Report to the 21st CCTF: Improving the stability of UTCr

G. Petit, A. Harmegnies, G. Panfilo

Laboratories participating to UTCr inquire about steps that they see in [UTCr-UTC(k)] when these are large (> 3-4 ns) and consistent between several laboratories with stable UTC(k). Such large steps seem to have occurred more frequently in the last 2 years. Of course the BIPM Time department already knows about these steps from the comparison [UTCr (Week) – (UTCr (previous week)]. However the quasi-automatic and quasi-real-time nature of the UTCr computation prevents from conducting detailed analysis on the spot. This report examines several issues related to these problems in UTCr and proposes some solutions.

1. The symptoms

Looking at the difference [UTCr-UTC] (plotted Figure 1) it seems clear that the situation has degraded since 2014. The standard deviation of the differences, which was 0.8 ns in 2013 and 1.1 ns in 2014, has increased to 1.4 ns in 2015-2016.

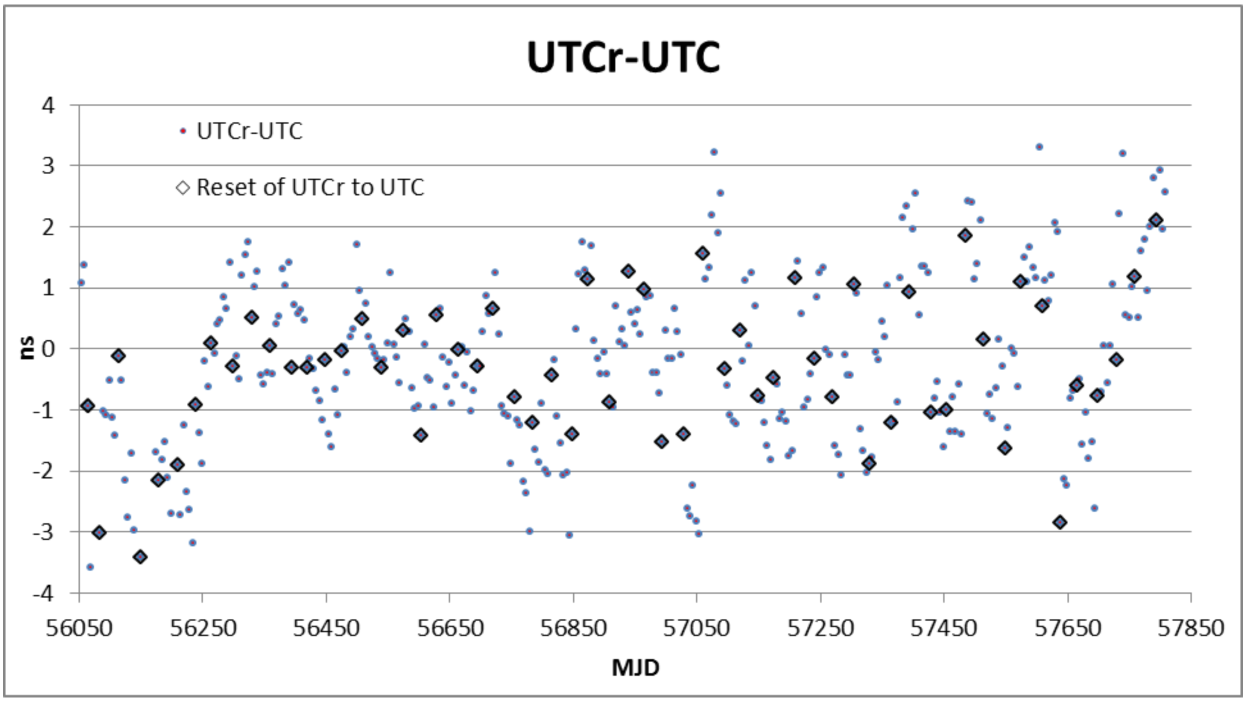

Figure 1 : [UTCr-UTC] since the origin of UTCr in May 2012 through early-2017. Diamonds correspond to the first date that includes the monthly "reset" of UTCr (see text).

Each month, in the first week of UTCr computation following the publication of Circular T, UTCr is "reset" to UTC by using the past month's [UTC-Clock] as past values which thus replace the [UTCr-Clock] values. In general, it is expected that this "reset of UTCr" provides a better agreement of UTCr with UTC. The values of [UTCr-UTC] for these reset dates are highlighted on Figure 1 and they also significantly degrade starting 2014.

2. Possible causes

2.1) As mentioned above, the situation has degraded since 2014 i.e. when the weighting algorithm has been changed in UTC and not in UTCr. This is likely a factor of explanation.

2.2) Some steps in [UTCr-UTC] have been documented to be associated to changes in time links / calibrations between the UTCr computation and the UTC computation. Such differences are unavoidable as the data used for UTC is the best data available for the whole month at the time of UTC computation, which happens after the corresponding weekly UTCr computations. On the other hand, each weekly UTCr computation uses data that may change from week to week. As an example, a change in the NTSC-PTB link explains a step in UTCr between weeks 1647 and 1648. It is however unlikely that this could be the dominant factor in explaining the behaviour seen in Figure 1.

2.3) Other factors may also be at work, linked to the monthly period of UTC computation:
In the present situation, the UTCr computation interval for one week ends on Sunday of the week and starts at the UTC standard date that is between 26 and 30 days before this Sunday. This “date charnière” is the end of the first interval of the past, which is used to compute the predicted rate of each clock over the computation interval. This method ensures that the computation interval has duration similar to that of the past intervals (30 days). This duration has been chosen to be consistent with the monthly UTC intervals.

On a week W of reset to UTC, data for the “date charnière” are [UTC-Clock], and this is also the case for week W+1 and W+2 (except in exceptional cases). In general, for weeks W+3 and W+4 (if the new Circular is not yet available), data from the date charnière are [UTCr-Clock].
For those weeks when [UTCr-Clock] is used for the date charnière (so called “UTCr-charnière” weeks), it is more likely that the predicted rate is biased with respect to a “UTC based” predicted rate (“UTC-charnière” weeks). Furthermore, due to the monthly nature of the computations, the sequence of “UTCr-charnière” weeks may repeat for some time, as in the example shown in Table 1 below.

Week	Date charnière	Computation UTCr	Type of Charnière at computation	Provides Charnière to
1645	57679	16/11/2016	UTC	1648
1648	57699	7/12/2016	UTCr	1652
1652	57724	4/1/2017	UTCr	1704
1704	57754	1/2/2017	UTCr	None

Any rate bias in the first “UTCr-charnière” week of each series is likely to yield a similar bias in the next week of the series, which may therefore yield systematic differences in [UTCr-UTC] over these weeks. When a “UTC reset” happens in the week following such “UTCr-charnière” weeks, this may cause a large step in UTCr.

An additional note is that, for the “UTC-charnière” weeks W and W+1, the date charnière is not the most recent date for which UTC is available. This means that UTCr extrapolates for a longer time than if the most recent UTC date was used as charnière (which would imply using a shorter computation interval).

3. Some tests

Tests have been carried out over 64 weeks, from UTCr weeks 1601 to 1712, providing 89 comparisons to UTC from MJD 57394 to 57834.

Several points should first be mentioned:

- Problems due to changes of links / calibrations (section 2.2) cannot be avoided. No attempt is made to estimate their possible average effect.
- It was realized during the tests that a non-negligible number of UTCr problems are due to errors in the submitted clock data; e.g. clock data in error in the UTCr report and corrected in the UTC report, or corrected by BIPM staff during the UTC computation.
- In these tests, it is impossible to exactly re-create the UTCr computations that yielded the published values of [UTCr-UTC(k)]. Rather the set of 64 weekly computations using the standard UTCr algorithm provides a “UTCrlike” scale, which has the same properties as UTCr in the comparison to UTC. [UTC-UTCrlike] serves as a reference to compare the tests results.

3.1) Test of changing the date charnière

In the first test, named "Lasteh", we vary the interval of computation to always use the most recent UTC date as charnière. This implies that the UTCr computation interval may vary from 10-15 days (just after the UTC computation) up to 40-45 days (just before the UTC computation) in extreme cases.

3.2) Test of the difference in weighting algorithm

It is complicated to fully implement in UTCr the same weighting algorithm as in UTC. However it is simple to implement a fix that uses the clock variances obtained from the most recent UTC computation to compute UTCr weights "à la UTC".
We combine both tests 3.1 and 3.2 to generate the "ZH+Lasteh" results

3.3) Results of tests

Figure 2 shows, compared to UTC, the reference "UTCrlike" and the two tests "Lasteh" and "ZH+Lasteh". The published UTCr is also shown to indicate its consistency with "UTCrlike".

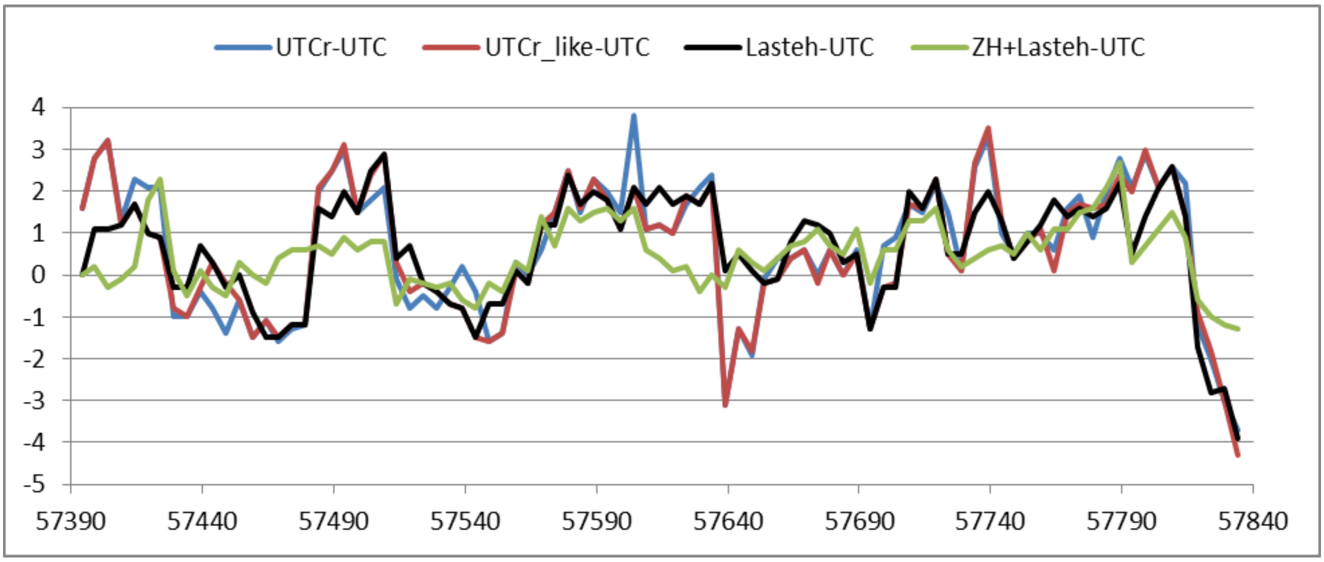

Table 2 summarizes the main statistical results over the 89 UTC dates.

	UTCr-UTC	UTCrlike-UTC	Lasteh - UTC	ZH+Lasteh - UTC
Stdev / ns	1.61	1.57	1.31	0.78
Largest step / ns	5.5	5.3	3.1	2.4

We see that using the most recent UTC date as the beginning of the interval of computation of UTCr ("Lasteh") already improves the consistency of UTCr with UTC. In addition, weighting clocks according to the most recent UTC computation ("ZH+Lasteh") restores the sub-ns consistency with UTC that was observed in 2013.
As a word of caution, these results have been obtained after detection and correction of a small number of errors in clock data reported for UTCr. This performance may not be fully achieved in the true UTCr computation where such detection and correction is not always possible.

4) Proposed plans

- In the short term, change the UTCr method of computation, switching to the approach used in the test "ZH+Lasteh".
- In the long term, the same (new) software should be used for UTC and UTCr.