DBBC Operations

Content

- Setting up the DBBC
- FS preparation for the DBBC
- Useful test with the DBBC at the station alone
  - test recordings
  - checks using Mark5B-Tools: vlbi2, bstate, bpcal
  - IF levels
- experiment preparation, drudg
- checks before the start
  - mostly the same as for analogue racks: clocks, IF counts, BBC counts, Tsys, maybe phase-cal
- checks during observations/recording
- antabfs files? currently no support by antabfs.pl
  - Dave has written a perl script that works with continues cal at Effelsberg, but wasn't tested for other observatories yet.

Setting up the DBBC

- Cabling the DBBC: IF, 1pps, 10 MHz, (80 Hz calibration?)
- Starting the DDC or PFB software (server) on the DBBC Windows PC
  - Newest version always available at http://www.hat-lab.com/hatlab/support
    currently v101 or v102 for DDC or v14 for PFB software
  - configuration file needs to be edit for your hardware installation

```
1 dbbc2_ddc_v101.bit 597.00 8      #  1st number indicates ADB1 or ADB2
1 dbbc2_ddc_v101.bit 682.00 8      # 2nd is the firmware file to use
1 dbbc2_ddc_v101.bit 853.00 8      #  3rd and 4th BBC default frequency and band width
1 dbbc2_ddc_v101.bit 938.00 8
1 dbbc2_ddc_v101.bit 597.00 8
1 dbbc2_ddc_v101.bit 682.00 8
1 dbbc2_ddc_v101.bit 853.00 8
1 dbbc2_ddc_v101.bit 938.00 8
1 dbbc2_ddc_v101.bit 597.00 8
1 dbbc2_ddc_v101.bit 682.00 8
1 dbbc2_ddc_v101.bit 853.00 8
1 dbbc2_ddc_v101.bit 938.00 8
1 dbbc2_ddc_v101.bit 597.00 8
1 dbbc2_ddc_v101.bit 682.00 8
1 dbbc2_ddc_v101.bit 853.00 8
1 dbbc2_ddc_v101.bit 938.00 8
0 fila10g_v2_0.bit                 # firmware for FILA10G (if present)
38000 38000 38000 38000            # initial target CoMos IF levels for AGC
100 100 100 100                    # phase calibration values
CAT2 1024                         # type of clock module CAT1|2 and sampling freq.
```

Setting up the FS for the DBBC

- The new FS releases 9.11.X now support the DBBC in DDC mode, that includes
  - drudg
  - Tsys calibration
  - ONOFF and FIVEPT measurements for gain calibration and pointing
  - fmset
  - a full description can be found in /usr2/fs/misc/dbbc.txt
- Six things need to be set-up to use the FS with DBBC: (1) equip.ctl, (2) skedf.ctl, (3) dbbad.ctl, (4) station.prc,(5) point.prc, and (6) .Xresources.
  → again a full description can be found in /usr2/fs/misc/dbbc.txt

Test at the station alone

- First functionality can be tested with the DBBC_client or from the FS
    - select different IF inputs for the ADBs and let AGC adjustment work, e.g.
      ifa # for query
      ifa=1,agc,1 # to set RF input 1, agc on, IF filter 1 (500-1000 MHz)
    - read out BBCs set different frequencies, …
      bbc01 # for query
      bbc01=596.00,a,16.00 # to set BBC freq=596 MHz, IFA, BBC band width = 16 MHz
- Test recordings are good to control the correct sampling (bit statistics), band pass shape, and pcal tones
- The Mark5B comes with a set of programs that allow to check the bit statistics (bstate), do auto- or cross correlations (vlbi2), and extract phase cal (bpcal).
    - *> bstate*

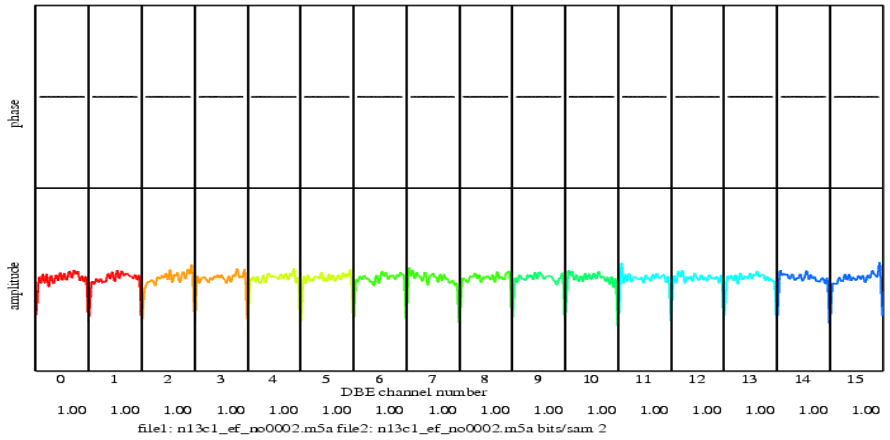
```
Usage: bstate <input m5b fname> <# frames>
> bstate n13c1_ef_no0002.m5a 200
ch    --     -      +      ++      --    -     +    ++   gfact
 0  88032 157895 160426  93647    17.6  32.1  31.6  18.7   1.00
 1  93899 151616 154405 100080    18.8  30.9  30.3  20.0   0.95
 2  92338 153774 156561  97327    18.5  31.3  30.8  19.5   0.97
 3  91497 154665 157139  96699    18.3  31.4  30.9  19.3   0.97
 4  84797 161299 163577  90327    17.0  32.7  32.3  18.1   1.03
 5  89860 155939 158073  96128    18.0  31.6  31.2  19.2   0.98
 6  88426 157547 159995  94032    17.7  32.0  31.5  18.8   1.00
 7  85429 160711 162749  91111    17.1  32.5  32.1  18.2   1.02
 8  89485 153806 157650  99059    17.9  31.5  30.8  19.8   0.97
 9  92445 150796 154915 101844    18.5  31.0  30.2  20.4   0.95
10  89559 153929 157131  99381    17.9  31.4  30.8  19.9   0.97
11  92958 151219 155066 100757    18.6  31.0  30.2  20.2   0.95
12  89607 153163 157750  99480    17.9  31.6  30.6  19.9   0.97
13  84856 158081 162791  94272    17.0  32.6  31.6  18.9   1.01
14  84164 159461 163177  93198    16.8  32.6  31.9  18.6   1.02
15  83381 159953 163898  92768    16.7  32.8  32.0  18.6   1.02
```

    - *> vlbi2*

```
vlbi file1 file2 -proctime proctime [-rev <0|1>] [-2bit <0|1>] [-tforce <0|1>]
  2bit:   1 to enable 2-bit input
  rev:    1 to reverse channels in the plot
  tforce: 1 to force correlation, ignoring timestamps
> vlbi2 n13c1_ef_no0002.m5a n13c1_ef_no0002.m5a -2bit 1
```

for autocorrelation

> gv dd1.pos



-
  ```
  > bpcal
  Usage: bpcal <input m5b fname> <tone freq (KHz)> <# frames>
  > bpcal n13c1_ef_no0002.m5a 2490 500

  integration time  0.078 sec
  ch  amp   phase(dg)
   0     1   153.7
   1     0   -93.5
   2     1    83.2
   3     2   -20.0
   4     1   -54.9
   5     2  -111.1
   6     0  -179.6
   7     1  -152.4
   8    12   -94.5
   9    11   -82.5
  10    11   -69.3
  11    12   -47.9
  12    12    24.3
  13    12   -58.8
  14    10  -154.2
  15     9   134.2
  ```

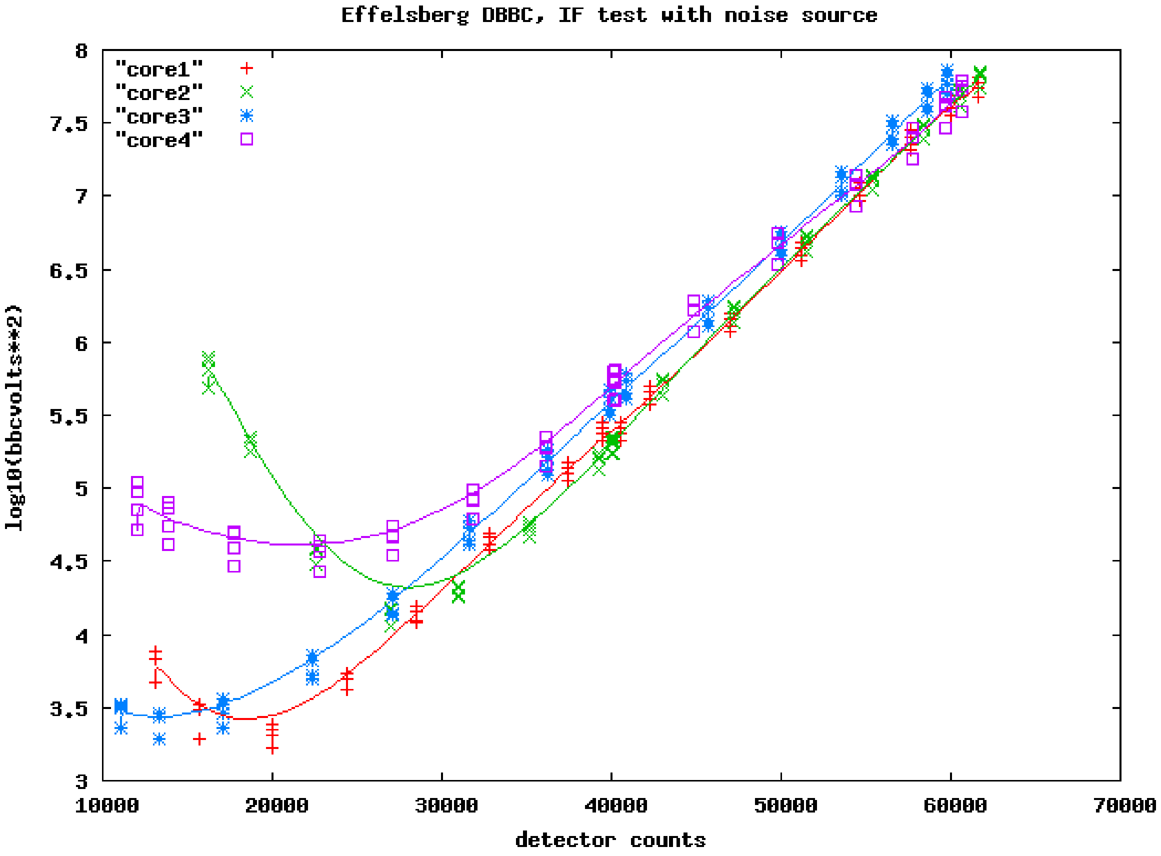
  shows phase cal in USB channels

-
  ```
  > bpcal n13c1_ef_no0002.m5a 2510 500
  integration time  0.078 sec

  ch  amp   phase(dg)
   0    11    97.6
   1    11    83.6
   2    12    80.6
   3    12    51.9
   4    10    13.2
   5    12    98.1
   6     9  -174.2
   7    11   -98.0
   8     1   -62.6
   9     0    26.8
  10     1    78.1
  11     1     5.9
  12     1   108.4
  13     0   135.0
  14     1  -145.3
  15     1  -142.0
  ```

  shows phase cal on LSB channels

- We have seen earlier that we can specify values for the target counts where the IF input power should be adjusted to.
  With an increasing number of DBBCs the best target IF levels seem to cluster around 35000 to 45000 counts, but it might be worth to test those for your DBBC.
  As an example:
  The Effelsberg DBBC was tested using the 500-1000 MHz IF of the 6cm receiver. Antenna was in zenith position looking at clear sky, therefore no major variations on the input level are expected.
  Then the attenuation was changed in steps of 2.5 dB, while checking detector counts, bbc counts and doing some short 10 sec recordings at the Mark5B to check phase cal amplitudes.
  The figures below show the BBC power against detector counts (Fig. 1) and the phase-cal amplitude calculated by bpcal over 0.15 sec against detector counts (Fig. 2). Since the DBBC was running in astro mode and the Mark5B+ at 512 Mbps (0xffffffff,2) only BBC01 to 08 were recorded.



- This can be run from a FS procedure like:

```
define  dbbciftest    00000000000x
" dbbc if test
!+1s
" if at att 0db
vdbbcifset=0
!+15s
collect32
"record if=att0
mk5=record=on:iftest_att0
!+10s
mk5=record=off
" if at att 5db
vdbbcifset=5
!+15s
collect32
"record if=att5
mk5=record=on:iftest_att5
!+10s
mk5=record=off
" if at att 10db
vdbbcifset=10
!+15s
collect32
"record if=att10
mk5=record=on:iftest_att10
!+10s
mk5=record=off
" if at att 15db
vdbbcifset=15
!+15s
collect32
```

  … and so on

- where vdbbcifset contains

```
define  vdbbcifset    00000000000x
ifa=1,$,1
ifb=1,$,1
ifc=1,$,1
ifd=1,$,1
enddef
```

- and collect32 reads out all 16 BBCs (32 since all are LSB and USB :-))

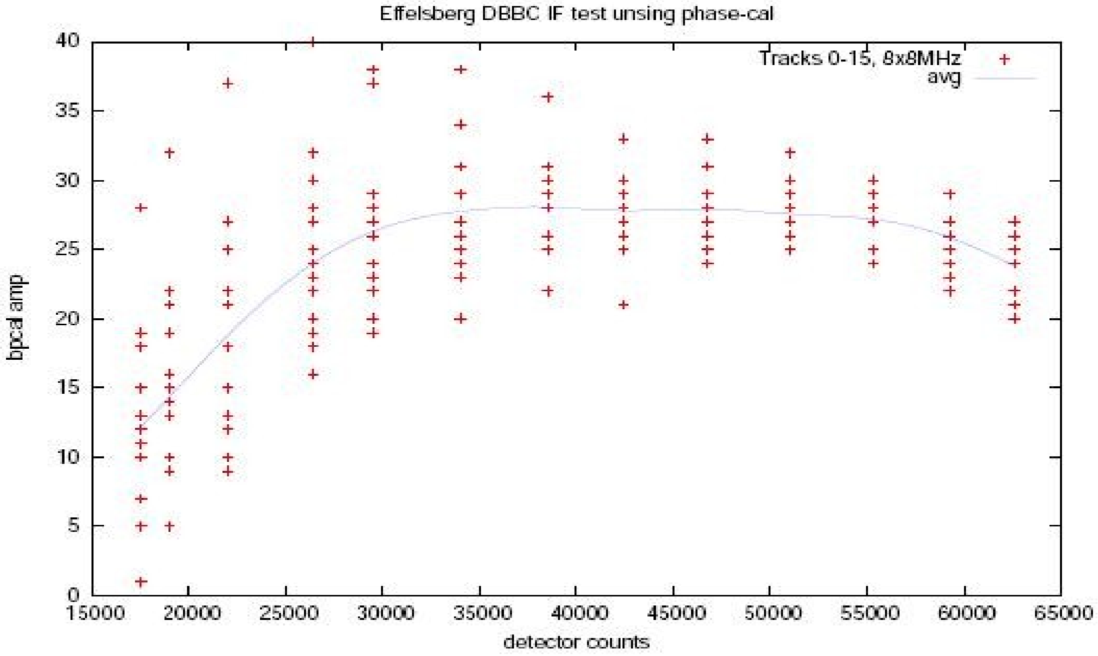


Figure 2: Phase-cal amplitude calulated by bpcal over 0.15 sec against detector counts.

Preparing for real observations

- **drudging the schedule**

```
> drudg n13c1.vex
drudg_rdctl: Reading system control file /usr2/control/skedf.ctl
   $SCHEDULES$SNAP $PROC $SCRATCH $PRINT $MISC

 DRUDG: Experiment Preparation Drudge Work (NRV & JMGipson 2013Jan23)
Version:  9.11.01
 Opening file /usr2/sched/n13c1.vex for schedule n13c1.vex
VREAD01 -- Got a VEX file to read, VEX_rev = 1.5;.
…...
 Output for which station (type a code,  :: or q to quit, = for all) ? Ef
VOBINP - Generating observations     0    31 scans in this schedule.
 WARNING! Using equipment from control file:
     Replacing rack none     by DBBC
     Replacing rec1 Mark5B   by Mark5B
     Replacing rec2 none     by none

 Equipment at EFLSBERG:
   Rack: DBBC     Recorder 1: Mark5B   Recorder 2: none
 Schedule will start with recorder 1.

 Select DRUDG option for schedule /usr2/sched/n13c1.vex at EFLSBERG
 1 = Print the schedule                7 = Re-specify stations
 2 = Make antenna pointing file        8 = Get a new schedule file
 3 = Make SNAP file (.SNP)             9 = Change output destination, format
 4 = Print complete .SNP file
 5 = Print summary of .SNP file       11 = Show/set equipment type
                                      12 = Make procedures (.PRC)
                                      15 = Data Transfer Overide
 51 = Print PI cover letter
 20 = Make fake lvex
 0  = Done with DRUDG
 ?
```

- Use option 3 to make the SNAP file and 12 for the procedures. The snp-file will look like every snp-file, but the prc-file now holds the setup for the DBBC

```
 ?3
SNAP output for EFLSBERG
 OK to purge existing file /usr2/sched/n13c1ef.snp (Y/N) ? Y
 Translation for EFLSBERG
   From file: /usr2/sched/n13c1.vex To snap file: /usr2/sched/n13c1ef.snp

 Equipment at EFLSBERG:
   Rack: DBBC     Recorder 1: Mark5B   Recorder 2: none
```

?12

```
Procedures for EFLSBERG
 PROCEDURE LIBRARY FILE /usr2/proc/n13c1ef.prc FOR EFLSBERG
 NOTE: These procedures are for the following equipment:
   Rack:       DBBC
   Recorder 1: Mark5B
     exper_initi  setup01      dbbc018     Converted IF A  to a1
Converted IF C  to b3
Converted IF A  to a1
Converted IF C  to b3
Converted IF A  to a1
Converted IF C  to b3
Converted IF A  to a1
Converted IF C  to b3
 ifd01
 Equipment at EFLSBERG:
   Rack: DBBC     Recorder 1: Mark5B   Recorder 2: none
```

- Note that at least for version 9.11.1 there are still some lines in the prc-file that might be worth to comment out (initialisation of ifc and loc).

```
define  ifd01        13100191309x
ifa=1,agc,1
ifb=3,agc,1
ifc=,agc,1
lo=
lo=loa,4240.00,usb,rcp,1.000
lo=lob,4240.00,usb,lcp,1.000
lo=loc,4240.00,usb,lcp,1.000
enddef
```