PROCEEDINGS OF SCIENCE

Developments in Spacecraft Doppler tracking software (SDtracker)

Guifré Molera Calvés[a,*] **and Mars Buttfield-Addison**[b]

[a] *School of Natural Sciences, University of Tasmania College Road, Sandy Bay, Tasmania, Australia*

[b] *School of Information and Communication Technology, University of Tasmania College Road, Sandy Bay, Tasmania, Australia*

E-mail: guifre.moleracalves@utas.edu.au, m.buttfield-addison@utas.edu.au

Arising from software developed for the 2005 Huygens probe landing, the first official version of SDtracker was used for the 2008 initial detection test of the ESA's Venus Express (VEX) space mission. In the 13 years since, SDtracker has been regularly updated to accommodate different radio telescope configurations, newer VLBI data formats, different hardware architectures, and more. Recently, a major upgrade was released to address a number of performance and usability issues. This newer version has a simplified installation process, removed or updated dependencies, and does the same job in significantly fewer lines of code. Herein, the features and scientific achievements of the improved software are discussed.

15th European VLBI Network Mini-Symposium and Users' Meeting (EVN2022)
11-15 July 2022
University College Cork, Ireland

*Speaker

1. Context

1.1 The Problem

The tracking of the Huygens probe during its descent and landing on Titan in January 2005 was a demonstration of the potential of spacecraft tracking by VLBI radio telescopes. In the experiment, over 20 telescopes around the globe tracked the probe at S-band and stored the data to physical disks, which were then transferred or shipped to JIVE to perform the necessary post-processing. VLBI data was used to calculate the state vectors with ultra precision, which were then used to determine the wind profile on Titan during the probe descent [1].

The Huygens experiment proved that astronomical antennas could be effectively used for radio navigation and planetary science. But it required the development of specialised software to streamline the operation—for independently-operated, VLBI-equipped systems to work in parallel to the space agency's own data collection.

1.2 The Solution

To address this need, the first iteration of the spacecraft Doppler tracking software SDtracker[1] was written in 2009, in a collaboration between JIVE and Aalto University. It was one of two projects to derive from that original codebase, and its methods and algorithms were published in [2]. The other development became the software correlator SFXC in use at JIVE today [3].

SDtracker has since been used extensively in collaborations with the European Space Agency (ESA) for S-band and X-band planetary missions—including ESA's own missions Venus Express [4], Mars Express [5], Rosetta, and BepiColombo, and the Chinese National Space Agency's (CNSA) Tianwen [6].

1.3 The Upgrade

SDtracker 2.0 eases the installation process for new users of the software through the inclusion of automated scripts for each supported platform. The dependencies have been minimised to make the installation optimal for a multi-core 64-bit machine (e.g. standard Flexbuff), as has become the modern standard. A number of dependencies were removed—including the Intel Integrated Performance Primitives (IPP) Legacy bridge functions, as were previously required to support the latest IPP core release—and others were updated to bring them in line with long-term support (LTS) releases. This included the Python component dependencies being brought up to Python 3.6 specification, and use of the latest `mark5access` release available in DiFX 2.7.

Finally, along with general optimisations for improved performance, a significant portion of the library's code has been removed; this includes the removal of vendored dependencies and extraction of unused or vestigial sections of code, to aid in future maintenance of the library.

2. Architecture

What is now SDtracker was once two separate projects: a software spectrometer (SWspec) and an optional spacecraft tracking extension (SCtracker). By 2021, the code and user base shared

[1] https://gitlab.com/gofrito/sctracker/

between the two had increased to a point where it made sense for the two to be incorporated into a single project.

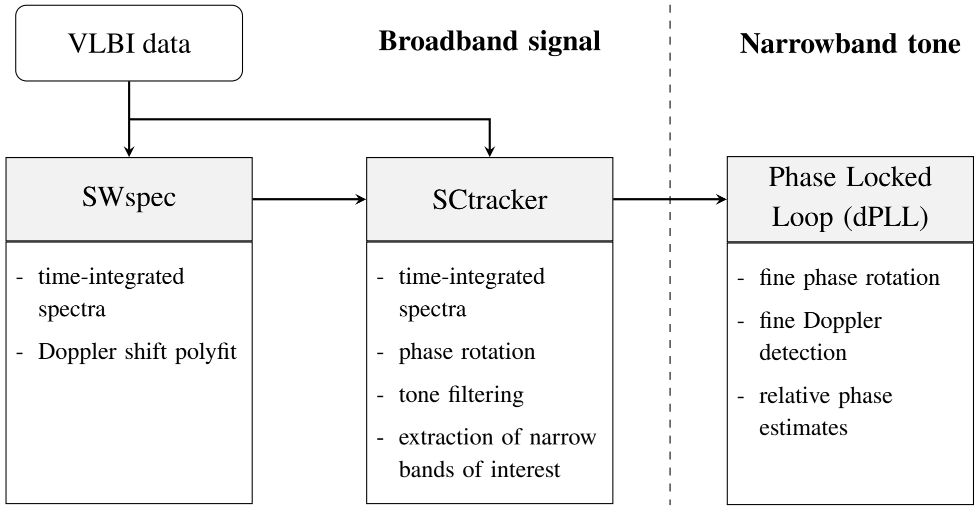

Figure 1: Process by which a broadband radio signal is processed through both SWspec and SCtracker. These two packages now form the core of SDtracker.

In an attempt to simplify ongoing maintenance, some 65,000 lines of code were removed. This was primarily done by removing layers of abstraction that had been architected for flexibility that was not needed in the final product. For example:

- The I/O functionality that could support either file or tee streams through a generic superclass for each of input and output, even though files were always used.
- The Core file and functions that existing in both IPP and MKL versions through a generic superclass, even though IPP was always required.
- The existence of multiple forms of Buffer and Phase Calibrator objects, even though only one of each were ever called for.

In each case, the generic intermediate class was replaced with the concrete implementation that was actually in use, and any alternatives discarded. Additionally, similar code between the SWspec and SCtracker components were made consistent and then centralised in a shared library that was then imported by both projects. As well as simplifying maintenance, this lead to a small improvement to build and test times.

3. Usability

The recent upgrade to SDtracker has improved its usability via:

- Automated installation scripts, replacing what was previously a series of command-line steps to install each required dependency.

- Clearer license and attribution terms, replacing the previous system of mixed and per-file licenses that made usage rights of the whole library unclear.
- Removal of legacy and platform-specific dependencies.

The software supports both Linux and OS X systems, to cover the breadth of standard Flexbuff-type data acquisition systems. This direct back end installation is necessary for the data processing to be remotely conducted on-site, to remove the need for high-volume data transfer to an external processing site.

4. Features and Applications

The software has been designed with traditional VLBI station hardware in mind. SDtracker is compatible with both legacy and modern VLBI data formats, though data processing nowadays is increasingly based on the VDIF standard alone. It also allows configuration of the back end with different bandwidths (from 2–128 MHz), and multiple frequency channels (1–16). The configuration can therefore adapt to support the end-user's intended application of the final data product.

Furthermore, the software also allows for use of non-standard astronomical hardware back ends. Development is in place to allow both complex and real multi-bit data inputs from Software Defined Radio (SDR) devices. Higher bit-sampling allows higher precision for tracking applications.

SDtracker has been used for a variety of near-field applications with radio telescopes, including:

- **Deep space planetary missions** SDtracker can be used to track any planetary spacecraft in the Solar System. The software generates precise measurements of the Doppler velocities and the residual phase of the spacecraft carrier signal.
- **Earth-bound satellites** SWspec is used to characterise the radio signals transmitted by satellites in Earth's orbit. This enables monitoring such as regular pattern of life assessment, RF characterisation and power/frequency anomaly detection.
- **Bistatic radar tracking** VLBI radio telescopes are used to receive the radio signals transmitted by ground antennas and reflected by objects of interest. This includes tracking of non-transmitting objects in near-Earth space–including asteroids or orbital satellite debris.
- **Single-dish spectroscopy** SWspec has been used in several campaigns to obtain the spectral profile of astronomical or planetary masers. The versatility of the software allows for tweaking of the integration time and spectral resolution, based on the target of interest.

5. Performance

The software's performance was generally increased thanks to the newer platform, increased multicore support, and use of the latest shared libraries. But, overall, the run times for the core function process were unfortunately increased for most use cases following this upgrade.

The primary reason for this comes from a series of bugfixes made to address previous memory leakage or access faults that let to frequent crashes of the software in certain operating conditions.

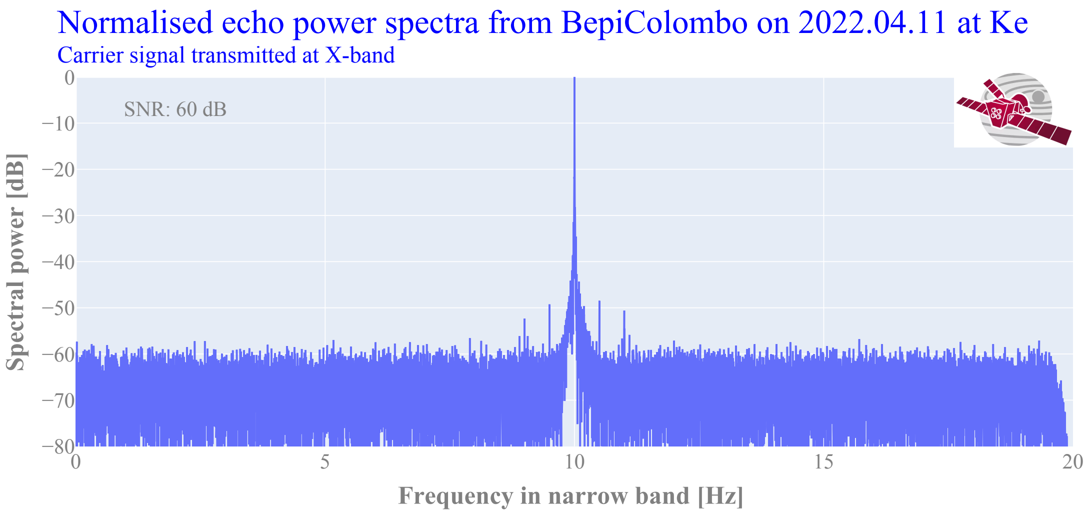

Figure 2: Normalised echo power spectra of the ultra-narrow band Ka-band carrier signal transmitted by ESA's BepiColombo (8.421 GHz) as captured by the Katherine radio telescope on 11 April 2022.

One key example was the sequence of steps in the core loop to perform rapid discrete Fourier transform (DFT) operations on large input data. In sequence, IPP functions would be called to calculate the size of buffers required for the upcoming operation, then this would be used to initialise the buffers, and finally the buffers would be used to do the DFT operation itself. Previously the first two steps happened only once per thread, with only the DFT being repeated each loop. This would crash whenever data segments was not evenly divisible by the buffer size, or if multiple data threads were included with different data properties. Moving the initialise step into the inner loop—the only way to prevent this edge case—added a costly operation to each DFT compute iteration. This was one of several similar changes that added steps to the core loop, the place where added work will impact performance most significantly, and ultimately these came at a cost of 10-30% additional run time for certain inputs.

The changes that had a slowdown effect were necessary to improve the stability of the tool for use with the full breadth of data that exists in the VLBI domain. However, performance is a high priority for the project moving forward, and this work has identified key opportunities where dramatic improvements may be achieved—such as through the planned SDtracker development to add GPU acceleration capability. This will work towards the goal of the project team to make SDtracker capable of real-time data processing during capture.

6. Closing Remarks

The software has been applied to myriad novel uses since its development, including water maser search in the Kronian moons, station performance analysis and real-time feedback, flare monitoring of spectral lines of methanol emissions, RF characterisation and pattern-of-life assessment of Earth-based satellites, bi-static radar tracking of asteroids, and space domain awareness. Detailed descriptions of these applications are available in [2].

SDtracker and its precursors have enabled a growing breadth of radio science over the past 17 years. The software is still in active development, and plans for the future prioritise ease of maintenance, novel software applications, and high performance. Ongoing improvements have made it easier for new users to adopt the software, allowing operators of astronomical radio telescopes or arrays to more easily adapt and respond to the growing demand for spacecraft, space weather, and satellite monitoring data.

Acknowledgments

SDtracker was developed by Sergei Pogrebenko, Jan Wagner and Guifré Molera Calvés, with contributions from Mars Buttfield-Addison and Nelson Nunes.

References

[1] J.-P. Lebreton, O. Witasse, C. Sollazzo, T. Blancquaert, P. Couzin, A.-M. Schipper et al., *An overview of the descent and landing of the Huygens probe on Titan*, *Nature* **438** (2005) 758.

[2] G. Molera Calvés, S.V. Pogrebenko, J.F. Wagner, G. Cimò, L.I. Gurvits, T.M. Bocanegra-Bahamón et al., *High spectral resolution multi-tone spacecraft Doppler tracking software: Algorithms and implementations*, *Publications of the Astronomical Society of Australia* **38** (2021) e065.

[3] A. Keimpema, M.M. Kettenis, S.V. Pogrebenko, R.M. Campbell, G. Cimó, D.A. Duev et al., *The SFXC software correlator for very long baseline interferometry: algorithms and implementation*, *Experimental Astronomy* **39** (2015) 259.

[4] Molera Calvés, G., Pogrebenko, S. V., Cimò, G., Duev, D. A., Bocanegra-Bahamón, T. M., Wagner, J. F. et al., *Observations and analysis of phase scintillation of spacecraft signal on the interplanetary plasma*, *Astronomy & Astrophysics* **564** (2014) A4.

[5] Bocanegra-Bahamón, T. M., Molera Calvés, G., Gurvits, L. I., Cimò, G., Dirkx, D., Duev, D. A. et al., *Venus Express radio occultation observed by PRIDE*, *Astronomy & Astrophysics* **624** (2019) A59.

[6] M. Ma, G. Molera Calvés, G. Cimò, P. Zhang, M. Xiong, P. Li et al., *VLBI data processing on coronal radio-sounding experiments of Mars Express*, *The Astronomical Journal* **162** (2021) 141.