The Development of a New Kalman-Filter Time Scale at NIST

Jian Yao, Thomas Parker, and Judah Levine
Time and Frequency Division, National Institute of Standards and Technology

ABSTRACT

We report on a preliminary design of a new Kalman-filter Hydrogen-maser time scale at NIST. The time scale is composed of a few Hydrogen masers and a Cs clock. The Cs clock is used as a reference clock, to ease operations with existing data. All the data used in this paper are real measurement data, except mentioned specifically. Unlike most other time scales, this time scale uses three basic time-scale equations, instead of only one equation. Also, this time scale can detect a clock error (i.e., time error, frequency error, or frequency drift error) automatically. A frequency step of 6.8×10^{-15} s/s (which typically corresponds to ~ 2^0C temperature change of a H-maser chamber) in a clock only leads to ~ 2.5 ns change in the time scale in 100 days, thanks to the advanced error-detection technique. A frequency-drift step of 5.4×10^{-21} s/s^2 in a clock only leads to ~ 11 ns change in the time scale in 100 days. These features make the new time scale stiff and less likely to be affected by a bad clock. Tests show that the time scale deviates from the UTC by less than ± 5 ns for ~100 days, when the time scale is initially aligned to the UTC and then is completely free running. Once the time scale is steered to an external frequency standard (such as a Cs fountain), it can maintain the time with little error even if the external frequency standard stops working for tens of days. At NIST, we have the Cs fountain running in 2015 September (23 days), 2015 December (14 days) and 2016 February (18 days). Although the Cs fountain runs for only 55 days in total, the time scale steered to the Cs fountain has a deviation of less than 4 ns (peak-to-peak) from the UTC, during MJD 57265 – 57500 (235 days). This can be helpful when we do not have a continuously-running fountain, or when the continuously-running fountain accidentally stops, or when optical clocks run occasionally.

I. INTRODUCTION

At timing laboratories, we want to have a time scale which is accurate, precise, and reliable [1]. By definition, a Cs fountain is naturally accurate. However, a Cs fountain is typically noisier than a Hydrogen-maser (H-maser) for an averaging time of less than several days. Thus, some people have formed a time scale composed of a continuously-running Cs fountain and an H-maser [2]. The short-term output of the time scale is determined by the H-maser, while the long-term output of the time scale is determined by the Cs fountain. In this way, the time scale is both accurate and precise. To improve reliability, some people add an additional back-up H-maser [2], to avoid the consequence of an H-maser failure.

In this paper, we propose a different architecture of a time scale. The time scale is composed of a Cs fountain and an ensemble of H-maser clocks. The ensemble of H-maser clocks, forming a free-running time scale using Kalman filter, is steered to the Cs fountain. Since we have a few H-masers, the short-term stability of the time scale is better than a single H-maser due to averaging. Also, it is easier to detect and mitigate unobvious clock errors (such as a small frequency step, or a small frequency-drift step), which improves the reliability of the time scale.

In Section II, we will discuss the basic principle of the time scale. Section III will test the performance of the free-running time scale. In Section IV, we will steer the free-running time scale to a Cs fountain and form the final time scale.

II. KALMAN-FILTER H-MASER TIME SCALE

The Kalman filter has been widely used to do prediction based on real-time measurements. The fundamental assumption in Kalman filter is that the noise type in both measurement and physical quantity is white. Although the clock behavior is only approximately white, the Kalman filter still works well [3].

Here, we build a new Kalman filter time scale, following Chapter 4 of [4]. The time scale is composed of a few H-masers and a commercial Cs clock. The Cs clock is used as a reference clock to ease operation with existing data. Each clock is characterized in terms of three parameters: time, frequency, and frequency drift. Although the Kalman filter is straightforward, we do not have measurements of each clock in the current situation. Instead, we only have measurements of the time difference between each clock and the reference Cs clock. Thus, we have three degrees of freedom: the reference clock's time, frequency, and frequency drift. To limit the freedom of the reference clock's time, we require that the weighted sum of the differences between the current time estimates and their predicted values is zero, which is called the basic time-scale equation [5-7]:

$$\sum_{i=1}^{N} w_i x_i(k+1) = \sum_{i=1}^{N} w_i x_i(k+1|k), \quad (1)$$

where w_i is the weight of clock *i* based on the statistics of prediction error, $x_i(k+1)$ is the time estimate of clock *i* at epoch *k*+1, and $x_i(k+1|k)$ is the predicted time of clock *i* at epoch *k*+1 based on the clock state at epoch *k*.

However, in many time scale algorithms, little effort is made to limit the freedom of the reference clock's frequency and frequency drift. The freedom of the reference clock's frequency can be viewed in this way: suppose the reference clock's frequency is changing in a specific pattern, our measurement results are still the same as long as other clocks' frequencies change accordingly. Thus, the reference clock's frequency can be any value, and therefore is a degree of freedom. The freedom of the reference clock's frequency drift can be viewed similarly. To limit the freedom of the reference clock's frequency and frequency drift, we need to extend the basic time-scale equation to three equations:

$$\begin{aligned} \sum_{i=1}^{N} w_{i\_x} x_i(k+1) &= \sum_{i=1}^{N} w_{i\_x} x_i(k+1|k), \\ \sum_{i=1}^{N} w_{i\_f} f_i(k+1) &= \sum_{i=1}^{N} w_{i\_f} f_i(k+1|k), \\ \sum_{i=1}^{N} w_{i\_d} d_i(k+1) &= \sum_{i=1}^{N} w_{i\_d} d_i(k+1|k), \end{aligned} \quad (2)$$

where *x* is time, *f* is frequency, and *d* is frequency drift. We need to have three sets of weights, w_{i_x} for time, w_{i_f} for frequency, and w_{i_d} for frequency drift. By limiting the freedom of the reference clock's frequency and frequency drift, the time scale has better long-term stability, because we give higher weights to those clocks that are more stable in frequency and frequency drift. As an extreme example, if a clock frequently has a big frequency drift variations or some frequency drift jumps, it will pull the whole ensemble significantly if we only use Eq. (1). However, if we use Eq. (2), we can give this clock a very small weight in w_{i_d}, so that it has tiny impact on the long-term stability of the time scale. We will show a real instance and discuss this issue further in Section III.

In addition to the above feature, the new Kalman-filter time scale has another new feature in comparison to the AT1 algorithm [5]. It can detect a clock time/frequency/frequency-drift error automatically. Thanks to the real-time estimation of time, frequency, and frequency drift, we can find a clock error quickly, and then remove the clock from the ensemble by giving the clock 0 weights in Eq. (2), before the clock pulls the ensemble too much. Also, it updates the statistics of each clock automatically. As an example, the time scale computes the standard deviation of frequency drift and adjusts the weights w_{i_fd} accordingly.

These two features make the new time scale stiff and less likely to be affected by a bad clock. We will test the performance of this time scale in the next Section.

III. PERFORMANCE OF FREE-RUNNING TIME SCALE

Now, we run the time scale for Modified Julian Date (MJD) 56650.0 – 56950.0. For test purposes, we only use four H-masers in the time scale, so that it is less complicated. At 56650.0, we initialize the time scale, by estimating the clock states (time, frequency, and frequency drift) with respect to UTC. In this way, the time scale is well aligned with UTC at the very beginning. Then we let the time scale completely free-run. The result is shown in Figure 1. We can see that after 100 days of free running (i.e., at 56750.0), the time scale is only ~ 3 ns away from UTC. This illustrates that the new time scale is quite stiff. Because the time scale is composed of H-masers and an H-maser can have some small change in frequency drift, the time scale typically has a non-zero frequency drift in the very long term (> 100 days) as indicated in Figure 1. The frequency drift is approximately 2.5×10^{-22} s/s^2, in this case.

Figure 2 shows the frequency drift of H-masers with respect to the time scale. Overall, all 4 H-masers have quite stable frequency drift. The mean values of frequency drifts are -3.5×10^{-22} s/s^2, -34.8×10^{-22} s/s^2, -167.8×10^{-22} s/s^2, and -7.4×10^{-22} s/s^2, for ST0005, ST0006, ST0007, and ST0022, respectively. However, we can still see some abnormal behaviors in frequency drift, from time to time. For example, there is a spike in ST0005 around MJD 56868.7. Checking the ST0005 chamber temperature data (see Figure 3), we find a temperature spike around MJD 56868.6. This confirms that our estimation of frequency drift is done properly and it can reflect the actual physical change of a clock. For another example, there is a change in the red curve of Figure 2 (ST0005) starting from around MJD 56915. At the same time, there is also a quick change in frequency of ST0005. By checking the chamber temperature for ST0005, we find that this clock change corresponds to the temperature change from 25.3 ^{0}C to 25.2 ^{0}C at around 56914. We should mention that the change in the chamber temperature is small (only 0.1 ^{0}C) and there is some delay for the whole H-maser to change to 25.2 ^{0}C. Also, the tiny frequency change of ST0005 is somehow correlated with the time noise. Thus, we practically observe 1-day latency of the clock error detection. As the last example, there seems some abnormal behavior in ST0006 at around MJD 56881. We do not observe any environmental change of this clock. This behavior may come from the mechanical or electronic components of the clock.

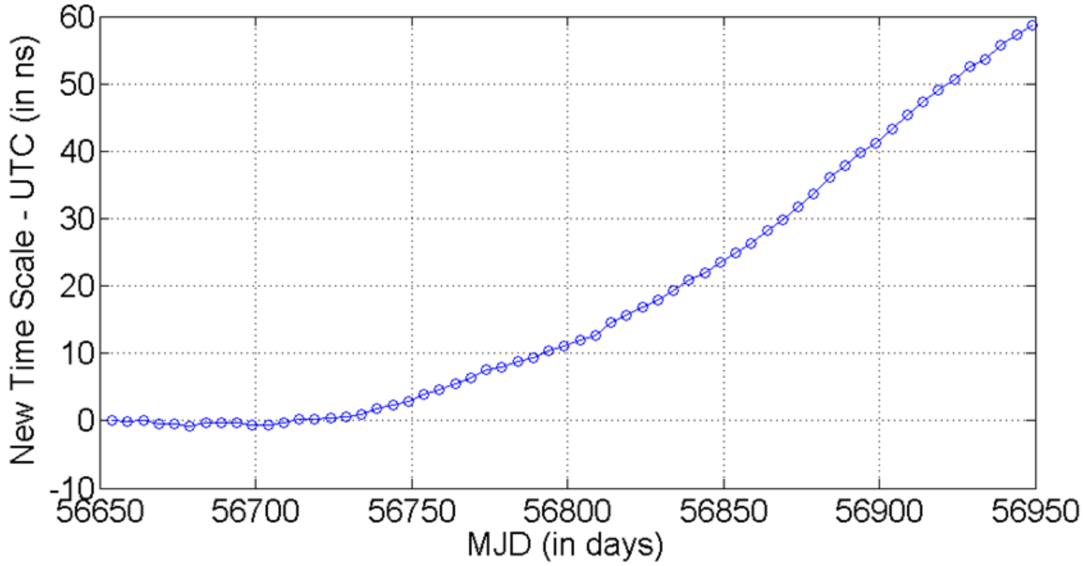

Figure 1. Free-running Kalman-filter H-maser time scale with respect to UTC.

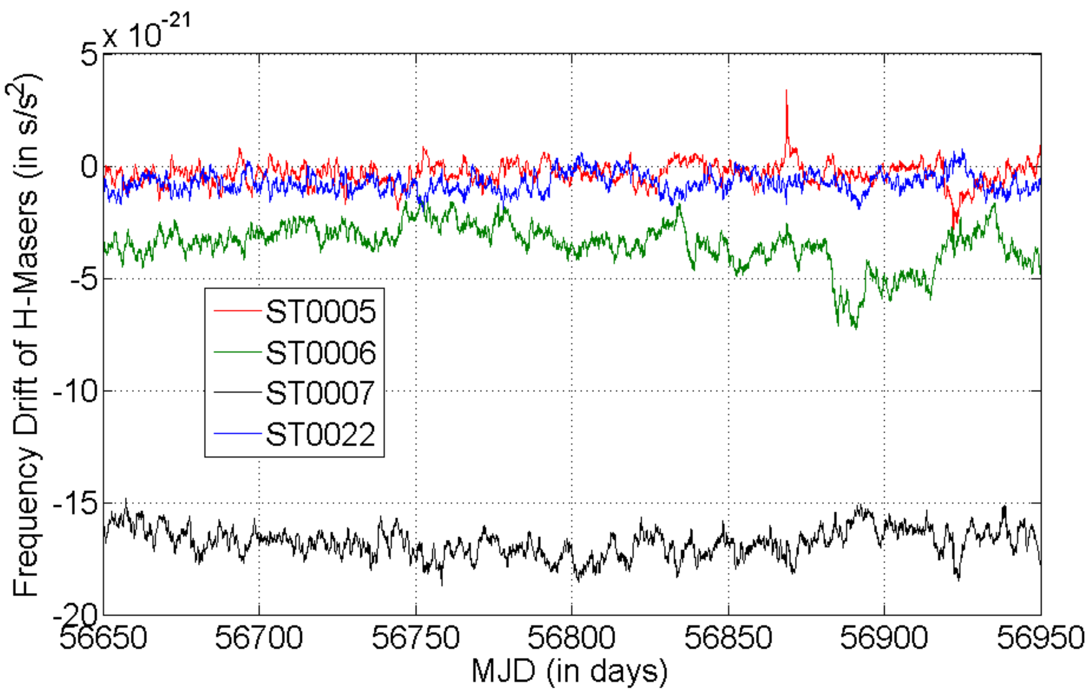

Figure 2. Frequency drift of H-masers.

As we have mentioned in Section II, once the clock error is detected, we give the clock 0 weights in the three equations of Eq. (2), no matter what the physical reason is. In this way, the time scale is not affected by a bad or abnormal clock. If the clock error disappears later, it gets back the weights it deserves.

From Figure 2, it is not easy to guarantee an H-maser to be always good. If we steer a single H-maser to the Cs fountain, it is quite likely to have a large timing error when the fountain stops working. In contrast, if we have an ensemble of H-masers, we can detect a bad clock and mitigate its impact. In this way, we may have still a tiny timing error, but we can avoid a big timing error.

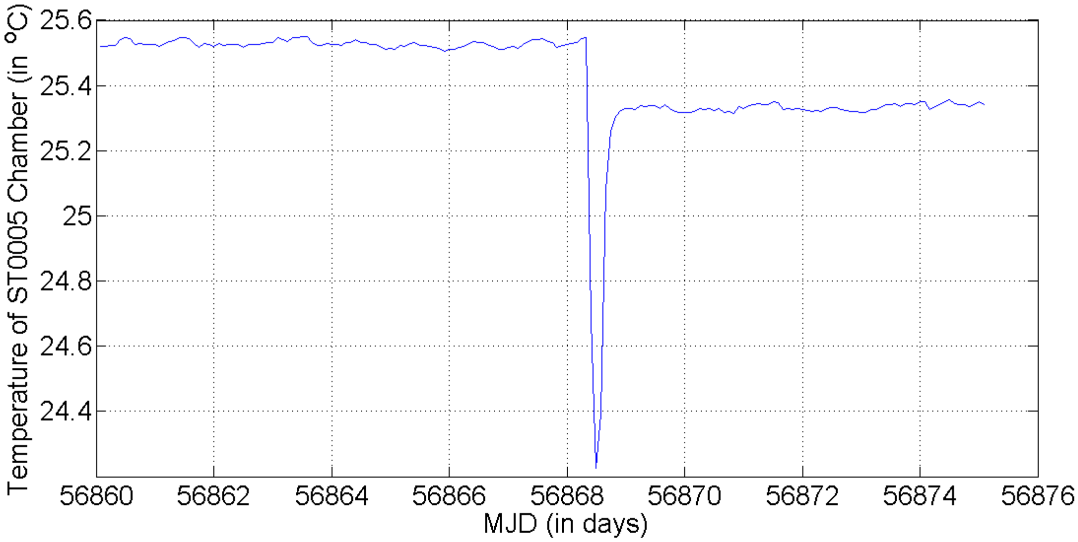

Figure 3. Chamber temperature for ST0005.

Next, we simulate a frequency jump and a frequency-drift jump in a clock, and see how the time scale output is affected by these jumps. Ideally, we want to detect the frequency/frequency-drift error and remove the clock from the ensemble immediately when the error occurs. Practically, the filtering process, kind of a low-pass filter, will lead to some time delay of reflecting the actual jump in frequency or frequency drift (because a jump contains all frequencies in the frequency domain). Also, we determine if there is a frequency/frequency-drift error, based on the criteria that the frequency/frequency-drift at the current epoch is greater than 4 times the standard deviation of the frequency or frequency drift. Thus, it is difficult to completely eliminate the impact of the frequency/frequency-drift error of a clock. Nevertheless, by giving the clock 0 weights in Eq. (2) once we determine that the clock is bad, we can mitigate the impact significantly.

Figure 4 shows an example. We artificially add a frequency jump of 6.8×10^{-15} s/s in ST0006 at MJD 56700.0. This frequency jump corresponds to around 2 ^{0}C H-maser chamber temperature change [8]. Because of this frequency jump, ST0006 is supposed to be 147 ns away from what ST0006 should be, at MJD 56950.0. If we assume that ST0006 has the same weight as other H-masers, a conventional time scale will be pulled by 36.7 ns at MJD 56950.0. This is the result of a conventional time-scale algorithm. From the red curve in Figure 4, we can see that the new time scale is pulled by only ~ 3.5 ns at MJD 56950.0. This is because we give ST0006 0 weights in Eq. (2) once we observe the abnormal behavior of ST0006. Thus, we mitigate the impact of a frequency jump by more than 90 %.

Figure 5 shows an example of a frequency-drift jump. Similarly, we artificially add a frequency-drift jump of 5.36×10^{-21} s/s^2 in ST0006 at MJD 56700.0. As we have seen in Figure 2, all the 4 clocks during the whole 300 days do not have such a large change in frequency drift. Thus, a jump of 5.36×10^{-21} s/s^2 is rare and not likely to happen in practice. This simulation result shows the performance of the new time scale, in such an extreme situation. ST0006 is supposed to be 1250 ns away from what ST0006 should be, at MJD 56950.0. This leads to the 25 % × 1250 ns = 312.5 ns change of a conventional time scale at MJD 56950.0, conventionally. From the red curve in Figure 5, we can see that the new time scale is pulled by only ~ 23.6 ns at MJD 56950.0, because ST0006 has 0 weights in Eq. (2) after the abnormal behavior is detected. Similar to the situation in the frequency jump, we mitigate the impact of a frequency-drift jump by more than 90 %.

From the above analysis, the new time scale is much stiffer than a conventional time scale, thanks to the three basic time-scale equations (i.e., Eq. (2)) and the advanced error-detection technique. Thus, it is nearly immune to abnormal clock behaviors, especially when the clock error is not large.

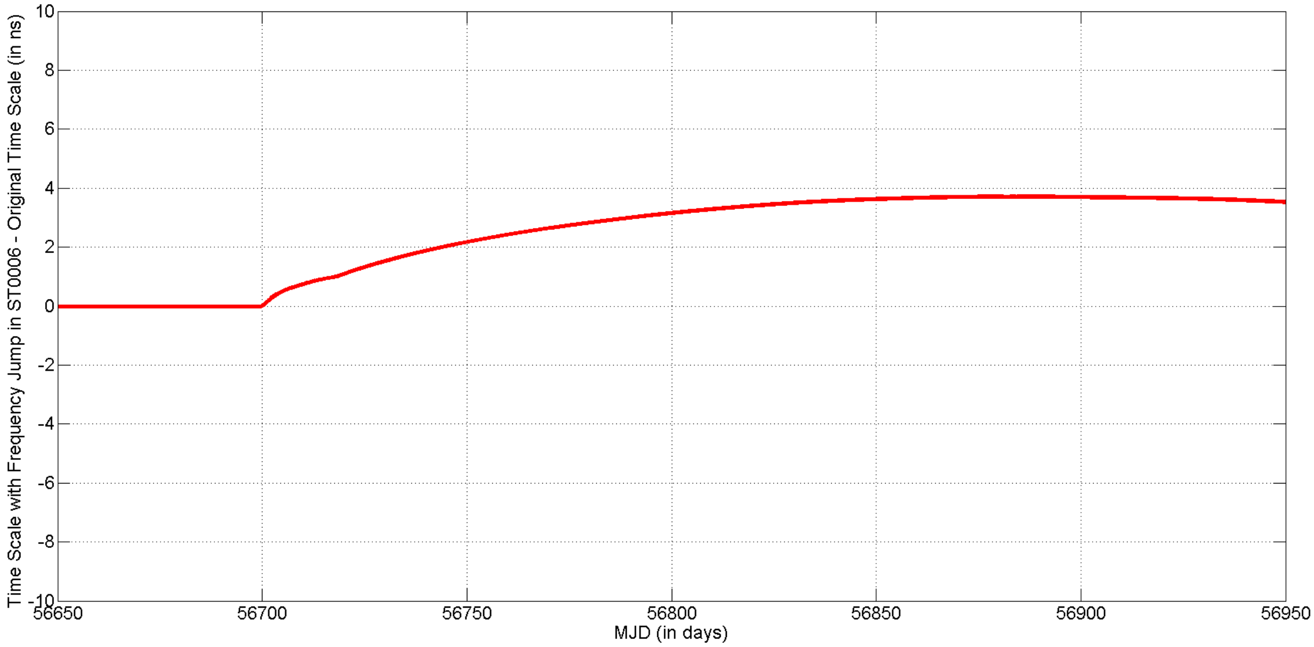

Figure 4. Response of the time scale to a frequency jump. We artificially add a frequency jump of 6.8×10^{-15} s/s in ST0006 at MJD 56700.0.

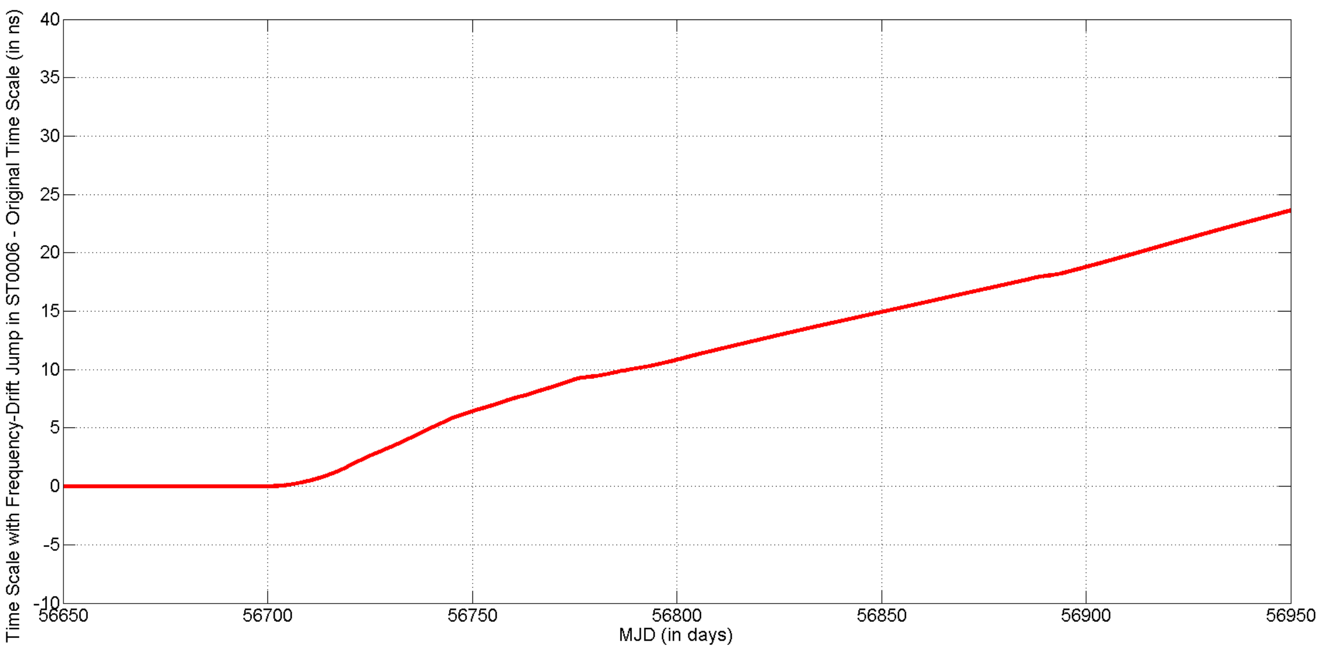

Figure 5. Response of the time scale to a frequency-drift jump. We artificially add a frequency-drift jump of 5.36×10^{-21} s/s^2 in ST0006 at MJD 56700.0.

IV. STEERING TIME SCALE TO CS FOUNTAIN

In Section III, we have demonstrated that the new time scale is stiff and cannot be easily pulled by a bad clock. Thus, it can serve as a good flywheel. If it is steered to a Cs fountain, the final output is accurate, precise, and reliable.

At NIST, we ran the NIST F1 Cs fountain during MJD 57264.9 – 57287.8, 57360.0 – 57374.0, and 57420.8 – 57439.0. We first run a free-running time scale during MJD 56980.0 – 57500.0, and then steer it to the Cs fountain (Figure 6). Note, the steering algorithm corrects both the frequency and the frequency drift of the free-running time scale, using the Cs fountain.

The free-running time scale is composed of six H-masers and a reference commercial Cs clock. The H-masers are ST0005, ST0006, ST0007, ST0010, ST0012, and ST0022. At MJD 56980.0, the time scale is well aligned with UTC. The blue curve shows the result of the free-running time scale (note: the blue, black, green, and red curve are the same during 56980.0 – 57264.9). We can see that the time scale has a very good match (< ±1 ns) with UTC during MJD 56980.0 – 57080.0. This again confirms that the time scale is good at keeping time for approximately three months once it is initially aligned with the UTC or a Cs fountain. After MJD 57080.0, the time scale starts to have a tiny positive frequency drift. Because of this frequency drift, the time scale starts to walk off from UTC, and at ~ MJD 57230, it is ~ 10 ns away from UTC. Then its frequency drift slowly becomes negative, and at MJD 57500, it is ~ −15 ns away from UTC.

Next, we steer the time scale to the first Cs fountain period (i.e., MJD 57264.9 – 57287.8). The result is shown by the black curve. Note, the black curve is the same as the red curve during 56980.0 – 57360.0. By comparing the black curve with the blue curve during MJD 57264.9 – 57500.0, we can find that because of using the Cs fountain, the frequency drift in the free-running time scale is removed. Also, the big frequency offset at 57264.9 in the blue curve is corrected by the Cs fountain. The black curve walks off from the UTC by less than ~ 5 ns, for the first 90 days (i.e., 57287.8 – 57377.8) after the time scale is steered to the fountain.

The green curve is the result of steering the free-running time scale to the two Cs fountain periods (i.e., MJD 57264.9 – 57287.8, and MJD 57360.0 – 57374.0). Note, the green curve is the same as the red curve during 56980.0 – 57420.8. Again, the green curve walks off from the UTC by less than ~ 5 ns, for the first 90 days (i.e., 57374.0 – 57464.0) after the time scale is steered to the fountain.

Last, the red curve shows the result of steering the free-running time scale to the three Cs fountain periods (i.e., MJD 57264.9 – 57287.8, 57360.0 – 57374.0, and 57420.8 – 57439.0). During 57264.9 – 57500.0, the red curve has a change of less than 4 ns, with respect to the UTC. Figure 7 shows the frequency-stability analysis of the free-running time scale and the time scale using three fountain periods, during 57264.9 – 57500.0. From the blue curve and the red curve in Figure 6 and Figure 7, the improvement of steering to the Cs fountain is significant in both time and frequency stability. Remember that we only run 55 days of Cs fountain, over the whole 235 days. In other words, with Cs fountain running for less than 25 % of the time, we successfully maintain the time within 4 ns. This opens a new possibility for designing a time scale. To achieve less than 5 ns error, we only require the Cs fountain to be run from time to time, instead of having a continuously-running Cs fountain. This can be helpful when we do not have a continuously-running fountain. Also, a continuously-running fountain may accidentally stop. By using our new time scale algorithm, we allow ~ 3 months for repairs. During the three months, the timing error is at most 5 ns, according to the tests in this paper. Most importantly, nowadays, NIST has many excellent optical clocks under active development. However, it is difficult to run optical clocks for a long time (> a few days), due to engineering obstacles. Because of this, many people believe that the optical clock cannot contribute to the time scale for the time being. However, from this example, we can imagine that occasional optical-clock running (e.g., 1 hour every week) could be very helpful to the long-term (e.g., > 15 days) performance of the time scale. Thus, instead of waiting until all engineering obstacles are solved, we should consider incorporating an optical clock into the time scale now.

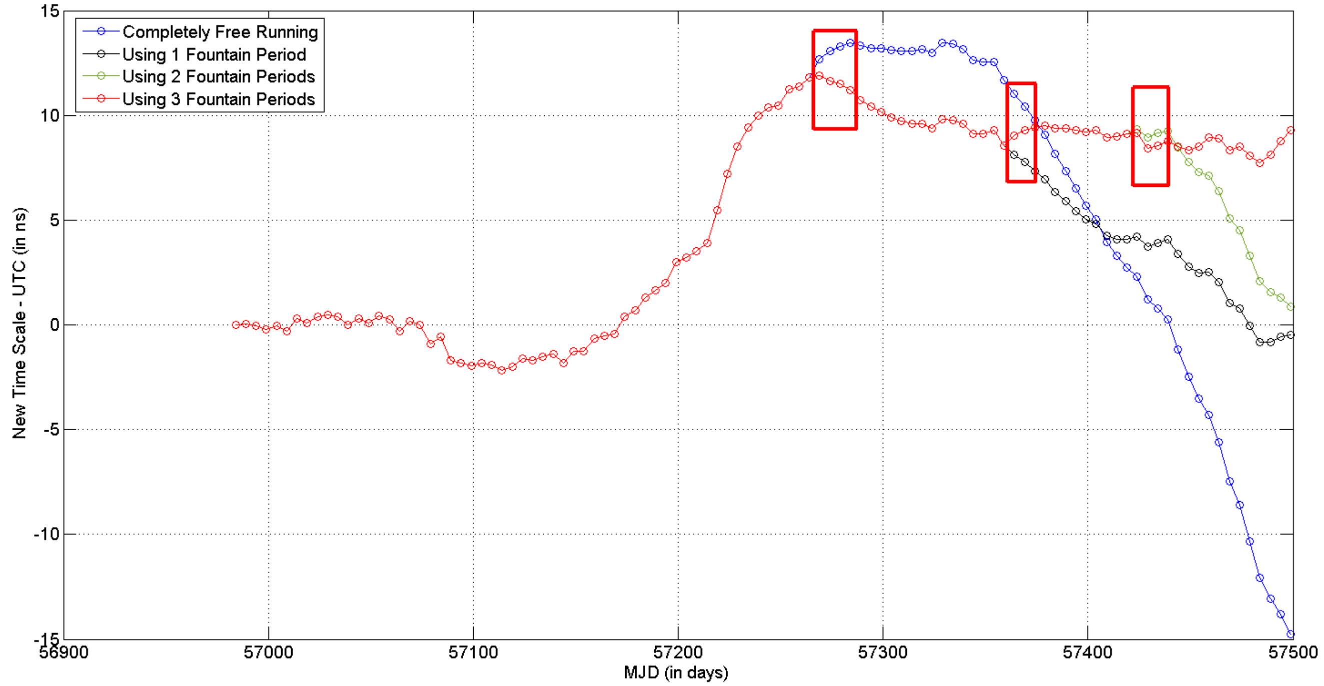

Figure 6. Steering the free-running time scale to the NIST F1 Cs fountain. Note, the blue curve is the same as the red curve during MJD 56980.0 – 57264.9; the black curve is the same as the red curve during MJD 56980.0 – 57360.0; the green curve is the same as the red curve during MJD 56980.0 – 57420.8.

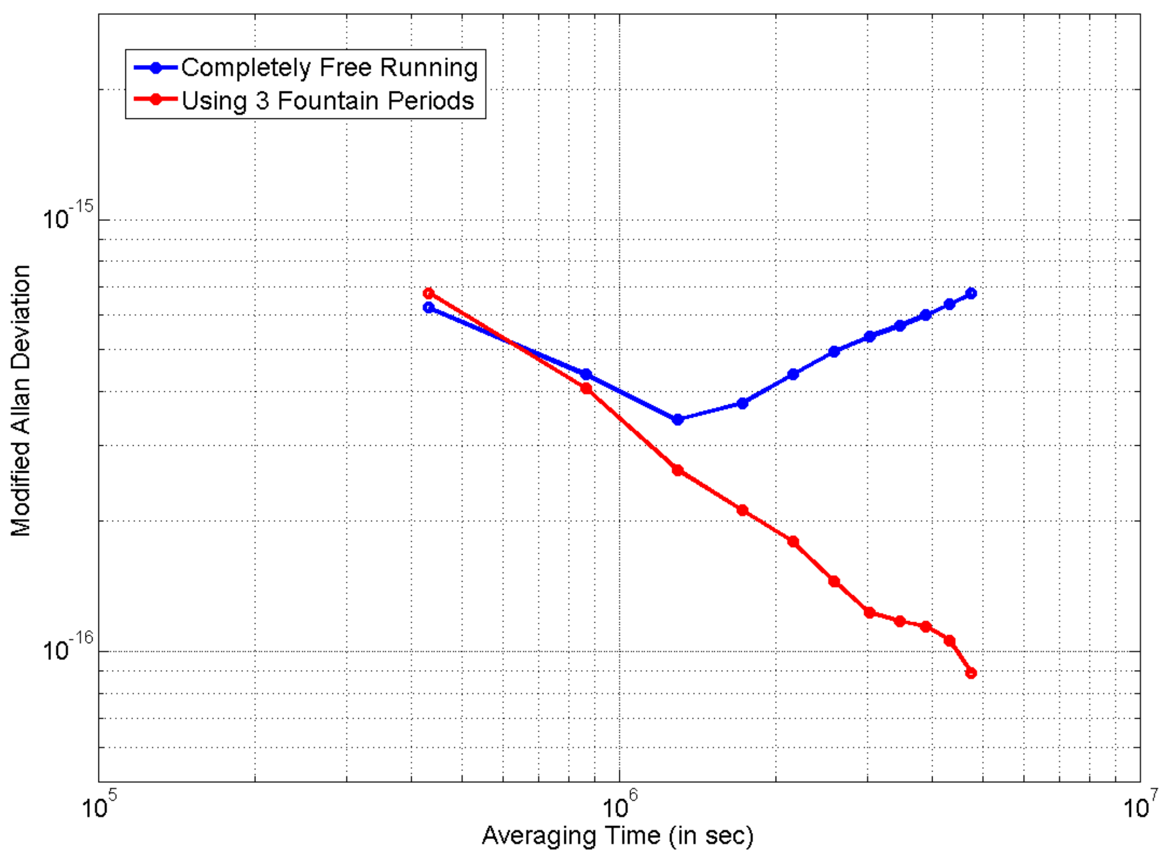

Figure 7. The frequency stability of the time scale using three fountain periods (red curve), versus the frequency stability of the free-running time scale (blue curve).

V. SUMMARY

This paper discusses a new Kalman-filter Hydrogen-maser time scale at NIST. The new time scale is designed to be stiff and nearly immune to an abnormal clock behavior. Tests show that when the time scale is initially aligned to the UTC and then is free-running, it deviates from the UTC by less than ± 5 ns for ~100 days. Thus, it is a good flywheel. Once the time scale is steered to a Cs fountain, it can maintain the time with little error (< 4 ns) even if the Cs fountain stops working for tens of days. Thus, an occasional-running Cs fountain is good enough to guarantee the accuracy of the time scale. This time scale algorithm also makes incorporating an occasional-running optical clock into the clock ensemble possible.

ACKNOWLEDGMENTS

The authors thank those people maintaining the NIST time scale, such as Stefania Romisch, Josh Savory, Jeff Sherman, Trudi Peppler, etc. The authors also thank Neil Ashby, Jeff Sherman, and Mike Lombardi for their comments.

REFERENCES

[1] L. A. Breakiron, "Timescale algorithms combining Cesium clocks and Hydrogen masers," Proc. 23rd Annual Precise Time and Time Interval (PTTI) Systems and Applications Meeting, pp. 297-302, 1991.
[2] G. D. Rovera, S. Bize, B. Chupin, J. Guena, P. Laurent, P. Rosenbusch, P. Uhrich, and M. Abgrall, "UTC(OP) based on LNE-SYRTE atomic fountain primary frequency standards," Metrologia, 53, pp. S81-88, 2016.
[3] L. Galleani, and P. Tavella, "On the use of the Kalman filter in timescales," Metrologia, 40, pp. S326-S334, 2003.
[4] A. Gelb, "Applied optimal estimation," MIT Press, 1974.
[5] J. Levine, "The statistical modeling of atomic clocks and the design of time scales," Rev. Sci. Instrum. 83, 021101, 2012.
[6] C. A. Greenhall, "Kalman plus weights: a time scale algorithm," Proc. 33rd PTTI Meeting, pp. 445-454, 2001.
[7] S. R. Stein, "Advances in time-scale algorithms," Proc. 24th PTTI Meeting, pp. 289-302, 1992.
[8] T. Parker, "Environmental factors and Hydrogen maser frequency stability," IEEE Transactions on UFFC, vol. 46, pp. 745-751, 1999.