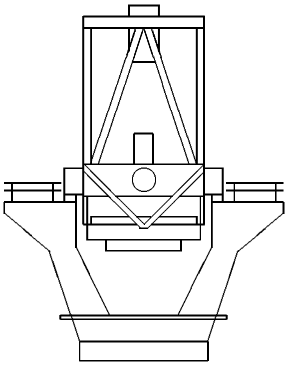
Isaac Newton Group of Telescopes

ALPHA-PLOT-2

PLOT
USERS' MANUAL



Version 2.0

January 12, 2000

Robert Laing, Robin Clark

Contents

List of Figures

PLOT Users' Manual **PLOT**

1 Introduction

This note describes **PLOT**, a stand-alone program for Alphastations which can be used to display and analyse log files produced by the Telescope Control System. It reads the data-files produced by all of the the TCS logging options and produces graphical output on any device or format supported by native PGPLOT, most usually X-windows and postscript.

Commands to be typed are given throughout in teletype font (e.g. `PLOT`).

2 Basic Concepts

2.1 Outline

The main steps in making a simple plot are:

1. Open a log file and review the header (`DATA`).
2. Read in the data values for the horizontal and vertical axes. Either individual quantities (`READ`) or their means (`MEAN`) or differences (`SUBTRACT`) can be read in.
3. Select a graphics device (`DEVICE`).
4. Set the plot limits (`BOX`, `LIMITS` or `SQUARE`).
5. Make the plot (`PLOT`).
6. Exit from the program (`EXIT`).

2.2 Buffers

PLOT works by reading up to three quantities from a log file into a buffer called DATA. The contents of this buffer can then be displayed or modelled, the results and residuals of modelling being stored in MODEL and RESIDUALS, respectively. The buffers are 3 $\times$ N arrays, where N (currently 100000) is the maximum number of data points. Each buffer is divided into 3 N-dimensional vectors, referred to as axes:

Axis 1 is the horizontal plot axis.

Axis 2 is the vertical plot axis.

Axis 3 is used for the independent variable in two-dimensional fits (see `FIT2D`) and is not plotted directly.

2.3 Log files and headers

The TCS command `LOG` can be used to produce data files of diagnostic information. These all have the same generic format: a header, followed by an array of data records. Details will vary according to the specific subset of data being logged and are telescope-specific (see Sections A and B). The header contains the information needed by **PLOT** to decode the data records, scale into appropriate units and display with sensible axis labels. The header contains the following information:

1. The name of the telescope.
2. The (UT) date.
3. The type of log file (encoder, autoguider or TV).
4. The number of data fields.

5. A list of the data fields. Each data field must have a variable name. This is a keyword of up to 8 characters which is used to identify the variable in **PLOT** commands and has an associated label of up to 20 characters which appears in plot labels. There is also an optional mechanism keyword of up to 8 characters, associated with a group of variables. This is used to specify arithmetic operations on groups of similar variables (e.g. averaging or subtraction of multiple encoder on the same drive axis).

6. The number of data records in the file.

The header of the current data file can be displayed using the `HEADER` command (this happens automatically when the file is first opened). The mechanism and variable names are listed, along with the plot labels. An example of the first few lines of a header display is:

```
INT                 19971017
Encoder log file
Number of data fields = 16

Key 1     Key 2     Label

          UTC       Time
HA        FERR1     HA Ferranti 1
HA        FERR2     HA Ferranti 2
```

This is an encoder log file for the INT obtained on 17th October 1997 with 16 data fields, of which the first 3 are shown. Key 1 is the mechanism and key 2 is the variable. The first field is universal time, UTC, for which there is no mechanism. The second and third fields are hour-angle encoders. They both have the mechanism name HA, and are identified individually by their variable names FERR1 and FERR2.

2.4 Units

Each quantity in a log file has an associated unit, given by a code in the header. This is used for plot labels, fitting calculations and command parameters. The units currently supported are shown in Table 1. When a quantity is read into the data buffer using `READ` or `MEAN`, then it will be displayed in the units associated with its header code. The same is true if the `SUBTRACT` command is used, unless the input quantities are in degrees or hours, in which case their difference will be displayed in arcsec. This is done because a common application for the program is to look at the differences between two independent encoders on the same axis, plotted against the reading from one of them. The most useful units in this case are degrees (or hours) for the reading and arcsec for the difference.

3 Starting PLOT

The following assumes that **PLOT** is to be run either from the TCS login menu or from an account on one of the Alphastations LPAS1–LPAS4 on La Palma or PICO and CORVO in Cambridge.

3.1 Running PLOT from a private account

The logical name of the directory containing the executable image and command files needed to run **PLOT** is PLOT_DIR. This should be set up as a system logical name. Check this by typing:
`DIR PLOT_DIR`
The files PLOT.EXE and PLOT_INIT.COM should be present. If they are not, you will need to find out the location of PLOT_DIR and use the DCL `DEFINE` command to set it. To set up the symbols needed by **PLOT**, type:

Table 1: Units supported by **PLOT**

Unit	Variables
Hours	Hour angle
Degrees	Azimuth, elevation, declination, gear phase
mm	Focus position
° C	Temperature
Microns	Displacement transducer readings
Days	Time
Degrees/s	Drive demands
Arcsec	Autoguider and TV data, drive position errors

`@PLOT_DIR:PLOT_INIT`
and, if you plan to make X windows plots:
`XDISPLAY`
To start the program, type:
`PLOT`
This should generate the prompt:
`PLOT>`

3.2 Running PLOT from the general user account

The general user account, GENUSER (password ING_GENUSER), can be used to run the PLOT program. When the PLOT option is chosen from the menu, a window is created in which PLOT is run, generating the prompt:
`. PLOT>`
When you exit from PLOT, you will be asked if you want to print your plot file. If you answer Y, you will be asked which printer you want to use (INT, WHT or JKT). To print the file again, run PLOT and immediately exit it, and answer the questions. This will only work if no-one else has used PLOT in the meantime and created a plot file themselves.

4 Simple use of PLOT

This section describes the use of the basic features of **PLOT** by means of a simple example. A transcript of the session reads:

```
RL>plot
PLOT>dir

Directory SYS$SYSDEVICE:[RL.PLOT.EXAMPLES]

ENC971017.DAT;1           25  20-OCT-1997 19:36:55.66

Total of 1 file, 25 blocks.
PLOT>data ENC971017.DAT
INT              19971017
```

```
Encoder log file
Number of data fields = 16

Key 1    Key 2    Label

         UTC      Time
HA       FERR1    HA Ferranti 1
HA       FERR2    HA Ferranti 2
HA       FERR3    HA Ferranti 3
HA       BALDWIN  HA Baldwin
DEC      HH1      Dec Heidenhain 1
DEC      HH2      Dec Heidenhain 2
DEC      BALDWIN  Declination Baldwin
CASS     ABS      Cassegrain Rotator
HA       POSERR   Hour Angle pos error
DEC      POSERR   Declinatio pos error
CASS     POSERR   Cassegrain pos error
HA       VEL_DEM  Hour Angle demand
DEC      VEL_DEM  Declinatio demand
CASS     VEL_DEM  Cassegrain demand
         WORMPHAS HA worm gear phase
    85 records in the data file
PLOT>dev xwindow
PLOT>read 1 utc
PLOT>read 2 dec hh1
PLOT>limits
PLOT>plot
PLOT>dev ps
Type <RETURN> for next page:
PLOT>plot
PLOT>dev xwindow
PLOT>mean 1 dec hh1 hh2
PLOT>subtract 2 dec hh1 hh2
PLOT>limits
PLOT>label Encoder difference log
PLOT>plot
PLOT>dev ps
Type <RETURN> for next page:
PLOT>plot
PLOT>exit
```

After starting up the program (as in the previous section), the `DIRECTORY` command is used to examine the files in the current directory [RL.PLOT.EXAMPLES]. There is one file (an INT encoder log for a long move in declination), and this is opened using the `DATA` command. The header listing shows 16 data fields (see Section B for a more detailed explanation). The first plot is of the value of one of the declination encoder heads against time. We start by selecting X windows as a graphics device with `dev xwindow`. We then read the two variables into axes 1 and 2 using `READ 1 UTC` and `READ 2 DEC HH1`. The plot limits are set around the data by: `LIMITS` and the plot is made with `PLOT`. To get a hard copy, we change the graphics device to a postscript file by typing `DEV PS` and repeat the `PLOT`. This gives a file pgplot.ps, which we copy somewhere safe from another window. The second example is slightly more complicated. We wish to plot the difference between the two declination encoder heads against their mean, in order to check consistency. To do this, we use the commands `MEAN` and `SUBTRACT`. We also add a comment using `LABEL`. The two resulting postscript plots are reproduced in Figure 1.

5 Modelling

5.1 General

5.2 Derotation Optics

The effects of misalignment errors of the Nasmyth derotation optics on the WHT may be modelled by fitting harmonic functions to image position measured using an autoguider. This is done by the command `FIT2D`. The procedure is as follows:

1. For the UES focal station, move the guide probe to the centre of the field. The GHRIL autoguider is fixed.
2. Position a star on the approximate centre of rotation using the TCS handset. The rotator should be tracking in DEROTATOR mode.
3. Tell the TCS which autoguider is in use using `AGSELECT` and start the autoguider centroiding with a short (<1 s) integration interval.
4. Turn on autoguider logging (`LOG AUTO ON`.)
5. Increment the demand sky position angle by 720° using the TCS handset (this corresponds to a full rotation of the optics.)
6. When the rotator has reached its final position, turn off the log with `LOG AUTO OFF`.
7. Start **PLOT** and set the default directory with `CD [WHT.DATA.AUTOGUIDER]`.
8. Identify the log file with `DIR`.
9. Open the file, read and plot the data:

   `DATA filename`

   `READ 1 AGX`

   `READ 2 AGY`

   `READ 3 UES POS` or `READ 3 GHRIL POS`

   `FIT2D` or `FIT2D RH`

   `SQUARE`

   `DEV XWINDOW` (or other graphics device)

   `PLOT`

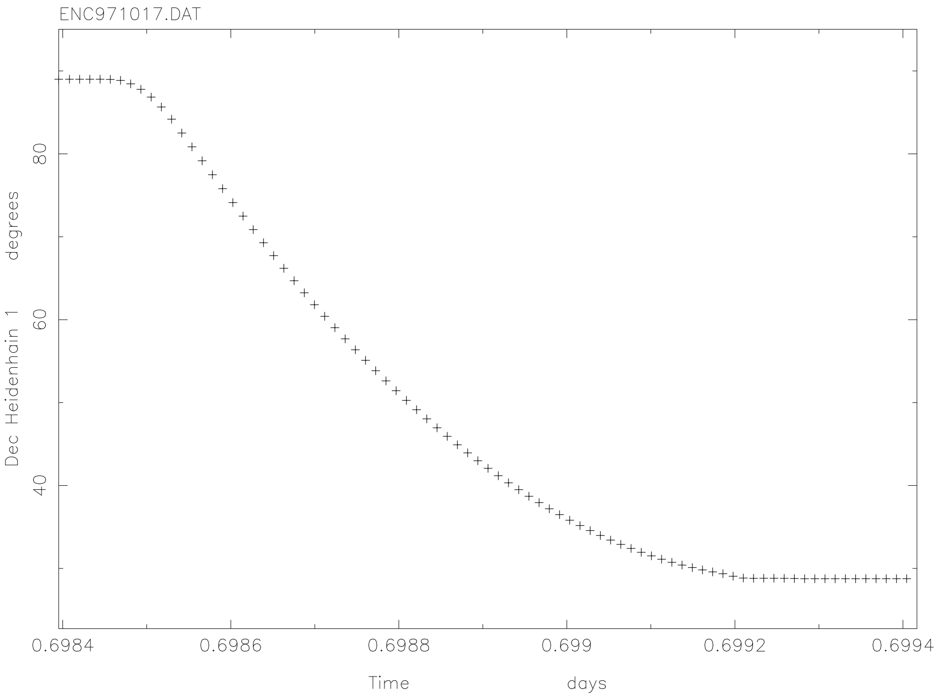
   `PLOT FIT`

10. Note the 6 numbers printed by `FIT2D`.

The optional handedness keyword for `FIT2D` sets the direction in the plot coordinate system corresponding to increasing mount position angle (this can be either clockwise or anticlockwise, depending on the autoguider set-up). It is know that the default (`LH`) works for the UES autoguider: GHRIL (for the IR and Optical de-rotators) requires `FIT2D RH`.

An example plot is shown in Figure 2. The fitted model is:

$$x = c_1 + c_3 \cos\theta + c_4 \sin\theta + c_5 \cos 2\theta + c_6 \sin 2\theta$$



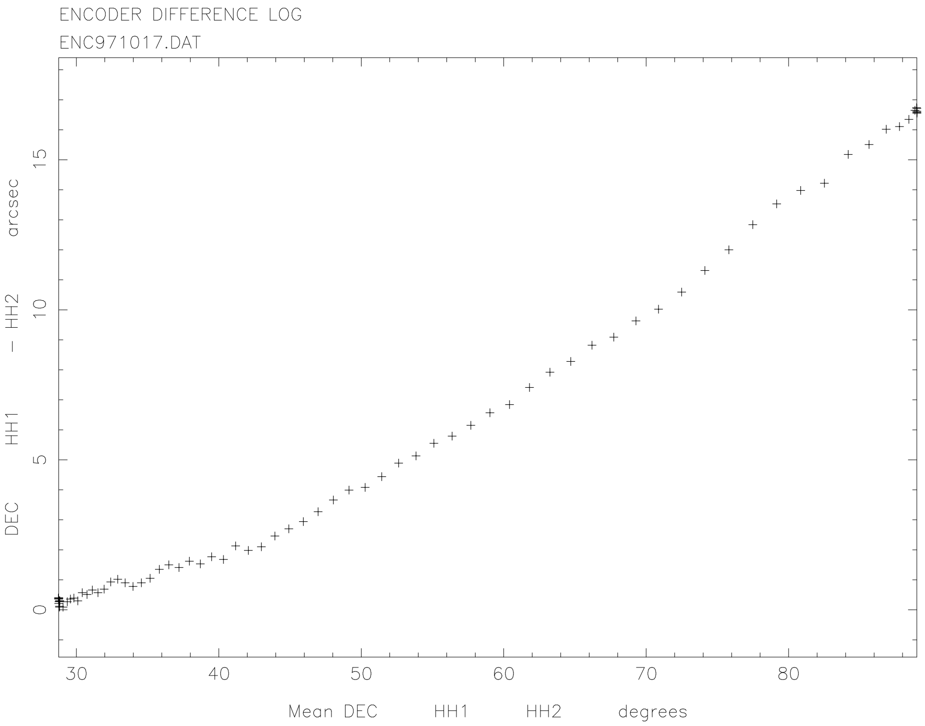


Figure 1: Two example plots from an INT encoder log, as described in the text.

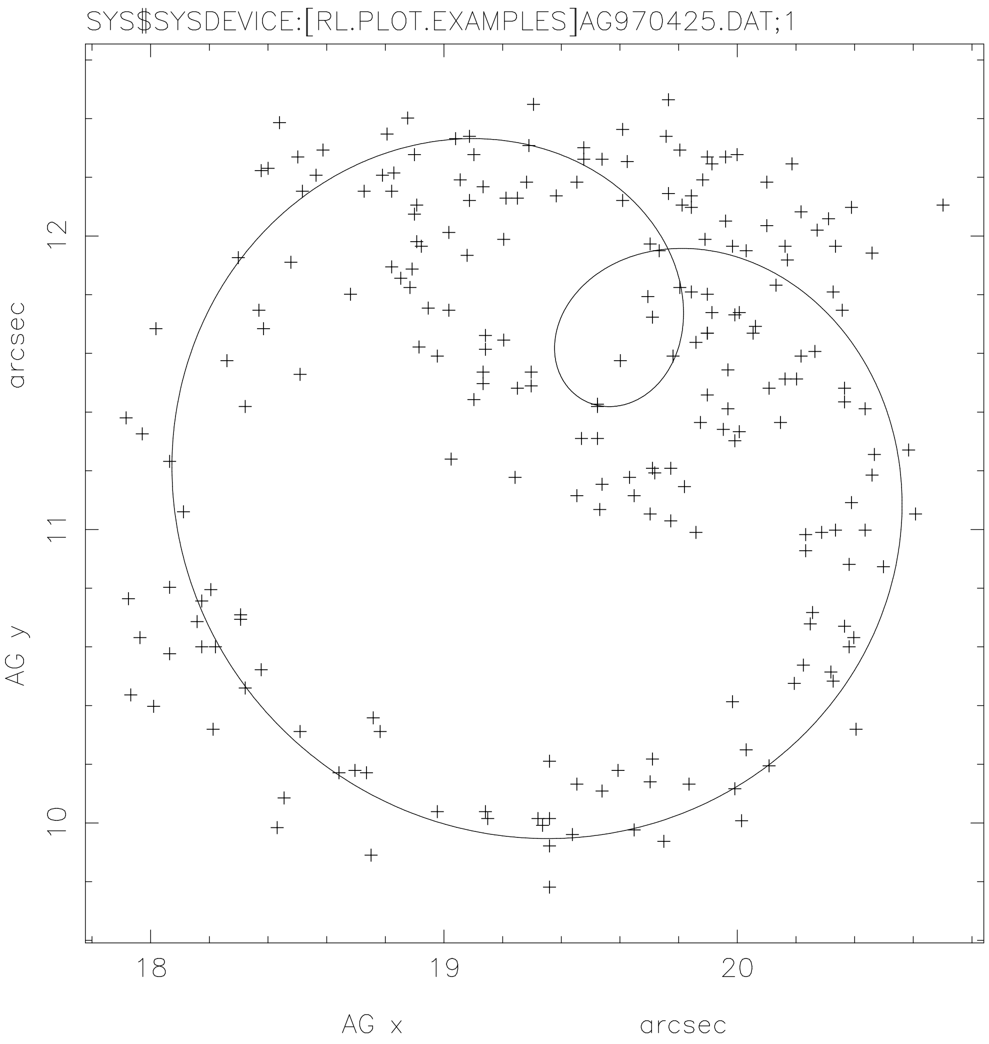


Figure 2: A plot of an autoguider log showing derotation optics errors at the UES focal station. The fitted model is shown as the solid line.

Table 2: Conversion between the output of `FIT2D` and the TCS correction parameters for the UES autoguiders. The coefficients c_3 and c_4 must use the radian (not the arc second) values. N represents the focal station and derotation mode.

TCS variable	Change to variable
POI_GD_NCOS1X(N)	$-c_3$
POI_GD_NSIN1X(N)	$-c_4$
POI_GD_NCOS1Y(N)	$+c_4$
POI_GD_NSIN1Y(N)	$-c_3$

$$y = c_2 + f[c_4 \cos\theta - c_3 \sin\theta + c_6 \cos 2\theta - c_5 \sin 2\theta]$$

where the suffices correspond to the order in which the coefficients are printed out and $f = +1$ for the default (LH) handedness and -1 for RH. c_1 and c_2 are the coordinates of the rotator centre on the autoguider chip. The first harmonic terms c_3 and c_4 must be included in the initialization file for the WHT (OWNINIT.FOR) after conversion to radians. The TCS variables be changed are given in Table 2. Note that the conversions from coefficient to TCS variable depend on the autoguider configuration: they have so far been determined only for the UES (moving probe and slit viewer.) The second harmonic terms are not included, since they can be removed by placing the star on the centre of rotation. The residual coefficients after careful calibration should be $\leq$0.05 arcsec.

5.2.1 How to Change the TCS variables

The TCS common block variables are changed in an iterative process to converge to a better and better de-rotation fit. The C3 and C4 values are added/subtracted to the current variables. When the fit is considered acceptable the TCS values are saved for inclusion in the OWNINIT.FOR file for a future release. An example GSEXAM script (See Page 18) may be modified and used to change the TCS variables. Note that the fortran index value (N) indicates the focal station and de-rotator mode in use.

5.3 Gear errors

6 Classified list of commands

6.1 General

- `CD` Change the current directory.
- `DIRECTORY` List the files in the current directory.
- `EXIT` Exit from **PLOT**.
- `HELP` Get help on a **PLOT** command.

6.2 Reading data

- `DATA` Define the data file to be read.
- `HEADER` List the header of the current data file.
- `MEAN` Take the mean of up to four data fields and write it into an axis of the data buffer.
- `READ` Read a data field and write to an axis of the data buffer.

- `SUBTRACT` Subtract two data fields and put the result in an axis of the data buffer.

6.3 Setting plot limits

- `BOX` Set a data range.
- `LIMITS` Find the extrema of the current data and set plot limits around them.
- `SQUARE` As `LIMITS`, but make axis lengths and scales identical.

6.4 Making a plot

- `DEVICE` Open a graphics device.
- `ERASE` Erase the current graphics device.
- `LABEL` Input a header to be written on the plot.
- `PLOT` Plot data, model or residuals.

6.5 Modelling

- `FIT2D` Fit first and second harmonic terms to derotation optics errors.

6.6 Text file output

- `OUTFILE` Open an ascii output file.
- `WRITE` Write data, model or residuals to an ascii output file.

7 Alphabetical Command Glossary

7.1 Syntax

All commands follow standard DCL syntax. Minimum match is supported for the first four characters of a command or keyword; following characters are ignored. Thus, `DE PS`, `DEVI PS` and `DEVILISH PS` will all be interpreted as `DEVICE PS`, whereas `D PS` is ambiguous and will generate an error message. Parameters (e.g. mechanisms, variables or graphics devices) must be given in full. None of commands, parameters or keywords are case-sensitive. Optional parameters are given in square brackets.

7.2 BOX

Set a data range for subsequent plots.

Format: `BOX xmin xmax ymin ymax`

Parameters: The parameters are the minimum and maximum values on the x and y axes of the plot, in units appropriate to the quantities to be displayed.

Example: `BOX 10.0 30.0 -3.5 3.5`

Comments: This command sets the limits of the area to be plotted in data units. No additional border is left (cf. `LIMITS`.)

7.3 CD

Change the default directory.

Format: `CD [directory_specification]`

Parameters: The directory specification is in standard VMS syntax and may be up to 40 characters long. Either absolute (e.g. [INT.DATA.ENCODERS]) or relative (e.g. [-.AUTOGUIDER]) specifications may be used. The latter are always defined with respect to the directory from which **PLOT** was started, *not* with respect to the current directory.

Defaults: If `directory_specification` is blank, then the current directory reverts to that from which **PLOT** was started.

Examples: `CD [INT.DATA.AUTOGUIDER]`
`CD`

Comments: This command changes the default directory used by `DATA` and `DIRECTORY`.

7.4 DATA

Open and validate a log file and display its header.

Format: `DATA filename`

Parameters: `filename` is the name of the log file produced by the TCS, in standard VMS syntax, up to 40 characters in length.

Defaults: If the extension is not specified, .DAT is assumed.

Examples: `DATA ENC971013`
`DATA [INT.DATA.AUTOGUIDER]AG980123.DAT;3`

Comments: This command opens the log file, reads and displays the header (see Sections 2.3, A and B for more details of the format) and checks the data format for consistency. The directory specification should either be set using `CD` or given directly to `DATA`, but *not* both.

7.5 DEVICE

Select and open a graphics device.

Format: `DEVICE device_name`

Parameters: `device_name` is the name of the graphics device as known to native PGPLOT *with the leading / removed* (this is done to avoid confusion with VMS qualifiers). The most useful devices are:

`XWINDOW`	an X window which does not persist for re-use after `EXIT`.
`XSERVE`	an X window which does persist for re-use.
`PS`	A postscript file, landscape orientation.
`VPS`	A postscript file, portrait orientation.

Examples: `DEVICE XWINDOW`
`DEV PS`

Comments: This command closes any currently-open graphics device and opens a new one. The postscript options always produce a file called pgplot.ps and `DEV PS` will overwrite any file of that name already present in the directory. Beware of overwriting a file before you rename it. If you plan to make X windows plots and are running from a private account, type `XDISPLAY` at the DCL prompt before starting **PLOT**. To get a list of supported devices type `DEVICE`, followed by `?` at the `_PGPLOT Device name:` prompt.

7.6 DIRECTORY

List the files in the current directory.

Format: `DIRECTORY`

Comments: This command executes the VMS command `DIRECTORY/DATE/SIZE` to give a listing of filenames, version numbers, sizes (in blocks) and last modification date for the current directory. Use `CD` to change the current directory from within **PLOT**. Note that write access is needed to the directory from which **PLOT** was started in order to create temporary files.

7.7 ERASE

Erase the current graphics device.

Format: `ERASE`

Comments: This command is useful to repeat a plot with a different label or with a different superposition of model and data, but without changing any other parameters.

7.8 EXIT

Exit from **PLOT** and return to the DCL prompt or TCS menu.

Format: `EXIT`

7.9 FIT2D

Fit a model to derotation optics errors.

Format: `FIT2D [handedness]`

Examples: `FIT2D`
`FIT2D RH`

Keywords: The optional handedness keyword can be either `LH` (default) or `RH`. It sets the direction of rotation corresponding to increase of mount position angle in the plot coordinate system and depends on the autoguider setup.

Comments: This command fits a 6-parameter model to image position data from an autoguider or TV. It is intended for modelling errors due to misalignment of image derotation optics, specifically at the WHT Nasmyth foci. The x and y coordinates and the rotator position angle must be read into axes 1, 2 and 3, respectively, before issuing this command. See Section 5 for more details.

7.10 HEADER

Display the header of the current data file.

Format: `HEADER`

Comments: This command produces a listing of the header for the current data file (identical to that output by `DATA` when the file is initially opened.

7.11 HELP

Provide information about the available PLOT commands.

Format: `HELP [topic [subtopic]...]`

Defaults: Lists all available commands.

Parameters: The command you wish to receive help on.

Keywords: None.

Examples: `HELP DEV GRAPH`

Comments: Invoking the `HELP` command initiates an interactive dialogue with the user interface HELP library, a normal VMS HELP library. In response to a Topic? or subtopic prompt you may:

- Type the name of the command/topic for which you need help;
- Type a question mark (?) to redisplay recently requested text;
- Press the RETURN key one or more times to exit from HELP;
- Press CTRL–Z once to exit from HELP;
- Abbreviate any topic name, but note that ambiguous abbreviations result in *all* matches being displayed.

7.12 LABEL

Input a comment to be printed on subsequent plots.

Format: `LABEL [Comment]`

Parameters: `Comment` is a text string of up to 70 characters which will be printed at the top of subsequent plots. Embedded spaces are allowed and should not be enclosed in quotation marks.

Default: If `Comment` is not specified, the label is left blank.

Examples:
```
LABEL Encoder log fit the 20th
LABEL
```

Comments: The plot label is blanked by the scaling commands `BOX`, `LIMITS` and `SQUARE` but not by `PLOT`, so successive executions of `PLOT` with different graphics devices will have the same label. To repeat a plot without the label use `LABEL` followed by `ERASE` and `PLOT`.

7.13 LIMITS

Set the plot limits to fit around the current data.

Format: `LIMITS keyword`

Keywords:

`DATA`	use the original data.
`MODEL`	use a model fitted to the data.
`RESIDUALS`	use the difference between data and model.

Defaults: `DATA` is the default keyword.

Examples:
```
LIMITS
LIMITS MODEL
```

Comments: The plot limits are set exactly equal to the range on the horizontal axis and 10% larger on the vertical one.

7.14 MEAN

Take the mean of up to four data items and read them into the specified axis.

Format: `MEAN axis mech var1 var2 [var3 [var4]]`

Parameters:

`axis`	is the axis into which the mean value is to be read (1, 2 or 3).
`mech`	is the mechanism name (all data to be averaged must have the same mechanism name).
`var1 - var4`	are the names of the variables to be averaged. The first two are required; the latter two optional.

Examples: `MEAN 1 HA FERR1 FERR2 FERR3`

Comments: This command is most often used to take the mean of several independent encoders on the same drive In this case, `mech` is the name of the drive (`HA`, `AZIMUTH`, etc.) and `var1 - var4` are the corresponding encoder names.

7.15 OUTFILE

Open a file for ascii output.

Format: `OUTFILE filename`

Parameters: `filename` is the output file name, in standard VMS syntax.

Examples: `OUTFILE JUNK.DAT`

Comments: `WRITE` is used to write ascii data to the file.

7.16 PLOT

Make a plot.

Format: `PLOT keyword`

Keywords:

`DATA`	plot the original data.
`MODEL`	plot a model derived by fitting the data on axis 2 as a function of that on axis 1.
`RESIDUALS`	plot the difference between data and model.
`FIT`	plot the results of a two- dimensional fit to the data on axes 1 and 2 as functions of that on axis 3.

Defaults: `DATA` is the default keyword.

Examples:
```
PLOT
PLOT RESIDUALS
```

Comments: `PLOT` will make a plot of the data on axis 2 against that on axis 1 on the specified graphics device. It requires that data have been read into both axes, and that the plot limits have been set. The `MODEL` and `RESIDUALS` keywords require that a model has been fitted to the data using `FIT1D`. Likewise, `FIT` requires `FIT2D` to be run first. The name of the log file from which the data were taken and (optionally) a comment input using `LABEL` are printed at the top of the plot.

7.17 READ

Read data into a specified axis.

Format: `READ axis [mech] var`

Parameters:

`axis`	is the number of the axis into which the data are to be read.
`mech`	is the name of the mechanism, if specified.
`var`	is the name of the variable. If the variable has no associated mechanism (e.g. UTC), then the `mech` field is left blank.

Examples:
```
READ 1 UTC
READ 2 HA FERR1
```

7.18 SQUARE

Set the plot limits so that both axes have the same length and scale and the data are just included.

Format: `SQUARE keyword`

Keywords:

`DATA`	use the original data.
`MODEL`	use a model fitted to the data.
`RESIDUAL`	use the difference between data and model.

Defaults: `DATA` is the default keyword.

Examples:
```
SQUARE
SQUARE MODEL
```

Comments: A 10% border is left around the data. This command is most useful where equal scales in the two axes are helpful, e.g. plotting autoguider and TV image data and derotation optics fits.

7.19 SUBTRACT

Subtract two data items and read the result into the specified axis.

Format: `SUBTRACT axis mech var1 var2`

Parameters:

`axis` is the axis into which the mean value is to be read (1, 2 or 3).

`mech` is the mechanism name (all data to be averaged must have the same mechanism name).

`var1` and `var2` are the names of the variables to be subtracted (in the sense `var1 − var2`).

Examples: `SUBTRACT 1 HA FERR1 FERR2`

Comments: This command is most often used to take the difference of two independent encoders on the same drive In this case, `mech` is the name of the drive (`HA`, `AZIMUTH`, etc.) and `var1` and `var2` are the corresponding encoder names.

7.20 WRITE

Write data to a file in ascii format.

Format: `WRITE keyword`

Keywords:

`DATA` write out the original data.

`MODEL` write out a model derived by fitting the data on axis 2 as a function of that on axis 1.

`RESIDUALS` write out the difference between data and model.

Defaults: `DATA` is the default keyword.

Examples: `WRITE`

`WRITE RESIDUALS`

Comments: The data are written out in three columns (corresponding to axes 1 – 3) in display units using Fortan F13.6 format. The file must previously have been opened with `OUTFILE` and at least one axis must have data.

A WHT log formats

A.1 Encoder logs

Table 3: WHT encoder log fields for the Cassegrain focus. At other foci, the mechanism name CASS is replaced by PRIME, UES or GHRIL as appropriate.

Mechanism	Variable	Description	Units
	UTC	Time	days (displayed as seconds)
AZIMUTH	ABS	Azimuth Absolute encoder	degrees
AZIMUTH	GEAR	Azimuth Gear encoder	degrees
AZIMUTH	ROLL	Azimuth Roller encoder	degrees
AZIMUTH	TAPE1	Azimuth Tape encoder head 1	degrees
AZIMUTH	TAPE2	Azimuth Tape encoder head 2	degrees
AZIMUTH	TAPE3	Azimuth Tape encoder head 3	degrees
AZIMUTH	TAPE4	Azimuth Tape encoder head 4	degrees
ALTITUDE	ABS	Elevation Absolute encoder	degrees
ALTITUDE	GEAR	Elevation Gear encoder	degrees
ALTITUDE	ROLL	Elevation Roller encoder	degrees
CASS	ABS	Cassegrain Absolute encoder	degrees
CASS	INC	Cass Incremental encoder	degrees
AZIMUTH	POSERR	Azimuth servo position error	arcsec
ALTITUDE	POSERR	Altitude servo position error	arcsec
CASS	POSERR	Cassegrain rotator servo position error	arcsec
AZIMUTH	VEL_DEM	Azimuth velocity demand	degrees / s
ALTITUDE	VEL_DEM	Altitude velocity demand	degrees / s
CASS	VEL_DEM	Cassegrain rotator velocity demand	degrees / s
	HORDIS1	Horizontal displacement transducer 1	microns
	HORDIS2	Horizontal displacement transducer 2	microns
	SECPOS1	Secondary mirror position transducer 1	microns
	SECPOS2	Secondary mirror position transducer 2	microns
	SECPOS3	Secondary mirror position transducer 3	microns

A.2 Autoguider and TV logs

Table 4: WHT autoguider log fields for the Prime focus. At other foci, the mechanism name PRIME is replaced by CASS, UES or GHRIL, as appropriate. TV log files have the same format except that the pixel coordinates are called TVX and TVY.

Mechanism	Variable	Description	Units
	UTC	Time	days (displayed as seconds)
	AGX	Autoguider x pixel coordinate	arcsec
	AGY	Autoguider y pixel coordinate	arcsec
AZIMUTH	POS	Azimuth	degrees
ALTITUDE	POS	Altitude	degrees
PRIME	POS	Prime rotator position angle	degrees
AZIMUTH	POSERR	Azimuth servo position error	arcsec
ALTITUDE	POSERR	Altitude servo position error	arcsec
PRIME	POSERR	Prime rotator servo position error	arcsec

B INT log formats

B.1 Encoder logs

Table 5: INT encoder log fields for the Cassegrain focus. At the prime focus, the mechanism name CASS is replaced by PRIME.

Mechanism	Variable	Description	Units
	UTC	Time	days (displayed as seconds)
HA	FERR1	HA Ferranti encoder head 1	hours
HA	FERR2	HA Ferranti encoder head 2	hours
HA	FERR3	HA Ferranti encoder head 3	hours
HA	BALDWIN	HA Baldwin encoder	hours
DEC	HH1	Dec Heidenhain encoder head 1	degrees
DEC	HH2	Dec Heidenhain encoder head 2	degrees
DEC	BALDWIN	Declination Baldwin encoder	degrees
CASS	ABS	Cassegrain Rotator encoder	degrees
HA	POSERR	Hour Angle servo position error	arcsec
DEC	POSERR	Declination servo position error	arcsec
CASS	POSERR	Cassegrain servo position error	arcsec
HA	VEL_DEM	Hour Angle velocity demand	degrees / s
DEC	VEL_DEM	Declination velocity demand	degrees / s
CASS	VEL_DEM	Cassegrain velocity demand	degrees / s
	WORMPHAS	HA worm gear phase	degrees

B.2 Autoguider logs

Table 6: INT autoguider log fields for the Prime focus. At the Cassegrain focus, the mechanism name PRIME is replaced by CASS.

Mechanism	Variable	Description	Units
	UTC	Time	days (displayed as seconds)
	AGX	Autoguider x pixel coordinate	arcsec
	AGY	Autoguider y pixel coordinate	arcsec
HA	POS	Hour Angle	hours
DEC	POS	Declination	degrees
PRIME	POS	Prime rotator position angle	degrees
HA	POSERR	Hour Angle servo position error	arcsec
DEC	POSERR	Declination servo position error	arcsec
PRIME	POSERR	Prime rotator servo position error	arcsec
	WORMPHAS	HA worm gear phase	degrees

B.3 De-rotator GSEXAM Example Calibration Script

```
! fjg 04 Apr 1999 - data - IR DEROT
EXA POI_GD_NCOS1X(4)
EXA POI_GD_NSIN1X(4)
EXA POI_GD_NCOS1Y(4)
EXA POI_GD_NSIN1Y(4)
DEP POI_GD_NCOS1X(4) = POI_GD_NCOS1X(4) + 0.615220E-06      ! -C3
DEP POI_GD_NSIN1X(4) = POI_GD_NSIN1X(4) - 0.791733E-06      ! -C4
DEP POI_GD_NCOS1Y(4) = POI_GD_NCOS1Y(4) + 0.791733E-06      ! +C4
DEP POI_GD_NSIN1Y(4) = POI_GD_NSIN1Y(4) + 0.615220E-06      ! -C3
EXA POI_GD_NCOS1X(4)
EXA POI_GD_NSIN1X(4)
EXA POI_GD_NCOS1Y(4)
EXA POI_GD_NSIN1Y(4)
```