

RootVis telemetry analysis framework

Luisa Faltenbacher* Tobias Göttfert* Armin Braun† Boris Grishechkin*

Arvind Kumar Balan‡

Deutsches Zentrum für Luft- und Raumfahrt e. V., German Aerospace Center

Münchener Straße 20, 82234 Weßling, Germany

A commonality of all space missions is the need to receive, process, archive, and analyze on-board telemetry of the spacecrafts involved. For long-running missions, the amount of data that needs to be preserved can reach hundreds of gigabytes. At the German Space Operations Center (GSOC), the RootVis framework is under development; it shall allow to process the full telemetry dataset of the GSOC satellite missions for analysis of the long-term behavior of the spacecraft. Typically each mission has slightly different concepts of analysis, visualization and format of its telemetry, so RootVis was conceptualized as a modular telemetry visualization tool. It is built around the core telemetry archive in the file format of the ROOT data analysis library, which—thanks to its efficient serialization—enables high performance data access. Thus, it is possible to handle large data sets with billions of data points in short time for all current and future GSOC missions.

I. Introduction

Typical satellite missions produce telemetry data archives during their operational lifetime whose size is not easily handled by traditional data visualisation tools. For example, a mission older than five years can have a size of its telemetry data archive of more than a hundred gigabytes. For these five years, a single telemetry parameter measured at a sampling rate of one per second already has more than one hundred million data points, which a telemetry analysis tool must be able to process. Since previously telemetry visualization and analysis tools used at the GSOC were mainly designed to support the missions in the launch and early orbit phase, visualizing telemetry for the whole lifespan of the mission was not easily possible for the subsystem engineers. Thus it became necessary to get a telemetry visualization and analysis tool that handles this amount of telemetry and is able to support all present and future missions at GSOC. Therefore the RootVis project was born in the year 2010.

RootVis is designed to facilitate analysis of long time spans and additional processing before displaying the data, but does not replace tools for interactive exploration of short-term plots. Currently RootVis supports telemetry visualization for the TerraSAR-X and TanDEM-X satellites, TET-1, the GRACE satellites and will support future GSOC missions as well.

II. Technical details

A. Libraries RootVis is based on

Since the ROOT framework¹⁻³ offers to “handle and analyze large amounts of data in a very efficient way”, it was decided to choose it as base library for RootVis. ROOT is an object oriented open source C++ library, in development at CERN research laboratory mainly by René Brun and Fons Rademakers. It is commonly used in high-energy physics to analyze datasets in the petabyte range. The ROOT specific storage format is a machine-independent compressed binary format that serializes the stored data in an efficient way. This offers the opportunity to analyze and visualize billions of data points in short time.

*Mission Planning System Engineer, Mission Operations Department, German Space Operations Center

†MCDS (Mission Control and Data Systems) Team Lead, Mission Operations Department, German Space Operations Center

‡PTS Subsystem Engineer, Mission Operations Department, German Space Operations Center

The ROOT library is programmed in C++ for one reason amongst others to have short computing times. RootVis itself is developed in the Python programming language,⁴ made possible via the PyROOT bindings to the ROOT library. PyROOT is the “Python extension module that allows the ROOT-user to interact with any ROOT class from the Python interpreter”⁵ and was developed, also at CERN by Wim Lawrijsen. Additionally, parts of RootVis also use NumPy⁶ functionality for data processing purposes in the Python domain.

B. Design

RootVis is a composition of several modules built around the RootVis telemetry archive (see Fig. 1), where all data since mission start is stored in the ROOT specific format.

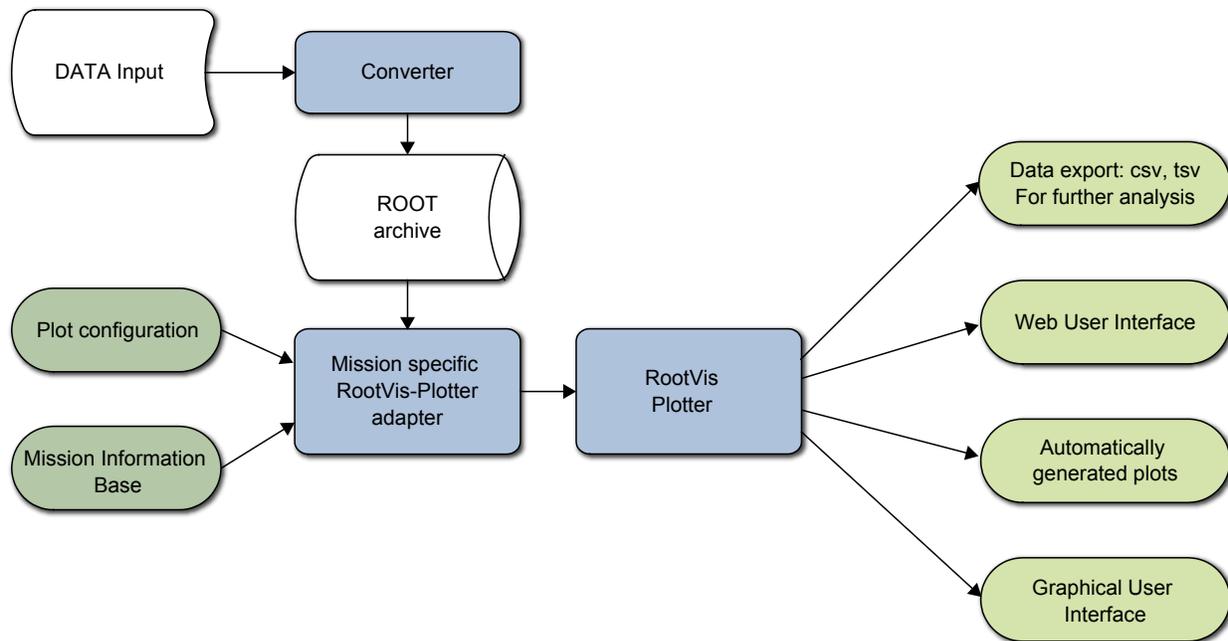


Figure 1. The diagram shows the design of RootVis. The applications are colored in blue. The archives are colored in white. All input information needed by RootVis is colored in green and the interface between the user and RootVis is colored in yellow. The directions of the arrows display the directions of the flow of information.

Once the telemetry is available it is converted from the input format into the ROOT-format. In addition to the data, some more input is needed to plot telemetry. For example the data type of the parameter values or the package name as they are defined in the telemetry/telecommand database are needed. An important input is the plot configuration, which configures the telemetry parameters displayed in the plot and additional information about processing, labeling and appearance. Here for example the type of the plot can be configured as histogram or line-plot, or a special plot with preprocessed data. Also the labeling of the axes or e.g. the position of the legend can be set there. The user specifies the plot configuration in the “yaml-format”. The whole information that is needed to visualize data points, including the data points themselves, will be prepared by several satellite specific modules and will be provided to the main plotting program that is responsible for the data visualization. The aspect that the plotting module itself does not need any information about the satellite or other satellite specific input makes it easier to add future missions later.

RootVis makes it possible for the engineer to browse through the predefined automatically generated plots via a website or a directory with image files. As required by the users the plots are generated either periodically at fixed times or automatically when new data is available. For the latter, the input data archive is monitored.

Additionally RootVis replaced the plotting back-end of an already existing graphical user web interface, to speed up the plotting process. The web interface communicates with its plotting engine via “xml-format,”

where the whole input RootVis needs is included.

Furthermore the engineer has the possibility to install the program on the local machine to interactively analyze telemetry.

III. Graphing modes

RootVis supports several options to visualize or analyze telemetry. The standard plot is a line-plot of one or more parameters over time. To process and display a plot as exemplified in Fig. 2, covering the time range of about four and a half years, takes round a minute.

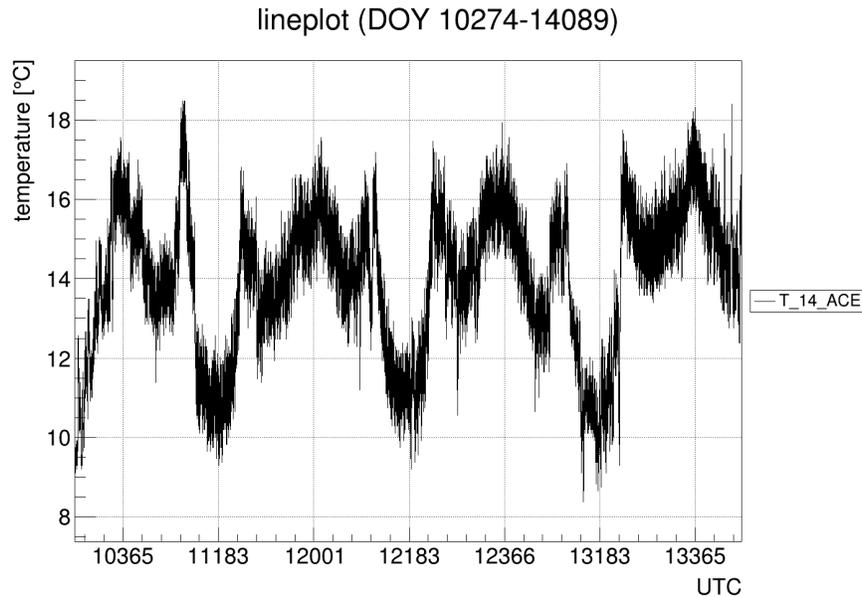


Figure 2. The image exemplifies a standard line-plot where a temperature measured on the TanDEM-X satellite is plotted over the time. On the x-axis the time is displayed in UTC, here the first two digits are the year and the following three digits are the day of year (DOY). The y-axis denotes a temperature. The plot covers a time range of about four and a half years. It takes a minute to process the nearly 9 million data points. In the title the start and end time are added automatically, the legend denotes the displayed parameter with its name.

For all plotting types the y-axis range is by default scaled meaningful on basis of the minimum and maximum values contained in the plot. Alternatively the user can define the axis-range in the plot configuration. To see parameters which are e. g. correlated to each other, covering different ranges or being described in different units, in one plot the user can define a second y-axis on the right side of the plot.

Sometimes it is requested to display the parameter limits. This information can be taken from the telemetry/telecommand database. Usually a set of parameters displayed in one plot are predicted to behave similar, and therefore mostly have the same limits. RootVis allows to define within the plot configuration for which parameter limits should be displayed, separately for hard or soft and upper or lower limits.

Additionally to the standard line-plots RootVis is able to plot one or more parameters over another parameter. Here the user must be aware that these parameters are in the same package, as defined in the telemetry/telecommand database, so that they have the same sampling rate.

If the user needs to display the distribution of the data in a defined time interval for analyzing the telemetry RootVis offers the possibility to create a histogram of the telemetry data. The user has the opportunity to decide whether the y-axis should have linear or logarithmic scale. Also the bin size can be defined by the user.

IV. Data processing

Besides visualizing the data as they are for analysis RootVis offers the possibility to conduct some analysis before plotting the telemetry.

A. Custom derived parameters

Instead of specifying which telemetry parameter to plot, RootVis also offers the possibility to define a formula involving one or more parameters. Thus, so-called derived parameters can be created on-the-fly within RootVis with a standard set of mathematical operations. As well here it is important that this calculation just contains parameters from the same package as defined in the telemetry/telecommand database, because it is expected by the implemented algorithm. Additionally for further analysis it is possible to export the plotted values in a file.

B. Bryant Plot

“Bryant^a plot” is a meanwhile well established name in the geophysical community. It is a way of plotting that can emphasize periodicities in the data. The Bryant plot combines two time axes with different scales in one plot. On one of the time axis the time is displayed as usual. On the other time axis the periodic interval is displayed; furtheron this axis will be called “secondary axis”. The values of the parameter are displayed in the third dimension.

This kind of plot is common to detect periodic events, as orbital or daily recurring events as well as seasonal effects like eclipses. For evaluating telemetry the natural periodicity of a low earth orbit satellite is that of its orbit and for a geostationary satellite it is the same as one day on earth. Therefore RootVis allows to chose between these two periods for the secondary axis.

If the RootVis user selected to display the orbital phase φ on the secondary axis he can choose between two different ways of its calculation. One way is the computation with the GPS information of the satellite; the position and the velocity. The other way is to derive the orbit phase φ from the orbit period T and a reference time t_{ref} , which denotes the time of an arbitrary ascending equator crossing point of the satellite. The orbit phase φ at a time t_p is calculated as follows:

$$\varphi = \begin{cases} \varphi' - 360^\circ, & \text{when } 180^\circ < \varphi' < 360^\circ \\ \varphi', & \text{when } 0^\circ < \varphi' \leq 180^\circ \end{cases} \quad (1)$$

$$\varphi' = ((t_p - t_{ref}) \bmod T) \frac{360^\circ}{T} \quad (2)$$

In equation 1 the orbit is assumed as a circular orbit.

The more accurate calculation of the orbit phase φ is the computation with the help of the GPS-position \vec{r} and GPS-velocity \vec{v} , which is recommended if this information is available. As initial point for this calculation a reference point \vec{s} is needed, its position vector must not be parallel to the normal vector of the orbit plane. The reference vector \vec{s} (Eq. 3) is chosen so that \vec{e}_x (Eq. 6) points from the orbit center to the most southern point of the orbit and \vec{e}_y (Eq. 5) points from the orbit center to the ascending equator crossing point of the satellite orbit. For a given position vector \vec{r} and velocity \vec{v} the orbit phase can be calculated to a reference point via (Eq. 7):

$$\vec{s} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \quad (3)$$

$$\vec{e}_z = \frac{\vec{r} \times \vec{v}}{\|\vec{r} \times \vec{v}\|} \quad (4)$$

$$\vec{e}_y = \frac{\vec{e}_z \times \vec{s}}{\|\vec{e}_z \times \vec{s}\|} \quad (5)$$

$$\vec{e}_x = \frac{\vec{e}_y \times \vec{e}_z}{\|\vec{e}_y \times \vec{e}_z\|} = \vec{e}_y \times \vec{e}_z \quad (6)$$

$$\varphi = \text{atan2}(\langle \vec{r}, \vec{e}_x \rangle, \langle \vec{r}, \vec{e}_y \rangle) \quad (7)$$

^anamed after Dr. Duncan Bryant of Rutherford Appleton Laboratory

In RootVis the third dimension is realized with a rainbow color map, where the dark blue range represents the lower values and the dark red range represents the higher values. Under- and overshoots are separately color-coded in black and pink, respectively.

Figure 3 displays an example to explain this plot type in more detail. In this plot, the orbital phase, was chosen for the secondary axis. For demonstration the z-position in GPS coordinates of the TET-1 satellite is displayed in the color-coded third dimension. This plot should demonstrate the correlation between the orbit phase and the z-position, as it shows that the z-position value reaches its maximum when the satellite is in the northernmost point in orbit and vice versa.

As the GPS-position and GPS-velocity in most cases are not in the same package as the desired telemetry parameter it is necessary to interpolate between the data points to map the orbit phase and the parameter.

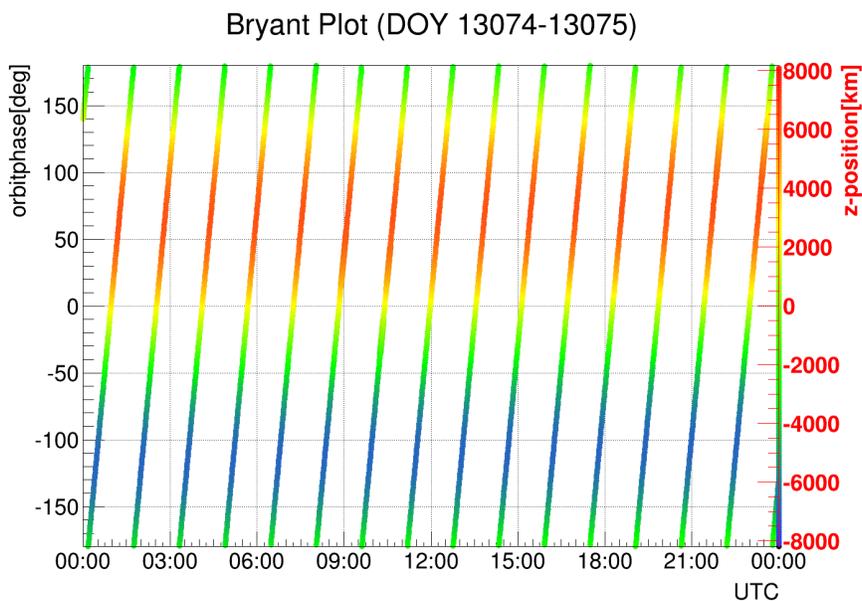


Figure 3. On the x-axis the time is displayed in UTC in hours and covers one day. On the y-axis the orbit phase which is correlated to the orbit time is displayed in degrees. The 0° point is defined as the ascending equator crossing point. On the third color-mapped dimension the GPS-position in direction of the pole-axis is shown.

Figure 4 shows color coded the rotation rate of the TET-1 satellite around the axis in flight direction in the satellite fixed coordinate system with a clear orbit periodicity over one month. The plot exhibits the regularity of the rotation rates' behavior, specific for the current mode,⁷ as well as the mode changes. In the first part of the time the plot covers can be seen that the satellite was in a special mode where the satellite was rotated on the north and the south-pole, what is represented with the two horizontal blue strips. That the mode changed in the last part show the gaps in the blue strips.

C. Data reduction

Since RootVis is designed for analysis of long-term telemetry, reducing the data before displaying it can improve readability of the resulting plot. A common way to reduce the data is to calculate mean values (P_j, T_j) (Eq. (8), Eq. (9)) over a defined time interval I , as exemplified in figure 5, in order to smooth out short-term behavior and see its long-term trend.

However, this shall not be performed at the expense of not spotting outliers in the data or washing out oscillations completely. The standard way of displaying “reduced data” in RootVis therefore is to plot minimum and maximum curves in addition to the curve of mean values. RootVis uses an “adaptive” way of averaging in order to cope with telemetry parameters that change their sampling rate, e. g. because the satellite can be switched into high- and low-rate modes. Therefore, the averaging is performed over a given time interval I rather than an absolute number of data points, which may result in non-equidistant points on the x-axis (usually the time axis). Future extensions that are planned shall handle weighting of data points depending on the sampling rate, so that many points at a higher sampling frequency do not outweigh fewer

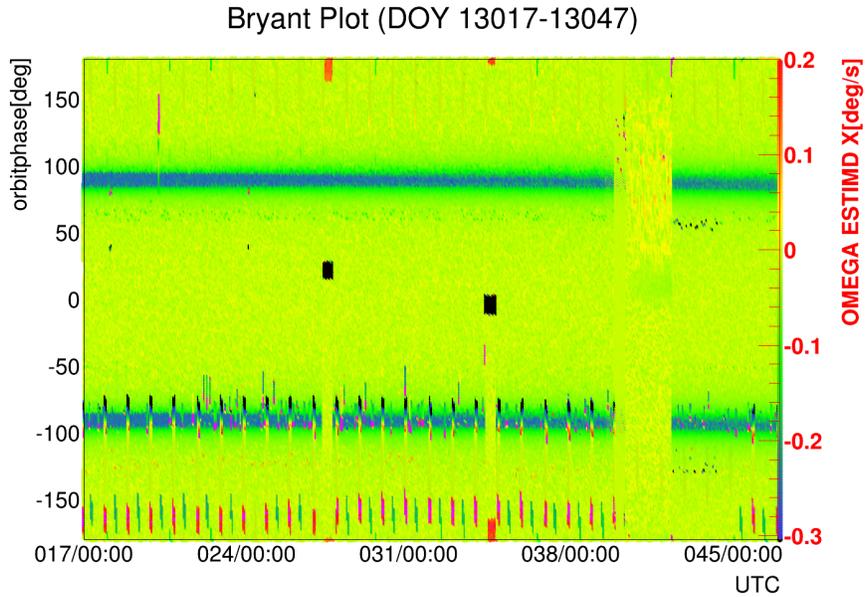


Figure 4. This plot shows the periodicity of the rotation rate of TET-1 in the satellite fixed coordinate system around the axis in flight direction over one month, which is drawn on the right axis. On the left axis is displayed the orbit phase in degrees.

points at lower sampling frequency that still represent a larger interval in time.

$$T_j = \frac{\sum_{i=1}^n t_{ji}}{n} \quad (8)$$

$$P_j = \frac{\sum_{i=1}^n p_{ji}}{n} \quad (9)$$

$$n = \text{the number of values in } I; n \in \mathbb{N}^+ \quad (10)$$

$$t_{ji} = [t_{start} + I(j-1), t_{start} + Ij[\quad (11)$$

$$\forall \{j \in \mathbb{N} \mid j \leq \frac{t_{stop} - t_{start}}{I} + 1\} \forall i \in \mathbb{N}^+ \quad (12)$$

where t_{start} is the first time value in the plot, set by the user, t_{stop} is the last time value in the plot, set by the user, I is the interval; the mean value should be calculated with the values covered by this interval and p_{ji} is measured value at the time t_{ji} .

V. Conclusion

After successfully implementing the user-requirements in the RootVis project, it now provides several useful features that help in visualizing and analyzing long-term telemetry. Additionally, the high performance data access enables the engineers to retrieve plots in short time, which also enables regular bulk production of an extensive set of automatically generated plots. It is planned to expand the forms of visualization with more different plotting types as well as the data processing parts.

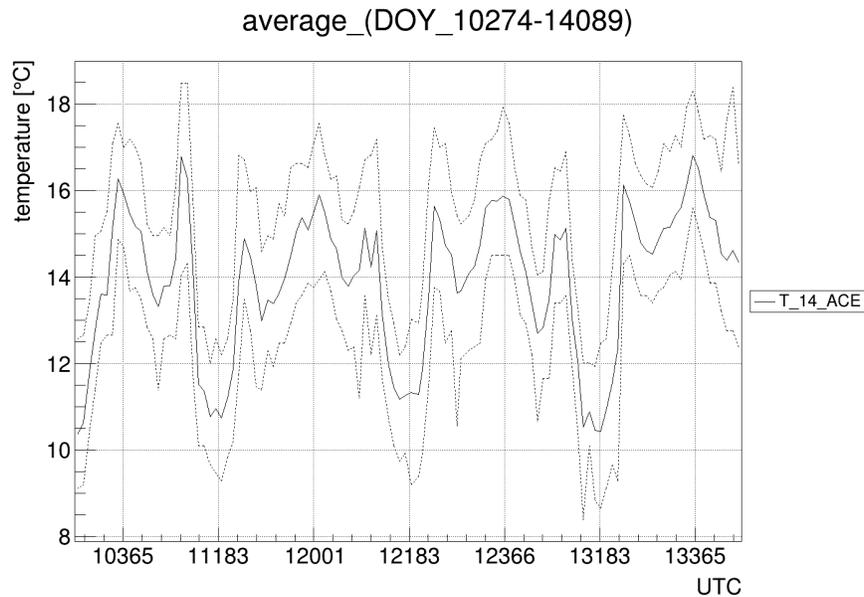


Figure 5. The parameters and the interval the plot covers are equal to those in the line-plot shown in figure 2. But instead plotted all data in this plot the mean values over 11 days with their minima and maxima are visualized. The continuous line connects the mean values. For every color of a continuous line two dotted lines in the same color exist. The upper one connects the maximum values and the lower line the minimum values of the parameter in the mean interval.

Glossary

GSOC German Space Operation Center

TerraSAR-X and TanDEM-X Earth observation radar satellites

TET-1 The “Technologieerprobungsträger” is a small German satellite for technical experiments in space, the abbreviation means Technology Experiment Carrier

GRACE Gravity Recovery and Climate Experiment

CERN European Organization for Nuclear Research, Geneva, Switzerland

References

¹Brun, R., Rademakers, F., “ROOT - An Object Orientated Data Analysis Framework”, *AIHENP*, 2 - 6 September 1996, Lausanne, Switzerland

²Antcheva, I., Ballintijn, M., Bellenot, B., Biskup, M., Brun, R., Buncic, N., Canal, Ph., Casadei, D., Couet, O., Fine, V., Franco, L., Ganis, G., Gheata, A., Gonzalez Maline, D., Goto, M., Iwaszkiewicz, J., Kreshuk, A., Marcos Segura, D., Maunder, R., Moneta, L., Naumann, A., Offermann, E., Onuchin, V., Panacek, S., Rademakers, F., Russo, P., Tadel M., “ROOT — A C++ framework for petabyte data storage, statistical analysis and visualization”, *Computer Physics Communications*, Volume 180, Issue 12, December 2009, Pages 2499–2512

³“ROOT”, URL: <http://www.root.cern.ch/> [cited 14 February 2014]

⁴“Python”, URL: <http://www.python.org/> [cited 14 February 2014]

⁵“PyROOT”, URL: <http://root.cern.ch/drupal/content/pyroot> [cited 14 February 2014]

⁶“NumPy”, URL: <http://www.numpy.org/> [cited 14 February 2014]

⁷Löw, S., Herman, J., Schulze, D., Raschke, C., “Modes and More - Finding the Right Attitude for TET-1”, *SpaceOps 2012 12th International Conference on Space Operations*, 11 - 15 June 2012, Stockholm, Sweden