Where — High Precision Positioning using Python

Michael Dähnn, Ingrid Fausk, Geir Arne Hjelle, Ann-Silje Kirkvik, Eirik Mysen
Norwegian Mapping Authority, Geodetic Institute

A combination of data from quasars (VLBI), lasers (SLR) and satellites (GPS and DORIS) is used to develop a coordinate system for the Earth, which is necessary to monitor the changes of the Earth's surface: Land rising after the last ice age, sea level increases and the tectonic plates moving several centimeters every year. Analysis from several research institutes around the world is combined in order to establish this coordinate system, known as the International Terrestrial Reference Frame (ITRF). Traditionally, most of this analysis has been done using Fortran code. At Kartverket (the Norwegian Mapping Authority) we are currently developing Where, a new software written in Python that will take part in this international collaboration.

PHOTO: GETTY IMAGES

PHOTO: BJØRN-OWE HOLMBERG

VLBI

Very Long Baseline Interferometry (VLBI) is a technique where several huge telescopes collects signals from radio sources in space, e.g. quasars. By comparing the arrival times of the signal at different telescopes we can calculate the distance between the telescopes.

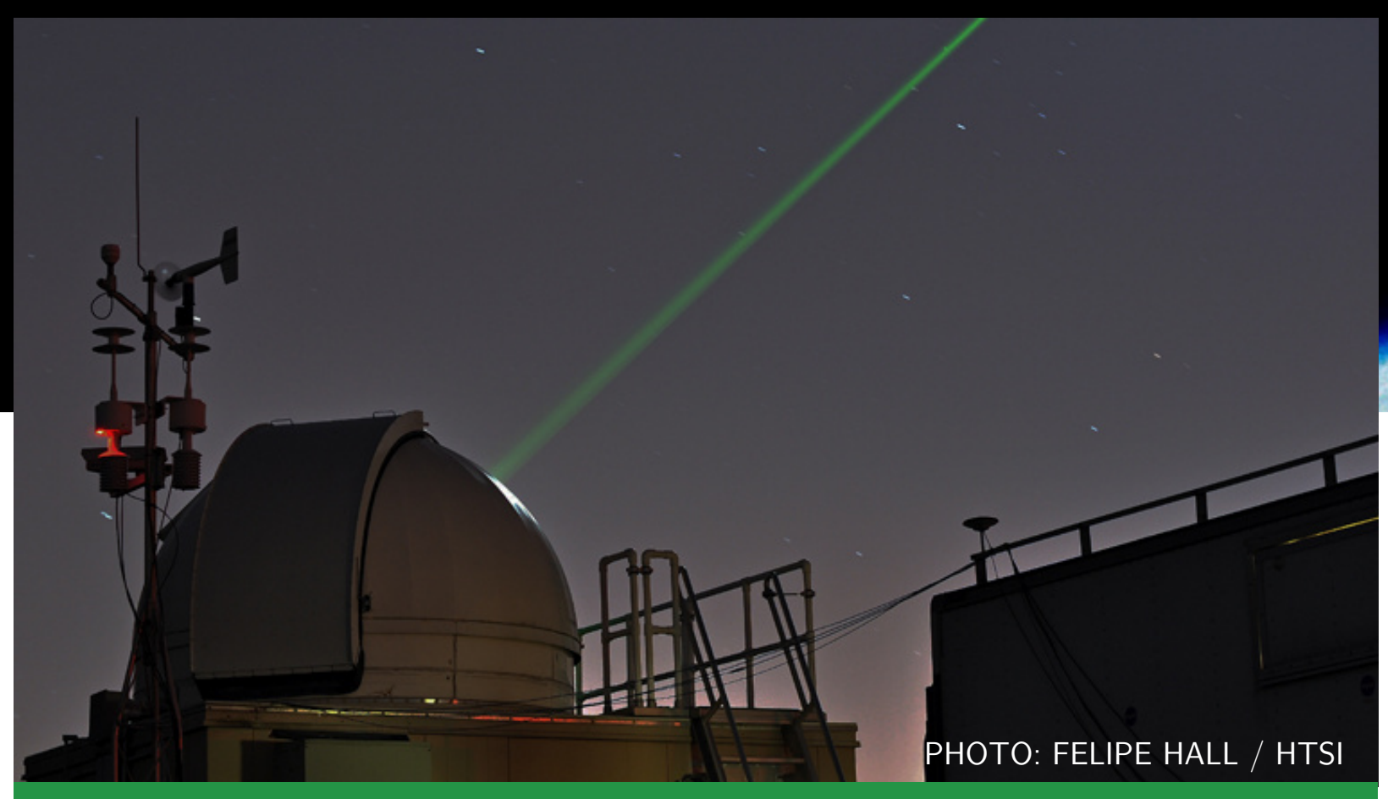


PHOTO: FELIPE HALL / HTSI

SLR

Satellite Laser Ranging (SLR) works by shooting at satellites with laser and measuring the time it takes for the laser to be reflected off a satellite and return to the laser station. These time measurements are used to calculate the distance from the laser station station to the satellite. Doing this over time gives very accurate satellite orbits.

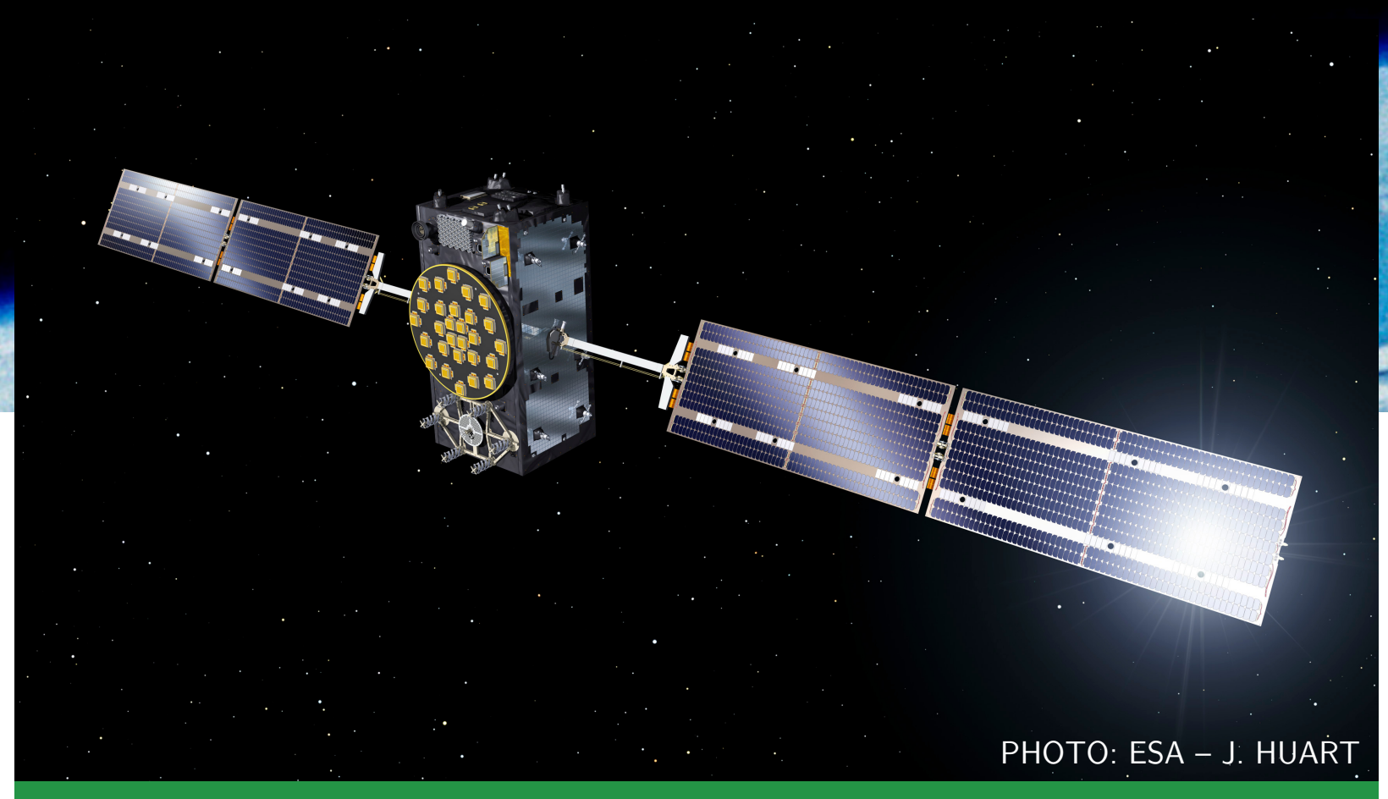


PHOTO: ESA – J. HUART

GNSS

Global Navigation Satellite Systems (GNSS) are networks of satellites used to accurately calculate positions on Earth. Such systems include the American GPS, the Russian GLONASS and the European Galileo-system. They all work by measuring the time a signal takes from the satellites to a receiver and calculating distances based on these.

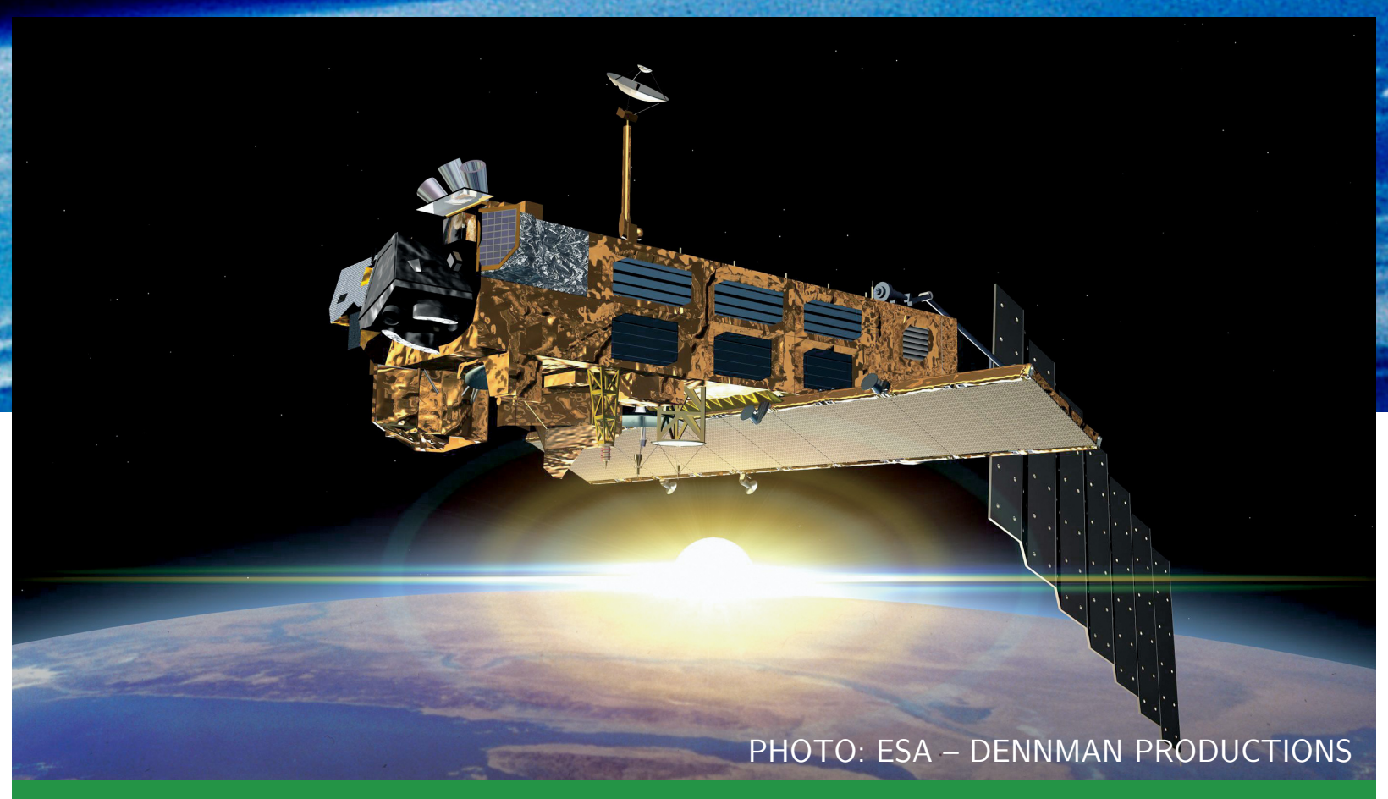


PHOTO: ESA – DENNMAN PRODUCTIONS

DORIS

Doppler Orbitography and Radiopositioning Integrated by Satellite (DORIS) is a system consisting of a network of ground-based beacons emitting signals that are picked up by satellites as they fly by. Satellite orbits and beacon positions can be calculated based on the Doppler effect, the frequency shift of the beacon signal caused by the satellite movement.

Where

Where is a software currently under development at the Norwegian Mapping Authority. Where will be able to process and analyse data from the space geodetic techniques VLBI, SLR, GNSS and DORIS, and combine these together to provide very precise positioning of geodetic stations around the world.

The International Terrestrial Reference Frame

The International Terrestrial Reference Frame (ITRF) is a coordinate system for the Earth, which is necessary for climate monitoring where we want to track minuscle movements over long time periods and for tracking satellites as they orbit the Earth. Every few years the ITRF is updated through a collective effort of several international research institutions. One of the goals of Where is to be a tool that can support the further development of the ITRF.

Python

As in many other research fields, Fortran is still heavily used in Geodesy. When we started work on Where we opted for using Python as our main programming language instead. Our motivations mainly came from Python's flexibility and ease of use, combined with the rich ecosystem that exists for doing data analysis as well as interfacing with established C and Fortran libraries.

Where is using the core of the Python datastack, including `numpy`, `scipy`, `pandas` and `matplotlib`, for a lot of functionality. In addition, we rely on more specialized packages like `astropy` and `jplephem` where special handling of geodetic data is needed. Furthermore, we are custom building important parts of the architecture, including a powerful caching mechanism and a flexible datastructure for handling the data.

The datastructure

We need an effective, flexible and powerful datastructure to handle the big amounts of data related to one model analysis, and have chosen to implement something we call a `where.dataset`. The `where.dataset` organizes data in rows each representing one observation and fields with data for that observation. Fields can be of different datatypes, where some are simply based on existing types like strings, `numpy`-arrays or `astropy`-times. Other fields are more specialized like a `position`-datatype that is represented by `numpy`-3-vectors, but knows how to transform themselves to other coordinate systems or calculate things like `azimuth` and `elevation` to other `positions`. The `where.dataset` is stored to disk using HDF5 (`h5py`) and JSON. Accessing fields can be done both using dict-notation: `dset['field']` and property-notation: `dset.field`.

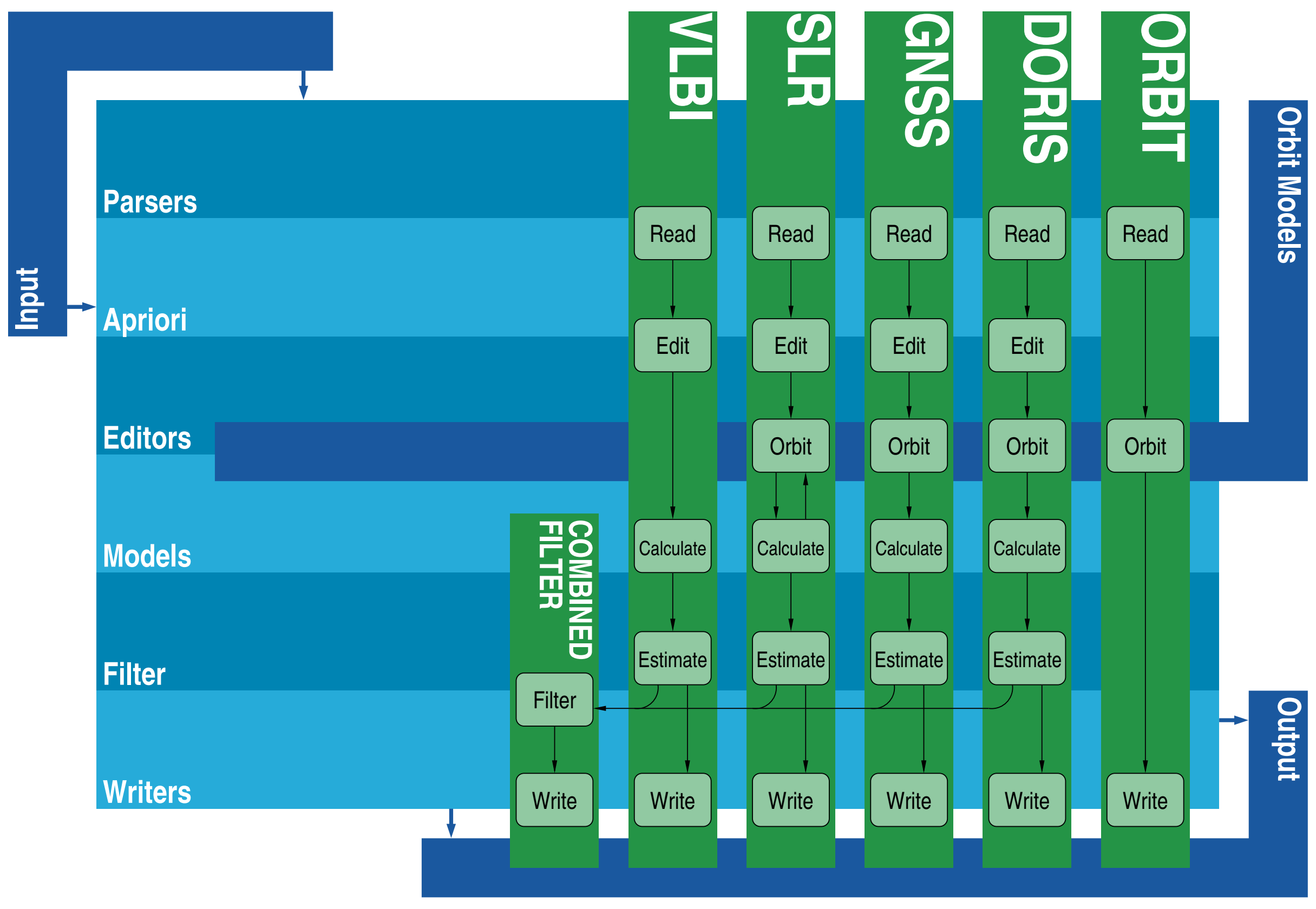


Figure: The software architecture.

Kartverket

Examples

Let us look at a few examples on how to run the Where software. One of our guiding principles is that the software should be as easy to use as possible.

A VLBI analysis

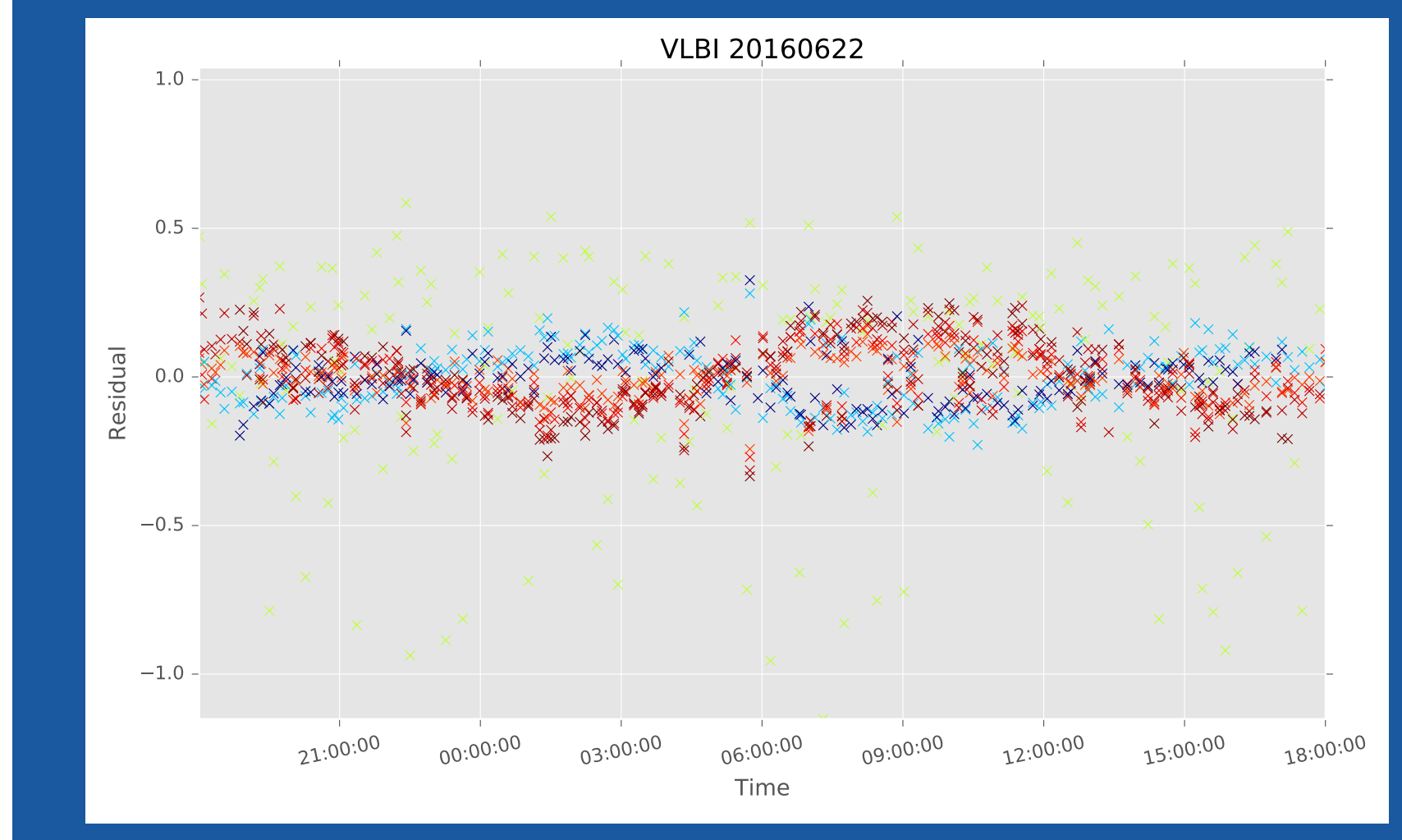
To start a VLBI analysis, we simply need to provide the date we want to analyse:

```
> where 2016 6 22 --vlbi
```

Doing this will show the default configuration for VLBI analyses, including options like which models to run, which ephemerides (positions for planets etc) to use and which output to create. All of these options can be changed, or we can accept the defaults and run the analysis.

As the model runs, all intermediary calculations are stored, allowing us to later look at data at each step. For instance can we compare raw observation numbers with calculated results, or look at the residuals before or after estimation. The simplest way to do this is to use the There GUI-companion:

```
> there
```



Residuals based on models (no estimation) from a VLBI analysis for the station at Ny-Ålesund, Norway. Colors indicate the other station in the baseline.

After a model run, we can also explore the data interactively. In practice, this means that we get access to the datasets and can use the Python datastack to manipulate them:

```
> where 2016 6 22 --interactive

Available datasets:
    v0, vlbi_calculate_XA_1
    v1, vlbi_calculate_XA_2
    v2, vlbi_edit_data_XA_0
    [...]

In [1]:  v1.residual
array([ 0.47276345, -0.05613388,
       -0.50012711, ...,  0.05031952,
       -0.00214976,  0.04906757])

In [2]: v1.unique(’station’)
[’BADARY’, ’MEDICINA’, ’NOTO’,
 ’NYALES20’, ’SVETLOE’, ’WETTZELL’,
 ’YEBES40M’, ’ZELENCHK’]
```

The final output of the analysis can be stored in many different formats. Sinex is typically used when sharing analysis with others. However, one interesting output format Where supports is Jupyter Notebooks. These allow us to for instance run each model independently, and easily create live demonstrations of the software highlighting particular parts of the analysis.

A GPS analysis

Doing a GPS analysis is done in the same way as the VLBI analysis. We only need to specify a different technique:

```
> where 2016 6 22 --gnss
```

The technique is specified as `--gnss` because Where will support analyses of several of the systems, including GLONASS and Galileo, in addition to GPS. Running Where like this will again bring up all the default options. For GNSS these options will now include how to handle satellite orbits and which stations to analyse.

Future

At the moment the software has implemented all models needed for VLBI and SLR analysis, as well as some of the GNSS models. We are currently working on the estimation and postprocessing needed in order to get the final useful results from the software.

The focus this fall will be on finishing the VLBI analysis and a simplified GNSS analysis. This includes much testing including comparing our results to that of other softwares. We will come back to the SLR and DORIS techniques later.