MATTHEW WILKINSON

SPACE GEODESY FACILITY,
HERSTMONCEUX, UK

Automatically and Consistently Detecting and Extracting SLR Measurements for Every Satellite Pass

British Geological Survey

Natural Environment Research Council

INTRODUCTION

Every SLR observation begins as the telescope tracks along a satellite's predicted path. Then the laser fires short pulses and the detector is armed for returning signals.

Acquiring returns requires clear skies and often some telescope alignment or correction to the prediction.

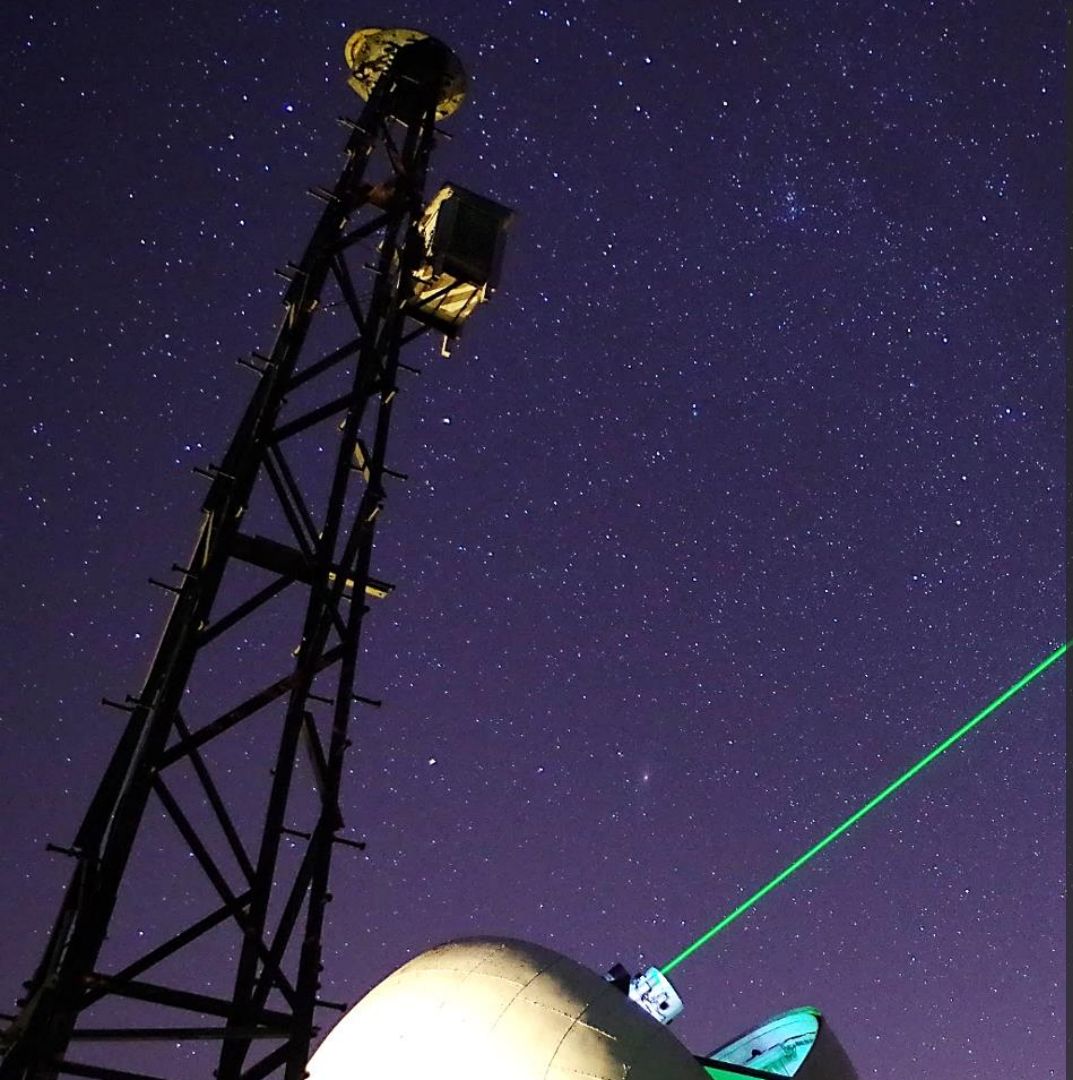
Later on, the recorded dataset must undergo post-processing to separate the laser range returns from the background noise.

INTRODUCTION

The separation of signal from noise is made more difficult with weaker return signals, intermittent data flows and greater background noise levels due to sky brightness.

A process for reliably and automatically extracting SLR returns from raw data files is under development and testing at the SGF, Herstmonceux.

A multi-processing central task manager was developed and is also described.



CONTENTS

- Automatic SLR data reduction
- Real-time track detection
- Initial residuals from orbit correction using orbitNP.py
- Track selection
- Return rate filtering
- Multi-processing task manager

WHY AUTOMATE TRACK EXTRACTION?

Many SLR stations in the ILRS network are automated to various degrees.

There are stations that operate automatic scheduling, automatic searching and automatic track detection.

Some SLR stations can be left unattended to track a schedule of satellites in the clear sky opportunities.

If such automated systems were allowed to take the final step of submitting data files then the reduction process would need to be very reliable.

WHY AUTOMATE TRACK EXTRACTION?

If a process has to be done manually, it will take some observer time. This task time accumulates over many passes, over days and years.

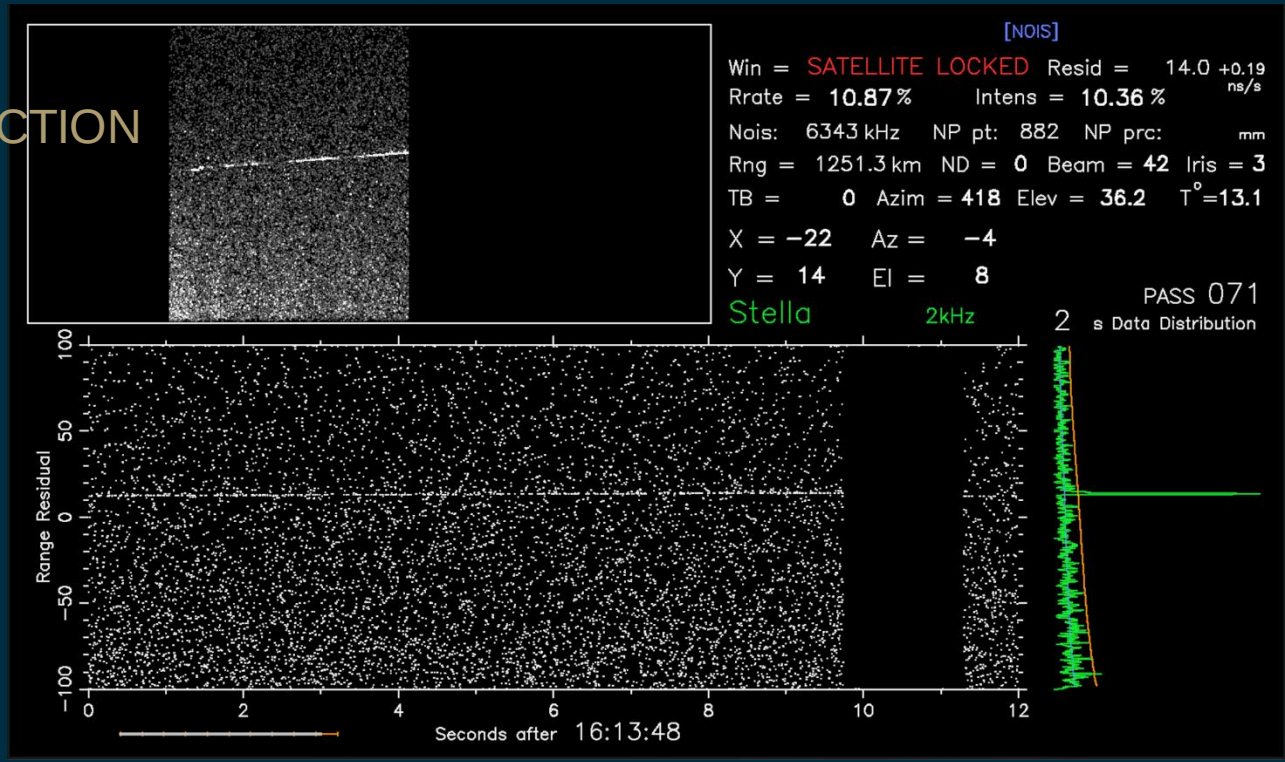
Manual processing must also wait for availability in the observing schedule. This could be a while given the increase in SLR targets and the significant impact of Lares-2 on the schedule.

This reduction process is being developed using the Riga A033-ET event timer which is installed at the SGF in parallel. The SGF is a kHz SLR station.

REAL-TIME TRACK DETECTION

Range residuals are plotted for the observer during a pass.

Satellite track shows in a short-term histogram plot of the residuals, which is continually refreshed.

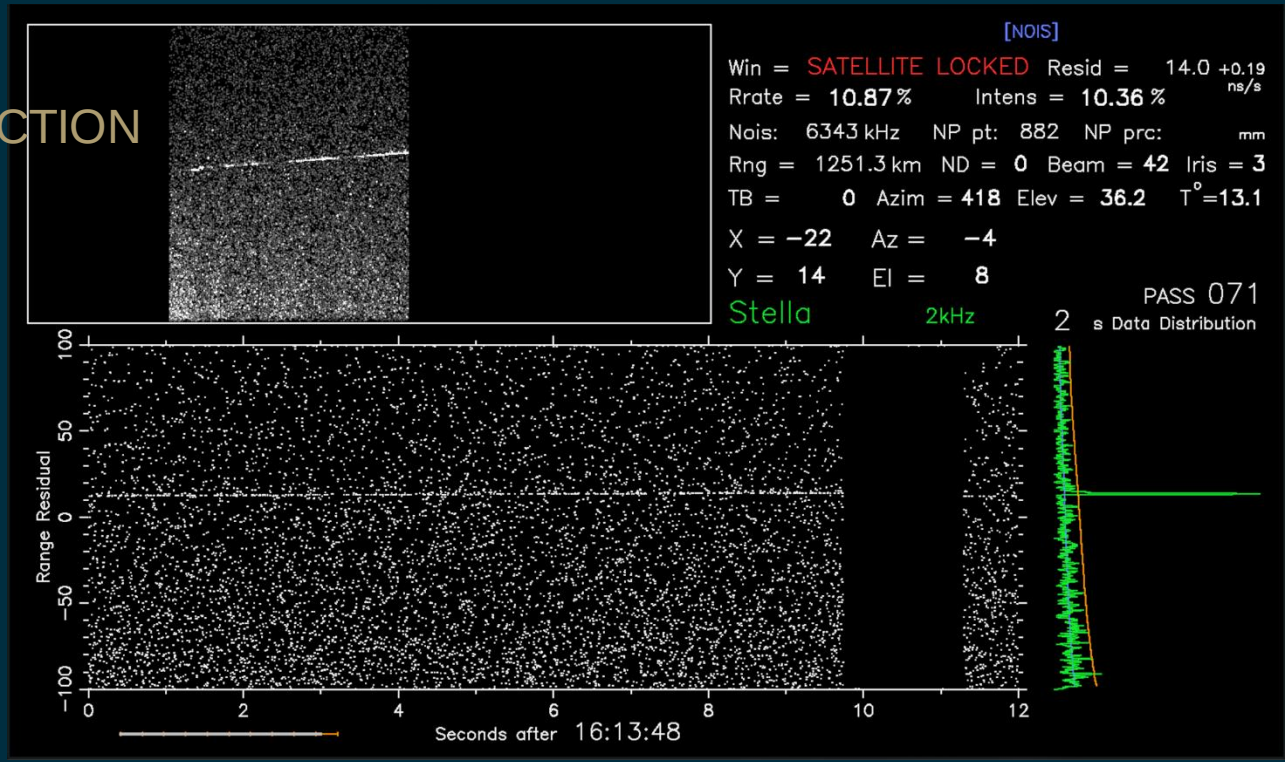


Track is identified in real-time from a peak above a set threshold. Track detection has been performed for many years at Herstmonceux.

REAL-TIME TRACK DETECTION

When track is detected, residuals are recorded and a polynomial is fitted to flatten returns for easy reduction.

In addition, for the new reduction process, track **ranges** are now recorded.



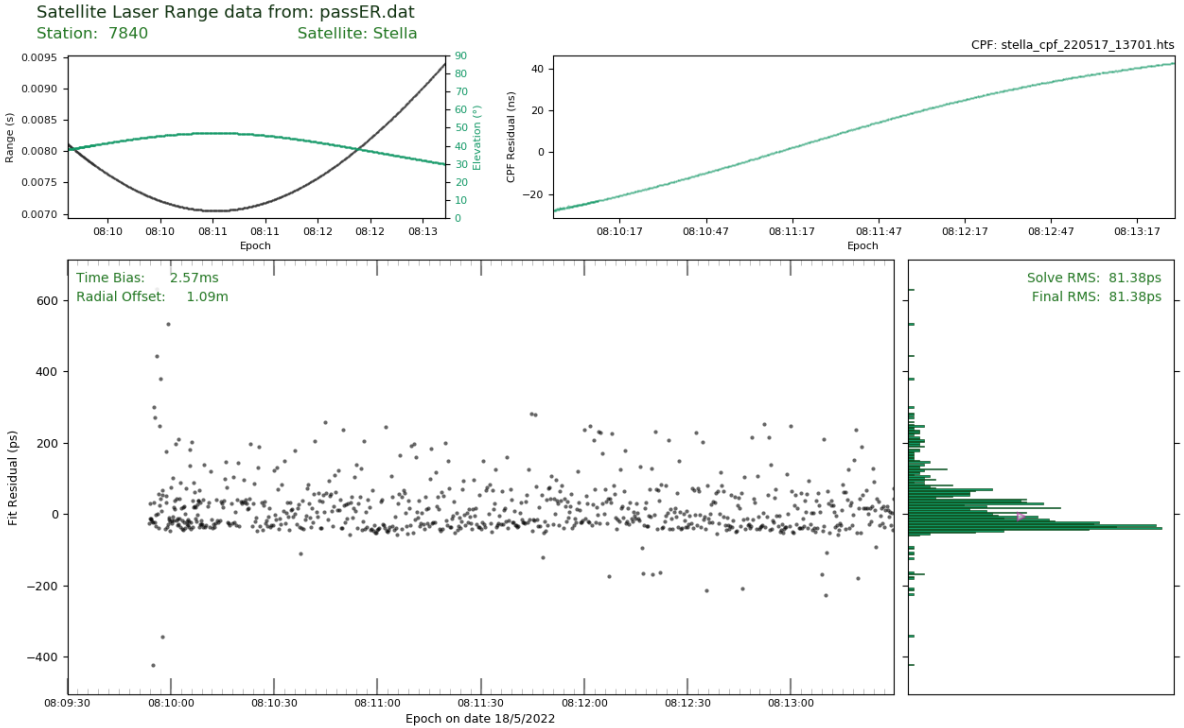
These ranges can be used in an orbit solution to produce residuals.

ORBITAL RESIDUALS

The epoch-range data recorded during the pass can be used as an input to the orbitNP.py program.

This flattens the track residuals and outputs the time bias and radial corrections.

This requires the pass calibration and meteorological records.



ORBITNP.PY

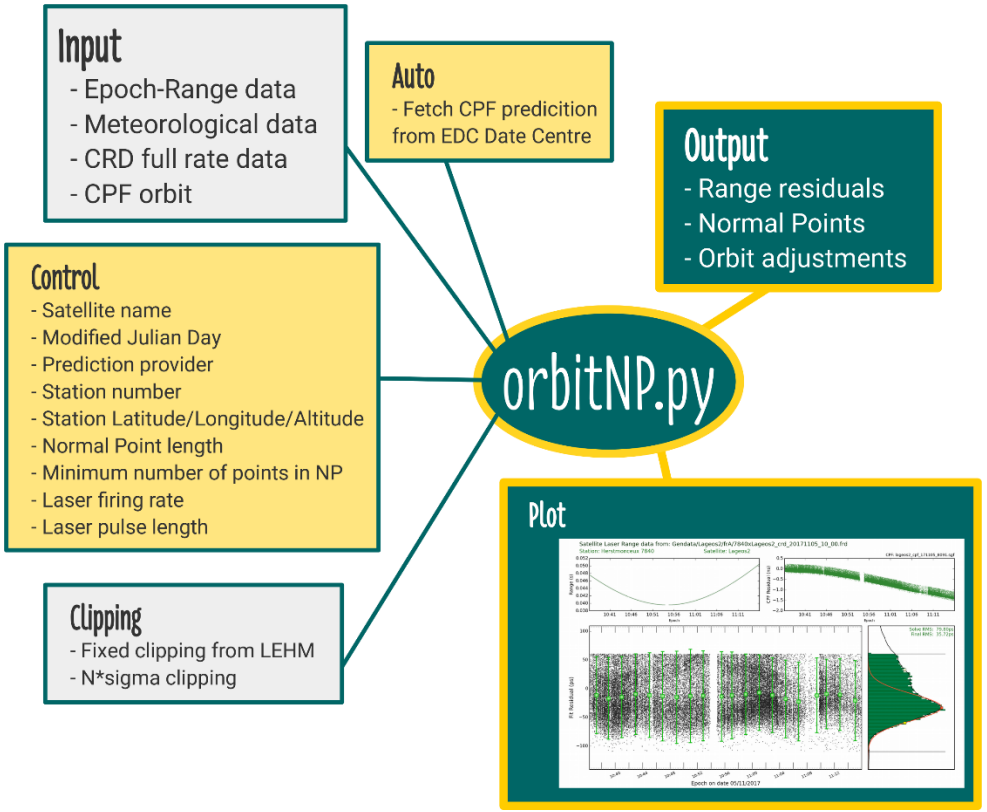
OrbitNP.py takes raw epoch-range data as an input.

It requires additional information such as the target name, modified julian day and a reference orbit CPF prediction.

To produce flattened range residuals, time bias and radial bias corrections to the reference orbit are solved for.

These can be output to a file for reference.

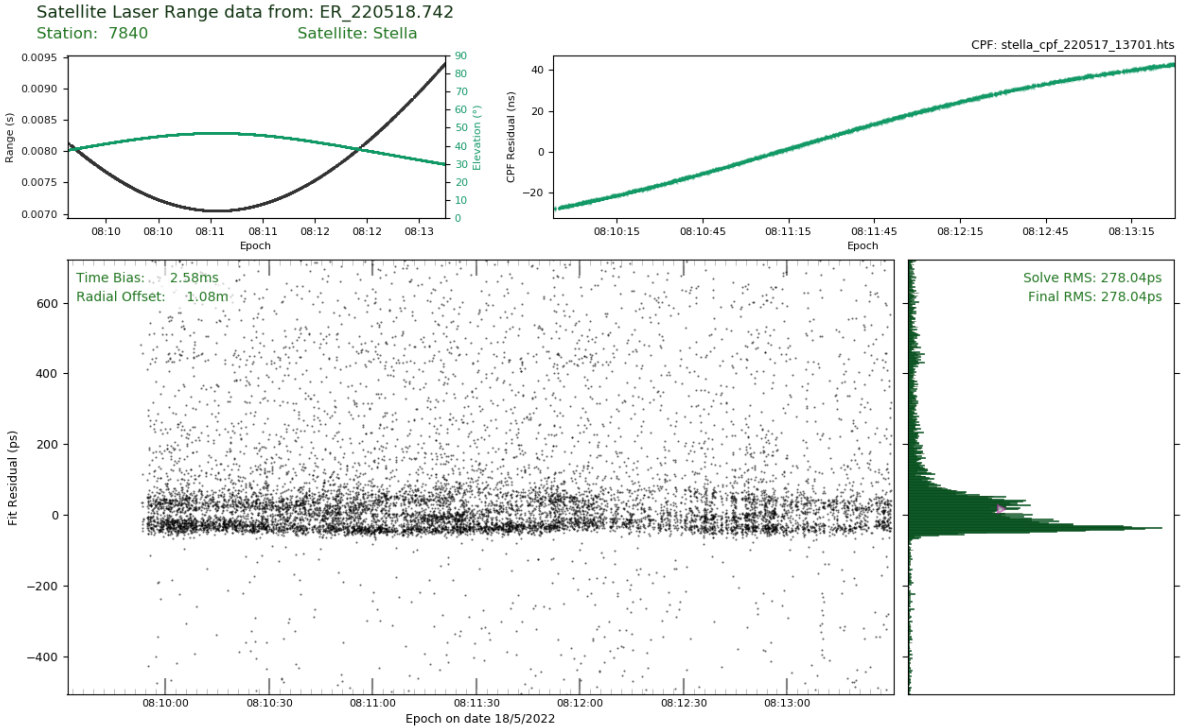
OrbitNP.py is available to download through the ILRS website.



ORBITAL RESIDUALS

The time bias and radial corrections to the orbit are calculated from the small track dataset.

These values can be applied to the whole pass dataset to produce flattened SLR residuals.

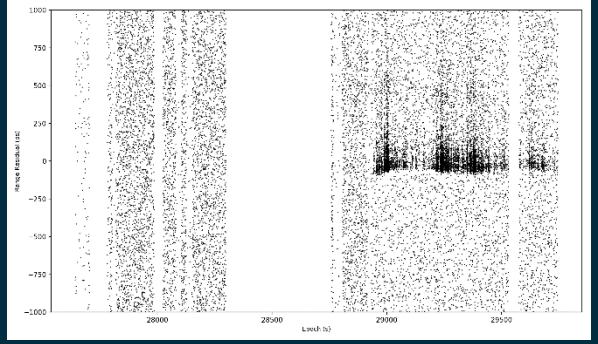


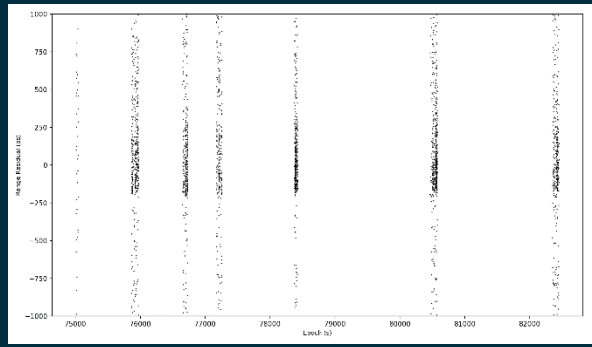
Using ranges from real-time track detection and the orbit corrections from orbitNP.py, flat residuals are automatically produced.

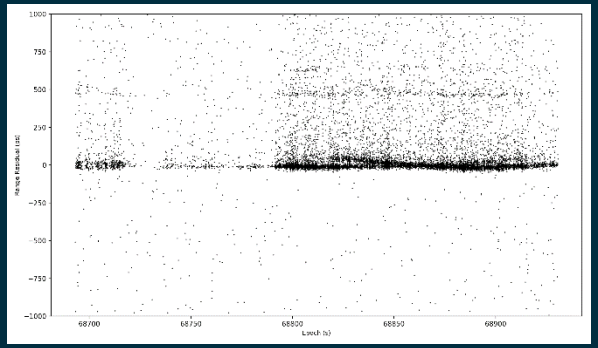
TRACK SELECTION

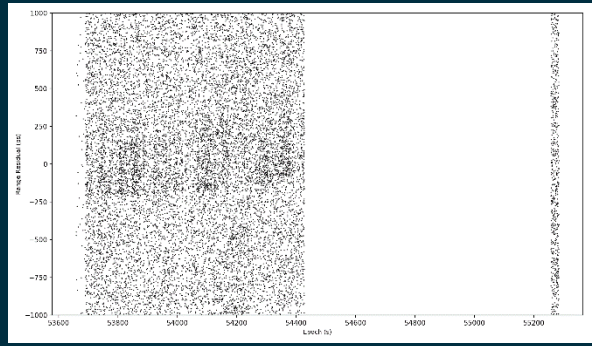
The residuals are automatically flattened. Here are some example passes.

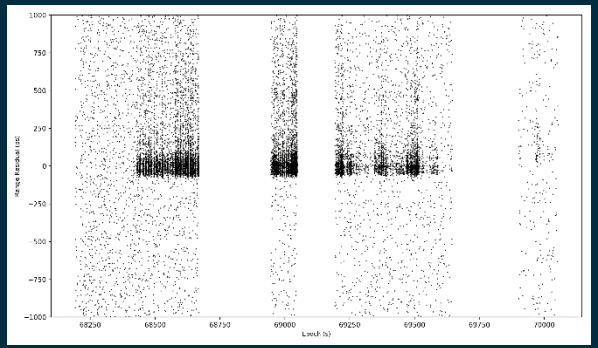
Now to make some decisions as to what is and what is not track.

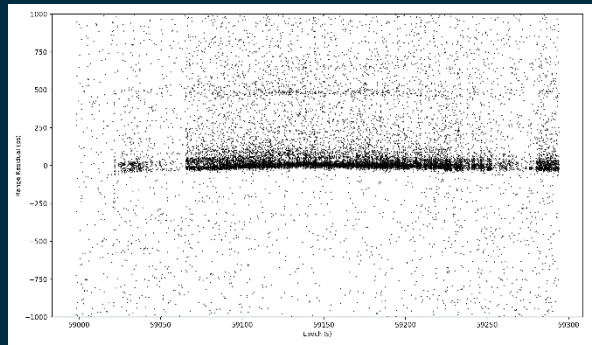








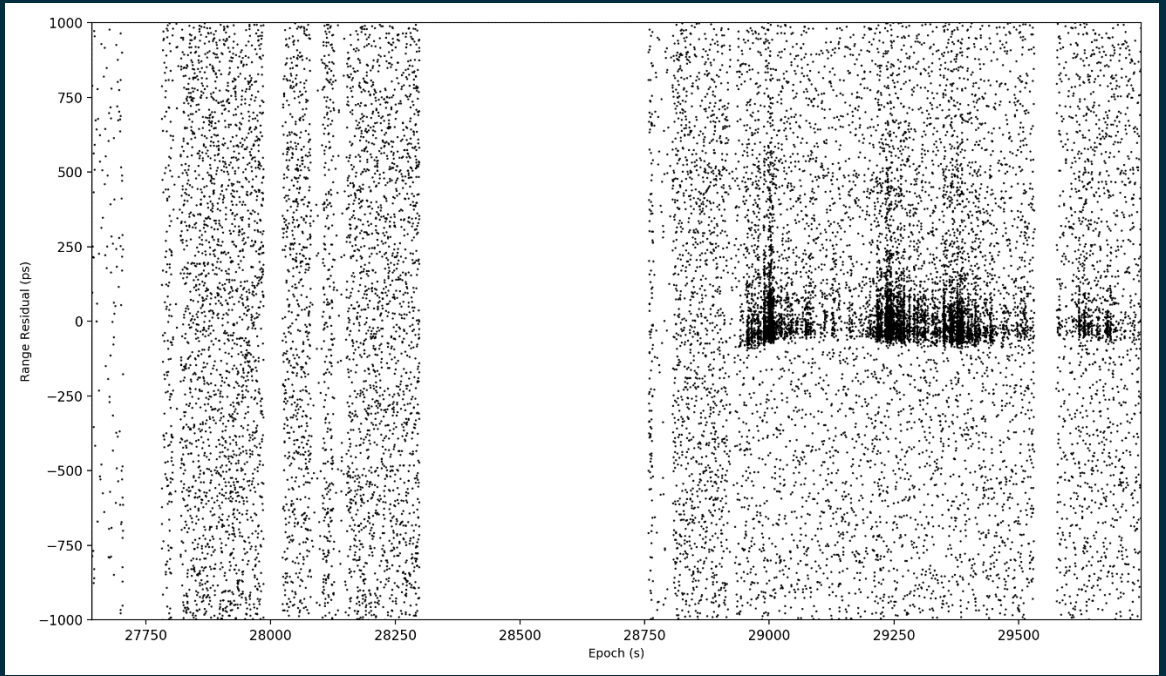




TRACK SELECTION

In this example, there are clearly areas of the residual plot that contain SLR returns. And there are areas that do not.

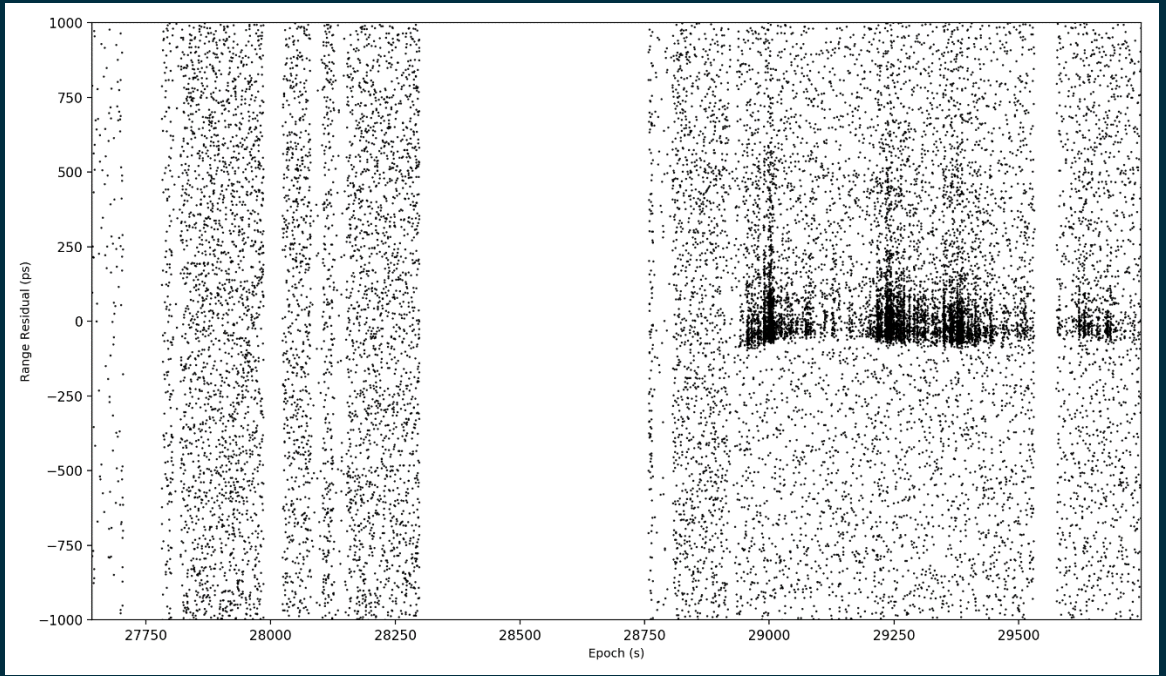
Track can be identified in a residual plot by considering relative densities.



TRACK SELECTION - METHOD

For each point record the interval to the Nth closest point within a narrow range residual window.

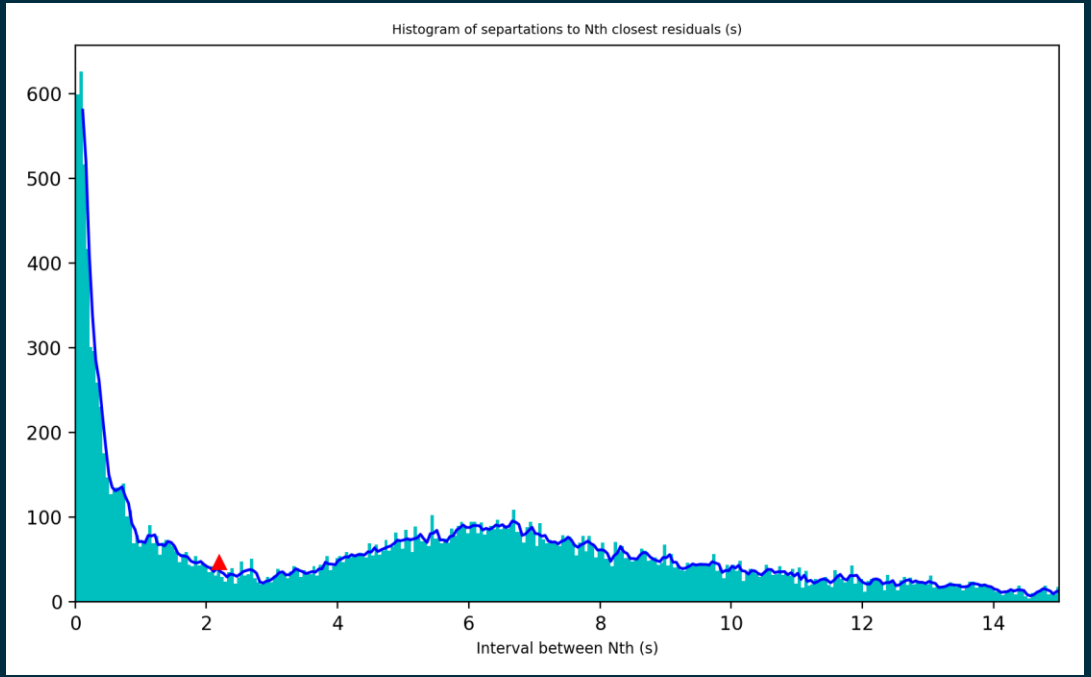
This will be very short for satellite track and longer for random noise.



TRACK SELECTION - METHOD

From a histogram of these values, the track points are the shortest values on the left of the plot.

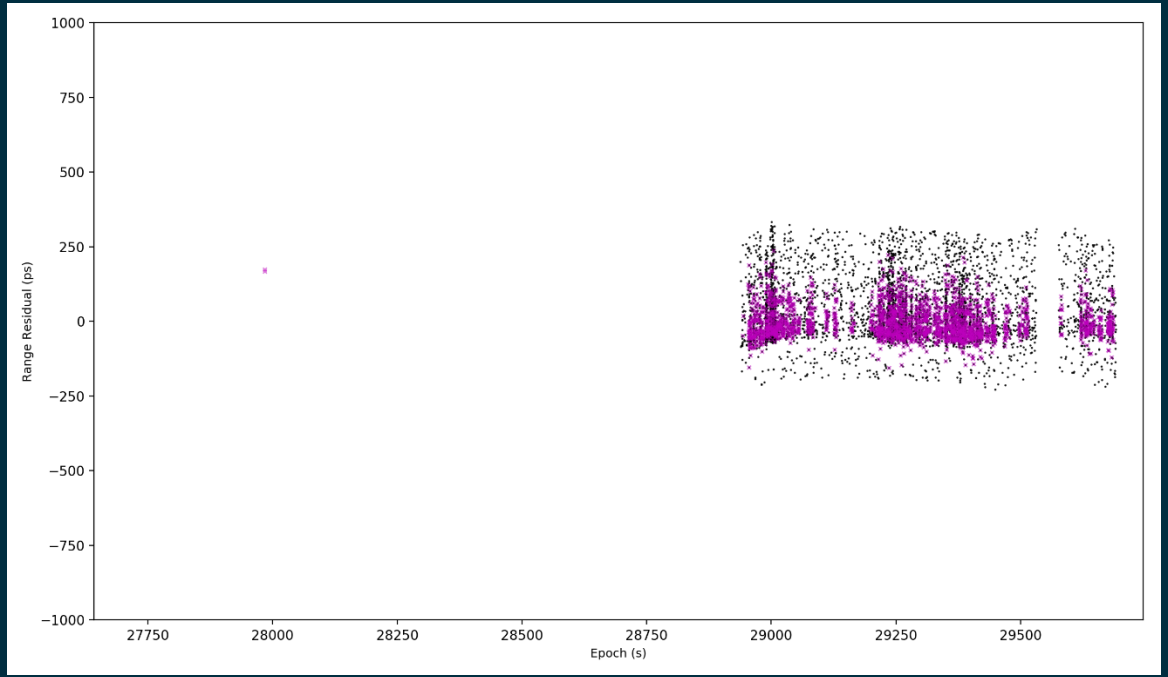
A cut-off point on the histogram needs to be defined. Below this point are satellite track returns.



TRACK SELECTION - METHOD

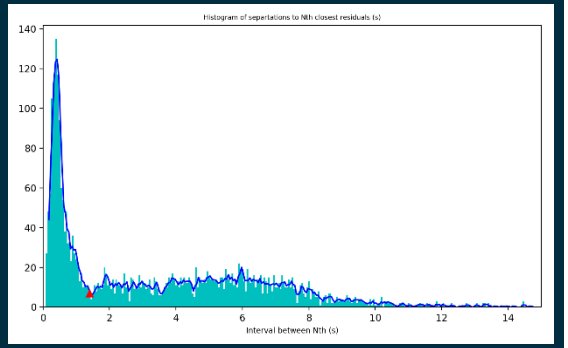
Selecting only the points below the cut-off point from the histogram gives the pink points.

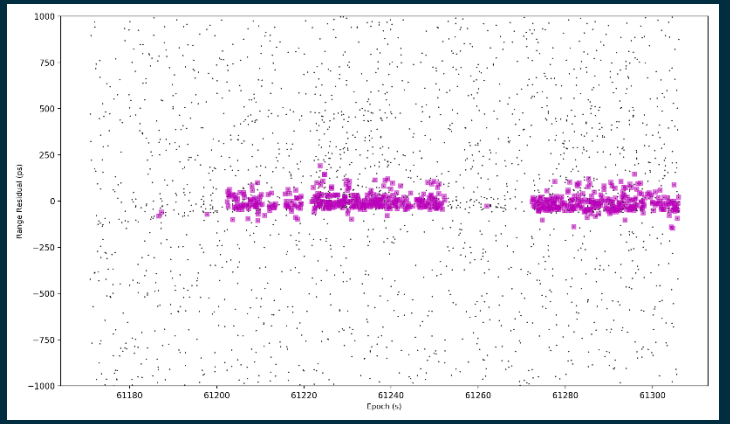
Selecting around these track points allows for the areas containing no track to be discarded.

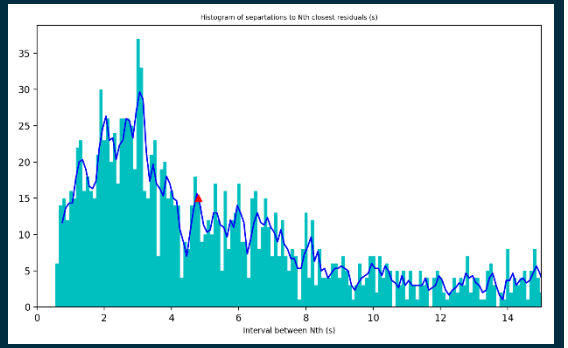


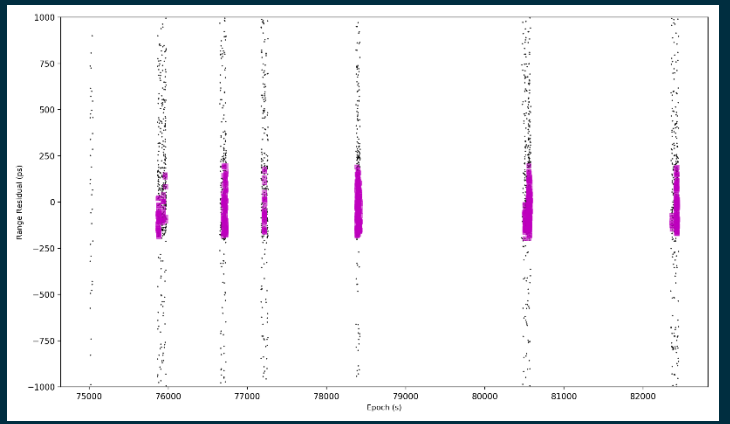
TRACK SELECTION

Other examples



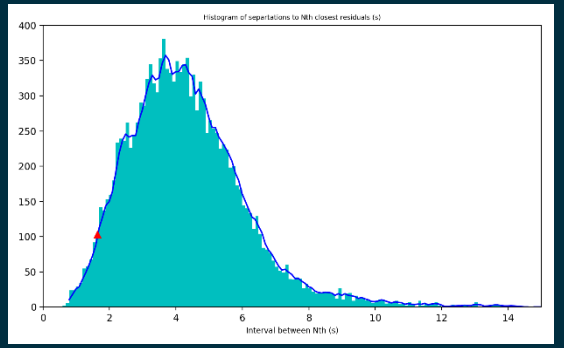


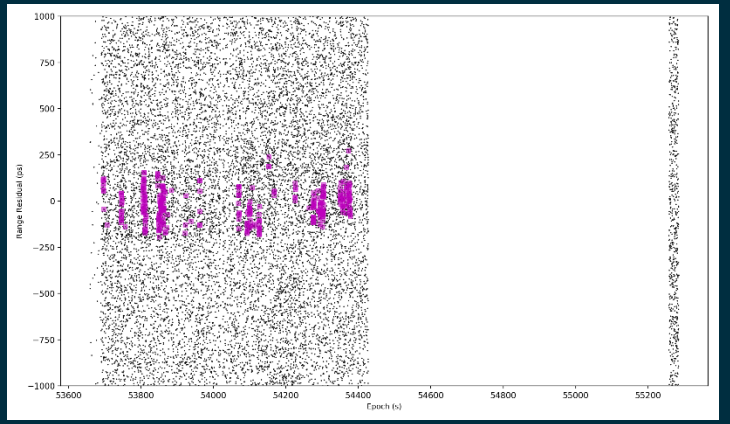


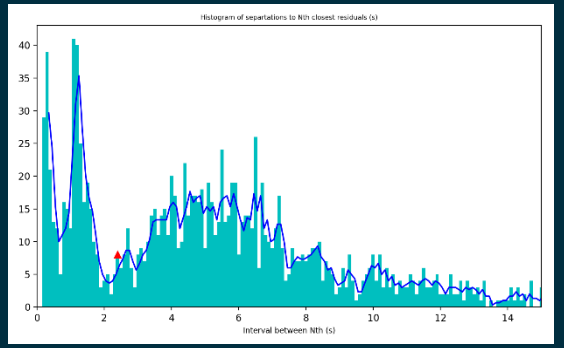


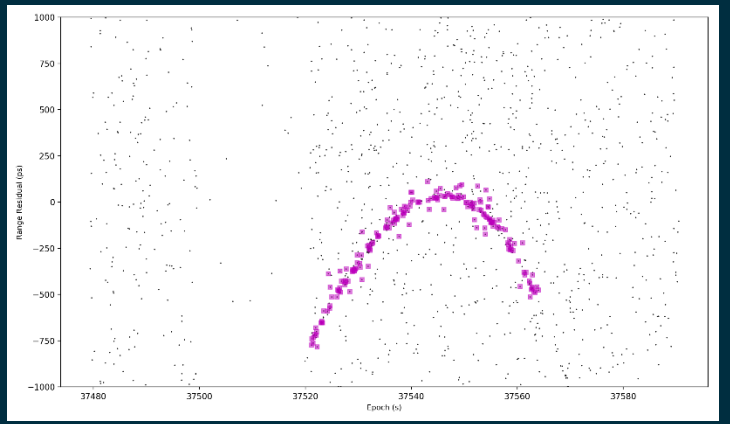
TRACK SELECTION

Other examples









RETURN RATE FILTERING

Traditionally, the SGF calculates return rate by binning data and dividing the track points over the number of shots.

Alternatively, at the 20th International Workshop on Laser Ranging in Potsdam, José Rodriguez presented a filter method based on Poisson statistics, which considered the time intervals between consecutive points.

https://cddis.nasa.gov/lw20/docs/2016/papers/45-Poisson_paper.pdf

RETURN RATE FILTERING

A new method for calculating return rates was developed that looks at the average time interval between consecutive returns.

For each return, average the time interval between consecutive points, for N returns before and after.

Inverting this value gives the number of points arriving per second and then return rate when it is divided by the number of shots.

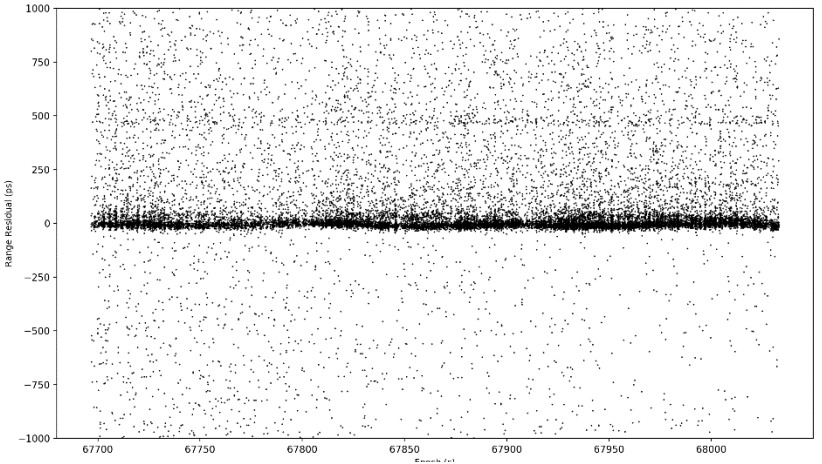
The number of shots is adjusted for any lost points recorded below the track.

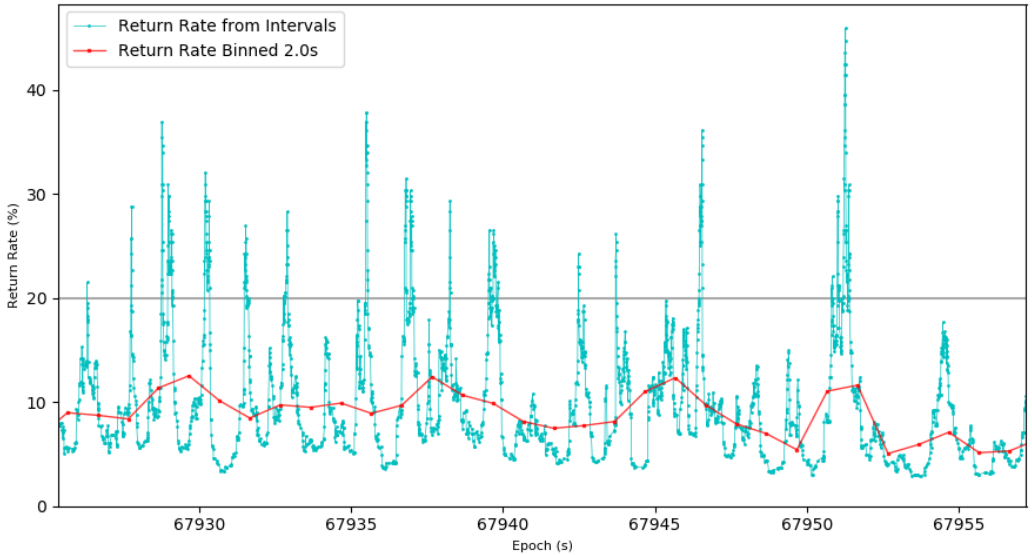
RETURN RATE FILTERING

This method has the advantage that every point is given a return rate value.

The example here shows a stable level of return rate as calculated by 2 second bin method.

The interval method show far greater variability with many short duration spikes in return rate reaching levels above the 20% threshold.



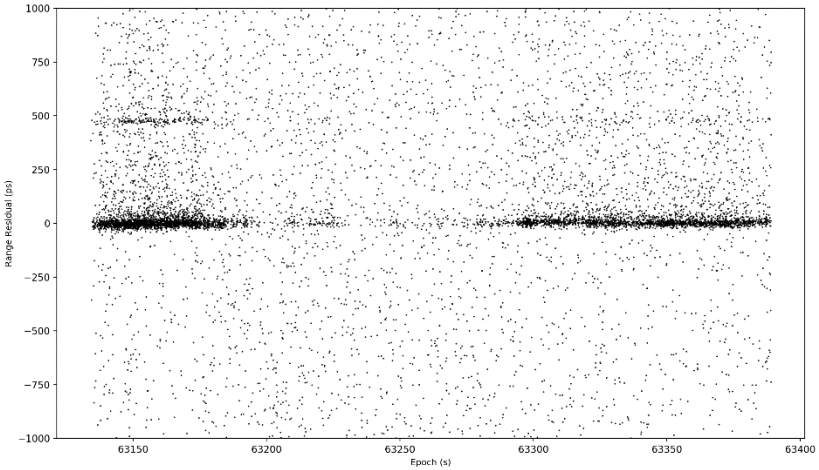


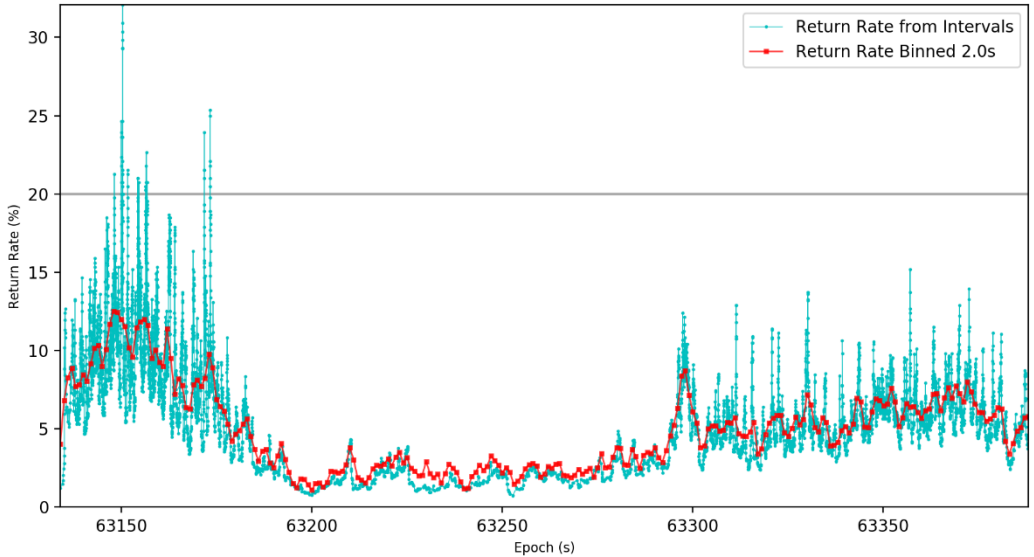
RETURN RATE FILTERING

This second example again shows agreement between the two methods.

But the interval method again shows much more return rate variability, with short term spikes.

Return rates can also be used to filter out areas of the plot that do not contain SLR track returns.





RETURN RATE FILTERING

Filtering for low levels of return rate, where there is little to no return signal is also possible.

However, a smoothed profile is preferable to allow the data to be more continuous.

Considering the background noise is useful for setting the lower threshold.

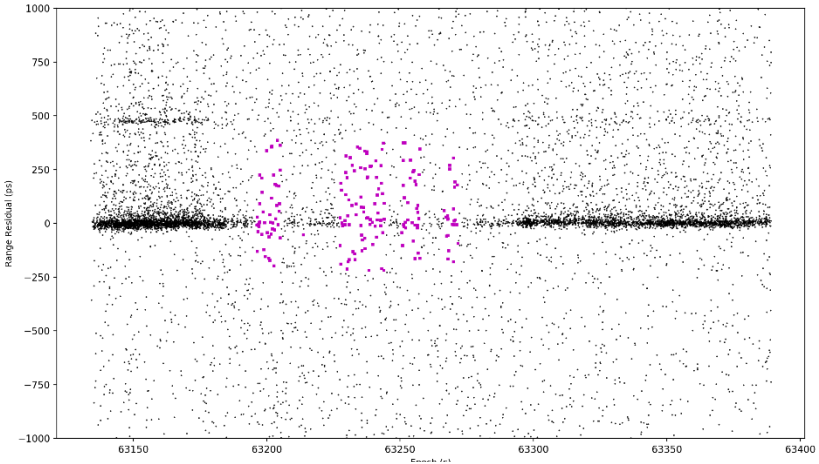
The background noise can be calculated by inverting the average residual difference from a point above the track. From this a return rate threshold can be set.

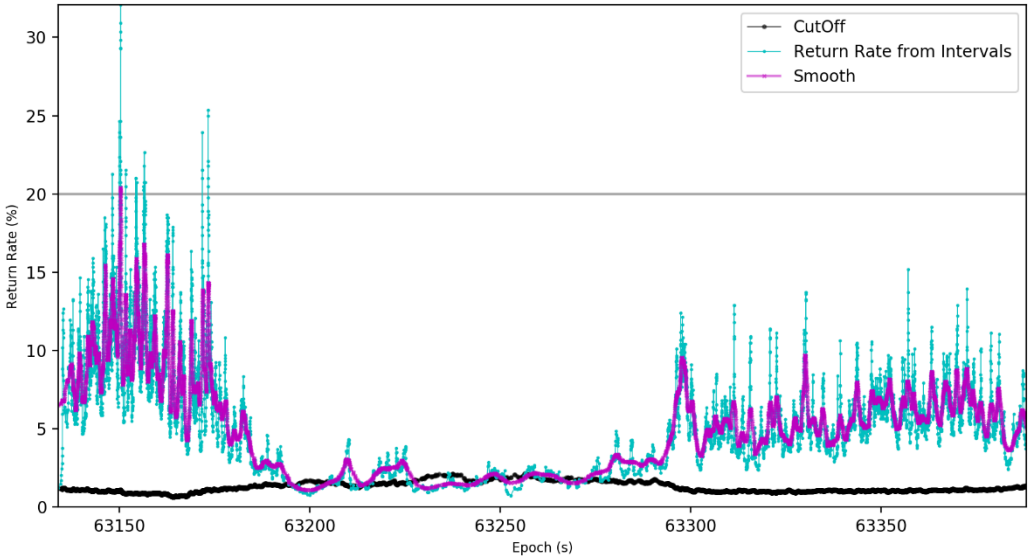
RETURN RATE FILTERING

The threshold from the background noise is plotted here in black.

The smoothed return rate is plotted in pink.

Where the return rate drops below the threshold, the corresponding points can be removed.



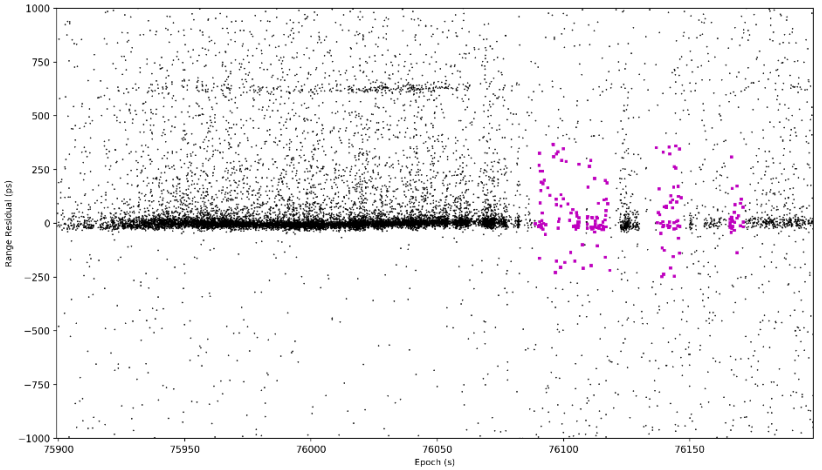


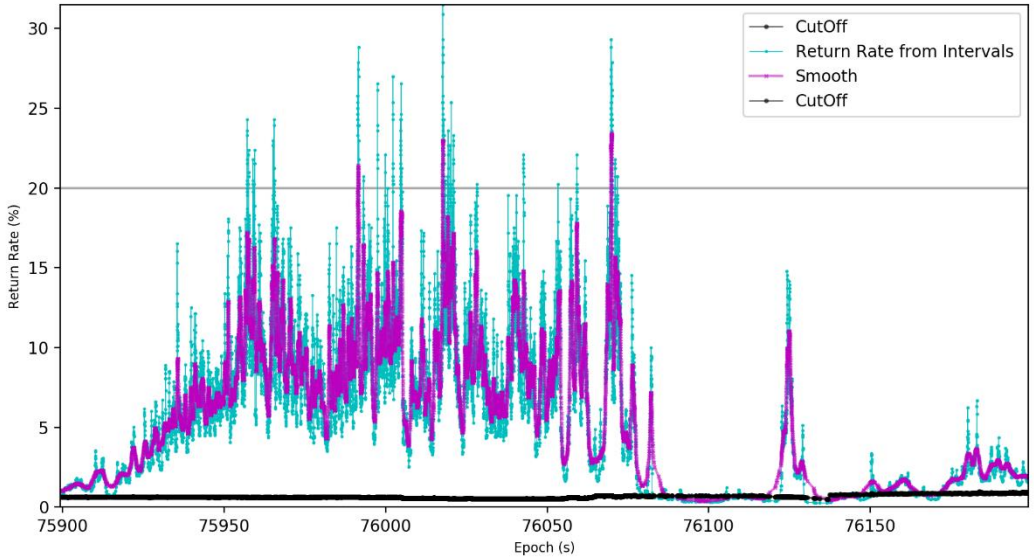
RETURN RATE FILTERING

Here is another example.

Threshold from the background noise is plotted here in black.

The smoothed return rate is plotted in pink.



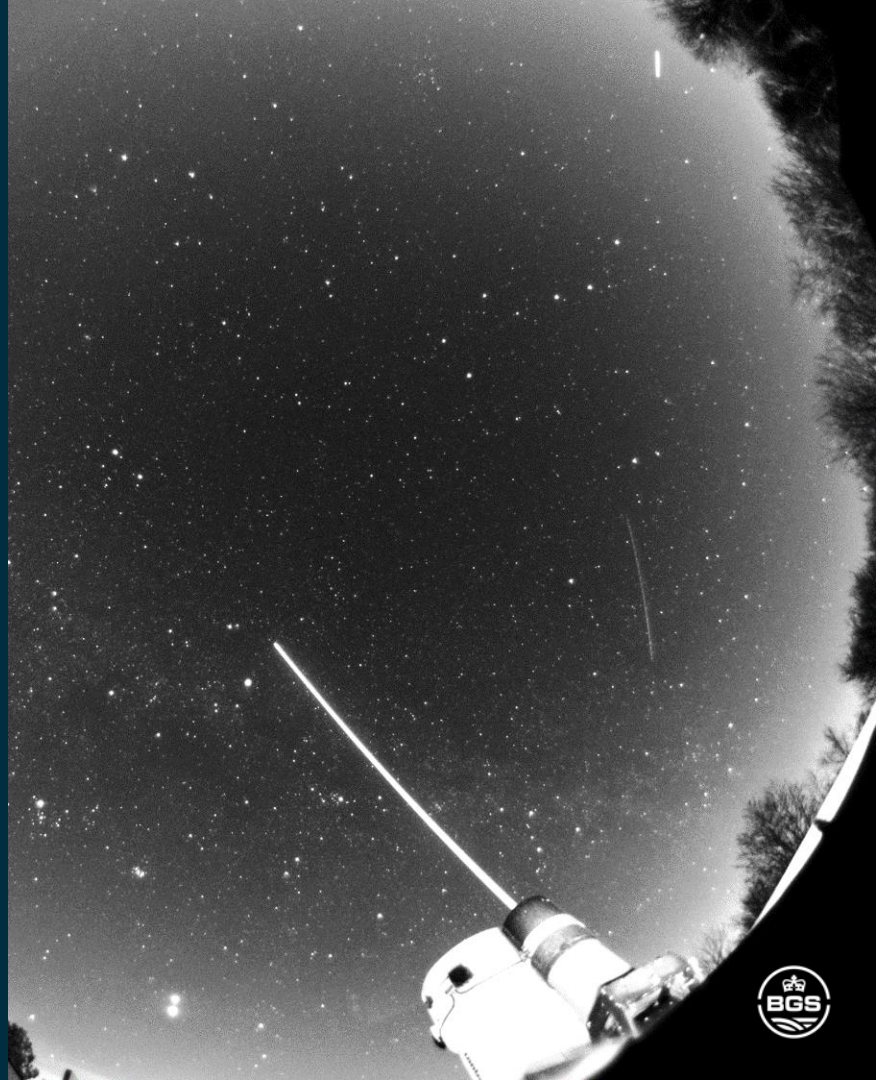


MULTI-PROCESSING TASK MANAGER

Before the reduction process can take place the required files need to generated and made available.

This is achieved using a multi-processing Python program. Routines are repeatedly run to check if new files exist and then to complete a task.

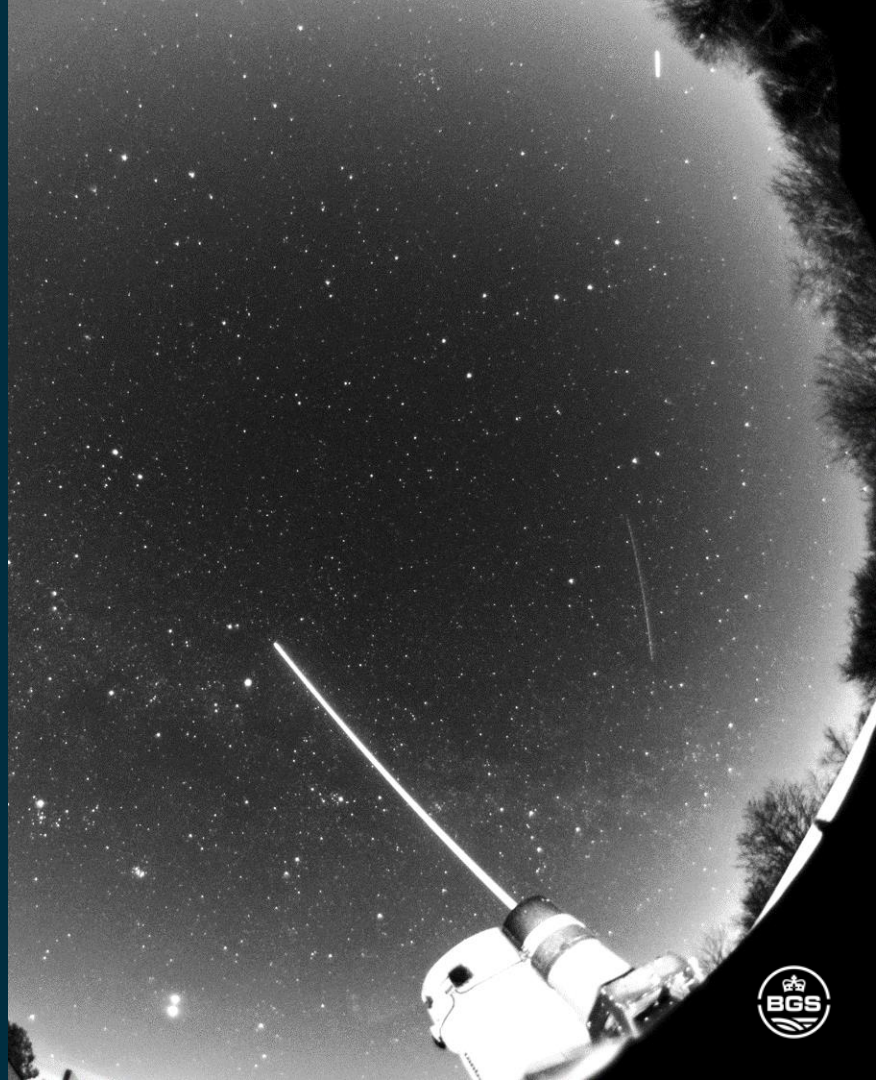
This program run continuously and prints to the screen as it executes its tasks.



MULTI-PROCESSING TASK MANAGER

A multi-processing Python program runs commands to:

- Write meteorological pass files
- Reduce new calibration data
- Write calibration pass files
- Collect status and track files
- Download CPFs
- Update maser corrections
- Reduce satellite files
- Archive the data and results



Conclusions

- Future Work

CONCLUSIONS

- Automatic reduction of passes is achievable for the majority and possibly for all SLR passes.
- Orbit correction is a stable way to produce flattened residuals.
- High return rate filtering should be carried out on short time scales to remove the spikes caused by atmospheric variability.

Future work

- Following some further testing. The process will start to routinely generate normal points.
- A close comparison will be made between the normal points from both event timers.

British Geological Survey

THANK YOU

Any questions?