

**Massachusetts Institute of Technology  
Haystack Observatory  
WESTFORD, MASSACHUSETTS 01886**

27 August 2008

To: Broadband Development Group

From: C. J. Beaudoin

Subject: New scripts and utilities for DBE1 setup and diagnostics

**Overview**

Described here are some useful Unix scripts, programs, and utilities that have been developed and implemented for the purposes of the BBDev effort, specifically for setting up GEO DBEs for VLBI experiments and diagnosing the appearance of spurious signals generated by the DBE.

**1. File Descriptions**

The following scripts and source code currently exist only in the "oper" home directory on the Westford Mark5B+ (192.52.53.109). In the interest of consolidating the scripts required to setup the DBE for recording and conducting autocorrelation diagnostics (i.e. appearance of spurious signals), the following scripts have been developed:

**~/initDBE**

Arms the DBE, initializes the gains, and synchronizes the DBE internal generators to the input 1pps and 5MHz signals. This script consolidates the following three commands documented in Arthur's DBE setup cookbook (BBDev memo #8):

```
~/dbe/dbers232 -arm  
~/dbe/dbers232 -gain 1000  
./dbe5bsetup2bit
```

**~/bin/bsutility**

This script is a utility for initializing and resetting B-state gains as well as viewing B-state statistics. Note that the need to manually parse the individual channel statistics and save various files has been eliminated. Storage of the channel statistics has been kept as an option.

Usage1: `bsutility reset [save]`

Collects data from the Mark5B and sets the bstate gains according to the stats. After the gains are set another collection is performed to observe the corrected BS stats. This input mode will also optionally save the BS stats used to set the gains in separate IF0 and IF1 files. The date and time are appended to the end of the filenames.

Usage2: `bsutility initgain <gain_value>`  
initialize all BS gains to `gain_value`

Usage3: `bsutility view`  
Collects data from the Mark5b and outputs IF0 and IF1 BS statistics to the screen. The BS gains are not set in this mode.

Usage4: `bsutility <IF0_BS_filename> <IF1_BS_filename>`  
reads the IF0 and IF1 files stored as described under Usage1 then sets the BS gains according to the saved BS stats.

## **~/bin/fftacs**

This is a binary executable compiled from the C-source code `fftacs.c` using the make script "makefftacs" (both the source code and make file are in `~/bin/source`). Using FFTs, the program calculates the autocorrelation spectrum of data stored in a `.m5b` file generated by a DBE running the GEO bitcode. "fftacs" generates ASCII output files `IF0_out` and `IF1_out` which contain the correlation values (currently unnormalized) and their corresponding RF frequencies which are determined based on the selected Nyquist zone.

Usage: `fftacs <M5B File> [-NZ <value>] [-FW <value>] [-NF <value>]`

-NZ: Nyquist Zone Flag (default 2), valid values are 1,2, and 3.

-FW: Noise Filter Width Flag (default 1024), # of samples of the ACF to keep when lowpass filtering the ACS. The sample count will be truncated to the next lower power of 2. This quantity is tantamount to specifying the number of lags in a time domain correlation processor.

-NF: # Frames Flag (default 1996), the total sample count<sup>1</sup> will be truncated to the next lower power of 2.

---

<sup>1</sup> In the Mark 5 file storage format there are a total of 2500 32-bit samples in a single frame. The total sample count is therefore given by  $2500 * NF$ .

## FFT Library Setup

The FFT function prototypes are located in the header file `"/usr/include/acml.h"` and compilation requires that the library files `"/usr/lib/libacml.so"` and `"/usr/lib/libacml.a"` be linked to using the flag `"-lacml"` (see `"makefftacs"`). The header and library files were obtained from the AMD website:

["http://developer.amd.com/cpu/Libraries/acml/downloads/Pages/default.aspx#downloads"](http://developer.amd.com/cpu/Libraries/acml/downloads/Pages/default.aspx#downloads) by selecting the file `"acml-4-1-0-ifort-32bit.tgz (19 MB)"` (The `"ifort"` fortran compiler reference in the filename is a misnomer, the files are applicable to C compilers as well). The contents of the `tgz` file are decompressed to a folder `"acml-4-1-0-ifort-32bit.tgz_FILES"` and the header and library files are located in the `"contents-acml-4-1-0-ifort-32bit.tgz_FILES/ifort32/include"` and `"contents-acml-4-1-0-ifort-32bit.tgz_FILES/ifort32/lib"` directories, respectively. The libraries for multi-processor architectures are located in the `"ifort32_mp"` folders. Once located, the header and library files are simply copied to the `"/usr/include"` and `"/usr/lib"` directories on the local machine, which will require root privileges. It is possible to specify the location of the header and library files in the source code and on the command line, respectively, if root access is not available.

## 2. Proposed Experimental DBE Setup

With the scripts described in the previous section (and assuming all necessary hardware connections have been made and `dimino` has been started), setting up the geodetic DBEs for an experiment can be completed by executing the following commands in sequence from the `oper` home directory:

- (1) `initDBE`
- (2) `bsutility reset`

Following setup of the DBE, the `bsutility view` command can be executed as many times as necessary in order to examine the stability of the `bstate` statistics.

## 3. Diagnosing DBE channel Autocorrelation Spectra

A 300K 50 ohm load amplified by 73 dB provides the necessary `-13 dBm/512 MHz` input PSD required by each input IF channel of the DBE1. After making the proper hardware connections and starting `dimino`, the autocorrelation spectra can be diagnosed by executing the following list of commands from the `oper` home directory:

- (1) Initialize the DBE  
`initDBE`
- (2) Perform any desired `bstate` operations  
`bsutility reset |`  
`bsutility initgain |`

```
bsutility <IF0> <IF1> |  
bsutility view |  
no bstate operation
```

(3) Calculate autocorrelation spectra

```
fftacs file.m5b
```

(4) Start gnuplot

```
gnuplot
```

(5) Set axes labels, title and grid

```
set xlabel "Frequency (MHz)"  
set ylabel "Unnormalized Correlation"  
set title "Autocorrelation Spectrum"  
set grid
```

(6) Plot all frequency channels of IF channel 'n' on single set of axes (figure (1))

```
plot 'IFn_out' with lines
```

(7) Plot frequency channel 'm' of IF channel 'n' on single set of axes (figure (2))

```
plot 'IFn_out' index m with lines
```

Plots can also be exported to an image file. GNUPLOT supports a wide variety of file types including postscript, png and gif. To see which file type are supported type the command:

```
set terminal
```

at the GNUPLOT command prompt. In order to export a plot to an image file, the following commands must be executed before issuing the plot command (step 6 or 7 above)

```
set terminal <image file type>  
set output "filename"
```

To restore plot output to the screen use:

```
set terminal X11
```

Note that 'gnuplot' must be installed on the local machine. Rich Crowley has this software and can install it remotely from his office machine. More information on gnuplot can be found at the URL [www.gnuplot.info](http://www.gnuplot.info)

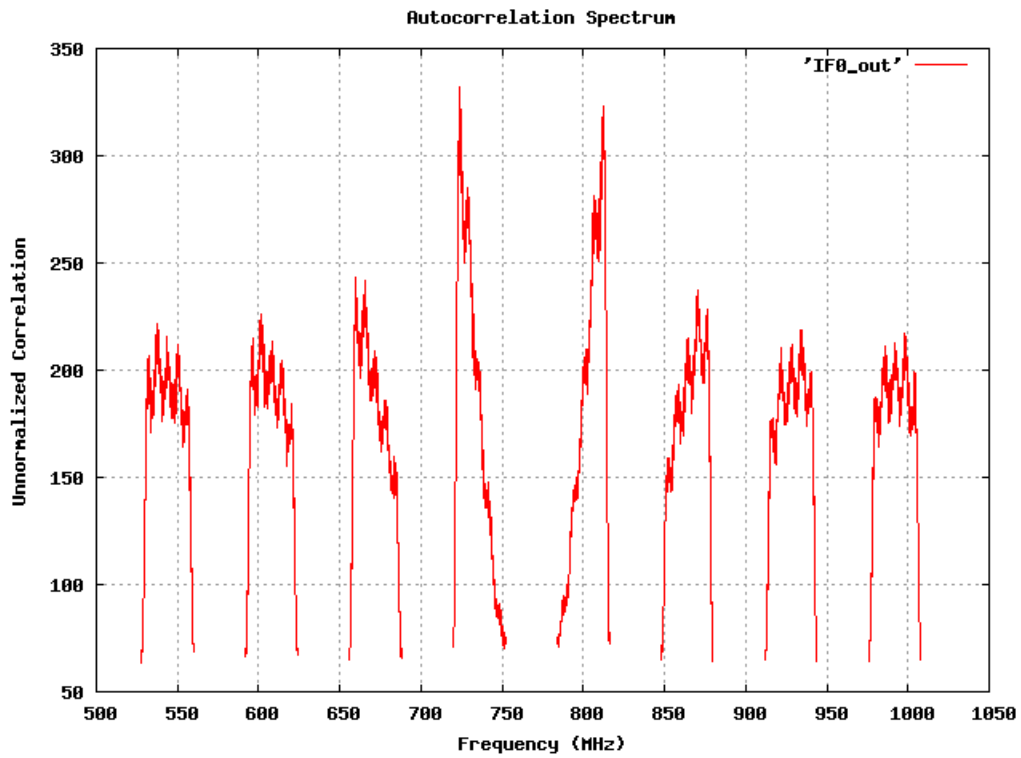


Figure 1: Example of autocorrelation spectrum plot created with GNUPLOT. All IF0 DBE Channels are shown.

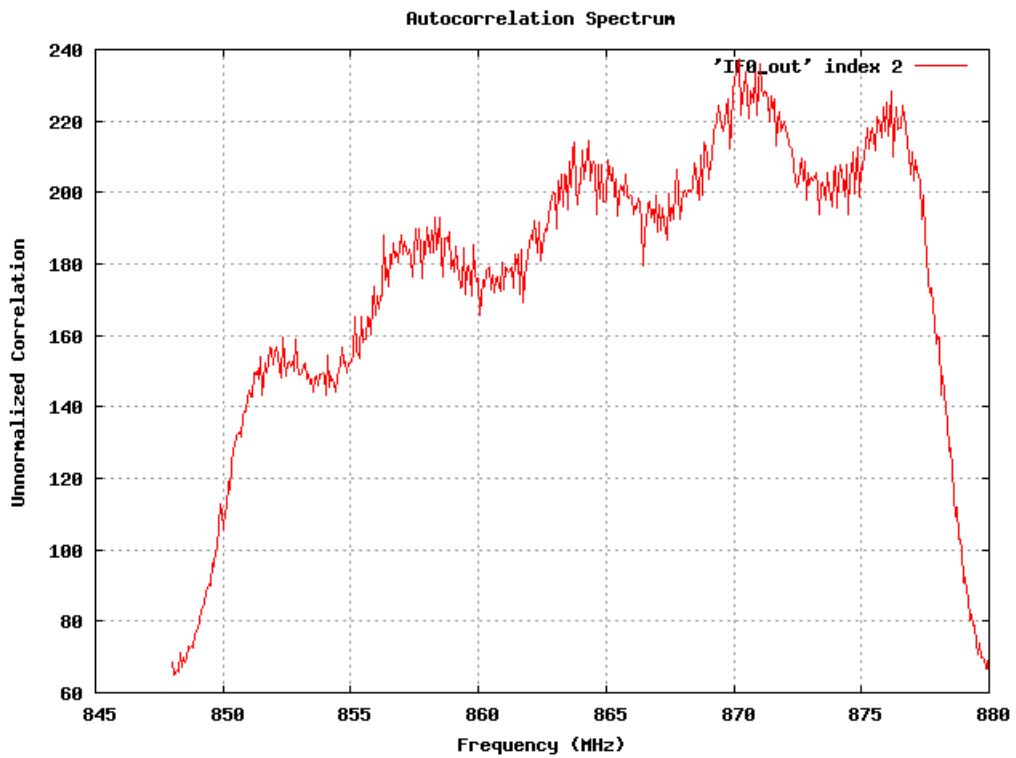


Figure 2: Example of autocorrelation spectrum plot created with GNUPLOT. IF0 DBE channel 3 is shown (GNUPLOT indexing starts at zero).