

plotpcal2

plotpcal2 is packaged the difx2fits package. It was introduced with DiFX-2.4 and currently only operates on DiFX output from DiFX-2.4 and later. It allows plotting of DiFX-extracted pulse cal tones in a number of ways. The program gets its input from the command line. The general structure of a command line is:

```
plotpcal2 [options] [operation] [tone selection] [data]
```

This utility is distributed within the difx2fits package.

[options] will eventually be more useful, but currently the only thing it can support is:

1. -lines

which replaces the scatter plot with a line plot

[operation] chooses what calculation to perform. Currently there are 4 options:

phase plot the phase of the tone(s) vs. time

amp plot the amplitude of the tone(s) vs. time

xy plot the real part vs. imag part

delay plot the delay as determined by differencing phases of 2 tones

[tone selection] is where things get complicated. For the first three operations, where a single tone is considered at a time the following syntax is currently supported. One or more of these can be included on the command line and they can be of different kinds. The operations that these are relevant for are amp, phase and xy.

rt=x,y "Record Channel Tone"

```
x is the record baseband channel (0-based)
y is the tone number in that baseband channel (0-based)
```

rf=x,y "Record Channel Freq"

```
x is the record baseband channel (0-based)
y is the frequency (MHz) of the desired tone
```

f=x "Frequency"

```
x is the frequency (MHz) of the desired tone(s)
Note: more than one tone may match. They will be orderd by record
channel
```

f=%x "Frequency mod x"

Select all tones from all recorded channels where
frequency mod $x = 0$ (MHz)

$r=x$ "Record channel"

x is the record baseband channel (0-based)
Note: all tones in that channel will be selected. Usually for USB the tones will be in order of increasing frequency and for LSB the tones will be in order of decreasing frequency

$t=x$ "Tone number"

x is the tone number in that baseband channel (0-based)
For all record channels, tone x will be selected. The ordering is by record number (starting with 0)

For the delay operation two tones are differenced. They must come from the same record baseband channel. The following tone selections can be used:

$rt=x,y,z$ "Record channel, tone 1, tone 2"

x is the record baseband channel (0-based)
 y is one tone from that record band (0-based)
 z is the other tone from that record band (0-based)
Delay is calculated from
$$(\text{Phi}(\text{tone1}) - \text{Phi}(\text{tone2})) / (\text{Freq}(\text{tone1}) - \text{Freq}(\text{tone2}))$$

$rt=x,\text{ends}$ "Extreme channels from one record channel"

x is the record baseband channel (0-based)
Selected tones are the highest and lowest frequency tones from the band. Sometimes this is not good because tones near the edge of the band can be corrupted by filter response / aliasing.

$rt=x,\text{good}$ "Channels near the end of one record channel"

x is the record baseband channel (0-based)
Selected channels try to be 1/8 of the band away from the edge frequency when possible.

$rt=x,\text{pairs}$ "Select pairs of channels across one record channel"

x is the record baseband channel (0-based)
For a channel with $n\text{Tone}$ tones, form $n\text{Tone}/2$ delays by considering first and second tones for first delay, third and fourth for second delay, ...

$t=x,y$ "Tone 1 and Tone 2 from all channels"

Like $rt=x,y,z$ but for all record channels. Will produce $n\text{Channel}$

```
delays ordered by record channel number.
```

t=ends "Extreme channels from all record channels"

```
Like rt=x,ends but for all channels. Will produce nChannel delays
ordered by record channel number.
```

t=good "Channels near the end of all record channels"

```
Like rt=x,good, but for all channels. Will produce nChannel delays
ordered by record channel number.
```

[data] is a list of DiFX derived pulse cal data. When produced they come from the .difx/ directories produced by DiFX. There will be a separate file per antenna per job.

When plotpcal is run it first sorts data by antenna name and makes one plot per antenna using the python matplotlib plotting package. All standard matplotlib features are available.

Canned Example

First [grab some data](#) (or use data you collected yourself)

Amplitude of a single tone (record channel 2, tone 0)

```
plotpcal amp rt=2,0 *.difx/PCAL*
```

Things to note: first sample is low, then three pairs of scans, each pair being a 5 MHz spacing followed by a 1 MHz spacing. 5 MHz spacing should have 5 times the voltage (as plotted) or 25x the power.

Note: there are two stations: HN and SC. When you close the first (HN) window, a second window will appear shortly for SC. Kill that to end the program.

Note: you can zoom, ...

Amp of 2 tones:

```
plotpcal amp rt=2,0 rt=4,0 *.difx/PCAL*
```

Amp of 2 tones (but only one tone for the 5 MHz mode):

```
plotpcal amp rt=2,0 rt=4,2 *.difx/PCAL*
```

Phase of 2 tones:

```
plotpcal phase rt=2,0 rt=2,1 *.difx/PCAL*
```

Note: tone 0 and tone 1 differ in actual frequency between two setups.

Phase of 2 tones on 5 MHz steps

```
plotpcal phase rf=0,%5 *.difx/PCAL*
```

Plot re vs im for all tones for one record band

```
plotpcal xy r=5 *.difx/PCAL*
```

Plot one delay for each record channel

```
plotpcal delay t=good *.difx/PCAL*
```

Note how much more precise (factor of 5ish) the 5 MHz pulse calcs are!

Except for 2 baseband channels from HN (cyan on top and magenta on bottom, corresponding to record channels 6 and 7), delays from a single IF match well, as expected. The mismatch here is curious.

From:

<http://www.atnf.csiro.au/vlbi/dokuwiki/> - **ATNF VLBI Wiki**

Permanent link:

<http://www.atnf.csiro.au/vlbi/dokuwiki/doku.php/difx/plotpcal>

Last update: **2017/10/20 00:21**

