Subj: Usuda - HALCA Range Prediction: The `orbitUsuda` program
From: Glen Langston
Date: 99 March 31

Summary

This document describes a simple test of the Usuda satellite prediction and a new program, `orbitUsuda`, to generate a satellite range prediction for Usuda based on the JPL Navigation Groups HALCA satellite orbit prediction. We find that the new frequency predictions from `orbitUsuda` are generally *internally* more self-consistent, when compared with the original predictions.

Background

Unknown problems existed in the Usuda Time Correction Files (TCFs), which resulted in large residual offsets ($\sim \pm 10\mu\ sec$) and rates ($\sim \pm 500 \times 10^{-12}\ sec/sec$) at correlation. Tests of the Usuda station hardware had found no problems and also found that all Usuda time constants used in the process of making a TCF were consistent with hardware measurements. (See OVLBI-ES memo 74, "Usuda Time Correction File Tests: Phase detection delays")

Yasuhiro Murata and Hideyuki Kobayashi had suspected there may be some in-accuracies in the original predictions due to the relatively high frequency residuals ($> 100Hz$) sometimes seen during Usuda tracking passes.

Nori Kawaguchi and his student Kazumasa Suzuki were making good progress towards reducing the high frequency residuals via a different method, changing the up link frequency so as to zero the down link frequency residuals. (Called the closed loop method.)

Satellite Prediction Summary

In order to understand the satellite prediction tests, the general function of the satellite prediction must be summarized. Each Space VLBI tracking station takes several input files including a file containing information to send a stable up link tone to the satellite. For Green Bank, the up link and down link tones are calculated from the satellite range (in seconds) and range rate (seconds/second). At the other stations, other equivalent forms of range rate prediction data are used.

For Usuda, the contents of the satellite range prediction file are a 6 line header describing the orbital elements used to calculate the rest of the prediction file, followed by a table with 7 columns. The table contains one entry for every second of the Usuda tracking pass. The 7 columns are:

Time Tag, t : Time at which the frequency prediction should be used. (DDDhhmmss format, where DDD is day of year and hhmmss are hours, minutes and seconds. Ie 1998 July 26 at 12 UTC is 207120000)

Up link range, $R_{up}(t)$: The distance (km) a photon of light will travel to reach the satellite, if transmitted at t. Note that this is not the instantaneous distance, because the satellite will move while the photon is traveling to the satellite. The time it takes for a photon to travel to the satellite is $\tau_{up} = R_{up}(t)/c$.

Down link range, $R_{down}(t)$: The distance (km) a photon of light will travel to the station from the satellite, if *received* at t. Note that this is not the instantaneous distance, because the station will move while the photon is traveling. The time it takes for a photon to travel to the station is $\tau_{down} = R_{down}(t)/c$.

Up link range rate : Rate of change of the up link satellite range (item 2) in units of km/sec. This item is not used at the Usuda station.

Down link range rate : Rate of change of the satellite down link range (item 3) in units of km/sec. This item is not used at the Usuda station.

Up link IF frequency, $\nu_{IF,up}(t)$: The IF frequency (Hz) corresponding to the RF frequency transmitted at time t such that when the up link is received at the satellite, the frequency is exactly 15.3 GHz. The IF frequency is up-converted (not multiplied) from IF to RF. The IF frequency is defined such that 65,000,000 Hz corresponds to a 15.3 GHz up link RF signal.

Down link IF frequency, $\nu_{IF,down}(t)$: The IF frequency (Hz) corresponding to the RF frequency *received* at time t assuming that the satellite transmitted a down link frequency of exactly 14.2 GHz. The IF frequency is up-converted (not multiplied) from IF to RF. The IF frequency is defined such that 65,000,000 Hz corresponds to a predicted down link RF signal of 14.2 GHz.

The design of the Space VLBI system does *not* require perfect up and down link predictions. The design requires the measurement of the residual frequency in order to correct for errors in the frequency and range predictions. The measurement of residual phase allows the satellite predictions to have errors as great as a few km. For this reason, the satellite predictions are not considered to be an important contribution to the residual delays and delay rates seen at correlation of the wide band data.

However, it is critical that the up link and down link predictions have the same relationship (Ie. use the same satellite prediction) or the measurement of residual phase is corrupted by *differences* in the predictions.

Tests

A test program, called `testPrediction`, uses a simple model to calculate the down link frequency prediction based only on the up link prediction and the satellite ranges. The purpose of this test was to check the consistency of the up and down link predictions. The program can not check the absolute accuracy of the satellite frequency predictions.

The model assumes that the up link frequency prediction is defined so that when the up link is transmitted at time t it arrives at the satellite with frequency exactly 15.3 GHz. The down link frequency is the frequency that will be received at the time t, assuming the satellite down link was 14.2 GHz. Therefore, using the up and down ranges, one frequency can be used to estimate the other frequency. Here the down link is calculated from the up link.

The rate of change of up link frequency is simply estimated as

$$\frac{d\nu_{up}(t)}{dt} \sim [\nu_{up}(t+1)) - \nu_{up}(t-1)]/2 \quad Hz/sec$$

where the ν_{up} values are RF frequencies calculated from the IF frequencies ($\nu_{IF,up}$) in the prediction table.

The down link RF frequency is estimated from the up link by the following equation.

$$\nu_{Est,down} = 15.3 \times 14.2/[\nu_{up} - ((\tau_{up} + \tau_{down}) \times \frac{d\nu_{up}}{dt})] \quad GHz$$

where τ_{up} and τ_{down} are the up and down link delays calculated from the prediction table.

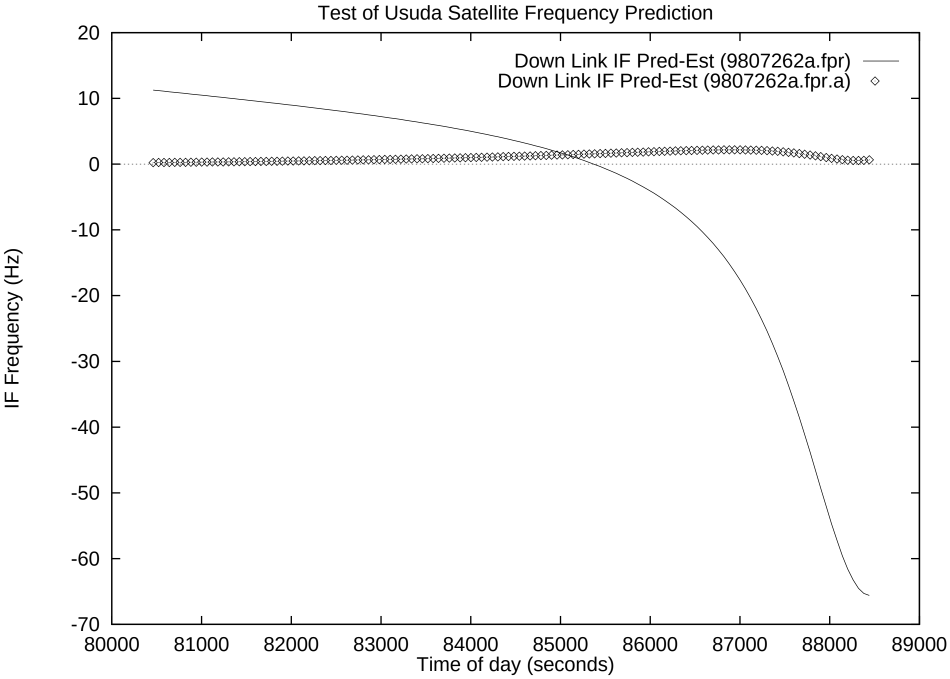

Figure 1: Tests of the ISAS Usuda-HALCA frequency predictions. The X-axis is the time of day (seconds) and the Y-axis is the difference between the down link frequency prediction and the estimated down link frequency based on the up link prediction. The solid line is a test based on ISAS prediction for Usuda tracking pass 9807262a. The thick line shows the difference based on the JPL HALCA prediction using `orbitUsuda`*. The difference between the down link prediction and the estimated down link value should be small.*

Figure 1 shows the `testPrediction` results for two predictions for the Usuda tracking pass on 98 July 26. The original prediction is contained in file names 9807262a.fpr and the prediction from `orbitUsuda` is in 9807262a.fpr.a.

Several points should be noted. The high differences in the down link prediction and estimated value for the original satellite prediction seem to smoothly change during the pass. There are large differences at the beginning and end of the pass which are of the same magnitude as the observed frequency residuals at Usuda station. up link/down link inconsistencies could cause high residual frequencies to be measured at the tracking station, but the residuals will not correctly measure errors in the predicted satellite range.

Also note that the original prediction appears to be more self-consistent than the new prediction for the central 10 % of the tracking pass.

Because the down link estimate is only approximates the full calculation of the motion of station and satellite while the link tone is transmitted, some errors might be expected. I have not found a simple way of calculating what the expected error in the approximation should be. (Ie. I have no evidence that the new prediction is as accurate as it could be.) Roughly, we would like the up link tone to have the same accuracy as the maser signal, approximately 1 part in 10^{12}, which corresponds to 0.015 Hz error on the 15.3 GHz up link. The difference between the prediction and the estimate is greater than this value.

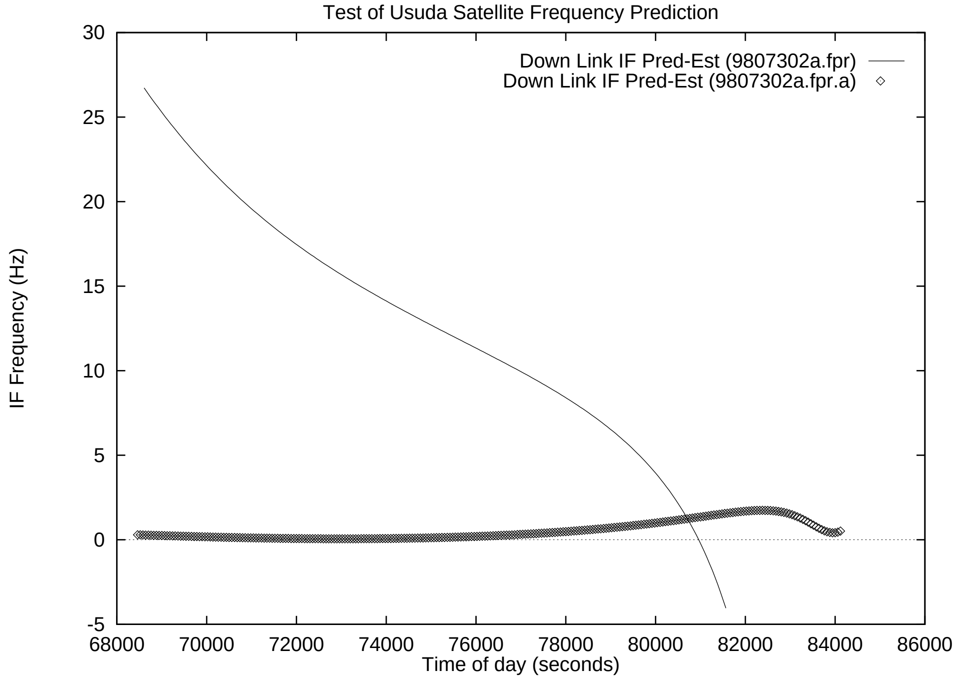

Figure 2: Tests of the ISAS Usuda-HALCA frequency predictions. The X-axis is the time of day (seconds) and the Y-axis is the difference between the down link frequency prediction and the estimated down link frequency based on the up link prediction. The solid line is a test based on ISAS prediction for Usuda tracking pass 9807302a. The thick line shows the difference based on the JPL HALCA prediction using `orbitUsuda`.

The new prediction appears to be generally more self-consistent that the original prediction, but an un-resolved question must be answered concerning the level of self-consistency required. Figures 2 and 3 show similar results as are shown in Figure 1.

It was possible that prediction inconsistencies caused the resulting Usuda TCFs to introduce large errors in the residual rate values. This was confirmed by creation of a TCF generation program which incorporates the differences between the ISAS prediction and the new predictions into calculation of the TCF values. The new TCF files have produced correlation results with small residual rates at the Penticton, Mitaka and Socorro correlators. There remains some questions about the relatively high clock offset values seen for some tracking passes. The new TCF program (`dt`) is the subject of OVLBI-ES memo 76.

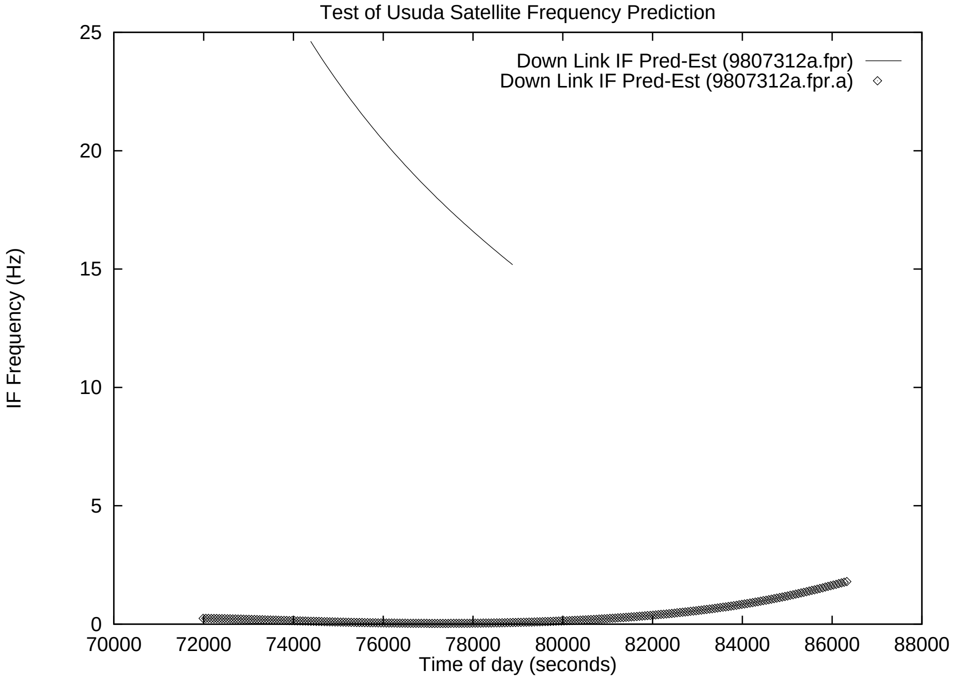

Figure 3: Tests of the ISAS Usuda-HALCA frequency predictions. The solid line is a test based on ISAS prediction for Usuda tracking pass 9807312a. The thick line shows the difference based on the JPL HALCA prediction using `orbitUsuda`*. The new prediction covers a larger time range that the original prediction.*

In 1999, the new predictions were used to control the Usuda station up link frequencies. Two of these tracking passes have yielded good residual rate results at the Socorro correlator. (The search continues for fringes from Usuda passes using the new predictions, when the closed loop mode was used for up link control).

Conclusions

A new program, `orbitUsuda` was written to calculate HALCA range and range rate to the Usuda tracking station based on the JPL Navigation orbit prediction data.

The smooth variation of the down link prediction and the estimated down link value based on the up link data suggest there is some small difference in the definition of these predictions. Ie. perhaps there is a difference in the definition of the satellite position such as the satellite ranges are the instantaneous (proper?) distance values.

The `orbitUsuda` program seems to generate results that are more self consistent than those generated by other methods. The new predictions were successfully used to correct frequency residual values measured during previous tracking passes and also during tracking passes in 1999. These tests do *not* check the absolute accuracy of either satellite range prediction.

Appendix A: orbitUsuda

The orbitUsuda program is very similar to a program used for the Green Bank Tracking station. The program uses the JPL NAIF software package (in fortran), and a set of "C" written in Green Bank and in Socorro, New Mexico.

The orbitUsuda program can read in a JPL Navigation prediction for any body, satellite or planet, and calculate the required records for the Usuda prediction. The program generates a dummy record for the orbital elements, because this information is not available in the JPL prediction.

The major part of the program written in Green Bank concerns calculation of the location of the tracking station in the J2000 coordinate system used in NAIF. This calculation requires knowledge of Earth rotation constants.

The program is different from the Green Bank version only in the hard coded coordinates of the tracking station (in an include file) and in a function to create the output file format. Both orbitUsuda and the Green Bank version require periodic maintainence, because the Earth rotation model must be updated for estimates of UT1-UTC and must include changes in the number of leap seconds. This maintainence needs to be done about once every six to twelve months. The Earth rotation model comes from the I.E.R.S. group via e-mail. Identical versions of this routine may be used in both Green Bank and Usuda programs.

The overall structure of the program orbitUsuda is described below including a summary of fortran and C modules used. The main program, orbitUsuda.f, is a fortran module, which calls many subroutines. The program orbitUsuda runs in a Sun Sparc Station environment, at Green Bank or at ISAS Sagamihara. I have complied and linked it successfully both at ISAS Sagamihara and in Green Bank. orbitUsuda uses pre-complied NAIF software libraries copied from Green Bank.

Like the prediction program used in Green Bank, it is somewhat difficult to correctly set the orbitUsuda input values. This problem is overcome by use of a C-shell script (called orbitVsop) which sets the orbitUsuda inputs for the specified JPL prediction. When run without arguments, orbitVsop prints out some documentation describing the input arguments, then locates the latest JPL Navigation prediction for HALCA (files ending with name ".bsp") and produces the Usuda frequency predictions base on that input file.

Sample output of the orbitVsop script is below:

```
% orbitVsop

orbitVsop: usage
orbitVsop <predictionFile> [<orbitFile>] [<startDate>] [<endDate>]
where <predictionFile> JPL NAIF binary SPK file name
 <orbitFile>           name of output file Usuda orbit file
 <startDate>           first date of orbit file to create
 <endDate>             last  date of orbit file to create
The latest NAIF files (*.bsp*) in current directory are:
-rw-rw-r--   1 glangsto vsop      1223680 Jan 26 21:22 990125-990202-p.bsp
-rw-rw-r--   1 glangsto vsop       804864 Jan 19 09:40 990118-990126-p.bsp
-rw-rw-r--   1 glangsto vsop       743424 Jan 15 20:55 990114-990122-p.bsp
```

```
-rw-rw-r--   1 glangsto vsop       819200 Jan 14 15:47 980723-980731-p.bsp
-rw-rw-r--   1 glangsto vsop       756736 Jan 12 10:02 990111-990119-p.bsp
Setting defaults

InFile = 990125-990202-p.bsp
Start, stop dates: 99Feb26             , 05DEC31
Start, stop times (radians): 0.000000, 2.617994
Orbiting body number       : -58
Tracking Downlink Position
Opening leapsecond kernel leapseconds.ker
Opening spk file name 990125-990202-p.bsp

Summary of spk file entries for body   -58

-------------------------------------------------------------------------------
Segment identifier:    VSOP Predicts P99025
Body       : -58                               Center    : 399
Reference  : DE-200                            SPK Type  : 1
From (UTC) : 1999 JAN 24 23:58:56  (Day 024)   From (ET) : -2.9505600000000E+07
To   (UTC) : 1999 FEB 01 23:58:56  (Day 032)   To   (ET) : -2.8814400000000E+07
-------------------------------------------------------------------------------
 START,STOP =   0  -2073664, INT =   1 UT1-UTC=  -0.44290077923226
```

Automatic Predictions

In Green Bank, Toney Minter wrote a Unix "cron job" which automatically retrieves the latest JPL Navigation predictions and produces Green Bank orbit range predictions.

At ISAS, a similar "cron job" runs on computer MAINOUMIto automatically create the Usuda - HALCA range predictions. All the executable programs, input data files and output predictions are located in the directory called "$BSPHOME" (currently /DATA/MAINOUMI_1/orbitUsuda). All the executable programs are kept in the directory $BSPHOME/bin. On MAINOUMI, a cron job runs the Unix script getVSOPbsp every hour to check whether a new HALCA prediction has been generated at JPL.

There are several steps to the process of determining whether a new file has been generated. The script or program executing this step is listed in parenthesis.

File list: Anonymously ftp into JPL computer `dots.jpl.nasa.gov` and gather a list of HALCA predictions. (`getVsopNewList`).

List check: For all recent predictions, check whether this prediction has already been retrieved. (`filterFileNames`).

Make List: Get all new predictions (`getVSOPbsp`).

Uncompress: For the most recent prediction, un-compress file (`getVSOPbsp`).

Predict Range: Calculate the Usuda station ranges as a function of time based on the JPL prediction. This is done for the entire JPL prediction time range, about one week. (`orbitVsop` and `orbitUsuda`).

Split: The Usuda operations assume that the range predictions for each pass is in a separate file. The single large file is split into small files and placed in directory $BSPHOME/in for input to the Usuda tracking station (`splitPrediction`).

E-mail: After a new file is processed an email is sent out indicating that new predictions are available (`getVSOPbsp`).

The crontab entry is below:

```
0 * * * * /DATA/MAINOUMI_1/orbitUsuda/bin/getVSOPbsp
```

Code structure

All programs developed for OVLBI in Green Bank used the SCCS version control system for keeping track of program versions. This facility has many good features, including the capability of re-creating any old version of a program by supplying a date to the “Check Out” command. I highly recommend this system for software development. The SCCS system is available as a standard part of the Sun Work stations environment. I used the SCCS system at ISAS and placed the files in SCCS directories ∼`glangsto/SCCS` and ∼`glangsto/offline/SCCS`.

All of the `orbitUsuda` fortran functions were written in Green Bank, many of the `orbitUsuda` C functions were taken from the VLBA antenna and correlator control project. Table 1 contains a list of all software modules used in the orbitUsuda program, and a short summary of the modules function.

Module Name	Function
orbitUsuda.f	Main routine
addnul.c	Fix Fortran/C string compatibility
cvrtuc.c	Convert string to upper case (VLBA)
extractName.c	Get file name from Space VLBI file name convention
gastm.c	Returns Greenwich Apparent Sidereal Time at Midnight (VLBA)
getnam.c	Returns ASCII string corresponding to UNIX environment variable
getutc.c	Read user input time ranges from UNIX environment variables
ll2xyz.c	Returns distance from LAT,LON,ELEV to center of Earth. (VLBA)
mjd2str.c	Converts a modified Julian day number to an ASCII string (VLBA)
nutate.c	Calculate nutation in longitude + obliquity for given time (VLBA)
obliq.c	Calculate MEAN obliquity at an arbitrary time, in radians (VLBA)
openOutRevision.c	Open a new version of file with .a etc extension
setCoordParms.c	Calculates X,Y,Z station values from LAT, LON, elevation
srclist.c	Look for sub-string in a longer string (VLBA)
str2mjd.c	Convert date string to modified Julian day number (VLBA)
str2rad.c	Convert time/angle string to radians (Ie "12h00m0s" = 3.14159) (VLBA)
stripWhite.c	Remove white characters from both ends of a string (VLBA)
time2str.c	Convert a radian value to time of day string (VLBA)
wusuda.c	Write out orbit prediction data
ztrlog.c	Read Unix environment values into a program
azel.f	Convert J2000 vector to antenna azimuth and elevation
cSpkez.f	Wrapper to allow a C subroutine to call a fortran subroutine
delUtc.f	Calculate the offset between UT1 and UTC
dispsp.f	Display summary information from a NAIF format prediction
fixtim.f	Convert from MJD, UTC to Ephemeris time
getcor.f	Calculate the matrix conversion from J2000 to Date coordinates
getgbp.f	Return position and velocity of station (J2000) at an input time
getlst.f	Calculate station LST from UTC and station location
getnut.f	Calculate Earth Nutation matrix for date
getpos.f	Return NAIF values for satellite position
getpre.f	Calculate the rotation angles between J2000 and Earth orientation
newton.f	Newton's method for calculating range and rates to satellite
nsetup.f	Process user input data; leap seconds Kernel file, etc
nutrot.f	Earth rotation matrix function
radec.f	Calculate Ra and Dec from J2000 position vector
spksum.f	Modified JPL NAIF software to print a prediction summary
vnorm.f	Utility to calculate unit vector from an arbitrary vector (VLBA)
MATHCNST.H	Standard mathematical and astrophysical quantities (VLBA)
MATH.I	Fortran include file with math constants.
STDDEFS.H	Standard data types used by VLBA (VLBA)
USUDACOORD.H	C include file containing Usuda coordinate information
USUDADIR.H	Definitions of output file directories
USUDA.I	Fortran include file containing Usuda coordinate information
vlb.h	Include file for some VLBA utility programs; (modified VLBA)
MakeOrbit	Make file, retrieves source from SCCS, compiles and links `orbitUsuda`