Users Guide for Kvis

Rachel Akeson
Interferometry Science Center

Version 1.11, Nov 29 2004

Contents

1 Introduction

This document describes the visibility data reduction for the Keck Interferometer. Raw interferometer data (called Level 0) is processed into reduced, time-averaged visibility data (called Level 1) by a software package called `Kvis`. This manual details the algorithms used and the inputs and outputs of the program. `Kvis` is run on all KI visibility science data as part of the standard pipeline processing and the reduced visibilities are made available to the users, so the general user will not need to run `Kvis` directly. A visibility data reduction guide for the science user is also available (http://msc.caltech.edu/KISupport/v2/V2reductionGuide.pdf).

`Kvis` is written and maintained by the Michelson Science Center in Pasadena. This manual can be obtained at http://msc.caltech.edu/software/Kvis/Kvis.html in either pdf or html.

It is not the purpose of this document to introduce the reader to the basic concepts of optical interferometry. The reader is referred to section 9 for a list of references. Section 2 will cover just the concepts and data types necessary to understand the visibility processing performed by `Kvis`. Users familiar with these concepts, particularly those who have used the Palomar Testbed Interferometer (PTI), can probably skip to section 3 which describes how to run `Kvis`. This document also contains information on the various `Kvis` modes and some details of the program organization.

This version of the manual applies to `Kvis` version 1.1. The `Kvis` version is included in the Level 1 `.info` file (Sec. 2.1).

1.1 Questions and bug reports

Questions, comments and bug reports on either `Kvis` or this manual should be sent to Rachel Akeson (rla@ipac.caltech.edu).

2 Data flow and processing

The beam combination for each baseline produces two outputs which are measured by a fringe tracker. One output is generally wide-band (WB), either a single pixel or several pixels combined and is used to track the fringe. This output is sometimes referred to as the "white light" channel. The second output is dispersed and directed to an array of pixels, which are referred to as the spectrometer channels. Visibilities are calculated for the wide-band pixel, the summation of the spectrometer channels and for individual spectrometer channels. In the channel numbering used by `Kvis` the wide-band pixel numbering starts at 0 and the spectrometer channels numbering starts at the number of wide-band channels. For more information on the data flow at KI see http://msc.caltech.edu/KISupport/dataOverview.html.

2.1 Data files

The raw data from KI is stored in archive files. The raw data volume is typically 10 Gbytes for night of visilibity observing and these data are stored in a binary format. An interface program, here called the API (application program interface), extracts and sorts the requested data from the archive file and returns it to `Kvis`. The location of the archive file must be specified as described in Sec. 3. The data channels to be returned by the API are contained in the `channels.params` file (Sec. 3.3). The archive file names give the start time for the file with the format yyyydddhhmm.archfile, where ddd is the day of the year and hh and mm are UT time. In addition the file yyyydddhhmm.tindex, which contains reference pointers to times within the archive file, is needed for the API.

The processed visibility data (Level 1 data) is output by `Kvis` and contains time-averaged data (see time terminology below) with the instrument calibrations applied. The Level 1 data files which are produced by `Kvis` are the `.sum, .spec, .anc, .calib,` and .info files which are from the corresponding `Kvis` output lines (SUM, SPEC, ANC, CALIB and INFO/ERROR). The content of these lines is described in Sec. 3.4 and at http://msc.caltech.edu/KISupport/v2/KIV2dataProducts.html.

2.2 Time terminology

The data processing in `Kvis` involves the following basic time units:

Sample This is the time over which one complete measurement of a fringe is made (a complete set of *zabcd* data, see Sec. 2.3, including multiple reads if applicable) and is the level at which fringe data is recorded in the archive. A typical value is 5 msec, corresponding to a fringe tracker rate of 200 Hz. `Kvis` obtains the fringe tracker rate from the data and this rate is reported on the SUM and SPEC output line (Table 5). This rate is automatically extracted from the data.

Frame Configurable length of time over which samples are combined. Spectrometer phases are referenced to the average WB phase over this time. This time is set in `ini.params` and has a default value of 0.1 sec. The visibility (V^2) is computed using incoherent and coherent estimates for each channel. This time scales affects the coherent V^2 and generally should not be longer than 0.1 sec for KI data.

Block Configurable length of time over which frames are combined. These data are recorded in the SUM and SPEC Kvis output (Sec. 3.4) for the wide-band and spectrometer channels respectively (and in the .sum and .spec Level 1 files). The squared visibility (V^2) is calculated coherently and incoherently for the WL channel, a weighted average of the spectrometer channels, and individually for each spectrometer channel. This time is set in `ini.params` and has a default value of 5 sec.

Integration Complete observation of a source, including all the calibrations discussed below, which typically contains 125 seconds of fringe data.

2.3 Visibility estimators

The fringe detector uses a four bin algorithm with path length modulation. See Colavita (1999) for a detailed description of this method. The detector array is read 5 times, before the modulation starts and after each quarter-wave of modulation. These reads are called z, a, b, c and d. Each of these reads consists of a number of sub-reads to reduce the effective read-noise. The basic output of the fringe detector, which is called a sample (Sec. 2.2), is the array counts for each channel and these values are read in by `Kvis`.

For each channel the integrated flux in each bin is calculated as

$$\begin{aligned} A = (a-z)/n_s \quad B = (b-a)/n_s \\ C = (c-b)/n_s \quad D = (d-c)/n_s, \end{aligned} \tag{1}$$

where n_s is the number of non-destructive reads of the detector. The fringe quadratures are then given by

$$\begin{aligned} X &= C - A \\ Y &= D - B \end{aligned} \tag{2}$$

and the total flux is

$$N = A + B + C + D. \tag{3}$$

A measure of the energy is given by

$$NUM = X^2 + Y^2. \tag{4}$$

From these values we calculate the squared visibility and phase

$$V^2 = \frac{\pi^2}{2}\frac{NUM}{N^2} \tag{5}$$

$$\phi = \arctan\frac{Y}{X}. \tag{6}$$

2.4 Calibration

Several internal calibration measurements are made during normal observations. Some are made at the beginning of the observing night or after a change of the fringe tracker parameters (DARK, BRIGHT) and others are taken once for every source (BACKGROUND, FOREGROUND, RATIO and RATIO2). DARK measurements are shuttered internally, BACKGROUND are off-source, BRIGHT are on an internal light source, FOREGROUND are on-source but off-fringe and RATIO/RATIO2 are on-source, off-fringe with either Keck I (RATIO) or Keck II (RATIO2) shuttered internally. RATIO/RATIO2) measurements are made as 5 second sequences are interleaved with 25 secs of fringe tracking data. For data reduction purposes DARK and BACKGROUND are equivalent (in the near-infrared).

The detector bias terms are measured and removed from the data on the sample time scale. The bias terms for X, Y and N (called BX, BY and BN) are calculated from the either the DARK or BACKGROUND data using equations 2–3. The X, Y, N values for each data sample are corrected by subtracting the bias terms and the corrected value of NUM is

$$NUM^* = (X - BX)^2 + (Y - BY)^2. \tag{7}$$

Two additional bias terms come from the photon noise and the read noise. The variance of the read noise term, Brn, is calculated from the average value of NUM^* during the DARK or BACKGROUND observations. The variance of the photon term is kN, where k is the number of detected counts per photoelectron. A high-level calibration observation (BRIGHT) is used to measure k which is given by

$$k = (NUM^* - Brn)/\hat{N}, \tag{8}$$

where the notation $\hat{N}$ is used to denote bias corrected values. The final corrected value for the energy measure can now be calculated as

$$\widehat{NUM} = NUM^* - k\hat{N} - Brn. \tag{9}$$

Control of when to calibrate the data and which data are used to calculate the bias values is described in Sec. 3.1.2. Note that the value of k currently used is set by internal measurements rather than nightly measurements due to the detector characteristics.

The RATIO and RATIO2 measurements are averaged for each integration and the ratio correction calculated as

$$RC = \frac{(1 + R)^2}{4R}, \tag{10}$$

where R is the ratio of photon counts for each arm. The ratio correction is calculated for each spectral channel and is included on the SUM line (Table 5) for wide-band and summed spectrometer outputs and on the SPEC line for the individual spectrometer channels. Note that data taken before September 2002 only contain RATIO data and not RATIO2. In this case the FOREGROUND and RATIO data are used to calculate the ratio correction.

2.5 Dewarping

If the path-length modulation does not match the wavelength of the channel, a correction must be applied to the fringe quadratures. For a stroke length in radians of s, where $s = 2\pi$ is a matched stroke, the dewarped values are given by

$$\begin{aligned}
X_c &= \frac{s}{2\pi}\left(\frac{X+Y}{\beta} + \frac{X-Y}{\alpha}\right) \\
Y_c &= \frac{s}{2\pi}\left(\frac{X+Y}{\beta} - \frac{X-Y}{\alpha}\right) \\
N_c &= N - \frac{\gamma}{2\alpha}(Y-X) \\
NUM_c &= NUM - 2GkN_c - \frac{\alpha\pi}{s}k(Y_c - X_c), \qquad (11)
\end{aligned}$$

where X, Y, N are the measured quantities and the constants are given by

$$\begin{aligned}
\alpha &= 2\sin(s/4) - \sin(s/2) \\
\beta &= 1 - \cos(s/2) \\
\gamma &= 2\sin(s/2) \\
G &= \left(\frac{s}{2\pi}\right)^2 \left(\frac{1}{\alpha^2} + \frac{1}{\beta^2}\right). \qquad (12)
\end{aligned}$$

The second term in equation 11 for NUM takes the place of the kN term in equation 9. This dewarping correction is applied to all data (fringe tracking and calibration) before the bias corrections discussed in Sec. 2.4 are applied. This results in a unbiased estimate of the visibility. Whether or not the dewarping correction is applied is controlled by the `doDewarp` parameter (Sec. 3.1).

2.6 Time and spectral averaging

`Kvis` combines groups of samples into frames (as configured, Sec. 3.1). The samples are filtered through a buffer to eliminate the data immediately before a transition from lock to semi-lock in the fringe tracker. The size of this buffer is set by `lockBufferSize` in `config.params` (Sec. 3.2) and corresponds to the averaging used by the fringe tracker when determining the lock state. Each good sample is dewarped and the bias corrections are applied. For the wide-band channel, an average phase is calculated for the frame. Coherent estimators for each frame are produced for each channel by derotating the fringe quadratures with respect to the WB average phase (ϕ_{WB}). For the spectrometer channels, ϕ_{WB} is scaled by the ratio of the WB and channel wavelengths.

$$\begin{aligned}
\hat{X}_{i\,coh} &= \langle \hat{X}_i \cos(\phi_{WB}\lambda_{WB}/\lambda_i) - \hat{Y}_i \sin(\phi_{WB}\lambda_{WB}/\lambda_i)\rangle \\
\hat{Y}_{i\,coh} &= \langle \hat{X}_i \sin(\phi_{WB}\lambda_{WB}/\lambda_i) + \hat{Y}_i \cos(\phi_{WB}\lambda_{WB}/\lambda_i)\rangle. \qquad (13)
\end{aligned}$$

(14)

The coherent value for NUM is given by

$$NUMC = (\hat{X}_{i\ coh})^2 + (\hat{X}_{i\ coh})^2 - (k\hat{N} + Brn)/N_{samps} \tag{15}$$

where N_{samps} is the number of samples in the frame and the incoherent value by

$$NUMI = \langle \widehat{NUM} \rangle, \tag{16}$$

where the average is calculated from the sample values.

In combining frames to form a block the values of $NUMI, NUMC$ and N are simply averaged. The coherent and incoherent values of V^2 for each channel are given by equation 5. The spectral line channels are combined to form a composite spectrometer visibility

$$V_{spec}^2 = \frac{\sum_i V_i^2 W_i}{\sum_i W_i} \tag{17}$$

where the weights W_i are set by the user (Sec. 3.1).

3 Running Kvis

`Kvis` is run from the command line with input arguments from both the command line and parameter files. The command line syntax is
> `Kvis startTime stopTime`

- startTime and stopTime can be in either decimal hours (UT) or hh:mm:ss format. If no times are given, the entire file will be processed. If only 1 time is given, it is used as the startTime.

The location of the input data is specified by setting the environment variable `DATA_DIR`. For example, if the data exist in the directory `/home/keck/dataArchive` then before running `Kvis`,
> `setenv DATA_DIR /home/keck/dataArchive`
for t-shell, or
> `DATA_DIR = ''/home/keck/dataArchive''`
for Bourne shells. If DATA_DIR is not specified, a default path (which is defined in `vis_main.h`) is used. If more than one archive file is present, the first file is used. A specific data file can be specified by setting DATA_DIR to the complete file name, using the `.archfile` extension.

The output of `Kvis` is to stdout and stderr. The level of output is controlled by parameters in the `ini.params` file (Sec. 3.1).

3.1 Parameters set by user

The parameters set by the user to control the data processing and output of `Kvis` are contained in the file `ini.params`. The file must be in the working directory, otherwise a default location will be searched. The location of the default file is set in `vis_main.cc`. If the file exists, but a given value is not present or an error is encountered during the read, a default value is used. The default values are chosen to be appropriate for the standard data reduction. A description of all the variables in `ini.params` in given in Table 1 and further discussion of the more detailed parameters is in the following sections.

The file format is taken from the PTI `vis.params` file and input routines, which were written by Brad Hines. The variables in the file can appear in any order within the header categories of grouping, processing and flags. The variable names are case insensitive and must be separated from the value by an =. Comments are designated with a ; at the start. The format of the file, along with the default values, is:

```
[grouping]
frameTime = 0.1
minFrameFill = 0.5
blockTime = 5.0
minBlockFill = 0.5

[processing]
specAvgType = 0
doCal = 1
calType = 3
saveCalData = 0
doDewarp = 1

[flags]
doPrintRaw = 0
doPrintSample = 0
doPrintFrame = 0
doPrintBlock = 1
doPrintSpec = 1
doPrintAnc = 1
```

3.1.1 Grouping parameters

The `frameTime` and `blockTime` parameters can be set to any value. However, if the requested time would cause the number of samples per frame or frames per block to exceed the allocated array sizes in the code, a message to this effect will be printed and the program will end. The array sizes can be increased by changing the values defined at the top of `vis_main.h` and recompiling.

The minimum filling fractions (`minFrameFill, minBlockFill`) specify the minimum fraction of good data within the specified frame or block time. This calculation is done in terms of the number of good items in the next lower processing level. For example, `minFrameFill = 0.5` means that one half of the samples within the frame must be good. At the block level, all frames are counted equally towards the `minBlockFill`.

3.1.2 Calibration parameters

Which data are used to calculate the bias values is controlled by the `calType` parameter.

calType = 0 The bias values are read from a file `calData.input` which must be in the working directory. The file format is 1 row per channel, with the WB values first. The bias values are given in the order BX, BY, BN, Brn, k. A file in the correct format called `calData.input` can be produced by using the `saveCalData` parameter. Using this parameter will destroy any previous versions of the file.

calType = 1 The bias values are calculated from averages of all the DARK and BRIGHT data present in the archive data file. The entire file will be searched, not just the time range specified on the command line.

calType = 2 Bias values for BX, BY, BN, Brn are calculated from the average of each BACKGROUND instance and for k using an average of all the BRIGHT data. For each block, the set of bias values is chosen from the BACKGROUND data closest in time.

		grouping	
frameTime	Length of frame in seconds		
	Allowed values:	real	
minFrameFill	Minimum fraction of frame with good data		
	Allowed values:	0 to 1	
blockTime	Length of block in seconds		
	Allowed values:	real	
minBlockFill	Minimum fraction of block with good data		
	Allowed values:	0 to 1	
		processing	
specAvgType	Sets weights used in spectrometer channel averaging		
	Allowed values:	0	weights = 1
		1	weights $\propto 1/\sigma^2_{V_i^2}$
doCal	Allowed values:	0	no calibration applied
		1	calibration applied
calType	Controls which calibration data is used		
	Allowed values:	0	data read from file `calData.input`
		1	use DARK,BRIGHT data
		2	use BRIGHT and closest BACKGROUND data
		3	use BRIGHT and median BACKGROUND data
		4	use BRIGHT, BACKGROUND and FOREGROUND data
		5	use BRIGHT, BACKGROUND and FOREGROUND data
saveCalData	Controls writing of calibration data to file		
	Allowed values:	0	no file written
		1	write data to file `calData.input`
doDewarp	Allowed values:	0	no dewarping calibration applied
		1	dewarping calibration applied
		flags	
doPrintRaw	prints all samples (WB only) regardless of data type		
doPrintSample	prints calibrated samples, one line per channel, grouped by frame		
doPrintFrame	prints calibrated frames, one line per channel		
doPrintBlock	prints WB and averaged spectrometer channels		
doPrintSpec	prints individual spec channel blocks		
doPrintAnc	prints ancillary telemetry		
	Allowed values:	0	do not print
		1	print

Table 1: Brief descriptions of all parameters in the `ini.params` file.

calType = 3 Bias values for BX, BY, BN, Brn are calculated from the average of each BACKGROUND instance and for k using an average of all the BRIGHT data. For each block, a set of 3 bias value sets is chosen based on closest time and each bias value is determined separately as the median value from those 3 sets.

calType = 4 As for calType = 3, except BX and BY are calculated from the FOREGROUND observations.

calType = 5 As for calType = 3, except BX, BY, Brn are calculated from the FOREGROUND observations. This method results in the removal of an estimated scintillation term from the data.

By default, for `calType` = 1 to 5 the entire input data file is searched for calibration data, regardless of the start and stop times specified. This loop through the data can take many minutes for data files lasting for several hours. If `setCalTime` is set to 1 (in `config.params`), the calibration processing will only use the data in the time range specified on the command line.

If no BRIGHT data are present, the values for k are set to the default value, which currently (May, 2003) is 0.26. This value can be changed using the DNPERELECTRON variable in `vis_main.cc`.

A memo describing how the default values of these parameters were determined using KI observations is available at http://msc.caltech.edu/KISupport/dataMemos/index.html.

3.2 Configuration parameters

Some of the parameters describing the configuration of the fringe tracker are read automatically by `Kvis`. Others processing parameters are contained in the file `config.params` which has the same format as `ini.params`. A brief description of these parameters is given below. In order for the data to be processed correctly, the user must ensure that `config.params` contains the values reflecting the state of the fringe tracker when the data was taken. `config.params` files are archived with the data files.

numSpecChannels	Number of spectrometer channels.
WBWave	Wide-band wavelength in microns.
strokeWave	Length of delay line stroke in microns.
channelWave	List of spectrometer channel wavelengths in microns. Format is comma separated values enclosed in curly brackets.
WBNumReads	The number of sub-reads for the wide-band channel.
specNumReads	The number of sub-reads for the spectrometer channels.
lockBufferSize	The number of samples used to determine the transition to and from lock and semi-lock within the fringe tracker software.
FTSMode	Toggles FTS mode (Sec. 7.1).
telemMode	zabcd telemtry type: 0 = images (before June 2001), 1 = sequence of doubles (June 2001 and later)
seqRead	Debugging mode used for archive files with bad index files.
setCalTime	If 1, search for calibration data only within times set on command line.
skipZCheck	Should be 0 for all files after Jan 2002.
LDLposition	The position of the long delay line. Only necessary if the LDL telemetry is not present (Oct 2003 and earlier).
maxLaserError	The maximum delay line position error for accepted data.

rtKvisMode Used only for Kvis run at the observatory (7.2).

3.3 Channel parameters

The `channel.params` file tells the API which channels to retrieve from the archive file. It is required for all Kvis versions 1.0 and later and is not used in all earlier versions. The file consists of a series of telemetry channels and parameters for each channel. Example lines are given below. The columns are:

Type name	
Full channel name	Must match name in archive file
Required	1 = required, 0 = optional
Default value	if optional (character string)
Trigger channel	1 = yes, 0 = no
Raw trigger channel	1 = yes, 0 = no
Bin size	For Rawdata channels only
Latency	in seconds, mainly for enum channels

All required channels must be present within the latency time given for a data sample to be returned. Optional channels will return the default value if data is not present in the archive file at the given time. The trigger channel is the channel used by the API to set the data sample times. The Rawdata type is a specific format for returning fringe tracker data that must be specially treated by the API. Based on the fringe tracker telemetry channels given in the file, `Kvis` will make the appropriate function calls for the real time system version.

Example lines:

```
FT_ServoMode FringeTracker[0].ServoMode 1 0 0 0 0 5
FT_TrackMode FringeTracker[0].CompleteControllerSet.TrackModeUpdate.TrackMode 1 0 0 0 0 5
FT_OperatingMode FringeTracker[0].OperatingMode 1 0 0 0 0 5
FT_CalibrationTag FringeTracker[0].CalibrationTag 1 0 0 0 0 5
Seq_StarId Seq[0].StarId 1 none 0 0 0 -1
Raw00 FringeTracker[0].CompleteControllerSet.ControllerData.RawData[0][0] 1 0 0 0 5 -1
Raw10 FringeTracker[0].CompleteControllerSet.ControllerData.RawData[1][0] 1 0 0 0 5 -1
Raw11 FringeTracker[0].CompleteControllerSet.ControllerData.RawData[1][1] 1 0 0 0 5 -1
Raw12 FringeTracker[0].CompleteControllerSet.ControllerData.RawData[1][2] 1 0 0 0 5 -1
Raw13 FringeTracker[0].CompleteControllerSet.ControllerData.RawData[1][3] 1 0 1 1 5 -1
FT_OPD0 FringeTracker[0].CompleteControllerSet.ControllerData.OPD[0] 1 0 0 0 0 -1
FT_OPD1 FringeTracker[0].CompleteControllerSet.ControllerData.OPD[1] 1 0 0 0 0 -1
DL0_metrology DelayLine[0].CompleteControllerSet.ControllerData.LaserPositionOPD 1 0 0 0 0 -1
DL0_ServoMode DelayLine[0].ServoMode 0 SM_TRACK 0 0 0 5
DL1_metrology DelayLine[1].CompleteControllerSet.ControllerData.LaserPositionOPD 1 0 0 0 0 -1
DL1_lasererror DelayLine[1].CompleteControllerSet.ControllerData.LaserError 0 0 0 0 0 -1
```

The full channel names for archive file telemetry can be found by running the `inventory` program. `channel.params` files are archived with the data files.

3.4 Output description

All output lines begin with a keyword identifying the contents of the line as informational (INFO), warnings (ERROR) or one of the following data output types: RAW, SAMPLE, FRAME, SUM, SPEC, ANC, CALIB. The SUM, SPEC, ANC, CALIB and INFO and ERROR lines are produced as part of the pipeline processing and are contained in the correspondingly named Level 1 files (except for ERROR which is in the .info file) The ERROR lines are sent to stderr rather than stdout. Which data output lines are written is controlled by the settings in `ini.params` (Sec. 3.1). The INFO and ERROR lines are always written. The RAW format is used primarily for engineering purposes and outputs the WB channel for all samples, regardless of the fringe tracker lock state. Due to the order in which the data are processed during calibration, the SAMPLE output has all samples in a frame for channel 0, followed by all samples for channel 1, etc., until all channels have been printed. Then the next set of samples is printed.

An easy way to create processed data files for plotting or other tasks is to `grep` the output of `Kvis` using the desired keyword. A header line is produced for each of the data output types. The headers begin with a # so that data output can be directed to plotting programs such as `gnuplot` without removing this line. Examples of each data output type are given below and the variables in each column are described in more detail in the tables in Appendix A.

The calibration flag in the sum and spec files has the following possible values:

0 ok
1 time difference between fringe data and background > CalTime
2 time difference between foreground and ratio > CalTime
only applies when using ratio1 and foreground, not ratio1 and ratio2
3 1 and 2 true
4 no ratio data within CalTime
5 1 and 4 true

where CalTime is currently set to 10 minutes.

4 Examples

4.1 Create a sum file

Many of the processing programs which follow `Kvis` (ex. `bFit, wbCalib`) use a so-called sum file of processed block data as their input. Here is one way to produce that file for an entire night of data.

Make sure `doPrintBlock` is set to 1 in `ini.params`.

```
> setenv DATA_DIR /proj/keckdata/sdata900/2001311
> Kvis | grep SUM > outputData.sum
```

4.2 Sample phase as a function of time (or delay)

A good method for making quick plots is to use `gnuplot`. This example use the WB channel (channel = 0), to use other channels, change the 0 in the $5 == 0 to the desired channel number. Note that the sample output can produce very large files, so it is probably best to limit the time range, using the command line arguments to `Kvis`.

Make sure `doPrintSample` is set to 1 in `ini.params`.

```
> setenv DATA_DIR /home/keck/dataArchive
> Kvis 02:00:00 02:30:00 | awk '$1 == ``SAMPLE'' && $5 == 0 {print $0}' > data.sample
```

```
> gnuplot
(for phase vs. time)
  gnuplot> plot 'data.sample' u 3:17
(for phase vs. delay)
  gnuplot> plot 'data.sample' u 5:17
```

4.3 Additional plotting

A set of plotting routines (called `sumplots`) utilizing the gnuplot plotting program was written for PTI by Andy Boden. These have been adapted for `Kvis` and are called `KvisPlot`. There are two modes for `KvisPlot`: 1) creating a series of standardized plots of the sum file and 2) creating one plot with user-specified axes from a file containing SAMPLE, FRAME or SUM lines. The input file can only contain one of the allowed types. The default behavior, invoked if `KvisPlot` is called with only a file name as an argument, is to produce the standard sum plots. The standard sum file plot are: V_i^2(WB), V_i^2(WB), N(WB), V_i^2(Spec), V_i^2(Spec), N(Spec), and jitter and other standard plots as a function of time in UT hours (see Table 5 for a descriptions). More information can be found at http://msc.caltech.edu/software/Kvis/KvisPlot.help and a description of the plots is at http://msc.caltech.edu/KISupport/v2/Level1Plots.html.

Standard sum file plots

```
>KvisPlot -all file.sum
(produces plots to screen)
>KvisPlot -all -hc file.sum
(produces a postscript file called file.sum.ps)
>KvisPlot -spec file.spec
(produces standard plots for spectrometer channels)
>KvisPlot -anc file.anc file.sum
(produces ancillary data plots as well as standard plots)
```

Single plot

```
>KvisPlot -one any.file
```

The user will then be prompted with a list of allowed variables to enter as the x and y-axis.

5 Known problems and limitations

- Times within `Kvis` are treated as UT hours with no date information. This means that multi-day data files (which do not normally occur) and files which contain the 24 UT to 0 UT transition are not handled properly (note: this boundary is during the day in Hawaii and so will not occur in standard night observing).
- All possible bad values of parameters in `ini.params` and `config.params` are not checked for.
- The numbers of spectrometer channels and wavelengths must be constant throughout the archive files (or the specified start and stop times if given). The fringe tracker rate may change.

6 Troubleshooting

Problem: No data lines are output when `Kvis` is run.

1. Check that at least one of the doPrint variables in the `ini.params` file is set to 1.

2. If you have specified times on the command line, check that data exists at these times. One way to do this is to look at the inventory file for the archive file you are using. This file will have a .chansum extension and the same numerical prefix as the archive file. If the inventory file does not exist, it can be created by running `inventory` after setting the ARCHIVEPATH environment variable to the desired archive file.

3. Check that the fringe tracker was locked during the specified time. It is possible that data are present but the fringe tracker was not locked. To see ALL fringe tracker data, set doPrintRaw in `ini.params` to 1. Beware that the output volume from this mode is substantial.

Problem: Error messages regarding missing telemetry.

Check the specification in `channels.params`. You may need to change the missing telemetry channels to optional. Note: These failed returns may come in the middle of the processing after the entire archive file has been seached for calibration data and the data processing pass through the archvie is begun.

7 Special Modes

7.1 FTS Mode

In FTS mode, the interferometer is set up to measure fringes on an internal light source. From a data processing point of view, the only difference is the conversion between laser card counts to delay in physical units. Setting `FTSMode` to 1 in `config.params` will change this conversion appropriately and will produce output lines with FTS headers. These lines contain the delay and the values of N for each channel, starting with the wide-band channel.

7.2 rtKvis mode

`Kvis` has a special mode for near real-time visibility estimates while observing and should ONLY be used at the Keck Observatory. This mode is specified by setting `rtKvisMode = 1` in `config.params`. The rtKvis script is described at http://msc.caltech.edu/software/Kvis/rtKvis.help. In rtKvis mode, `Kvis` processes the calibration data as it is found, rather than in a separate pass through the data. The averaging times for the calibration data are the same as for fringe data and no averaging is done across calibrations (for example, all RATIO integrations are processed separately, rather than averaged). In addition, when in this mode, `Kvis` checks the current time and the processed time and sleeps to avoid running past the end of the archive file. A stop time is set within the program so that the processing finishes at 18 UT (8 am Hawaii Standard Time).

8 Programming details

This section includes some details on how the `Kvis` program functions, but is not meant to be a complete description. See Appendix B for a diagram of the program subroutines.

8.1 Adding new channels

To add a new telemetry channel of a known telemetry type to the API, there are 4 steps.

1. Add the channels to `channels.params`. The format of this file is specified in Sec. 3.3.
2. Add the channel to the archive pointer list in vis_main.cc. The format is `result = archivePtr.KAI_tie("FT_OPD0", &FT_OPD0);` where the first argument is the first column of the channels.params file and the second in the name given the vis_main.h.
3. Add the channel to vis_main.h. Using the same names from vis_main.cc and the appropriate data type, add the channel to vis_main.h. Example: `DoubleType FT_OPD0;`
4. Transfer the values from the API variables to local variables using the getValue method. Example: `WLSample->fastDelay = DL1_metrology.getValue();`

9 References

- General optical and infrared interferometry
  - Shao, M. and Colavita, M.M. 1992, *ARAA*, 30, 457.
  - Selected papers on Long Baseline Stellar Interferometry, ed. P. Lawson, 1997, SPIE Milestone series, volume 139.
  - Principles of Long Baseline Stellar Interferometry, ed. P. Lawson, Course notes from the 1999 Michelson summer school (available at http://sim.jpl.nasa.gov/library/coursenotes.html).
- Visibility data reduction
  - Colavita, M.M. 1999, *PASP*, 111, 111.
  - V^2 reduction guide, http://msc.caltech.edu/KISupport/v2/V2reductionsGuide.pdf
- Technical desciption of the Keck Interferomter
  - Colavita, M.M. and Wizinowich, P., 2003, SPIE vol. 4838, p. 79

A Output file descriptions

The following tables contain the descriptions of the column format for the output formats. See http://msc.caltech.edu/KISupport/v2/KIV2dataProducts.html for descriptions of the INFO and CALIB output formats.

Raw samples

Column	Variable	Description
1	RAW	File indentifier
2	time	UT time (decimal hours)
3	delay	delay line position (meters)
4	z	Raw values as read from IR array not yet divided by number of sub-reads
5	a	
6	b	
7	c	
8	d	
9	sflag	servo mode value
10	tflag	track mode value
11	cflag	calibration mode value

Table 2: Description of columns in the RAW output.

Samples

Column	Variable	Description
1	SAMPLE	File indentifier
2	date	month/day/year/hour:min:sec
3	time	UT time (decimal hours)
4	source	source name
5	delay	delay line position (meters)
6	channel	WL = 0, spectrometer starts at 1
7	z	Raw values as read from CCD not yet divided by number of sub-reads
8	a	
9	b	
10	c	
11	d	
12	X	fringe quadrature (A – C)
13	Y	fringe quadrature (B – D)
14	N	total flux (A + B + C + D)
15	NUM	energy ($X^2 + Y^2$)
16	V^2	squared visibility
17	phase	unwrapped fringe phase (radians)
18	opd	optical path difference as measured by fringe tracker (μm)

Table 3: Description of columns in the SAMPLE output. Internal calibrations and dewarping applied to X,Y,N and NUM.

Frames

Column	Variable	Description
1	FRAME	File indentifier
2	date	month/day/year/hour:min:sec
3	time	UT time (decimal hours)
4	source	source name
5	delay	delay line position (meters)
6	channel	WL= 0, spectrometer starts at 1
7	# samp	Number of samples averaged in this frame
8	X	fringe quadrature (A – C)
9	Y	fringe quadrature (B – D)
10	N	total flux (A + B + C + D)
11	NUMI	incoherent energy ($X^2 + Y^2$)
12	NUMC	coherent energy ($X^2 + Y^2$)

Table 4: Description of columns in the FRAME output.

Blocks

Column	Variable	Description
1	SUM	File indentifier
2	date	month/day/year/hour:min:sec
3	time	UT time (decimal hours)
4	source	source name
5	FDL	fast delay line position (meters)
6	LDL	long delay line position (meters)
7	N(WB)	wide-band photons (in DN)
8	V_i^2(WB)	incoherent wide-band squared visibility
9	V_c^2(WB)	coherent wide-band squared visibility
10	λ(WB)	wide-band wavelength (microns)
11	N(S)	average spectral channel photons (in DN)
12	V_i^2(S)	incoherent V^2 from weighted average of spec channels
13	V_c^2(S)	coherent V^2 from weighted average of spec channels
14	λ(S)	weighted average of spec channel wavelengths (microns)
15	jitter	rms of difference between sample phases (radians)
16	# frames	number of frames in this block
17	# locks	number of lock breaks in this block
18	fracTime	fractional time locked on fringe
19	FT rate	fringe tracker sample rate (Hz)
20	RC(WB)	wide-band ratio correction
21	RC(S)	weighted average spec ratio correction
22	flag	calibration flag
23	baseline	baseline designation

Table 5: Description of columns in the SUM output.

Spectral channel blocks

Column	Variable	Description
1	SPEC	File indentifier
2	date	month/day/year/hour:min:sec
3	time	UT time (decimal hours)
4	source	source name
5	FDL	fast delay line position (meters)
6	LDL	long delay line position (meters)
7	jitter	rms phase jitter (radians)
8	FT rate	fringe tracker sample rate (Hz)
9	flag	calibration flag
10	baseline	baseline designation
11	nch	number of spectral channels
12	λ	wavelength for channel 1 (μm)
13	N	number of photons (in DN) for channel 1
14	$V_i^2(1)$	incoherent V^2 for channel 1
15	$V_c^2(1)$	coherent V^2 for channel 1
16	RC(1)	ratio correction for channel 1
$\vdots$	$\vdots$	$\vdots$
nch*5 + 7	λ	wavelength for channel nch (μm)
nch*5 + 8	N	number of photons (in DN) for channel nch
nch*5 + 9	V_i^2	incoherent V^2 for channel nch
nch*5 + 10	V_c^2	coherent V^2 for channel nch
nch*5 + 11	RC	ratio correction for channel nch

Table 6: Description of columns in the SPEC output.

Ancillary

Column	Variable	Description
1	ANC	File indentifier
2	date	month/day/year/hour:min:sec
3	time	UT time (decimal hours)
4	source	source name
5	V_i^2(WB)	incoherent wide-band squared visibility
6	jitter	rms of difference between sample phases (radians)
7	N(WB)	wide-band photons (in DN)
8	N(WB) rms	wide-band rms/mean
9	WB ratio	Wide-band ratio (Keck 1/Keck 2)
10	Spec ratio	Summed spectrometer ratio (Keck 1/Keck 2)
11	ST(K1)	Keck 1 Star tracker mean intensity (DN)
12	ST(K2)	Keck 2 Star tracker mean intensity (DN)
13	ST rate	Star tracker rate (Hz)
14	Az(K1)	Keck 1 azimuth (radians)
15	Elev(K1)	Keck 1 elevation (radians)
16	Az err(K1)	Keck 1 azimuth error (radians)
17	El err(K1)	Keck 1 elevation error (radians)
18	Az(K2)	Keck 2 azimuth (radians)
19	Elev(K2)	Keck 2 elevation (radians)
20	Az err(K2)	Keck 2 azimuth error (radians)
21	El err(K2)	Keck 2 elevation error (radians)
22	AO cnts(K1)	Keck 1 Mean wave-frint sensor counts
23	AO rate (K1)	Keck 1 AO correction loop rate (Hz)
24	AO DT gain(K1)	Keck 1 AO tip-tile loop gain
25	SO DM gain(K1)	Keck 1 AO deformable mirror loop gain
26	AO cnts(K2)	Keck 2 Mean wave-frint sensor counts
27	AO rate (K2)	Keck 2 AO correction loop rate (Hz)
28	AO DT gain(K2)	Keck 2 AO tip-tile loop gain
29	SO DM gain(K2)	Keck 2 AO deformable mirror loop gain

Table 7: Description of columns in the ANC output.

B Program layout and data flow

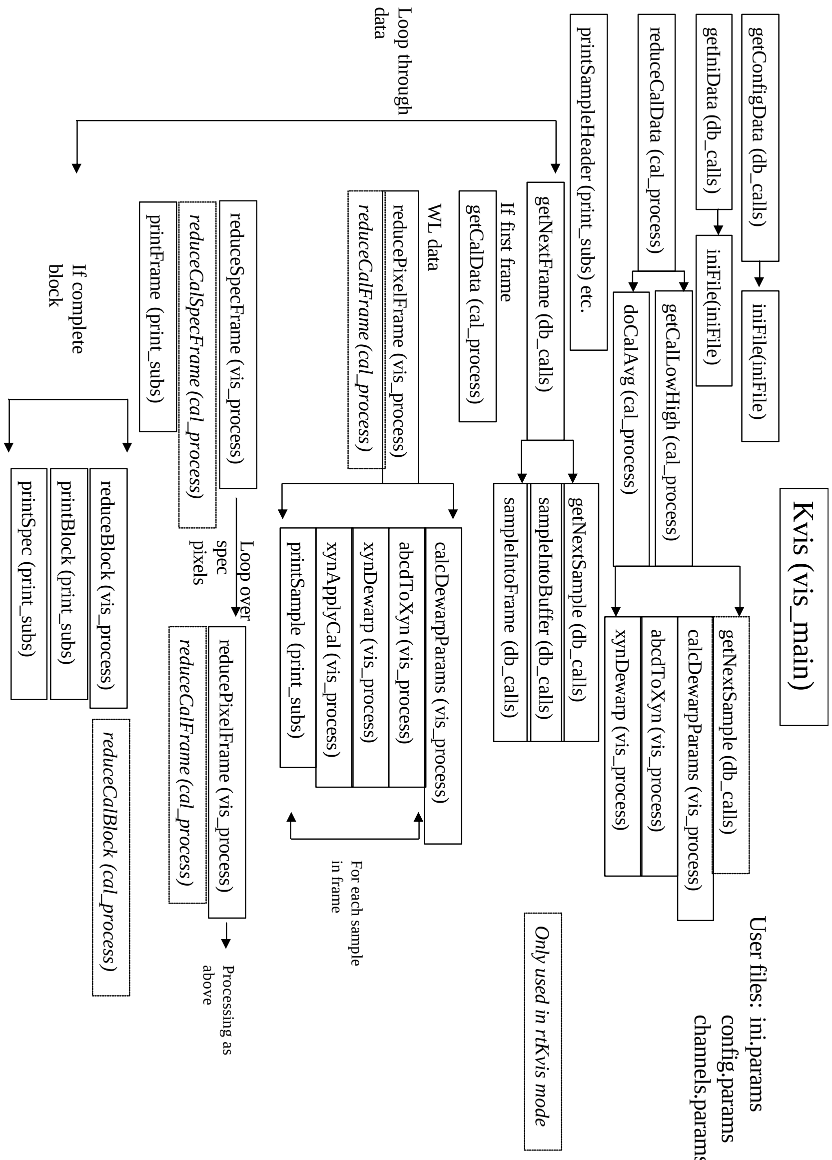

Figure 1: Flow chart of Kvis subroutines