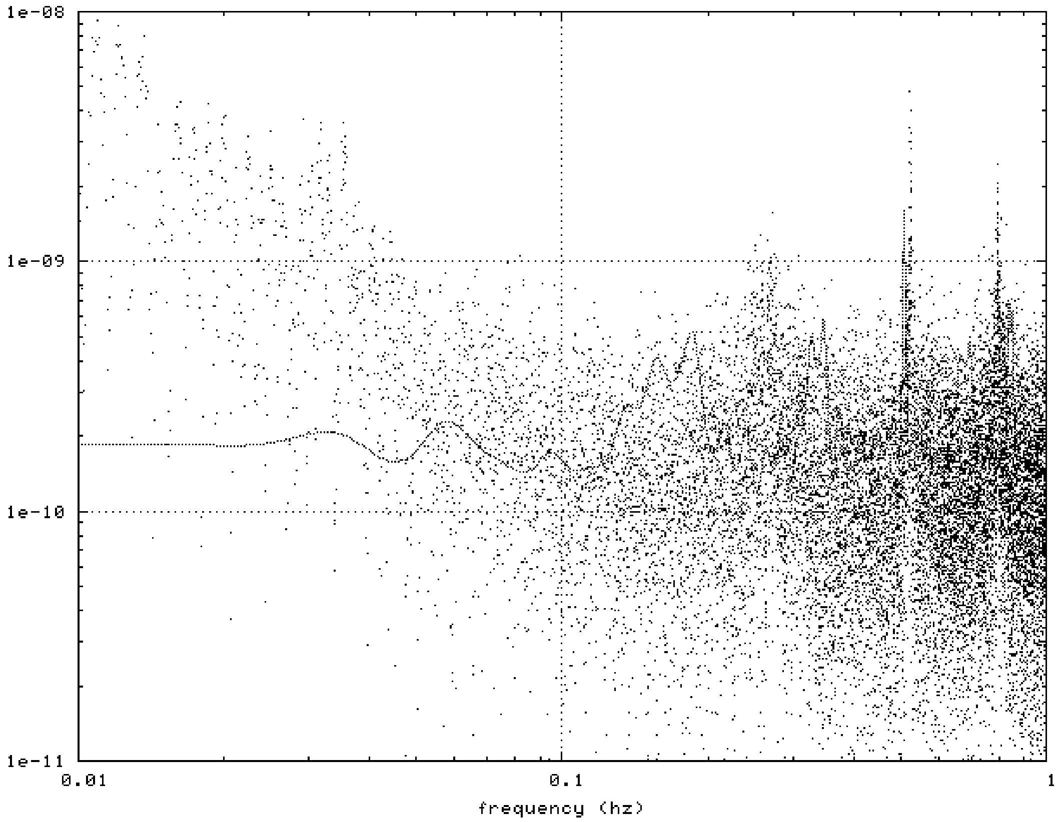


Illustration 37 SAP servo **-** *expanded*

An expanded view is shown in figure 37; this suggests that the power spectrum has some components corresponding to gondola oscillations, but no component is visible for balloon

27 Resolution expected 5A/4096=1.22mA

oscillations.
This is probably because the sap torque is produced by the gyro pickoff (see pag 21) after passing it through a filter

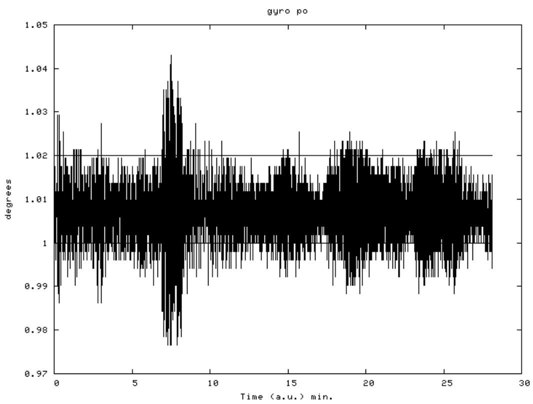


Illustration 38 SAP gyro pickoff vs time

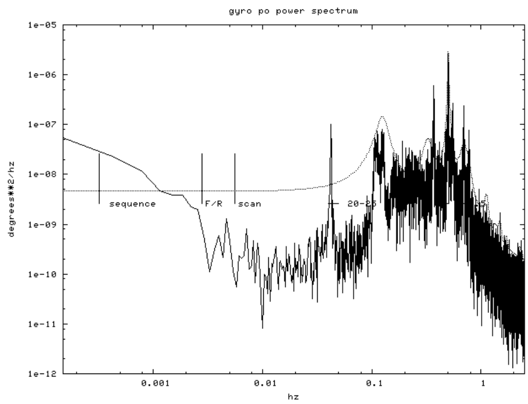


Illustration 39 SAP gyro pickoff - Power spectrum

Similar considerations apply to the gyro pickoff[28] (see figure 38 and 39)

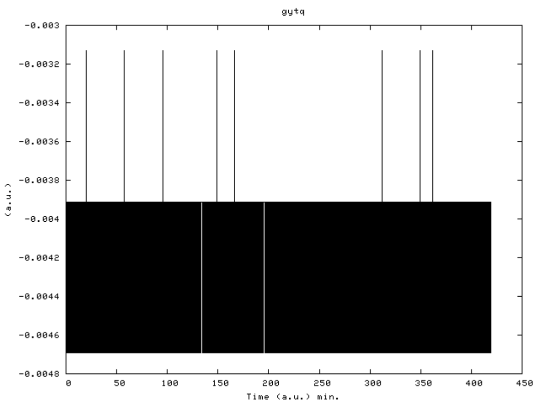


Illustration 40 SAP gyro torque

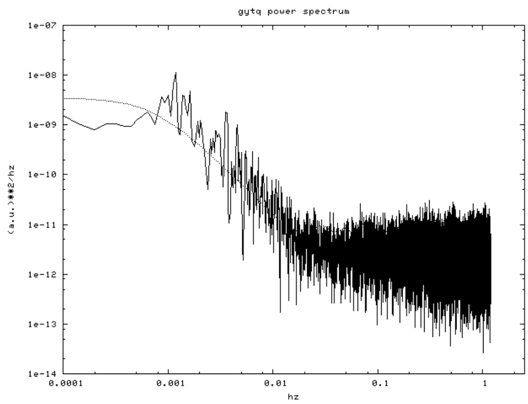


Illustration 41 SAP gyro torque - Power spectrum

As well at to the gyro torquer (see figures 40 and 41), Note that the gyro torque signal (which comes from the outer loop, the slow one) has no high frequency component as it should be[29].

28 This is the residual sway of stabilized SAP see libretto pag 19
29 There is a note from kelly stating that it comes directly from the filtered inclinometer (pag 18); The two power spectra are closely similar.

```
i    #-----------------------------------------------------------------------
#
#
# SAFIRE SCI
#
#
#-----------------------------------------------------------------------
BEGIN
eb90

 FILETIME LINCOM 1 FILEFRAM 0.432 0.0
 FTIME FRAMETIME FILEFRAM 2.31481481

 AID_BIT        BIT         AI          0

 FASTGAIN       SHR         AI          1           0x0003
 CARTPOS        SHR         AI          4           0xffff
 CARTSPD        DIFF        CARTPOS 4096
 SR1                        SHR         R1          1           0xffff
 SR2                        SHR         R2          1           0xffff

 clsang         SHR         NOWCNF      4    0x0fff
 cmaskrel       BIT         NOWCNF      3
 cmaskins       BIT         NOWCNF  2
 cbbrel         BIT   NOWCNF            1
 cbbins     BIT   NOWCNF  0


 nlsang         SHR         NXTCNF      4    0x0fff
 nmaskrel       BIT         NXTCNF      3
 nmaskins       BIT         NXTCNF  2
 nbbrel         BIT   NXTCNF            1
 nbbins     BIT   NXTCNF  0

 cd0sg                      SHR         NOWSLW      0           0x000f
 cd1sg                      SHR         NOWSLW  8    0x000f

 nd0sg                      SHR         NXTSLW      0           0x000f
 nd1sg                      SHR         NXTSLW  8    0x000f

 maxpath        SHR         PTHLSR  4    0x0fff
 phi                        BIT         PTHLSR      2
 phf                        BIT         PTHLSR      1

 BOOTSTRP       BIT         DCFLAGS     0
 STANDBY        BIT         DCFLAGS     1
 STARTUP        BIT         DCFLAGS     2
 ACTIVE         BIT         DCFLAGS     3
 SHUTDOWN       BIT         DCFLAGS     4
 CARMOV         BIT         DCFLAGS     5
 CARDOWN        BIT         DCFLAGS     6
 ZEROSCH        BIT         DCFLAGS 7
 INWINDOW       BIT         DCFLAGS     9
 EMER_BRK       BIT         DCFLAGS     10
 CLOCKW         BIT         DCFLAGS     11
 ACLOCKW        BIT         DCFLAGS     12
 CMDPEND        BIT         DCFLAGS 13
 CRASH                      BIT         DCFLAGS     14
 BADCMD         BIT         DCFLAGS     15

 laserpow       LINCOM      1           LASPOW      -6.8E-4  22

######################################################################
### SAP
```

```
SFR         BIT     SAPSTS    0     # Servo Fast Response
ZCR         BIT     SAPSTS    1     # Zero Crossing
MAN         BIT     SAPSTS  2   # 1 Manual, 0 Auto
NORM        BIT     SAPSTS    3     # 1 Normal, 0 Init
SHTR        BIT     SAPSTS    4     # SHTR    Open
HTR7        BIT     SAPSTS    5     # Heater 7 ON
HTR6        BIT     SAPSTS    6     # Heater 6 ON
HTR5        BIT     SAPSTS    7     # Heater 5 ON
SMOTON      BIT     SAPSTS 10   # SAP MOTOR ON
SGYON       BIT     SAPSTS 11   # SAP Gyro Motor ON
SIHTRON     BIT     SAPSTS 12   # SAP Inclinometer Heater ON
SIHTREN     BIT     SAPSTS 13   # SAP Inclinometer Heater Enable
SEHTRON     BIT     SAPSTS 14   # SAP Encoder Heater ON
SEHTREN     BIT     SAPSTS 15   # SAP Encoder Heater Enable


inclin      LINCOM  1   INCLIN    3.0518E-4   -10.0
lvdt        LINCOM  1   LVDT    3.0518E-4   -12.5
sha_enc     LINCOM  1   SHA_ENC -0.00549316   109.187622
elereg      LINCOM  1   ELEREG    -0.00549316   109.187622
gyro_po     LINCOM  1   GYRO_PO 1.2207E-4   -4
gyro_po_    LINCOM  1   GYRO_PO 1.1719E-3   -38.4
fiinc       LINCOM  1   FIINC   1.2207E-3   -50.0
gycomp      LINCOM  1   GYCOMP  1.8311E-3   -60.0
sapsts      LINCOM  1   SAPSTS    1.0   0.0
gyexc       LINCOM  1   GYEXC   1.5259E-4   -5.0
mgyac       LINCOM  1   MGYAC   1.5259E-4   -5.0

##################################################################
### TEMPERATURES
m1t1        LINCOM  1   M1T1    1.5259e-3   -50.
m1t2        LINCOM  1   M1T2    1.5259e-3   -50.
m2t1        LINCOM  1   M2T1    1.5259e-3   -50.
m2t2        LINCOM  1   M2T2    1.5259e-3   -50.
m1t1k       LINCOM  1   M1T1    1.5259e-3   223.15
m1t2k       LINCOM  1   M1T2    1.5259e-3   223.15
m2t1k       LINCOM  1   M2T1    1.5259e-3   223.15
m2t2k       LINCOM  1   M2T2    1.5259e-3   223.15
mm1t        LINCOM  1   MM1T    3.0518E-3   -124.2
mm2t        LINCOM  1   MM2T    3.0518E-3   -124.2
edect       LINCOM  1   EDECT   1.5259E-3   -24.2
inclt       LINCOM  1   INCT    1.5259E-3   -24.2
shenct      LINCOM  1   SHENCT  1.5259E-3   -24.2
lasert      LINCOM  1   LASERT  6.1040E-5   -2.4
coffint     LINCOM  1   COFFINT 1.5259E-3   -24.2
airt        LINCOM  1   AIRT    1.5259E-3   -24.2
oinstt      LINCOM  1   OINSTT  1.5259E-3   -24.2
mbrkt       LINCOM  1   MBRKT   3.0518E-3   -124.2
telet       LINCOM  1   TELET   3.0518E-3   -124.2
ceut        LINCOM  1   CEUT    3.0518E-3   -124.2
gyrot       LINCOM 1    GYROT   3.8147E-3   -125.0
brtt        LINCOM  1   BRTT    1.0   0.0
stct        LINCOM  1   STCT    1.0   0.0
preampt     LINCOM  1   PREAMPT 1.0   0.0

##################################################################
### CURRENT MONITOR
ceucur      LINCOM  1   CEUCUR  0.15625   -20.0
gytq        LINCOM  1   GYTQ    4.8828E-5   -2.0
sapsvr      LINCOM  1   SAPSVR  3.0518E-4   -12.5
mshtle      LINCOM  1   MSHTLE  6.1035E-4   -20
mlibm       LINCOM  1   MLIBM   6.1035E-5   -2.0
```

```
#################################################################
### VOLTAGE MONITOR
 vm5                          LINCOM  1      VM5     3.0518E-4         -10
 vmp15               LINCOM 1                VMP15          9.1553E-4       -30
 vmm15               LINCOM 1                VMM15          9.1553E-4       -30
 vmp28               LINCOM 1                VMP28          1.2207E-3       -40


#################################################################
###
 pcbias          LINCOM        1             PCBIAS         1.0             0.0
 incl1                        LINCOM  1     INCL1  -3.05176E-3             145.0  #x
 incl2                        LINCOM  1     INCL2   3.05176E-3             -145.0 #y


#################################################################
### Cryo
 d2v1_                        SHR           D02V1          0               0x00ff
 d0v1_                        SHR           D02V1          8               0x00ff
 d2v2_                        SHR           D02V2          0               0x00ff
 d0v2_                        SHR           D02V2          8               0x00ff
 d2t_                         SHR           PREAMPT 0   0x00ff
 d0t_                         SHR           PREAMPT        8               0x00ff
 d2dc_                        SHR           D02BS          0               0x00ff
 d0dc_                        SHR           D02BS          8               0x00ff

 d0v1                         LINCOM        1              d0v1_           0.15625                   -20.0
 d1v1                         LINCOM        1              d2v1_           0.15625                   -20.0
 d0v2                         LINCOM        1              d0v2_           0.15625                   -20.0
 d1v2                         LINCOM        1              d2v1_           0.15625                   -20.0
 d0t                          LINCOM        1              d0t_            7.8125E-2       -10.0
 d1t                          LINCOM        1              d2t_            7.8125E-2       -10.0
 d0dc                         LINCOM        1              d0dc_           7.8125E-2       -10.0
 d1dc                         LINCOM        1              d2dc_           7.8125E-2       -10.0


 DAID_BIT      LINCOM  1      AID_BIT       1              16392
 DINTNUM         LINCOM       1             INTNUM         1               16392
 DCARMOV         LINCOM       1             CARMOV  1   16392


END 0 0
```

```
ii   #ifndef __HSK_OP_H

#define __HSK_OP_H
static char *hsk_op_SccsId = "@(#) hskop.h 1.6 6/14/93";
#include "defs.h"
#include "bitmask.h"
#define OUT_OF_RANGE    9999.
#define ALARM   TRUE
#define ITSOK   FALSE
#define RELEASED        2
#define INSERTED        1
/*-------------------------------------------------------------*/
#define E_INT_NUMBER(value)     (value)
/*-------------------------------------------------------------*/
#define E_MASK_STATUS(value)    ((value)&TWO_B)
#define E_BB_STATUS(value)      ((SHIFT((value),-2))&TWO_B)
/*      12 bits : 0 -> -180 , 4096 -> 180 )     */
/*      revision        18/5/92         sign of angle in 0x0800,
                                        was on          0x0100
```

```
*/
#define ASTEP   (360./4096)
/*
#define E_LS_ANGLE(value)      ((value > 0 ) ? ((value >> 4) * ASTEP)\
                                              : (-1 * (((-value) >> 4) * ASTEP)))
*/
#define E_LS_ANGLE(value)      ((short)((value)&0xfff0)/16 *ASTEP)
/*-----------------------------------------------------------------*/
#ifdef REV_0
#define E_GAIN_BITS(value)     ((((value)&1)<<2)+(((value)&4)>>2)+  \
                               ((value)&2))
#else
#define E_GAIN_BITS(value)     (value)
#endif
#define E_SLOW_GAIN_0(value)    ((value)&FOUR_B)
#define E_SLOW_GAIN_1(value)    ((SHIFT((value),-4))&FOUR_B)
#define E_SLOW_GAIN_2(value)    ((SHIFT((value),-8))&FOUR_B)
#define E_SLOW_GAIN_3(value)    ((SHIFT((value),-12))&FOUR_B)
/*-----------------------------------------------------------------*/
#define E_MAX_PATH(value)       ( SHIFT((value),-4) * 0.032 )
/* laser off -> bit set -> (~(FALSE)) -> TRUE = ALARM   */
#define E_LASER_STATUS(value)   (~(__F_T((value),LASER_BIT)))
/* phi/f off -> bit clear ->  TRUE = ALARM      */
#define E_PHF_STATUS(value)     (__F_T((value),PHF_BIT))
#define E_PHI_STATUS(value)     (__F_T((value),PHI_BIT))
/*-----------------------------------------------------------------*/
/*      following macros return ALARM ( = TRUE) if bit is set
     but for CC e CCW which return ALARM when the bit is clear */
#define E_UNUSED_COMMAND(value) (~(__F_T((value),UN_ACCEPTED_BIT)))
#define E_CRASH(value)          (~(__F_T((value),CRASH_BIT)))
#define E_PENDING_COMMAND(value) (~(__F_T((value),PENDING_BIT)))
#define E_CCW_STOP(value) (__F_T((value),NOT_ANTI_CLOCKWISE_BIT))
#define E_CW_STOP(value)        (__F_T((value),NOT_CLOCKWISE_BIT))
#define E_EM_STOP(value) (~(__F_T((value),EMERGENCY_BREAK_BIT)))
#define E_IN_WIN(value)         (~(__F_T((value),IN_WINDOW_BIT)))
#define E_ZERO_SEARCH(value)    (~(__F_T((value),ZERO_SEARCH_BIT)))
/*      following macro returns direction       */
#define E_SHUTTLE_DIRECT(value) (((value)&DOWN_BIT) ? \
                                                        REVERSE : FORWARD)
/*      following macros return TRUE when the bit is set        */
#define E_IS_MOVING(value)      (~(__F_T((value),MOVING_BIT)))
#define E_SHUT_DOWN(value)      (~(__F_T((value),SHUT_DOWN_BIT)))
#define E_ACTIVE(value)         (~(__F_T((value),ACTIVE_BIT)))
#define E_START_UP(value)       (~(__F_T((value),START_UP_BIT)))
#define E_STAND_BY(value)       (~(__F_T((value),STAND_BY_BIT)))
#define E_BOOTSTRAP(value)      (~(__F_T((value),BOOTSTRAP_BIT)))
/*-----------------------------------------------------------------*/
     /*      CLOCK HAS NO DEFINE     */
/*----------------------------------------------------------------*/
/*      linearly interpolate range 0 +20 A over bit 128 - 256   */
#define E_CEU_CURRENT(value)    LIN_INT(0.0,20.0,H_8_B,H_8_B, \
                                              ((value)&EIGHT_B))
/*-----------------------------------------------------------------*/
#define E_DIGITAL__1(value)     ((value)&EIGHT_B)
/*-----------------------------------------------------------------*/
/*      the macro linearly interpolates for the temperature
range -51.34 , 86.23 assuming a 10 bits ADC with range 0 - 1023
                               (-10 +10 unipolar ADC)
range -50.00 , 50.00 assuming a 12 bits ADC with range 0 - 4095
                               (-10 +10 bipolar ADC)              */
#ifdef REV_0
#define E_MASKX_TX(value)       LIN_INT(-51.34,86.23,F_10_B,0,\
                                              (SHIFT((value),-6)))
#else
```

```
#define E_MASKX_TX(value)       LIN_INT(-50.,50.,F_12_B,0,\
                                        (SHIFT((value),-4)))
#endif
#define E_MASK1_T1(value)       E_MASKX_TX(value)
#define E_MASK1_T2(value)       E_MASKX_TX(value)
#define E_MASK2_T1(value)       E_MASKX_TX(value)
#define E_MASK2_T2(value)       E_MASKX_TX(value)
/*------------------------------------------------------------*/
/*      the macro linearly interpolates for the temperature
range -24.20 , 75.70 assuming a 10 bits ADC with range 0 - 1023
                        (-10 +10 unipolar ADC)
range -24.20 , 75.70 assuming a 12 bits ADC with range 2048 - 4095
                        (-10 +10 bipolar ADC)           */
#define E_AD590_10_T(value)     LIN_INT(-24.2,75.7,F_10_B,0,\
                                        (SHIFT((value),-6)))
#define E_AD590_12_T(value)     LIN_INT(-24.2,75.7,H_12_B,H_12_B,\
                                        (SHIFT((value),-4)))
/*------------------------------------------------------------*/
#ifdef REV_0
#define E_M_MASK1_T(value)      E_AD590_10_T(value)
#define E_M_MASK2_T(value)      E_AD590_10_T(value)
#else
#define E_M_MASK1_T(value)      E_AD590_12_T(value)
#define E_M_MASK2_T(value)      E_AD590_12_T(value)
#endif
/*------------------------------------------------------------*/
#define E_SAP_DECK_T(value)     E_AD590_10_T(value)
/*------------------------------------------------------------*/
#define E_INCLIN_T(value)       E_AD590_10_T(value)
/*------------------------------------------------------------*/
#define E_SHA_ENC_T(value)      E_AD590_10_T((value))
/*------------------------------------------------------------*/
#define E_LASER_T(value)                E_AD590_10_T(value)
/*------------------------------------------------------------*/
#define E_COFFIN_T(value)       E_AD590_10_T(value)
/*------------------------------------------------------------*/
#define E_AIR_T(value)          E_AD590_10_T(value)
/*------------------------------------------------------------*/
#define E_BS_INSTR_T(value)     E_AD590_10_T(value)
/*------------------------------------------------------------*/
#define SWITCH_OFFSET           0x48e8
/*      interpolate 0 - 360 degrees over bit range 0 - 65536    */
#define E_SHA_ENC(value)        (\
                                        (SWITCH_OFFSET-(unsigned)(value))\
                                        *360./65535.)
/*------------------------------------------------------------*/
/*      interpolate 0 - 360 degrees over bit range 0 - 65536    */
#define E_ELEV_REGISTER(value)  (\
                                        (SWITCH_OFFSET-(unsigned)(value))\
                                        *360./65535.)
/*------------------------------------------------------------*/
/*      interpolate -60 - +60 degrees over bit range 0 - 4096   */
#define E_GYRO_COMP(value)      LIN_INT(-60.,60.,F_12_B,0, \
                                        (SHIFT((value),-4)))
/*------------------------------------------------------------*/
/*      following macros return TRUE when bit is set    */
#define E_SAP_FAST(value)       (~(__F_T((value),SAP_FAST_BIT)))
#define E_SAP_ZERO(value)       (~(__F_T((value),SAP_ZERO_BIT)))
#define E_SAP_MANUAL(value)     (~(__F_T((value),SAP_MANUAL_BIT)))
#define E_SAP_NORMAL(value)     (~(__F_T((value),SAP_NORMAL_BIT)))
#define E_SAP_SHTR(value)       (~(__F_T((value),SAP_SHUTTER_ON_BIT)))
#define E_SAP_HTR7(value)       (~(__F_T((value),SAP_HEATER7_ON_BIT)))
#define E_SAP_HTR6(value)       (~(__F_T((value),SAP_HEATER6_ON_BIT)))
#define E_SAP_HTR5(value)       (~(__F_T((value),SAP_HEATER5_ON_BIT)))
```

```
#define E_SAP_MOTOR(value)      (~(__F_T((value),SAP__MOTOR__ON_BIT)))
#define E_SAP_GYROM(value)      (~(__F_T((value),SAP_GYR_MOT_ON_BIT)))
#define E_SAP_INCL_HTR_S(value) (~(__F_T((value),SAP_INC_HTR_ON_BIT)))
#define E_SAP_INCL_HTR_E(value) (~(__F_T((value),SAP_INC_HTR_EN_BIT)))
#define E_SAP_ENC_HTR_S(value) (~(__F_T((value),SAP_ENC_HTR_ON_BIT)))
#define E_SAP_ENC_HTR_E(value) (~(__F_T((value),SAP_ENC_HTR_EN_BIT)))
/*-------------------------------------------------------------*/
/* linearly interpolates in range -5 +5 V over 1024 bits        */
#define E_GYRO_EXCIT(value)     LIN_INT(-5.0,5.0,F_10_B,0, \
                                        (SHIFT((value),-6)))
/*-------------------------------------------------------------*/
/* linearly interpolates in range -125 +125 C over 1024 bits    */
#define E_GYRO_T(value)         LIN_INT(-125.0,125.0,F_10_B,0, \
                                        (SHIFT((value),-6)))
/*-------------------------------------------------------------*/
/* linearly interpolates in range -5 +5 V over 1024 bits        */
#define E_M_GYRO_AC(value)      LIN_INT(-5.0,5.0,F_10_B,0, \
                                        (SHIFT((value),-6)))
/*-------------------------------------------------------------*/
#define E_M_BRAKE_T(value)      E_AD590_12_T(value)
/*-------------------------------------------------------------*/

#define E_BARATRON_T(value)     E_AD590_12_T(value)
/*-------------------------------------------------------------*/
#define E_TELESCOPE_T(value)    E_AD590_12_T(value)
/*-------------------------------------------------------------*/
#define E_SENSOTEC_T(value)     E_AD590_12_T(value)
/*-------------------------------------------------------------*/
#define E_CEU_T(value)          E_AD590_12_T(value)
/*-------------------------------------------------------------*/
/* linearly interpolates in range -10 +10 V over 4096 bits      */
#define OP12(value)             LIN_INT(-10.0,10.0,F_12_B,0, \
                                        (SHIFT((value),-4)))
/*-------------------------------------------------------------*/
/*     zero is moved to offset 2.5 V ; if NORMAL_BIT is set
     datum is converted to 1 unit = 2.5 V , otherwise it
     is converted to 9.6 unit = 2.5 V                   */
#define E_GYRO_PICKOFF(value)   ( E_SAP_NORMAL(value) ? \
                                        ( ( OP12(value) - 2.5 ) / 2.5 ) : \
                                        ( ( OP12(value) - 2.5 ) * (9.6/2.5) ) )
/*-------------------------------------------------------------*/
/*     zero is moved to offset 2.5 V ;
     datum is converted to 0.4 unit = 2.5 V         */
#define E_GYRO_TORQUE(value)    (( OP12(value) - 2.5 ) * (0.4 / 2.5))
/*-------------------------------------------------------------*/
/* linearly interpolates in range -10 +10 V over 4096 bits      */
#define E_INCLIN(value)         ( OP12(value) )
/*-------------------------------------------------------------*/
/*     zero is moved to offset 2.5 V ;
     datum is converted to 4 unit = 1.0 V           */
#define E_FILT_INCLIN(value)    ( ( OP12(value) - 2.5 ) * 4. )
/*-------------------------------------------------------------*/
/*     zero is moved to offset 2.5 V ;
     datum is converted to 1 unit = 1.0 V           */
#define E_LVDT(value)           ( ( OP12(value) - 2.5 ) )
/*-------------------------------------------------------------*/
/* linearly interpolates in range -10 +10 V over 4096 bits      */
#define E_VOLT_MON_5(value)     OP12(value)
/*-------------------------------------------------------------*/
/* linearly interpolates in range -30 +30 V over 4096 bits      */
#define E_VOLT_MON_P15(value)   (OP12(value)*3)
/*-------------------------------------------------------------*/
/* linearly interpolates in range -30 +30 V over 4096 bits      */
```

```
#define E_VOLT_MON_M15(value)   (OP12(value)*3)
/*-------------------------------------------------------------*/
/* linearly interpolates in range -40 +40 V over 4096 bits      */
#define E_VOLT_MON_P28(value)   (OP12(value)*4)
/*-------------------------------------------------------------*/
/*      zero is moved to offset 2.5 V ;
     datum is converted to 1 unit = 1.0 V           */
#define E_SAP_SERVO(value)      ( OP12(value) - 2.5 )
/*-------------------------------------------------------------*/
/*      useful conversion values        */
/*      revision of     TOR_2_MB        was 1014/760
                                        PSI_2_MB        was 69.95
*/
#define TOR_2_MB                (1000./750.065)
#define PSI_2_MB                (1000./14.5038)
#define KPA_2_MB                (6.895*PSI_2_MB)
/*-------------------------------------------------------------*/
/*      revision        to take into account half range work of ADC      */
/*      sensotec has max output at 0xbfe0 (shifted) corresponding
     to 5 V ; for overpressure we output out_of_range value ,
     otherwise we fit linearly between 0 and 1. psi over
     the output range and then convert to mbar              */
#define MAX_SNS                 (unsigned )0xbff0
#define E_SENSOTEC_P(value)     (((value) > MAX_SNS) ? \
                                        OUT_OF_RANGE :\
                                        (LIN_INT(0.,1.,(0x0bff-0x0800),0x0800, \
                                        (SHIFT((value),-4))))*PSI_2_MB)
/*-------------------------------------------------------------*/
/*      sensotec has max output at 0xffe0 (shifted) ; for over-
     pressure we output out_of_range value , otherwise we fit
     linearly between 0 and 10.000 torr over the output range
     and then convert to mbar                           */
#define MAX_BRT                 (unsigned )0xffe0
#define E_BARATRON_P(value)     (((value) > MAX_BRT) ? \
                                        OUT_OF_RANGE :\
                                        (LIN_INT(0.,10.000,(0x0fff-0x0800),0x0800, \
                                        (SHIFT((value),-4))))*TOR_2_MB)
/*-------------------------------------------------------------*/
/* linearly interpolates in range -10 0 Volts over 2048 bits
                        conversion made assuming -10 V @ 0 mbar  and
                        0 V at 1034 mbar */
#define E_LASER_P(value)        ((2048.-((value)>>4))/2048*1034)
/*-------------------------------------------------------------*/
/* linearly interpolates in range -20 +20 A over 4096 bits
     still requires conversion to pressure units            */
#define E_M_SHUTTLE(value)      (2.*OP12(value))
/*-------------------------------------------------------------*/
/* linearly interpolates in range -2 + 2 A over 4096 bits
     still requires conversion to pressure units            */
#define E_M_LIMB_AMP(value)     (0.2*OP12(value))
/*-------------------------------------------------------------*/
/*      fit -20 +20 degrees over range 0 - 4096 */
#define INCLIN_SCALE    10
#define NULL_OUTPUT     4.5
#define FULL_SCALE      (2*NULL_OUTPUT)
#define E_OUR_INCL_X(value)     ((-LIN_INT(-10.,10.,F_12_B,0,\
                                        (SHIFT((value),-4))) + NULL_OUTPUT) \
                                        * INCLIN_SCALE)
#define E_OUR_INCL_Y(value)     ((LIN_INT(-10.,10.,F_12_B,0,\
                                        (SHIFT((value),-4))) - NULL_OUTPUT) \
                                        * INCLIN_SCALE)
/*-------------------------------------------------------------*/
/*      this is bourdon pressure
     range 0. 1000. millibar over half 12 bit range          */
```

```
#define E_SPARE_AN_1(value)     LIN_INT(0.,1000.,H_12_B,H_12_B, \
                                        (SHIFT((value),-4)))
    /*   line added 2 march 1993          */
#define E_SODEME(value)         E_SPARE_AN_1(value)
/*----------------------------------------------------------*/
    /*   modified and added 2 lines 2 march 1993 */
#define E_SPARE_AN_2(value)     (value)
#define E_OPTDSK_PR_A(value)    (LIN_INT(0.,15.,H_12_B,H_12_B, \
                                        (SHIFT((value),-4)))*PSI_2_MB)
#define E_OPTDSK_PR_D(value)    (((value)>>2)&0x3f)
/*----------------------------------------------------------*/
    /*   modified and added 1 line 2 march 1993  */
#define E_SPARE_AN_3(value)     (value)
#define E_MAGNETOMETER(value)   ((1-SHIFT((value),-4)/4096.)*180)
/*----------------------------------------------------------*/
#define E_SPARE_AN_4(value)     OUT_OF_RANGE
/*----------------------------------------------------------*/
/*   preamp : T in K is volt * 100   */
/*   nota
    20 V -> 255 scalini => 1 scalino -> 20/255= .078 V
    la calibrazione da .01 V per grado e cioe' 1 bit -> 8 gradi
*/
#define NIBBLE(value)           (((float)(value)/EIGHT_B)*20.-10.)
#define E_PREAMP_T(value)       (NIBBLE(value)  * 100 - 273.4)
#define E_DET_VOLT(value)       (((float)(value)/EIGHT_B)*40. - 20.)
#define E_PREAMP01_T(value)     E_PREAMP_T(UPPER_BYTE(value))
#define E_PREAMP23_T(value)     E_PREAMP_T(LOWER_BYTE(value))
#define E_PC_BIAS(value)        (0.2*(LIN_INT(-10.,10.,EIGHT_B,0,\
                                        (LOWER_BYTE(value)))) -1.5)
#define E_DET0_VOLT1(value)     E_DET_VOLT(UPPER_BYTE(value))
#define E_DET2_VOLT1(value)     E_DET_VOLT(LOWER_BYTE(value))
#define E_DET0_VOLT2(value)     E_DET_VOLT(UPPER_BYTE(value))
#define E_DET2_VOLT2(value)     E_DET_VOLT(LOWER_BYTE(value))
#define E_DET1_VOLT1(value)     E_DET_VOLT(UPPER_BYTE(value))
#define E_DET3_VOLT1(value)     E_DET_VOLT(LOWER_BYTE(value))
#define E_DET1_VOLT2(value)     E_DET_VOLT(UPPER_BYTE(value))
#define E_DET3_VOLT2(value)     E_DET_VOLT(LOWER_BYTE(value))
#define E_DET0_BIAS(value)      NIBBLE(UPPER_BYTE(value))
#define E_DET2_BIAS(value)      NIBBLE(LOWER_BYTE(value))
#define E_DET1_BIAS(value)      NIBBLE(UPPER_BYTE(value))
#define E_DET3_BIAS(value)      NIBBLE(LOWER_BYTE(value))
/*   offset value 16385 reevaluated from AEU tests made in
    IROE, sec. 2.3.1                    */
#define E_INTERF(value,aid)     (((long)SHIFT((value),-1)-16385)<< \
                                        (3-(SHIFT((aid),-1)&TWO_B)))
#define E_OPTI_STATUS(value)    ((value) & OPTI_WORD)
#define E_OPTI_IS_STDBY(value)  ((value) & OPTI_STDBY)
#define E_OPTI_IS_RECOR(value)  ((value) & OPTI_RECOR)
#define E_OPTI_IS_READ(value)   ((value) & OPTI_READ)
#define E_OPTI_IS_DIAG(value)   ((value) & OPTI_DIAG)
#endif
```

iii We first find speed in arc-sec/sec (samples are given at a rate of 0.1 hz, one per 10 sec) which is the converted to m/sec ; 1 arc-sec corresponds to (6378 * 2 * π)/(360*3600) * 1000; from this we may find the acceleration in m/sec^2 ; deviation in degrees will then be approximatively given by the ratio of this acceleration with gravity acceleration.