

fourfit user's manual

Version 1.0

Roger Cappallo

MIT Haystack Observatory

September 18, 2019

Contents

1	Introduction	2
1.1	Why fringe-fit?	2
1.2	Correlator windows	2
2	History	3
3	Program control	3
3.1	command line	3
3.2	control file	3
3.3	set mechanism	4
4	Program output	4
5	Architecture	4
5.1	control files	4
5.2	fringe search	5
5.2.1	grid search	5
5.2.2	interpolation	5
5.2.3	complex phase factor	5
5.2.4	program structure	6
6	Algorithms	6
6.1	Ionosphere	6
6.1.1	Smoothing	7
6.2	Combining Linear Polarization Products	7
7	Fringe plots	8
	Appendix A Command line options	9
	Appendix B Control file keywords	11
B.1	Selector keywords	11
B.2	Syntactic keywords	11
B.3	Action keywords	12
B.4	Keyword semantics	13
B.4.1	Scan Selection	13
B.4.2	Filtering – <i>determine whether or not each AP is accepted</i>	13
B.4.3	search – <i>control the fringe-searching process</i>	14
B.4.4	corrections – <i>apply corrections to the data, either before or after fit</i>	14

B.4.5	switches and other miscellaneous functionality	16
B.4.6	other symbols	16
B.4.7	Special Formats	16
B.4.8	General Guidelines	17
Appendix C	Multitone phase-cal	17
C.1	Introduction	17
C.2	Multitone Features	17
C.3	Mathematical Underpinnings	18
C.4	Syntax and Usage	18
C.5	Examples	19
Appendix D	File-based phase corrections	19
Appendix E	Mk4 file system	20
Appendix F	Subversion history	20
Appendix G	Adding new parameters	20
Appendix H	Calling graphs	21

1 Introduction

This manual is intended to be useful for a varied audience, which will include both users and developers. Hopefully it is organized enough so that any specific user can find the information they need, although of course the ultimate arbiter is the source code, which should be consulted for further detail.

1.1 Why fringe-fit?

One might reasonably ask why fringe-fitting is even necessary, given that the data have already been "correlated". The reason is that correlation is done with a very specific model, which incorporates information about source coordinates, station coordinates, epoch, frequency sequence, polar motion, earth rotation, clock-settings, etc. Many of these parameters contain significant errors, which will manifest themselves as non-zero residual delay and phase, varying in time. It is the task of fringe-fitting to remove as much of this residual signal as possible by estimating corrections to the intermediate quantities of group delay and delay rate.

Once corrections to delay and rate are estimated, they can then be recombined with the delay and rate predicted by the model, in order to yield a "total" observable. This total is relatively insensitive to errors in the original model. For example, if we define O as the observed (true) value and C as the computed model value, then the residual data value is $O - C$, and we have the relation $O = C + (O - C)$. Small changes to C are canceled out by compensating changes in $O - C$, at least over the region where changes remain linear.

1.2 Correlator windows

Not only is it desirable to keep the model as close to reality as possible, in order to stay within the linear region, but there are also some natural limitations inherent in the correlation process, which cause a finite range of possible residual values around the model. For example, a lag correlator has a finite number of

lags, and there is a maximum lag between the two datastreams, typically of order of 1 or 2 μ s. The models must agree with reality to this level, or the correlated signal will fall outside the range of correlation. An FX correlator has similar limitations. If the delay rate residual is too large, then the fringe phase may wrap around a half or a full turn per accumulation period, causing loss of fringes or weakened fringes, respectively. The natural multiband delay window is set by the spacing of the frequencies of the channels. It is the inverse of the GCD of the all of the channel frequency differences.

For all three windows the natural window can be further restricted in the post-processing. This is done in order to speed up the fringe search, or possibly to restrict the range of a parameter based upon externally-supplied information.

2 History

In the late 70ÅTs Alan Rogers developed a program called FRNGE derived from the original VLBI2 fringe-fitting program. It was written in fortran and designed to be efficient on an HP-21MX (later renamed to HP-1000) minicomputer. This computer had only 32K x 16 bit words, and was extremely slow by 21st century standards. Just performing the fringe-fitting on small scans in several minutes time was a real tour de force.

As both hardware and software technology improved it became apparent that FRNGE would need to be rewritten, and so fourfit was created by Colin Lonsdale, Roger Cappallo, and Cris Niell in the early 90ÅTs. The basic algorithms were adopted directly from FRNGE, while the I/O, control architecture, and plotting were heavily modified. The programming language was changed from fortran to C.

3 Program control

3.1 command line

The *fourfit* command line has a number of options, which are detailed in Appendix A. Depending on the chosen options, each execution of *fourfit* could process any combination of:

- one or more scans
- one or more baselines
- one or more frequency bands
- one or more polarization products

The most commonly used options include -t to invoke test mode, which doesn't create any output files, -b to specify a particular baseline to process, -p to pop up a fringe plot on the screen, and -c to specify a control file name.

3.2 control file

Control files are the principal means by which *fourfit* is usually controlled. There is normally one control file created per experiment, with finer control of processing provided within an experiment by using optional selectors, such as specifying a time range or particular scans, within the control file. Control file information can in turn come through three different routes, in the order they are read:

1. there is the capability (not normally used) of having a default control file. If it exists, it is accessed by an environment variable called `$DEF_CONTROL`. It might prove useful if the program defaults need to be changed on a regular basis.
2. a control file that may be referenced through the `-c` option of the *fourfit* command line
3. a convenience feature that allows immediate use of control file commands on the run line (see **set** below)

If the same parameters are set in more than one location it is the last setting that is used.

3.3 set mechanism

As explained above, the final control file input to the program is anything on the command line following **set**, if that is present. Since it comes last, the information - if it matches the selector criteria - overwrites any corresponding earlier information. For example, the experiment control file might specify the singleband delay range over which to search, but one could override it by appending **set sb_win 0.56 0.56**, which would force it to use the value of 0.56 μ secs. This mechanism provides a convenient way to see the effect of quick changes to parameters, since it avoids having to edit the control file.

4 Program output

At the end of each fringe fit data are written to type 2 files within the current fileset, unless the program is run in test mode with `-t`. These files contain all of the information relevant to the fringe fit, and become the input for subsequent astronomic or geodetic processing. Optionally, a fringe plot may be popped up in a window, or perhaps printed, in order to facilitate data quality analysis.

5 Architecture

5.1 control files

Internally, the control files are implemented as a chained list of structures, which are created and appended in the order that the control file information is encountered by the program (i.e. default control file first, then the main control file, then any **set** commands.) The control block structures each have four selectors, which control whether or not a particular block will be applied to data. The selectors are (baseline, source, frequency group, and time range), with wildcard matching allowed. A fifth "pseudo-selector" is a single-character station code such as K. It is expanded internally into two identical parameter blocks, one matching baseline K? and the next matching ?K. In addition to the selectors there are many parameters that can be applied to the data, such as filtering parameters, calibration corrections, or search ranges.

The control file is parsed by a finite-state machine, which can at times react to errant input files so as to produce syntax error messages that are a little hard to interpret. The offending input line is always printed, though, to help the user zero in on the problem.

The principal routines used in the control file parsing are:

- `create_fsm` – sets up a structure with the finite-state-machine information containing the semantics of the control file commands
- `parse_cmdline` – parses the command line options

- `parse_control_file` – high level routine that reads the control file, sets up the fsm, uses lex, and invokes the parser
- `lex` – does a lexical analysis, changing keywords into tokenized information
- `parser` – does the heavy lifting of applying the fsm semantics to the tokenized output of lex, in order to create `c_blocks`

5.2 fringe search

The fringe search in *fourfit* is a two-stage process:

1. Search over a large 3-dimensional grid of (singleband delay, multiband delay, delay-rate) to find the approximate location of the maximum of the delay resolution function.
2. Form a 5x5x5 cube of correlation amplitude values centered on the location found in the grid search, and perform a 3 dimensional interpolation to find the peak value. Note: this section describes the *simultaneous* interpolation, not the now-deprecated *iterative* interpolation.

5.2.1 grid search

A search is made through a large number of grid points in a 3-dimensional volume with axes of singleband delay, multiband delay, and delay-rate. The function being maximized is the coherent sum of all of the correlator output complex visibility points over frequency and time. The grid points are evenly spaced, which allows the search to be done efficiently via FFT's. For example, along the delay-rate axis the transform of the complex phasors converts from a function of time to a function of fringe frequency. The best fit can be found by inspecting the fringe frequency values to find the maximum. Similar conjugate searches in the delay domains are done: the cross-power spectral points within the channels determine the singleband delay peak, and the phases across the different channels determine the multiband delay peak.

5.2.2 interpolation

Once the location of the maximum grid point is known, then an interpolation step refines the values in the neighborhood of that (singleband delay, multiband delay, delay-rate) triple. In the interpolation step a direct summation of the complex visibility values is performed, by rotating the data by a complex phase factor (see next section). This factor incorporates the effect of the offsets along each of the three axes of a 5x5x5 cube, which is centered on the values (sbd, mbd, dr) from the grid search. After the cube is formed, the 3 coordinates of the maximum of the visibility magnitude is found via iterative application of a simultaneous 3-D interpolation. The iteration step is implemented by searching for the maximum of an interpolated 11x11x11 cube of visibility magnitudes. Each iteration has a 11x11x11 cube that is a factor of 5 smaller, until all 3 coordinate spacings are less than 0.0001 of the original 5x5x5 spacing.

5.2.3 complex phase factor

In the interpolation of the best-fit (mbd, sbd, dr) values and also in the plotting of the data as fit by the program, there is a complex rotation applied to the data. This factor has the form $e^{-i2\pi\theta}$, where θ is given by

$$\theta = f\Delta t\hat{\tau} + (f - f_{ref})\tau_{mb} + b\tau_{sb} \quad (1)$$

where:

f	frequency of a data point
f_{ref}	reference frequency
Δt	offset in time of data from ref. time
$\dot{\tau}$	residual delay rate
τ_{mb}	residual multiband delay
τ_{sb}	residual singleband delay
b	(0.25, 0, -0.25) for (usb, dsb, lsb) data

5.2.4 program structure

The overall program structure can be seen in the calling graphs of Appendix H.

6 Algorithms

6.1 Ionosphere

In the S/X geodetic system independent group delay estimates are made for X-band and S-band, and a linear combination of these delays, $\tau_g = 1.081\tau_x - 0.081\tau_s$, is used to give a group delay measurement that is largely free of ionospheric effects. One drawback of this system is that both the S and X band observations have to have a high enough signal-to-noise ratio so that a good detection can be made in each band. In contrast, the VGOS system is designed to work with very short scans, in which the individual bands may not provide a reliable detection, and in which all 4 bands need to be combined coherently in a fit. Since this fit also spans a wide frequency range, it is necessary to fit and remove the differential ionosphere from the group delay estimate.

The phase contribution in radians due to the ionosphere, $\Delta\phi$, is

$$\Delta\phi = -8.448 \times 10^9 \Delta TEC / f \quad (2)$$

where f is the observing frequency in Hz, and $\Delta TEC \equiv TEC_a - TEC_b$ is the differential TEC for baseline ab in TEC units ($1 \text{ TEC unit} = 10^{16} \text{ elec} / m^2$). *NOTE! This definition of the sense of the differential quantity as reference minus remote is (unfortunately) the opposite of the standard used throughout the rest of fourfit.* Since phase is only measured modulo 2π , there is a non-linear dependence of phase upon the ΔTEC parameter in the model, which restricts the manner in which a fit might be performed.

The original algorithm used was that a range of coarse values were specified, the maximum within the range was found, then a second pass fine search was done around the maximum. Recently the *fourfit* code has been modified in a modest way, in order to increase the ionosphere search to be 3 level - (coarse, medium, fine). The intent is to allow efficient searches over a wider range of ΔTEC .

It is currently parameterized to allow a flexible coarse search, though it is recommended that the user employs a 4 TEC unit spacing of the grid points over a wide range of ΔTEC values. The fringe fit in Figure 1 used:

- coarse: 11 points from (-70, -30)
- medium: 11 points with 2 unit spacing = +/- 10 units about the coarse peak
- fine: 11 points with 0.4 unit spacing = +/- 2 units about the medium peak
- final: parabolic interpolation of three fine points around the fine point peak

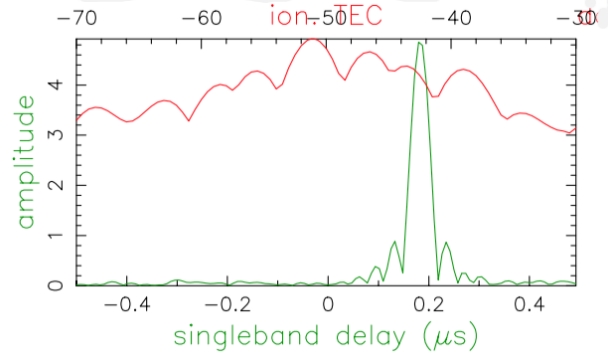


Figure 1: Ionospheric fit - correlation amplitude as a function of differential TEC in TEC units (red curve). A complete fringe fit at multiple points for trial values of the ionospheric differential TEC is performed, and the code finds the interpolated maximum.

The medium and fine spacings are automatically generated; as before, the user only supplies the coarse range and npts. Though the efficiency of this algorithm could possibly be improved, it has proven a very robust mechanism for finding the maximum value.

6.1.1 Smoothing

There is an option, invoked by `ion_smooth` true (default: false) in the fourfit control file. If invoked, the code will do a 2-pass (as opposed to 3-pass) search of TEC space, looking for the amplitude maximum. The coarse spacing is determined by `ion_win` and `ion_npts`, as before:

$$\text{coarse_step} = (\text{ion_win}[1] - \text{ion_win}[0]) / (\text{ion.npts} - 1)$$

The smoothing code takes the grid points from the coarse search and performs a convolutional smoothing, using a half-cycle of a cosine function as the smoothing kernel. It then finds the maximum of the smoothed, interpolated new points. Execution of this step takes negligible time.

A final fine search is then done about the smoothed maximum, of double width (of the non-smoothed fine search - i.e. 24 pts) at 0.4 TEC unit spacing.

The intent of this code is to make use of the information present over a broader range of TEC's than just the peak. In some cases a little noise might make one (unsmoothed) peak higher than another, but with smoothing the peak will be brought up by the strength of the surrounding points, at least in theory.

6.2 Combining Linear Polarization Products

In order to achieve reasonably good feed characteristics across the full frequency range, the VGOS design uses linearly polarized feeds. This adds significant complexity to the signal processing tasks, as all 4 polarization products must be produced and then combined in an optimal fashion to estimate group delay. A pseudo-Stokes I (Intensity) observable can be formed, which is accurate to first order in the polarization leakage D-terms (Corey 2011). The pseudo Stokes-I observable is formed as

$$I = (\overline{X_a \star X_b} + \overline{Y_a \star Y_b})\cos(\delta p) + (\overline{X_a \star Y_b} - \overline{Y_a \star X_b})\sin(\delta p) \quad (3)$$

where δp is the differential parallactic angle between sites a and b , and $\overline{X_a \star Y_b}$ is (for example) the time-averaged correlation product of site a 's X polarization with site b 's Y polarization.

7 Fringe plots

Fringe plots are a very useful tool for the analyst to examine a single scan for possible defects. There is a wealth of information on a single page, which though daunting at first, can prove a very efficient use of the human ability to see graphical patterns.

Appendices

A Command line options

Command line syntax:

```
fourfit [-a] [-b BB:F] [-c controlfile] [-d display device] [-f value] [-m value] [-n value]
[-p] [-r afile] [-s naps] [-tux] [-P polar_pair] [-T trefoffs] [-X] data file list [set <control
file syntax statements>]
```

All arguments except the data file list are optional. The data file list is required, unless the [-r afile] option is used, when the afile replaces the data file list. If present, the "set" argument and the commands which follow it must come last. All option flags must appear before the data file list, though the flags can come in any order.

Here are two examples of command-line invocations of fourfit, with an explanation of what they do:

```
fourfit -txas -m 1 -c control 018-234505 set mb_win -0.0034 .004 freqs a b
```

Test mode, xwindow display, accounting switched on, cross power spectrum plot switched on, moderately verbose, use control file named "control" in current working directory, process all data in scan directory 018-234505, override multiband delay search window and select channels 'a' and 'b' only.

```
fourfit -r refr_list -c control -d hardcopy -b AT:S
```

Process all data referenced by type 2 lines in the A-file named "refr_list", use control file "control", print the fringe plot on the default printer, process only baseline AT frequency subgroup S.

Option Flags:

-a If specified, this option switches on accounting of CPU time and wall-clock time used in the various parts of fourfit. When the program finishes, it produces a summary of these timing statistics.

-b BB:F Allows the user to override the control file with a specification of the baseline and/or frequency group to be processed. The syntax is flexible. 0, 1 or 2 characters before the colon refer to the baseline (one character is interpreted as a station), and 0 or 1 character after the colon is interpreted as the frequency subgroup. You can use the control file wildcard character '?' in the baseline, but remember to protect it from the C-shell either by escaping it with a backslash ' \' or enclosing the entire -b argument in single quotes. If you wish only to specify the baseline, the colon may be omitted. An error in the -b flag argument causes the flag to be ignored, and fourfit will continue execution.

-c controlfile Specifies the file which contains parameters to control the operation of the program. If absent, fourfit will use only the file pointed to by the environment variable DEF_CONTROL, which in turn defaults to /dev/null as defined in the HOPS/setup.csh file. Any parameters set in a control file specified with the -c option override the default file values. A description of the syntax of the control file, with an example, can be found later in this document.

-d display_device Upon completion of a fringe fit, fourfit can optionally display the results using postscript. The valid choices for "display_device" are:

diskfile:file.ps save the plot in "file.ps"

hardcopy send the plot directly to lpr

pshardcopy print the plot via pplot_print

xwindow show the plot in an X11 window

psscreen the same, but allow GS_* options

- f first_channel overrides the default first channel (0) to facilitate plotting when there are more than 16 channels (see -n)
- m value This flag controls the verbosity of the program via the integer argument "value", which ranges from 3 (virtually silent except for major errors) to -3 (incredibly verbose, of use only to the authors of the program). The default is 2.
- n nchans When there are many channels (see also the -f flag) this tells how many channels to put on the plot. Note that neither -f nor -n affect the actual fringe fit, just the plotting thereof.
- p Simplest way to pop up a fringe plot. This is equivalent to "-d psscreen".
- r afile Puts fourfit into "refringe" mode. Fourfit refringing is driven by an A-file input, which overrides any correlator root files directly specified on the command line (i.e. the latter are ignored when the -r flag is specified). The input A-file, of which there can be only one, may contain root, corel or fringe lines, but only the fringe lines are used to determine which data to process, by baseline and frequency subgroup. Obviously, the -r flag is inconsistent with the -u flag, and specifying both is an error. Note that for afiles using HP-1000 (version 1) line formats, fourfit has to pre-check the disk for the existence of the type 2 files. The data area is controlled by the DATADIR environment variable. It defaults to the value of \$CORDATA.
- s naps This parameter controls how many AP's are merged together into each plotting segment. Thus the number of time points shown in the phase, amplitude, and validity plots is so controlled. Additionally, the ph/seg and amp/seg statistics are calculated based upon the stated number of AP's in each segment.
- t This flag places fourfit in test mode. Everything works as normal, except that the output file is not written to disk, and the root file is not updated. This is useful when experimenting with different fringe-fitting strategies, in order to avoid cluttering up the disk.
- u Normally, fourfit processes all data consistent with the data file list and the control information. When this flag is specified, fourfit will also check the information in the type-2100 record of the root to see if the data have already been processed by fourfit. If so, the data in question are skipped. The "u" stands for update mode.
- x This is equivalent to "-d xwindow".
- P pq Controls polarization processing. If pq is a 2 character string, then pq is one of sixteen cross-polarization products, formed by p and q each being one of {R, L, X, Y}. Alternatively, pq can be a single character, 'I', which forms the Stokes pseudo-I mode combination of the products {XX, YY, XY, YX}.
- T trefoffs If this option is invoked, the fourfit reference time will be calculated by taking the nominal scan start time from the ovex file and adding trefoffs (which is an integer number of seconds) to it.
- X Forces fourfit to write cross-power spectra into type 230 records. This option is typically used for import into AIPS.

Arguments:

data file list This mandatory argument or arguments tells fourfit which data files to process. The format of the data file specification is the standard one for all MkIV software. You may specify individual filenames, scan directories which contain data files, experiment directories, or any combination of these three. In the latter two cases, fourfit will descend the directory tree looking for data files to add to its internal list of files to process. Only root files need be specified. The data files actually fringe-searched are determined by the combination of the root files specified and the restrictions imposed by the control file or control parameter

list (see below). In the absence of such restrictions, all data associated with the specified root files are processed.

Beware of trying to specify too many files or scan directories, as it is possible to overflow the Unix argument list buffer on large experiments. In such cases, specify the experiment directory instead.

B Control file keywords

B.1 Selector keywords

Keyword	Value	
station	1 character	
baseline	2 characters	
source	string of 1-8 chars	
f_group	1 character	Notes:
scan	UT-epoch (special format), or: < UT-epoch > UT-epoch UT-epoch1 to UT-epoch2 (inclusive time range)	

1. The wildcard character '?' can be used for any or all characters within the baseline, the source name, or the frequency group. So long as the other, non-wildcarded characters all match the scan will be selected.
2. The station selector keyword is provided for the user's convenience, but internally within the program there is no station keyword. Internally, each use of the station keyword results in two matching control blocks, so that if station Z (commands) is equivalent to if baseline Z? (commands) followed by another block with if baseline ?Z (same commands). This subtlety is not normally apparent to the user, but could arise, e.g., if one tried some compound condition involving both a station and a baseline, such as:

if station Q and baseline QC

Such combinations are probably best avoided.

B.2 Syntactic keywords

```

if
else (not yet implemented)
and
or
not
( ) (not yet implemented)
<>
to
?
```

B.3 Action keywords

Keyword	Value		
adhoc_amp	float		
adhoc_file	string		
adhoc_file_chans	string		
adhoc_period	float		
adhoc_phase	'sinewave', 'polynomial', or 'file'		
adhoc_poly	≤ 6 floats/integers (mixture OK)		
adhoc_tref	float	pc_phases	n char string, followed by n floats
dc_block	'true' or 'false' (def: false)	pc_phases_l	n char string, followed by n floats
dec_offset	float	pc_phases_r	n char string, followed by n floats
delay_offs	n char string, followed by n floats	pc_phases_x	n char string, followed by n floats
dr_win	2 floats	pc_phases_y	n char string, followed by n floats
est_pc_manual	int	pc_freqs	n char string, followed by n floats
fmatch_bw_pct	float	pc_mode	'normal', 'ap_by_ap', 'manual', or 'multit
freqs	n chars	pc_period	int
gates	n char string, followed by 2n floats	pc_tonemask	n char string, followed by n ints
gen_cf_record	'true' or 'false' (def: false)	ra_offset	float
interpolator	'iterate' or 'simul' (def: simul)	ref_freq	float
ionosphere	float	samplers	int, followed by up to 8 strings
ion_npts	int (def: 0)	sampler_delay_l	up to 8 floats
ion_smooth	'true' or 'false' (def: false)	sampler_delay_r	up to 8 floats
ion_win	2 floats	sampler_delay_x	up to 8 floats
lsb_offset	float	sampler_delay_y	up to 8 floats
mb_win	2 floats	sb_win	2 floats
mbd_anchor	'sbd' or 'model' (def: model)	skip	'true' or 'false'
notches	2n floats	start	integer
optimize_closure	'true' or 'false' (def: false)	station_delay	float
passband	2 floats	stop	integer
period	int	switched	'scan_start' or 'each_minute'
pc_amp_hcode	float	t_cohere	float
pc_delay_l	float	use_samples	'true' or 'false'
pc_delay_r	float	weak_channel	float
pc_delay_x	float		
pc_delay_y	float		
vbp_correct	'true' or 'false' (def: false)		
vbp_fit	'true' or 'false' (def: false)		
vbp_coeffs	5 floats/integers (def: 0.0)		
vbp_file	string		

– *deprecated commands – for backward mk4 compatibility only:*

index	n ints - alternate method used to select data channels, based on the original corel index number. Not as well supported as freqs, and is currently being phased out.
max_parity	float (tape only)
x_crc	'keep' or 'discard' (tape only)
x_slip_sync	'keep', 'discard', or an integer (tape only)
y_crc	'keep' or 'discard' (tape only)
y_slip_sync	'keep', 'discard', or an integer (tape only)

B.4 Keyword semantics

B.4.1 Scan Selection

skip – if this is set to true in the body of an `if` block, then any scans matching the if conditions will be skipped. Note: as of 99.2.19 *fourfit* will not properly skip data if `f_group` is specified.

B.4.2 Filtering – *determine whether or not each AP is accepted*

freqs controls which frequency channels get included in the fit. The letters a-p correspond to the order that the frequencies appear in the root file (assuming 16 channels). With no suffix, DSB is implied, if both sidebands are present. A plus suffix denotes USB, a minus is used for LSB. After 26 channels, the uppercase alphabet is used, then 10 digits, finally '\$' and '%' (i.e. 64 channels).

start start time for data to be included

stop stop time for data to be included

Arguments of start and stop are integers with an optional minus sign. A positive integer is interpreted as an absolute time in seconds past the hour (of the scan start time). When a minus sign precedes the start time it is considered to be a time relative to, and later than, the scheduled scan start. Similarly, a negative stop time precedes the scheduled scan stop time, by the indicated number of seconds.

switched turns on (frequency) switched mode, which discards some AP's and keeps others, depending on a gating waveform

period period in seconds of the gating waveform

gates for each freq. channel, the starting delay and duration, in seconds, of the gating waveform

passband lower and upper bounds (in MHz) of the spectral passband of data to be accepted, specified as RF frequencies. If the lower bound is greater than the upper bound, the range wraps around – allowing a band in the middle to be excluded.

notches a list of non-overlapping [lower,upper] bounds pairs (in MHz) to exclude from the spectral passband (passband may be applied prior to removal of these notches). Note that the amplitude modification calculus isn't sufficiently sophisticated to detect overlaps of passband and notches, so be sure to keep your surgeries disjoint. (A large number of notches is supported and you will get a complaint if you exceed it.) As with passband, the spectral data is rescaled so that the amplitude observable is in some sense preserved.

dc_block if set to true, zero out lowest cross-power spectral channel; useful for suppressing DC bias

– *deprecated commands – for backward mk4 compatibility only:*

<code>max_parity</code>	maximum allowable fraction of bytes in error
<code>x_slip_sync</code>	maximum number of frames w/ re-sync in reference station
<code>y_slip_sync</code>	(same for remote station)
<code>x_crc</code>	either keep or discard AP if reference time has CRC error
<code>y_crc</code>	(same for remote station)

B.4.3 search – *control the fringe-searching process*

sb_win single band delay search window bounds, in us

mb_win multiband delay search window bounds; if the upper multiband delay bound (2nd number) is less than the lower bound (1st number), then *fourfit* performs a "wrap-around" search, in order to handle the case of a delay near to the multiband (semi-) ambiguity.

dr_win delay-rate search window bounds, in us/s

ion_npts number of evaluation points in ionospheric coarse search. Either 0 (the default value) or 1 disables the ionosphere model.

ion_smooth iff true, use alternative ion search strategy in which tec tabular points are smoothed mid-search

ion_win ionospheric coarse search window in TEC units

ra_offset apply right asc.offset (asec) to re-center search windows

dec_offset apply declination offset (asec) to re-center search windows

interpolator selects method of fit interpolation. Classically, an iterative search has been done over sbd, mbd, drate, one dimension at a time, for 3 cycles. The simultaneous mode constructs a 5x5x5 cube of data points and does a 3D quintic interpolation.

B.4.4 corrections – *apply corrections to the data, either before or after fit*

pc_mode specify phase_cal mode: - normal (model linear in time is extracted from the data) - manual (specified totally by the user) - ap_by_ap (phase cal is extracted independently for each AP) DEPRECATED: use normal or manual with pc_period 1 or more - multitone (all tones in band are coherently fit, and phase is extrapolated to the center of the band).

pc_phases phase-cal phases in deg, for each of the listed freq channels; these offset phases are added to the underlying model, as specified by pc_mode, above. If 2 polarizations are present, the same values are applied to both pols.

pc_phases_l specified in same manner as pc_phases, but the tone phases so specified are applied only to the first pol (L, X, or H)

pc_phases_r specified in same manner as pc_phases, but the tone phases so specified are applied only to the second pol (R, Y, or V)

pc_phases_x synonym for pc_phases_l (see)

pc_phases_y synonym for pc_phases_r (see)

pc_freqs phase cal tone frequencies in KHz, for each of the listed freq channels iff not in range -64..64. Inside of this range, the value is interpreted as a tone number, with 1 being the 1st USB tone, 2 being the 2nd USB tone, etc. Negative tone numbers are used for LSB tones.

pc_period in multitone mode (only), the phase can be estimated and applied over each pc_period ap's, thus removing slopes or other drifts in pcal (default is 5)

pc_tonemask - in multitone mode (only): The values for pc_tonemask form a bit-masked map of which tones to *exclude* for this frequency channel. Thus 1 excludes the lowest tone, 2 the next lower tone, 4 the 3rd lowest tone, etc. A value of 5, for example, would exclude the lowest and the 3rd lowest tones (perhaps 10 KHz and 2.01 MHz).

pc_delay_l a time value in ns representing the difference between the

pc_delay_r the travel time from the feed phase center to the pcal injection point, minus the the travel time from the pcal pulse generator to the injection point. It is specified separately for the two polarization senses.

pc_delay_x

pc_delay_y synonyms for pc_delay_l and pc_delay_r (see)

lsb_offset additive phase in degrees, for the LSB relative to the USB; often necessary when correlating VLBA data against Mk 3

ref_freq specifies a frequency in MHz at which the phase delay is determined (default is total LO of first frequency)

adhoc_phase specify mode of ad hoc phase corrections. No corrections are made if this isn't present, or is set to false.

adhoc_period For ad hoc sinewave model; the period in integer seconds.

adhoc_amp " " " " amplitude in degrees of phase.

adhoc_tref For both ad hoc phase models; the reference time in seconds past the most recent hour.

adhoc_poly For the ad hoc phase polynomial model; From 1-6 coefficients describing a power-series model in time. (*deg/secⁿ*)

adhoc_file Name of the file containing phases in the ad hoc file mode.

adhoc_file_chans String of channel labels for phases (columns) in the ad hoc file.

use_samples If true, use the sampler statistics (aka state counts) to normalize the raw correlation sums to the equivalent analog correlation. This is specific to the mk4 hardware correlator, as the state information is not otherwise available.

ionosphere specified per station, in TEC ($10^{16}el/m^2$) units

t_cohere coherence time used in fringe fit (default is infinite). The principal use case for t_cohere is to cope with limited atmospheric coherence at high frequencies. *fourfit* will add together multiple delay-rate channels in an incoherent manner, effectively smoothing the correlation function over delay rate.

delay_offs delay offsets (ns) to be applied to each of the listed freq channels. This correction is made prefit, similar to pcal.

samplers the number of samplers, followed by the freq channel identifiers of channels that share a common sampler are grouped together in strings. In multitone mode only, the averaged tone-derived differential delays are applied to all channels sharing a sampler. In VEX2 the plan is to get this equipment-interconnection information from the VEX file.

optimize_closure modifies fine fringe search so as to minimize the contribution of non-closing delay errors to the closure phase; can result in poorer single-baseline fits

station_delay a priori guess at the delay of the pcal path, from maser to the digitizers (ns). Recommended to use sampler_delay_l/r/x/y instead.

mbd_anchor controls the basis for choosing the mbd ambiguity when forming the total. If 'sbd' then the ambiguity closest to the singleband delay is chosen; if 'model' then the ambiguity closest to the a priori model is chosen.

sampler_delay_l indexed by sampler, it is the center of the window in which the pcal delay ambiguity will be resolved. Just like the old **station_delay**, but now broken down by sampler and polarization, since their cabling and filters will lead to different delays. For LCP, in units of ns.

sampler_delay_r same as the above, but for RCP, instead of LCP.

sampler_delay_x synonym for **sampler_delay_l**.

sampler_delay_y synonym for **sampler_delay_r**.

vbp_correct if true, apply video bandpass correction.

vbp_fit if true, fit for and print video bandpass correction.

vbp_coeffs numerical coefficients for video bandpass correction. The algebraic model used is a cubic polynomial plus a $1/f$ term. If the five coefficients supplied are called a through e , then the model is $\Delta\phi = af^3 + bf^2 + cf + d + e/f$. The coefficients are in degrees of phase.

vbp_file (NYI) if specified, use file-based video bandpass correction from given file (supersedes algebraic model correction).

B.4.5 switches and other miscellaneous functionality

fmatch_bw_pct associate freqs < this percentage of bw together (25% default)

pc_amp_hcode generate an H code iff any pcal amps less than this (0.005 default)

weak_channel the ratio of **single_channel_amp** to **coherent_sum_amp** below which a G code is assigned to the scan (0.5 default)

gen_cf_record if true, saves the full control file in the fringe record

est_pc_manual if nonzero, estimates manual **pc_phase_?** and **delay_offs_?** values. A value >0/<0 estimates ref/rem station values. The magnitude is a mask of bits indicating what and how to make estimates. 0x01 phase, 0x02 median channel sbd, 0x04 average sbd channel, 0x08 use total sbd value, 0x10 use original per-channel sbd delay, 0x20 activates heuristics to discard outliers and 0x40 reports phase measurement as a **pc_phase_offset_?** value. Calculations are made with channels in the -f and -n range (inclusive). Conflicting commanding on delay options does nothing.

chan_ids followed by a string of channel labels and an array of RF frequencies. Allows override of the default single character channel labels. The labels are applied iff the corresponding RF freq matches to 0.01 MHz or better.

B.4.6 other symbols

?	wild card character - for single char within baseline, source or f_group
keep	32767
discard	0
true	1
false	0

B.4.7 Special Formats

UT-epochs: UT-epochs are expressed in the format ddd-hhmmss, where all 10 characters are necessary, including leading 0's if appropriate. This format will match that of a scan directory, if the UT-epoch that is being specified is an actual scan time.

B.4.8 General Guidelines

1. White space is ignored; i.e., multiple spaces and line feeds all collapse to a single space.
2. Multiple commands per line are fine.
3. Comments: anything from an asterisk through the end of the line is ignored.
4. Nested ifs are not allowed (or necessary). Nested parentheses in an if condition are fine (not yet implemented). As of 94.1.16, parentheses are not supported. The logical operators, in decreasing order of precedence are (not, and, or).
5. Wildcard "?" matches any single character for f_group, station, or baseline, any string (of up to 8 characters) for source, and any time-value for scan.
6. Phase cal and delay offsets are treated station by station. If not in a "station context", then values are applied to remote stn only.
7. Only freqs that are chosen for both stations in a baseline are present in the fit.
8. If multiple if-blocks match a particular passes' choice of baseline, f_group, source, and scan criteria, then the later values assigned to each parameter overwrite the earlier ones.

C Multitone phase-cal

C.1 Introduction

In the past, there were three different phase calibration modes that were employed in fourfit, with each having its own strengths and applicable usage cases. A brief summary of their characteristics is given in Table 1. This memo describes a new mode, called multitone mode, which introduces new capabilities, as well as incorporating most of the functionality of the 3 previous modes.

mode description advantages disadvantages normal any single tone extracted by corr. hardware simplicity poor snr due to ignored tones manual each channel's phase is manually specified can be determined by peaking on a strong calibrator and used for a whole experiment; no thermal noise added labor-intensive; requires calibrator scan; doesn't follow scan to scan variation ap_by_ap pcal phase for a single tone is extracted and applied independently for each AP able to track a 'wandering' phase introduces extra noise into fringe phase; poorest snr Table 1 Prior phase cal modes in fourfit

C.2 Multitone Features

- multiple tones within each channel: the coherent sum of all tones within the channel's bandpass is formed, a best-fit delay determined, the phase is evaluated at mid-band, and applied
- masking of tones by channel: in order to avoid contamination of the phase calibration by strong interfering tones, it is possible to select for use any subset of the tones within the band, by way of a excluded tone mask
- flexible averaging over time: since phase corrections can vary over time within a scan, perhaps not even in a linear fashion, the multitone phasecal correction can be segmented over time. The integration period is user-adjustable so that the phase can be tracked with a minimum effect on the noise added into the interferometric fringes by pcal phase noise.

- additive phases by channel: for flexibility it is sometimes useful to apply an extra additive phase, on a channel by channel basis

C.3 Mathematical Underpinnings

Let us define:

- N number of tones within the band
- f_n frequency of the n th phase cal tone within the band (Hz)
- f_c center frequency of the band (Hz)
- ρ_n complex phasor of the n th tone, as measured
- τ_n best-fit delay (s) to tones across the band
- P_n coherent sum at mid-band

To find the best-fit delay we take the complex phasors for all of the included tones within the band, which due to their means of generation by a short pulse - all fall nicely on a grid of equally spaced points in frequency, and find the peak (delay) of their FFT. In order to increase accuracy, the phasors are first 0-padded, which is equivalent to using a sinc interpolator on the delay function. Using the best-fit delay, we then counter-rotate the individual phasors to the center frequency (since we are correcting the mean channel phase) and form their sum:

$$P_n = \sum_{n=0}^{N-1} \rho_n e^{-i2\pi(f_n - f_c)\tau_n} \quad (4)$$

The phase of this complex result, $\arg(P_n)$, is used to correct the fringe phase of the channel.

C.4 Syntax and Usage

A fourfit user invokes and controls multitone mode by adding one or more lines to the fourfit control file. The stations to which a particular specification is applied are determined by the normal fourfit control file context rules. The applicable stations could be established, for example, by a construct of the form if <station> or if <baseline>, or the scope could be global (specified prior to any ifs). The relevant commands, all of which are optional, are

- **pc_mode multitone** command to specify multitone mode
- **pc_period <period>** an integer specifying the averaging period in AP's (default is 5 ap's). pcal ap's are typically 1 sec in duration. The optimal value to use depends on the pcal signal strength and the variation in time of the pcal signal. Smaller/shorter values add more thermal noise in the phase correction but track the phase well, larger/longer values add less Gaussian noise but are susceptible to phase drifts.
- **pc_tonemask <chan_specifier> <mask1> <mask2>** an integer tone exclusion mask is given for each of the specified channels; each bit within an exclusion mask corresponds to a tone within the channel; if set, bit 0 omits the 1st tone from the fit, bit 1 omits the 2nd tone, and so on. This is useful for leaving out RFI-corrupted tones from the fit across the channel.
- **pc_phases <chan_specifier> <ch1_phase> <ch2_phase>** specifies, for each named channel, an additional phase to be added to the values normally extracted by multitone mode.

C.5 Examples

Example 1: comparison of normal and multitone phase cal modes when the tone sequence occurs at different points within different bands. When using normal mode the correction phase depends on the placement of the tone that is being extracted, due to the delay from injection point to where the waveform is sampled.

D File-based phase corrections

For some applications, such as mm-wave VLBI where the atmospheric coherence time scale is short, it is advantageous to be able to make a priori phase corrections in a fringe fit. Typically the phases will be derived through some other avenue, such as other instruments (e.g. water-vapor radiometers) or information from other polarization senses. For full flexibility, one can make corrections for each frequency channel as a function of time within each scan.

The full time-frequency array is specified in a flat 2-dimensional ASCII file, one file per each station. Typically, though not necessarily, a single file would be used for an entire experiment. The file would be given a name that identifies the station whose data it contains.

Each line in the file begins with an epoch, and then has entries for all channels for that epoch, represented by space-delimited phases in degrees. The UT epoch is in decimal days since the beginning of the year. (0300UT on Jan 2 is 1.125). Comments begin with an asterisk, and everything including and beyond an asterisk on each line is stripped prior to reading.

Thus an example of a rather short phase file (having only 4 channels, with data points each 10 sec) might appear as:

```
* Westford phases for chans at 221120.0 221152.0 221184.0 221216.0

223.041667 13.2 -39.3 57.1 177.0
223.041782 9.7 -36.2 77.0 -176.5
223.041898 8.8 -37.7 92.4 -171.9
```

The new fourfit control file commands to support this capability are:

```
adhoc_phase file          * denotes that file mode pcal will be used
adhoc_file <file_name>    * relative or absolute file name containing phases
adhoc_file_chans <channel_list> * where channel_list is ff chan codes
```

An example of a block of a fourfit control file might be:

```
if station E
  adhoc_phase file
  adhoc_file pcal_x3970_wf
  adhoc_file_chans abcd
```

Algorithm

The time points are considered to be either instantaneous values, or the result of a linear average of the phase, symmetric about the specified epoch. The algorithm within fourfit performs a linear interpolation between the specified points, and then finds the average value of the piece-wise linear function so derived over each accumulation period in a fringe fit. If the ap data extends beyond the range of the piecewise linear pcal function, then the function is extrapolated linearly (if possible) to cover the ap.

The phase values are considered to be equivalent to measured pcal phase for a station. Thus the value will be removed from the data phase prior to the fringe fit. In other words, the data phase will be counter-rotated by the given pcal phase. The ad hoc file-based phases are applied in addition to any extracted phase cal (as invoked by `pc_mode` multitone, normal, or manual). In order to zero out the extracted pcal, one can simply use `pc_mode manual` without any phases being specified in a `pc_phases` command, as the default is 0.

Note that fourfit channel codes (abcde) have an implied context of the mk4 fileset it is being applied to. It is possible to recorrelate with a changed vex file, and have the resulting channel codes change. This shouldn't present a problem with a modicum of caution on the part of the fringe fitter.

E Mk4 file system

<mk4 file system document goes here>

F Subversion history

<history of svn mods to go here>

G Adding new parameters

To add a new (simple) parameter, one must modify the source code in a number of routines. Even though there are a number of places to change, the structure of the code ensures that the changes are generally straight-forward. The necessity of changing the routines marked as optional depends on the specifics of the parameter that is being added.

1. **control.h** most likely the parameter value needs to be kept in struct `c_block`
2. **parser.h** most likely you'll need to provide a new token for the parameter
3. **param_struct.h** (optional) possibly add new parameters to this global structure
4. **init_tokens.c** maps your tokens to the control file strings
5. **default_cblock.c** set the default value
6. **nullify_cblock.c** set to recognizable null value
7. **copy_cblock_parts.c** a line to copy it when non-null
8. **parser.c** appropriate fsm changes
9. **precorrect.c** (optional) changes prior to fringe search
10. **create_fsm.c** (optional) if your change changes the fsm
11. **../.. /help/fourfit.doc** document what the new parameter does
12. **../.. /data/ff_testdata/chk..sh** (optional) consider adding a test to make check in order to verify that what you changed does what you want (now and into the future).

H Calling graphs

The overall calling structure of the program can be seen in the following graphs.

