Journal of the Korean Astronomical Society
52: 207 ∼ 216, 2019 October

https://doi.org/10.5303/JKAS.2019.52.5.207
pISSN: 1225-4614 · eISSN: 2288-890X
http://jkas.kas.org

VLBI Network SIMulator: An Integrated Simulation Tool for Radio Astronomers

Zhen Zhao, Tao An, and Baoqiang Lao
Shanghai Astronomical Observatory, Key Laboratory of Radio Astronomy, Chinese Academy of Sciences, 200030 Shanghai, China; antao@shao.ac.cn

Received July 29, 2018; accepted August 21, 2019

Abstract: In this paper we introduce a software package, the Very long baseline interferometry Network SIMulator (VNSIM), which provides an integrated platform assisting radio astronomers to design Very Long Baseline Interferometry (VLBI) experiments and evaluate the network performance, with a user-friendly interface. Though VNSIM is primarily motivated by the East Asia VLBI Network, it can also be used for other VLBI networks and generic interferometers. The software package not only integrates the functionality of plotting (u, v) coverage, scheduling the observation, and displaying the dirty and CLEAN images, but also adds new features including sensitivity calculations for a given VLBI network. VNSIM provides flexible interactions on both command line and graphical user interface and offers friendly support for log reports and database management. Multi-processing acceleration is also supported, enabling users to handle large survey data. To facilitate future developments and updates, all simulation functions are encapsulated in separate Python modules, allowing independent invoking and testing. In order to verify the performance of VNSIM, we performed simulations and compared the results with other simulation tools, showing good agreement.

Key words: techniques: interferometric — methods: numerical

1. Introduction

The East Asia Very Long Baseline Interferometry (VLBI) Network (EAVN, An et al. 2018), consisting of 21 radio telescopes in China, Japan and Korea, has performed astronomical observations since mid–2018. The diverse sub-array configurations and frequency setups of the EAVN allow to cover a wide range of research areas including astronomical masers (e.g. hydroxyl, methanol, water and SiO masers in Galactic objects and extragalactic megamasers), active galactic nuclei (AGN), pulsars and transients (e.g. supernovae, gamma-ray bursts), space exploration, astrometry and geodesy. The EAVN is expected to promote regional collaborations in East Asia. Such academic collaborations will offer great opportunities for new discoveries in astronomy and space science, and form a successful model for international academic collaborations with a sustainable operation scheme.

Along with the formal operation of the EAVN, a set of user assistance software packages, having functions for evaluating the network performance, would be needed and helpful for users to prepare proposals and is also necessary for expanding the user community. The diversity and versatility of the EAVN configurations demand such a tool to be sufficiently flexible and expandable for future development.

Currently, there are several auxiliary tools widely used in the VLBI community for different purposes. SCHED (Walker 2015) is designed by the National Radio Astronomy Observatory (NRAO) of the USA and is mainly used for scheduling VLBI observations. It supports plotting the (u, v) coverage of a given array and time range, displaying the corresponding dirty beam and the change of telescope elevation angle with time, thus assisting astronomers in choosing preferable telescopes. SCHED can also show multiple source scans in time sequence, which facilitates the arrangement of time blocks to optimize the (u, v) coverage within the allocated time period. Since SCHED is configured through importing a pre-defined file in keyin free format, any parameter change or setup adjustment requires restarting the program. SCHED is written in the FORTRAN programing language and lacks the flexibility to adapt to new VLBI telescopes. European astronomers are developing new interfaces in order to support the globalisation of VLBI.[1] Another widely-used software, Difmap (Shepherd 1997), is a successor of the Caltech VLBI package (Readhead & Wilkinson 1978; Readhead et al. 1980). Its characteristic features are interactive operations including editing, hybrid imaging, self-calibration, and automated pipeline capacities. Difmap provides both a scripting language and commands for operations. Although these software packages are still widely used in the VLBI community, their architectures and algorithms are outdated. Upgrades based on advanced programing languages such as Python are preferrable. For example, the aperture synthesis simulator (APSYNSIM) (Martí-Vidal 2017) is a Python-based software package, providing an interactive tool to visualize the aperture synthesis and perform educational-level simu-

Corresponding author: T. An

[1] JUMPING JIVE: http://jumping.jive.eu

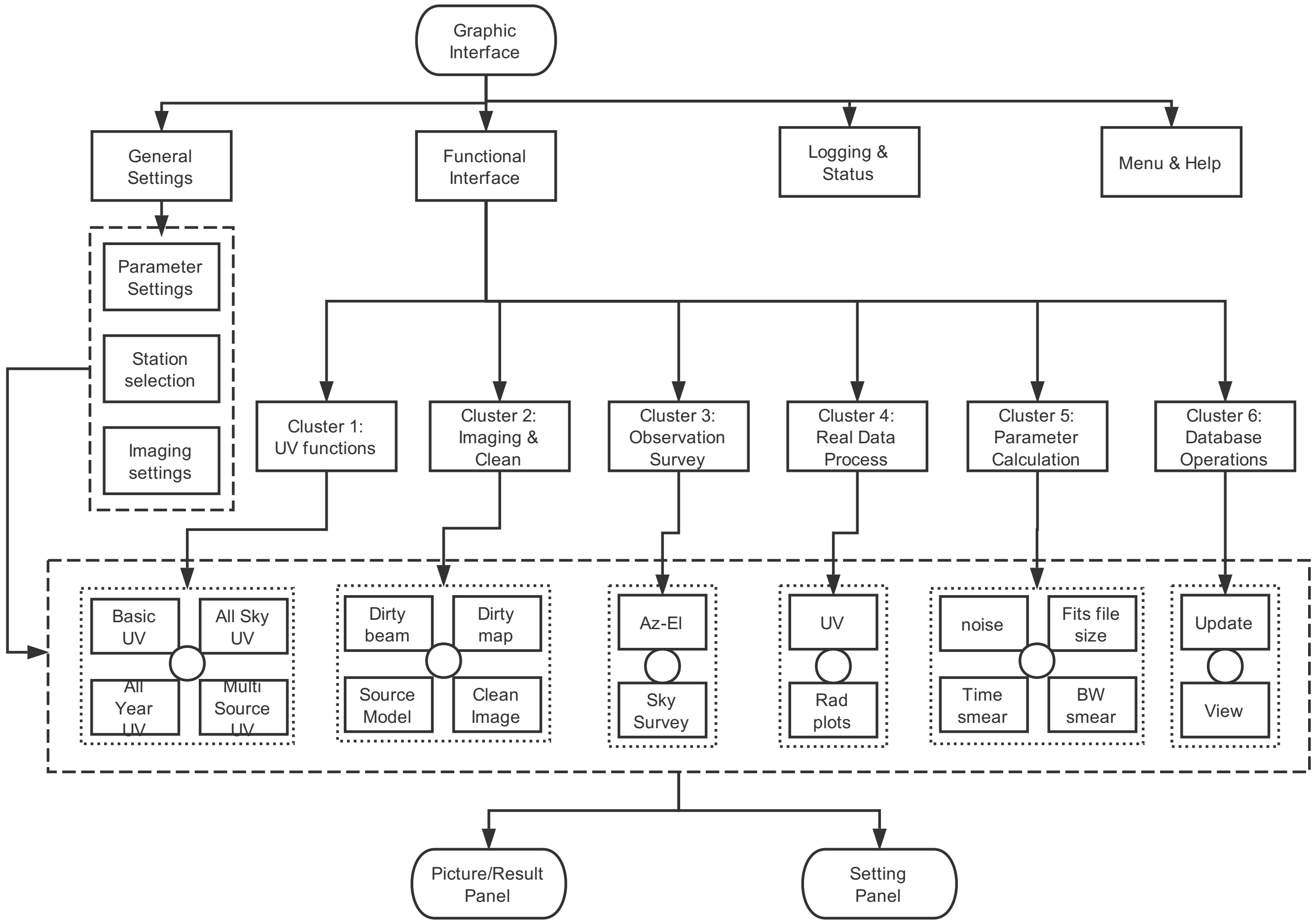

Figure 1. The software framework of VNSIM. This diagram shows the structure of the graphical user interface. Detailed description of the main functions are given in Section 3. VNSIM currently comprises six function clusters including 14 sub-functions in total.

lations which are helpful for non-radio astronomers.

Besides these auxiliary tools for ground-based VLBI networks, some other software packages were developed with extra functions tailored to supporting space VLBI. These include: the Space VLBI Assistance Software (SPAS) developed by the Satellite Geodetic Observatory of the Institute of Geodesy (Frey et al. 1998), and *Fakesat* (Smith et al. 2000) for the VLBI Space Observatory Programme (Hirabayashi et al. 2000) proposal preparation; the Astronomical Radio Interferometer Simulator (ARIS) developed by the Japan Aerospace Exploration Agency (Asaki et al. 2007, 2009) for Japan's second-generation space VLBI project VSOP2, which adds some new functions to assess the impacts of a variety of error sources on the image quality; the FakeRat software package developed by the Astro Space Center of the Lebedev Physical Institute (Zhuravlev 2015) and used for the Russian RadioAstron (Kardashev et al. 2013) space VLBI mission. The Shanghai Astronomical Observatory of China once proposed the space millimeter-wavelength VLBI array (SMVA) programme (Hong et al. 2014) and the dual-element low-frequency space VLBI observatory (An et al. 2019a,b). In order to support the SMVA, a simulation software (An et al. 2016) was developed to provide fundamental space–ground and space–space VLBI (u, v) coverage simulations.

The major functions of existing software packages adapted to specific applications have been discussed and compared in An et al. (2016). Making these tools work for a new VLBI network such as the newly operational EAVN, major modifications have to be made. Moreover, a module-independent, highly scalable, flexible, and user-friendly solution is highly desirable. For this purpose, we developed a new software package, the VLBI Network SIMulator (VNSIM), which integrates most commonly-used simulation programs and offers a user-friendly interface for radio astronomers. It is a cross-platform Python-based software package with high scalability and reusability. Each function has been separately designed and independently implemented for the sake of future extension. VNSIM allows for presenting (u, v) plots of multiple sources along with their dirty maps to evaluate the imaging performance. The

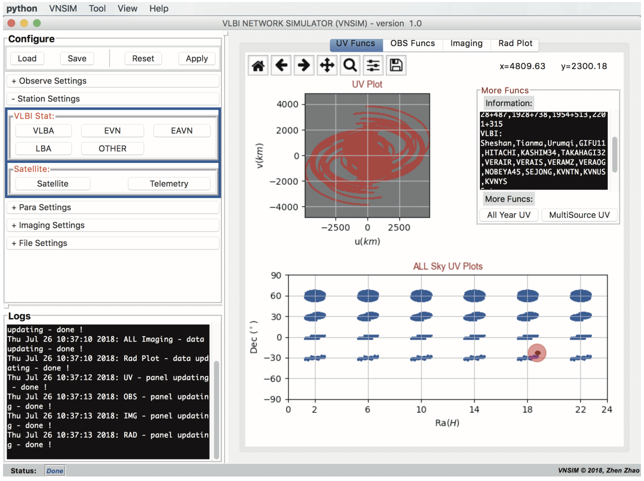

Figure 2. The main interface of VNSIM. It comprises three functional zones: configuration zone, log zone, and display zone. The plot in the top-right panel shows the EAVN (u, v) coverage for the quasar 0134+329. (u, v) coverages of other sources can be invoked by selecting them in the configuration panel (Observe Settings). The bottom-right panel shows simulations of the EAVN (u, v) coverages of sources at different RA and Dec coordinates. The red circle marks the position of the Sun on the sky at the observation epoch.

data of stations and sources are dynamically managed by the SQLite database, and the parameter configuration can either be saved and loaded directly or be specified through the interactive graphical user interfaces (GUI). Multi-processor acceleration is supported, enabling a significant decrease of the processing time for large datasets. Although VNSIM has initially been designed for EAVN, it can easily be adapted to other VLBI networks or other generic interferometers.

The remaining part of this paper is organized as follows. Section 2 describes the overall design concept of VNSIM. Details of the main functions are presented in Section 3. In Section 4, we demonstrate the major functions of VNSIM and compare our results with those of other simulation tools. A summary is given in Section 5.

2. The VNSIM

The VNSIM is programmed in the Python language (Python 3.6 or higher) and can be implemented on all common operation systems, including Microsoft Windows, Linux and Mac OS. The source package can be downloaded from the Github,[2] detailed documentation is provided on the corresponding wiki page.[3]

2.1. Overall Design

To fulfill the requirements for simulating VLBI experiments, we propose the following design considerations:

- High scalability and reusability. Each function cluster is independently encapsulated into a single Python file to facilitate testing and future extensions. In this sense, the design follows the Model View Controller (MVC) architectural pattern. We divide the development into three interconnected parts: an SQLite database for data management, a GUI designed by Tkinter for user interactions, and function clusters for parameter calculations.

- Dynamic database management. The information on VLBI stations and astronomical sources is stored and managed by an SQLite database which is implemented through a GUI, enabling users to add, delete and modify existing data records in a convenient way. It also allows for appending self-defined stations for the purpose of simulating new VLBI networks. The related database currently consists of four tables: source table, station table, satellite table, and telemetry station table; the last two are only needed for space VLBI missions. Data can either be inserted through the GUI record by record or be directly added completely through importing external files in the `.txt` or `.csv` formats.

- User-friendly interface. Both graphic and command line interfaces are supported in VNSIM for different-level users with different usage habits. In either mode, the parameter configurations can be saved and loaded for repeated use. All resulting images can be saved in a variety of image formats. Besides, indicative logging information and dialogues are displayed on the GUI to help users track the run status.

- Run performance. VNSIM is compatible with generic interferometers, such as the Very Large Array (VLA) and the upcoming Square Kilometre Array (SKA). VNSIM contains some new advanced functions which are suitable for (u, v) coverage simulations of large-scale sky surveys with these many-element interferometers and for long-term evaluation of the (u, v) coverage of key targets in space VLBI simulations. These functions involve high volumes of calculations and produce large-size images, typically consuming considerable computing resources. To tackle these problems, we implemented parallel processing. In addition, multiple threads can be separately created to maintain

Table 1
Main Python scripts and their functions.

Python Scripts	Functions
Func_uv.py	Basic UV
Func_uv_advanced.py	All-sky, all-year, multi-srcs UV
Func_obs.py	Az-El, Sky survey
Func_img.py	Beam, map, clean map, model
Func_survey_all.py	Pipeline to survey multiple sources
Func_cal.py	Parameter evaluation
Func_db.py	Database editor
Func_gui.py	A GUI tool

[2] https://github.com/ZhenZHAO/VNSIM
[3] https://github.com/ZhenZHAO/VNSIM/wiki

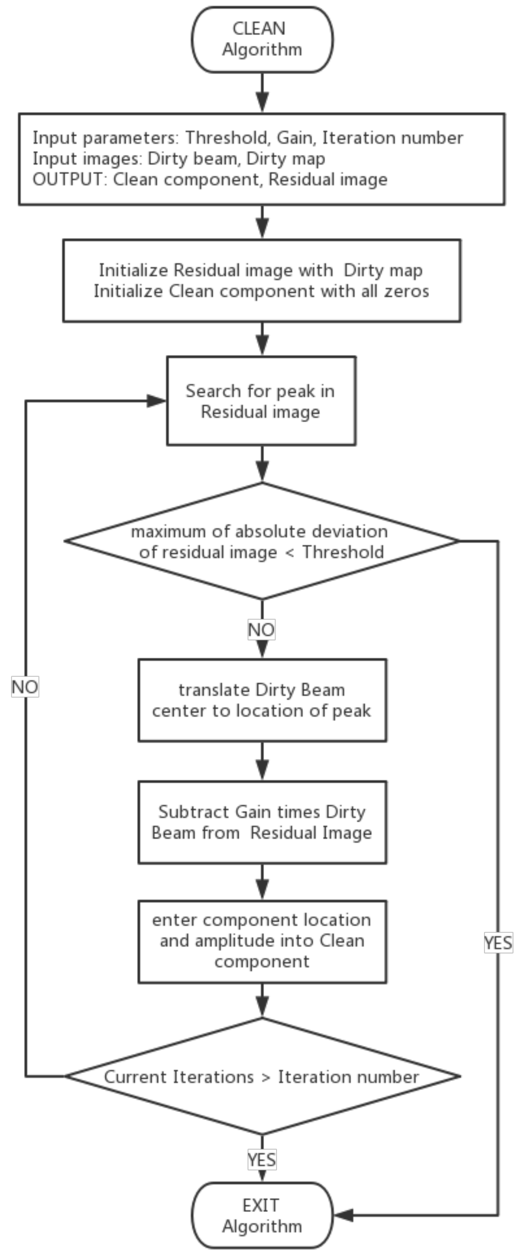

Figure 3. Flowchart of the CLEAN loop.

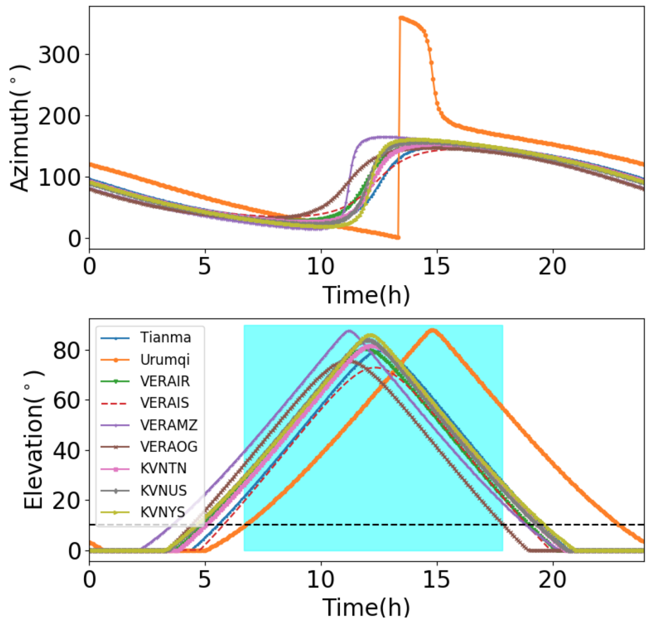

Figure 4. Variation of azimuth and elevation angles of selected telescopes with time. The horizontal black dashed line represents the cutoff angle of 10° (this value can be changed manually), the red shaded area indicates the preferred observation time range for the given target source. The telescopes used in this simulation are KVN, Tianma, VERA, and Urumqi.

graphical interfaces and functional calculations as well.

- Functional designs. To deliver a complete solution, VNSIM aims to integrate most useful simulation functions, including the (u, v) coverage plots for a single source and multiple sources, all-sky (u, v) plots of a large sample, image simulations of dirty beam, dirty map and CLEAN map, scheduling setups showing the change of azimuth and elevation of VLBI telescopes with observing time, and the visibility amplitude distribution as function of projected (u, v) distance. Some other functions related to space VLBI are also considered but will be implemented in future development stages, including changes of space VLBI (u, v) coverage with the precession of satellite orbits. Besides, VNSIM also enables users to estimate the VLBI network performance, including the sensitivity estimate with the given time range and data rate.

2.2. Software Framework

As shown in Figure 1, the framework of VNSIM essentially consists of six function clusters, parameter configuration and other common GUI components such as status bar, menu bar, and logging information. The six clusters cover four fundamental simulation functions, including (u, v) coverage plotting, dirty beam and dirty map imaging, observation survey and simulation of visibility data (clusters 1–4), and two additional functions with parameter evaluation and database management (clusters 5–6). The function clusters are mutually independent, each of them can be directly invoked after appropriate configuration and run separately.

Table 1 shows the eight main Python scripts corresponding to the aforementioned functions. All scripts are provided with command-line options and arguments. (u, v) plots and imaging-related functions require users to specify the observation settings in the configuration file. The *Func_survey_all.py* script provides a simulation routine for survey observation, which not only creates the image results of (u, v) plots, dirty map and clean map, but also records some quantitative information including the beam size, preferable observation time range, and dynamic ranges. Advanced users

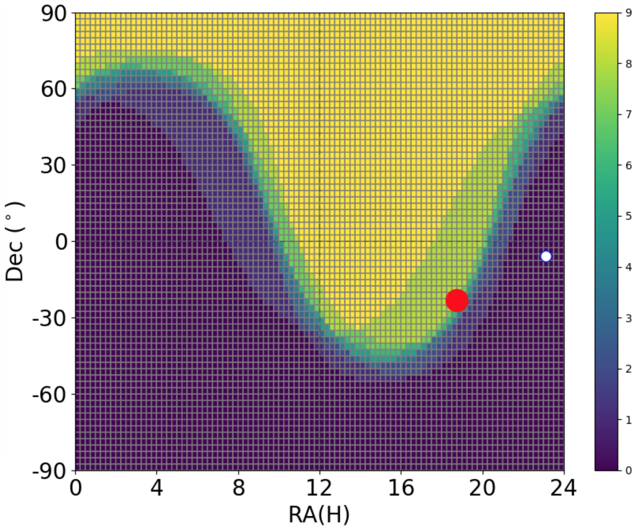

Figure 5. Number of visible stations as function of source position for a given sky survey. The red and white circles mark the positions of the Sun and the Moon, respectively. The color index denotes the number of the visible stations.

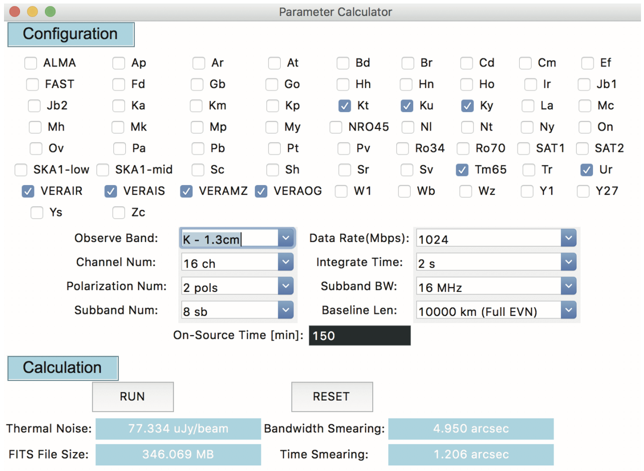

Figure 6. The VNSIM parameter evaluation GUIs.

can run the program in command-line mode, while beginners may interact with a friendly GUI.

As for Figure 2, the main area of the VNSIM GUI is divided into three parts: log panel, configuration panel and result panel. Four main simulation functions locate at different tabs on the result panel, while two additional functions can be accessed through the 'Tool' menu. The configuration panel comprises five configuring items (Observe Settings, Station Settings, Para Settings, Imaging Settings and File Settings), and only one of them is visible with the others folded up by default. The log panel shows the run information in real-time to help track the VNSIM running status. The whole interface is resizable and can be adjusted according to the display window on different computer screens.

To accelerate the processing of multiple sources, we consider the following multi-threading and multi-processing strategies. For the sake of clarity of the interface, the GUI only contains sample image demonstration. Therefore we allocate two threads to make the GUI and underlying data calculations run separately. The script, *Func_uv_advanced.py*, can provide the all-year round (u, v) plots, all-sky (u, v) plots, and multiple-source (u, v) plots. These are very time-consuming. Since the acceleration of multi-threading in Python is strictly limited by the Global Interpreter Lock (GIL), we adopt multiprocessing to optimize the computing-intensive tasks in VNSIM. To reduce the overhead of creating sub-processes, a process pool is pre-created and the number of sub-processes can be user-defined (or adopt the CPU core number by default). Each sub-process can deal with a complete combination of parameter settings, and the inter-communication among sub-processes is performed via asynchronous queue objects. Eventually, all run results are gathered and parsed at the main process. A test result is presented on the Github wiki page.[4] The computing speed is roughly proportional to the core number. Likewise, the script, *Func_uv_advanced.py*, is also accelerated by multiprocessing. Actually, to further reduce the overhead of inter-communications between sub-processes, we implement both the calculation and image-generation inside each sub-process. This is feasible owing to the complete independence among different parameter settings.

3. Major Functions

VNSIM currently comprises six function clusters including 14 sub-functions in total, described in detail below.

3.1. Simulation Functions

3.1.1. (u, v) Coverage Plotting

The projection of the separation between any two antennas in a VLBI network (the 'baseline', B) on a plane perpendicular to the direction of the observed radio source can be decomposed into east–west and north–south components, represented by u and v, respectively. Given the position $(x_\lambda, y_\lambda, z_\lambda)$ of baseline vector B_λ in an (X, Y, Z) coordinate system, the components (u, v, w) are obtained by (Thompson et al. 2007),

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} \sin H_s & \cos H_s & 0 \\ -\sin\delta_s \cos H_s & \sin\delta_s \sin H_s & \cos\delta_s \\ \cos\delta_s \cos H_s & -\cos\delta_s \sin H_s & \sin\delta_s \end{pmatrix} \begin{pmatrix} x_\lambda \\ y_\lambda \\ z_\lambda \end{pmatrix}, \quad (1)$$

where H_s and δ_s are the hour angle and declination of the target position, and $(x_\lambda, y_\lambda, z_\lambda)$ are in units of the observing wavelength. Continuous observation of a radio source projects any given baseline onto a portion of an arc of an ellipse on the (u, v) plane, resulting in characteristic curved lines in the (u, v) plane. Multiple stations of VLBI networks jointly form a set of baselines, providing the overall (u, v) coverage.

In addition to plotting the above fundamental (u, v) coverage of a single source in a fixed time period,

[4] https://github.com/ZhenZHAO/VNSIM/wiki/test-uv-advance

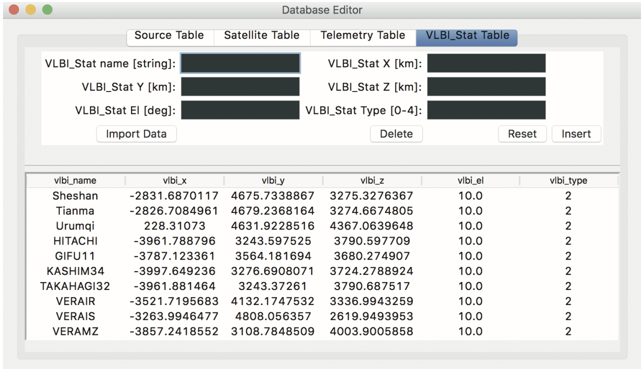

Figure 7. The GUI of the database editor provides a high quality, visual and open source tool to create, delete and edit database files. Users can either edit the records via the GUI, or import a telescope or a source table from an external file.

three additional functions related to (u, v) coverage are implemented in VNSIM: (i) all-year-round (u, v) plotting can create twelve (u, v) coverage plots, each simulating observations on the first day of each month. This function is designed to visualize the (u, v) coverage evolution of space VLBI due to satellite orbit precession. The code is partly complete; we will incorporate it into the VNSIM once the testing is completed. (ii) By evenly dividing the whole sky into 5×6 blocks, an all-sky (u, v) plotting function shows the survey ability of selected combinations of VLBI stations in a given observing period. This is useful for scheduling surveys of large samples. In the example shown in the bottom-right panel of Figure 2, the source at $\delta = 0°$ is marginally observable by the EAVN; high-declination sources have good (u, v) coverages. (iii) Similarly, a function for multiple-source (u, v) plotting displays the (u, v) coverages of multiple selected sources to assess the performance of a given VLBI network. This is invoked by clicking the button 'MultiSource UV' in the 'Tool' menu.

3.1.2. *Imaging*

The intensity distribution I_ν of a radio source at observing frequency ν can be estimated by the two-dimensional Fourier transform of the spatial coherence function $V_\nu(u, v)$ (Thompson 1999),

$$I_\nu(l, m) = \iint V_\nu(u, v) e^{2\pi i(ul+vm)} du dv, \tag{2}$$

where l, m are the sky coordinates. However, in practice V_ν is not continuous but is sampled at particular positions in the (u, v) plane. Then, we actually have

$$I_\nu^D(l, m) = \iint V_\nu(u, v) S(u, v) W(u, v) e^{2\pi i(ul+vm)} du dv, \tag{3}$$

where $I_\nu^D(l, m)$ is the 'dirty image'; $S(u, v)$ denotes the sampling function which takes the values 0 or 1 as function of (u, v) position; $W(u, v)$ represents the weighting functions (we currently consider uniform weighting in VNSIM). Besides, the observed intensity distribution is

Table 2
Simulation parameter settings.

	Parameters	Settings
General	Time Scan length Frequency	12 h 5 min 22 GHz
Source	Main Others[a]	M87 0202+319, 0529+483, 1030+415, 1128+385, 1418+546, etc.
Station[b]	CVN JVN KVN NRO VERA	Tianma, Urumqi, Sheshan Gifu, Hitachi, Kashima, Takahagi Sejong, Tamna, Ulsan, Yonsei Nobeyama Iriki, Ishigakijima, Mizusawa, Ogasawara
Imaging	Iteration Gain Threshold	100 0.2 0.001

[a] VNSIM allows for processing multiple sources simultaneously. In the examples, we aslo add 10 bright and compact AGNs to test the multi-source processing.
[b] The stations are the same as in An et al. (2018) for comparison.

often expressed as a convolution of the source intensity distribution and the synthesized beam (i.e., the point spread function) of the interferometer B_θ,

$$I_\nu^D(l, m) = I_\nu * B_\theta, \tag{4}$$

where B_θ is given by

$$B_\theta(l, m) = \iint S(u, v) W(u, v) e^{2\pi i(ul+vm)} du dv. \tag{5}$$

Given a source model (e.g., point source, Gaussian) and proper observation configurations in simulations, VNSIM can be used to create the corresponding (u, v) coverage plot, dirty beam B_θ, and dirty image I_ν^D. We also integrate the CLEAN algorithm (Högbom 1974) in VNSIM to perform quick inspection of the image quality. As shown in Figure 3, the CLEAN deconvolution algorithm can pass two controlling parameters, the 'Threshold' and 'Iteration Number' which control when the iterative procedure is stopped. Eventually, the CLEAN beam image can be obtained by convolving the CLEAN component map with a synthesized CLEAN beam and then adding the generated image back into the residual image to produce the final CLEAN image. As mentioned before, this function is useful for evaluating the performance of a new VLBI network, or selecting optimized network configuration. In future, more realistic noise models will be considered in order to bring the simulation closer to practical observing conditions.

3.1.3. *Observations*

In the third cluster (Figure 1), 'Observation Survey', we implement two types of simulations. The first one plots

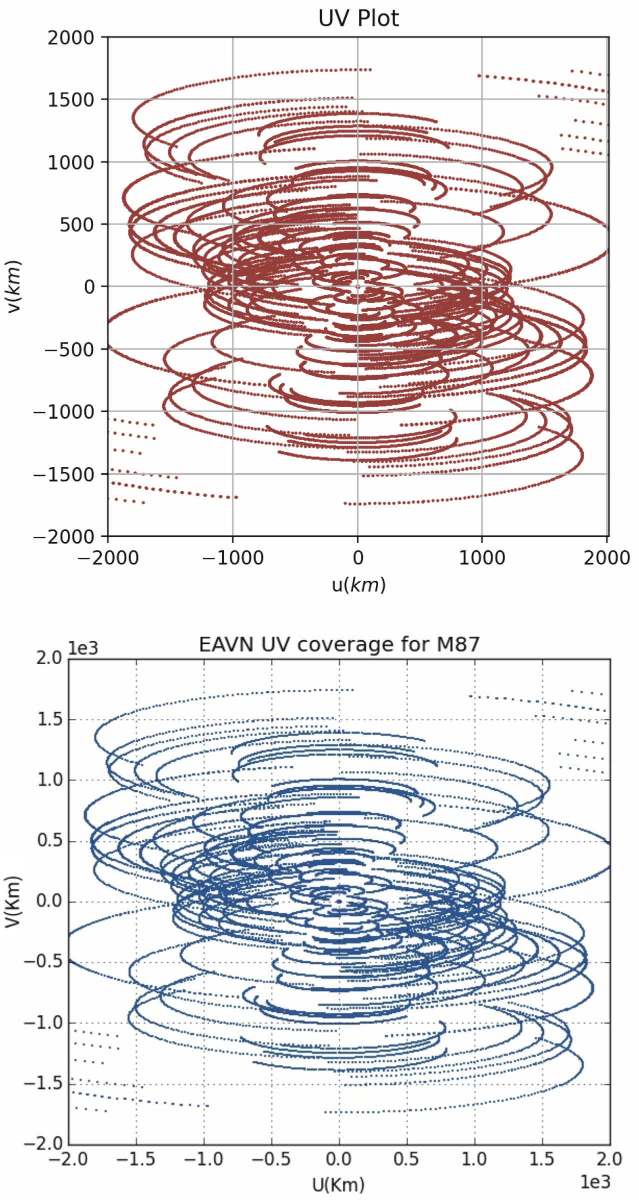

Figure 8. Comparison of (u, v) coverages of EAVN derived from VNSIM (*top*, the present paper) and from the literature (*bottom*, An et al. 2018). This diagram displays the inner 2,000 km region of dense (u, v) coverage for the galaxy M 87. They are consistent.

the azimuth and elevation angles of selected telescopes as a function of time within the observation time interval. As Figure 4 shows, it can identify the best observing time interval for a specific target source – meaning here the time interval during which the largest number of telescopes can operate at elevations higher than 10°. The other is the sky survey function, which visualizes whether different parts of the sky are visible by a VLBI network or not at a certain epoch. An an example, Figure 5 shows the number of visible stations as function of sky position along with the positions of Sun and Moon during the observing time.

Table 3
Four source models.

No.	Relative RA (mas)	Relative Dec (mas)	Flux density (Jy)
1	0.0	0.0	1.0
2	10.2	−13.8	0.5
3	−7.0	10.0	1.5
4	−11.0	−5.5	10.0

3.1.4. Processing Real Observation Data

VNSIM is not only regarded as a simulation and demonstration tool, but also it allows for processing real observing data. For example, by extracting the station position and observed visibility data, VNSIM can draw the plots of (u, v) coverage of the data and the corresponding dirty beam. As a useful demonstration, VNSIM plots the visibility amplitude as function of projected (u, v) distance, which is useful to assess the visibility quality, diagnose bad data points, and gain a rough knowledge of the source structure (resolved or unresolved).

3.2. Additional Functions

3.2.1. Parameter Evaluation

VNSIM provides an toolbox for estimating the performance of a VLBI network. The parameters included are image thermal noise Ω, bandwidth-smearing-limited field of view F_{bw}, time-smearing-limited field of view F_{time}, and an estimate of the FITS file size C.

The image thermal noise (in units of Jy beam^{-1}) can be calculated by

$$\Omega = \frac{1}{\eta} \frac{S_e}{\sqrt{\frac{rT_{obs}}{2}}}, \tag{6}$$

where r and T_{obs} are the data rate in bits per second (bps) and on-source time in seconds, respectively; η denotes the VLBI efficiency factor (commonly $\eta = 0.7$). S_e is given by

$$S_e = \left(\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} \frac{1}{SEFD_i \times SEFD_j} \right)^{-1/2}, \tag{7}$$

where $SEFD_i$ denotes the system equivalent flux density (SEFD) of the ith telescope in units of Jansky. Similar to the European VLBI Network (EVN) Calculator (Paragi 2012), we consider 2-bit sampling ($N_{bit} = 2$) by default. Thus the data rate setting and the bandwidth setting must be consistent, i.e.,

$$r = BW_{sub} N_{sub} N_{pol} N_{bit} \cdot 2 \ . \tag{8}$$

Based on the user guide of the EVN,[5] we evaluate the field of view in units of arcsec as follows. Firstly, the

[5] http://www.evlbi.org/user_guide/fov/node4.html

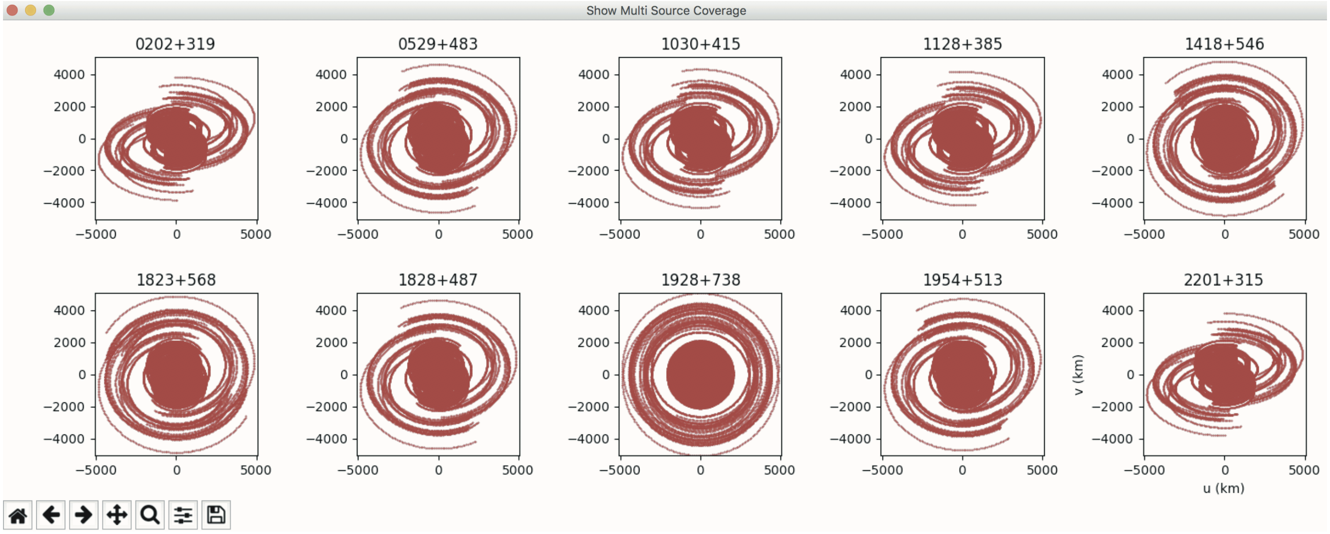

Figure 9. An example of (u, v) coverage plotting of multiple sources. This function is useful for surveys of large samples. This figure shows 10 sources observed with the same VLBI network configuration. Larger numbers of sources can be handled by means of parallel processing.

field of view limited by the bandwidth smearing effect is given by

$$F_{bw} = \kappa_1 \frac{N_{ch}}{L_{bl}\, BW_{sub}}, \tag{9}$$

where N_{ch}, L_{bl}, and BW_{sub} denote the number of channels, the maximum baseline length in km, and the sub-band bandwidth in MHz, respectively. κ_1 is a constant; its digital value is 49,500. The field of view limited by the time-smearing effect is given by

$$F_{time} = \kappa_2 \frac{L_\lambda}{L_{bl}\, T_{int}}, \tag{10}$$

where L_λ, L_{bl} and T_{int} represent the wavelength in meters, the maximum baseline length in km, and the integration time of the correlation in seconds, respectively. The constant κ_2 usually equals $18''.56$. Note that these values are calculated by taking into account a loss of 10% in the response to a point source.

Following the instructions of the EVN user guide,[6] the size of the resulting FITS file can be calculated like

$$C = 1.75 \frac{N_{sta}^2\, N_{pol}\, N_{sub}\, N_{ch}}{131,072} \frac{1}{T_{int}} \frac{T_{obs}}{3600}, \tag{11}$$

where N_{sta}, N_{pol}, and N_{sub}, N_{ch} denote the number of stations, polarizations, sub-bands, and channels per sub-band, respectively. T_{obs} and T_{int} are the total observation time and the integration time of the correlation in units of seconds, respectively. More specifically, a single board will output 48 kByte of data per integration, which consists of 32 kByte lag-based correlation data and 16 kByte header information. Considering only the correlation-function data portion, it yields 0.5 MByte per integration for the whole correlator. The EVN correlator has 32 correlator boards, each currently handling a total of 4096 lags. Thus the maximum frequency points for each corelator is 65,536. We note that other correlators may have different configurations. The observation frequency point N_{freq} can be calculated as

$$N_{freq} = \frac{N_{sta}^2}{2} N_{pol}\, N_{sub}\, N_{ch} \tag{12}$$

Hence the equivalent number of the correlator, N_c, can be calculated like

$$N_c = \frac{N_{freq}}{65,536} = \frac{N_{sta}^2\, N_{pol}\, N_{sub}\, N_{ch}}{131,072}. \tag{13}$$

Consequently, the capacity in GB for one hour of observations is

$$C_h = N_c \times \frac{0.5}{1024} \times \frac{3600}{T_{int}} = 1.75 \frac{N_c}{T_{int}}. \tag{14}$$

Accordingly, the capacity C, in the units of GB, for an observation T_{obs} seconds long is

$$C = C_h \frac{T_{obs}}{3600}. \tag{15}$$

The parameter evaluation interface of the VNSIM is shown in the top panel of Figure 6 . Commonly used telescopes are included in the default list. Other telescopes can easily be added via the database. The configuration parameters can be selected from the drop-down menu. The on-source time text frame accepts manual input. After clicking the button 'RUN', the calculated parameters are shown in the bottom lines. In Figure 6, typical stations of the EAVN at K band were selected and the four parameters discussed above were calculated and displayed on the GUI.

[6] http://www.evlbi.org/user_guide/fov/node8.html

3.2.2. *Database Management*

Compared to other existing software packages, VNSIM features a user-friendly GUI, allowing for easy access and operation. We selected SQLite as the SQL database engine. It is self-contained, highly reliable, open source and full-featured. The graphic editor of the database management is shown in Figure 7. The related database consists of four tables that are responsible for manipulating the corresponding data of the sources, satellites, VLBI stations and telemetry stations, respectively. Taking the VLBI station table as an example, users can delete old records or insert new records easily through the GUI operations. In addition to inserting new data record by record, users can also import multiple records through loading an external file containing properly formatted data.

4. Example Experiments

In this section, we regard two main functions, (u, v) coverage and source imaging, to demonstrate the operation of VNSIM.

4.1. *(u,v)* Plotting of a Single Source

The (u, v) coverage is essential for evaluating the VLBI network performance and predicting the image quality. We first use VNSIM to create the (u, v) coverage plot of a single source, and compare it to the output of other software packages. The radio galaxy M87 is chosen as the target. As VNSIM supports parallel processing of multiple sources, we also import ten more sources in the source list to test the multi-source processing. The experiment parameters used in this simulation are listed in Table 2. To ease comparison, these parameters are the same as the ones adopted in Figure 2 of An et al. (2018). The 22-GHz frequency band, one of the operational frequency bands in the first EAVN open-use session, is chosen. Sixteen telescopes are included in the simulation. The 12-hour full-track (u, v) coverages of M87 are shown in Figure 8. VNSIM obtained exactly the same (u, v) plot (Figure 8, left panel) as the one shown in An et al. (2018). We note the (u, v) plot in An et al. (2018) was generated by using the UVSIM software (An et al. 2016), whose results have been compared with SCHED. The consistent results in Figure 8 confirm the accuracy of VNSIM and its compatibility with other similar software tools.

4.2. *(u,v)* Plotting of Multiple Sources

One of the new practical functions included in VNSIM is the multi-source (u, v) coverage plotting. It allows for calculating the (u, v) coverages of multiple sources with pre-defined observing configurations through one-shot runs and either displays the results in the interface or saves them in external files. Figure 9 demonstrates an example of the multi-source (u, v) plotting function. In this experiment, we calculated the (u, v) coverages of ten radio-loud active galactic nuclei selected from Cheng et al. (2017). The configuration is summarized in Table 2 and Section 4.1 The observing frequency is 22 GHz. The total observing time interval for each source in the simulation is 12 h. As all these sources are at high declination, they are visible by the EAVN most of the time. In particular, 1823+568 and 1928+738 show almost circular (u, v) coverages. Moreover, the maximum number of sources is not limited to ten. More sources are allowed and the calculations can be accelerated by multiprocess parallelization. Apart from displaying the results in the GUI, the output can also be exported to Encapsulated PostScript files or files in other formats.

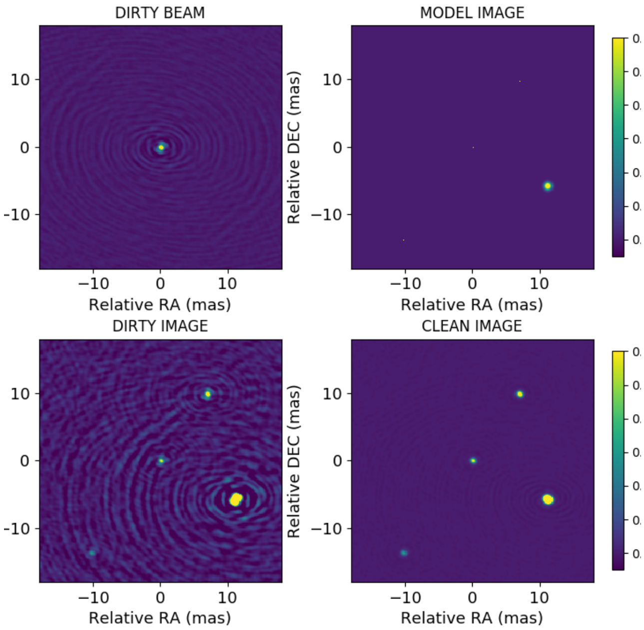

Figure 10. Image simulation. The major and minor axis lengths of the dirty beam *(top left)* are 1.01 mas and 0.85 mas, respectively. *Top right:* Distribution of four simulated sources. *Bottom left:* Dirty image. The sidelobes around each source are clearly seen. *Bottom right:* Deconvolved image after 100 iterations of CLEANing. No significant sidelobes can be seen, the dynamic range is 705.

4.3. Dirty and CLEAN Images

VNSIM is not only a simple display tool of (u, v) coverage, but also a powerful software package enabling image simulations for a visual evaluation of VLBI network imaging performance. To demonstrate this functionality, we created four artificial target sources whose parameters are listed in Table 3. We applied the EAVN station and general settings shown in Table 2. The dirty beam and source models are displayed in the top panel of Figure 10. Source 1 with a flux density of 1 Jy is a point source, located at the image center corresponding to the sky position RA=03h19m48.160s and Dec=41°30′42″.10. Sources 2 and 3 are point sources located to the southeast and northwest of Source 1. Their corresponding flux densities are 0.5 Jy and 1.5 Jy, respectively. Source 4 is a spatially extended Gaussian source, with a size of $1.5 \times \theta_{\rm beam}^{\rm min}$ and a total flux density of 10 Jy, located southwest of the image center. Figure 10, bottom-left panel, depicts the corresponding dirty image. The sidelobes around each source are clearly visible. The bottom-right panel shows the deconvolved image after 100 iterations of CLEANing. The sidelobes have been reduced substantially, and the dynamic range increases from 11 (in the dirty image) to 705 (in the CLEAN image). Deeper CLEANing may further decrease the rms noise.

5. SUMMARY

In this paper, we have introduced VNSIM, a software tool aiding VLBI network simulations. The motivation is to provide an integrated software package to aid radio astronomers in preparing observation schedules and to gain a preliminary evaluation of the interferometer's performance. Compared to existing simulation tools, VNSIM not only integrates commonly used functions but also supplements new features supporting large surveys containing multiple sources. The source code is written in Python, the individual programs are easy to extend and update. By design, VNSIM is not limited to VLBI networks, but is also applicable to connected-element interferometers. Considering the usability for non-VLBI astronomers without much interferometric knowledge, VNSIM has been designed to provide user-friendly interfaces and convenient database management. All parameters can be user-specified to adapt to different simulation scenarios. The comparison of the simulation results from VNSIM to those from other tools verifies the consistency between them. The current version of VNSIM provides functionalities covering the key requirements for scheduling and evaluating ground-based interferometric observations. More sophisticated functions are under development. In the future, we aim to provide space VLBI simulations that consider realistic constraints on observations.

ACKNOWLEDGMENTS

The simulations were performed on the China SKA Regional Data Center Prototype funded by the Ministry of Science and Technology of China (grant No. SQ2018YFA040022) and the Chinese Academy of Sciences (CAS, grant No. 114231KYSB20170003). T.A. thanks the youth innovation promotion association of the CAS. B.Q.L. acknowledges financial support by the National Natural Science Foundation of China (No. U1832104, 11703069). The authors are grateful to Zsolt Paragi for his help with VLBI network parameter evaluation, and Sandor Frey for his constructive comments on simulations of space VLBI observations.

REFERENCES

An, T., Lao, B., Wang, J., et al. 2016, Space Millimeter-wavelength Very Long Baseline Interferometry Simulation Software, Proc. 8th IEEE International Conference Communication Software and Networks, 274

An, T., Sohn, B. W., & Imai, H. 2018, Capabilities and Prospects of the East Asia Very Long Baseline Interferometry Network, Nat. Astron., 2, 118

An, T., Jaiswal, S., Mohan, P., Zhao, Z., & Lao, B. 2019, A Cosmic Microscope to Probe the Universe from Present to Cosmic Dawn – Dual-element Low-frequency Space VLBI Observatory, Chinese J. Space Sci., 38, 242

An, T., Hong, X., Zheng, W., et al. 2019, Space very long baseline interferometry in China, Adv. Space Res., in press (arXiv:1901.07796)

Asaki, Y., & Miyoshi, M. 2009, A Future Submillimeter-wave Space VLBI, in: Approaching Micro-Arcsecond Resolution with VSOP-2: Astrophysics and Technologies, 402, 431

Asaki, Y., Sudou, H., Kono, Y., et al. 2007, Verification of the Effectiveness of VSOP-2 Phase Referencing with a Newly Developed Simulation tool ARIS, PASJ, 59, 397

Cheng, X. P., An, T., Hong, X. Y., et al. 2017, The Most Compact Bright Radio-loud AGNs – II. VLBA Observations of Ten Sources at 43 and 86 GHz, ApJS, 234, 17

Frey, S., Fejes, I., Heitler, G., et al. 1998, Space VLBI Assistance Software – New Version, IAU Colloquium, 164, 399

Hirabayashi, H., Hirosawa, H., Kobayashi, H., et al. 2000, The VLBI Space Observatory Programme and the Radio-astronomical Satellite HALCA, PASJ, 52, 955

Högbom, J. A. 1974, Aperture Synthesis with a Nonregular Distribution of Interferometer Baselines, A&AS, 15, 417

Hong, X., Shen, Z., An, T., Liu, Q. 2014, The Chinese Space Millimeter-wavelength VLBI Array – Step toward Imaging the Most Compact Astronomical Objects, Acta Astronautica, 102, 217

Kardashev, N. S., Khartov, V. V., Abramov, V. V., et al. 2013, "Radioastron" – A Telescope with a Size of 300 000 km: Main Parameters and First Observational Results, Astron. Rep., 57, 153

Smith, J. G., et al., JPL Contribution to the VSOP Mission, Advances in Space Research, 26, 637

Martí-Vidal, I. 2017, APSYNSIM: An Interactive Tool To Learn Interferometry, arXiv:1706.00936

Paragi, Z. 2012, EVN Calculator, http://old.evlbi.org/cgi-bin/EVNcalc

Readhead, A. C. S., Wilkinson, P. N. 1978, The Mapping of Compact Radio Sources from VLBI Data, ApJ, 223, 25

Readhead, A. C. S., Walker, R. C., et al. 1980, Mapping Radio Sources with Uncalibrated Visibility Data, Nature, 285, 137

Shepherd, M. C. 1997, Difmap: An Interactive Program for Synthesis Imaging, in: Astronomical Data Analysis Software and Systems VI, 125, 77

Thompson, A. R. 1999, Fundamentals of Radio Interferometry, Synthesis Imaging in Radio Astronomy II, 180, 11.

Thompson, A. R., Moran, J. M., Swenson, G. W. 2007, Interferometry and Synthesis Imaging in Radio Astronomy (New York: Wiley & Sons)

Walker R. C. 2015, SCHED, http://www.aoc.nrao.edu/software/sched/

Zhuravlev, V. I. 2015, Fakerat Software in the International Interferometric Radioastron Project with Very Long Ground–Space Bases, Cosmic Research, 53, 216