Rubidium clock drift correction for HK timing system

Claire Dalmazzone, 19th March 2025

Contents

- Results of first real-time tests
- SF monitoring: first tests
- Parallel SF monitoring and time stamps correction

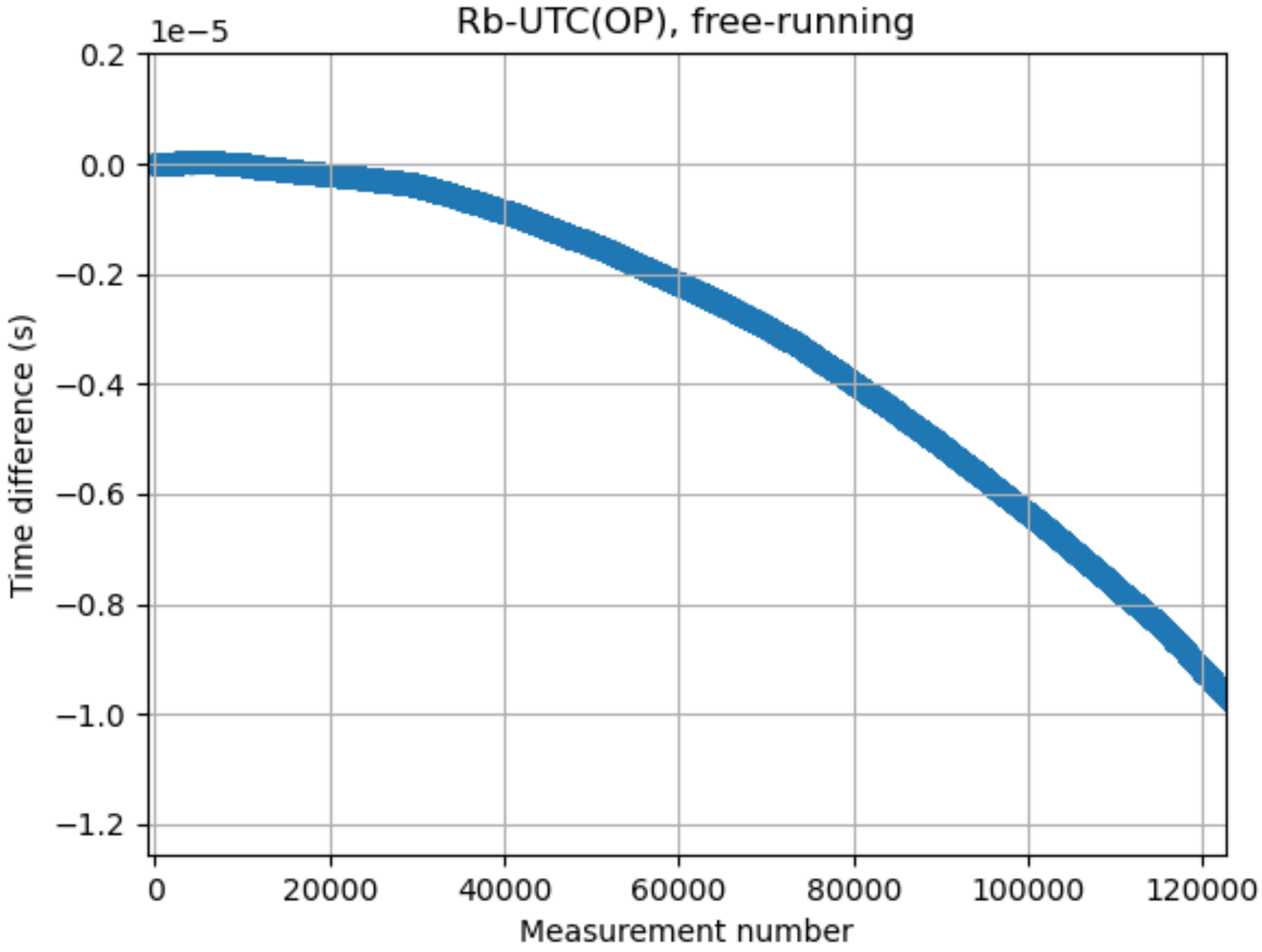
Results of first real time tests

Results

Hyper-Kamiokande

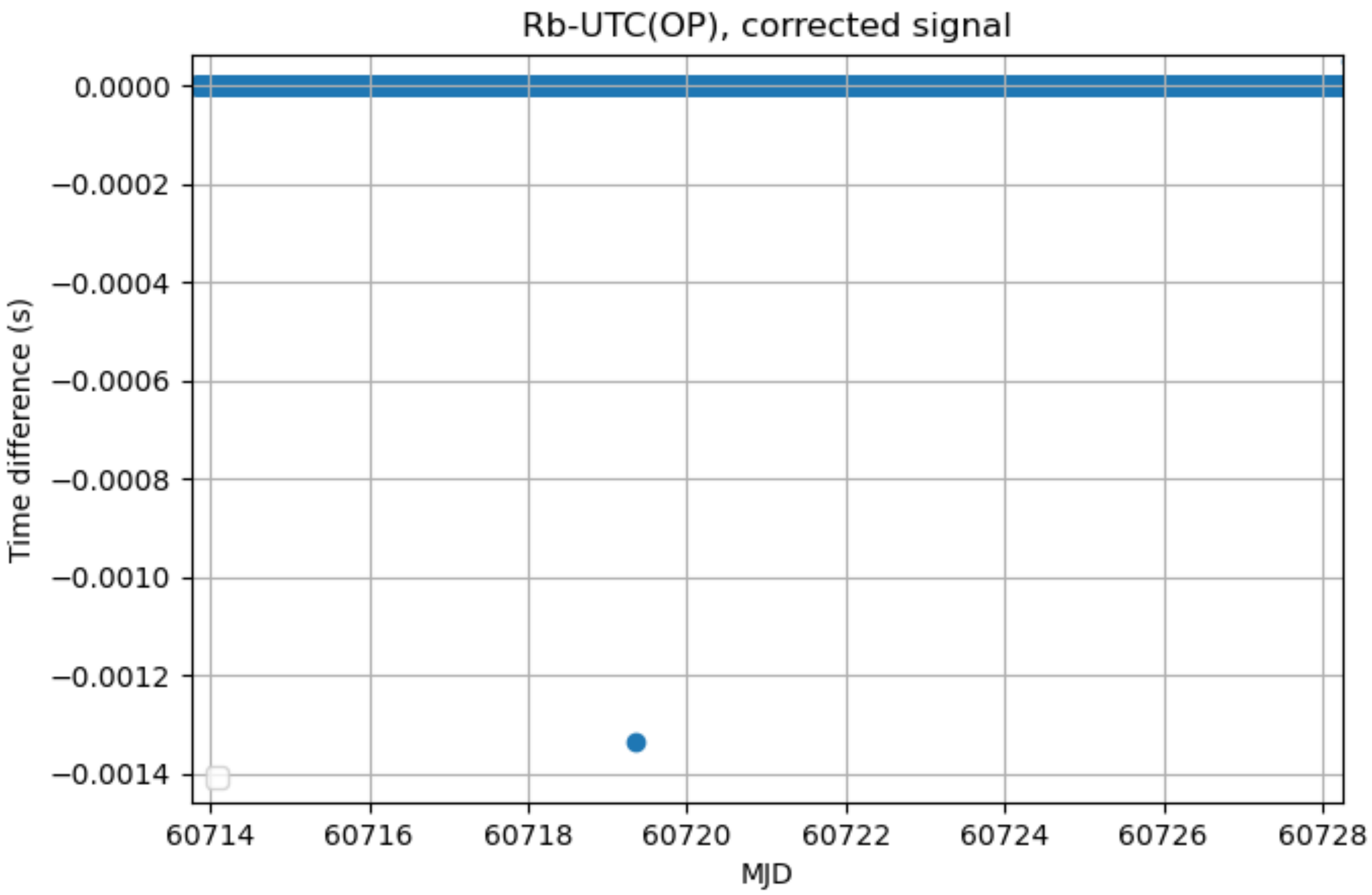
- First test of real-time correction was interrupted by a jump in the SYRTE PPS signal
- Was still able to collect a few weeks of data
- Apart from an outlier (see next slide), the real-time correction seems to work:
    - No sign of remaining drift in the OASD until $\tau = 4 \times 10^5$ s
    - Mean residual is (-1.3 ± 3.6) ns (excluding the outlier)

Results

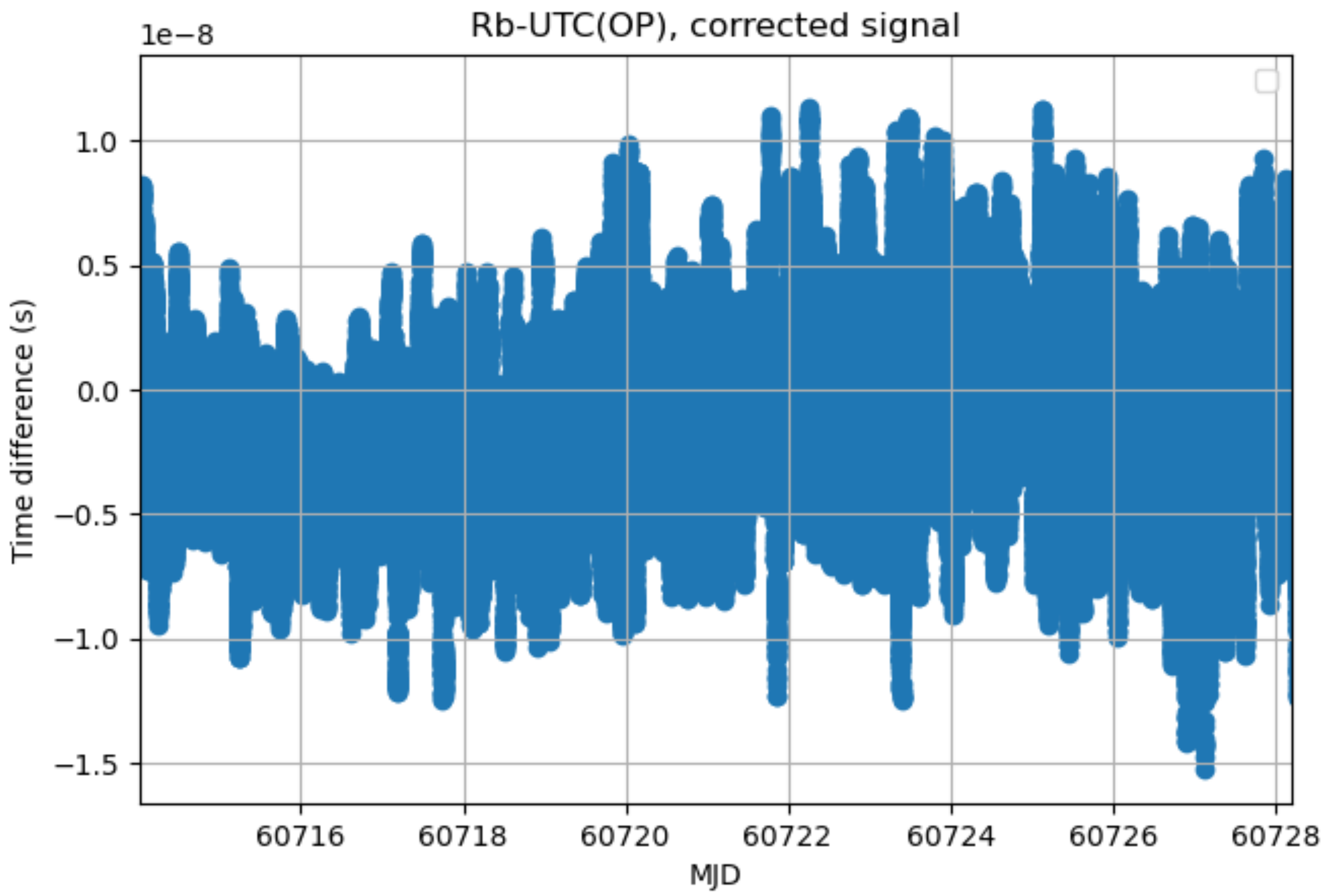


Rb-UTC(OP), free-running
1e−5
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
−1.2
Time difference (s)
0
20000
40000
60000
80000
100000
120000
Measurement number

Results

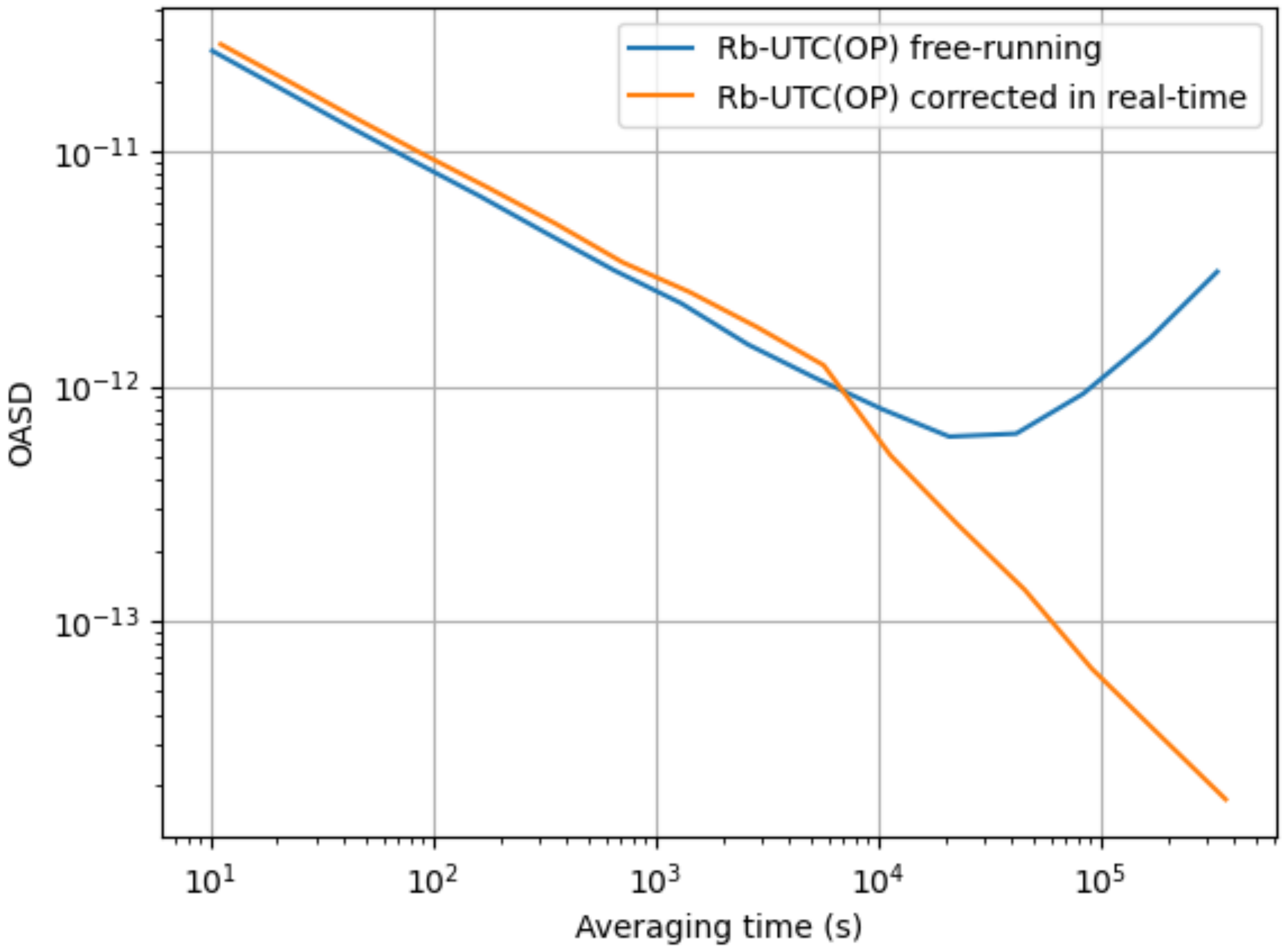


Rb-UTC(OP), corrected signal
0.0000
−0.0002
−0.0004
−0.0006
−0.0008
−0.0010
−0.0012
−0.0014
Time difference (s)
60714
60716
60718
60720
60722
60724
60726
60728
MJD



Rb-UTC(OP), corrected signal
1e−8
1.0
0.5
0.0
−0.5
−1.0
−1.5
Time difference (s)
60716
60718
60720
60722
60724
60726
60728
MJD

Results



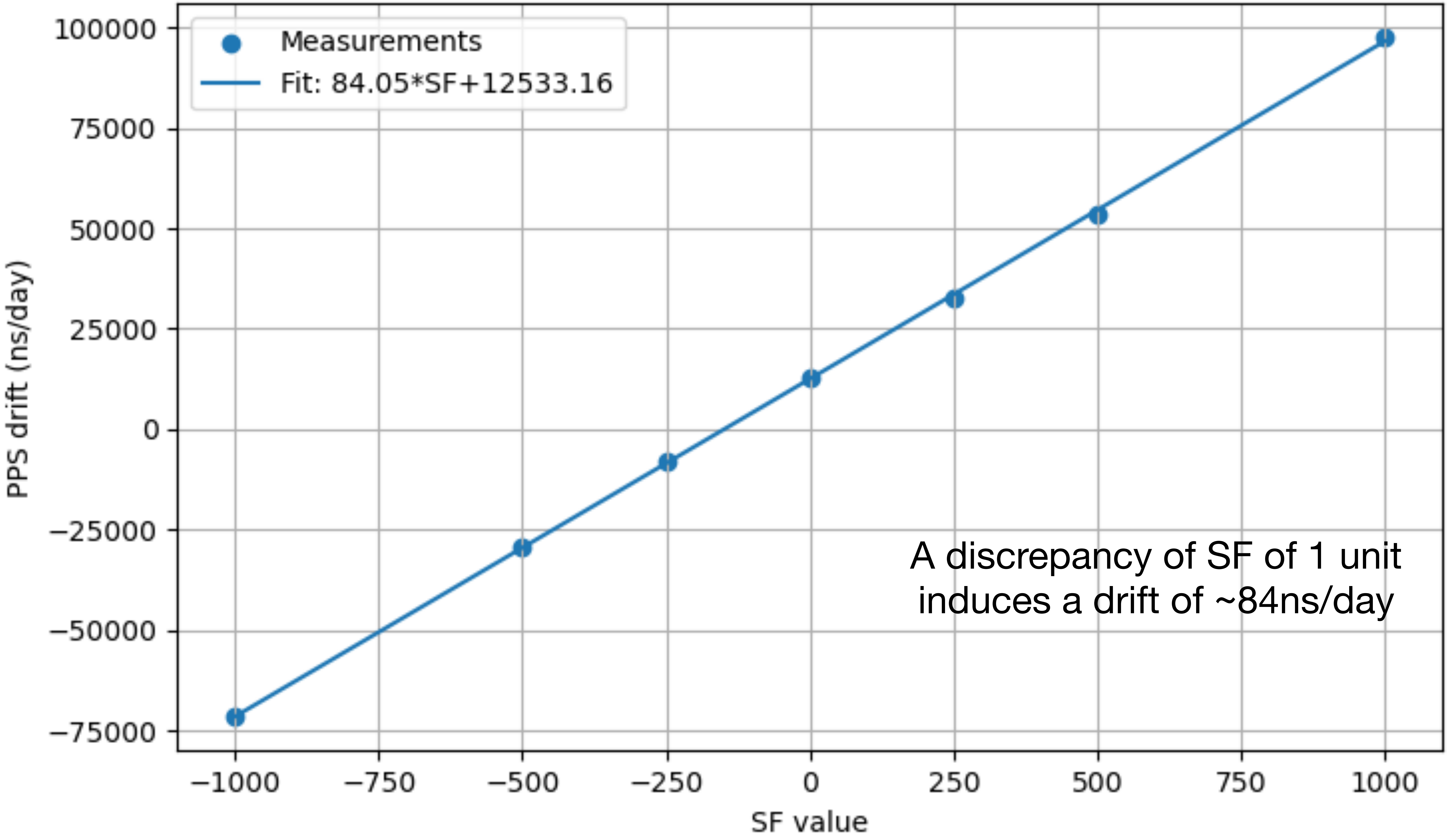
Rb-UTC(OP) free-running
Rb-UTC(OP) corrected in real-time
10^{-11}
10^{-12}
10^{-13}
OASD
10^{1}
10^{2}
10^{3}
10^{4}
10^{5}
Averaging time (s)

SF monitoring

SF monitoring

- The frequency drifts quite rapidly, leading to a ~10 microseconds PPS drift in two weeks
- This can cause issues in the long term as the range of the Septentrio measurement is limited to ± 1 ms. Going beyond this range will cause jumps in the Septentrio measurements and bias the correction
- This effect can be mitigated by implementing a real-time monitoring of the frequency: the frequency drift is monitored using the septentrio measurement and in case of a too big drift, the correction decides to change the frequency of the clock by changing the SF parameter

SF monitoring



A discrepancy of SF of 1 unit induces a drift of ~84ns/day

SF monitoring

However we need to be careful about

- Short term stability of the signal: the SF parameter should not be changed too often and by a too large number as it might deteriorate the signal's jitter?
- Compatibility with our correction algorithm. When the SF parameter is changed, old Septentrio measurements are not useful anymore to predict future behaviour…
- Other concerns?

SF monitoring

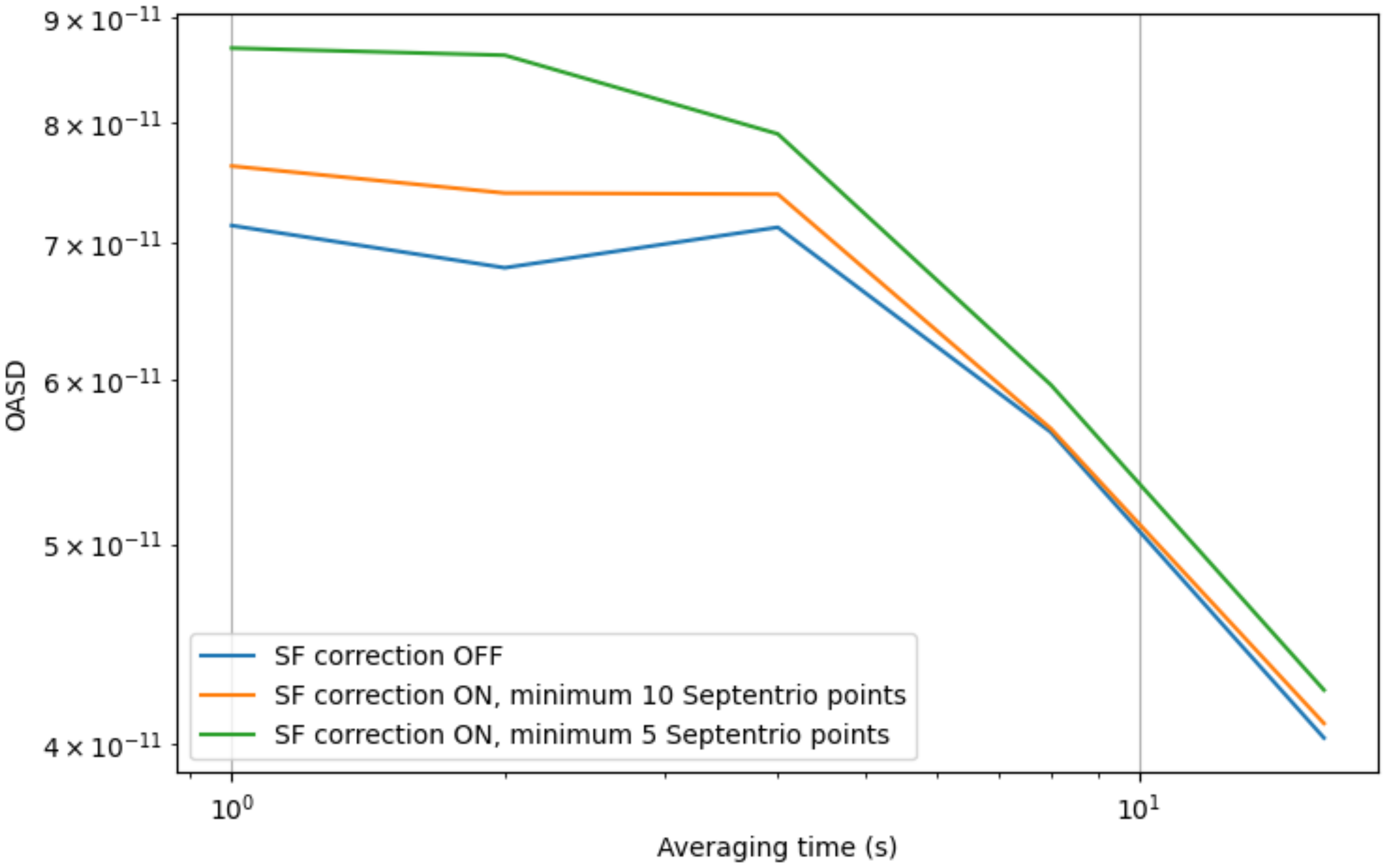
Performed first test with minimal features:

- If the drift measured from the Septentrio measurements (linear fit of last points) is bigger than an upper limit (currently 140 ns/day), SF is changed.
- The new value of SF is SF_new=SF_old-drift//82 ns/day (over-correction is intentional)
- The correction will forget the old septentrio measurement except the last one. It will also update the correction coefficients to take into account the impact of the change of SF on the PPS drift.

SF correction

Impact on short term stability

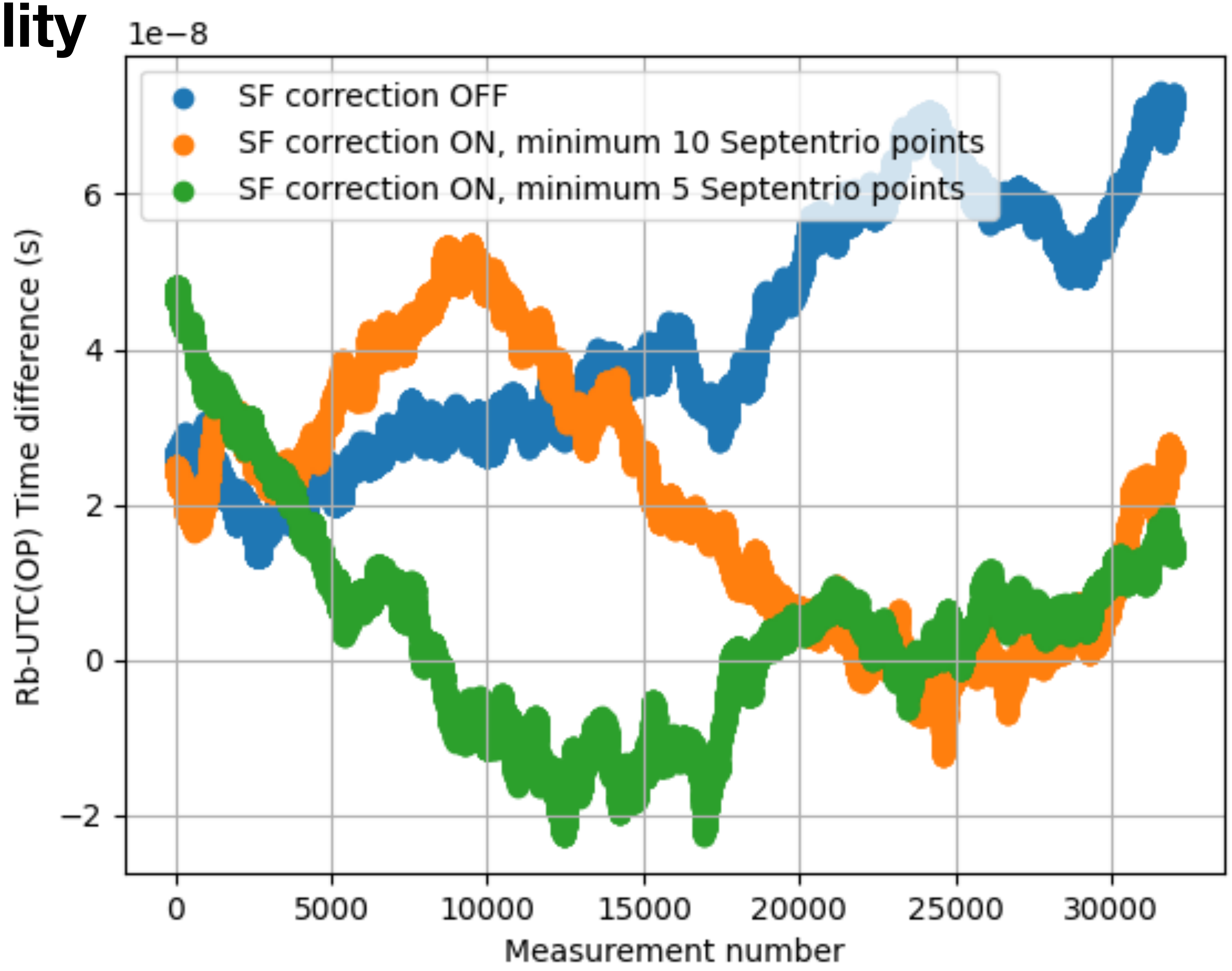
Impact on short term stability is small. Still it is better to limit the frequency of the change of SF.



SF correction

Impact on short term stability

Impact on short term stability is small. Still it is better to limit the frequency of the change of SF.



SF monitoring

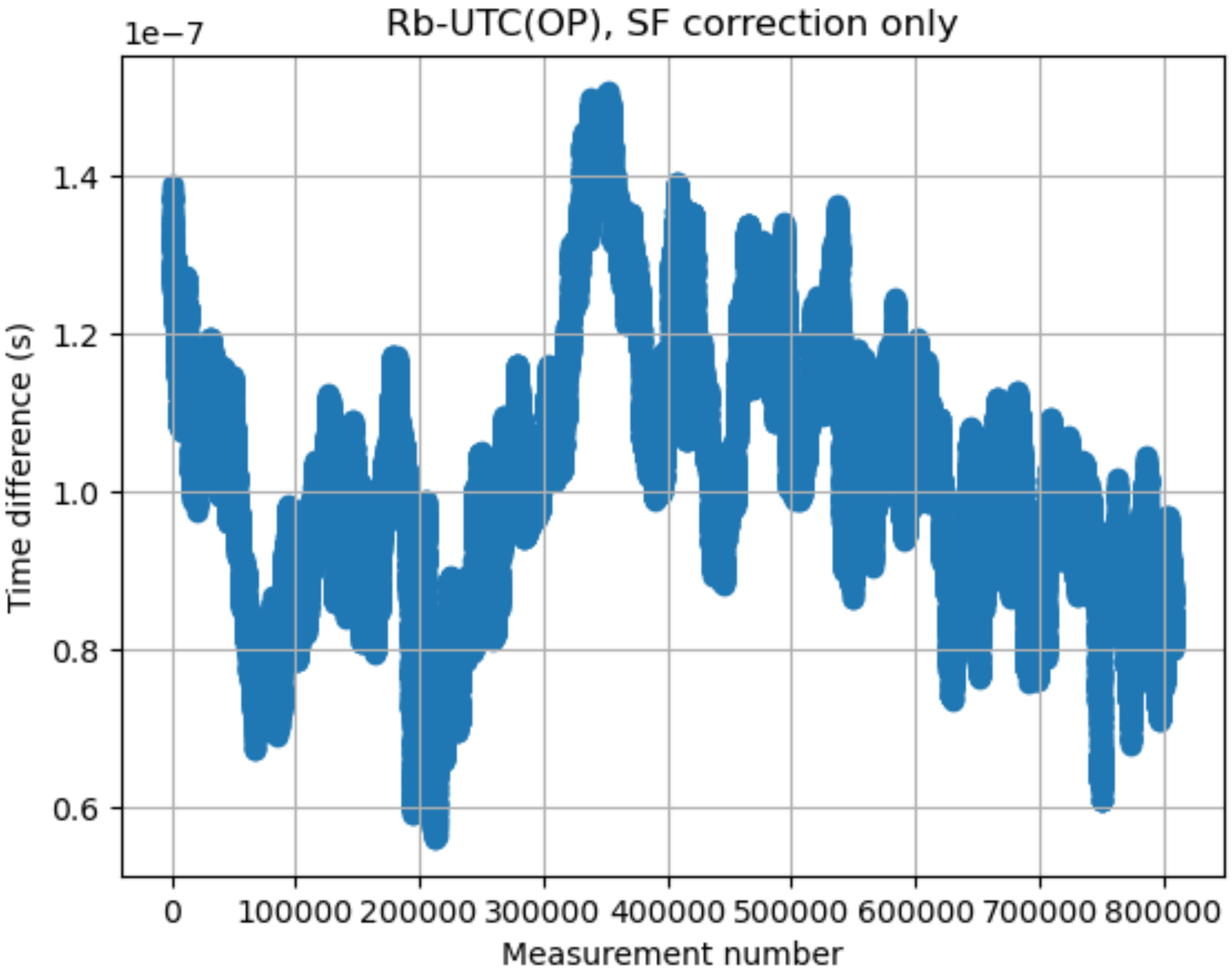
Impact on time correction

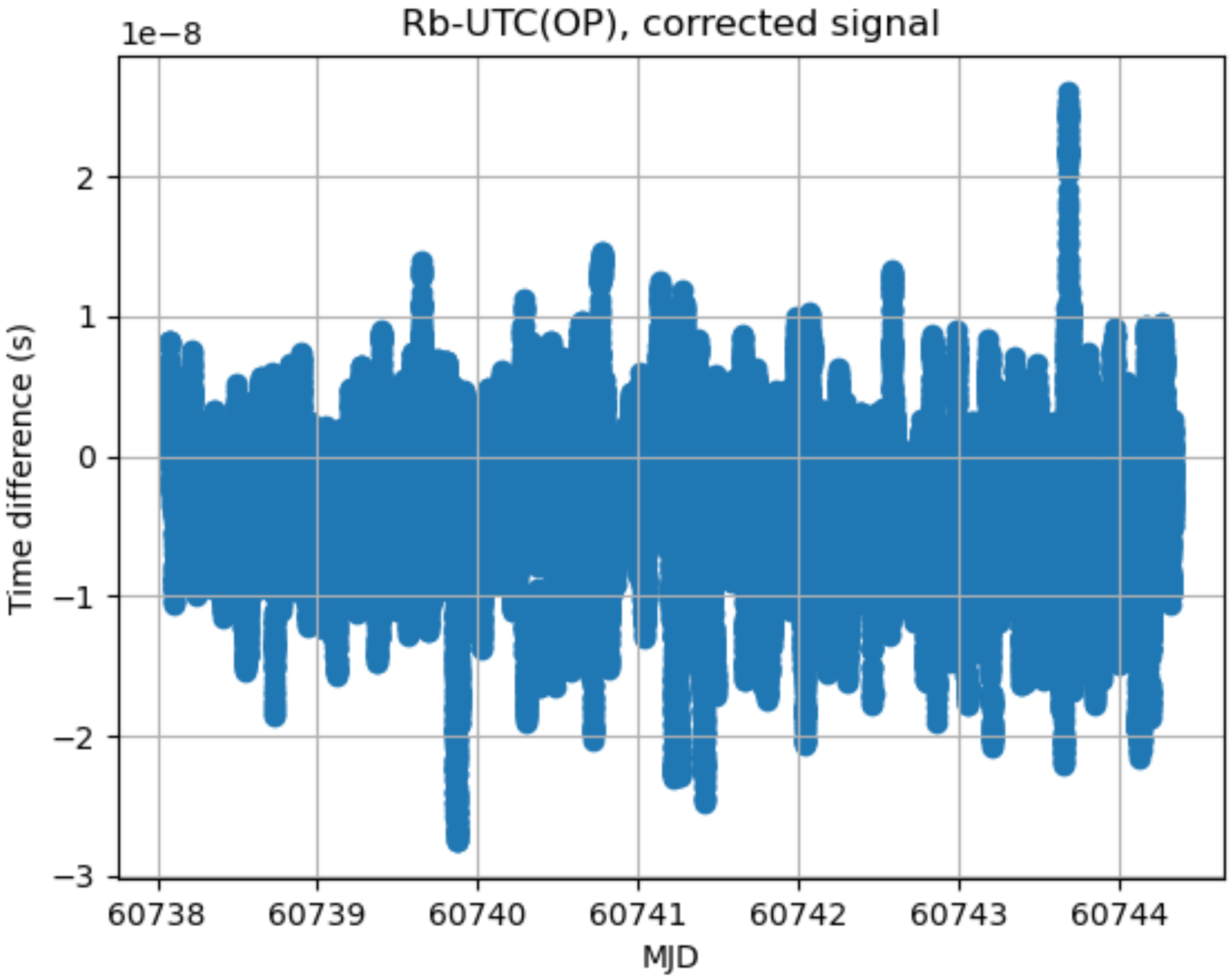
Performed first test with the following features:

- The drift has to measured with the last 10 Septentrio measurements to be able to change SF (same length as for the usual correction).
- The queue of Septentrio measurement to use t update correction is re-initialized after a SF change
- SF change is limited to +-5 units
- Doing in parallel the SF correction and the time signal correction

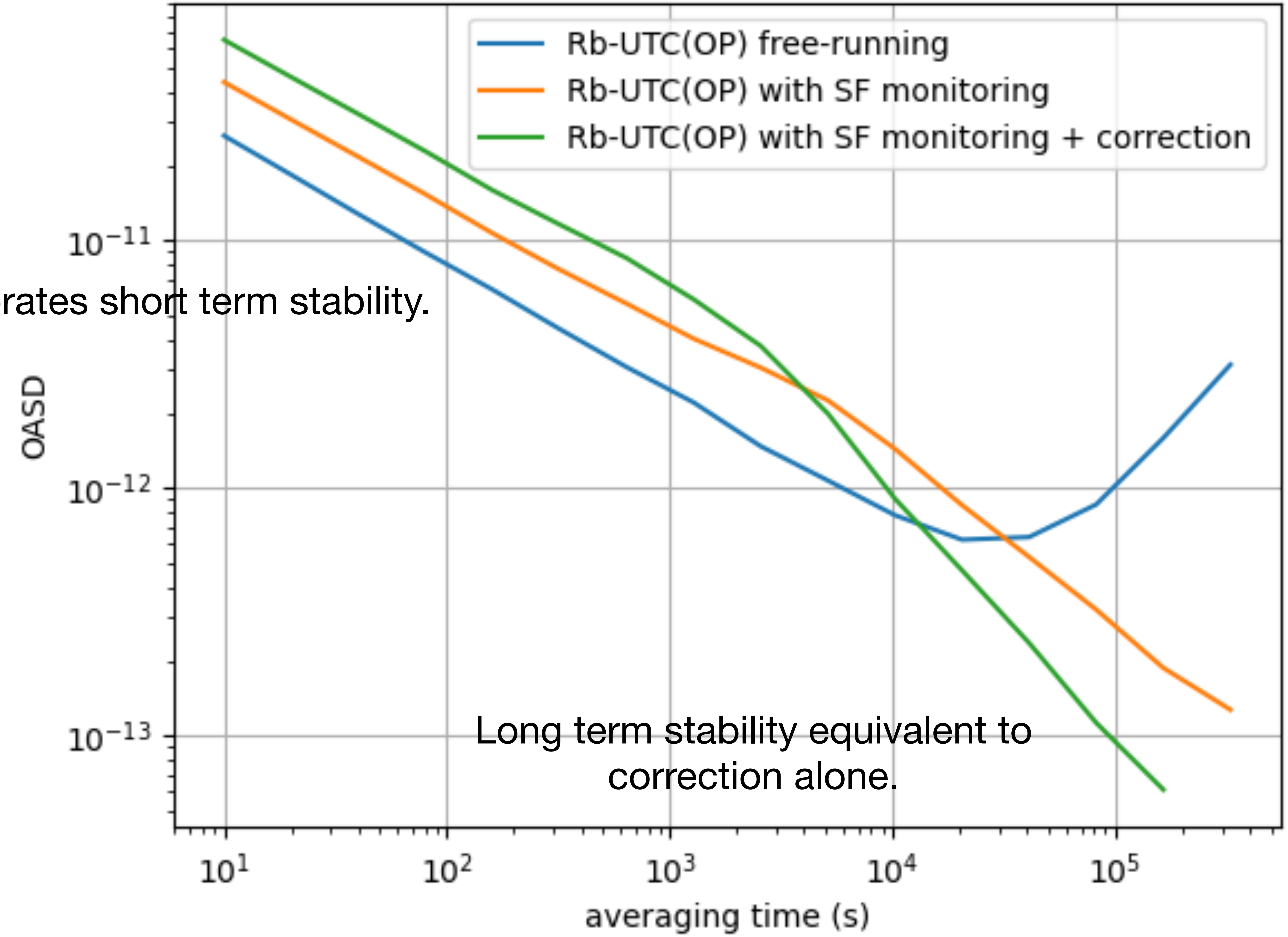
SF monitoring

Impact on time correction





SF monitoring



Deteriorates short term stability.

Long term stability equivalent to correction alone.

SF monitoring

Impact on time correction

Conclusions so far

- Good solution to limit the drift of the free-running signal (useful for the reliability of the Septentrio measurements)
- But could deteriorate short term stability and performance of the time signal correction
- Performances could be better with finer features (see ideas on next slides)

SF correction

Possible improvements

- Short term stability problem could get better if we require a minimum number of Septentrio measurements in the queue to update the correction coefficients (except at initialisation).

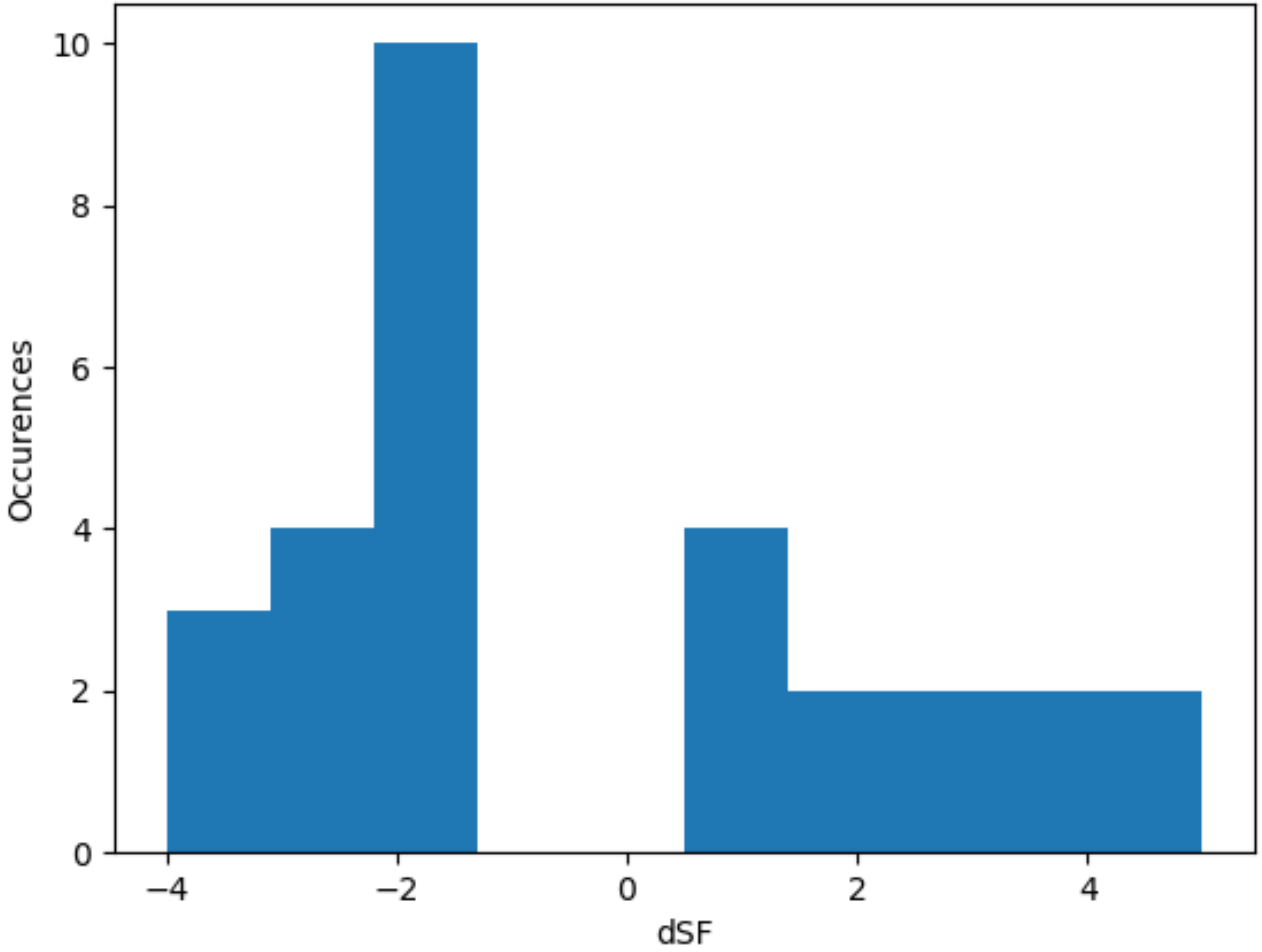
- Correcting SF changes the stability of the signal from which we optimised the correction time window of our time signal correction. With a SF correction in parallel, this optimisation could be revised.

- Alternatively, we could have two queues of Septentrio points: one for SF correction (longer) and one for time signal correction (~3 hours long). This is motivated by the fact that we would want SF correction to correct the deterministic drift and time signal correction to correct the RW.

SF correction

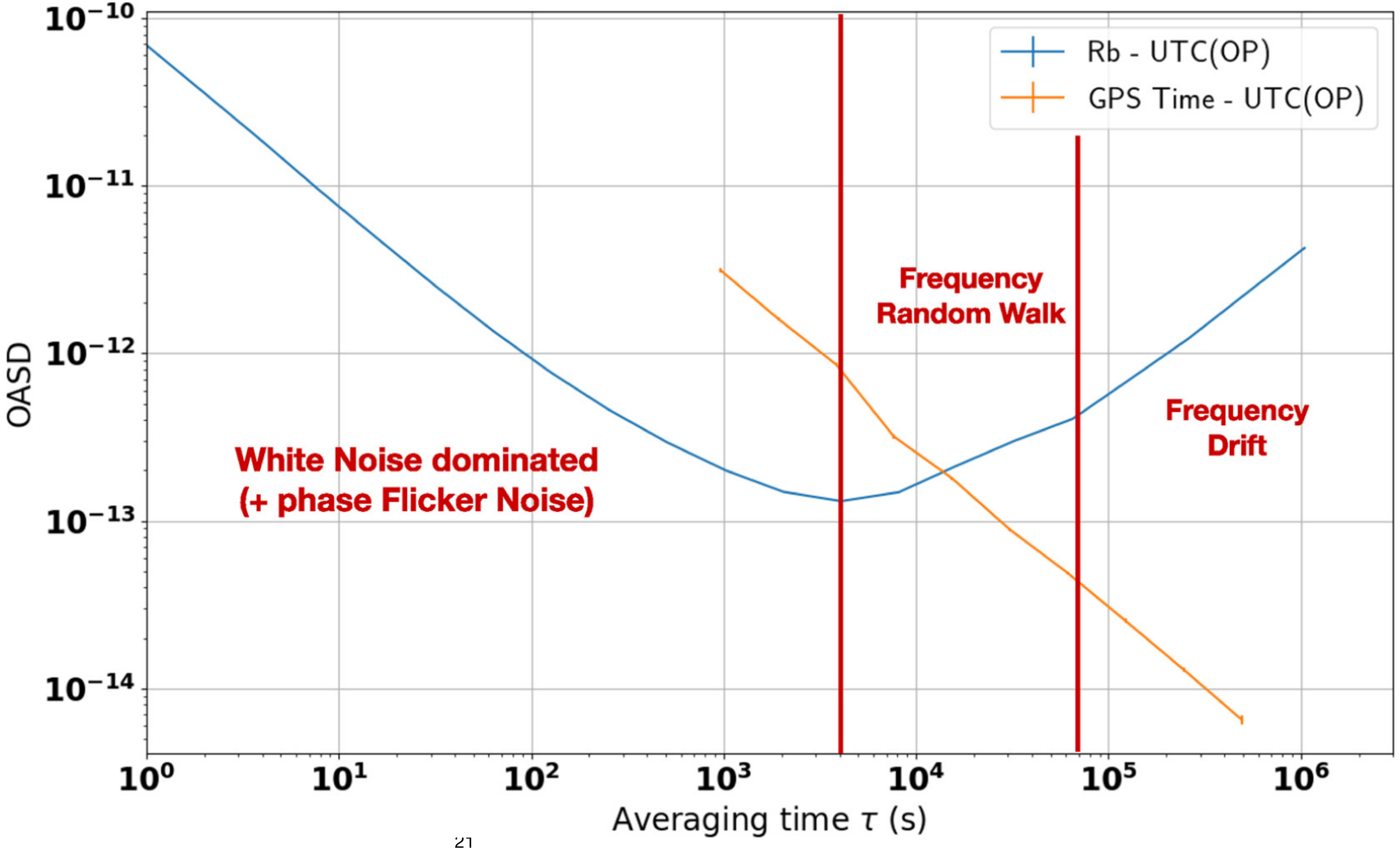
Possible improvements

- The repartition of the amount of change of SF during the ~100 hours run show that we clearly over-correct SF (mean is -0.4).
- By correcting less often we could improve the short term stability?



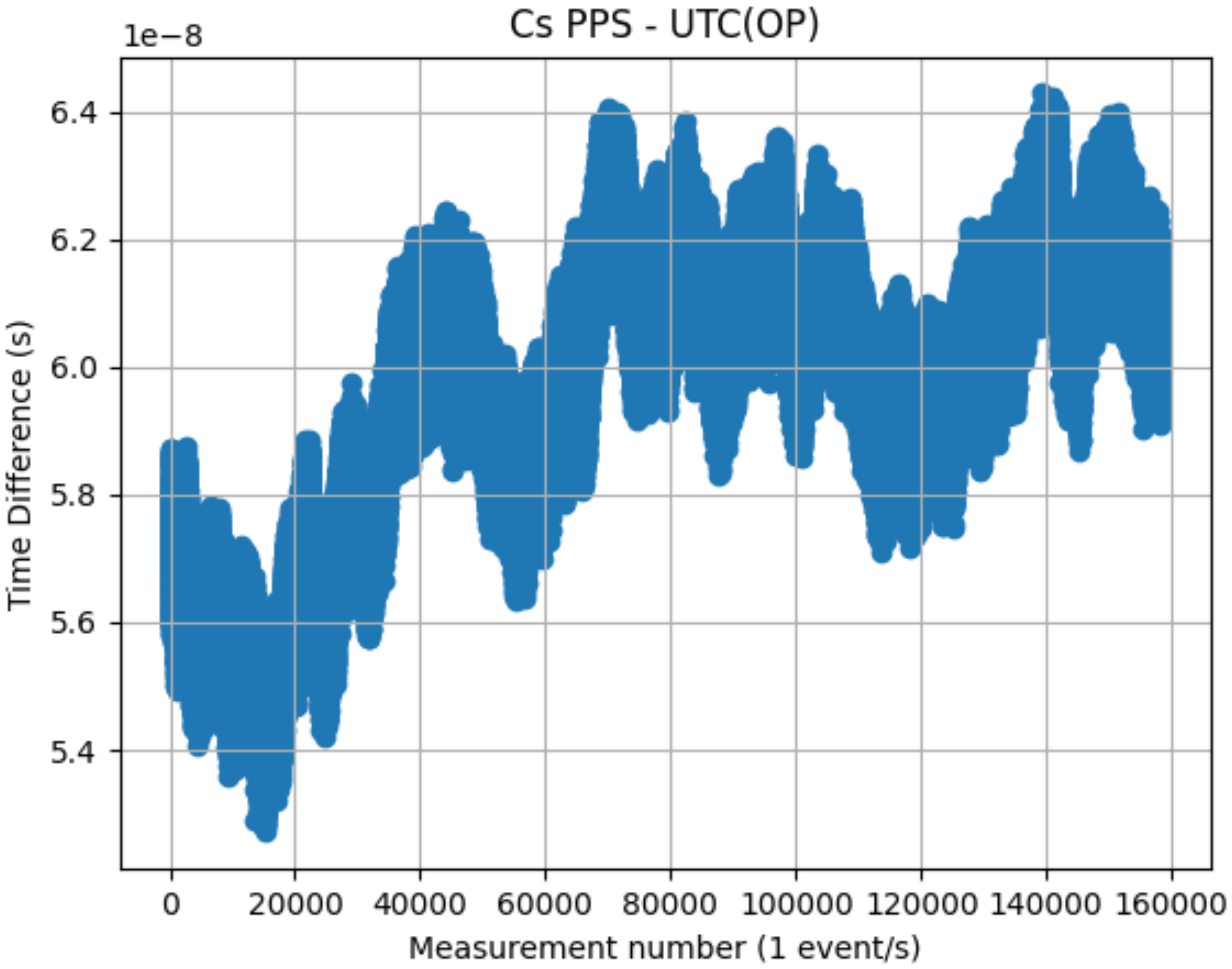
Frequency drift

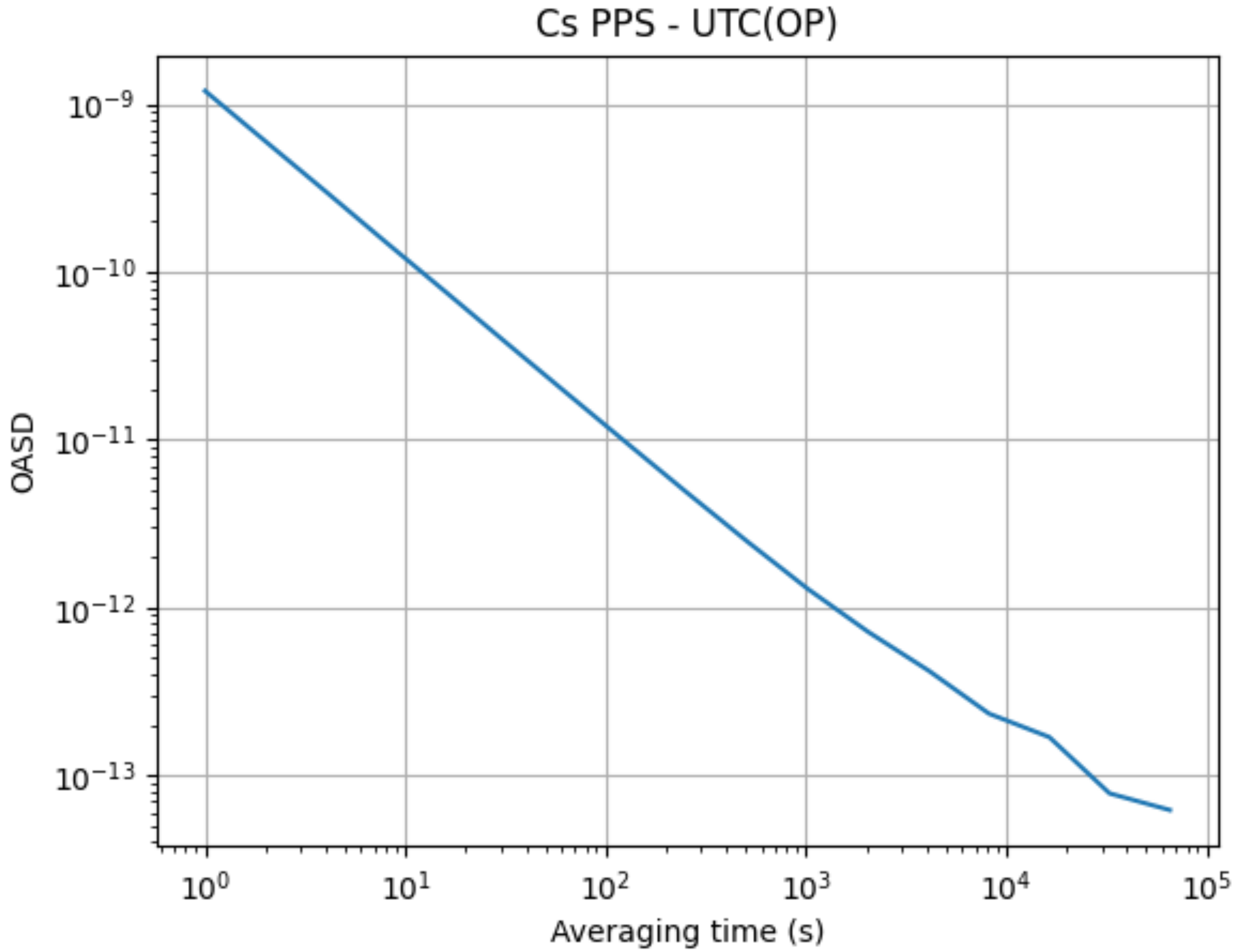
Frequency drift starts to limit ASD around 7x10^4s (~19.5 hours ~73 Septentrio measurements)



First results with Cs

First tests with the Cs





Lot of white noise (>10 times more than Rb)! Could it be due to the counter's configuration: wrong impedance?
Starting another run with a 1M impedance (instead of 50k)

Conclusion

Hyper-Kamiokande

- Apart from one outlier (coming the application of the correction, to investigate) real-time method looks successful
- Parallel SF monitoring and correction seems possible but will require fine-tuning
- Several solution possible, what to prioritise?
- Another concern: There is still ~50-60ns difference between UTC(OP) received via Refimeve and GNSS Time. Do we know where this comes from?