DiFX Correlation at the Stations

Alessandra Bertarini,

IGG University of Bonn & MPIfR Bonn

DiFX

DiFX -> Distributed FX correlator

DiFX is a software correlator

DiFX is a free downloadable software from:

http://cira.ivec.org/dokuwiki/doku.php/difx/installation

DiFX needs IPP libraries (IPP requires licence).

For one baseline (zero-baseline test) DiFX can be installed on a 4-core machine.

Typical running time for 256 Mb/s, 1 bit sampling, 1 pol and 40 s of data: 5 min.

Might need to install HOPS package too.

DiFX File Requirements

DiFX reads data in: Mark 4/VLBA format,

Mark 5B format,

VDIF format,

LBA format

DiFX needs one valid VEX file => you will need to generate one.

To check results: AIPS → too complex for the purpose

sniffer → dunno how to use it

HOPS fourfit → my favourite option!

VEX template: you can re-use the same template file only changing:

1) in $SCHED: start time

2) in $EOP: the eop's can be set all to zeros

3) in $CLOCK: clock_early and start time

4) Adding all over the "new" digital station (done once in template)

5) check "track" assignment

Example - $SCHED: start time

In the old file is:

```
scan No0001;
   start=2012y138d17h00m00s; mode=3mmpol; source=OJ287
   station=Ef:    0 sec:  440 sec:    0.000 GB:   :         : 1;
   station=On:    0 sec:  440 sec:    0.000 GB:   :         : 1;
endscan;
```

Suppose new observation starts at DOY 260 at UT 15h30m 10 s of data, Onsala digital (Od) vs Onsala analogue (On):

```
scan No0001;
   start=2012y260d15h30m00s; mode=3mmpol; source=OJ287
   station=Od:    0 sec:  10 sec:    0.000 GB:   :         : 1;
   station=On:    0 sec:  10 sec:    0.000 GB:   :         : 1;
endscan;
```

Note avoid less than 3 s of data (useless complication!)

EOP: VEX example for observation on DOY 179.

OLD:

```
$EOP;
def EOP0;
 TAI-UTC= 34 sec;
 A1-TAI= 0 sec;
 eop_ref_epoch=2012y121d;
 num_eop_points=1;
 eop_interval=24 hr;
 ut1-utc  = -0.543703 sec;
 x_wobble = 0.006280 asec;
 y_wobble = 0.368810 asec;
enddef; [..]
def EOP4;
 […]
enddef;
```

NEW:

```
$EOP;
def EOP0;
 TAI-UTC= 34 sec;
 A1-TAI= 0 sec;
 eop_ref_epoch=2012y177d;
 num_eop_points=1;
 eop_interval=24 hr;
 ut1-utc  = 0.0 sec;
 x_wobble = 0.0 asec;
 y_wobble = 0.0 asec;
enddef;
def EOP4;
 […]
enddef;
```

Note: DiFX needs EOPs for 5 days of which two prior to the observation

Example - $CLOCK: clock_early and start time

In the old file is:

gps-fmout

```
$CLOCK;
def On;
clock_early=2012y123d17h00m :  13.08  usec:2012y123d17h00m0s: 0;
enddef;
def Od;
clock_early=2012y123d17h00m :  15.06  usec:2012y123d17h00m0s: 0;
enddef;
```

Suppose new observation starts at DOY 260 at UT 15h30 m:

```
$CLOCK;
def On;
clock_early=2012y260d15h30m :  20.01  usec:2012y260d15h30m0s: 0;
enddef;
def Od;
clock_early=2012y260d15h30m :  21.05  usec:2012y260d15h30m0s: 0;
enddef;
```

DiFX - VEX File

Adding throughout the "new" digital station

Suppose in VEX there is Onsala (On) => add Od:

```
$MODE;
def GEOSX-SX;
ref $FREQ = GEOSX-SX01:Aa:Bb:Cc:On;
ref $IF =  GEOSX-SX01: Aa:Bb:Cc:On;
ref $TRACKS = Mark4:Aa:Bb:On;
ref $TRACKS = Mark5B:Cc;
[..]
```

Becomes:

```
$MODE;
def GEOSX-SX;
ref $FREQ = GEOSX-SX01:Aa:Bb:Cc:On:Od;
ref $IF =  GEOSX-SX01: Aa:Bb:Cc:On:Od;
ref $TRACKS = Mark4:Aa:Bb:On;
ref $TRACKS = Mark5B:Cc:Od;
[..]
```

NOTE: DBBC track is Mk5B!

In the old VEX:

```
def ONSALA;
 site_type = fixed;
 site_name = ONSALA;
 site_ID = On;
 site_position = 3370605.9823 m : 711917.5286 m: 5349830.7678 m;
 [..];
enddef;
```


In the new VEX:

```
def ONSALA
 [..]
enddef;

def ONDBBC
site_type = fixed;
 site_name = ONDBBC;
 site_ID = Od;
 site_position = 3370605.9823 m : 711917.5286 m: 5349830.7678 m;
 [..];
enddef;
```

NOTE: station positions are identical! CALC will nevertheless shift the stations (!). No worries, just tiny fringe rate !

Check "track" assignment: VEX speaks (still) tape language!

Mk 4	VSI=geo	VSI=astro
1US	0	0
1UM	1	1
2US	2	2
2UM	3	3
3US	4	4
3UM	5	5
4US	6	6
4UM	7	7
5US	8	8
5UM	9	9
6US	10	10
6UM	11	11
7US	12	12
7UM	13	13
8US	14	14
8UM	15	15

Mk 4	VSI=geo	VSI=astro
1LS	16	16
1LM	17	17
2LS	-	18
2LM	-	19
3LS	-	20
3LM	-	21
4LS	-	22
4LM	-	23
5LS	-	24
5LM	-	25
6LS	-	26
6LM	-	27
7LS	-	28
7LM	-	29
8LS	18	30
8LM	19	31

DiFX - VEX File

Check “track” assignment: VEX speaks (still) tape language!

Mk 4	VSI=geo	VSI=astro
9US	21	-
9UM	22	-
10US	23	-
10UM	24	-
11US	25	-
12UM	26	-
12US	27	-
13UM	28	-
13US	29	-
14UM	30	-
14US	31	-

In VEX enter VSI output + 2!

i.e. 1US: VSI output = 0 → VEX TRACK = 0 + 2 = 2

TRACKS sorting in VEX:

```
$FREQ;
def GEOSX-SX01;
chan_def = &X:8212.99 MHz:U:8.000 MHz:&CH01:&BBC01:&U_cal;
[..]
enddef;
[..]
$TRACK;
def Mark5B;
fanout_def = A : &CH01 : sign : 1 : 02 ;
[..]
enddef;
```

USB

From tables above:

Mk 4	VSI=geo	VSI=astro
1US (BBC01)	0	0

VSI output = 0, i.e. TRACK = 02

DiFX - V2D File

DiFX requires different files, based on VEX

v2d = vex-to-difx creates those files.

More info:
http://cira.ivec.org/dokuwiki/doku.php/difx/vex2difx

Same v2d template for all tests, only to be changed:

1) Filelist
2) Integration time
3) No. spectral channels

How to create filelist:

Directory2filelist : DiFX program to create filelists

directory2filelist /path-to-data/ Mode > output.filelist e.g.:

directory2filelist /raid1/exp1/ Mark5B-512-8-2 > on.filelist

format Mbps bbc no. nbit

```
/raid1/exp1/No0001 56054.708345 56054.708935
/raid1/exp1/No0002 56054.709479 56054.710058
```

MJD

NOTE: Mk5B MJD in filelist is offset by 1000 days → filelist needs editing (use linux editors!).

layout of v2d file:

vex = vex file name

antennas = two letter code of the participating stations (e.g. antennas = OD, ON)

singleScan = True

tweakIntTime = True

suggested values for zero-baseline test

```
SETUP r1534
{
  tInt = integration time in second (e.g. 0.2 s, 1 s ...)
  doPolar = True/False
  nChan = no. spectral channels (e.g. 128, 512, 1024)
}
```

(max for fourfit)

DiFX - V2D File

Layout of v2d file cont.:

```
RULE clock{
    scan = scan name (e.g. No0001)
    setup = r1534
}
ANTENNA AB
{
   filelist = ab.filelist
}
```

In fringe test mostly only one scan

Run DiFX

1) run the program vex2difx: *vex2difx*

vex2difx r1534.v2d

vex2difx creates the files .input, .calc, ...

2) run the correlator using the script *startdifx*:

startdifx r1534_1.input

Difx creates a directory called r1534_1.difx

3) run *difx2mark4* to create the files for fourfit:

difx2mark4 r1534_1.difx (will create a directory 1234)

4) Run *fourfit:*

fourfit -pt -c cf_1234 No0001

Fringe Fit: Control File

cf_1234 is fourfit control file.

Tells fourfit what to do.

Basic layout:

`pc_mode normal` (pcal applied)

`sb_win -256.0 256.0 mb_win -2.0 2.0 dr_win -30.e-4 30.e-5`

sb_win: sbd search window bounds (µs)

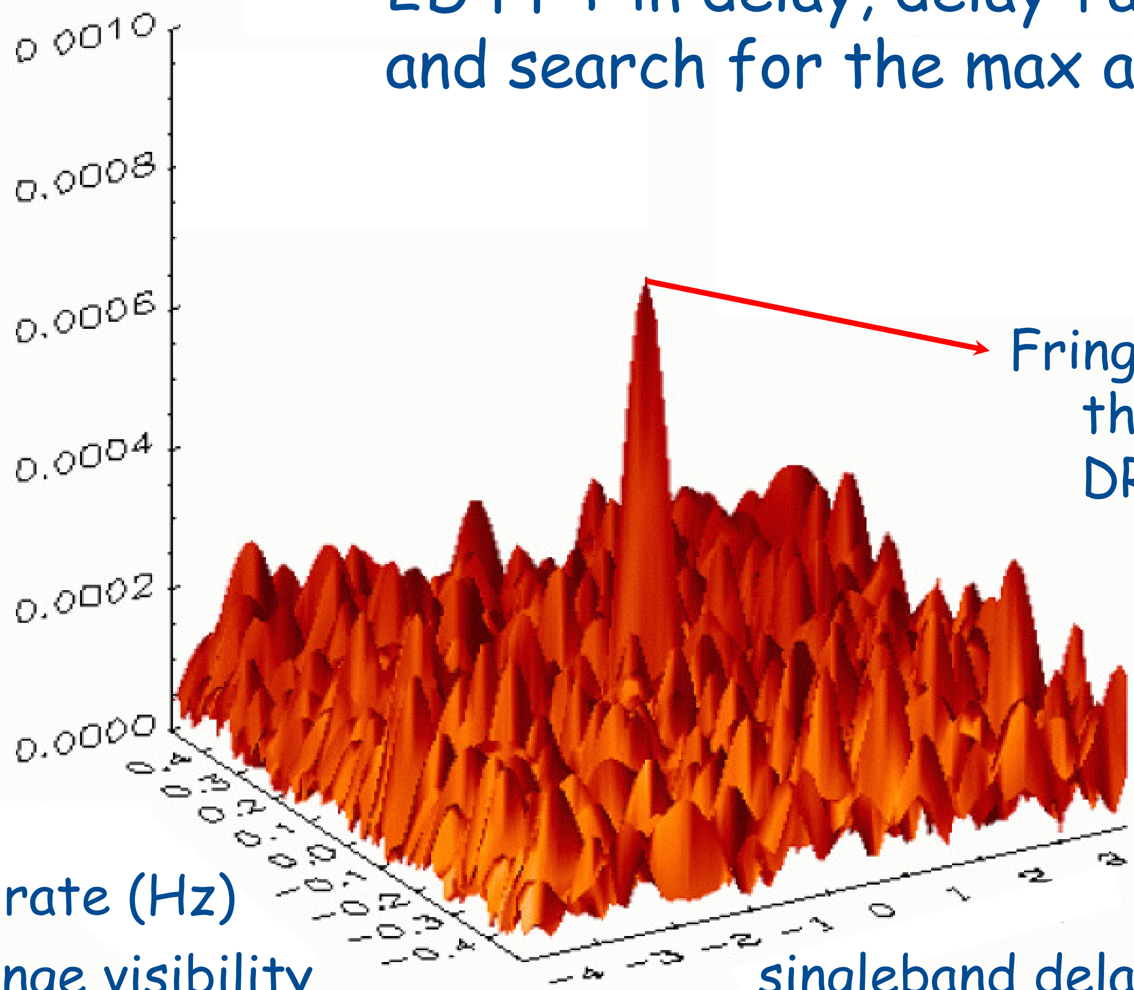
mb_win: mbd search window bounds (µs)

dr_win: delay rate search window bounds

Keep the parameters as above to have a huge window.
If not specified fourfit defaults to a small window !

Fringe Fit: Real Fringe Search

2D FFT in delay, delay-rate space and search for the max amplitude



Corr. amplitude

Fringes are located at this position in SBD, DR space

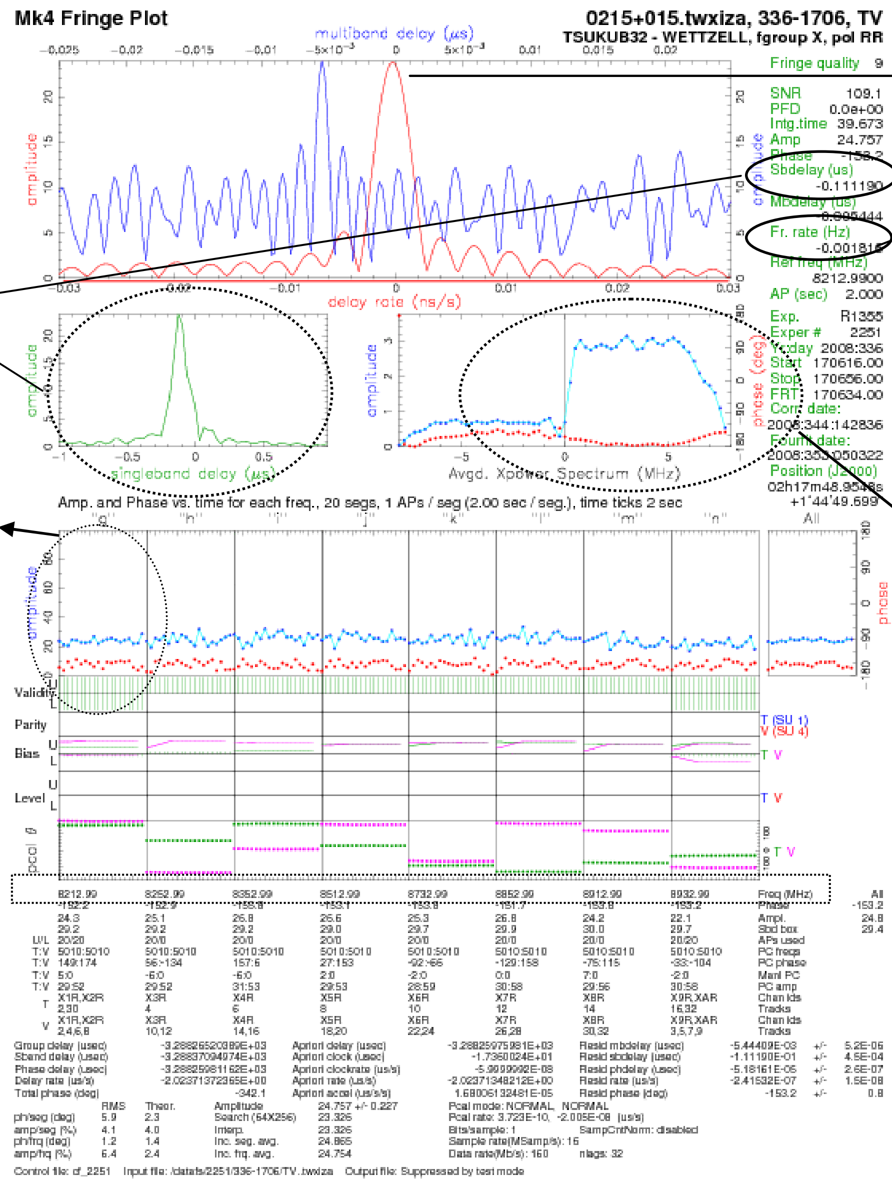
Fringe rate (Hz)

(FT fringe visibility with respect to time)

singleband delay (µs)

Image courtesy of K. Kingham

Fourfit



Delay rate.

Single band delay (µs)

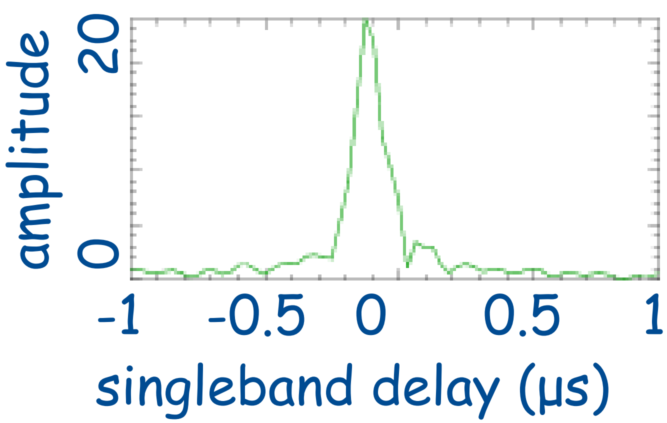
Fringe rate (Hz) = Delay Rate • Sky freq.

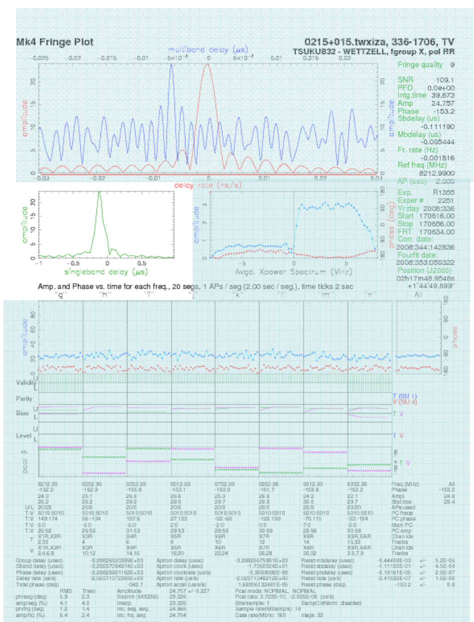
Phase (red) & amp (blue vs time for every BBC

FT of lag spectrum

Sky freq.

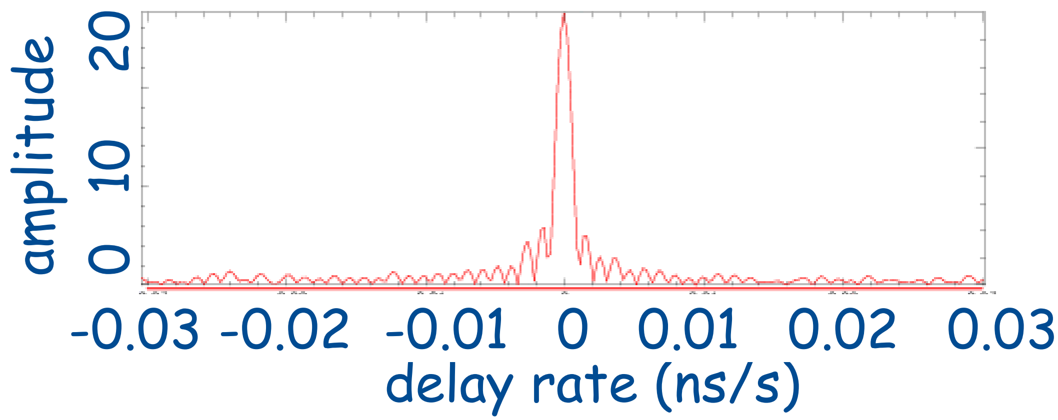
Fourfit: Lag Spectrum

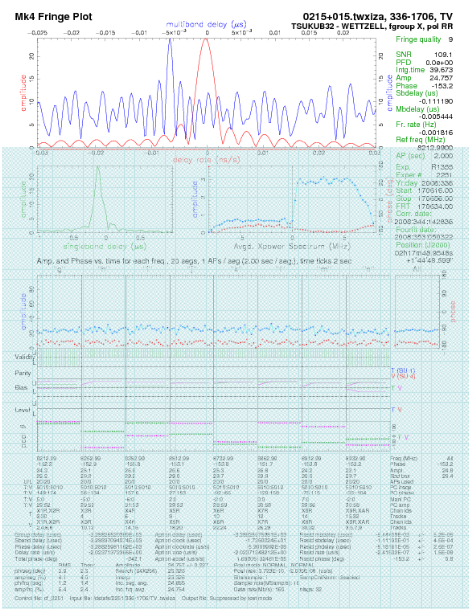




- Lag spectrum: output of the correlator integrated over the scan duration.
- Lag spectrum shown is lag spectra of all BBC stacked.
- 8 MHz/BBC => 16 Msample/s => sample period = 1 / 16 Msample/s = 0.0625 µs => 0.0625 µs * 32 lags = 2 µs SBD window width
- Indicates residual correlator model errors, part of which can be absorbed in the clock offset.

Fourfit: Delay Rate





FR is the Fourier transform of fringe visibility with respect to time.

DR = FR / Observing frequency.

DR window = [1 /(2 * AP)] / Obs. Freq.

DR tells how fast the fringes move away from the phase centre due to correlator model error. It can be absorbed in the clock rate.

Fourfit: Wordy Part

universitätbonn

0215+015.twxiza, 336-1706, TV

TSUKUB32 - WETTZELL, fgroup X, pol RR

Ref-Rem, Band, Polarization

SNR =

Peak amp / σ

Fringe quality 9

Depends on amp. & phase rms vs frequency and vs time.

SNR 109.1

PFD 0.0e+00

Intg.time 39.673

Amp 24.757

Phase -153.2

Mean visibility amp. & phase

Prob. of false detection. i.e. that a noise spike exceeds the signal amp.

Sbdelay (us) -0.111190

Mbdelay (us) -0.005444

Fr. rate (Hz) -0.001816

Residual SBD (μs)

Residual MBD (μs)

Residual FR (Hz)

Ref freq (MHz) 8212.9900

AP (sec) 2.000

Exp. R1355

Exper # 2251

Yr:day 2008:336

Start 170616.00

Stop 170656.00

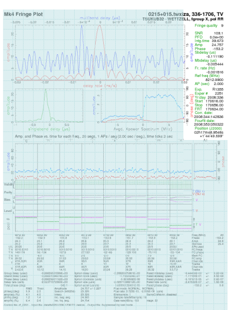
FRT 170634.00

Corr. date: 2008:344:142836

Fourfit date: 2008:353:050322

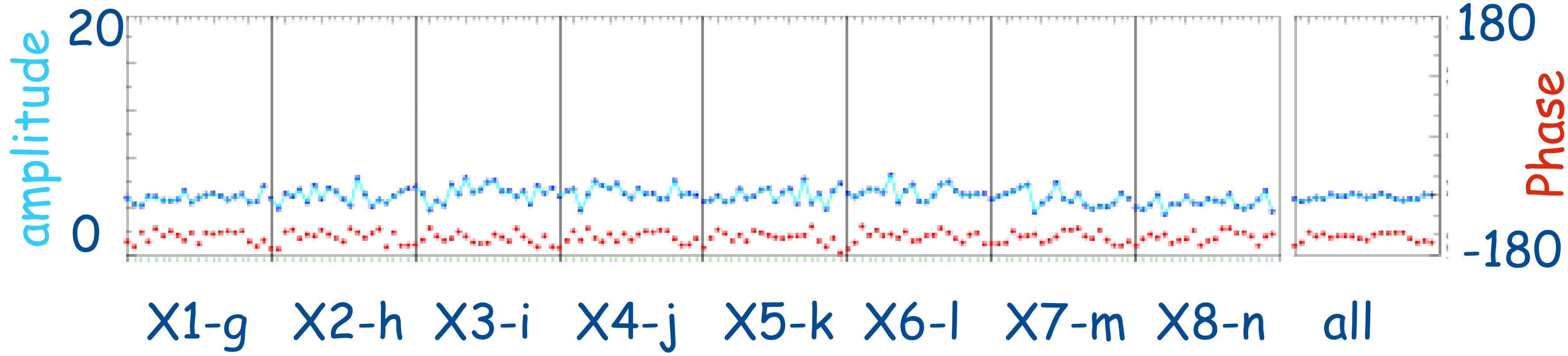
Position (J2000) 02h17m48.9548s +1°44'49.699"

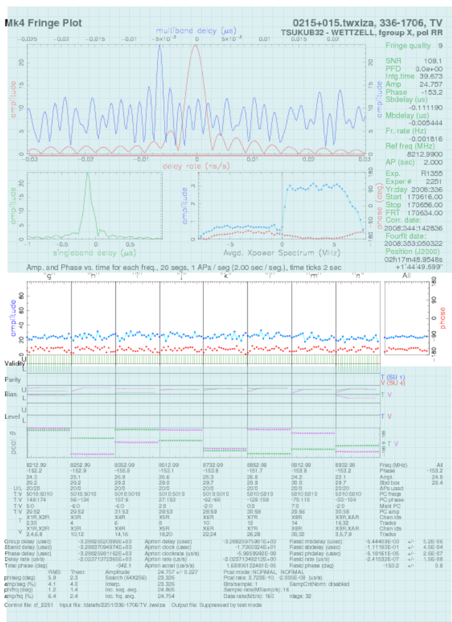
Accumulation Period length, Fourfit Reference Time, …



Fourfit: Amplitude & Phase vs Time

Amp. & Phase vs time (AP) for each frequency





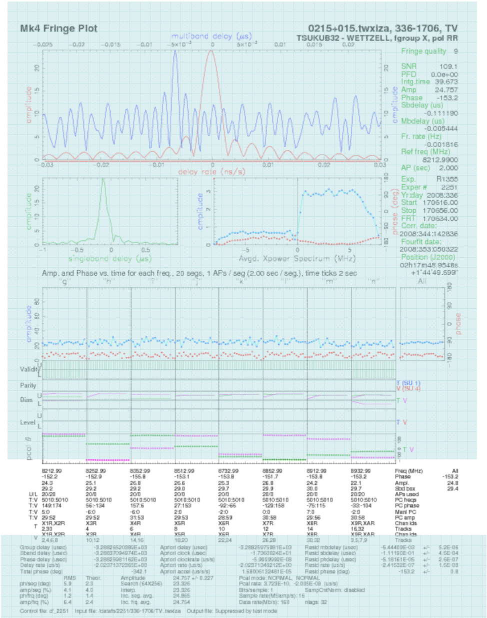
Every dot represents the phase (red) and amplitude (blue) of the visibility for every segment (~ AP).

Data are already fringe fitted and pcal has been applied.

Every BBC/VC channel is represented.

Fourfit: Wordy Part II

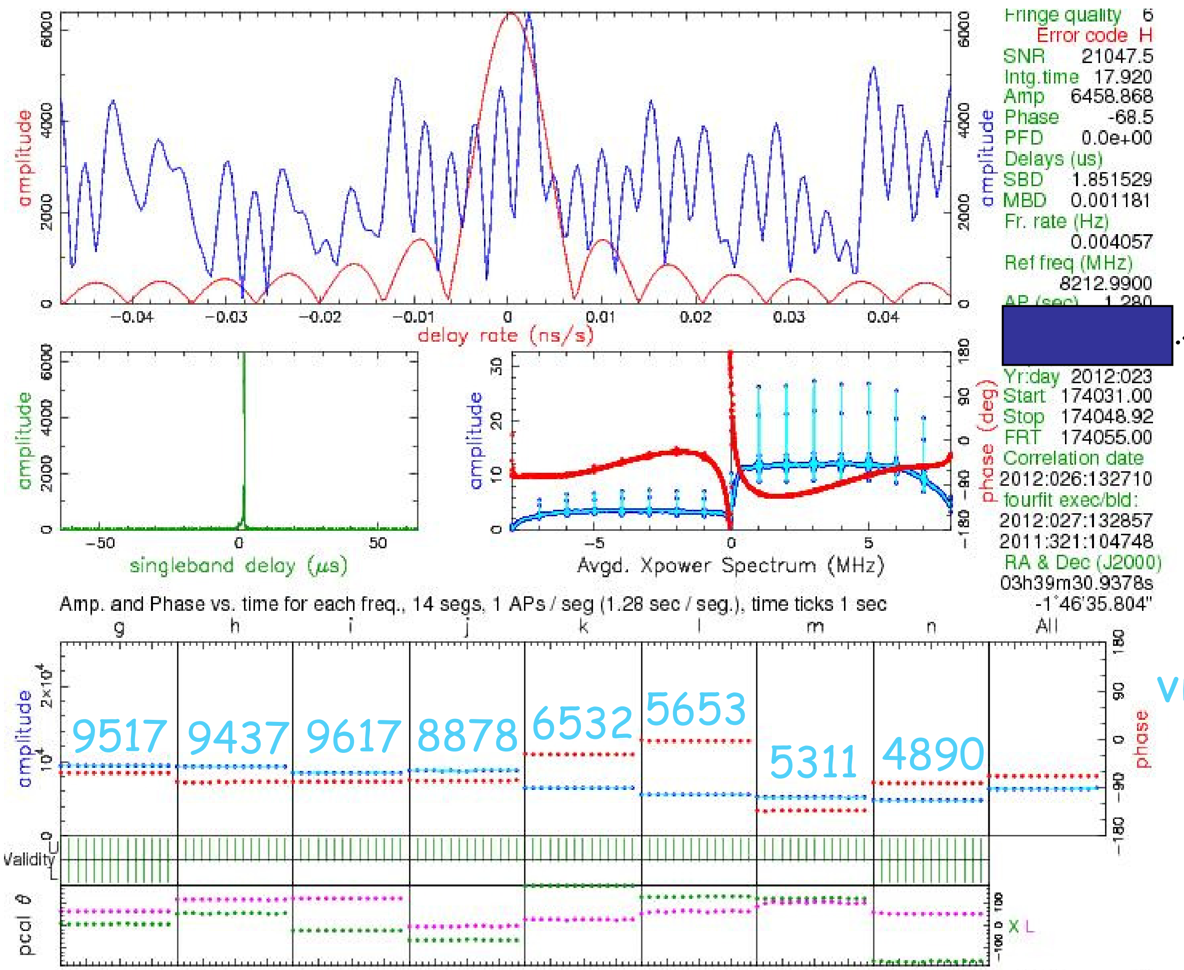
	8210.99	8220.99	8250.99	8570.99	Freq (MHz)	All
	-93.0	-95.7	-99.0	-96.7	Phase	-96.5
	291.1	304.7	308.8	301.2	Ampl.	301.9
	35.6	35.6	35.8	35.7	Sbd box	35.7
U/L	13/13	13/0	13/0	13/13	APs used	
B:N	2010:2010	2010:2010	2010:2010	2010:2010	PC freqs	
B:N	-145:143	-147:30	-14:69	-33:-172	PC phase	
B:N	0:0	0:0	0:0	0:0	Manl PC	
B:N	33:96	33:94	33:93	35:72	PC amp	
B	X1R,X2R	X3R	X4R	X9R,XAR	Chan ids	
	2,4,6,8	10,12	14,16	3,5,7,9	Tracks	
N	X1R,X2R	X3R	X4R	X9R,XAR	Chan ids	
	2,4,6,8	10,12	14,16	3,5,7,9	Tracks	



For zero-baseline tests you want to check the amplitudes: they need to be all about 10000 counts.

Any cause of SNR loss is a problem!

Fourfit: Example



	8212.99	8252.99	8352.99	8512.99	8732.99	8852.99	8912.99	8932.99	Freq (MHz)	All
	-62.5	-79.1	-78.5	-76.5	-27.4	-2.4	-132.5	-81.3	Phase	-68.5
	9517.1	9436.8	8616.9	8877.7	6532.2	5653.4	5311.3	4899.8	Ampl.	7326.2
	2108.2	2109.1	2109.2	2109.2	2108.7	2108.8	2108.6	2107.7	Sbd box	2108.2
U/L	14/14	14/0	14/0	14/0	14/0	14/0	14/0	14/14	APs used	
X	5010	5010	5010	5010	5010	5010	5010	5010	PC freqs	
L	5010	5010	5010	5010	5010	5010	5010	5010		
X:L	6:61	53:113	-24:119	-66:-5	175:25	127:60	120:99	-162:50	PC phase	
X:L	0:0	0:0	0:0	0:0	0:0	0:0	0:0	0:0	Manl PC	
X	25	24	25	25	22	24	23	23	PC amp	
L	21	19	20	18	6	2	3	7		

censored!

vis. amplitudes

No good!

Likely causes of SNR Loss

1) Internal clock calibration -> DBBC comes with a tool to perform it.

2) DBBC covers 0-500 MHz whereas Mark4 IFs cover 100-200 MHz and 220-500 MHz:

- if frequencies close to band edge are observed, DBBC could be sampling (@ 8-bits !) junk that is outside Mark 4 rack passband.

-amplifiers can become non linear devices: possible harmonics of the LO or IF input signal lie outside Mark 4 rack passband due to deliberately chosen IF bandwidth (Mark 4 IFs cut the first harmonic, whereas DBBC passes some harmonics).

3) There could be non-linearity in DBBC amplifiers before the A/D converter.

Copy raw data (~ 3 MB) onto file and check the data with linux command *od*:

`od -tx4` file name

output is like:

frame no.

time stamp:MJD & second of day

fractional second & header error check

```
0000000 abaddeed bead0001 0974ad5f f00abf01
0000016 0d645d49 57143f17 3a19c152 a0ec5b58
```

$\underbrace{\texttt{0000016}}_{\text{byte no. in file}}$ $\underbrace{\texttt{0d645d49 57143f17 3a19c152 a0ec5b58}}_{\text{data}}$

`od` byte no. in file

...

data

ABADDEED => header sync word (every 10000 bytes)

if lots of hex are zeroes -> no input to DBBC

Other Tools to Check Data

Use `mark5access` library (part of DiFX, but should be possible to install them as stand-alone):

`m5d`: decode data (valid for all data kinds that DiFX reads).

`m5test`: decode data headers and data (valid for all data kinds that DiFX reads).

`m5bstate`: state counts summary (valid for all data kinds that DiFX reads).

`m5spec`: forms total power for each baseband channel in the file (never used by me!).

Other Tools to Check Data

```
m5d /path/file.m5b Mark5B-256-16-1 10 ➔
Mark5 stream: 0x89e130
stream = File-1/1=/data10/r1/nyalesund/r1538_ny_171-1212a
format = Mark5B-256-16-1 = 2
start mjd/sec = 97 43922.000000000
frame duration = 312500.00 ns
framenum = 0
sample rate = 16000000 Hz
offset = 0
framebytes = 10016 bytes
datasize = 10000 bytes
sample granularity = 1
frame granularity = 1
gframens = 312500
payload offset = 16
read position = 0
data window size = 1048576 bytes
-1   1   1   1  -1   1  -1  -1  -1  -1   1  -1  -1  -1   1  -1
 […]
10 / 10 samples unpacked
```

Other Tools to Check Data

```
m5test /path/file.m5b Mark5B-256-16-1
Mark5 stream: 0x89e130
stream = File-1/1=/data10/r1/nyalesund/r1538_ny_171-1212a
format = Mark5B-256-16-1 = 2
start mjd/sec = 97 43922.000000000
frame duration = 312500.00 ns
framenum = 0
sample rate = 16000000 Hz
offset = 0
framebytes = 10016 bytes
datasize = 10000 bytes
sample granularity = 1
frame granularity = 1
gframens = 312500
payload offset = 16
read position = 0
data window size = 1048576 bytes
frame_num=2 mjd=97 sec=43922 ns=000625000.0 n_valid=2 n_invalid=0
 [..]
frame_num=335990 mjd=97 sec=44026 ns=996875000.0 n_valid=335990
1679990000 / 1679990000 samples unpacked
```

Other Tools to Check Data

`m5bstate /path/file.m5b Mark5B-2048-16-2` →

Ch	--	-	+	++	--	-	+	++	gfact
0	3937	2332	14736	3995	15.7	9.3	58.9	16.0	1.10
1	3921	8576	8552	3951	15.7	34.3	34.2	15.8	1.10
2	3968	8521	8580	3931	15.9	34.1	34.3	15.7	1.10
3	3833	8597	8651	3919	15.3	34.4	34.6	15.7	1.12
4	3857	8573	8628	3942	15.4	34.3	34.5	15.8	1.11
5	3951	8559	8518	3972	15.8	34.2	34.1	15.9	1.10
6	3947	8642	8416	3995	15.8	34.6	33.7	16.0	1.10
7	3991	8543	8525	3941	16.0	34.2	34.1	15.8	1.10
8	3961	8656	8430	3953	15.8	34.6	33.7	15.8	1.10
9	3934	8582	8531	3953	15.7	34.3	34.1	15.8	1.10
10	3896	8651	8615	3838	15.6	34.6	34.5	15.4	1.12
11	3909	8764	8458	3869	15.6	35.1	33.8	15.5	1.11
12	3971	8613	8531	3885	15.9	34.5	34.1	15.5	1.11
13	3988	8561	8370	4081	16.0	34.2	33.5	16.3	1.09
14	3844	8580	8679	3897	15.4	34.3	34.7	15.6	1.12
15	4002	8445	8581	3972	16.0	33.8	34.3	15.9	1.09

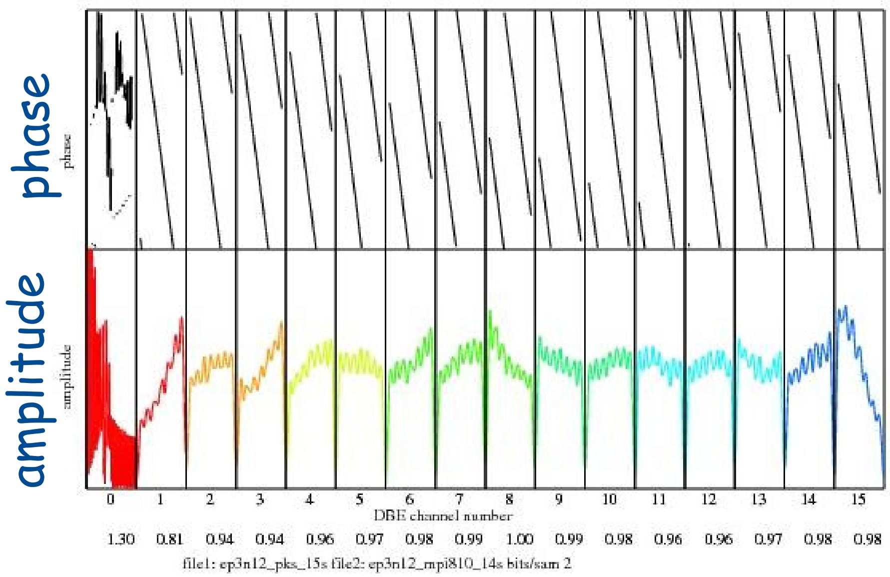
Other Tools to Check Data

Programs downloadable from Haystack:

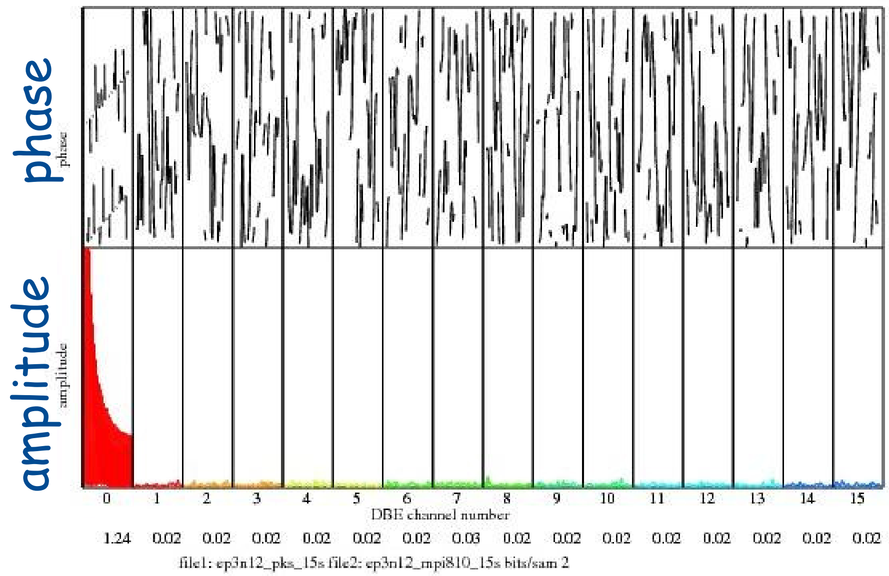
`vlbi2` only for 16-channels 2 bit sampling

`bstate` only for 16-channels 2 bit sampling

`vlbi0` only for DBE (or equivalent channel assignment)



Fringes



No fringes

Conclusion

DiFX is probably too complex to use to do zero-baseline testing, but required if correlating analogue (Mark IV) vs DBBC.

A solution could be to use DBBC only: inject same noise to two DBBC inputs: IF1 and IF2 tuned at the same frequency --> Cross-correlation between two DBBC cores. Then modified vlbi0 is probably the best option.

To check the data, mark5access library might be easier to use.