Rubidium clock drift correction for HK timing system

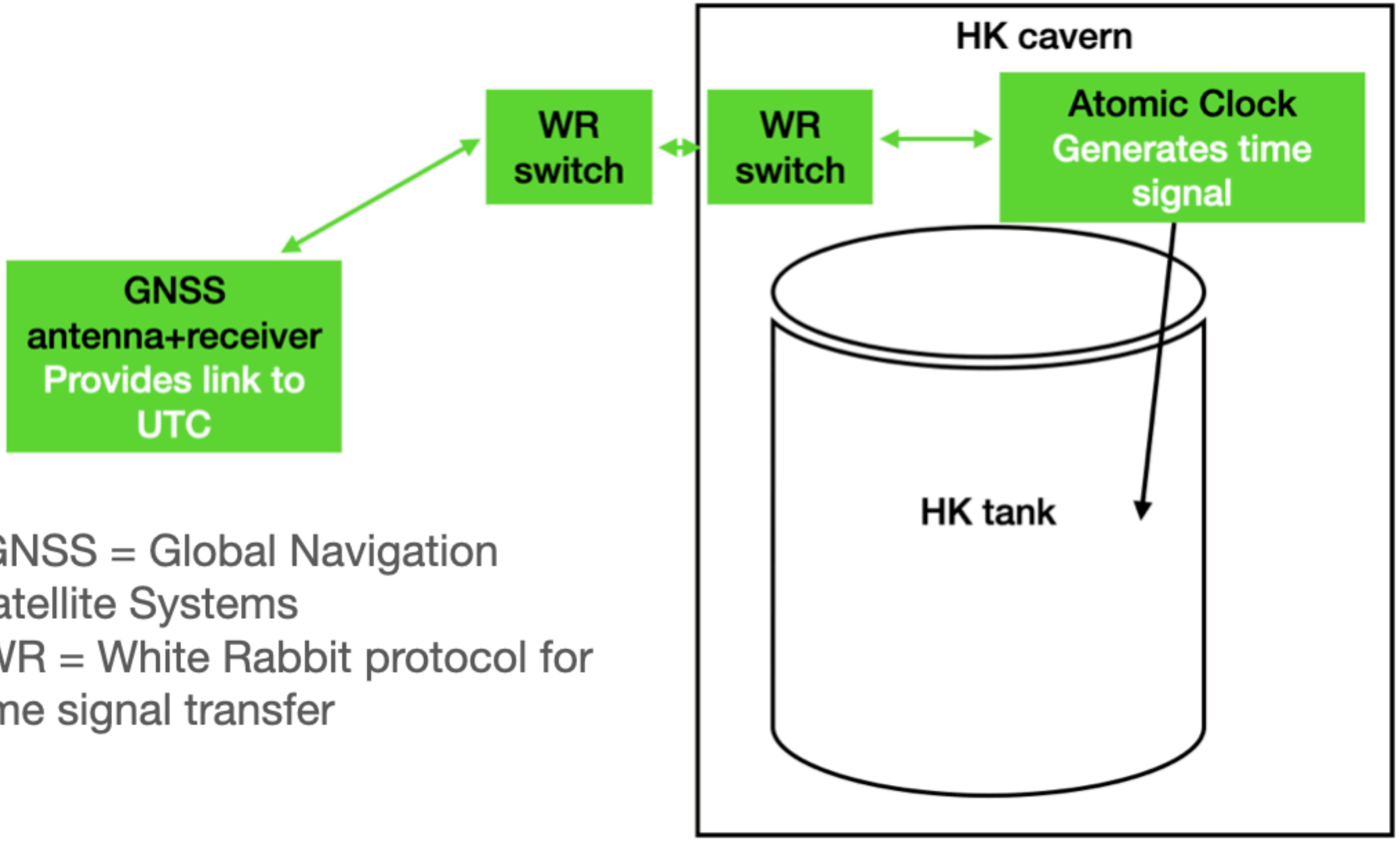
Claire Dalmazzone, 12th February 2025

Contents

- Timing in HK
- Rubidium clock drift correction
- Real-time implementation
- Validation on data

Time generation system

- Signal generated by a free running Rubidium atomic clock
- GNSS receiver continuously measures difference Rb - GNSS Time
- Measured difference used to correct Rb signal in order to keep a synchronisation below 100ns.



*GNSS = Global Navigation Satellite Systems
*WR = White Rabbit protocol for time signal transfer

Foreseen setup for HK

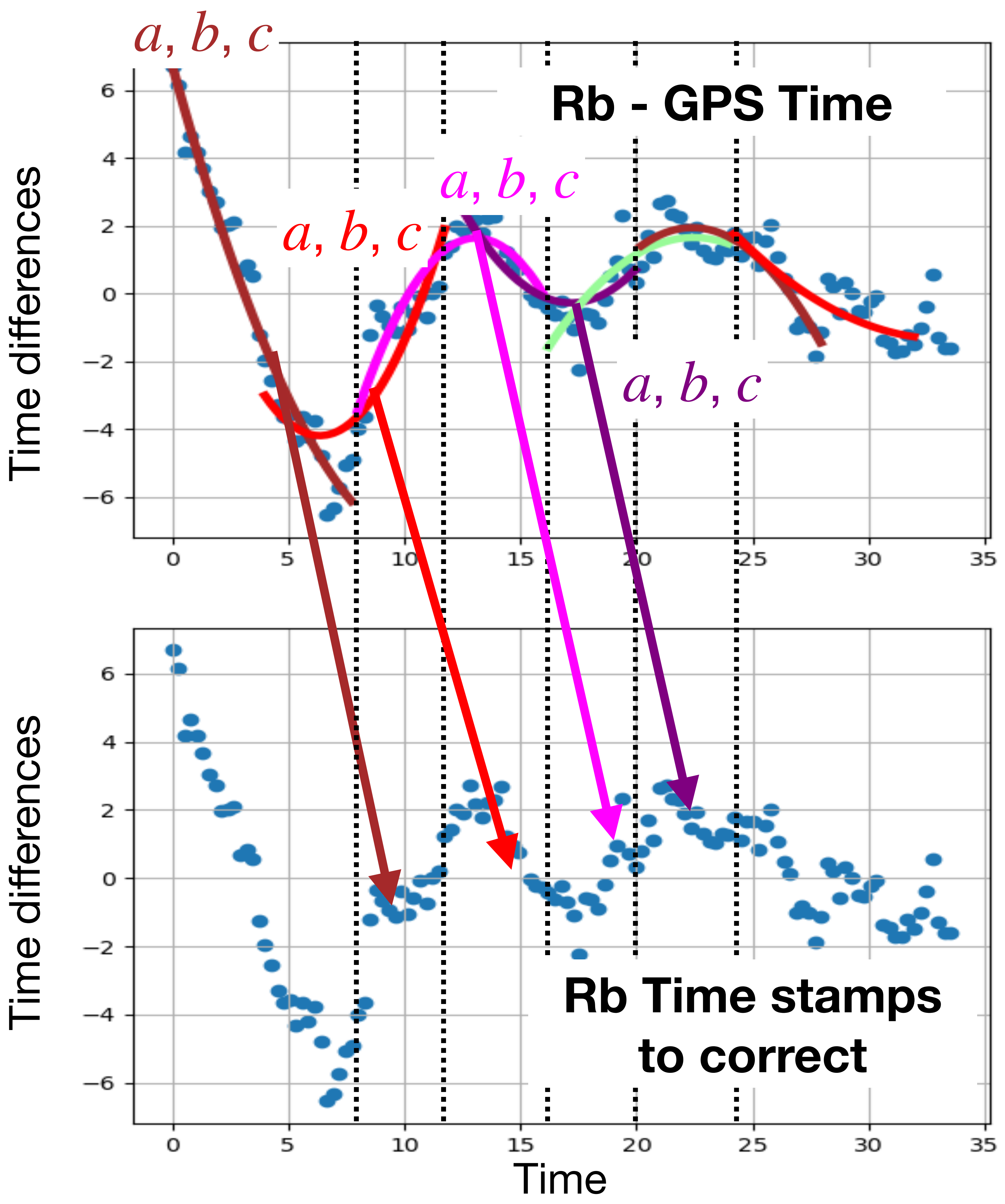
Rubidium clock drift correction

Drift correction

Hyper-Kamiokande

- Rubidium atomic clocks have a good short term stability but their frequency drifts with time
- The comparison to GNSS allows to estimate this drift and correct it
- To correct it in real-time, we can extrapolate the measured drift to the near future
- In practice, we receive on measurement every 16minutes. The correction to apply is thus changed every 16 minutes.

Online correction



Drift correction

- Fit the Rb - GNSS Time measured by Septentrio receiver with piece-wise polynomial functions of time:

$$\forall t \in [t_k, t_k + \Delta t],\ t_{Rb} - t_{GNSS} = b_k \cdot t + c_k$$

- Correct Rb time signal by subtracting the fit result extrapolated t the near future:

$$\forall t \in\ ]t_k + \Delta t, t_k + \Delta t + \boxed{\delta t}],\ t_{Rb,corr} = t_{Rb} - (b_k \cdot t + c_k)$$

Determines frequency at which we recalculate the correction coefficients: should be as small as possible for better efficiency (one Septentrio epoch ~16min).

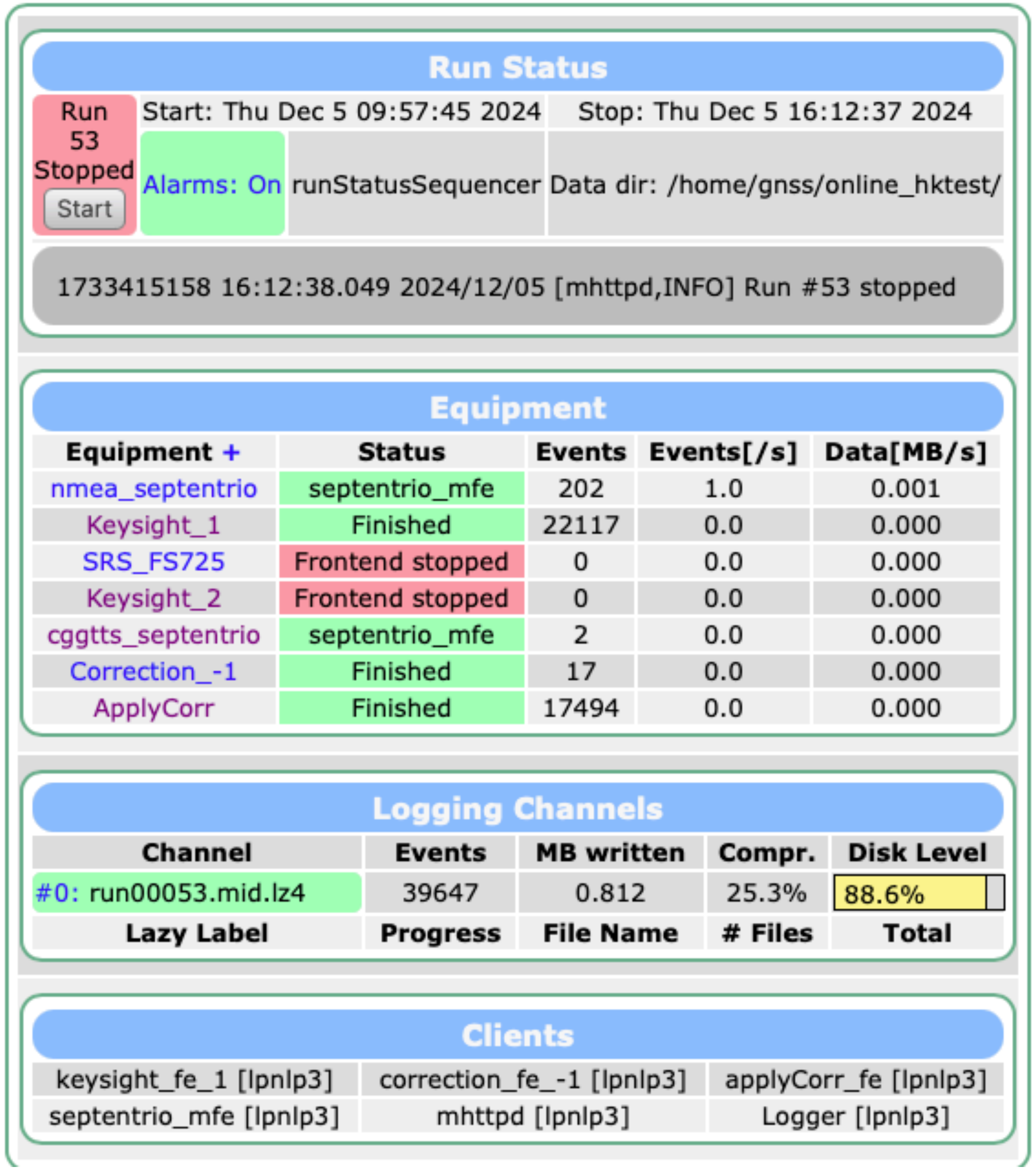
Real-time implementation

Midas

- Currently using Midas for our data-takings
- Midas is a data acquisition system, used by T2K ND280 for instance
- It works with frontends (programs written in python or C++) that can be launched via a webpage, and an online database (ODB) containing all information related to the internal operation of the data acquisition and any user information related to the configuration of the experiment.
- The frontend programs can access and edit the ODB during a run. Each time it does so, this is also saved in the midas file written by the corresponding run.

Correction

- Done with the **Correction** Frontend, script correction_fe.py



Run Status

Run 53 Stopped Start	Start: Thu Dec 5 09:57:45 2024		Stop: Thu Dec 5 16:12:37 2024
	Alarms: On	runStatusSequencer	Data dir: /home/gnss/online_hktest/

1733415158 16:12:38.049 2024/12/05 [mhttpd,INFO] Run #53 stopped

Equipment

Equipment +	Status	Events	Events[/s]	Data[MB/s]
nmea_septentrio	septentrio_mfe	202	1.0	0.001
Keysight_1	Finished	22117	0.0	0.000
SRS_FS725	Frontend stopped	0	0.0	0.000
Keysight_2	Frontend stopped	0	0.0	0.000
cggtts_septentrio	septentrio_mfe	2	0.0	0.000
Correction_-1	Finished	17	0.0	0.000
ApplyCorr	Finished	17494	0.0	0.000

Logging Channels

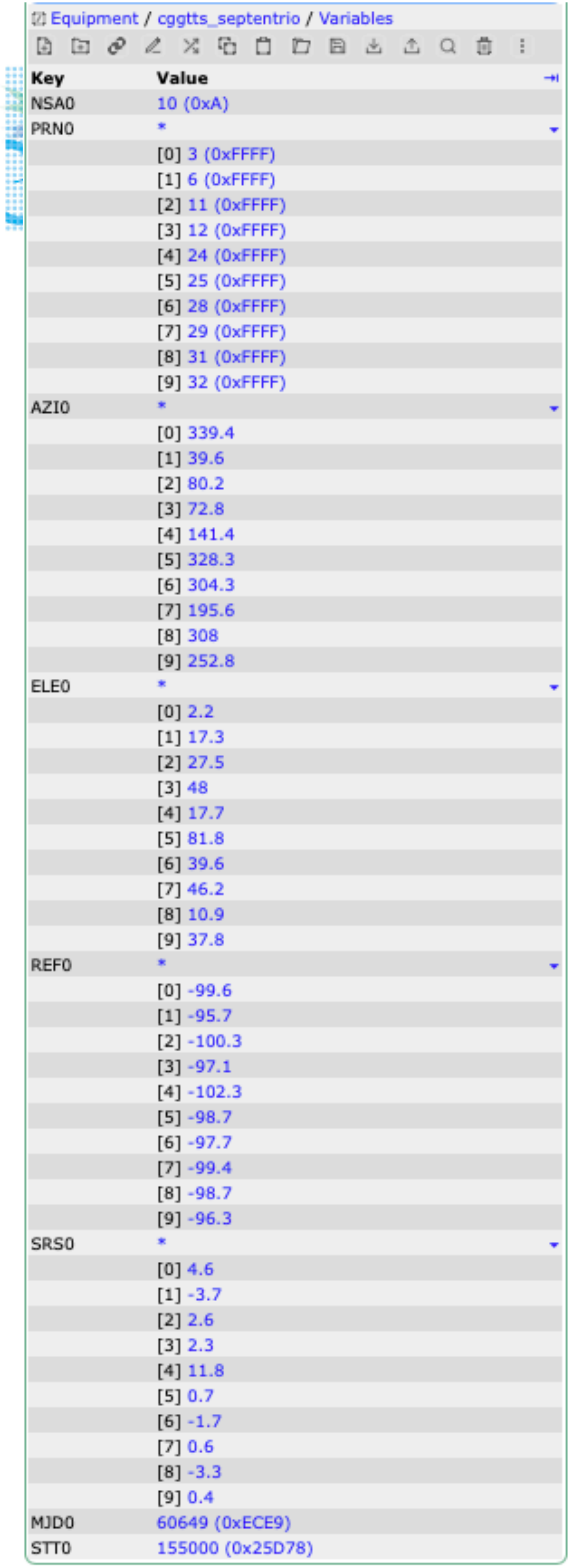
Channel	Events	MB written	Compr.	Disk Level
#0: run00053.mid.lz4	39647	0.812	25.3%	88.6%
Lazy Label	Progress	File Name	# Files	Total

Clients

keysight_fe_1 [lpnlp3]	correction_fe_-1 [lpnlp3]	applyCorr_fe [lpnlp3]
septentrio_mfe [lpnlp3]	mhttpd [lpnlp3]	Logger [lpnlp3]

Correction

- Done with the **Correction** Frontend, script correction_fe.py
- It continuously reads the ODB of the **cggtts_septentrio** frontend.
- Every time a new measurement is available, it updates the correction coefficients

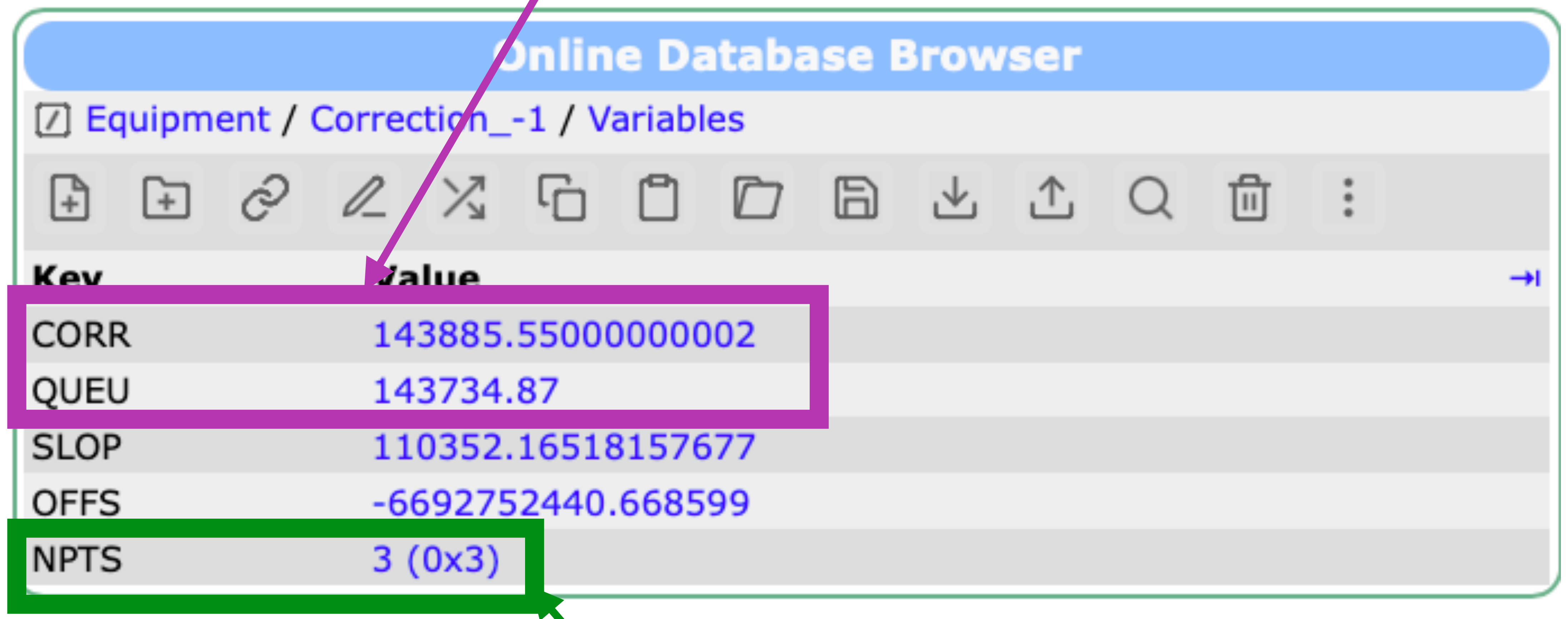


Correction

Hyper-Kamiokande

- Done with the **Correction** Frontend, script correction_fe.py
- It continuously reads the ODB of the **cggtts_septentrio** frontend.
- Every time a new measurement is available, it updates the correction coefficients
- Stores the coefficients in its ODB

Should be deleted



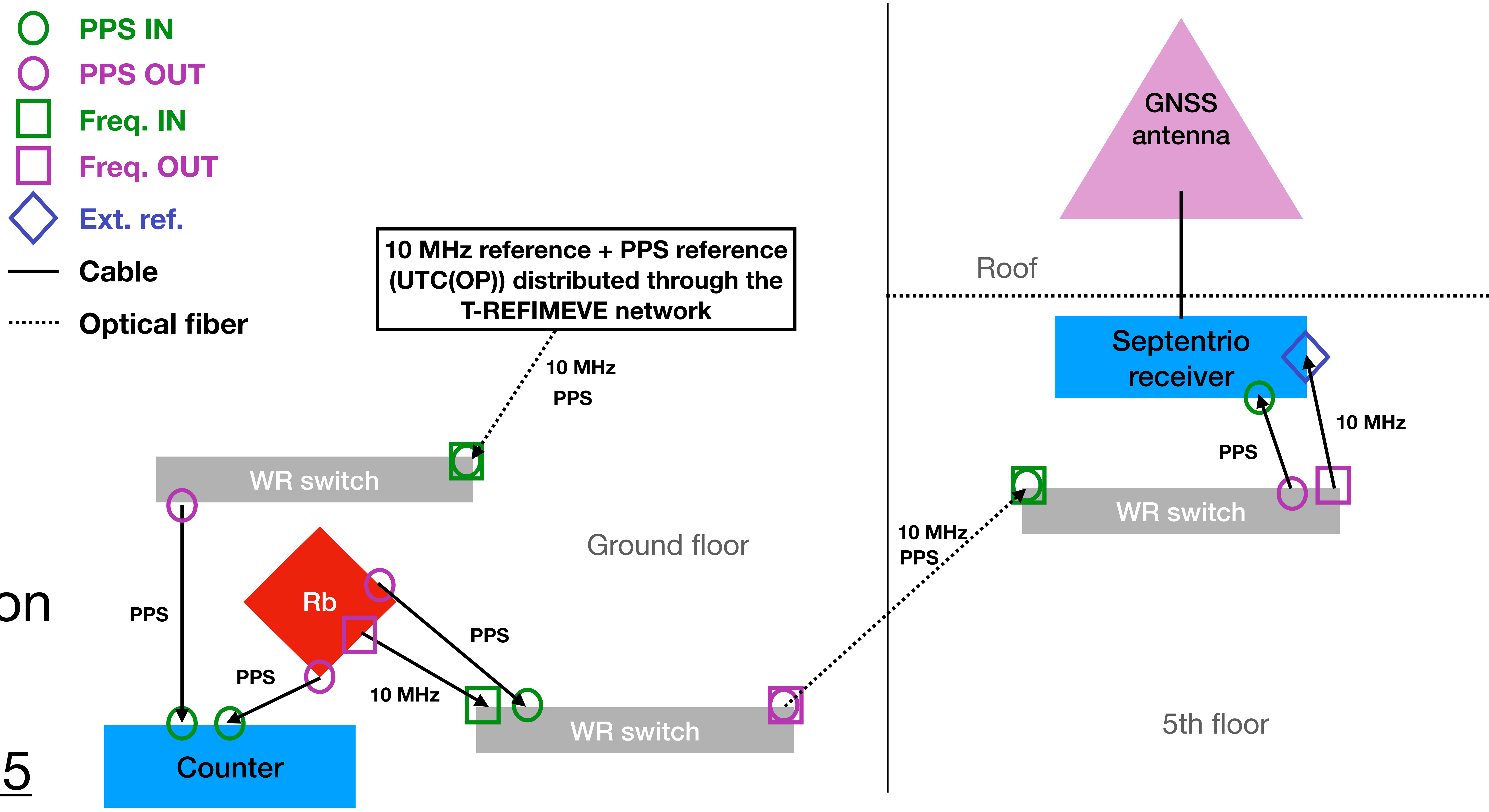
Number of Septentrio measurements to use.
Should be in Settings, not Variables…

Validation on data

Validation on data

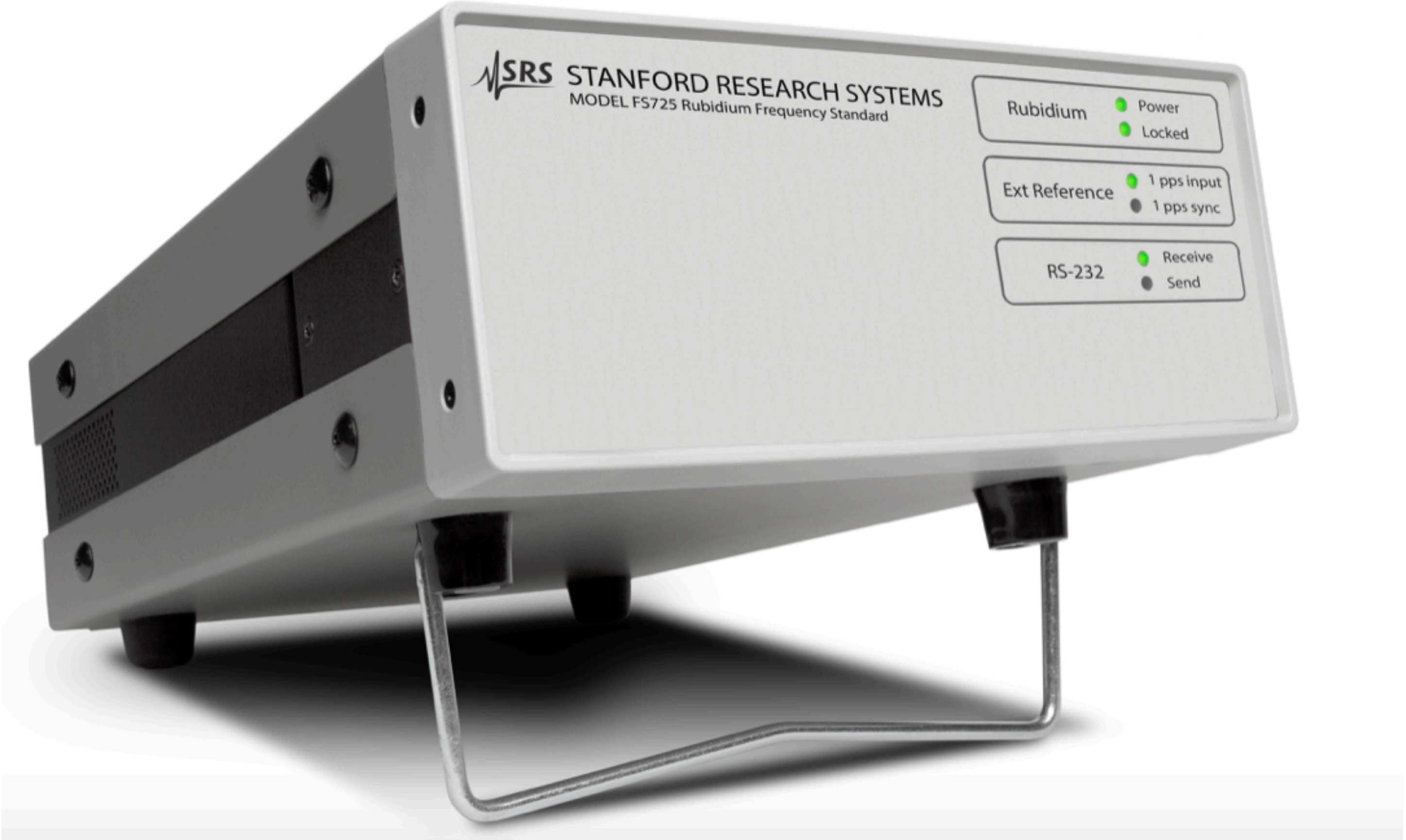
- Simultaneous Rb - GPS Time and Rb - UTC(OP) measurement
- Use Rb - GPS Time to extract correction
- Apply correction to both measurements
- Check residual differences and Allan Standard Deviation
- Results of the offline validation: arXiv:2407.20825

*UTC(OP): French official realisation of UTC made by SYRTE lab (3km from LPNHE)



Setup@LPNHE Paris

Validation on data

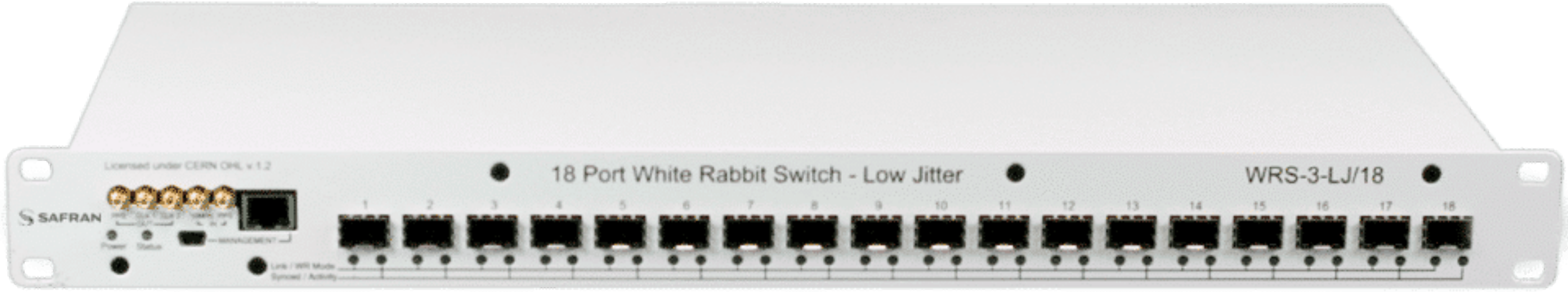


Rubidium atomic clock: FS 725 from Stanford Research System

GNSS receiver: Septentrio PolarRx5

Antenna: Septentrio PolaNt Choke Ring

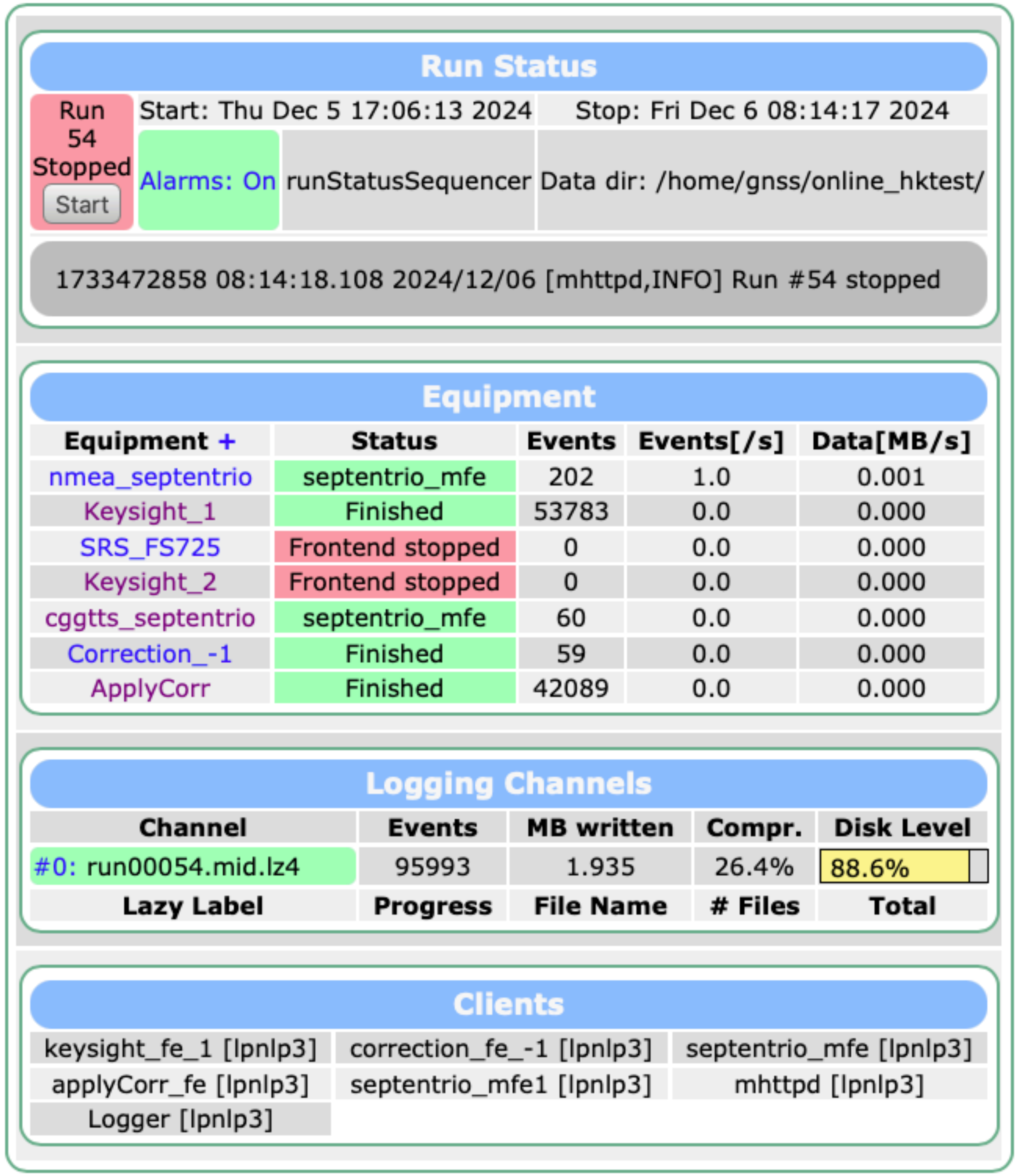
Counter: Keysight 53220a



White Rabbit Switch

Correction test

- To test the correction, added a **ApplyCorr** frontend that applies it to a time signal measured by a **Keysight** frontend



Run Status

Run 54 Stopped Start	Start: Thu Dec 5 17:06:13 2024		Stop: Fri Dec 6 08:14:17 2024
	Alarms: On	runStatusSequencer	Data dir: /home/gnss/online_hktest/

1733472858 08:14:18.108 2024/12/06 [mhttpd,INFO] Run #54 stopped

Equipment

Equipment +	Status	Events	Events[/s]	Data[MB/s]
nmea_septentrio	septentrio_mfe	202	1.0	0.001
Keysight_1	Finished	53783	0.0	0.000
SRS_FS725	Frontend stopped	0	0.0	0.000
Keysight_2	Frontend stopped	0	0.0	0.000
cggtts_septentrio	septentrio_mfe	60	0.0	0.000
Correction_-1	Finished	59	0.0	0.000
ApplyCorr	Finished	42089	0.0	0.000

Logging Channels

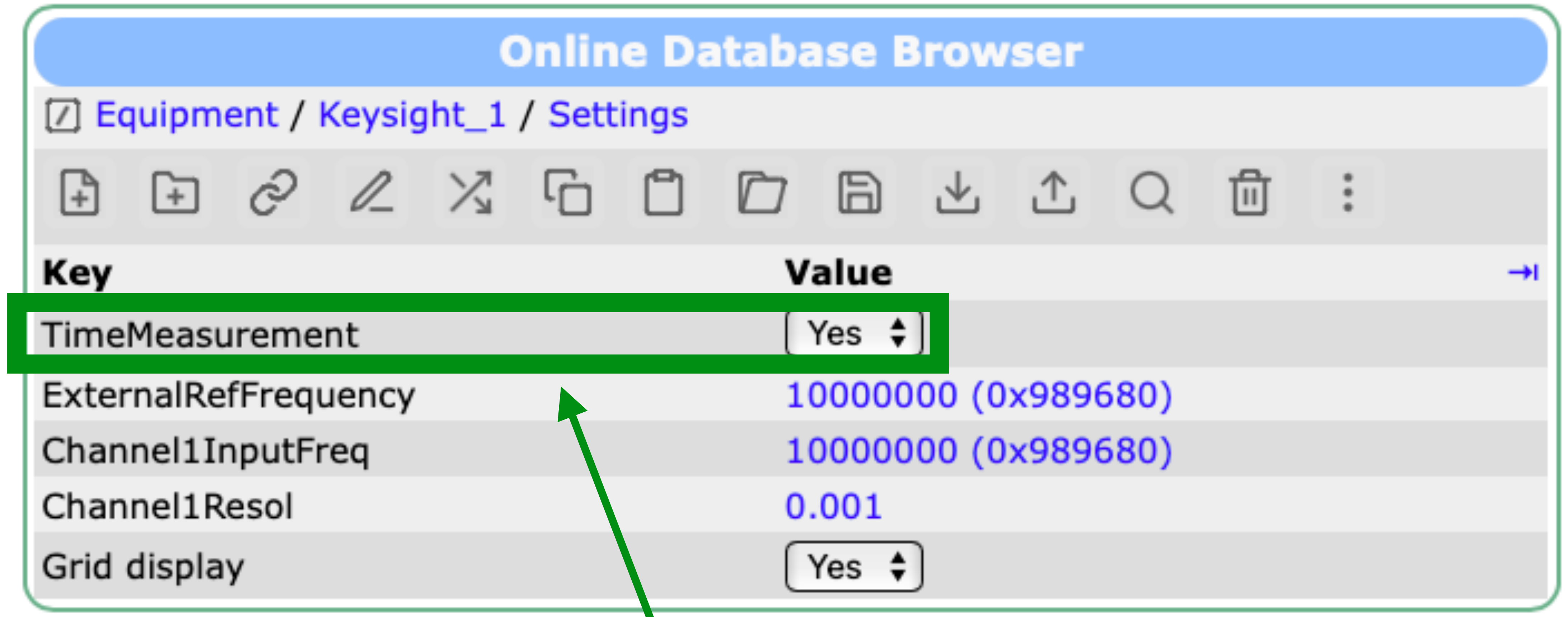
Channel	Events	MB written	Compr.	Disk Level
#0: run00054.mid.lz4	95993	1.935	26.4%	88.6%
Lazy Label	Progress	File Name	# Files	Total

Clients

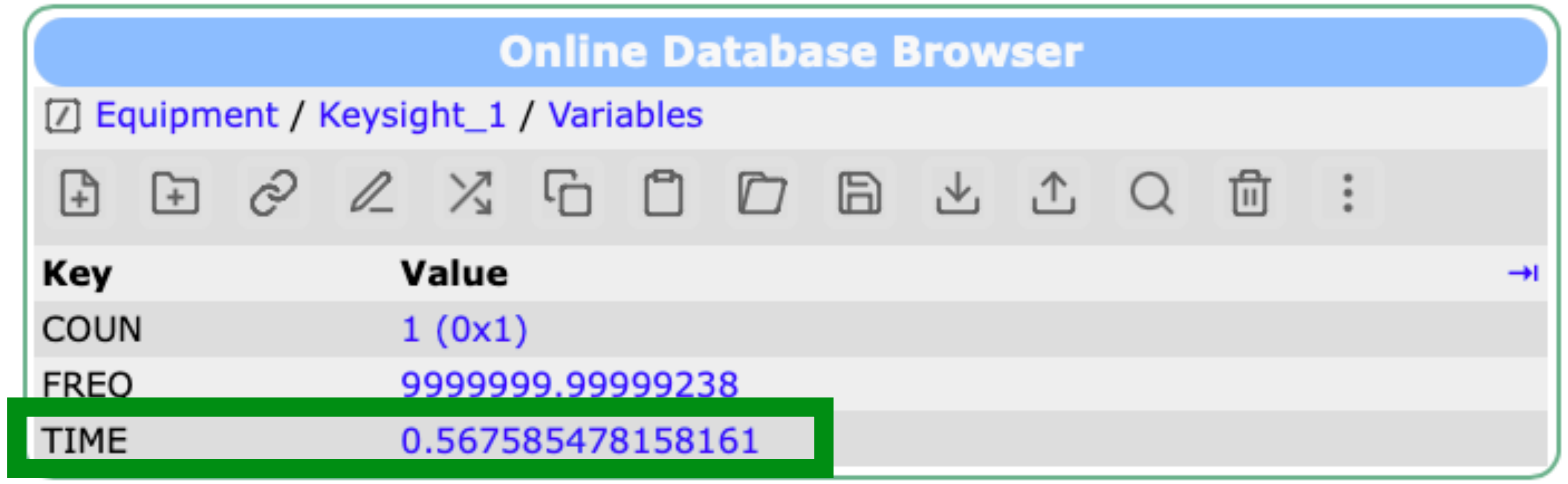
keysight_fe_1 [lpnlp3]	correction_fe_-1 [lpnlp3]	septentrio_mfe [lpnlp3]
applyCorr_fe [lpnlp3]	septentrio_mfe1 [lpnlp3]	mhttpd [lpnlp3]
Logger [lpnlp3]		

Correction test

- To test the correction, added a **ApplyCorr** frontend that applies it to a time signal
- It continuously reads the ODB of the **Keysight_1(2)** frontend.



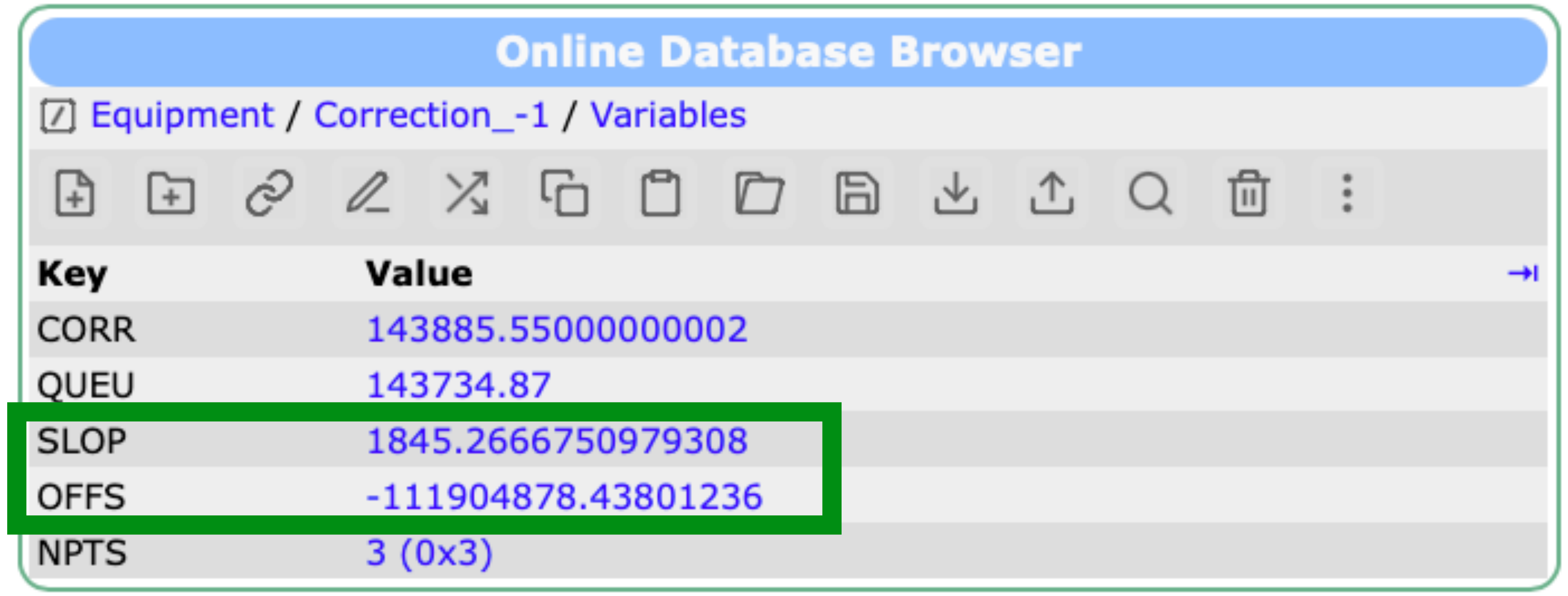
Checks if it is a Time measurement

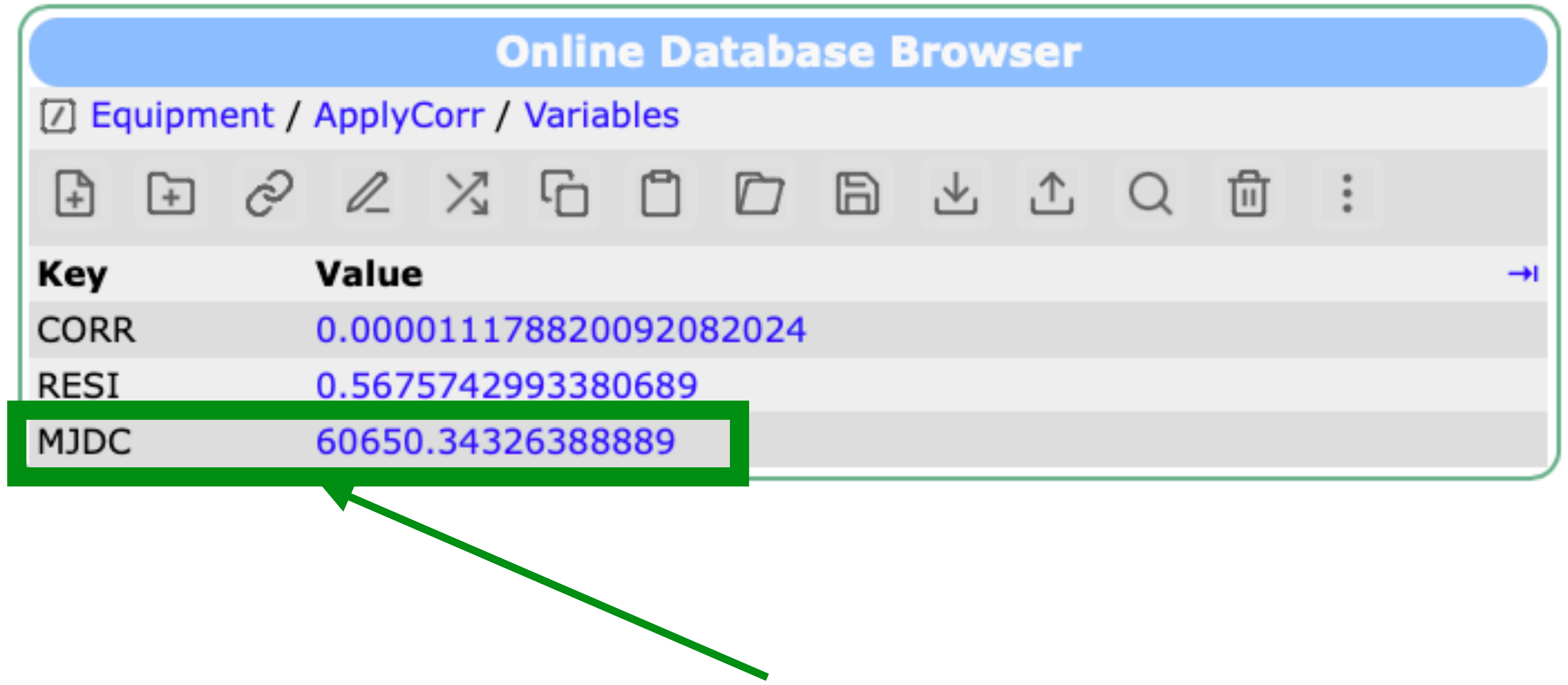


Reads the TIME value only

Correction test

- To test the correction, added a **ApplyCorr** frontend that applies it to a time signal
- It continuously reads the ODB of the **Keysight_1(2)** frontend.
- Every time a new **Keysight** measurement is available, it reads the current value of the correction coefficient in the **Correction** ODB and stores both the computed correction and the residual (key sight measurement - correction)

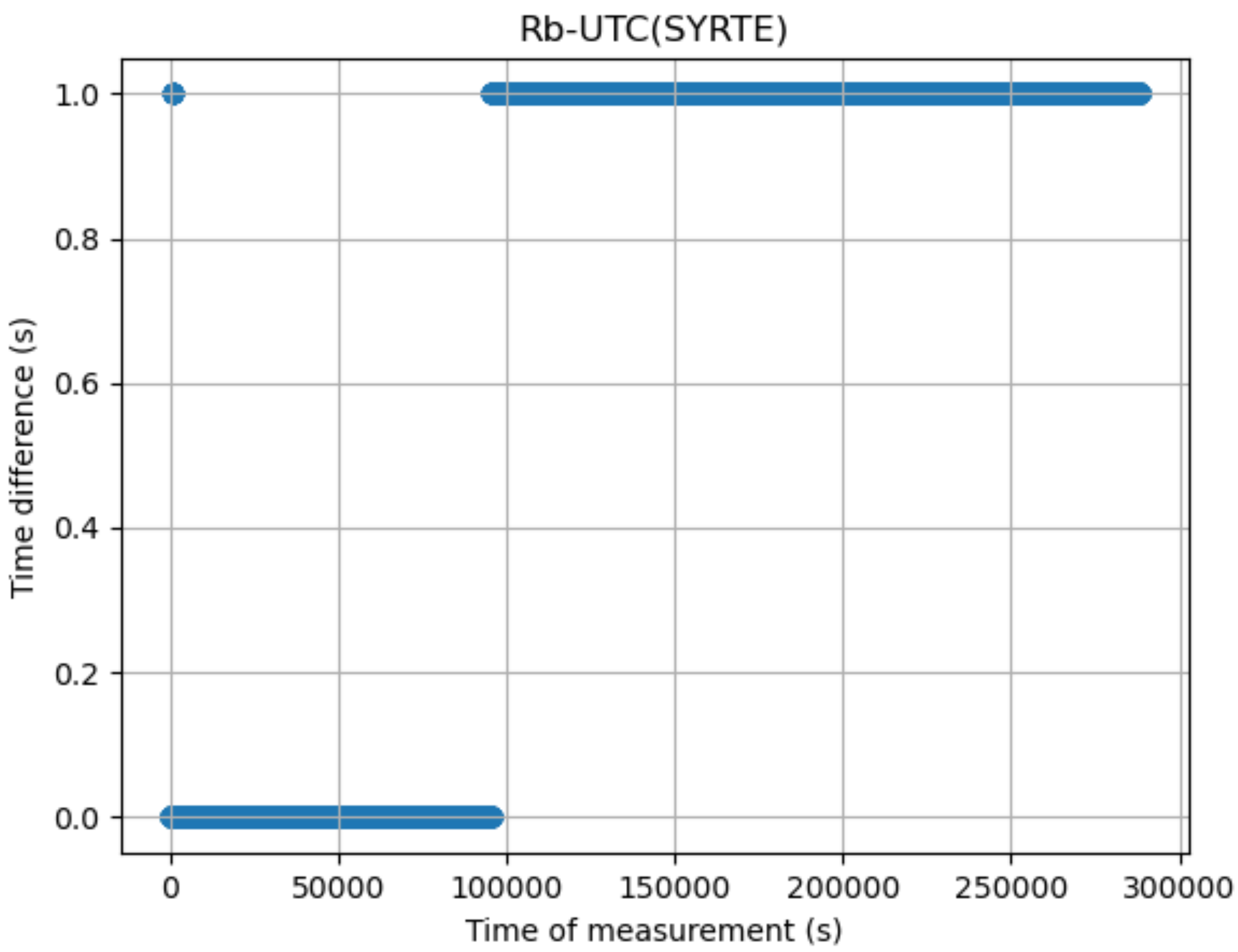




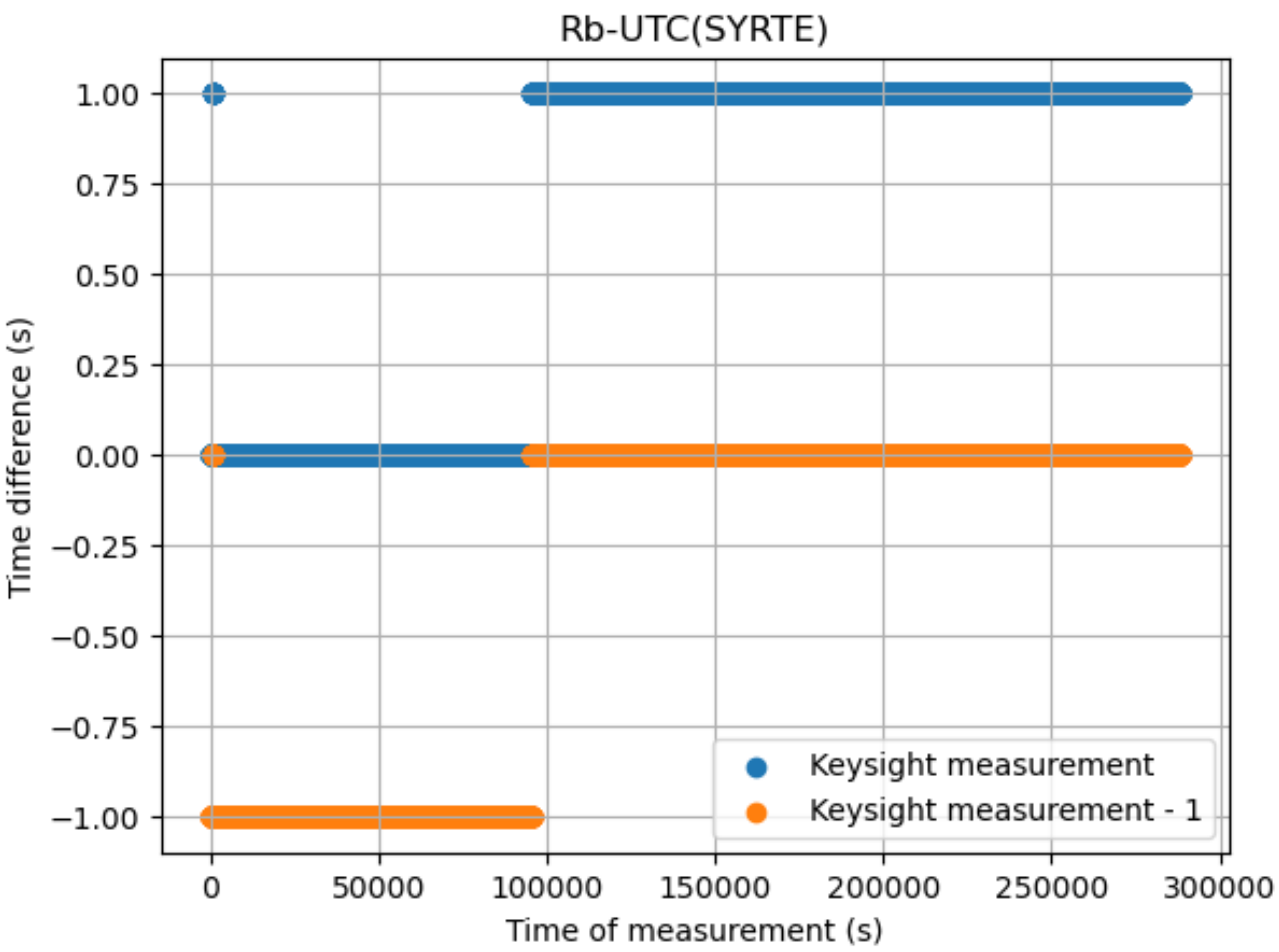
Date (in MJD) of the applied correction

Results

Hyper-Kamiokande



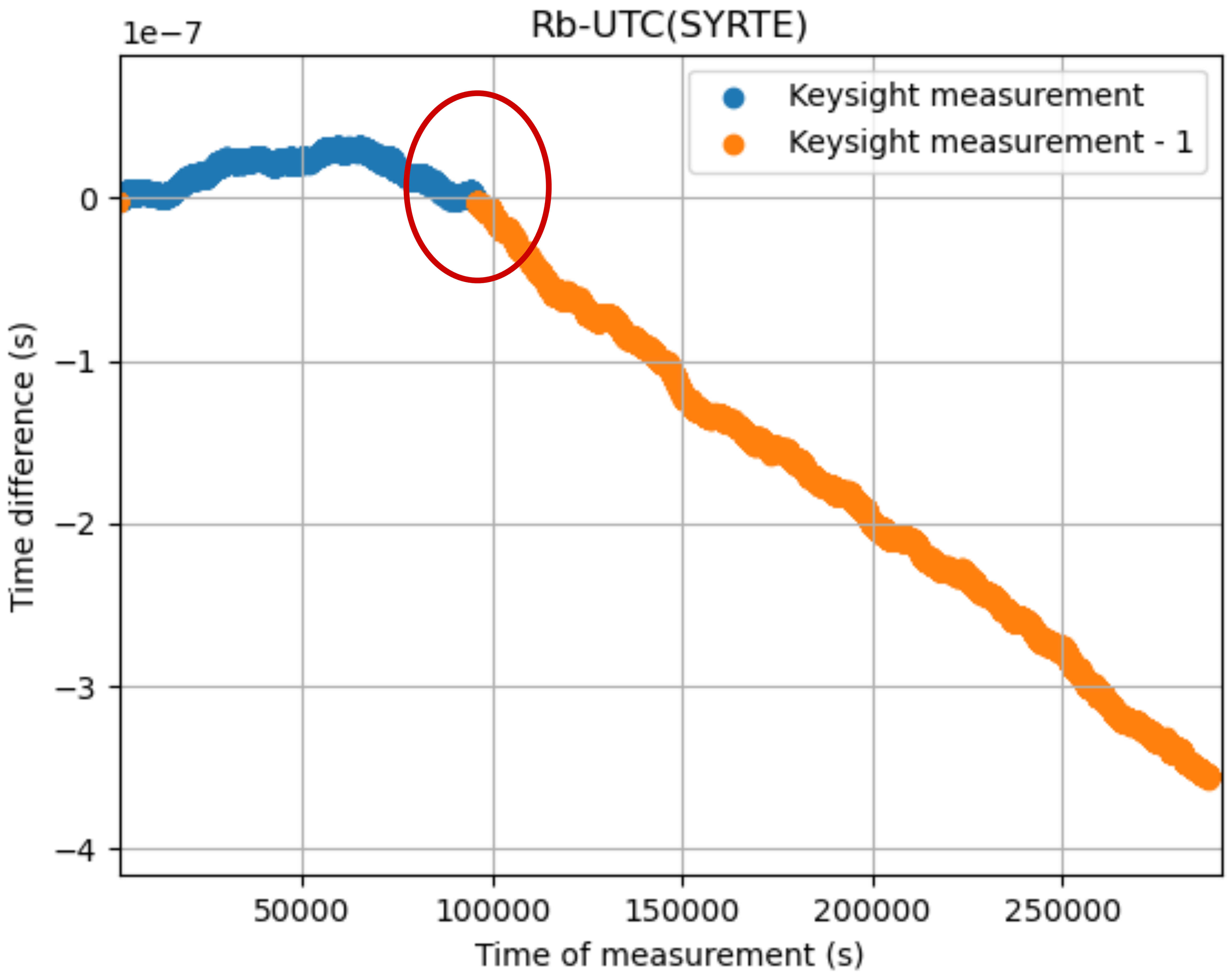
When the time difference becomes too negative, the order of the PPS switches



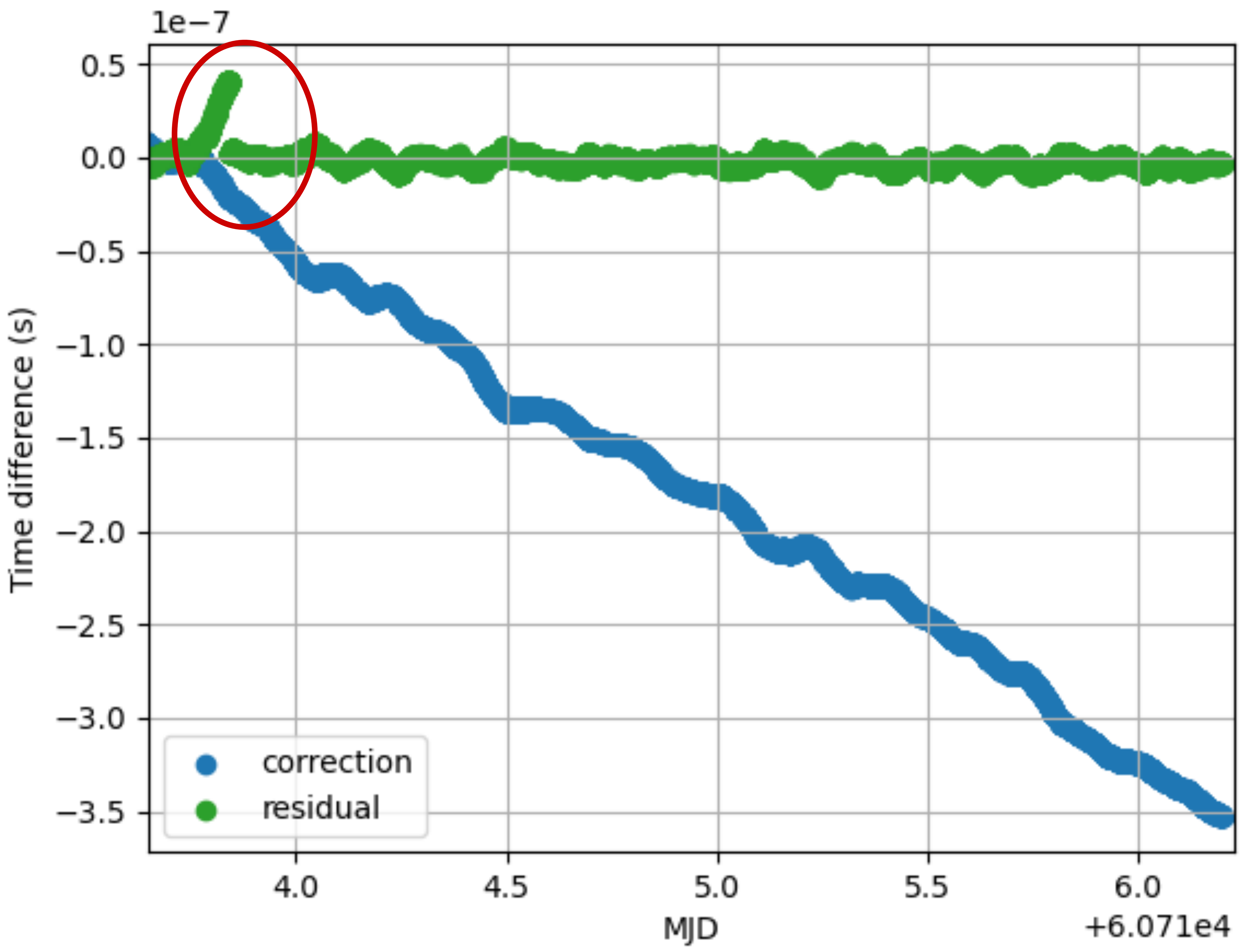
Take this into account by applying correction to “measurement-1s” when needed

Results

Bug when the Rb-UTC(SYRTE)<0

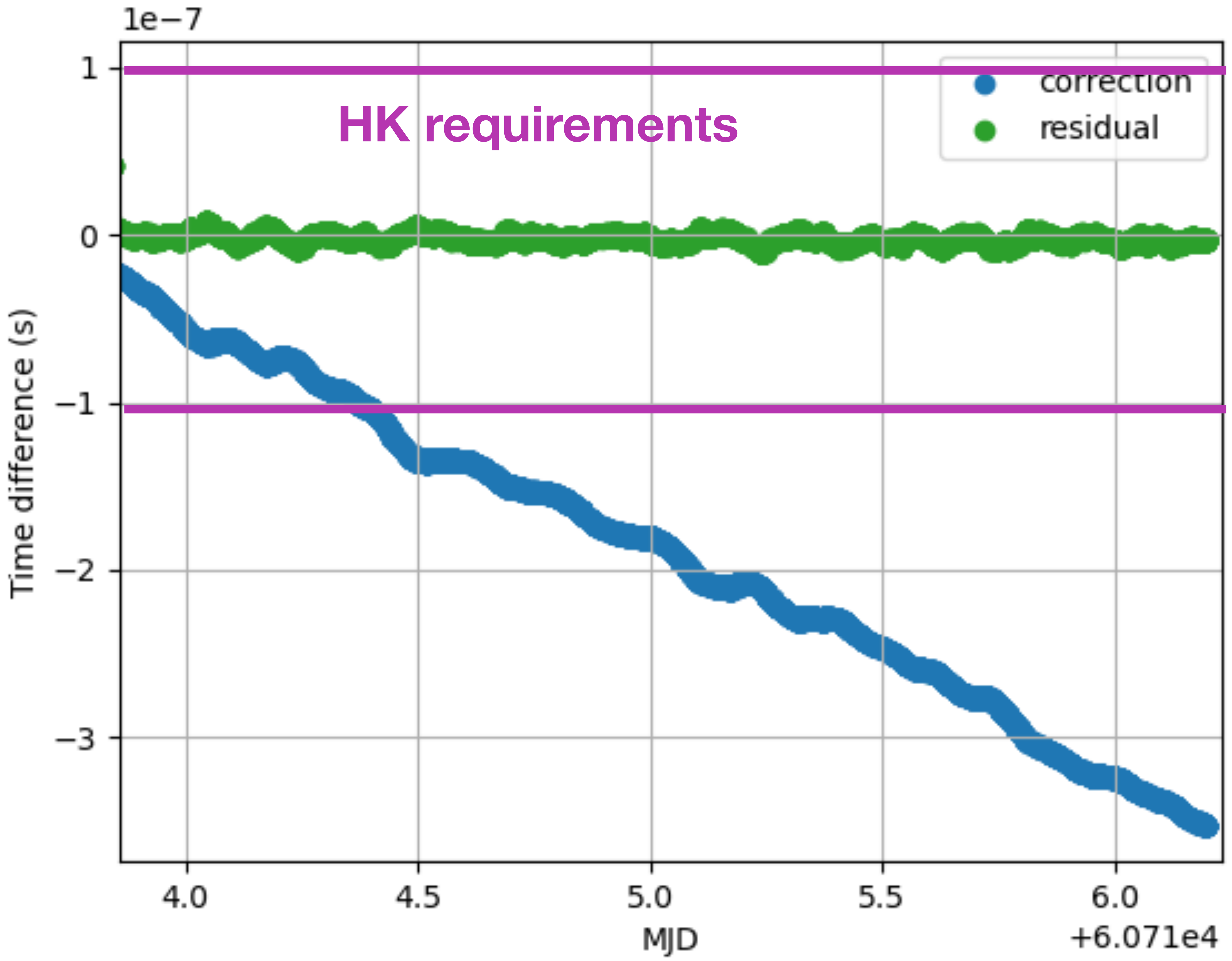


Distribution to which the correction should be applied

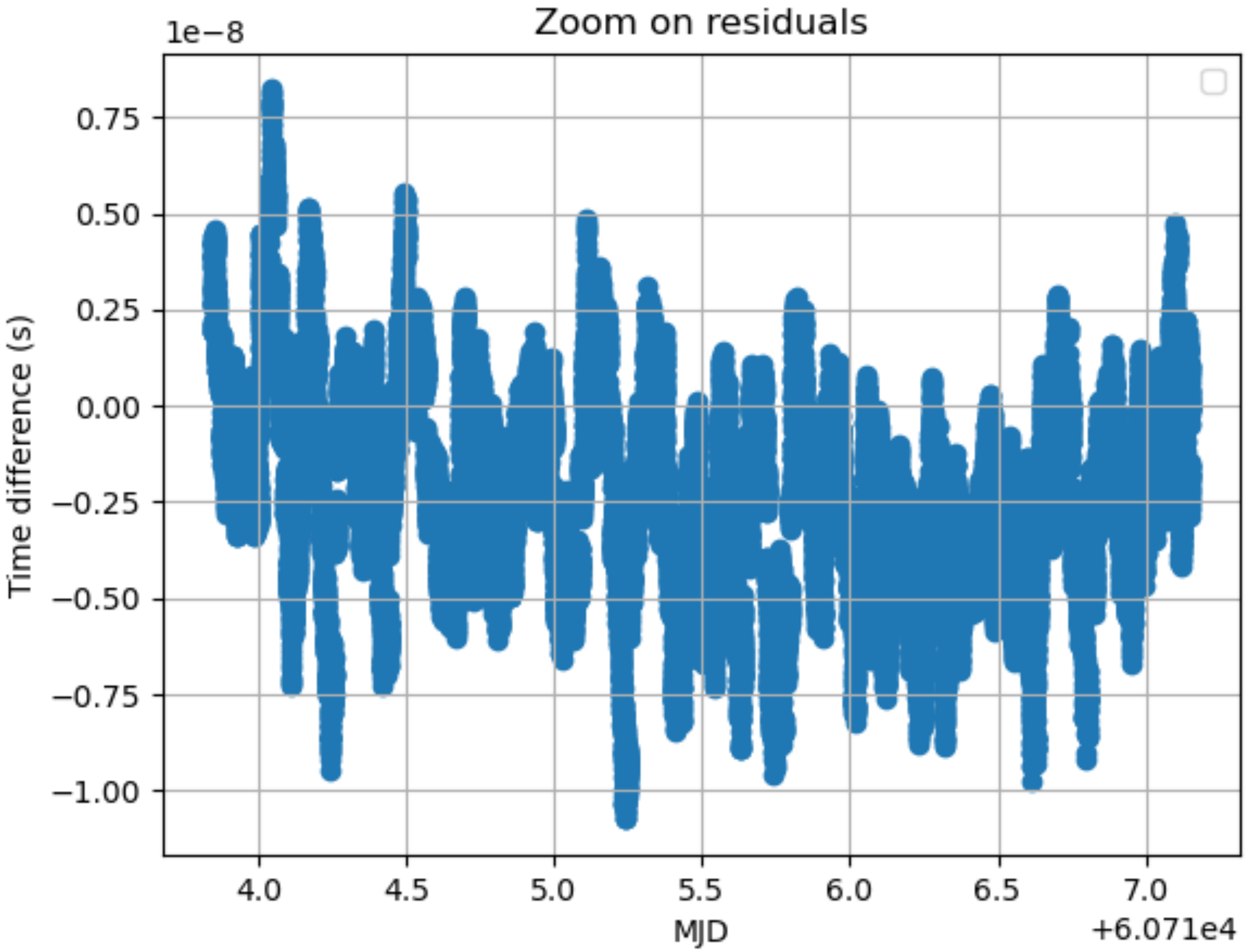


After bug fix, no more drift

Results



After bug fix



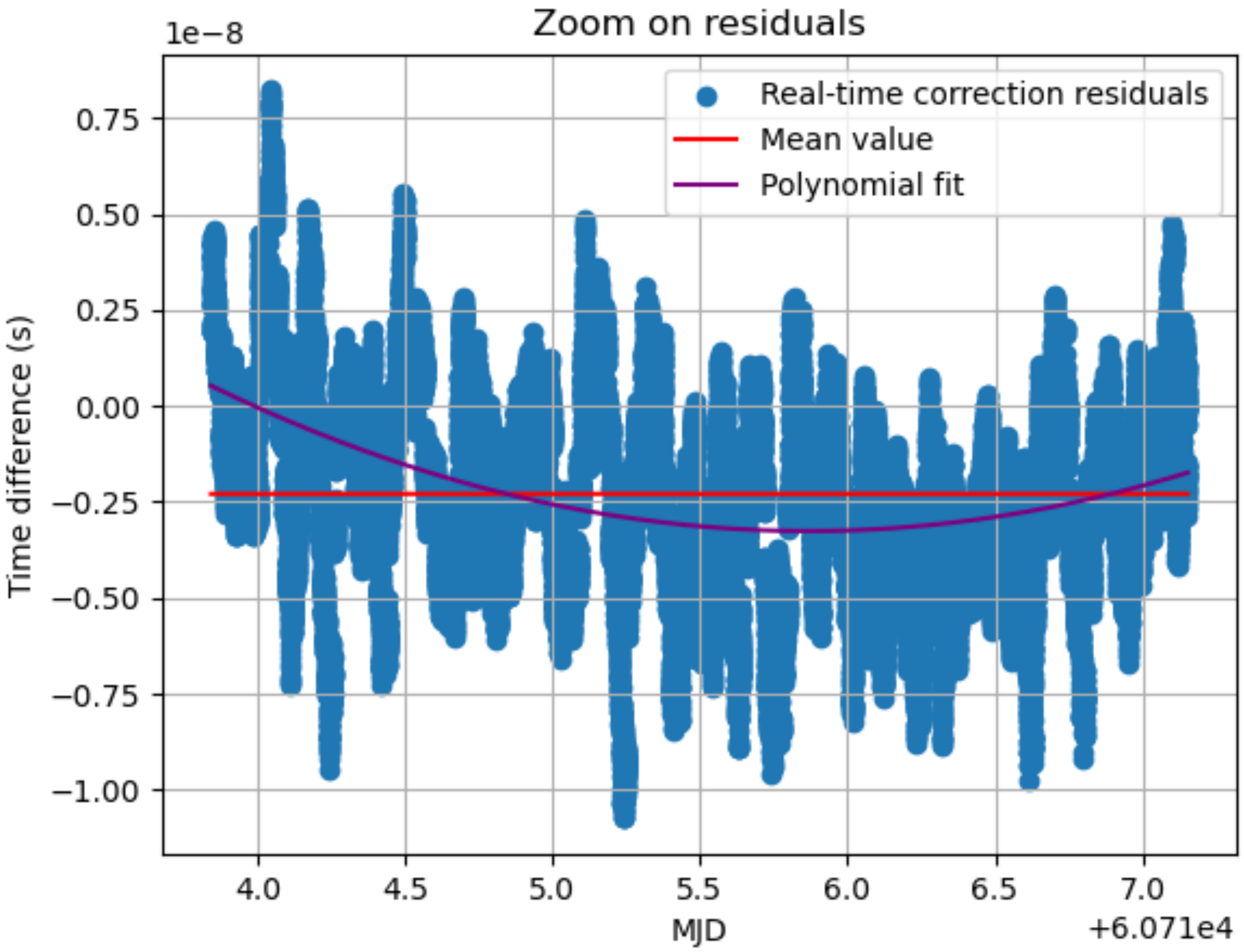
Conclusion

Hyper-Kamiokande

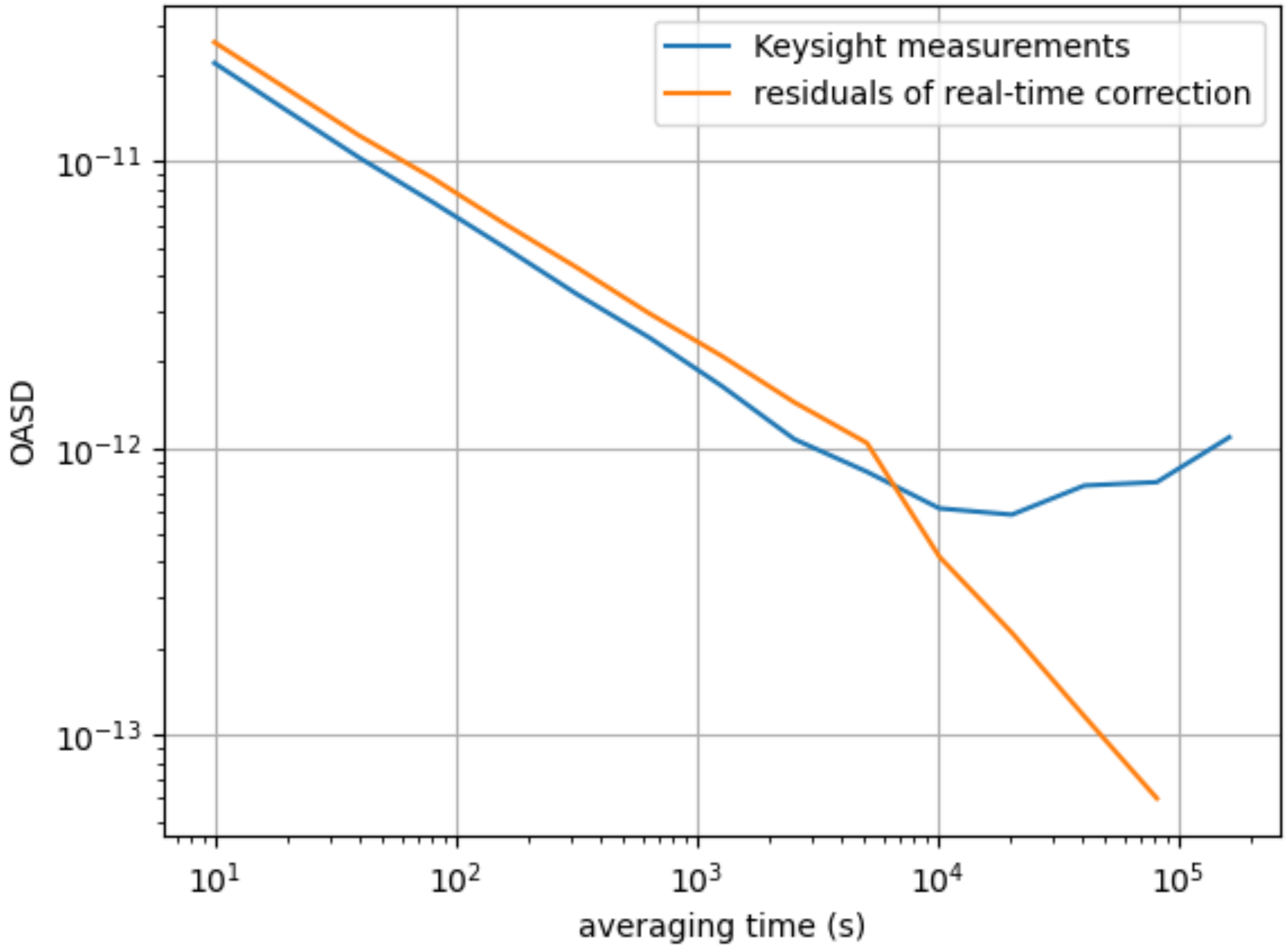
- Need to collect more statistics to make sure there is no residual drift
- Features to implement in the future:
  - Alarm system (frontend crash etc.)
  - Error estimation for the correction (could be useful in case of low satellite coverage)
  - Use GNSS comparisons to estimate when to change the Rubidium frequency

Thank you!

Results


Zoom on residuals
1e−8
Real-time correction residuals
Mean value
Polynomial fit
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
Time difference (s)
4.0
4.5
5.0
5.5
6.0
6.5
7.0
MJD
+6.071e4

Results



Keysight measurements
residuals of real-time correction
10^{-11}
10^{-12}
10^{-13}
OASD
10^1
10^2
10^3
10^4
10^5
averaging time (s)