How to use "susipoint" to find siderostat pointing solutions

W. J. Tango

5 September 2007

1 Introduction

The purpose of finding good pointing solutions is to minimize the time needed to acquire stars. Due to a variety of mechanical effects the quality of the pointing will gradually deteriorate, and consequently a redetermination of the pointing parameters needs to be done occasionally for all siderostats.

These notes are intended as a user's manual for `susipoint`. The theory on which the method is based, and a discussion of the iterative procedure outlined here, can be found in the companion document *The SUSI Astrometric Model and "the pointing problem."*

The program uses pointing data that are saved in special pointing files. The data can be collected during a normal observing session or one devoted specifically to collecting pointing data. The SUSI control programs `astromod` and `sidcon` have been modified to facilitate this.

I have developed a procedure for using `susipoint` that (a) rejects outliers, (b) finds the pointing solution, and (c) is easy to use. All one needs is the `susipoint` program and a text editor (`emacs`, `vi`, `kdedit`, ...). A spreadsheet program like Excel is also very useful for visualizing the results.

It is not necessary to know what the various model parameters are but users should be aware that, except for θ_2, they are all small; unusually large values for any of the other parameters can indicate problems either with the fitting process or the data. The expected values for the various model parameters are summarized here:

- The autocollimation offsets should be very small ($\sim 0.01^\circ$). These offsets are included `susipoint` on an experimental basis, and it is probably best to leave them set to 0 (the default).
- The δ_1, δ_2 and θ_1 angles are of the order of 0.1°.
- The angle θ_2 specifies a "line of nodes" and can have any value between 0° and 360°.
- The encoder offsets are typically of the order of $1^\circ \sim 2^\circ$.
- Except for θ_2 any of these angles can be negative.
- The two angles θ_1 and θ_2 are ambiguous; the solutions $(-\theta_1,\ \theta_2)$ and $(\theta_1,\ \theta_2 + \pi/2)$ are the same.

2 Collecting pointing data

2.1 How to collect pointing data

The pointing data are saved by `sidcon`, the program that controls the siderostats. There are three menu items that control the acquisition of pointing data (the commands in brackets can be entered from the `sidcon` command line if one prefers). The items are located in the TRACKING menu. The commands and their operation are:

- "Save pointing data" [`spd`] creates a pointing data file and data will be saved every 10 seconds whenever the station is tracking. *If the pointing file for the particular siderostat already exists, it will be overwritten and any previous data will be lost.*
- "Close pointing file" [`cpf`] closes the current pointing data file.
- "Reopen pointing file" [`rpf`] is used when you wish to append to an existing pointing file. No data will be lost.

The operation of these commands is pretty much self-explanatory. However, it is important to note that *the* `spd` *command will destroy any previous pointing data for the particular siderostat.* Normally, the `spd` command should only be used once during the night when the file is initially created. Use the close and reopen commands to stop and resume data collection.

Note also that the collection of pointing data is automatically suspended when the siderostat is not locked. It is *not* necessary to close the pointing file when slewing to another star, for example. If the siderostats lose lock because of cloud, etc., no data will be collected. *If* `sidcon` *crashes because of a power failure, RTP errors, etc., use "reopen pointing file" not "save pointing data" to continue collecting pointing data!*

> **CAUTION:** When a program writes to a disk file the operating system normally saves the data to a buffer. Data is written to the physical disk periodically by the operating system. If a program crashes all the data in the buffer is lost and the amount of data lost is unpredictable. For this reason it is advisable to close and reopen the pointing files occasionally. The buffer is always flushed when a file is closed.

Pointing data can be collected at any time. However, if the purpose is to improve a pointing solution the most important thing is to observe a number of stars over as much of the sky as possible. A good strategy is to select a group of stars that have roughly the same right ascension but cover a wide range of declination. Start collecting data when the stars are in the East, and check them regularly as they move to the West. Where practical data should be collected using both the North and South periscope configurations.

Prior to observing, do a careful alignment and record the current autocollimation positions in the `sidstats` file using the command

```
>susiedit sidstats
```

It is *not* necessary to collect long data runs. Five to ten minutes per object is probably sufficient. The close/reopen functions make this relatively easy. The same principle applies when observing a "science" target: collect ten minutes of pointing data every 30 minutes or so. This keeps the size of the pointing files manageable and minimizes the amount of editing that needs to be done on the files.

Please watch the elevation and azimuth encoder displays before and during the collection of pointing data! Do not collect data if the encoder display "jumps" suddenly to a whole number of degrees and note any odd behavior in the log. The pointing files use the declination and hour angle to identify points, so any comment should include the hour angle.

Of course one can collect data from more than one siderostat during the night.

N.B.: It is essential that the collection of pointing data does not impact on the stability of the siderostat servo loop and consequently all the pointing data operations are implemented using low level functions. This is the reason why:

- There is no on-screen messaging to indicate when the pointing files are open.
- The pointing files are saved in the "`tmp`" directory used by the astrometric model.
- Data points in the pointing file are identified by the star's declination and hour angle, rather than the more user friendly HR number and time.

2.2 What to do at the end of a session

At the end of an observing session with a particular siderostat, use the "close pointing file" command to close the pointing file.

At the end of the night the pointing files must be copied to the standard SUSI data directory.

The pointing files are located in the directory

```
/usr/local/susi/tmp/
```

and are called

```
xx_point.dat
xx_model.dat
```

where `xx` is the station code (e.g., `n1`, `s3`). The files that have been used during the night should be copied to

```
~susi/data/pointing/YYMMDD/
```

where `YYMMDD` is the observation date. There may be several pointing files in the `tmp` directory; copy only the ones that were updated during the night.

2.3 Analysis

The data should be copied to the analysis directory in the usual way. There is a directory called

```
~susi/analysis/pointing
```

Navigate to this directory and run the command

```
cp -r ~susi/data/pointing/YYMMDD .
```

to copy the pointing directory `YYMMDD`. If you have pointing data from several nights you might want to create the directory `YYMMdd-DD` where `dd` and `DD` are the inclusive dates. The `xx_point.dat` files from the various nights should be *merged* into a single file called something like `xx_merged.dat`. The `xx_model.dat` files should all be the same for the different nights so any one of them can be used.

If there are data for more than one siderostat make subdirectories for each siderostat; for example, if there is pointing data for N1 and S3 create the subdirectories `YYMMDD/n1` and `YYMMDD/s3`.

3 Synopsis of susipoint

The usage is:

```
>susipoint datafile modelfile results cleandata
```

where

datafile is a standard SUSI pointing data file. It should contain no more than 5000 data points.

modelfile is a file listing the starting values for the model parameters. The program uses these as the initial guess when it searches for the best fit. There is a control flag associated with each parameter. If the flag is 0 the parameter will be held fixed. If the flag is 1 `susipoint` will include the parameter when it searches for the best fit.

If *all* the control flags are 0 `susipoint` uses the parameters given in the file to calculate the residuals. No optimization is performed.

Normally one would use the model file `xx_model.dat` generated by the astrometric model. The fixed/free flags will initially be all zero.

results has three sections:

1. The first section contains statistics about the fitting procedure. This information is echoed to the console.
2. The next section lists the fitting parameters in the standard "model file" format. This section can be cut out to make a new model file, if necessary.
3. The third section is a listing of the azimuth and elevation residuals.

cleandata is a copy of the original data file, with any "dodgy" points commented out.

A full optimization with 1000 data points takes about 30 seconds.

4 Analysis of the data

CAUTION: The first two fitting parameters labeled `acol az offset` and `acol el offset` are developmental and should normally be kept fixed and set to zero.

Prior to analyzing the data you may need to edit or prune the raw pointing data file. In most cases this shouldn't be necessary. As noted previously it may be advantageous to merge the data from several nights. Pruning and merging may be done using your favorite editing program. A spreadsheet utility (Excel or the Open Office equivalent) or IDL are particularly useful, since their graphical capabilities make it easier to identify dodgy points. As an example let's assume that we have a set of pointing data for siderostat S2. Let's call the initial merged and edited file `s2_0.dat`.

The recommended procedure involves a minimum of two passes with `susipoint`. In the first pass, all the model parameters are held fixed and `susipoint` calculates the residuals with respect to the current pointing solution. The `xx_model.dat` files initially have all the fixed/free flags set to fixed, so the first step is to run the command

```
susipoint s2_0.dat s2_model.dat results1 s2_1.dat
```

The `results` file details the results of the fitting procedure, and lists the residuals. The file `s2_1.dat` contains the original data with any dodgy points commented out.

If there are no rejected points after this pass, the analysis is repeated, but this time the parameters are allowed to vary. The result should be a new and improved model for the siderostat. The rms errors should be of the order of 10 arcsec or less.

If there are a significant number of dodgy points the second pass is run on the "cleaned" data set, which has the bad points removed:

```
susipoint s2_1.dat s2_model.dat results2 s2_2.dat
```

This will produce a preliminary solution for the siderostat model. A new model file should be created with the new solution (this can be done by cutting and pasting from the `results2` file). Let's call this file `s2_model2.dat`.

The residuals for the original, full data set are recomputed using this preliminary solution, with the fitting parameters all held fixed.

```
susipoint s2_0.dat s2_model1.dat results3 s2_3.dat
```

More often than not the number of bad points will fall to 0, indicating that the current pointing model is bad. The data are processed a final time, now allowing the parameters to be optimized:

- Edit the `s2_model1.dat` file and change the fixed/free flags to 1.
- Rerun the last `susipoint` command.

This procedure is best explained using a flow chart. See Fig. 4.1.

It is strongly advisable to plot the residuals (azimuth vs. elevation residuals) for the final fit. One may wish to experiment with fixing one or more parameters, etc.

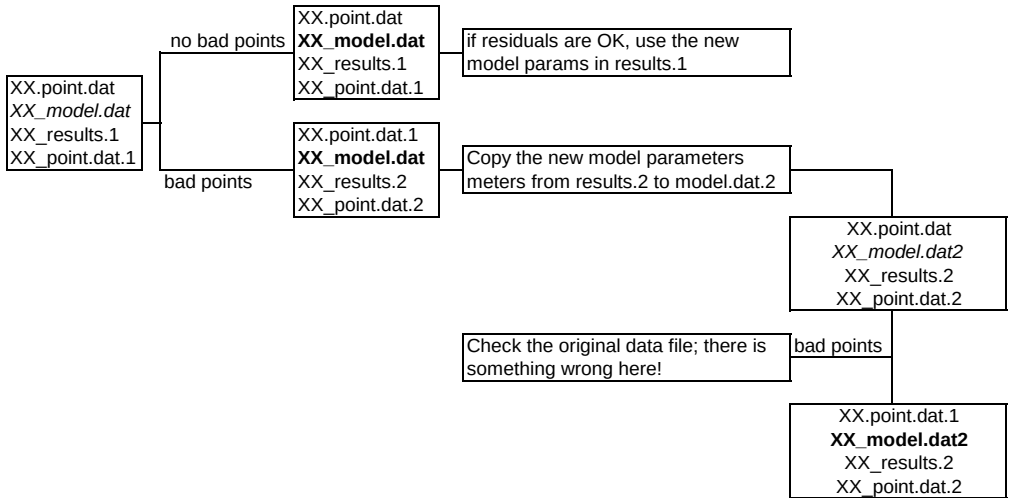

Figure 4.1: A flow chart illustrating how the pointing data should be processed. Each box with the four data files means "run `susipoint` using these files." *Italics* means that the flags in the model file should all be set to zero (no optimization); **bold** indicates that the flags are set to 1 (optimization turned on).

As noted above, there is an ambiguity in the angles θ_1 and θ_2. It is not unusual for θ_2 to jump by approximately 180° from one iteration to another.

If the solution looks good, run the command

```
>susiedit sidstats
```

Scroll or search to find the data block corresponding to the relevant siderostat, and change the model parameters to the ones you have just found (note: `susiedit` uses the `vi` editor; sorry about that). Save the file when you have finished. You will be prompted for a comment. Type in something like this:

```
N4: updated pointing 14/04/05. All sky, 1442 points.
S3: updated pointing 14/04/05. All sky, 1352 points.
```

Note that there is a `README` template file in `susi/pointing`. When you have finished, please copy this file to your analysis directory and edit in order to document what you have done.