Introduction

RadioAstron has two modes of synchronization: 1) local synchronization with a H-maser reference and an on-board rubidium reference, and 2) "closed-loop" with synchronization using the H-maser references at Pushchino and Greenbank ground stations.

The H-maser reference was depleted in July 2017. This left the rubidium as the only on-board reference.

Astro Space Center (ASC) reports in RadioAstron Newsletter #33 that the synchronization mode with the on-board reference produced interferometric fringes, but best results were achieved in the closed-loop mode.

Closed-loop mode of synchronization is the default for AO5 and AO6 observations.

The ASC has a dedicated software correlator for RadioAstron experiments. The ASC correlator already implements a delay model that incorporates closed-loop mode. The software correlator is described by Likhachev et al. (2017).

The DiFX software correlator supports space VLBI ("RA-DiFX"). It too is used for correlating RadioAstron experiments. However, currently RA-DiFX does not support the closed-loop mode (01/2019).

Source code location

This page keeps track of implementing closed-loop support into the existing DiFX RadioAstron source code branch.

The current not yet modified RA-DiFX source code resides in https://svn.atnf.csiro.au/difx/master_tags/DiFX-RA-1.0.0/ and is installed in /cluster/difx/DiFX-RA-1.0.0/, selectable on the cluster with 'dra100'.

The new script for applying closed-loop corrections to DiFX/CALC IM files before the actual correlation run resides in https://svn.atnf.csiro.au/difx/sites/MPIfR/oneoff/raPatchClosedloop.py

Updates

22Nov2018 : Brief exchange with J. Anderson (RA-DiFX) on patching closed-loop delays into DiFX CALC model .im files

17Dec2018 : Test data set RK18CJ received at Bonn, no documentation

07Jan2019 : First look at raw data and .vex; created v2d and vex.obs; ground-only, no fringes yet in K nor C band; no station clock infos

15Jan2019 : Created 'raPatchClosedloop.py' script to apply closed-loop corrections to DiFX/CALC IM prior to running actual DiFX correlation job. Script in DiFX Trunk ./sites/MPIfR/oneoff/

25Jan2019 : Mikhail found RadioAstron fringes in the rubidium-mode scans! Had to set GT delays in VEX to zero, and include them (in units of usec) in .v2d clockfudge. Strong fringe in C-band (fringe_RA_Yebes_Cband_rubidium.pdf). For the closed-loop scans and poly, might need to combine zeroth order coeff of ASC with CALC, replace rest with the higher order coeffs from ASC.

28Jan2019 : Closed-loop fringes K-band first scan (fringe_RA_GBT_Kband_closedloop.pdf). Not as strong as anticipated. Not stable over time, strong fringe mainly in 75 sec towards end of first scan (fringe_RA_GBT_Kband_closedloop_sec600-675.pdf).

25Apr2019: Had a telecon with the ASC correllator on the rk18cj rubidium and closed-loop parts. From MPI were Jan Wagner and Misha Lisakov. From ASC was Andrey Andrianov.

09May2019: Revisited the dual-band VEX file. Shifted GBT (former ref ant) and other K-band ground stations +17.20 us. Got fringes now both K and C, both in coherent and in rubidum. Improved raPatchClosedLoop.py to allow a residual rate to be fed back to the polynomial. Tried the suggested -3.0e-10 sec/sec rate offset to no avail, rates already low before that.

16May2019: Fixed the delay jumps introduced by merging DiFX delay (0th order) with ASC coeffs (1st..Nth order) and an optional user delta-rate.

Now K-band closed loop fringe phase is much improved throughout the scan (fringe_09h15m_K_noDeltaRate.pdf with as-is ASC poly, fringe_09h15m_K_withDeltaRate.pdf with user-specified added residual '-r 2.4886e-06') and C-band has fringes too (fringe_09h15m_C_withDeltaRate.pdf). AIPS FRING solint=1;solsub=0.5 results in SNR>50 with a nanohertz rate residual that has a linear trend over the scan. With these solutions and a solint=2 fit the SNR is around 80.

Structure in fringe phase vs time (HOPS) and rate over time (AIPS) shows that the ASC poly set is indeed the initial poly set and needs iterative refinement, but currently it is unclear how to best to this. PIMA can fit delay, rate, acceleration, but: a) how to segment a scan into 60s pieces? b) how to fit higher-order terms?

20May2019 Compared uvw coordinates in AIPS UVPLT (ASC reference FITS, DiFX FITS where ASC uvw poly data ignored and CALC9 retained i.e. directly from ephemerides). The uvw coordinates do NOT match. Time against uv-dist overlaps nicely. However, coordinates are of opposite track direction / from the conjugate position.

20May2019 Implemented loading of PIMA-fitted residuals for rate and acceleration (1st, 2nd order). Resulted in an improvement in SNR~11 to SNR~24 in 600s. The latter HOPS fringe plot shows that not all of the phase wander could be compensated. Might have to find a way to fit the higher-order terms (jerk/3rd, 4th, 5th) that the priori polynomials from ASC do contain.

Test data set

Data are on the cluster under /data/rk18cj/ and /Exps2/rk18cj/

Raw data

/data/rk18cj/rdf/GBTS_20172920*.rdf	RadioAstron raw 4-channel 1-bit data in RDF(?) format
/data/rk18cj/ra/GBTS_20172920*.m5b	RadioAstron raw RDF converted to raw Mark5B
/data/rk18cj/gb/ NRAO+312_005?_RK18CJ_GB_No000?	GBT, 4-thread 1-channel 2-bit VDIF
/data/rk18cj/{ys,mc,ir,nt}/*	Yebes and others stations, 1-thread 4-channel 2-bit VDIF
/data/rk18cj/tr/*	Torun, unknown format

Extra files

README	"This is a RadioAstron experiment raks18cj to test the RadioAstron correlation for the closed-loop (coherent mode) observations."
filelist_RA_v2d.txt	spacecraft / ground station clock breaks in undocumented notation
rdf/ GBTS_2017_292_rk18cj.txt	RDF file 1-bit statistics in 4-channels for all four scans
RA_C_COH.TXT	RadioAstron delay polynomials for each scan
RA_C_COH_uvw.txt	RadioAstron polynomials for u, v, and w

Missing information:

1. reference FITS visibility data set, to be able to compare ASC results against DiFX
2. clocks and rates used for the ground stations in the ASC reference correlation
3. EOPs used in the ASC reference correlation
4. details on the polynomials; is RA_C_COH.TXT in seconds, does it include ground-to-space delay?

Correlation details

Details of DiFX correlation are below.

Scans

Source	ra1cm2	ra6cm2	RadioAstron reference
0716+714 @ 292-0900	GT GB M TR	GT IR NT YS	free-running
0716+714 @ 292-0915	GT GB M TR	GT IR NT YS	free-running
0716+714 @ 292-0930	GT GB M TR	GT IR NT YS	onboard rubidium
0716+714 @ 292-0945	GT GB M TR	GT IR NT YS	onboard rubidium

Freq setup of RA: four channels, all are LCP polarization, frequencies and sidebands as below:

```
def RA4Freq;
     sample_rate =  32.000 Ms/sec;  * (1bits/sample)
     chan_def = :  4836.00 MHz : U :  16.00 MHz : &CH01 : &BBC01 :
&NoCal; *Lcp
     chan_def = :  4836.00 MHz : L :  16.00 MHz : &CH02 : &BBC01 :
&NoCal; *Lcp
     chan_def = : 22236.00 MHz : U :  16.00 MHz : &CH03 : &BBC03 :
&NoCal; *Lcp
     chan_def = : 22236.00 MHz : L :  16.00 MHz : &CH04 : &BBC03 :
&NoCal; *Lcp
enddef;
```

Delays for GT/RA: Mikhail found that delays in earlier VEX attempts should be set to zero in the VEX, and instead have to be included in the .v2d under the clockfudge entries in units of microseconds:

```
SC_GS_clock_break=start@2017y292d07h56m06s/sync@2017y292d07h56m06s/
clockfudge@0.0 # To do
SC_GS_clock_break=start@2017y292d09h00m00s/sync@2017y292d09h00m00s/
clockfudge@0.0 # To do, closed loop
SC_GS_clock_break=start@2017y292d09h15m00s/sync@2017y292d09h15m00s/
clockfudge@0.0 # To do, closed loop
SC_GS_clock_break=start@2017y292d09h30m00s/sync@2017y292d09h30m00s/
clockfudge@-3.30469 # Fringes, rubidium
SC_GS_clock_break=start@2017y292d09h45m00s/sync@2017y292d09h45m00s/
clockfudge@-3.25000 # Fringes, rubidium
```

Status

Station	Format	Delay	Rate	Notes
Noto @6cm	VDIF/8032	fs log "gps-fmout/-48.48e-06" fringe: -37.48us	?	ground fringes
Yebes @6cm	VDIF/8032	fringe: +6.0us	?	ground fringes
Irbene @6cm	VDIF/8032	fringe: 0.0us (ref)	?	ground fringes
RadioAstron/ GBTS @6cm	Mark5B-128-4-1	?	?	6cm fringes in rubidium scan
GBT @1.3cm	(automatic, multi-threaded VDIF)	fringe: 0.0us (ref)	?	ground fringes
Medicina @1.3cm	VDIF/8032	fringe: -21.0us	?	ground fringes
Torun @1.3cm	MKIV1_4-256-4-2	fringe: -13.5us	?	ground fringes, in RR but not in LL

Delay polynomials

Produced by DiFX calcif2 from .calc --> .im:

1. starts at integer second
2. defaults to 5th order, 6 coefficients
3. coeffs seem to be given in order of a0, a1, a2, a3, a4, a5 (for p(t) = a0 + a1*t + a2*t^2 + ...)
4. .im SCAN 0 NUM POLY: 9 total number of coefficients sets over the scan
5. .im SCAN 0 POLY <0..8> MJD: 58045 valid from MJD

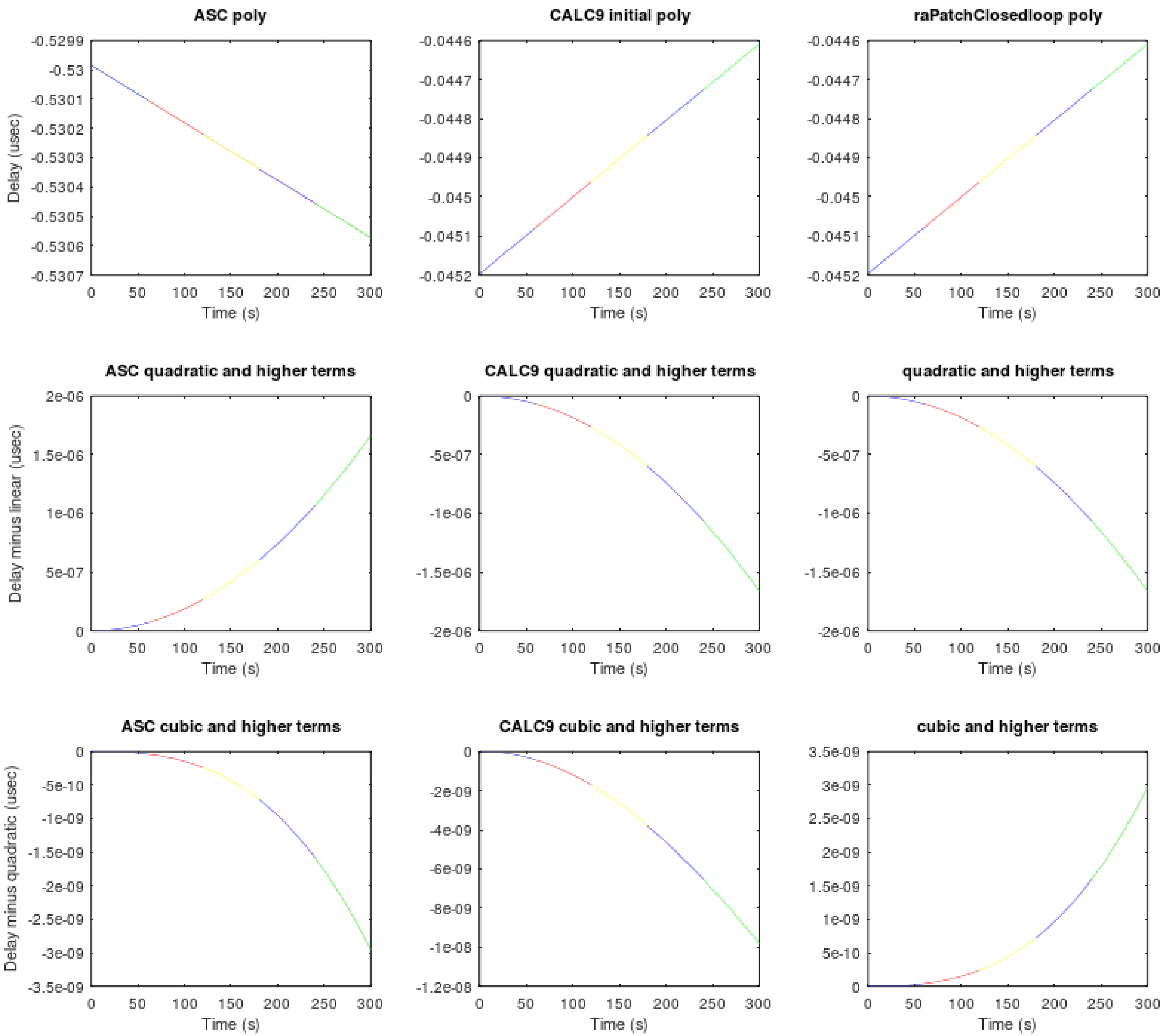
6. .im SCAN 0 POLY <0..8> SEC: 32400 valid from second-of-day
7. .im SRC 0 ANT <n> DELAY (us): coefficients for antenna n
8. .im SRC 0 ANT <n> U (m): coefficients for antenna n; similarly for V (m) and W (m)

DiFX calcif2 divides the day (sec 0...86399) into fixed intervals. The intervals start from second 0 of the day. In non-RA DiFX Trunk the length of each interval is 120 seconds. VEX scan start times do not affect the placement of the polynomial intervals. In other words, polynomials are valid in 2-minute intervals, and these are not necessary aligned with the VEX scans.

Provided by ASC in RA_C_COH.txt and RA_C_COH_uvw.txt:

1. starts at integer second, valid over a start/stop time region
2. .txt start = 19/10/2017 09h00m00s stop = 19/10/2017 09h01m00s
3. defaults to "6th order" (.txt: order = 6), 6 coefficients given
4. coeffs seem to be given in order of a0, a1, ..., a5 just like in DiFX .im file
5. .txt P0 = -5.28201618845368e-001 ... P5 = -1.67944819726498e-019
6. delays are given in seconds (vs DiFX/calc in microseconds), sign might differ from DiFX
7. polynomial coeffs are for direct replacement of CALC .im coeffs; "This are final values of the delay model, accounting for additives (but without accounting for the signal travel time from the satellite to the tracking station - this one should be [or should not be, if DiFX calculates it] be added to the first term of the polynomial)"

An illustration of evaluated ASC and CALC9 delay polynomials is given below for 5 consecutive segments. The figure can be re-generated in Matlab / Octave with dlyPolyCompare.m.



Essential notes from the ASC correlator

Based on the telecon 2019-Apr-25

1. ASC correlator calculates a fringe SNR as the fringe amplitude divided by the noise spread, max - min, not the rms. Hence, SNR should be higher with MPIfR DiFX than that from the ASC correaltor.

UPD (ML, 2020-11-11): But, surprisingly, ASC Corr sets higher SNR cutoff for detections -- around 11.4 in their units.

2. Station coordinates in the IDI FITS file are wrong. This is a bug of a convertor. But the correlation is done with proper parameters taken from the .vex file (calculated to the date of observation). The polynomials for the closed-loop more are also calculated with correct parameters.

3. A difference in the EOP parameters is negligible and could be due to different versions of EOPs used. But the EOPs are downloaded from the same place as we do.

4. ASC correlator refers delay and rate offsets measurements to the beginning of a scan.

5. For the Rubidium mode correlation, a constant Rate offset should be introduced: d_rate = -3 * 10e-10 sec/sec **(sign may be oposite, check which works for us.)**

6. Both delay offset and rate offset should be added to the ASC closed-loop polynomials. This contradicts with previous strategy.

SPECIFIC COMMENTS ON RK18CJ

1. Rubidium

ASC correlator found fringes at both 6 and 1 cm. MPIfR yet succeeded only at 6 cm.

- At 6 cm, ASC and MPIfR results agree., both SNR and delay RA-YS.
- At 1 cm, we will make another try to transfer 6 cm delay|rate values to 1 cm. Otherwise, no issues were found

2. Closed loop (Coherent) mode

ASC correlator found fringes at both 6 and 1 cm. MPIfR yet succeeded only at 1 cm.

- At 1 cm, SNR @ MPIfR is really lower somewhy. ASC had not introduced any a priori delay offset to the GBT (which we know to be ~17 musec). Hence our 3 musec delay offset agrees with their 20 musec offset.
- The procedure in ASC for producing polynomials is the following:
    - Take a first order guess of the polynomial
    - correlate, find a fringe, add delay|rate offsets to a setup file (like we do with a vex or v2d file for DiFX)
    - recalculate polynomials with offsets taken into account
- Hence, polynomial coefficients should be valid for the offsets that were used by the ASC correlator (~~For real observations, we will never have a recalculated poly. Need to perform the following test: take first order guess poly for rk12cj from ASC and try to correlate with it.~~ That was an initial poly wich we got from ASC.)

Details on implementation in DiFX

The normal utilities in DiFX are used to set up a RA correlation job.

An initial CALC9 delay model is produced by running calcif3 as usual. This initial model can then be patched to contain the replacement delay & uvw polynomial data supplied by ASC. Patching is done with https://svn.atnf.csiro.au/difx/sites/MPIfR/oneoff/raPatchClosedloop.py which takes as input the original calcif3 .im file, the ASC poly text files, and some optional parameters, and produces a new file .im.closedloop.

```
$ calcif3 --allow-neg-delay rk18cj_1.input
$ raPatchClosedloop.py RA_C_COH.TXT rk18cj_1.im
```

The raPatchClosedLoop.py script reads the input .im file and detects all delay polynomial blocks for the spacecraft. Some consistency checks are included to make sure the ASC polynomial files and the CALC9 .im are compatible (poly start time, time interval). The .im polynomial blocks are then updated such that:

1. scan start initial delay (0th order term) is taken from the original .im file as predicted by CALC9 based upon V2D & VEX
2. intra-scan poly segment delays (0th order terms) are recomputed based on (1) and the cumulative sum of delay changes over previous poly segments, this ensures time continuity on polynomial segment boundaries
3. delay rate (1st order term) and higher-order poly coefficients are taken from the ASC files, *not* from VEX or V2D

To use the new .im.closedloop file in a DiFX correlation run:

```
$ mv rk18cj_1.im rk18cj_1.im.openloop
$ cp rk18cj_1.im.closedloop rk18cj_1.im

$ startdifx --dont-calc rk18cj_1.input
```

The startdifx option --dont-calc is necessary to avoid potentially overwriting the closed-loop -patched .im file.

The output of correlation can be processed as usual.

If fringe SNR or residual phase behaviour needs to be improved across the duration of scans, the a priori delay polynomial can be refined via fringe fitting in PIMA. After an initial DiFX correlation with just the a priori polynomial data the resulting visibility data can be difx2fits exported (in one band) to a FITS file which then is fringe fitted using PIMA.

The PIMA config file needs to have "*MAX_SCAN_LEN: 60*" in order for the fringe fit results to be consistent with the 60 second interval of CALC9 and ASC delay polynomial segments.

The resultant PIMA .fri file and its delay rate and acceleration residuals can be applied to the ASC a priori polynomial text file to produce a next iteration polynomial text file that can be used in a recorrelation.

```
$ difx2fits --override-version -x -d --clock-merge-mode drop
Kband_iter0.fits

$ pima_fringe.csh rk18cj K load
$ pima_fringe.csh rk18cj K fine

$ raPatchClosedloop.py -P rk18cj_k.fri RA_C_COH.TXT
Wrote RA_C_COH.TXT.rev1

$ raPatchClosedloop.py RA_C_COH.TXT.rev1 rk18cj_2.im
$ mv rk18cj_2.im rk18cj_2.im.openloop ; cp rk18cj_2.im.closedloop
rk18cj_2.im
$ startdifx --dont-calc rk18cj_2.input
```

```
$ difx2fits --override-version -x -d --clock-merge-mode drop
Kband_iter1.fits
```

For the iterations it is sufficient to have just Radioastron and GBT/Yebes/other reference station in the correlation runs. Once the residuals for Radioastron look good enough a full correlation with all the stations can be carried out.

Real-life use experience

So far, closed-loop correlation according to the procedure described above was performed on two experiments: GS042 and GG083B. The detailed procedure can be found on the experiment pages.

Several convenience scripts were written to facilitate and speed up file handling procedures. Those are available in /Exps2/gg083b/v1

One particular caveat was found:

in the .v2d file, the clocks are defined by lines like this one:

```
SC_GS_clock_break=start@2018y015d14h09m49s/
sync@2018y015d14h10m11s/clockfudge@8.33
```

The first datetime value `2018y015d14h09m49s should be set to the scan beginning (as in .vex file). This is the time of a clock break.`

`The second datetime value 2018y015d14h10m11s` is printed out by a `convert_RDF_to_M5B`.py script. There is no firm understanding of why this value should be like that. But if changed even by a second, the fringe may disappear.

Hence, do not change it.