Post-VME VLBA Observing of Moving Sources

VLBA Sensitivity Upgrade Memo 49

Walter Brisken

12 Dec 2019

1 Background

The VLBA has had the ability to track moving sources for many years. This capability has been used many times over the last 2 decades in projects such as the Spacecraft Navigation Pilot Project (which brought these capabilities to maturity), subsequent spacecraft astrometry (such as the Huygen's probe descent into the atmosphere of Titan and improvement of the outer solar system ephemeris through tying the location of Saturn's center of gravity to that of the inertial quasar frame), and asteroid radar observations. Observations of the Juno spacecraft orbiting Jupiter continue. In these past observations, the VLBA-specific VME control file (the `crd` file produced by `sched`) contained station-specific topocentric proper motions for each moving source, enabling a piece-wise linear description of the path of an object on the sky. With the advent of the retirement of the VME's in late October, 2019, this mechanism to convey object motion to the control software was lost. In this memo, the replacement approach is documented.

2 Executing VLBA observations

This section describes the information flow for data pertaining to execution of a VLBA observation. A description of the simpler case in which there are no moving objects is described first. Following that are descriptions of required changes to the steps in which moving (and/or near-field) object observing is being performed.

2.1 Observing non-moving sources

The following steps briefly describe the data flow for "normal" VLBA observing:

1. The PI produces a `.key` file containing a high-level description of the observation to be executed. This file contains enough information to specify choreography of the antennas' motion, the recorders, and tuning.

2. `sched` is run on the `.key` file. This creates as its primary output a human-readable `.sum` file with detailed scheduling information and the computer-readable `.vex` file. A `.oms` file is created which is used by operations to populate the OMS database; it is used later in the process to form the skeleton `.v2d` file used for correlation. Historically important `crd` files are created but these are not used anymore.

3. VLBA Operators run `queueVex` on the `.vex` file. This operation does the following:

   (a) calls `vex2script`, which creates station-specific `.py` files that are run by the `executor` program at each station's control computer,

   (b) uses `scp` to transmit these files to each station,

(c) and updates each station's `observ.tx` file to place it in the observing queue.

4. The `ObserveScriptMonitor` at each site will tell the `executor` process to execute each `.py` file at the appropriate time.
5. After observing, VLBA Operators collect monitor data with `rdbetsm`.
6. After the Mark6 media has arrived in Socorro, the modules are indexed by the Tape Librarian.
7. Analysts prepare correlator control files:
   (a) `vex2obs` applies EOP, clock, and module information to the `.vex` file, creating the `.vex.obs` file,
   (b) `oms2v2d` is run, which is used to create a template DiFX correlation file based on `sched`'s output; this file links to the `.vex.obs` file, which contains most of the information needed by the correlator,
   (c) updates to correlation parameters, zoom bands, source or antenna locations, . . . , can be added to the `.v2d` file at this time.
8. The VLBA Operators run the correlator.
9. Analysts convert the correlator output to FITS-IDI format, merging in monitor data in the process.
10. Data is "sniffed" for quality control.
11. Data is entered into the NRAO archive.
12. PI is notified of data availability.

Some of these steps are documented in more detail in VLBA Sensitivity Upgrade memo 44 (at https://library.nrao.edu/public/memos/vlba/up/VLBASU_44.pdf).

2.2 Moving sources: the `.key` file

PIs wanting to observe a moving source need to provide to `sched`, via the `.key` file, information on the object's trajectory. This is done through specification of an ephemeris file and indication of the object number within that file. Two formats are currently supported, both of which provide the location of the object in three-dimensional space, allowing near-field corrections to be done properly.

In a `sched` source catalog (either directly in the `.key` file or via external files to be included), four parameters must be specified. If the ephemeris is described via a JPL Binary SPice kernel (BSP) file, a source entry may look like:

```
satinit /
  satname = 'JUNO'
  satnum = 61
  kerfile = naif.0012.tls
  satfile = juno_190812.bsp
  /
endsat /
```

Here `satname` is the source name that will be used later in the schedule file, `satnum` is the numerical identity of the object within the ephemeris file, `kerfile` is a file containing information on leap seconds (available from https://naif.jpl.nasa.gov/pub/naif/generic_kernels/lsk/), and `satfile` is the ephemeris file itself.

NORAD Two Line Element files (TLEs) are the other supported ephemeris file type. These files are specific to Earth-orbiting objects. Use of the BSP format is recommended whenever possible as the TLE data format is not officially publicly documented; it has been reverse-engineered by various groups. Further, the information in these files needs to be "propagated" through orbit modeling, a process that has generally accepted best practices, but multiple standards within implementation details. Finally, the precision of the data that can be stored in the two-line elements is limited. In practice, TLEs from http://celestrak.com/NORAD/elements/ that are not more than a few days old tend to have precision better than an arcminute, sufficient to point VLBA at frequencies up to 25 GHz, but perhaps insufficient at higher frequencies. See https://www.celestrak.com/NORAD/documentation/tle-fmt.php and links within for information and caveats about TLE data. The `sched` input describing a TLE-based ephemeris looks something like:

```
satinit /
  satname = 'ISS'
  satnum = 61
  kerfile = naif.0012.tls
  tlefile = iss.tle
  /
endsat /
```

Here all of the parameters are in common to the BSP case except `tlefile` is used to specify the ephemeris file.

Note that the above instructions were valid prior to the change to retirement of the VME; no change here should be noticeable to the PI. Also note that ISS (International Space Station) is a bad example here as it is too low for significant mutual visibility between VLBA stations and it is moving across the sky at a rate faster than the VLBA slew rate.

2.2.1 Caveat

When operating with VME control, there was an option to explicitly specify a proper motion (and epoch of position) for a slowly moving source. This capability, as of this writing, is no longer possible.

2.3 Moving sources: the `.vex` file

The VEX 1.5 specification (https://vlbi.org/vlbi-standards/vex/), which is currently used by the VLBA, and most VLBI systems worldwide, does not support general orbital motions. Within the VLBA, a defined, but not otherwise used parameter of the `SOURCE` descriptions in the `.vex` file is used. This parameter, `source_type`, is set to include two values, the name of the ephemeris file and the source id number, e.g.:

```
def JUNO;
     source_name = JUNO;
     source_type = spk_pre_190529_190812.bsp : -61;
     ra = 16h58m07.7347183s; dec = -22d20'56.581713"; ref_coord_frame = J2000;
```

Notes:

1. The above example is stripped of extraneous details; `sched` will include additional comments in this `SOURCE` section.

2. A nominal right ascension and declination is provided; this is a meaningless value that should not be used in downstream processing.

3. BSP files contain tables that link one object (either a planet, spacecraft, moon, or barycenter) to one other object. These can be chained together. For example, the ephemeris may have the orbit of a moon around Jupiter, the orbit of Jupiter around the solar system barycenter, and Earth's orbit around the solar system barycenter. From these three linkages, the state vector for the Jupiter moon as seen from Earth can be computed. The BSP file must contain full linkage between certain solar system objects to be used by sched and the VLBA correlator: solar system Barycenter (object id 0), Earth barycenter (object id 3), and Earth (object ID 399). A BSP file containing only the orbit of an object with respect to the earth will need to be "merged" with the solar system ephemeris so that all of the necessary state vectors can be calculated. This merging can be performed with the JPL Spice Toolkit utility called `spkmerge` available at https://naif.jpl.nasa.gov/naif/toolkit.html .

2.4 Moving sources: vex2script and the .py file

The `executor` process controls the VLBA antennas based on a python control script. This code has been inherited from the EVLA project. There are provisions within this system to provide a polynomial description of the apparent topocentric coordinates of the source as a function of time in the form of a series of polynomials. Each scan on the target source can be described by a different polynomial. By default each polynomial is computed using a 5 term expansion. This looks something like the following within the `.py` file:

```
source_JUNO_orbit = []
source_JUNO_orbit.append( { 'interval' : (58737.957292,58737.964850),\
   'ra' : [4.442421, 0.001067, -0.000497, -0.000626, 0.000947],\
   'dec' : [-0.390070, -0.000041, -0.000677, -0.002409, -0.009512]} )
source_JUNO_orbit.append( { 'interval' : (58737.958935,58737.966505),\
   'ra' : [4.442423, 0.001065, -0.000500, -0.000687, 0.004828],\
   'dec' : [-0.390070, -0.000043, -0.000689, -0.002465, -0.009997]} )
.
.
.
source_JUNO_orbit.append( { 'interval' : (58738.117662,58738.125243),\
   'ra' : [4.442575, 0.000823, -0.001053, 0.002023, 0.145498],\
   'dec' : [-0.390119, -0.000944, -0.010147, -0.098244, -1.225929]} )
```

This data structure is referenced later in the `.py` file through command sequences resembling:

```
subarray.setSource(source17)
subarray.setInterferometerModel(
  MakeTopocentricInterferometerModel(source_JUNO_orbit[0]))
recorder0.setPacket(0, 0, 36, 5008)
```

```
subarray.setRecord(mjdStart + 213*second, mjdStart+263*second,
  'No0003', obsCode, stnCode)
subarray.execute(mjdStart + 197*second)
```

vex2script makes use of an external command line program called OrbitGenerator to compute these polynomials. This calculation is transparent to operations as long as the .vex file has the correct file reference and that the ephemeris file is located in the directory where vex2script is run.

2.5 Moving objects: correlation

Correlation of moving objects requires explicit intervention by Data Analysts preparing the .v2d file. Specifically, the moving source needs to be linked to the ephemeris file and the object number. Between the time of observation and time of correlation improved ephemerides may have become available. If so, those newer ones should be used for correlation.

The SOURCE entry in the .v2d file should be amended with ephemFile and ephemObject, as in the example below:

```
SOURCE JUNO { calCode=Z ephemFile=juno_190812.bs ephemObject=61 }
```

Provided the ephemeris file is located in the same directory as the .v2d file, correlation should proceed properly.

3 A note about planets

sched and the executor recognize some source names as being planetary (e.g., VENUS, JUPITER, ...). These objects can be useful sources during pointing observations but are rarely useful targets for VLBI due to theie low surface brightness at radio frequencies. No explicit mention of ephemeris files is needed; sched and executor both have internal solar system ephemerides which are used for these objects. In the .key file, one can add in the schedule, for example:

```
INTENT = PLANET_VENUS
```

to make use of this functionality.

4 Acknowledgements

Thanks to Jim Jacobs and Barry Clark for making code changes to vex2script and executor in support of the new capability. Thanks to Vivek Dhawan for nudging me to write this memo. Thanks to Vivek Dhawan, Craig Walker, Justin Linford and Barry Clark for useful suggestions based on a draft version of this memo.