Digital Calibration of Cluster EFW

Bjørn Lybekk, University of Oslo
Michael Thomsen

Editor: Bjørn Lybekk

Contents

This page intentionally left blank.

1 Introduction

1.1 Background

The electric field and wave experiment (EFW) of the Cluster project is designed to measure the electric field and plasma density. The instrument will measure the wave and quasi-static electric fields in the spin plane of the four Cluster spacecraft with high time resolution. Voltage/current sweeps can also be made to measure both electron temperature and density. The three magnetic field signals from the search coil sensors are also available in the EFW experiment. The sensors consist of 16 spherical probes, four for each of the four spacecraft. The probes can be operated in pairs to measure the voltage between probes or the voltage between a single probe and the spacecraft can be measureed. The probes can also be used as low impedance probes, Ampere meter, to measure the current from the plasma to the probe. The satellite spin and comparison to other instruments can be used to verify the symmetry between probes.

1.2 Purpose of the Digital Calibrations

The purpose of the EFW digital calibrations is: 1. to partially verify the instrument performance with respect to the user requirements. 2. to partially provide instrument response functions needed in the scientific data analysis. The digital calibrations shall provide all needed information except phase response.

1.3 Scope of the Digital Calibrations

The scope of the digital calibration is to derive the instrument calibrations from the input stimulus to the resulting telemetry values except the phase response. This is schematically illustrated in Figure 1. Tests on unit and sub-unit levels, as well as the phase response, are covered by the analog tests, [Ref. 1]. The digital calibration products constitute a sub-set of the complete EFW instrument tests. The complete digital instrument tests are described in [Ref. 13].

2 Instruments

2.1 Test periods

The instruments were tested at the Cluster test lab at CRPE in Velizy / Paris. The Cluster I and Phoenix is only mentioned in this section. The rest of the document is for Cluster II.

CLUSTER I

Test period	*WEC5*	*WEC1 (deployed)*	*WEC2 (depl.)*	*WEC3 (depl.)*	*WEC4 (depl.)*
930531-930702	FM1	FM01 (5m)	FM02 (5m)	FM03 (45m)	FM04 (45m)
930920-931009	FM2	FM05 (5m)	FM06 (5m)	FM07 (8m)	FM08 (8m)
940106-940114	FM3	FM09 (6m)	FM10 (6m)	FM17 (9m)	FM18 (9m)
940521-940522	FM4	FM13 (6m)	FM14 (6m)	FM11 (9m)	FM12 (9m)
941025-941103	FM5	EM1 (8m)	EM2 (8m)	FM15 (8m)	FM16 (8m)

PHOENIX

Test period	*WEC5*	*WEC1 (deployed)*	*WEC2 (depl.)*	*WEC3 (depl.)*	*WEC4 (depl.)*
961122-961128	Phoenix	FM15 (6m)	FM16 (7m)	EM1 (8m)	EM2 (9m)
961128-961129	Phoenix	EM1 (8m)	EM2 (9m)	FM15 (6m)	FM16 (7m)

CLUSTER II

Test period	*WEC5*	*WEC1 (deployed)*	*WEC2 (depl.)*	*WEC3 (depl.)*	*WEC4 (depl.)*
980823-980907	FM-6	S/N 201 (6m)	S/N 202 (6m)	S/N 203 (5m)	S/N 204 (5m)
981120-981130	FM-7	S/N 205 (8m)	S/N 206 (9m)	S/N 207 (7m)	S/N 208 (8m)
990409-990420	FM-8	S/N 209 (6m)	S/N 210 (6m)	S/N 211 (7m)	S/N 212 (7m)
990814-990823	FM-9	S/N 213 (7m)	S/N 214 (8m)	S/N 215 (8m)	S/N 216 (9m)
991020-991025	FM5	FM #15 (6m)	FM #16 (7m)		

2.2 Equipment

An outline of the equipment used is drawn in Fig. 1.

The equipment consist of the following parts:

- EFW computer: A HP Vectra Pentium II with Solaris 2.5.1 operative system.
- WEC-SUN: A SUN SPARC 1 with Solaris 2.4 operative system.
- Stimuli PC: A 486 PC running DOS, (see [Ref. 7]).
- EFW Stimuli Rack: (see [Ref. 9]): This rack contains:
  - EFW Stimuli Box, (see [Ref. 8]).
  - EFW HP 3325B Synthesizer / Function Generator
  - EFW HP 8904A Multifunction Synthesizer

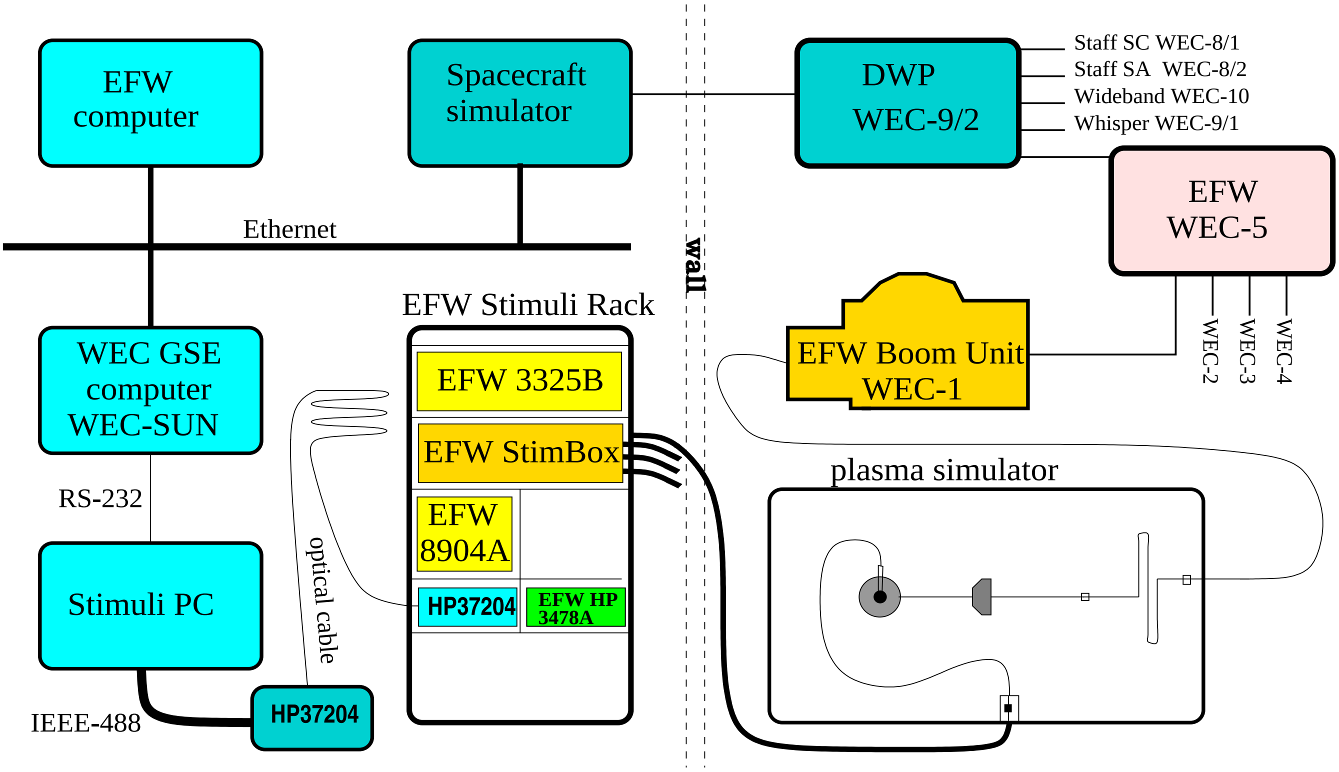

Figure 1: The WEC EFW test setup at CRPE

- EFW HP 3478A Multimeter
- EFW HP 37204(A/B) HP-IB Extender

- EFW HP 37204(A/B) HP-IB Extender and an optical fibre cable.
- The Spacecraft Simulator
- EFW instruments (WEC 1, 2, 3, 4 and 5)
- The four EFW Plasma Simulator Boxes and cables to the Stimuli Rack.

3 Test programs

3.1 Programs run on the EFW computer

On the EFW computer the *gse* program is run.

This program was written by Sten Eckholm (ESTEC), Henrik Berle (ESTEC), Michael Thomsen (ESTEC) and Bjørn Lybekk (UiO), (see [Ref. 1]). The program consists of different modules communicating with each other using interprocess communication.

The gse program consist of the different modules:

- **gse** is the overall server program.

- **tlm** receives telemetry data from the WEC-SUN. This client reads data from the decommutaed data file /users/egse/wec/efw/efw.d on the WEC-SUN. The directory where the incoming data file is located is mounted as an external harddisk at the EFW computer (using NFS).
- **tlc** sends telecommands to the EFW instrument, (via the WEC-SUN).
- **stimrun** controls the Stimuli Rack and reads multimeter data from the HP 3478A.
- **seqrun** executes the test/calibration sequences.

3.2 Test procedures

The test procedures are defined in [Ref. 4]) . The calibration part is the CPT 7 test. This test consist of:

EFIELD TESTS:

Name	*amplitude*	*freq.range*	*plasma sim.*	*quantities*	*sampl.freq*	*filter.freq*
7.1.1	set A	0.3-50 Hz	5M——11pF	V1L V2L V3L V4L	25Hz	10 z
7.1.2	set A	10-900 Hz	5M 11pF	V1M V2M V3M,V4M	450Hz	180Hz
7.1.3	set A	10-900 Hz	5M 11pF	V12M, V34M	450Hz	180Hzdiff
7.1.4	set A	200-47.25 kHz	10k	V1H V2H V3H V4H	9kHz	4kHz
7.1.5	set B	400-36 kHz	10k	V12H V34H	18kHz	8kHz
7.1.6	set C	700-189 kHz	10k	V1U V2U V3U V4U	36kHz	32kHz
7.1.7	set D	500-500 kHz	10k	BPC BP34	25Hz	Bandpass
7.1.8	set E	50 kHz	10k	BPC BP34	25Hz	Bandpass
7.1.9	set F	700-90 kHz	5M 11pF	V1U V2U V3U V4U	36kHz	32kHz

DENSITY TESTS:

Name	*amplitude*	*freq.range*	*plasma sim.*	*quantities*	*sampl.freq*	*filter.freq*
7.2.1	set F	200 - 47.25 kHz	5 pF	V1H V2H V3H V4H	9 kHz	4 kHz
7.2.2	set F	400 - 36 kHz	5 pF	V12H V34H	18 kHz	8 kHz

amplitudes:

set A	1Vp-p	16Vp-p	100Vp-p	30VDC+1Vp-p	60VDC+1Vp-p	
set B	0.3Vp-p	1Vp-p	16Vp-p	100Vp-p	30VDC+1Vp-p	60VDC+1Vp-p
set C	1Vp-p	16Vp-p	30VDC+1Vp-p	60VDC+1Vp-p		
set D	0.15Vp-p	0.3Vp-p	1Vp-p	10Vp-p		
set E	0.05Vp-p	0.1Vp-p	0.15Vp-p	0.2Vp-p	0.25Vp-p	0.3Vp-p
	0.35Vp-p	0.4Vp-p	0.45Vp-p	0.5Vp-p		
set F	1 Vp-p	16 Vp-p				

3.3 Data sampling

The instrument mode (quantity etc.) is specified via a list of telecommands sent to the instrument. The general description of the EFW telecommands is given in [Ref. 3])

Sub-sections 3.3.1 to 3.3.7 give the specific telecommands used during the tests.

3.3.1 V1L, V2L, V3L and V4L at 25 Hz

```
Used in test CPT 7.1.1.

functions: set_V1_2L_25Hz()
           set_V3_4L_25Hz()

TELECOMMANDS        Explanation
----------------------------------------------------------------------
INDEX  34          Set LX
QTY    V1L V2L
QTY    V3L V4L
QTY    BPC B34
QTY    EOL
LXFMT  34

INDEX  38          Set HX
QTY    V1L V2L
QTY    EOL
HXFMT0 38
```

The HX buffer size is 50 words. This will give 25 V1L V2L sample pairs each second. The test has to be run twice. The first time we use "QTY V1L V2L" in the HX list and the second time we use "QTY V3L V4L".

3.3.2 V1M, V2M, V3M and V4M at 450 Hz

```
Used in test CPT 7.1.2.

WEC-SUN must command DWP to set EFW to Tapemode 3.
S/C-sim must be set to samplinq frequency 450 Hz.
Then HXFMT3 list will give the quantities at 450 Hz ,

The instrument names used by GetData() will then be
"V1M", "V2M", "V3M" and "V4M".
```

3.3.3 V12M and V34M at 450 Hz

```
Used in test CPT 7.1.3.

WEC-SUN must command DWP to set EFW to Tapemode 1.
S/C-sim must be set to samplinq frequency 450 Hz.
Then HXFMT1 list will give the quantities at 450 Hz ,
```

The instrument names used by GetData() will then be
"V12M" and "V34M".

3.3.4 V1H, V2H, V3H and V4H at 9 kHz

Used in test CPT 7.1.4.

function: set_V1_V4H_9kHz()

```
TELECOMMANDS          Explanation
---------------------------------------------------------------------
                      Enable playback in Normal mode
INDEX  34             Enable LX playbak
QTY    V1L V2L
QTY    V3L V4L
QTY    BPC B34
QTY    EOL
LXFMT  34

INDEX  38             Enable HX playback
QTY    V12L V34L
QTY    EOL
HXFMTO 38

INDEX  40             define Burst list
BQTY   BV1H BV2H      (p. 16)
BQTY   BV3H BV4H
BQTY   EOL
BFMT   40             Select this burst list
BFREQ  4              0x4; x-ss-fff = 0-00-100; ss:Normal, fff:9kHz, (- is 0)
BCHIRP 0              100% continous sampling
BPAGES 17             ssss.cccc = 0001.0001; 1 search -, 1-64kB collection page
BTRIG  65             0x041; rp.aaaa.tt = 01.0000.01 r:no burst repeat
                         p:automatic playback, aaaa:not adjust new trlgger,
                         tt:immediate trigger
```

3.3.5 V12H and V34H at 18 kHz

Used in test CPT 7.1.5.

function: set_V12_34H_18kHz()

```
TELECOMMANDS          Explanation
---------------------------------------------------------------------
                      Enable playback in Normal mode
INDEX  34             Enable LX playbak
QTY    V1L V2L
QTY    V3L V4L
```

```
QTY    BPC B34
QTY    EOL
LXFMT  34

INDEX  38          Enable HX playback
QTY    V12L V34L
QTY    EOL
HXFMT0 38

INDEX  40          define Burst list
BQTY   BV34H BV12H (p. 16)
BQTY   EOL
BFMT   40          Select this burst list
BFREQ  21          0x15; x-ss-fff == 0-01-101; ss:Split, fff:18kHz, (- is 0)
BCHIRP 0           100% continous sampling
BPAGES 17          ssss.cccc = 0001.0001; 1 search -, 1-64kB collection page
BTRIG  65          0x041; rp.aaaa.tt = 01.0000.01 r:no burst repeat
                      p:automatic playback, aaaa:not adjust new trlgger,
                      tt:immediate trigger
```

3.3.6 V1U, V2U, V3U and V4U at 36 kHz

Used in test CPT 7.1.6.

```
functions: set_V1_2U_36kHz()
           set_V3_4U_36kHz()
```

```
TELECOMMANDS        Explanation
---------------------------------------------------------------------
                   Enable playback in Normal mode
INDEX  34          Enable LX playbak
QTY    V1L V2L
QTY    V3L V4L
QTY    BPC B34
QTY    EOL
LXFMT  34

INDEX  38          Enable HX playback
QTY    V12L V34L
QTY    EOL
HXFMT0 38

INDEX  40          define Burst list
BQTY   BV1U BV2U  (p. 16)
BQTY   EOL
BFMT   40          Select this burst list
BFREQ  54          0x36; x-ss-fff == 0-11-110; ss:Null, fff:36kHz
BCHIRP 0           100% continous sampling
BPAGES 17          ssss.cccc = 0001.0001; 1 search -, 1-64kB collection page
```

```
BTRIG   65          0x041; rp.aaaa.tt = 01.0000.01 r:no burst repeat
                       p:automatic playback, aaaa:not adjust new trlgger,
                       tt:immediate trigger
```

The test has to be run twice. The first time we use "BQTY BV1U BV2U" and the second time we use "BQTY BV3U BV4U".

3.3.7 SCX, SCY, SCZ and BP34 at 9 kHz

```
Used in test CPT 2.5

function: set_SCX_Y_Z_9kHz()

TELECOMMANDS           Explanation
----------------------------------------------------------------------
                    Enable playback in Normal mode
INDEX   34          Enable LX playbak
QTY     V1L V2L
QTY     V3L V4L
QTY     BPC B34
QTY     EOL
LXFMT   34

INDEX   38          Enable HX playback
QTY     V12L V34L
QTY     EOL
HXFMTO 38

INDEX   40          define Burst list
BQTY    BSCX BSCY   (p. 16)
BQTY    BSCZ BBP34
BQTY    EOL
BFMT    40          Select this burst list
BFREQ   4           0x4; x-ss-fff = 0-00-100; ss:Normal, fff:9kHz, (- is 0)
BCHIRP 0            100% continous sampling
BPAGES 17           ssss.cccc = 0001.0001; 1 search -, 1-64kB collection page
BTRIG   65          0x041; rp.aaaa.tt = 01.0000.01 r:no burst repeat
                       p:automatic playback, aaaa:not adjust new trlgger,
                       tt:immediate trigger
```

3.4 Format of data files

From each of the tests (described in 3.2) the data is stored in files on the EFW computer. The data files are stored in the *gse/log/cluster/data* directory. The data files from the CPT 7 tests are named *cpt7.dat.##* . *##* denotes the number 00 to 99. *cpt7.dat.00* is the first file created.

The data file consists of records. A record contains data from one stimuli setting. The first record from CPT 7.1.1 is the 512 V1L and V2L data points recorded when the stimuli was a sine wave with amplitude 1 Vp-p and frequency 0.3 Hz.

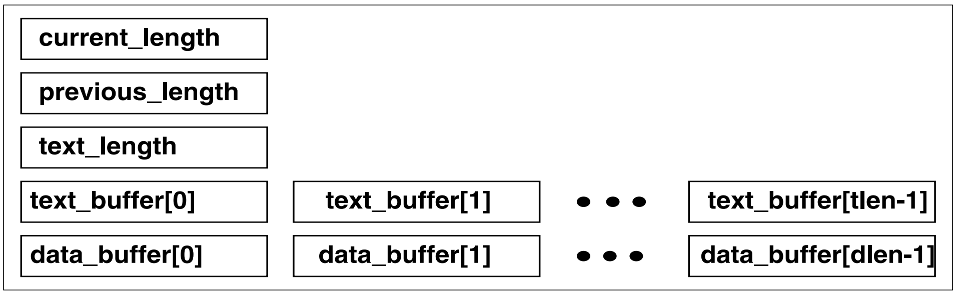

Figure 2: This figure shows the format of one data record. Each box is one integer as defined in Solaris x86. This is denoted as word in the text.

The first word (*current_length*) in each record contains the number of words in this record. The second word (*previous_length*) contains the number of words in the previous record. The first record in the file has *previous_length* equal 0. The last record in the file is an empty record with only three words, (*current_length, previous_length and text_length*). In the last record *current_length* is 0.

The first part of the record is the *text_buffer*. The *text_buffer* contains the "compressed" text header from the text string *text_header[]*. Each word contains two characters. The simple C "compression" algorithm is:

```
tlen = 0;
for (i = 0; i < strlen(text_header)/2; i++)
{
    text_buffer[tlen] = (text_header[i*2] << 8) + text_header[i*2+1] ;
    tlen++;
}
text_length = tlen;
```

The number of words in the *text_buffer[]* is stored in *text_length.* In the text header string the *$* character is used to identify the identifiers (Id.). The format of the text header is:

Id.	*Explanation*	*Example*
$T	section name	$TCPT 7.1.1
$F	EFW filter	$F10 Hz
$R	sampling rate (Hz)	$R25
$N	number of quantities	$N2
$Q	quantity names and number of samples	$QV1L(544)V2L
$v	stimuli amplitude in volt (Vp-p AC)	$v1.0
$o	stimuli amplitude DC offset (V DC)	$o30.0
$f	stimuli frequency (Hz)	$f50.0
$m	measured multimeter value ACV or DCV	$m2.453435ACV

Not all identifiers have to be present. The *$N* number of quantities (*n_quant*) and *$Q* number of samples (*n_samples*) are used to decode the data part of the record.

As an example we want to read the file and store the data in the integer matrix *data[][]*. Let the first index contain the sample number and the second index contain the quantity number. The C algorithm will be:

```
k = 0;
for (i = 0; i < n_quant; i++)
{
    for (j = 0; j < n_samples; j++)
    {
        data[j][i] =  data_buffer[k]; k++;
    }
}
```

The **pbmotif/pbrun** program running at the EFW computers , successfully decodes the data files.

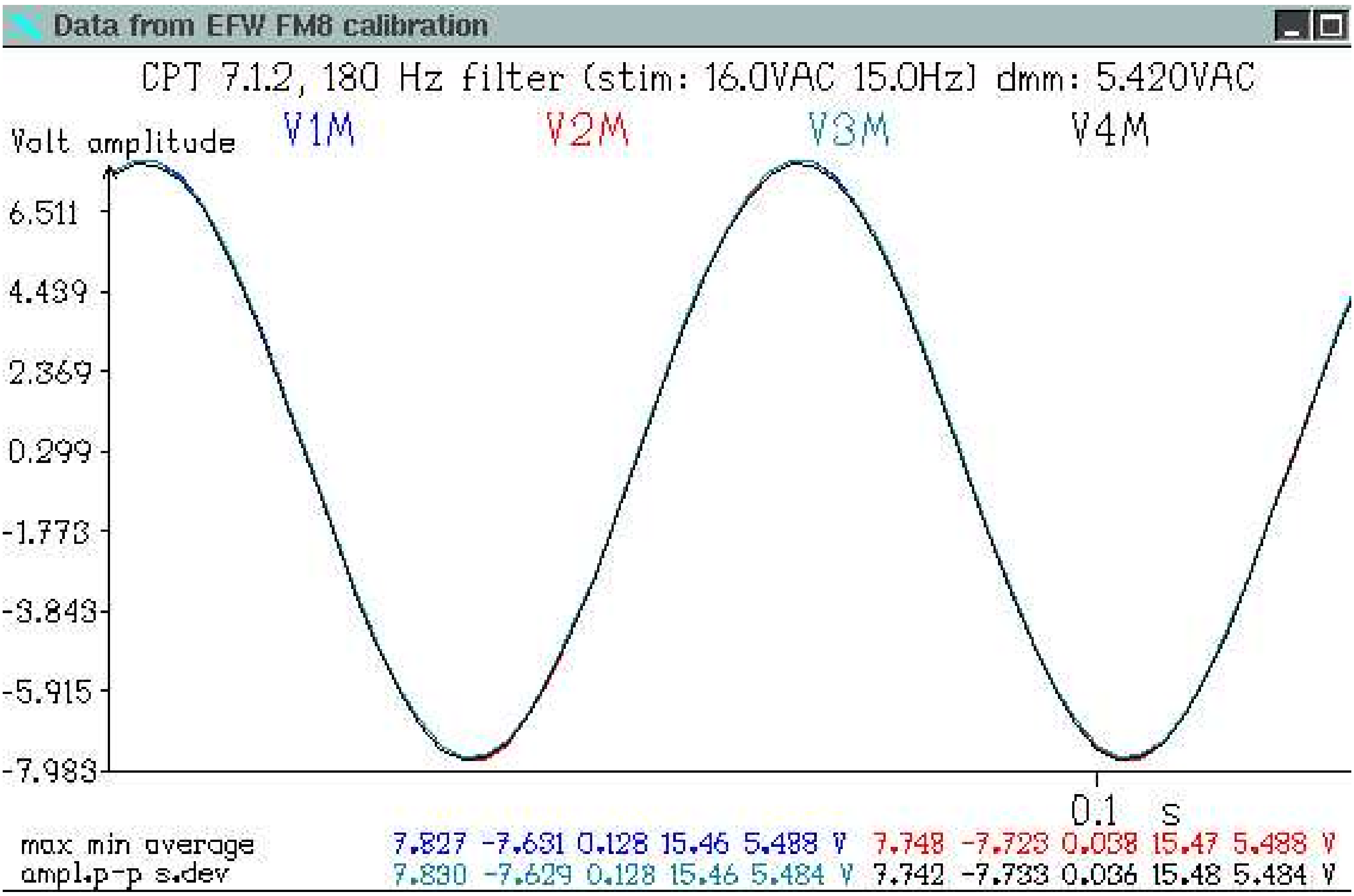

Figure 3: This figure shows the graphic output window of the **pbmotif/pbrun** program.

However, when running at a SUN workstation a rearrangement of the integer is necessary. Let *d* represent an integer read from the data file. The algorithm below is used in order to transfer the integer into the SUN Sparc format:

```
    data_buffer[] =  ((d & 0xFF000000) >> 24)
                  |  ((d & 0x00FF0000) >> 8)
                  |  ((d & 0x0000FF00) << 8)
                  |  ((d & 0x000000FF) << 24);
```

The data files are stored at the SUN workstation, aurora.uio.no. Local backups are stored on CD-ROMS, TR4 and DAT tapes. New backups are taken every week by the University of Oslo Data Centre, (Usit).

4 Analysis

4.1 Creating list files

The rather complex form of the raw data files described in 3.4 is not suited for creating summary plots (see TBW) or EFW calibration files (see TBW). Program **cblist** reads the test data files and creates new ASCII data files. These ASCII files are named *list files*. The *list file* contain one text line from each record (see Figure 2). An example of one line from such a *list* file is:

```
$R: 0  $SECTION: CPT 7.1.1 $N: 2 $QUANT: V1L V2L $RATE: 25 Hz $SAMPLES: 512 \
%WAVE: sine %AMPL: 1.0 Vp-p %OFFSET: 0.0 Vdc %FREQ: 0.3 Hz                  \
%DMM: 40.9 mVac (rms) &MAX[0]: 136 &MAX[1]: 123 &MIN[0]: -336 &MIN[1]: -347 \
&AVG[0]: -104 &AVG[1]: -108 &AMPL[0]: 472 &AMPL[1]: 470                     \
&SDEV[0]: 162 &SDEV[1]: 162 &FFT-FREQ[0]: 0.292969 &FFT-FREQ[1]: 0.292969   \
&PSD-AMPL[0]: 458.121743 &PSD-AMPL[1]: 458.450138
```

The special symbols used in *list file* is:

Symbol	*Explanation*
\	as last character in a line denotes that the line continue
$	mark the start of satellite instrument identifier
%	mark the start of an applied stimuli identifier
&	mark the start of a data identifier
:	mark the end of an identifier.

The format of the *list file* is:

Item	*Explanation*
$R:	record number in data input file
$SECTION:	the name of the test
$N:	number of quantities in this record. This value is referred as N below.
$QUANT:	which quantities are used, space between each quantity
$RATE:	sampling frequency
$SAMPLES:	number of samples in this record, n
%WAVE:	stimuli waveform, default is sine
%AMPL:	stimuli amplitude, default is Vp-p, (options Vp-p, Vrms)
%OFFSET:	stimuli amplitude offset, default is Vdc
%FREQ:	stimuli frequency, default is Hz, (options Hz, kHz, MHz)
%DMM:	measured stimuli amplitude using the HP 3478A Multimeter default is Vac (options mVac, Vac, mVdc, Vdc). The AC measure is in Vrms
&MAX[i]:	maximum data value in this record, quantity i, i = 1, ...,N
&MIN[i]:	minimum data value in this record, quantity i, i = 1, ...,N
&AVG[i]:	average data value in this record, quantity i, i = 1, ...,N
&AMPL[i]:	peak to peak amplitude (i.e. &AMPL[i]: = &MAX[i]: - &MIN[i]:) quantity i, i = 1, ...,N
&SDEV[i]:	standard deviation, quantity i, i = 1, ...,N $sdev = sqrt\sum_{j=0}^{n-1}(data[j] - avg)^2$
&FFT-FREQ[i]:	denotes peak frequency in the FFT, quantity i, i = 1, ...,N default is Hz,(options Hz, kHz, MHz)
&PSD-AMPL[i]:	Amplitude peak to peak computed from the psdtoampl() function using a GAUSSIAN window.

The **psdtoampl** function is using the Fast Fourier Fransform to correct the peak to peak amplitude. Compared to the peak to peak amplitude calculated from maximumand minimum in the sampled data, the value returned from the *psdtoampl* function does not contain any contribution from the noise and variation in the DC level.

4.2 Power Spectrum Estimates

In order to obtain the best possible values for the amplitude of the sampled sine-wave stimuli signal, we made use of power spectral density (**PSD**) analysis.

To obtain the power spectrum, we used functions from TsLib, (see [Ref. 6]) . Once we have the normalized spectrum, one way to obtain the amplitude is to add a few bins centred around the one containing the signal etc. ....

Since we know the frequency of the signal , there is a more accurate way, which will pick out the stimuli signal even when there is a lot of noise.

Knowing all the parameters of the **PSD** analysis, and the stimuli frequency it is possible to calculate :

1. The bin that contains the signal.

2. How much the signal is offset from the centre-frequency of that bin.
3. A factor to correct the value of the bin for the signals offset from the centre frequency of the bin.

If the stimuli frequency lies inside the range *[0,fs/2)* the bin that contains the signal is easily found as :

$$bin = \frac{fst}{fsN}$$

Where *fst* is the stimuli frequency, *fs* the sampling frequency and *N* the number of samples used in the FFT. *bin* of course has to be rounded to the closest integer.

If the stimuli frequency lies above *fs/2* (Nyquist freq.) then the power is folded back inside the range *[0,fs/2)*. We therefore fold the stimuli frequency and use that for calculating the bin.

We then find the centre frequency (*fcenter*) of that bin as :

$$fcenter = \frac{binfs}{N}$$

The offset of the stimuli freq. in bin's from that centre freq. is then :

$$s = \frac{(fst - fcenter)N}{fs}$$

The correction factor comes from our knowledge of what window was applied to the data before the FFT.

As the Fourier transform of the product of the data and it's window, is equal to the convolution of the Fourier transform of the data and the Fourier transform of the window it follows that for the **PSD** :

$$W(s) = 1/(Wss[\sum_{k=0}^{N-1} e^{2\Pi isk/N} Wk]^2)$$

Where *W(s)* is the window centred around the centre freq. of a given bin, and s is the offset in bin's from that centre freq.

Wss is the "window squared and summed" :

$$Wss = N \sum_{j=0}^{N-1} Wj^2$$

Where *W* indexed *k* and *j* in the above is the window applied to the data before the FFT. So in the case of a rectangular window it is just 1. However, the rectangular window results in the risk of a high degree of leakage from noise sources into the bin containing the signal. And worse, there is quite some effect when the signal isn't commuting, (not a whole number of periods in the sampled interval)

We found that we got good results from using a Gaussian window on the data, which then makes W:

$$W(j) = e^{-(2.0(j-N/2)/N)^2 6.12}$$

Now the correction factor is simply *1/W(s), s* being the offset in bin's, we found above.

Due to the way the PSD has been normalized we need to include the width of a bin in the correction factor which then becomes :

$$C = fs/W(s)N$$

In order to find the peak to peak (p-p) amplitude of the sampled stimuli signal, we now simply multiply the value of the bin with the following :

$$AMPL_{p-p} = \sqrt{PSD(bin)}C2\sqrt{2}$$

Where *PSD[bin]* is the power in the bin which contains the signal.

For a more in depth discussion on PSD, FFT etc. see [Ref. 10] chapters 12 and 13.

The man pages of TsLib (see [Ref. 6]) contains descriptions of the FFT and PSD functions used to obtain the power spectrum, as well as a description of the Gaussian window we used.

5 Calibration Tables

5.1 Calibration tables definition

The The overall calibration files syntax is defined in [Ref. 2] The EFW specific part is explained in this section.

5.1.1 EFW calibration file syntax

BEGIN
<Instrument block>
<Model block>
<Date of update>
<Version number>
<Valid from date>
<Boom length>
<Calibration block>
END

There will be one block for each change in instrument status. Each block is a complete set of calibration data. The elements of the block have the following meaning :

< Instrument block>, *<Model block>*, *<Date of update>*, *< Version number>* and *< Valid from date>* are defined in [Ref. 2].

<Boom Length> , gives the length of the booms when the calibration took place.

<Calibration block>, gives the actual calibration data including stimuli information and results.

The semantics of a block is :

```
<Boom Length> ::=[ <Boom> , <Boom> , <Boom> , <Boom> ]
        <Boom> ::= <Boom number> <Length>
                <Boom number> ::= [ BOOM 1 — BOOM 2 — BOOM 3 — BOOM 4 ]
                <Length> ::= [ 0.0 - 51.0 ]

<Calibration block> ::= <Calibration data>
<Calibration data> :: [ START ] <keyword>
            [ MODE ] <mode>
            [ QTY ] <quantities>
            <sfreq>
            <stim>
            <content>
            [STOP] <keyword>


<keyword> ::= [ FREQ_RESPONSE | OFFSET | NOISE |
                PHASE | STEP_RESPONSE | PULSE_RESPONSE ]

<mode> ::= [ EFIELD | DENSITY ]

<quantities> ::= [ V1L V2L V3L V4L | V1M V2M V3M V4M | V12M V34M |
                   V1H V2H V3H V4H | V12H V34H | V1U V2U V3U V4U ]

<sfreq> ::= [ SAMPLING_FREQ ] <sampling freq>
        <sampling freq> :: = [ 25 | 450 | 9000 | 18000 | 36000 ]

<stim> ::= <ampl> <offset> <wave>
        <ampl> ::= AMPL <amplitude value>
                <amplitude value> ::= float Vpp
        <offset>::= OFFSET <offset value>
                <offset value> ::= float Volt
        <wave> ::= WAVE [ sine | square | pulse ]

<content> ::= <stim> <result>
            <stim> ::= [ <frequency> ]
                    <frequency> ::= stimuli frequency
            <result> ::= [ <amplitude> ]
                    <amplitude> ::= measured amplitude Vp-p/TM for each
                                    quantity in <quantities>
```

5.1.2 EFW calibration file

One calibration file/table for each calibrated EFW instrument has been created. The files are is in ASCII format. The name of the files are:

C*s*_CT_EFW.C*yyyymmdd*_V*vvv*.cal

s denotes satellite number, (1,2,3,4)
yyyymmdd denotes year month day, and
vvv denotes version number.

Satellite instrument	*Number used in space*
FM6	2
FM7	3
FM8	4
FM9 (/ FM5)	1

The *Number used in space* is the satellite numbers used on the data files from Dornier and ESOC. It is also the numbers used in the ISDAT program. The satellite number s in C*s*_CT_EFW.C*yyyymmdd*_V*vvv*.cal , is the *Satellite instrument* , (6, 7, 8 and 9).

In this test the amplitude of the stimuli was set to 1.0 Vp-p (and 16.0 Vp-p) on the HP 3325B. However, some attenuation is present, and the stimuli signal is measured with a HP 3478A multimeter. The multimeter value is recorded in the data files, (see above). If the multimeter value is m [Vrms] the amplitude peak to peak A of the signal is :

$$A = 2m\sqrt{2}$$

The peak to peak data value of the applied sine wave is computed. This was the *&PSD_AMPL[i]* identifier in the list files. Let d = *&PSD_AMPL[i]*. The calibration value c in `Volt / TM` (i.e. `Volt / ''bits''`) is then:

$$c = A/d$$

The value c is the value listed in the calibration tables.

The program **calfile** reads the list files and creates new calibration file.

An example of the calibration table for the Cluster EFW FM6 is:

```
BEGIN
EFW
F6
1998-11-10              # date of update of this block
0.1                     # version number
1998-11-10 17:34:10     # valid from date
#
BOOM 1 50.0 , BOOM 2 50.0 , BOOM 3 50.0 , BOOM 4 50.0 #EFW FM7 94-07-11
#
```

```
START FREQ_RESPONSE
#
MODE EFIELD
#
QTY  V1L V2L V3L V4L
SAMPLING_FREQ 25
AMPL 1.0
OFFSET 0.0
WAVE sine
#
# frequency  V1L volt/unit  V2L volt/unit  V3L volt/unit  V4L volt/unit
#
    0.3      2.077949E-03   2.080682E-03   2.076840E-03   2.077211E-03
    1.0      2.081994E-03   2.085327E-03   2.082018E-03   2.083462E-03
    1.8      2.090587E-03   2.093532E-03   2.092361E-03   2.092551E-03
    3.0      2.111305E-03   2.116353E-03   2.115329E-03   2.115634E-03
    4.0      2.138269E-03   2.145867E-03   2.140829E-03   2.143635E-03
    6.0      2.210153E-03   2.218606E-03   2.217391E-03   2.216129E-03
    7.5      2.298838E-03   2.307338E-03   2.306901E-03   2.305791E-03
    10.0     3.352716E-03   3.368029E-03   3.278352E-03   3.265485E-03
    15.0     4.046196E-02   4.034507E-02   4.029720E-02   4.063203E-02
    20.0     3.421543E-01   3.618976E-01   3.354878E-01   3.516558E-01
    ...      ... . . . .    ... . . . .    ... . . . .    ... . . . .
#
QTY  V1L V2L V3L V4L
SAMPLING_FREQ 25
AMPL 16.0
OFFSET 0.0
WAVE sine
#
# frequency  V1L volt/unit  V2L volt/unit  V3L volt/unit  V4L volt/unit
#
    0.3      2.083590E-03   2.084641E-03   2.083382E-03   2.083595E-03
    1.0      2.087327E-03   2.088756E-03   2.087902E-03   2.088110E-03
    1.8      2.096762E-03   2.099045E-03   2.097726E-03   2.097968E-03
    3.0      2.118278E-03   2.122315E-03   2.120471E-03   2.120867E-03
    4.0      2.144989E-03   2.151001E-03   2.148538E-03   2.149163E-03
    ...      ... . . . .    ... . . . .    ... . . . .    ... . . . .
STOP FREQ_RESPONSE
... . . . . . . .
END
```

5.2 Calibration results

The function ***EfwReadCalFile()*** reads the calibration file into a data structure. This function also calculates the *passband* in each quantity. This *passband* of the quantity is defined as: The selected part from the calibration table is the part when the frequency of the stimuli signal was below the half of the sampling frequency. The *passband* is the average of the three lowest values of this part.

These values from the calibration is in *mV / telemetry unit*. The *EfwReadCalFile()* is used in the ISDAT program (see [Ref. 5]) . Normally the *passband* value is used as the calibration constant.

The tables below gives the *passband* calibration values for the different filters (in *mV / telemetry unit*).

10 Hz filter in EFIELD mode

instrument	*filter*	*stimuli*	*V1L*	*V2L*	*V3L*	*V4L*
F6	10 Hz	1.0Vp-p	2.0835	2.0865	2.0837	2.0844
F6	10 Hz	16.0Vp-p	2.0892	2.0908	2.0897	2.0899
F6	10 Hz	100.0Vp-p	2.1068	2.1086	2.1072	2.1075
F6	10 Hz	1.0Vp-p 30.0VDC	2.1558	2.1577	2.1557	2.1549
F6	10 Hz	1.0Vp-p 60.0VDC	2.1509	2.1511	2.1507	2.1509
F7	10 Hz	1.0Vp-p	2.0529	2.0525	2.0536	2.0532
F7	10 Hz	16.0Vp-p	2.0593	2.0597	2.0594	2.0590
F7	10 Hz	100.0Vp-p	2.0699	2.0701	2.0700	2.0694
F7	10 Hz	1.0Vp-p 30.0VDC	2.1568	2.1573	2.1573	2.1559
F7	10 Hz	1.0Vp-p 60.0VDC	2.1499	2.1507	2.1507	2.1515
F8	10 Hz	1.0Vp-p	2.0922	2.0925	2.0932	2.0914
F8	10 Hz	16.0Vp-p	2.0998	2.0996	2.0990	2.0984
F8	10 Hz	100.0Vp-p	2.1115	2.1113	2.1108	2.1101
F8	10 Hz	1.0Vp-p 30.0VDC	2.1581	2.1568	2.1590	2.1560
F8	10 Hz	1.0Vp-p 60.0VDC	2.1511	2.1531	2.1510	2.1508
F9	10 Hz	1.0Vp-p	2.0924	2.0920	2.0929	2.0951
F9	10 Hz	16.0Vp-p	2.0974	2.0976	2.0975	2.0979
F9	10 Hz	100.0Vp-p	2.1089	2.1091	2.1077	2.1083
F9	10 Hz	1.0Vp-p 30.0VDC	2.1597	2.1589	2.1587	2.1577
F9	10 Hz	1.0Vp-p 60.0VDC	2.1521	2.1545	2.1510	2.1515

180 Hz filter in EFIELD mode

instrument	*filter*	*stimuli*	*V1M*	*V2M*	*V3M*	*V4M*
F6	180 Hz	1.0Vp-p	2.1104	2.1103	2.1118	2.1110
F6	180 Hz	16.0Vp-p	2.1019	2.1024	2.1032	2.1035
F6	180 Hz	100.0Vp-p	2.0972	2.0975	2.0985	2.0988
F6	180 Hz	1.0Vp-p 30.0VDC	2.1526	2.1543	2.1537	2.1541
F6	180 Hz	1.0Vp-p 60.0VDC	2.1475	2.1471	2.1471	2.1478
F7	180 Hz	1.0Vp-p	2.0726	2.0728	2.0735	2.0734
F7	180 Hz	16.0Vp-p	2.0862	2.0863	2.0882	2.0882
F7	180 Hz	100.0Vp-p	2.0951	2.0952	2.0974	2.0974
F7	180 Hz	1.0Vp-p 30.0VDC	2.1526	2.1526	2.1552	2.1522
F7	180 Hz	1.0Vp-p 60.0VDC	2.1470	2.1460	2.1490	2.1477
F8	180 Hz	1.0Vp-p	2.1098	2.1110	2.1120	2.1107
F8	180 Hz	16.0Vp-p	2.1089	2.1083	2.1104	2.1104
F8	180 Hz	100.0Vp-p	2.0880	2.0874	2.0895	2.0895
F8	180 Hz	1.0Vp-p 30.0VDC	2.1535	2.1529	2.1542	2.1554
F8	180 Hz	1.0Vp-p 60.0VDC	586.6265	402.3580	458.5737	407.6893
F9	180 Hz	1.0Vp-p	2.0750	2.0722	2.0765	2.0756
F9	180 Hz	16.0Vp-p	2.0969	2.0960	2.0975	2.0977
F9	180 Hz	100.0Vp-p	2.0833	2.0825	2.0842	2.0844
F9	180 Hz	1.0Vp-p 30.0VDC	2.1549	2.1548	2.1545	2.1553
F9	180 Hz	1.0Vp-p 60.0VDC	2.1488	2.1474	2.1489	2.1490

180 Hz differential filter in EFIELD mode

instrument	*filter*	*stimuli*	*V12M*	*V34M*
F6	180 Hz diff	1.0Vp-p	1.0539	1.0558
F6	180 Hz diff	16.0Vp-p	1.0540	1.0554
F6	180 Hz diff	100.0Vp-p	1.2653	1.2658
F6	180 Hz diff	1.0Vp-p 30.0VDC	1.0737	1.0771
F7	180 Hz diff	1.0Vp-p	1.0466	1.0468
F7	180 Hz diff	16.0Vp-p	1.0390	1.0394
F7	180 Hz diff	100.0Vp-p	1.2669	1.2670
F7	180 Hz diff	1.0Vp-p 30.0VDC	1.0763	1.0774
F8	180 Hz diff	1.0Vp-p	1.0563	1.0564
F8	180 Hz diff	16.0Vp-p	1.0553	1.0556
F8	180 Hz diff	100.0Vp-p	1.2503	1.2504
F8	180 Hz diff	1.0Vp-p 30.0VDC	1.0771	1.0786
F9	180 Hz diff	1.0Vp-p	1.0565	1.0566
F9	180 Hz diff	16.0Vp-p	1.0509	1.0512
F9	180 Hz diff	100.0Vp-p	1.2631	1.2632
F9	180 Hz diff	1.0Vp-p 30.0VDC	1.0780	1.0781

4 kHz filter in EFIELD mode

instrument	*filter*	*stimuli*	*V1H*	*V2H*	*V3H*	*V4H*
F6	4 kHz	1.0Vp-p	2.1202	2.1201	2.1194	2.0816
F6	4 kHz	16.0Vp-p	2.1207	2.1205	2.1196	2.0425
F6	4 kHz	100.0Vp-p	5.2266	5.3250	5.1425	5.3854
F6	4 kHz	1.0Vp-p 30.0VDC	2.1452	2.1443	2.1438	2.1096
F6	4 kHz	1.0Vp-p 60.0VDC	2.1386	2.1382	2.1352	2.0937
F7	4 kHz	1.0Vp-p	2.1219	2.1206	2.1192	2.1215
F7	4 kHz	16.0Vp-p	2.1217	2.1207	2.1197	2.1205
F7	4 kHz	100.0Vp-p	5.6182	5.5875	5.6157	5.5138
F7	4 kHz	1.0Vp-p 30.0VDC	2.1464	2.1455	2.1444	2.1456
F7	4 kHz	1.0Vp-p 60.0VDC	2.1402	2.1401	2.1379	2.1382
F8	4 kHz	1.0Vp-p	2.1215	2.1201	2.1211	2.1195
F8	4 kHz	16.0Vp-p	2.1214	2.1205	2.1201	2.1209
F8	4 kHz	100.0Vp-p	5.2462	5.3203	5.3571	5.3296
F8	4 kHz	1.0Vp-p 30.0VDC	2.1468	2.1453	2.1444	2.1447
F8	4 kHz	1.0Vp-p 60.0VDC	2.1401	2.1368	2.1393	2.1390
F9	4 kHz	1.0Vp-p	2.1238	2.1229	2.1232	2.1226
F9	4 kHz	16.0Vp-p	2.1228	2.1220	2.1215	2.1222
F9	4 kHz	100.0Vp-p	5.5272	5.2980	5.4139	5.4933
F9	4 kHz	1.0Vp-p 30.0VDC	2.1465	2.1483	2.1443	2.1451
F9	4 kHz	1.0Vp-p 60.0VDC	2.1393	2.1387	2.1402	2.1409

8 kHz filter in EFIELD mode

instrument	*filter*	*stimuli*	*V12H*	*V34H*
F6	8 kHz	0.3Vp-p	0.0108	0.0110
F6	8 kHz	1.0Vp-p	0.0128	0.0130
F7	8 kHz	0.3Vp-p	0.0108	0.0107
F7	8 kHz	1.0Vp-p	0.0131	0.0131
F8	8 kHz	0.3Vp-p	0.0106	0.0106
F8	8 kHz	1.0Vp-p	0.0128	0.0129
F9	8 kHz	0.3Vp-p	0.0108	0.0107
F9	8 kHz	1.0Vp-p	0.0129	0.0129

32 kHz filter in EFIELD mode

instrument	*filter*	*stimuli*	*V1U*	*V2U*	*V3U*	*V4U*
F6	32 kHz	1.0Vp-p	2.1170	2.0506	2.1221	2.1183
F6	32 kHz	16.0Vp-p	2.1165	2.0314	2.1194	2.1137
F6	32 kHz	1.0Vp-p 30.0VDC	2.1418	2.0898	2.1465	2.1464
F6	32 kHz	1.0Vp-p 60.0VDC	2.1357	2.0894	2.1366	2.1364
F7	32 kHz	1.0Vp-p	2.1219	2.1190	2.1249	2.1192
F7	32 kHz	16.0Vp-p	2.1196	2.1197	2.1218	2.1200
F7	32 kHz	1.0Vp-p 30.0VDC	2.1451	2.1467	2.1480	2.1455
F7	32 kHz	1.0Vp-p 60.0VDC	2.1364	2.1365	2.1405	2.1404
F8	32 kHz	1.0Vp-p	2.1213	2.1215	1.9788	2.1148
F8	32 kHz	16.0Vp-p	2.1191	2.1199	2.1171	2.0963
F8	32 kHz	1.0Vp-p 30.0VDC	2.1446	2.1454	2.1451	2.1447
F8	32 kHz	1.0Vp-p 60.0VDC	2.1372	2.1391	2.1392	2.1361
F9	32 kHz	1.0Vp-p	2.1232	2.1234	2.1250	2.1192
F9	32 kHz	16.0Vp-p	2.1210	2.1223	2.1222	2.1200
F9	32 kHz	1.0Vp-p 30.0VDC	2.1467	2.1491	2.1498	2.1477
F9	32 kHz	1.0Vp-p 60.0VDC	2.1413	2.1423	2.1412	2.1381

10 Hz filter in DENSITY mode

instrument	*filter*	*stimuli*	*V1L*	*V2L*	*V3L*	*V4L*
F6	10 Hz	1.0Vp-p	2.0528	2.0115	2.1118	2.0483
F7	10 Hz	1.0Vp-p	2.0797	2.0848	2.0909	2.0860
F8	10 Hz	1.0Vp-p	2.1252	2.1238	2.1267	2.1239
F9	10 Hz	1.0Vp-p	2.1287	2.1283	2.1282	2.1289

180 Hz filter in DENSITY mode

instrument	*filter*	*stimuli*	*V1M*	*V2M*	*V3M*	*V4M*
F6	180 Hz	1.0Vp-p	2.0350	1.9896	2.0273	2.1187
F7	180 Hz	1.0Vp-p	2.1276	2.1358	2.1403	2.1364
F8	180 Hz	1.0Vp-p	1664.5073	929.7358	1548.8130	682.9595
F9	180 Hz	1.0Vp-p	2.1345	2.1351	2.1361	2.1388

4 kHz filter in DENSITY mode

instrument	*filter*	*stimuli*	*V1H*	*V2H*	*V3H*	*V4H*
F6	4 kHz	1.0Vp-p	1.9804	1.4061	1.8435	1.7034
F6	4 kHz	16.0Vp-p	2.7001	2.5654	2.6611	2.6967
F7	4 kHz	1.0Vp-p	1.6512	1.4090	1.6563	1.5744
F7	4 kHz	16.0Vp-p	2.6707	2.5931	2.5928	2.6141
F8	4 kHz	1.0Vp-p	1.6773	1.3829	1.6859	1.6212
F8	4 kHz	16.0Vp-p	2.6353	2.5228	2.5957	2.5804
F9	4 kHz	1.0Vp-p	1.6345	1.4212	1.6281	1.5850
F9	4 kHz	16.0Vp-p	2.6097	2.5126	2.5326	2.6145

8 kHz filter in DENSITY mode

instrument	*filter*	*stimuli*	*V12H*	*V34H*
F6	8 kHz	1.0Vp-p	0.0121	0.0122
F6	8 kHz	16.0Vp-p	0.1855	0.1840
F7	8 kHz	1.0Vp-p	0.0123	0.0123
F7	8 kHz	16.0Vp-p	0.1864	0.1862
F8	8 kHz	1.0Vp-p	0.0120	0.0121
F8	8 kHz	16.0Vp-p	0.1857	0.1861
F9	8 kHz	1.0Vp-p	0.0118	0.0119
F9	8 kHz	16.0Vp-p	0.1855	0.1855

6 Noise Measurements

6.1 Digital Noise Measurements

The noise measurements in the Digital Calibration is done in the CPT 3 test, see [Ref. 4]). The **pbmotif/pbrun** program is used to read the raw data files from the CPT 3 tests. The data from one measurement is shown in the Fig. 4.

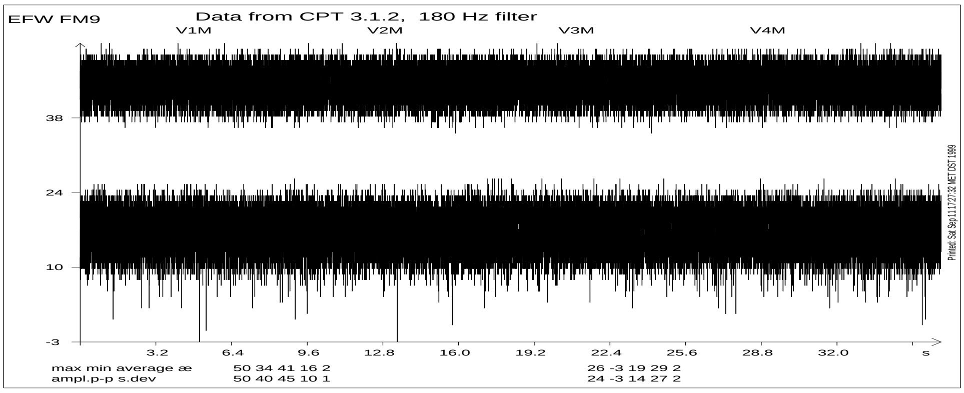

Figure 4: This figure shows the noise measurement of the EFW F9 instrument. The data from the 10 Hz filter is shown.

The noise values in the table below for each satellite are the peak-to-peak values of these measurements.

Noise values in TM units of the 10 Hz filter in EFIELD mode

instrument	*filter*	*V1L*	*V2L*	*V3L*	*V4L*
F6	10 Hz	6	12	6	11
F7	10 Hz	5	12	4	12
F8	10 Hz	7	19	11	20
F9	10 Hz	6	22	8	15

Noise values in TM units of the 180 Hz filter in EFIELD mode

instrument	*filter*	*V1M*	*V2M*	*V3M*	*V4M*
F6	180 Hz	9	16	8	13
F7	180 Hz	9	17	7	14
F8	180 Hz	11	22	12	25
F9	180 Hz	16	29	10	27

Noise values in TM units of the 180 differential Hz filter in EFIELD mode

instrument	*filter*	*V12M*	*V34M*
F6	180 Hz	65	39
F7	180 Hz	60	40
F8	180 Hz	28	23
F9	180 Hz	33	24

Noise values in TM units of the 4 kHz filter in EFIELD mode

instrument	*filter*	*V1H*	*V2H*	*V3H*	*V4H*
F6	4 kHz	7	17	6	14
F7	4 kHz	5	12	6	16
F8	4 kHz	9	21	7	20
F9	4 kHz	9	19	8	20

Noise values in TM units of the 8 kHz filter in EFIELD mode

instrument	*filter*	*V12H*	*V34H*
F6	8 kHz	9	23
F7	8 kHz	9	24
F8	8 kHz	11	28
F9	8 kHz	12	26

Noise values in TM units of the 32 kHz filter in EFIELD mode

instrument	*filter*	*V1U*	*V2U*	*V3U*	*V4U*
F6	32 kHz	15	18	12	21
F7	32 kHz	13	17	13	17
F8	32 kHz	20	27	182	62
F9	32 kHz	12	22	12	20

References

[1] Sten Eckholm. Proposal for EFW EGSE at WEC and S/C level. Technical report, ESTEC, May 1990.

[2] Parrot Michel (editor). WEC Calibration Files. Technical Report CWD-LPCE-DD-0001, LPCE, March 1995.

[3] P Harvey. Cluster II EFW , Command and Telemetry Description , Version 2.1, Revision H. Technical report, Space Sciences Laboratory, University of California, Berkeley, January 1999.

[4] A Lundgren. CPT comprehensive performance test. Technical report, IRF-U, July 1998.

[5] A. Lundgren and G. Holmgren. ISDAT programmers guide. 2. DBH instrument modules. Technical Report CWD-IPG-002, IRF-U, February 1994.

[6] A. Lundgren and A. Mälkki. ISDAT reference. scientific subroutine library. Technical Report CWD-SUM-005, IRF-U, March 1994.

[7] B Lybekk and M Thomsen. Cluster WEC stimuli-PC program. Technical report, Department of Physics, University of Oslo, 1993.

[8] Torleif A Sten. Electronic ground support equipment for the Cluster electric field and wave experiment. Technical Report UIO-PHYS-93-37, University of Oslo, October 1992.

[9] Torleif A Sten and Espen Trondsen. Documentation of the Cluster EGSE built at the University of Oslo. Technical Report UIO/PHYS/99-02, ISSN-0332-5571, University of Oslo, March 1999.

[10] W.T. Vetterling W. H. Press, S. A. Teukolsky and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1992.