Invited Review Article: The statistical modeling of atomic clocks and the design of time scales

Judah Levine
Time and Frequency Division and JILA, National Institute of Standards and Technology and University of Colorado, Boulder, Colorado 80305, USA

(Received 20 June 2011; accepted 28 August 2011; published online 23 February 2012)

I will show how the statistical models that are used to describe the performance of atomic clocks are derived from their internal design. These statistical models form the basis for time scales, which are used to define international time scales such as International Atomic Time and Coordinated Universal Time. These international time scales are realized by ensembles of clocks at national laboratories such as the National Institute of Standards and Technology, and I will describe how ensembles of atomic clocks are characterized and managed. [http://dx.doi.org/10.1063/1.3681448]

I. INTRODUCTION

A time scale is a procedure for assigning names to consecutive instants. The foundation of every time scale is a periodic event that is used to define the basic time interval. The time at any epoch is then the number of periodic events that have elapsed since the epoch that was chosen as the origin of the scale. The time scale may also include a mechanism for interpolation between occurrences of the periodic events that are the basis of its definition.

Astronomical observations have always been an important part of every time scale.[1] The solar day and the solar year have been used since antiquity, and many societies also used observations of the periodic variation in the positions of moon and the stars to measure time intervals.[2] The periods of these events are not commensurate, and calendars, which are often based on several astronomical events, are complex as a result. Clocks in general, and the definition of the second in particular, were originally thought of as a method of interpolating between consecutive astronomical events, which were considered the primary definition of time and time interval.

Contemporary time scales have reversed the definition, and use the second as the primary unit of time interval;[3] astronomical periods are now expressed in terms of the definition of the length of the second. In this paper, we will focus on how the second is defined in principle and on how it is realized in practice. However, astronomical events still play a central role in timekeeping, so that every time scale must still address the complexities of the calendar. I will not deal with these complexities nor will I discuss how the various time scales that are currently used address them.[4] These are complicated topics, and a detailed discussion of them would take us far outside the scope of this paper.

II. THE DEFINITION OF THE SECOND

The second is currently defined as the time interval realized by counting 9 192 631 770 cycles of the frequency associated with the hyperfine transition in the ground state of an unperturbed cesium 133 atom.[5] The length of the second defined in this way was chosen to be roughly equivalent to the previous astronomical definitions.[6]

Since the measurement of the hyperfine transition perturbs the atoms, the magnitude of the perturbation must be estimated and removed. A primary frequency standard is a device that is constructed so as to minimize the perturbation resulting from the measurement process and to facilitate estimating the magnitude of the residual perturbation so that it can be removed from the data.

A number of groups have proposed changing the definition of the second. For example, advances in measurement technology have made it feasible to use an optical frequency to define the second rather than the microwave frequency of the current definition.[7] A higher frequency for the definition of the second is better in principle, but there are many technical problems to be overcome before an optical transition could be used. Although these discussions are in a preliminary stage, it is likely that the definition of the second will be modified within a decade. However, for reasons that I will discuss later in this section, it is unlikely that this change in the definition will eliminate the need for time scale algorithms and measurement protocols.

A practical realization of the second must address two types of problems. The first is a problem of principle: the frequency measured by an observer depends on the velocity of the receiver with respect to the source (the Doppler effect), and on any difference in gravitational potential between the source and the observation point (the gravitational "redshift"). These effects must be included in any definition, because primary frequency standards are located on the rotating Earth, and sources and detectors may be at different gravitational potentials even if they are at rest with respect to each other.

The original concept that the second should be defined based on the transition frequency of an unperturbed cesium could be extended to require that the atom be at rest at a location where the gravitational potential is 0. It is not possible or practical to realize this purist definition, and some sort of compromise is needed. The current definition of the second is based on the frequency of cesium atoms located at rest relative to the rotating geoid, which is an equipotential

surface of the gravitational potential that is roughly equivalent to mean sea level.[8] Frequency standards not located on the geoid (in Boulder, Colorado, for example, about 1600 m above the geoid) must be corrected for this height offset. The correction in fractional frequency is approximately 2×10^{-16}/m relative to an observer on the geoid. Frequency standards located on a moving platform (such as a satellite) must also be corrected for the Doppler shift of the frequency when the signal is observed by an observer at rest on the geoid.

Although the corrections themselves can be quite large relative to the other contributions to the uncertainty of the frequency of the primary standard, the uncertainties in these corrections are not significant at present. However, the importance of these uncertainties will grow as more accurate frequency standards are developed, and deterministic and stochastic variations in the position of a frequency standard with respect to the geoid may ultimately limit the realization of the second. For example, the gradients in the lunar and solar gravitational potentials produce approximately diurnal and semi-diurnal tides in the solid Earth that have an amplitude of about 0.3 m. These tidal effects introduce corresponding diurnal and semi-diurnal variation in the frequency of a primary standard when measured by an observer at a distant location on the surface.

The second difficulty is a problem of implementation rather than one of principle. A primary frequency standard is designed to minimize the effects of various systematic frequency offsets between the frequency realized by the device and the frequency of the unperturbed atom. In addition, the residual frequency offsets must be estimated. It is difficult to operate a primary frequency standard continuously as a clock and simultaneously satisfy these requirements, and many primary frequency standards operate only intermittently with the data used to calibrate a continuously running clock that is used to disseminate the time and frequency information.

The continuously running clock, on the other hand, must be designed to be reliable, with a frequency stability that is optimized for the interval between calibrations by the primary device. The accuracy of this clock is less important, because it will be calibrated by the primary device. Since a single clock may fail with little warning, a robust design generally uses the data from the primary frequency standard to calibrate an ensemble of clocks rather than just a single device. Thus, the need for a time scale, a process for defining the time of an ensemble of clocks in a statistically robust way, is born.

The need for a time scale as a flywheel between evaluations by a primary frequency standard is unlikely to disappear as newer, more accurate primary frequency standards are developed. In fact, the opposite may occur because the laboratory prototypes of many advanced primary standards operate only intermittently and with a relatively short duty cycle. This will require an even more stable flywheel time scale that can be used to transfer the accuracy of the primary device to users.

The remainder of the paper is devoted to a discussion of how such hardware time scales are currently realized. As we will see in the following discussion, there is no obviously best implementation for a time scale – every implementation has some weakness, and is therefore a compromise among competing and incompatible goals.

III. STATISTICAL TOOLS AND THE DEFINITION OF SYMBOLS USED IN THE TEXT

In the following discussion, we will present a number of statistical tools that are often used to characterize time and frequency data. The most important concept is the Allan (or two-sample) deviation, which provides an estimate of the RMS variation of the average frequency of a clock (relative to some other device) measured over two (generally consecutive) equal time intervals. Since the time difference of the clocks (rather than their frequency difference) is often the primary observable, the Allan deviation equivalently provides an estimate of the second-difference of the time difference of a clock with respect to another device.

The Allan deviation is usually expressed as a function of the averaging time between the time-difference observations used to compute this quantity. (Alternatively, it is the averaging time used to estimate the frequency difference.) The Allan deviation is a root-mean-square quantity and does not provide any information on the distribution of the data that are used to compute it. Therefore, it is much less useful when the data contain deterministic variations or steps in time or frequency. In addition, it is a measure of frequency stability rather than time or frequency accuracy, since constant time or frequency differences between the two devices cancel in the computation of the differences and do not contribute to the result.

The Allan deviation can also be used to estimate the time deviation of a clock given its frequency variation. The time deviation is typically called TDEV. The dependence of the Allan deviation and TDEV on the averaging time used to compute them (specifically the slope of a log-log plot of the variance as a function of the averaging time) provide information on the characteristics of the variance of the data.

In many cases, the dependence of the Allan deviation on the averaging time can be approximated by a polynomial function with only a few terms. The terms in the polynomial with negative powers of the averaging time tend to be most important at shorter averaging times, while the terms with positive powers tend to dominate at longer times. Thus, the log-log plot of the variance as a function of averaging time typically has a "U" shape, which can be approximated by a series of straight-line segments. The bottom of the "U" is often called the "flicker floor" because it is the domain of averaging times where an increase in averaging time does not produce a corresponding improvement in the stability of the device. Increasing the averaging time beyond this domain degrades the stability. From the physical point of view, the flicker floor is the point where the simple stationary, random-noise model of the time differences is no longer an accurate description of the data. While the variance of the data is no longer characterized by a simple stationary random-noise model in this domain, it cannot be characterized by a deterministic variation either. From the perspective of Fourier frequency, the noise in this domain is increasingly "red" in character. The increase in the power spectral density of the noise at low Fourier frequencies means that relatively short segments of data may appear to

TABLE I. An explanation of all of the symbols and the notation used in the text.

x	The time difference between a clock and a reference in units of seconds
y or f	The frequency difference between a clock and a reference. The symbol y is in units of seconds/second (dimensionless) and f is the frequency in Hz.
D	The frequency aging between a clock and a reference in units of seconds^{-1}
	Upper case values are measurements or administrative parameters, lower case values are estimated or calculated values
t	A particular instant of time in units of seconds
τ, Δt	A time interval between measurements or calculations, measured in seconds
σ	Average prediction error of clock in ensemble calculation
ε	Prediction error of clock in ensemble calculation
ξ, η, ζ, v	Standard deviations of noise terms, assumed to have zero mean
W	Weight of clock in ensemble average
(k^-)	The parameter was evaluated immediately before time t_k
(k+)	The parameter was evaluated immediately after time t_k
MJD	Modified Julian day number. An integer value that is incremented at 0000 UTC every day. It is commonly used in time scale calculations because it eliminates the complexities of time difference calculations when using the more common year-month-day notation. For example, 1 January 2011 corresponds to MJD 55 562.
Subscript j, m, n	An index number identifying a particular clock
Subscript e	The parameter is with respect to the ensemble average
Subscript r	The index of the reference clock for a measurement system
Subscript s	A parameter used for clock steering
Subscript k or 0 or (k)	The parameter was measured or evaluated at epoch t_k or at the origin t_0
Superscript ^	The estimate of a parameter
Superscript $\sim$	The error or uncertainty of the estimate
$\rightarrow$, also boldface	Vector quantity
= over a quantity, also boldface	Matrix
K and K′	Kalman gain matrices
S	Kalman state vector
N	Kalman noise vector
P, Q, C	Kalman covariance matrix
H	Kalman measurement matrix
O	Kalman measurement vector

have a deterministic character, but the appearance is misleading because the deterministic parameters vary depending on the length and epoch of the data set.

Flicker noise is present in all devices at some level. It has no simple physical explanation, and is characterized by the independence of the Allan variance on the averaging time as discussed above or by the equivalent 1/f dependence of the power spectral density on the Fourier frequency of the data. The time-domain and frequency domain descriptions are equivalent. If the power spectral density of the frequency in the flicker domain is given by P_f/f then the Allan variance in this domain is approximately $2 \times \ln(2) \times P_f$.[9] For a more comprehensive review of these concepts, see the previous review article.[10]

Table I, above, contains a listing of all of the symbols used in the text.

IV. THE PROPERTIES OF ATOMIC CLOCKS

In order to understand the design of time scales, it is helpful to understand the properties of the atomic clocks that are used to construct them. Although the details vary over a wide range, all atomic clocks consist of an oscillator whose frequency is stabilized relative to the transition frequency in some atomic (or molecular) system. The separation between the oscillator and the frequency discriminator is particularly easy to see in cesium atomic clocks, and we will use this separation as the model for all atomic clocks. The same general principles will apply even for clocks in which the two functions are not so clearly separable, such as lasers or oscillating hydrogen masers, whose frequencies are determined both by the properties of the atoms and by the resonant cavity used to sustain the oscillations. The sensitivity of the output frequency to the properties of the resonant cavity degrades the theoretical purity of the definition of an atomic clock and introduces practical frequency offsets and aging as we will describe below.

An atomic clock consists of three distinct systems: (1) the "physics package," which produces atoms in the lower state of the clock transition; (2) the "electronics package," which generates the frequency needed to induce the transition in the reference atoms, which detects these transitions, and which then locks the frequency of the generator to the maximum of the transition rate, and (3) the output system, which converts the frequency of the atomic transition to one that is more suitable for comparisons with other devices or for driving clock hardware. Typical frequency outputs are 5 MHz and 1 Hz, but other frequencies can also be used. We will always assume

TABLE II. The frequencies of five nominally identical cesium standards relative to atomic time as maintained at NIST.

Clock	Frequency (s/s)
1	8.71×10^{-14}
2	2.37×10^{-14}
3	8.05×10^{-14}
4	4.12×10^{-14}
5	1.90×10^{-13}

that the statistical properties of the output do not depend on its frequency.

The frequency output of the device is controlled by the atomic transition frequency in first order. However, the transition frequency is perturbed even for a cesium device in which the atoms are interrogated while they are in a beam in a vacuum system. The perturbations are caused by residual electric and magnetic fields, by collisions with other atoms, and by similar effects. The Doppler shift, and the interaction between the atoms and the electromagnetic field that induces the clock transition must also be considered. These perturbations are likely to vary from one frequency standard to the next one (even among nominally identical devices) so that we would expect that even the members of an ensemble of identical devices would have significantly different frequencies. Furthermore, these frequency offsets are likely to change slowly with time, so that it is not possible to calibrate them and remove them once and for all. Table II shows the frequencies of five nominally identical cesium frequency standards with respect to the National Institute of Standards and Technology (NIST) clock ensemble. This variation among identical devices is typical, and these values exhibit both white and random-walk frequency variations as a function of epoch. The fact that all of the devices had positive frequencies at the instant the data were recorded is coincidental. Figure 1 shows the frequency variation of a typical cesium standard with respect to the NIST ensemble.

The electronics package also contributes to the frequency stability of the device. There is shot noise in the detection of the atoms and noise in the control loop that locks the oscillator to the atomic transition frequency. These effects introduce corresponding fluctuations in the output frequency of the device. These fluctuations are generally both more rapid and less predictable that the frequency fluctuations due to the physics package described above. The separation is never perfect, and it is often not possible to deduce the source of the frequency fluctuations from their power spectrum.

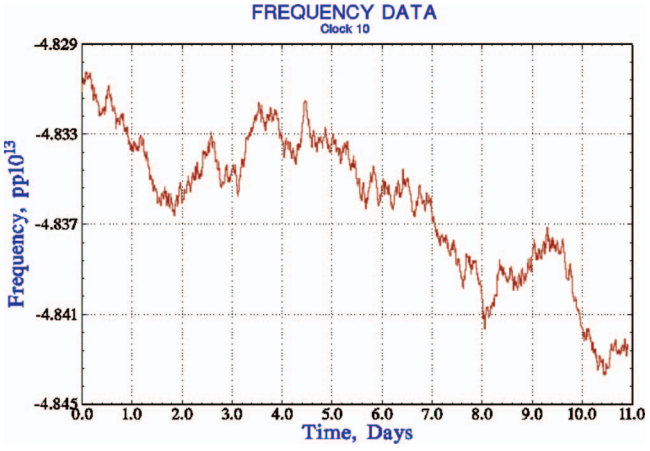

FIG. 1. (Color online) The frequency of a cesium frequency standard with respect to the NIST clock ensemble.

Finally, the counting circuitry that generates clock pulses from the oscillator locked to the atomic transition and any circuit that uses these pulses operate with only a finite signal to noise ratio. The measurement system may also be sensitive to the ambient temperature, so that a measurement of the time interval between consecutive output pulses will show fluctuations that are not related to the frequency fluctuations of the device. It is most important to separate this contribution from the ones described above, since it generates noise in the *time* signals with no corresponding *frequency* fluctuations in the clock.

Although all of these effects result in fluctuations in the consecutive time differences between two nominally identical devices, the characteristics of the fluctuations due to each cause are different, and time scale algorithms are designed to exploit these differences in the characteristics.

V. CLOCK MODELS

If we use the previous discussion as a guide, we are led to characterize a clock in terms of 3 deterministic parameters: x(t), its time difference, measured in seconds, between its output and the output of a second device at epoch t; y(t), its frequency difference, measured as a dimensionless parameter (units of seconds/second) at epoch t, and d(t), its frequency aging at epoch t, measured in units of s^{-1} (seconds per second squared). For simplicity, we consider that the second device used as the reference is perfect, so that the parameters describe the properties of the device we are characterizing. The epoch, t, is also derived from the same perfect reference device.

We model the evolution of these parameters from time t−Δt to t by

$$x(t) = x(t-\Delta t) + y(t-\Delta t)\,\Delta t + \frac{1}{2} d(t-\Delta t)(\Delta t)^2 + \xi, \tag{1}$$

$$y(t) = y(t-\Delta t) + d(t-\Delta t)\Delta t + \eta, \tag{2}$$

$$d(t) = d(t-\Delta t) + \zeta, \tag{3}$$

where ξ, η, ζ are the stochastic contributions to the time difference, frequency offset, and frequency aging, respectively. The stochastic contributions are assumed to be uncorrelated, zero-mean processes with specified variances. For example, we assume that all 3 noise parameters satisfy equations of the form

$$\langle \xi(t) \rangle = \langle \eta(t) \rangle = \langle \zeta(t) \rangle = 0, \tag{4}$$

$$\langle \xi(t)\,\xi(t') \rangle = \xi^2 \delta(t-t'), \tag{5}$$

$$\langle \xi(t)\,\eta(t) \rangle = 0, \tag{6}$$

with relationships similar to Eqs. (5) and (6) for the other variables and for all t and t′. The initial values for the variances of the noise parameters are estimated either from ancillary measurements or from the known characteristics of the clocks and the measurement system, and the algorithm updates these initial estimates on each measurement cycle.

This model is chosen in part because it has an intuitive connection to the physical design of the hardware components and in part because it supports the separation of the noise variance in a natural way, which is also derived from the design of the hardware. However, it is not the only possible choice either for the deterministic or stochastic components of the clock model. In a later section we will discuss a finite impulse response model, which models the current time difference as a linear combination of previous observations. This is a particular type of the more general ARIMA (auto-regressive integrated moving average) models.[11] These methods are often useful for modeling the performance of a single clock, but are much less frequently used to model an ensemble of them, and we will not consider them in detail here.

In general, time scale measurement systems observe x(t), the time difference as a function of epoch on a regular, periodic basis. Some systems also can incorporate occasional, less frequent measurements of y(t) directly into the ensemble calculation. Since the frequency difference data are often received only on an irregular basis, they are often incorporated into the time scale by a separate steering algorithm, which is outside of the normal time-scale calculations. We will discuss steering algorithms below, and limit the current discussion to the much more common configuration of a system that processes only periodic (or nearly periodic) time difference data.

The time scale algorithm predicts the value of x(t) based on previous computations, compares the predicted values to the measurements, and updates the parameters of each clock based on this comparison. The time scale algorithm must separate the contributions of each of the terms on the right side of Eq. (1) for each clock in order to do this. The measurement strategy needed to realize this requirement as accurately as possible depends on many parameters, and I will present a number of detailed examples to illustrate the method.

VI. MEASUREMENT STRATEGIES

Consider a high-performance commercial cesium frequency standard (see Table III) whose time differences are being measured by the use of a dual-mixer measurement system.[12] A dual-mixer system computes the time difference between two frequency standards by measuring the phase difference between the RF outputs of the standards at a frequency such as 5 MHz. The resolution of the phase measurement is enhanced by down-conversion of the radio frequencies to a much lower frequency of order 10 Hz. The down-conversion is implemented by mixing each signal with a frequency that is synthesized with an offset from one of the clocks being measured. The clock used for this purpose is generally the reference clock for the hardware as described below. Dual-mixer systems were originally realized by analog techniques and physical mixers, but the same principle of enhancing the resolution of the time difference measurement after down-conversion can be realized digitally.[13] The increase in resolution that results from the lower frequency must be balanced in practice by low-frequency 1/f noise in the measurement system. The increase in the resolution of the time difference measurement implies a correspondingly increased stability requirement in the hardware before the down-conversion process, since any delay variations in this part of the system affect the measurements in first order.

TABLE III. Characteristics of typical commercial cesium and rubidium standards.[a]

	Cesium standards	
Parameter	Standard device	High performance device
Accuracy	1×10^{-12}	5×10^{-13}
Allan deviation@1s	1.2×10^{-11}	5×10^{-12}
Allan deviation@100 s	2.7×10^{-12}	8.5×10^{-13}
Allan deviation@10^4 s	2.7×10^{-13}	8.5×10^{-14}
Allan deviation@10^5 s	8.5×10^{-14}	2.7×10^{-14}
Allan deviation@5 days	5×10^{-14}	1×10^{-14}
	Rubidium standards	
Parameter		
Initial accuracy	5×10^{-11}	
Frequency aging	5×10^{-11} per month, 5×10^{-10} per year	
Allan deviation@1 s	2×10^{-11}	
Allan deviation@10 s	1×10^{-11}	
Allan deviation@100 s	2×10^{-12}	

[a]Notes:
1. The Allan deviation for cesium standards generally does not improve for averaging times longer than 5 days, so that the last value in the table is the "flicker floor" for the device.
2. Actual cesium devices typically exceed the values shown above by a factor of at least 2 or 3. For example, compare the accuracy specification above with the values in Table II.
3. The Allan deviation for rubidium devices does not improve significantly for averaging times longer than a few hundred seconds, so that the last value in the table is approximately the "flicker floor" of the device.
4. The stability of rubidium standards may be degraded by fluctuations in the ambient magnetic field and, to a lesser extent, by fluctuations in the ambient temperature.

To estimate ξ, the noise in the measurement process, we can use the measurement hardware to measure the time difference between a clock and itself. Figure 2 shows the TDEV of the time difference measured between a clock and itself for two pairs of channels in the measurement system currently used at NIST. When we discuss Kalman filters below, the analysis model will include a contribution to this noise that arises from the measurement hardware and an independent component that originates in the clock itself. The experiment we describe here would be sensitive only to the noise of the measurement hardware. The origin of the noise can be an important distinction in some applications (where the output of the clock is used to drive a frequency multiplier, for example), but it is generally not too important for time scales, since the algorithms almost always use measurement strategies in which this component of the noise is small compared to the other contributions from the variation in the frequency and the frequency aging. A typical result would be that $\xi \leq 10^{-12}$ s for all averaging times less than 1 or 2 days. (Since TDEV is a measure of stability and not an indication of accuracy, it is not necessarily true that the "better"

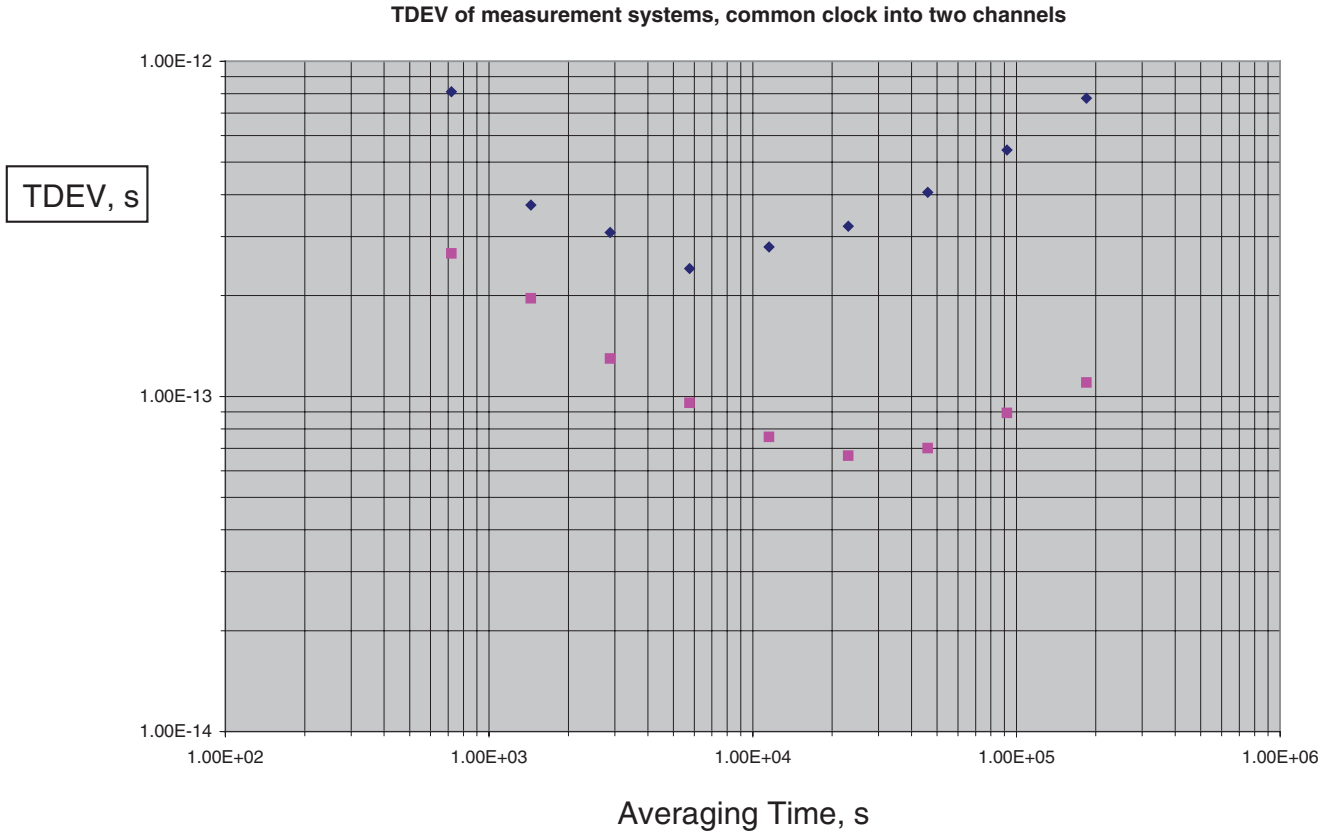

FIG. 2. (Color online) The time deviation of the measurement system, observed by measuring the time difference between the same clock connected to two pairs of channels. The differences in the temperature sensitivities of the two channels and gradients in the room temperature contribute to the variation.

results in Fig. 2 actually represent better hardware. It is equally possible, and probably more likely, that the two "better" channels happen to have the same dependence on ambient temperature changes, which is attenuated by the differencing, while the "poorer" channels have responses that are less well balanced at the particular time we made the measurements. In addition, the magnitude of TDEV for any τ greater than the minimum value is computed by averaging the data in blocks as specified in the definition for the modified Allan variance statistic, which is the basis for TDEV. This is an important point, because the time-difference data are often not used by the time scale in the same way they were averaged to compute the statistic. The estimate of the magnitude of the noise parameter in the previous sentence is a conservative estimate based on these considerations.) The increase in the value of TDEV for longer averaging times is probably an indication of a response to temperature fluctuations or possibly to aging in the measurement hardware. The temperature gradient between measurement channels is pretty small, so that the observed variation is more likely to be caused by a different admittance to temperature by the different channels.

Laboratory time-scale algorithms always operate at the low end of the range of averaging times shown in Fig. 2, but that does not eliminate the longer term changes in the response of the hardware, which are modeled by the algorithm as differential frequency noise or frequency aging. These data do not depend on the details of the clock itself, and so they set the minimum uncertainty of any single time-difference measurement for any type of clock.

The quieter pair of channels in Fig. 2 shows a TDEV that is approximately characteristic of white phase noise at shorter averaging times. Therefore, it is possible to reduce the impact of the measurement noise in these channels by the use of closely spaced measurements of the time difference. For the commercial cesium device we are considering, d $\sim$ 0 and |y| $\leq 2 \times 10^{-13}$. We will choose y $= 2 \times 10^{-13}$ to illustrate the details of the design of the algorithm.

If we measure the time difference x(t) every 0.1 s, the contributions of y and d to any time difference will be smaller than the contribution of ξ. In other words, we are measuring so rapidly that the parameters of the clock do not change between measurements. (Note that in order to satisfy this requirement, it is the frequency offset of the device and not the variance of that quantity that is important here.) The variance in the measured time differences is then mostly due to ξ, and this procedure can be used to provide a real-time estimate of this parameter. Since the variance of the noise parameter ξ changes slowly, examining the average for outliers could also be used as a preliminary error detection method. The distribution of the time difference measurements is approximately a Gaussian random process in this domain, so that the standard deviation of the mean improves as the square root of the number of measurements that contribute to it. Therefore, we can improve the measurement noise by approximately 3.3 ($\sqrt{10}$) by averaging consecutive groups of 10 measurements spaced 0.1 s apart. If we used this method as a preliminary check for outliers, and if we set the threshold for an outlier to be 3$\times$ the standard deviation of the group, then the threshold for

an outlier would be approximately equal to ξ. That is, if the standard deviation of the group of 10 measurements exceeds our estimate of the expected variance for one of them, then the ensemble is presumed to contain an outlier. The usual strategy is to examine the group for the measurement that has the greatest offset from the mean and assume that it is the outlier.

The full improvement estimated in the previous paragraph may not be realized in practice, since the standard deviation of the mean of the measurements is comparable to the contribution of the frequency offset, y, in this case, and the value of y may not be known *a priori*. (This strategy must be re-considered when there are costs associated with the measurements. Averaging groups of 10 measurements increases the cost by a factor of 10 but decreases the noise only by $\sqrt{10}$, so that the cost/benefit ratio of averaging rapid measurements is very unfavorable. The cost/benefit ratio is even less favorable when the noise has flicker characteristics.)

Although some improvement would still be achieved by averaging repeated measurements through the other pair of channels, this strategy would not work as well as we might wish, since the TDEV of that pair has an intermediate slope between white and flicker noise variations. (As we discussed above, increasing the averaging time in the flicker noise domain does not produce a corresponding improvement in the estimate of the measurements.) In any case, this is not a widely used strategy for the type of time scale we are discussing, since the contribution of the measurement noise is already smaller than the clock noise for the range of averaging times that are commonly used for time scales. However, it is very important in many network-based algorithms, since the measurement noise is much larger than the clock noise in those configurations.

If we know the frequency offset of the device under test, the averaging strategy described above can be extended by including the deterministic frequency offset into the ensemble of measured time differences. That is, before we include them in the average time difference, we adjust the later time differences by the time dispersion resulting from the known frequency offset. The limitation on the averaging time is now set by the noise in the frequency offset rather than the frequency offset itself. The improvement realized by this strategy is often not worth the effort, since the standard deviation of the mean of time differences improves only as the square root of the number of contributors to the average. In practice, any averaging scheme must also deal with outliers and error detection, and this adds considerable complexity when the deterministic frequency offset must be included in the estimation process.

In order to estimate the frequency offset of the device, we must choose a time interval between measurements so that the contribution of y to the measured time difference is larger than the measurement noise. That is,

$$y\Delta t \gg \xi. \tag{7}$$

If we use the typical values for this configuration, the minimum measurement interval is given by $\Delta t \gg 10^{-12}/2 \times 10^{-13} = 5$ s. The NIST ensemble uses a measurement interval of 720 s, so that the deterministic frequency offset would result in a time dispersion of about $720 \times 2 \times 10^{-13} = 144$ ps. For averaging times less than about 1 day, the frequency fluctuations of a cesium device are about $2 \times 10^{-12}/\tau^{1/2}$, so that the time dispersion at 720 s due to these frequency fluctuations is approximately $2 \times 10^{-12} \times \tau^{1/2} = 54$ ps. The 1 ps measurement noise is not important in this configuration. The contribution of the frequency dispersion is smaller than that of the deterministic frequency, but not negligible. The dominant contribution to the variance is Gaussian frequency fluctuations, which will be an important assumption when we discuss ensemble algorithms.

If the time differences were measured by the use of a simple time interval counter, the measurement noise would be much larger – of order 0.1–0.5 ns, so that the time interval between measurements would have to be larger by about a factor of 100–500 to satisfy Eq. (7). Time scales that use this type of hardware often use a measurement interval of 1 h to satisfy this requirement.

If we use the parameters above, the deterministic frequency offset contributes approximately $2 \times 10^{-13} \times 3600 = 720$ ps at an averaging time of 1 h, and the frequency dispersion contributes approximately 120 ps to the measured time difference. The ratio of the deterministic frequency contribution to the stochastic frequency contribution has increased by $\sqrt{(3600/720)} = 2.2$ relative to the previous averaging time, but this averaging time was driven more by the need to satisfy Eq. (7) than to increase the significance of the contribution of the frequency variance.

These calculations illustrate the trade-offs in choosing a measurement interval. The estimation process is easy in this simple case, since the contributions of both the deterministic frequency and the frequency dispersion become larger and easier to measure as the averaging time is increased, and there is no down-side to simply making the measurement interval longer and longer. This situation will change when frequency aging and non-Gaussian frequency variations must be included.

Estimating the frequency difference between two devices situated in different laboratories is a much more difficult problem because the measurement noise is significantly larger and the accuracy of the standards is better. It is difficult to make the measurement noise at 1 s much less than 0.1 ns, so that comparing primary frequency standards that differ in fractional frequency by 10^{-15} requires about 1 day of averaging under the best of circumstances. The comparisons will become more difficult and require longer averaging times as more accurate primary frequency devices are developed. The increase in the averaging time needed to satisfy Eq. (7) also will stress the requirement that the measurement noise be well characterized as a Gaussian random process, since this assumption is already not completely accurate even at an averaging time of 1 day.

At the other end of the spectrum, if the channel noise is large enough (for example, a channel that uses a packet-switched network such as the Internet to compare two clocks) then the channel noise will be so much larger than the contribution of the frequency fluctuations of an atomic clock that the channel noise may dominate the noise budget at all reasonable averaging times, and Eq. (7) can never be satisfied. The time transfer noise of the channel translates into a frequency

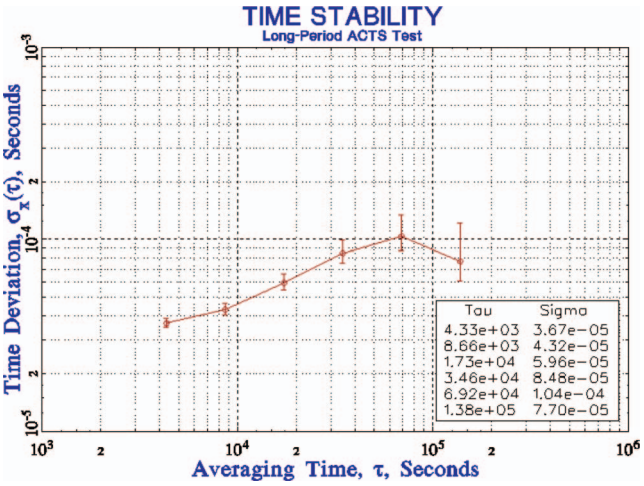

FIG. 3. (Color online) The time stability (TDEV) of a voice-grade telephone line as a function of averaging time. Note the poorer stability at periods near 1 day.

noise of order $\xi/\Delta t_{max}$, where Δt_{max} is the maximum averaging time that can be used. In many situations, this maximum averaging time is set by the onset of non-Gaussian processes. From the point of view of a user there may be no advantage to having an atomic clock as the remote reference if its frequency, y, and frequency stability, η, are much better than the effective frequency noise of the channel, $\xi/\Delta t_{max}$, and a much cheaper remote clock would work just as well.

For example, Fig. 3 shows the time stability (TDEV) of a dial-up telephone line, which is typically more stable than a packet-switched network connection. The value of TDEV is not less than 30 μs for all averaging times and is about 100 μs for averaging time near 1 day. The time fluctuations of the channel contribute about 10^{-7} to the effective frequency noise at short periods and about 10^{-9} for an averaging time of about 1 day. These values are much larger than the frequency fluctuations of any atomic clock, so that, from the point of view of a user who is using this channel to compare the local clock with the remote one, the synchronization process would not benefit from the fact that an atomic clock is being used as the remote reference. The same comment would apply to a network time server that uses data from a global positioning system (GPS) satellite. Even a simple GPS receiver delivers timing information with an uncertainty that is significantly less than 1 μs. This is not as accurate as a directly connected atomic clock, but is much better than the TDEV of the network connection used by a user.

VII. ESTIMATING THE CLOCK FREQUENCY

The estimate of the average frequency offset over the interval between the time-difference measurements would be given by the first difference of the measurements

$$y(t) = \frac{x(t) - x(t - \Delta t)}{\Delta t}. \tag{8}$$

The averaging time, Δt, is chosen to be long enough so that the measurement noise make only a small contribution to the estimate, while at the same time being short enough so that the contribution of frequency aging to the time difference can be ignored. In that case, the variance in these data is determined primarily by η – in other words, we are in the domain where the variance in the time differences is characterized by white frequency noise, and we would now average consecutive *frequency* estimates to reduce the impact of η. The time differences are *not* a Gaussian random variable in this domain and averaging *them* is no longer the optimum strategy. Most atomic-clock time scales operate in this regime.

The previous discussion assumed that the frequency fluctuations of the clock were well characterized as a Gaussian random process so that the uncertainty in the frequency offset of the clock could be reduced by averaging consecutive first-differences of the measured time differences. The assumption of Eq. (8) is that the frequency aging contribution to the time differences is small compared to the other contributions. This was easily realized for a cesium frequency device, since our simple model of those devices used a frequency aging term of zero.

The situation becomes more complicated when a hydrogen maser is considered, since its frequency noise is small and it often has significant frequency aging. For example, suppose $y = 2 \times 10^{-13}$, $d = 10^{-21}$ s^{-1}, and $\eta = 2 \times 10^{-15}$. (Rubidium atomic clocks also have significant frequency aging. The frequency aging is about 5×10^{-11}/month or 2×10^{-17} s^{-1} and the frequency noise is about 2×10^{-12}. The frequency noise is too large for the time scale of a national laboratory, and the frequency aging also presents a problem because it is not stationary.) If we use an averaging time of 720 s, which is used in the NIST ensemble, the contribution of the frequency aging of a hydrogen maser to the time difference is about $0.5 \times 10^{-21} \times 720^2 = 2.6 \times 10^{-16}$ s, which is negligible compared to any other term in the expression. Thus, the use of Eq. (8) is justified. The contribution of the frequency aging to the measured time differences does not exceed even the measurement noise until an averaging time of about 44 000 s or about one-half of a day.

However, even when the contribution of the frequency aging makes only a small contribution to the estimate of the time difference, the assumption that the frequency variance can be well characterized as a Gaussian variable is inadequate for sufficiently long averaging times. From Eq. (2), the contribution of the frequency aging will be comparable to the Gaussian frequency noise for averaging times such that

$$d(t)\Delta t \approx \eta. \tag{9}$$

If we use the values in the previous paragraph, $\Delta t = 2 \times 10^6$ s, which is about 3 weeks, and longer averaging times would be needed in practice to increase the confidence in the determination of the aging parameter.

The long averaging times that are needed to get a robust estimate of the frequency aging of a hydrogen maser introduce a number of practical difficulties. The first problem is that because a robust estimate of the frequency aging requires a significant averaging time, there is a period of time following a cold start when the model is not correct, and predictions of the measured time differences based on the model will have undesirable start-up transients.

The second problem is that the frequency aging is rarely a simple constant – it has stochastic variation as well. Some

of this variation is a Gaussian random process and can be adequately modeled by the noise parameter ζ in Eq. (3) or, equivalently, by extending the definition of the frequency noise parameter, η, to include a non-Gaussian contribution, which may be more intuitively appealing but which is more difficult to handle statistically. However it is modeled, the frequency aging is usually more complicated than the model defined by Eqs. (2) and (3), especially at longer averaging times.

The details depend on the device, but the conclusion that the model equations for frequency standards are inadequate at sufficiently long averaging times is device independent. We would expect that ensembles of such devices would have the same limitation – any algorithm that "averages" the frequencies of real devices will have stochastic frequency fluctuations at sufficiently long averaging times that are not easily modeled. Adding additional terms to the Taylor series of the time differences as a function of the averaging time (Eq. (1)) generally does not help. These terms depend on higher and higher powers of Δt, so that the coefficients of these terms can be estimated only by the use of longer and longer averaging times, and these estimates are corrupted by the non-Gaussian variation in the lower-order terms at these longer periods.

Therefore, there is a symbiotic relationship between time scales and primary frequency standards. The former are needed as flywheels to provide real-time time signals and to provide a stable short-term frequency reference to assist in the evaluation of the systematic offsets of a primary frequency standard, while the latter are needed to provide a frequency reference at longer averaging times when the stochastic models of the time scale become inadequate. From the statistical point of view, the cross-over between a clock ensemble and a primary frequency standard is reached when the frequency uncertainty of the primary frequency standard is comparable to the frequency aging of the clock in the ensemble. The accuracy of the NIST primary frequency standard is $\leq 10^{-15}$, so that the cross-over is on the order of 1 month.

The offset between a primary frequency standard and the frequency of an ensemble of clocks is used to steer the frequency of the ensemble. We can either incorporate the offset by adding it as an administrative change to the parameters of the ensemble or we can use the offset to control a physical clock signal that is outside of the ensemble algorithm. I present a few general considerations here and will discuss the question of steering in more detail in a later section.

The International Bureau of Weights and Measures (Bureau International des Poids et Mesures (BIPM) in French) uses the former method, since its time scales have no realization as physical outputs – the free-running frequency of the international atomic clock ensemble (EAL – Échelle Atomique Libre), which is composed of commercial cesium clocks and hydrogen masers, is steered based on the input from primary frequency standards to produce the International Atomic Time scale, TAI. The steering parameters are adjusted periodically, and are published in the BIPM Circular T.[14] The current (May 2011) frequency steering value (EAL-TAI) = 6.563×10^{-13} and the aging of the steering correction is about -0.007×10^{-13}/month during the first months of 2011 (see *Note Added in Proof* at the end of this paper).

The steering applied to EAL to produce TAI is a result of the frequency aging of EAL. To put this issue in perspective, consider that the frequency of EAL aged by about 3.1×10^{-15} during calendar year 2008.[15] (Based on the data from the first months of 2011, the aging rate has increased significantly since 2008.) Consider that the calculation of EAL was based on data from about 400 clocks and that about 100 of them were hydrogen masers, so that the masers might contribute a weight of order 25% to EAL. If each of the masers had a frequency aging comparable to the maser at NIST (10^{-21} s^{-1}) and if the aging parameters were uniformly distributed about 0, then the standard deviation of the aging of the ensemble might be of order 10^{-22} s^{-1} or 3.2×10^{-15} yr^{-1}. If the weight of the masers was 25% of EAL, the masers would have been responsible for a frequency aging of EAL of about 8×10^{-16} yr^{-1}. This result is not very different from the observed frequency aging during 2008. Although this calculation has many approximations and assumptions, it illustrates the effect of frequency aging in masers and the need to model it accurately for time scales that are designed for maximum long-term stability.

The steering method used at NIST uses two methods. The frequency aging of the NIST masers is estimated relative to the data from the NIST primary frequency standard, and these estimates are inserted into the time scale, where they are treated as constant values (see following discussion). The estimate of the time and frequency offsets between the output of the NIST atomic clock ensemble and TAI is implemented as a steering correction applied to offset the time and frequency of a physical clock. The steered output is used as the time reference for all of the NIST services. The parameters of the ensemble are not modified. The provisional value for the steering correction is generally published 2 months in advance, and the final value is published about 30 days in advance in the NIST time and frequency bulletin, which is available on the web.[16] I will discuss steering algorithms in more detail below.

VIII. CLOCK ENSEMBLES

The previous discussion outlined the general principles of atomic clocks and how ensembles of them are likely to behave. We now turn to a discussion of how a clock ensemble is computed.

The models discussed in Secs. VI and VII assumed that the clock being characterized was being compared to an ideal, perfect reference. In practice, the clock is compared against an ensemble of similar devices with the assumptions that (a) the performance of the ensemble is better than any one of its members and (b) the noise of the ensemble time used as the reference is not correlated with the noise of the device under test. The first requirement is not too difficult to satisfy for a reasonably large ensemble, but the second one is more troublesome, since the device being characterized is typically also a member of the ensemble used as the reference for the time-difference measurements. The most serious version of this problem is when all of the clocks in the ensemble have a common-mode offset, which cannot be detected since it cancels in all of the measured time-differences. Even without a

common-mode offset, the clock being characterized always looks too good in this situation, since it contributes data to the ensemble average that is being used to characterize it. This problem is especially troublesome when the ensemble has only a few member clocks. We will return to this point later.

There are many different ensemble algorithms, but all of them start with the measurements of the time differences between each of the member clocks and one of them that is designated as the reference device for the hardware. In general, the reference device is chosen for its reliability and longevity rather than for any special statistical properties.

The frequency and frequency aging parameters of each clock are estimated with respect to the ensemble. Since the input data are time differences, all of these parameters have an arbitrary additive constant that cannot be determined from "inside" the estimation process. In practice, these additive constants are determined from external calibration data; they can be administratively adjusted to steer the ensemble without affecting its internal dynamics. This is the method used to realize TAI from EAL as described in Sec. VII.

The AT1 algorithm used at NIST is typical of the algorithms that are used by many national laboratories, and we will describe it in detail. A variation of the algorithm called ALGOS is used by the BIPM to compute EAL and TAI.[17] We will also discuss algorithms based on the Kalman filter paradigm. Taken together, these two designs are the basis for almost all of the time scale algorithms used at present.

In the following discussion, the measured time difference between the reference clock, r, and clock, j, at epoch t_k will be designated as $X_{rj}(k)$. The sign convention is that a positive value means that the time of the reference clock is ahead of the time of clock j. In other words, a positive value implies that the tick of the reference clock at some epoch occurs before the tick of clock j for the same epoch. There is nothing special about the reference clock, and its time difference is reported by the hardware as $X_{rr}(k) = 0$. We assume that the epochs are specified with negligible uncertainty. This is not a difficult requirement to realize, since the frequency offsets for atomic clocks are very small, so that small errors in the epoch produce only a very small contribution to the measured time differences. For example, if we take the measurement noise to be on the order of 1 ps and the maximum offset frequency of any clock to be $\leq 10^{-11}$, then specifying the epoch to within 0.1 s is adequate to have the contribution of the error in the epoch to be of the same order as the measurement noise. The actual accuracy in the determination of τ_k is much better than this limit.

The measurements are normally equally spaced in epoch at an interval of τ seconds, so that the measurement epochs can be expressed recursively by $t_k = t_{k-1} + \tau$. (The occurrence of a leap second complicates matters, since the physical elapsed time between measurements that span the leap second differs from the value computed from the time tags by ± 1 s. This discrepancy is a one-time effect and is handled by special code in AT1. This problem becomes more serious as the spacing between measurements is decreased.) The current implementation at NIST uses a value of $\tau = 720$ s (12 min), which easily satisfies the requirement of Eq. (7), so that we can ignore the noise in the measurement system and Eq. (9), so that the frequency aging can be treated as a constant during the calculation. The BIPM uses an interval of 5 days between measurements, partially because the noise in the time difference measurements between laboratories is much larger than the noise of the dual mixer system used at NIST and partially for historical reasons. The exact value used at NIST is not critical and the value that is used is chosen mostly for computational convenience, since it is an exact decimal fraction of an hour.

The measurement interval used by the BIPM was shortened from ten days to five days in 1996. Although the statistics of the measurement processes could justify a shorter measurement interval, this is unlikely to be realized for administrative reasons. Based on the previous discussion, we expect that the frequency aging is constant (or 0) over the measurement interval and that the variance in the measurement data can be modeled as white frequency noise. The BIPM algorithm, which does not explicitly include frequency aging in the model, is marginal in this respect, as I have discussed in detail in Sec. VII. The frequency aging of a hydrogen maser is typically about 10^{-21} s^{-1}. For an averaging time of five days, the frequency aging contributes about 93 ps to the measured time difference, and the frequency of the maser ages by about 2.6×10^{-15} over the one-month interval used to report the international atomic time scale. These quantities are small, but they are not negligible, and they become more important as longer averaging times are used. For example, the ALGOS algorithm can estimate the frequency of a contributing clock from the most recent six months of data. For most cesium standards, the frequency fluctuations for this averaging time correspond to flicker or random-walk statistics, so that the optimum estimate for the current interval is the value at the end of the previous one. However, the frequency aging of a hydrogen maser is even more important over this averaging time, and treating the frequency of a hydrogen maser as constant over this interval is not optimum.

The BIPM has done a number of studies and simulations that demonstrate the impact of the aging of the hydrogen masers that contribute to the computation of EAL.[18] Including these terms in the time scale algorithm should improve the accuracy of EAL and reduce its frequency aging. This will also reduce the need for applying steering corrections to TAI and should permit TAI to realize the SI second with a much smaller frequency offset term than is needed at present.

I will consider equally spaced measurements to simplify the notation. However, the algorithm does not depend on equally spaced data, and τ in the following equations can be replaced by the actual interval between the current measurement and the previous one. Even when the procedure is designed for equally spaced measurements, different values of τ happen occasionally when the hardware fails and some number of measurement cycles are lost. The measurement cycle is re-synchronized when the hardware is restarted, so that the gap is an exact integer multiple of τ. The larger interval across the gap is handled with no special processing.

The time of each clock with respect to the ensemble at epoch t_k is modeled recursively in terms of its time offset, frequency offset, and frequency aging at the previous epoch

t_{k-1} by

$$\hat{x}_{je}(k) = x_{je}(k-1) + y_{je}(k-1)\tau + 0.5d_{je}(k-1)\tau^2. \tag{10}$$

This equation is analogous to Eq. (1), except that the parameters on the right-hand side are the time offset, the frequency offset and the frequency aging, respectively, of clock j with respect to the *ensemble* rather than with respect to a physical device, and the result is an *estimate* of the time difference of clock j with respect to the ensemble at the current epoch with the estimates at the previous epoch used as a prediction. The ensemble time need not be realized in a physical device; however, we could reverse the interpretation of Eq. (10) and suggest that it is an estimate of the time of the ensemble relative to physical clock j. There are N equations of this type – one for each member of the ensemble, including the hardware reference clock. However, there are only N-1 time difference measurements, so that these equations do not provide a unique definition of the ensemble time.

Each one of the measured time differences can be combined with the corresponding model equation for that clock to compute a prediction of the time of the reference clock with respect to the ensemble at the current epoch. Thus,

$$\widehat{X}^j_{re}(k) = \hat{x}_{je}(k) + X_{rj}(k) \tag{11}$$

is a prediction of the time of the reference clock with respect to the ensemble at the current epoch based on the model for clock j and the measurement of the time difference between the reference clock and clock j. There is one of these equations for each clock in the ensemble, including the reference clock, where it is simply an identity, since the corresponding time difference $X_{rr}(k) = 0$.

The first term on the right-hand side of Eq. (11) is the estimate of the current time difference of the clock with respect to the ensemble with the deterministic characteristics of each of the member clocks determined from previous computations. If the models are an accurate representation of the performance of the physical clocks, the N estimates in Eq. (11) differ only by the Gaussian frequency noise of each clock and, to a much lesser extent by the Gaussian noise in the measurement process. (The AT1 algorithm assumes that the frequency aging, which is included in the model for each clock, is either a constant or zero over the measurement interval of 720 s.) With this assumption, the provisional, statistically robust estimate of the time of the reference clock with respect to the ensemble is the weighted sum of these estimates over all of the members of the ensemble,

$$\widehat{X}_{re}(k) = \sum_{j=1}^{N} w_j(k)\widehat{X}^j_{re}(k). \tag{12}$$

The reference clock contributes to the sum on the right-hand side of Eq. (12) just like any other clock. As above, we could just as easily consider this equation as providing a preliminary estimate of the ensemble time as an offset from the physical reference clock. The weight of each estimate is computed from the average prediction error of the clock over the previous cycles (defined below). On the assumption that the estimates are normally distributed, the optimum weight for each clock is the inverse of its variance

$$w_j(k) \sim \frac{1}{\sigma_j^2(k)}. \tag{13}$$

The weights are normalized so that they sum to 1 by the use of the normalization constant $\sigma^2(k)$:

$$\sigma^2(k)\sum_{j=1}^{N} w_j(k) = 1,$$

so that

$$\frac{1}{\sigma^2(k)} = \sum_{j=1}^{N} w_j(k) = \sum_{j=1}^{N} \frac{1}{\sigma_j^2(k)}, \tag{14}$$

$$w_j(k) = \frac{\sigma^2(k)}{\sigma_j^2(k)}.$$

The quantity $\sigma(k)$ is the standard deviation of the ensemble at time t_k. In this model, adding even a relatively poor, low-weight clock improves the standard deviation of the ensemble computed by Eq. (14). As we will discuss below, the weight used in Eq. (12) for any clock may be limited by administrative considerations to a value less than this.

The prediction error for each clock is the difference between the estimate computed in Eq. (11) for that clock and the ensemble average of all of these estimates as computed by Eq. (12):

$$\varepsilon_j(k) = \widehat{X}^j_{re}(k) - \widehat{X}_{re}(k). \tag{15}$$

The prediction error on this measurement cycle is compared to the average prediction error over previous cycles, by the use of the following statistic:

$$\kappa_j(k) = \frac{|\varepsilon_j(k)|}{\sigma_j(k)}. \tag{16}$$

Case 1: $\kappa_j(k) \le 3$. That is, the prediction error on this measurement cycle is within 3 standard deviations of the average prediction error over previous cycles. Accept the estimate of the ensemble time from this clock and continue.

Case 2: $3 < \kappa_j(k) < 4$. The prediction error is significantly larger than the average value over the previous cycles. Decrease the weight of this clock in the ensemble average (Eq. (12)). In the following expression, the original weight computed in Eq. (14) is $w_j^0(k)$, and it is replaced in the recomputation of the ensemble average by

$$w_j(k) = (4 - \kappa_j(k))w_j^0(k), \tag{17}$$

which de-weights the clock linearly from its value derived from the prediction error for $\kappa_j(k) = 3$ to zero when $\kappa_j(k) = 4$. Set a flag to show that the clock has been de-weighted at this epoch. Return to Eq. (12) to re-compute the ensemble average time with this new weight.

Case 3: $\kappa_j(k) \ge 4$. Set the weight of this clock to zero, return to Eq. (12) and re-compute the average time of the reference clock with respect to the ensemble. Set a flag to show that this clock has been dropped from the ensemble average at this epoch.

The decision to accept a clock if its prediction error is not greater than three sigma and reject it completely if its prediction error is greater than or equal to four sigma is somewhat arbitrary. It is derived from operational experience and is based on the usual compromise between accepting a clock that should be rejected and rejecting a clock that should be accepted. The intermediate case is designed to gradually deweight a clock whose prediction error is close to the edge of acceptability rather than to drop it suddenly when it crosses the threshold of acceptability. This prevents small fluctuations in the performance of a clock that is near the reset threshold from causing significantly larger transients in the ensemble average.

If more than one clock satisfies the conditions of case 2 or case 3, then the ensemble average may not be correct. Assume that only one clock is in error, perform the operation only on the clock with the largest value of κ and re-compute the ensemble average by means of Eq. (12) with the modified weight for that clock. A clock whose weight has been modified by case 2 or case 3 is not tested again on a subsequent loop. When no further failures are detected, continue with the subsequent section on parameter updates.

IX. ADMINISTRATIVE LIMIT ON THE WEIGHTS

The algorithm described above is potentially unstable if one of the clocks is significantly more stable than the others so that its prediction error is consistently smaller than the errors of the other members of the ensemble. The same effect can happen if the combination of the measurement noise and the prediction errors conspire to decrease the prediction error of one of the clocks. Since the weight of a clock in the ensemble average is derived from its prediction error, a clock with a small prediction error has a high weight, and this produces a significant correlation between the time of the clock and the average time of the ensemble. The prediction error for such a clock is always too small, since it contributes to both terms on the right-hand side of Eq. (15). In an extreme situation, this can result in a positive feedback loop, in which a clock that is initially somewhat better than the others eventually is given a weight close to 100% and effectively takes over the ensemble. This is especially serious in a small ensemble.

Since the positive feedback loop results from the correlation between the ensemble average of the prediction errors and the contribution of a high-weight clock, one solution is to compute the effects of this correlation and increase the prediction error (and thereby decrease the weight of the clock in the average) to account for it.[19] This method will be discussed below in the parameter update section. A second solution is to limit the maximum weight that any clock can have in the ensemble average. This maximum weight for the NIST ensemble is set at 30%. From Eq. (14), if the maximum weight is to be limited to 0.3, then

$$w_j(k) = \frac{\sigma^2(k)}{\sigma_j^2(k)} \le 0.3, \tag{18}$$

so that

$$\sigma_j(k) \ge 1.83\sigma(k).$$

If the σ for a clock is smaller than this limit, the weight of the clock in Eq. (12) is set to 0.3, which effectively sets σ for that clock to be the value calculated in Eq. (18). The normalization constant is then re-computed with this reduced weight.

The ALGOS algorithm has a similar administrative limit. The weight that any clock can have in that algorithm is currently limited to 2.5/N, where N is the number of clocks in the ensemble calculation.[20] In other words, the ALGOS algorithm allows the best clocks in the ensemble to have a weight that is 2.5× the value that would be used (1/N) if all of the clocks were equal. The maximum weight of a clock in ALGOS is currently somewhat less than 1%.

Since the weights are normalized so that the sum is 1, reducing the weight of a good clock below what its statistical performance would predict implicitly transfers the weight to poorer clocks and gives them more weight than they deserve. The statistical performance of the scale is then not as good as it could be if the administrative limit were not enforced. This degradation in performance is considered an acceptable price to pay for avoiding the possibility of having one clock take over the scale.

The administrative limit must always be larger than 1/N, so that the NIST administrative limit implies that the NIST ensemble must always have more than 3 clocks. Ensembles of less than 3 clocks are possible in principle, but they are not used in practice because there is no method of assigning the prediction error in case there is a problem. Even an ensemble of 3 clocks can be marginal in this respect.

X. PARAMETER UPDATES

When the algorithm described above is finished, we have an estimate of the time of the reference clock with respect to the ensemble based on the measurements of all of the other clocks whose prediction errors were not too large. We can also consider this datum as the final, *unique*, realization of the ensemble paper time as an offset from a physical clock. Since we have measured the physical time differences between the reference clock and all of the other clocks in the ensemble, we can also realize the ensemble time by combining these physical time differences with the calculated offset of the reference clock from the ensemble. Thus, the ensemble time can be realized as a time offset from any of the physical clocks that were used to compute it, and the ensemble frequency is the evolution of this time difference on consecutive measurement cycles. In general, none of the physical clocks directly realizes either the ensemble time or the ensemble frequency. Equation (12) (combined with the definition of the weights in Eq. (14), including the administrative weighting limits) is very important, since it defines the paper time of the ensemble with respect to the time of a physical clock. It is a necessary adjunct to the N-1 measured time differences. The definition of the ensemble time is not uniquely determined without this additional constraint.

The next step is to evaluate the adequacy of the model that was used to estimate the time difference of each of the physical clocks with respect to the ensemble. The previous results already confirmed that the prediction errors of all of the clocks that contributed to the ensemble on this cycle were

reasonably consistent with the previous estimates of their time, frequency, and frequency aging, where "reasonably consistent" implies the evaluations based on Eq. (16) discussed above. We now divide the residual prediction error for each clock into two components: a deterministic value that is used to adjust the model parameters, and a stochastic value that is removed by averaging.

The time of the reference clock with respect to the ensemble is set to the estimate computed above in Eq. (12) where the sum uses the weight for each clock as computed by Eq. (14) and as modified by the subsequent tests and administrative limits,

$$x_{re}(k) = \widehat{X}_{re}(k). \tag{19}$$

This expression can be used to compute the final value for the prediction error of each clock as the difference between the time of the reference clock with respect to the ensemble predicted by clock j and the ensemble average of these predictions from Eq. (19):

$$\varepsilon_j(k) = \widehat{X}^j_{re}(k) - x_{re}(k). \tag{20}$$

Equation (20) is evaluated for every clock in the ensemble, including the reference clock, which is not treated in any special way. These updated estimates will be identical to the provisional values if the prediction errors of all of the clocks were within the acceptable limit of three times the value for the corresponding σ so that no clock was de-weighted because it was close to the maximum threshold or reset because its prediction error exceeded the threshold.

The first step in the parameter update process is to deal with any clock whose weight was reduced to zero because its prediction error was too large. We model these clocks as having had a simple time step since the last computation. We adjust the time of the clock with respect to the ensemble so that it matches its value based on its current measured time difference. If clock m was reset on this cycle, then

$$x_{me}(k) = x_{re}(k) - X_{rm}(k). \tag{21}$$

We do not modify its other parameters on this cycle. If this assumption is accurate, then the clock will return to the ensemble with its parameters unchanged on the next measurement cycle and its prediction error will return to be within the expected range.

If the error is not due to a simple time step, then the previous action is unlikely to fix the problem. For example, if the frequency or the stability of the clock changed since the last measurement cycle, then its behavior probably will not be modeled by a single time step. It will most likely be reset repeatedly on subsequent measurement cycles, which effectively removes it from the ensemble, since its weight is set to zero repeatedly. This is generally an indication of a hardware failure, and the ensemble simply calls for human assistance but continues to run with the failed clock effectively removed from the calculation. This strategy is a necessary consequence of the fact that AT1 is a real-time ensemble; more sophisticated responses are possible for an ensemble that does not run in real time. For example, the parameters of the clock could be administratively adjusted based on other data and the scale can be re-computed.

The next step in the process is to update the parameters of each clock that was not reset on the current measurement cycle.

(1) The time of each clock with respect to the ensemble is set to the computed time of the reference clock with respect to the ensemble and the measured time difference between each clock and the reference clock

$$x_{je}(k) = x_{re}(k) - X_{rj}(k). \tag{22}$$

This updated time is used in the following calculation. Equations (21) and (22) are identical – the time of a clock with respect to the ensemble is a direct consequence of the definition of the ensemble and the measured time differences, and this is true whether or not the clock behaved as we predicted it would based on previous data. This procedure implicitly assumes that a measurement of the *physical* time difference between two clocks is an accurate measure of the difference in the time *state* between them. In other words, that the measurement noise is negligible. The Kalman time-scale algorithm, which we will discuss below, relaxes this requirement.

(2) The frequency of the clock with respect to the ensemble is set in two steps. The first step estimates the frequency over the time interval since the last measurement cycle by means of a simple first-difference of the times of the clock with respect to the ensemble

$$f_{je}(k) = \frac{x_{je}(k) - x_{je}(k-1)}{\tau}. \tag{23}$$

Based on the time interval between measurements and the model of the clock, the frequency estimated by Eq. (23) is assumed to have a variance that is due only to white frequency noise, because the measurement interval has been chosen to be long enough so that the measurement noise is much smaller than the frequency noise of the clock and short enough so that the contribution of the frequency aging term to the time differences can be considered as a constant that cancels in the difference. Therefore, the average of these calculations is an unbiased estimate of the frequency of the clock with respect to the ensemble. On the other hand, the number of frequency estimates included in the average must be limited because we will not satisfy the requirement that the frequency aging is a constant or that the frequency variance can be characterized as white frequency noise if we go back too far in time. Since the algorithm is implemented recursively, we use a recursive implementation of the finite average, which "forgets" older frequency estimates with a dimensionless time constant w_y. We then add the contribution of the frequency aging to obtain

$$y_{je}(k) = \frac{w_y y_{je}(k-1) + f_{je}(k)}{1 + w_y} + d_{je}(k-1)\tau,$$

$$y_{je}(k) = y_{je}(k-1) + \frac{1}{1+w_y}(f_{je}(k) - y_{je}(k-1)) + d_{je}(k-1)\tau. \tag{24}$$

The second form of Eq. (24) is algebraically equivalent to the first form, but casts the frequency update in a form that will be easier to compare to the Kalman formalism to be discussed below. In the second form of Eq. (24), the first term is the previous estimate of the frequency and the second term is the weighted difference between this estimate and the value computed on the current cycle. The weighting factor is defined as follows.

The time constant used in the exponential filter above is determined from the statistics of the clock performance as the time interval over which the frequency noise can be considered as having a white spectrum. If T_j is the time constant for clock j in seconds determined in this way, then

$$w_y = \frac{T_j}{\tau}. \tag{25}$$

The time constant is about 4 days ($w_y = 480$ measurement cycles) for a standard-performance cesium standard and about 10 days ($w_y = 1200$ measurement cycles) for a high-performance device when the standards are operated in an uncontrolled temperature environment. The performance is significantly better if the environmental parameters (mostly ambient temperature) are controlled, and the time constants can be increased in this case. The effect of Eqs. (24) and (25) is to partition the measured variance in the frequency of the clock with respect to the ensemble: frequency fluctuations with periods less than T_j are attenuated, while those greater than this time modify the frequency of the clock used to predict the time difference on future measurement cycles. This strategy is not optimum if the frequency fluctuations over the averaging time τ are characterized by a random walk, which is the reason that the EAL algorithm uses a different frequency estimator in which the frequency for any interval is the last value at the end of the previous one.

The update of the prediction error is also computed in two steps. The first step computes the integrated prediction error for clock j over the last 24 h ending at the current time t_k:

$$S_j(k) = \sum_{t_m = t_k - 24\text{h}}^{t_k} \varepsilon_j(t_m), \tag{26}$$

where the prediction error for each cycle given by Eq. (20). This value is corrected for the correlation effect discussed above[21] and the result is passed through an exponential filter with a fixed time constant of 31 days,

$$w_s = \frac{\tau}{86400}\,\frac{1}{1 - w_j(k)}, \tag{27}$$

$$\sigma_j^2(k) = \frac{31\sigma_j^2(k-1) + w_s S_j^2(k)}{31 + w_s}. \tag{28}$$

The first expression in Eq. (27) scales the measurement interval measured in seconds to a fraction of a day. The second term increases the effective prediction error for a clock that has a large weight in the ensemble average. As we discussed above, the prediction error is always too small for such a clock, since it makes a significant contribution to the ensemble that is also being used to evaluate its time difference. Since the maximum weight of a clock in the ensemble is limited to 0.3, the correlation correction in Eq. (27) increases the observed variance by a factor of up to 1.43. Equation (28) is similar in form to the first term in the first form of Eq. (24), so that it could be manipulated to obtain an equation analogous to the first term in the second form of that equation. As in Eq. (24), this form emphasizes the role of the "innovation" – the difference between the current result and the previous running average.

The time constant of 31 days is an administratively chosen constant that is based more on experience than on a rigorous analysis. The value of this time constant partitions the changes of the prediction error. Long-period changes in the prediction error modify the sigma of the clock, which affect both its weight on each cycle and the threshold for the reset algorithm. Shorter- period changes in the prediction error have less effect on the sigma. The result is that slow changes in the prediction error affect the weight of the clock, while more rapid changes are more likely to result in a reset.

The AT1 algorithm does not estimate or update the frequency aging for the reasons discussed above. The aging parameter is generally 0 for cesium standards and must be determined outside of the algorithm for hydrogen masers or rubidium standards. Although the frequency aging for a hydrogen maser is typically on the order of 10^{-21} s^{-1}, it is very important to include it in the model. The contribution of the aging term to the model of the measured time difference is very small over the measurement interval used at NIST, and setting it to zero does not impact the prediction error or the weight of the clock. However, the frequency estimates will by systematically wrong by a small amount, and the ensemble time will exhibit frequency aging as a result.

XI. SUMMARY OF THE AT1 ALGORITHM

The AT1 algorithm uses the prediction error of each of its member clocks in a number of different ways. Large, short-term prediction errors are used for error detection and are used either to reduce the weight of a clock in the ensemble average or to remove it entirely if the prediction error is large enough.

Prediction errors that do not trigger the reset procedure can modify both the model parameters of a clock through the parameter update procedure (Eqs. (21)–(24)) and the weight of the clock through Eq. (28). The time constant in Eq. (28) is generally longer than the time constant of Eq. (24) for most masers and for cesium standards when the environmental parameters are not well controlled, so that the parameter update procedure tends to modify the model parameters of a clock so as to track the changes in the prediction error faster than the weighting algorithm will deweight the clock. The time constants for cesium standards in Eq. (24) can be significantly longer than the value in Eq. (28). The relationship between these time constants is a consequence of the assumption that intermediate-term prediction errors are to be modeled as frequency changes rather than as changes in the long-term weight of the clock. The frequency changes are further divided into short-term random variations, which are attenuated, and longer-term variations that modify the model. The transition between the two responses is defined by Eqs. (24) and (25). The actual values used in any

configuration must be determined based on a consideration of the underlying noise type as a function of averaging interval.

The AT1 ensemble time is defined so that the weighted sum of the differences between the predicted and measured time differences of the members of the ensemble that are not reset will be zero. This is the appropriate method for combining the estimates from the member clocks, given the assumptions that were used to construct the clock models. The weighted average will attenuate the random contributions to the measured time differences, but it cannot eliminate them completely. (In the simple case where the random contributions are just white noise and where all of the clocks have equal weight, the improvement is just proportional to the square root of the number of clocks in the ensemble. Making a significant improvement to an ensemble is therefore an expensive business.) In addition, it cannot cope with inadequacies in the clock models themselves. The AT1 ensemble time will therefore behave very much like a single clock with parameters derived from the weighted average of its members. We would expect that the AT1 ensemble time would have significant variation at longer periods because its member clocks do too, and this is observed in practice.

The performance of the AT1 algorithm is determined by the assumptions that are used to justify Eq. (12), which computes the ensemble time with respect to the physical reference clock as the weighted sum of the estimates of each of the members. The calculation will do the wrong thing if the distribution of time differences is biased because of a correlation among high-weight clocks or if the distribution has a mean of zero but a bi-modal distribution. In both of these situations, the algorithm may choose to reset the wrong clocks. Even when the assumptions of Eq. (12) are well justified, the interaction between the inevitable measurement noise and the weighting algorithm means that the time scales computed by two measurement systems observing the same physical ensemble of clocks will slowly diverge in time and frequency. Based on our experience at NIST, this divergence can be approximated as a differential frequency aging of the two nominally identical time scales of about 4×10^{-23} s^{-1} (about 10^{-16}/month). The variance of the frequency aging can be approximated by flicker frequency modulation.

The performance of AT1-type algorithms at longer periods could be enhanced in principle if the calculation were done retrospectively rather than in real time. The ALGOS algorithm, which is very similar to AT1 and which is used by the BIPM to compute EAL and TAI is computed retrospectively for this reason. However, the calculation of EAL has not solved the problem of long-term frequency aging, and it depends on data from the primary frequency standards to estimate and remove it. In addition, a retrospective scale is not useful for a timing laboratory, which must provide time and frequency services in real time. Many of the current users of time and frequency information (navigation, telecommunications, ...) depend on real-time data, so that real-time ensembles are likely to become more important in the future.

An important weakness of AT1-type algorithms (especially, the real-time implementations of the procedure) is that the model equations and the ensemble calculation favor clocks with better short-term stability even if those clocks have significant longer-term frequency aging. The Kalman algorithm is intended to address this shortcoming, and we will now discuss a generic form of this method.

XII. THE KALMAN TIME-SCALE ALGORITHM

In this section, we discuss a "generic" Kalman time-scale algorithm. Our discussion is based on Gelb.[22] A number of algorithms of this type are described in the literature.[23,24] They all share the same basic principles, although they vary in detail, especially in estimating the weights assigned to the members of the ensemble and in how they deal with outliers and clock errors.

The Kalman time scale algorithm uses the same model equations that we presented above in Eqs. (1)–(3). The model is usually presented by the use of matrix formalism. We define a 3-component state vector: $\mathbf{S_j}(k) = (x_j(k), y_j(k), \text{and } d_j(k))$, where the components are the time difference, frequency difference, and frequency aging for clock j with respect to the ensemble at epoch t_k as before. The noise parameters also are expressed as a 3-component vector: $\mathbf{N_j}(k) = (\xi_j(k), \eta_j(k), \zeta_j(k))$. The evolution of the state vector as a function of time is determined by the 3×3 transition matrix, $\mathbf{\Phi}$:

$$\overline{\overline{\Phi}} = \begin{bmatrix} 1 & \tau & \frac{\tau^2}{2} \\ 0 & 1 & \tau \\ 0 & 0 & 1 \end{bmatrix} \tag{29}$$

With this notation, Eqs. (1)–(3) for clock j are expressed as

$$\vec{S}_j(k) = \overline{\overline{\Phi}}\vec{S}_j(k-1) + \vec{N}_j(k), \tag{30}$$

where $\tau \equiv \Delta t = t_k - t_{k-1}$. It is convenient to choose a constant time interval between measurements, but this is not a requirement of the algorithm. There are N such equations, one for each clock in the ensemble. We can combine all of these N equations into a single matrix equation by defining a 3N-component vector for the state **S** and a 3N-component vector for the noise parameters **N**. Thus,

$$\vec{S} = [x_1, y_1, d_1, x_2, y_2, d_2, \ldots, x_N, y_N, d_N], \tag{31}$$

$$\vec{N} = [\xi_1, \eta_1, \zeta_1, \xi_2, \eta_2, \zeta_2, \ldots, \xi_N, \eta_N, \zeta_N]. \tag{32}$$

The transition matrix for the entire ensemble is composed of N 3×3 sub-matrices along the diagonal, with zeroes everywhere else. Each sub-matrix is of the form of Eq. (29). The transition matrix is a function only of the time interval between measurements.

We observe the state of the system at time t_k. The Kalman formalism could support observations of any of the components of the state vector, but the most common arrangement is the situation we considered above: a measurement of the time differences between each of the clocks and one of them, which is designated as the reference clock for the hardware. The hardware reference clock was not handled differently from any other clock in AT1, but the situation is somewhat more complicated here, as will be shown below.

Suppose that the reference clock is clock 1 and that there are 3 clocks in the ensemble, so that the state vector has 9

components. There are 2 measured time differences: x_1-x_2 and x_1-x_3, so that the measurement vector has 2 components. The measurement vector **O** is related to the state vector **S** by the vector equation,

$$\vec{O}(t) = \overline{\overline{H}}\vec{S}(t) + \vec{v}(t), \tag{33}$$

where the vector **v** specifies the noise of the measurement process, which is assumed to be a white noise process. Note that this formalism supports two contributions to the white phase noise of the time differences: the phase noise of the clock itself, ξ, and the phase noise of the measurement process, **v**. This is an important difference between this type of algorithm and the AT1 algorithm discussed above. The Kalman algorithm allows for the possibility that we do not observe the state directly – we see it only through Eq. (33) with the additional noise vector v. The AT1 algorithm implicitly assumed that this white phase noise contribution was negligible and that the measurements directly observed the time differences of the state vectors themselves.

The **H** matrix specifies the relationship between the state vector and the observations, and it has a simple form when the observations are the two time differences specified above for the 3-clock ensemble

$$\overline{\overline{H}} = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}. \tag{34}$$

The noise of the measurement process depends on exactly how the measurements are made. In the simplest case, we have two identical time-interval counters that measure the two time differences, x_1–x_2 and x_1–x_3. The variance of the measurement noise is the same in both channels, but the noise in each channel is not correlated with the other one. The vector **v** is simply

$$\vec{v} = \sigma_v \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \tag{35}$$

where σ_v is the magnitude of measurement noise in each channel of the time difference hardware. (Although clock 1 is used to measure the time difference in both channels, Eq. (35) represents the measurement noise of the *channel*, rather than the noise of the clock.) Since the two channels are not correlated, the expectation values of the cross terms are 0. The covariance of the measurement noise in the simple case is given by

$$C_v(t) = E[\vec{v}(t)\vec{v}^T(t)] = \sigma_v^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{36}$$

If the time differences are measured by a dual-mixer configuration, the signal from each clock is mixed with a local oscillator and the time difference between two signals is then measured at the lower intermediate frequency output of the two mixers. Again, we assume that all of the measurement channels are identical, so that the diagonal elements of the covariance matrix are of the form

$$(x_1 - x_2)(x_1 - x_2) = x_1^2 + x_2^2 = 2\sigma_v^2,$$

while the off-diagonal cross terms are of the form

$$(x_1 - x_2)(x_1 - x_3) = x_1^2 = \sigma_v^2,$$

so that the covariance matrix is

$$C_v(t) = \sigma_v^2 \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \tag{37}$$

where we have assumed that the expectation value of the cross terms $x_j x_k$ is zero for all j and k. The covariance in Eq. (37) includes cross-terms because each signal is mixed with the local oscillator before the measurement, which adds noise that is not present in the simpler model that yielded Eq. (36). In both cases, the covariance of the measurement process is a characteristic of the system and has no time dependence.

The covariance of the clock noise vector **N** is defined in the same way and is computed in terms of the white noise parameters ξ, η, and ζ for each clock. The noise parameters of each of the clocks are assumed to be uncorrelated, so that the cross-products of terms from different clocks are all 0. (This assumption is identical to the assumption that justified the computation of the ensemble average time in AT1. Both algorithms will do the wrong thing when this is not correct.) The covariance matrix of the noise vector is computed as above

$$\overline{\overline{Q}}(t) = E[\vec{N}(t)\vec{N}^T(t)]. \tag{38}$$

The Q matrix is initially diagonal; the diagonal elements q_1, q_2, q_3 are the variances of the noise parameters ξ, η, and ζ, respectively. It does not depend on time. However, the covariance matrix evolves during the interval between measurements and its off-diagonal elements become non-zero. This is simply a mathematical representation of the fact that the variance of the frequency of the device contributes to the time difference, etc. If the interval between measurements is τ, the covariance matrix evolves during that interval according to[24]

$$C = \int_0^\tau \phi(t) Q \phi^T(t) dt. \tag{39}$$

Using ϕ from Eq. (29), the covariance matrix is symmetric. The diagonal elements are

$$\begin{aligned} C_{11} &= q_1\tau + q_2\frac{\tau^3}{3} + q_3\frac{\tau^5}{20}, \\ C_{22} &= q_2\tau + q_3\frac{\tau^3}{3}, \\ C_{33} &= q_3\tau, \end{aligned} \tag{40}$$

and the cross-terms are

$$C_{12} = C_{21} = q_2\frac{\tau^2}{2} + q_3\frac{\tau^4}{8},$$

$$C_{13} = C_{31} = q_3\frac{\tau^3}{6},$$

$$C_{23} = C_{32} = q_3\frac{\tau^2}{2}.$$

From the Allan variance perspective, the q_1, q_2, q_3 parameters represent the white frequency noise, random-walk frequency noise, and random-run frequency noise, respectively.

We now consider the evolution of the state of the ensemble, starting at time t_{k-1}. We first estimate the state of the ensemble at time t_k just before the measurements are made. This estimate is denoted by $\hat{S}$(k-). The estimate of the state of the

ensemble immediately after the measurements is denoted by $\hat{S}$(k+). We look for an estimate of the ensemble state immediately after the measurements are made as a linear combination of the state just before the measurements and the measurement data. That is,

$$\hat{\boldsymbol{S}}(k+) = \boldsymbol{K'}_k \hat{S}(k-) + \boldsymbol{K}_k \boldsymbol{O}(k), \tag{41}$$

where K′ and K are weighting matrices (to be defined below) applied to the previous state and the current measurements, respectively, at the epoch t_k.

The estimates of the states before and after the measurement can be divided into two pieces: the true state at time t_k and the prediction error of the state just before and just after the measurement. The error in each estimate is denoted by the tilde,

$$\hat{S}(k+) = S(k) + \tilde{S}(k+), \tag{42}$$

$$\hat{S}(k-) = S(k) + \tilde{S}(k-). \tag{43}$$

Replace O(k) in Eq. (41) with Eq. (33), and then replace the left side of Eq. (42) with Eq. (41). Then replace S(k−) with Eq. (43). Re-arrange the terms to get an estimate of the prediction error just after the measurements have been performed (the matrix **1** is the identity matrix),

$$\tilde{S}(k+) = \left(K'_k + K_k H_k - \bar{\bar{1}}\right) S(k) + K'_k \tilde{S}(k-) + K_k v_k. \tag{44}$$

Find the conditions needed to make the expectation value of this prediction error equal to zero. The expectation value of the measurement process, which is the third term on the right side is 0 by assumption. The expectation value of the second term will be zero if (*and only if*) the prediction of the state just before the measurement was unbiased. If these two conditions are satisfied, then the expectation value of the entire expression will be zero if the expectation value of the first term is zero, which defines a relationship between the two weighting matrices,

$$K'_k = \bar{\bar{1}} - K_k H_k. \tag{45}$$

If we substitute this value into Eq. (41), we obtain

$$\hat{\boldsymbol{S}}(k+) = \hat{S}(k-) + K_k[O(k) - H_k \hat{S}(k-)]. \tag{46}$$

The first term is the value of the state just before the measurement is made. It is computed from the state at time (k−1) and the matrix **Φ** defined above. The second term is the difference between this value and the observations. In other words, the second term is the difference between what we expected to find based on the previous state and what we actually measured. It is often called the innovation for this reason, and the parameter K_k is the Kalman gain matrix – the weight that scales the difference between what we expected to find at time t_k based on the previous state and what we measured. The previous state vector is multiplied by the measurement matrix **H** so that the two terms in the brackets are the same. In the case we are considering, **H** extracts the differences in the time states of the various clocks from the state vector **S**, which is what we have measured in **O**. However, the formalism would support other types of measurements.

We choose K_k to minimize the weighted scalar sum of the diagonal terms of the covariance matrix of the prediction error just after the measurements have been performed. The result for the Kalman gain is in the literature,

$$K_k = P_K(-) H_k^T \left[H_k P_k(-) H_k^T + C_v\right]^{-1}, \tag{47}$$

where C_v is the covariance matrix of the measurement noise **v** (Eq. (36) or Eq. (37)), and P_k(-) is the covariance matrix of $\hat{S}(k-)$. It is the sum of two terms: the sum of the evolution of the covariance matrix from the previous iteration, C(n−1) and the contribution from the current calculation, Q. Thus,

$$P_K(-) = \phi C(n-1) \phi^T + Q. \tag{48}$$

Using the Kalman gain and the previous covariance matrix, the updated covariance matrix is given by

$$C(n) = [1 - K_k H]\, P_K(-). \tag{49}$$

This result is mathematically complex and physically obscure.

XIII. A SIMPLE EXAMPLE OF THE KALMAN ALGORITHM

I will demonstrate the operation of the Kalman algorithm using a simple example. Suppose that we wish to compare two systems each of which can be characterized as having only a single parameter in its state. For example, a measurement of the time difference between two clocks whose frequencies are identical and have no deterministic or stochastic variation. The Kalman model is particularly simple in this case: the time differences are modeled as arising only from white phase noise and do not depend on the time of the measurements or the interval between consecutive measurements. If $x(t_n)$ is the time difference between the two devices at epoch t_n, then

$$x(t_n) = x(t_{n-1}) + \xi(t_{n-1}), \tag{50}$$

where ξ is a white noise process with zero mean and variance Q, so that its auto-correlation is

$$R_\xi(t_n, t_{n-1}) = Q\delta(t_n - t_{n-1}) \tag{51}$$

for all n. From Eq. (48), the evolution of the covariance is

$$P_K(-) = C(n-1) + Q. \tag{52}$$

We measure the time difference, $z(t_n)$, between the two devices at time t_n and the measurement process has a noise $v(t_n)$, which is a zero-mean random process with variance $M = \sigma_v^2$. The matrix H is simply 1 in this case, so that

$$z(t_n) = x(t_n) + v(t_n). \tag{53}$$

If we substitute Eq. (52) into Eq. (47), the Kalman gain is given by

$$K_k = \frac{P_K(-)}{P_K(-) + M} = \frac{C(n-1) + Q}{C(n-1) + Q + M}. \tag{54}$$

If we substitute this value for the Kalman gain into Eq. (46), we obtain a prediction of the current time difference state

$$\hat{x}\,(t_n) = x\,(t_{n-1}) + \frac{C\,(n-1)+Q}{C\,(n-1)+Q+M}[z\,(t_n) - x\,(t_{n-1})]. \tag{55}$$

Equation (55) can be re-arranged to yield

$$\hat{x}\,(t_n) = \left\{\frac{1}{C\,(n-1)+Q} + \frac{1}{M}\right\}^{-1}\left\{-\frac{x(t_{n-1})}{C\,(n-1)+Q} + \frac{z(t_n)}{M}\right\},$$

which illustrates that the updated state is the weighted combination of two independent statistical variables – the previous state and the current measurement. From Eq. (49), the updated covariance is

$$\begin{aligned} C(n) &= \frac{M}{C(n-1)+Q+M}\,[C(n-1)+Q] \\ &= \left\{\frac{1}{C(n-1)+Q} + \frac{1}{M}\right\}^{-1}. \end{aligned} \tag{56}$$

The fraction in the first form of Eq. (56) is always less than 1, so that the process of making the measurement reduces the covariance. The improvement in the covariance becomes smaller and smaller as the measurement noise increases, and becomes negligible when the measurement noise is much larger than the inherent noise of the devices themselves. The second form of Eq. (56) shows that the reciprocal of the updated variance is the usual combination of the variances of two statistically independent random variables.

XIV. THE COMPOSITE CLOCK ALGORITHM

The composite clock algorithm is used to characterize the clocks in the satellites and at the monitoring stations of the global positioning system. It is based on the Kalman formalism we have discussed above, but it is more complex because it also estimates the orbital parameters of the satellites and a number of other parameters. I will focus only on the portion of the algorithm that estimates the clocks.

The input data to the algorithm are pseudo-ranges: the measured time differences between the signals transmitted from each of the satellites to each of the ground stations in turn as the satellite comes into view of each of the stations. The model assumes that each time difference is an estimate of the states of the clock in the satellite and the clock in the ground station and has an additional, uncorrelated random noise contribution as described by Eqs. (33) and (34). The random noise of the measurement process is assumed to have zero mean and known covariance.

All of the clocks in the ensemble are modeled using a Kalman algorithm as in Eqs. (29)–(32) without the frequency aging terms. Thus, the transition matrix, $\boldsymbol{\phi}$ is 2×2 rather than 3×3 as in the previous discussion, and the other vectors are correspondingly smaller. As we have discussed above, the impact of frequency aging becomes more important at longer averaging times, so that ignoring this term can limit the maximum averaging time that can be used by the estimator. The composite clock algorithm currently computes a new estimate every 15 min, so that the effect of frequency aging is not a problem.

The algorithm considers that the state of each clock is a sum of the state of an ideal clock and the difference between the actual state of the clock and the state of the ideal one. Since the ideal clock is common to all of the members of the ensemble, it cancels in the time differences, and the Kalman equations of the reduced clocks are unchanged by subtracting the state of this ideal clock. In fact, subtracting ANY time value from the states of all of the clocks leaves the ensemble equations unchanged. This is a direct result of the fact that the ensemble algorithm uses the time differences between pairs of clocks; all time-scale algorithms share this property.

In order to resolve this ambiguity, the algorithm defines the implicit ensemble mean, which is the weighted average of the times of the corrected clock estimates.[26]

The sum that is used to construct the implicit ensemble mean is similar in spirit to the ensemble time defined in the AT1 algorithm, but the weighting algorithm is different. In both cases, the mean is constructed after removing the deterministic component of the time of each clock, and the differences between the implicit ensemble mean and the corrected time of each clock is just white noise in principle. Therefore, no clock exactly realizes the ensemble mean in principle, although the differences between the implicit ensemble means realized by each of the member clocks are small if the Kalman state parameters accurately reflect the performance of the clocks.

Since the implicit ensemble mean is the weighted average of several clocks, its stability should be better than any of the contributing members in principle. However, we would expect that it would share the long-period divergence of other time scales for the same set of reasons – the lack of perfect modeling of the long-period parameters of the clocks, the interaction between the measurement noise and the prediction error, the failure of the assumption that the measurement noise is a white phase-noise process, etc. The last of these problems can be particularly troublesome, since the pseudo-range measurements are affected by the tropospheric refractivity and by multi-path reflections at the receiver. Both of these effects can be difficult to estimate accurately.

The long-period stability and accuracy of the composite clock algorithm is maintained by steering the ensemble time to Coordinated Universal Time (UTC) as maintained at the U.S. Naval Observatory (USNO). In addition, a prediction of the time difference between the composite clock scale and UTC(USNO) is broadcast by each satellite. I will discuss the details of steering algorithms in a subsequent section.

XV. COMPARING THE KALMAN AND AT1 TIME SCALES

An important point in the derivation of the Kalman method is the discussion that derived Eq. (46) from Eq. (44). The derivation depends on the fact that the expectation value of the prediction error just before the measurement was unbiased. This requirement is difficult to realize and even more difficult to verify. To complicate matters, it is quite likely that

a biased estimate at any iteration will propagate into the future without administrative intervention.

The generic Kalman algorithm has nothing equivalent to the reset procedure of AT1. This is not particularly surprising, since a reset procedure is a non-statistical administrative intervention in the functioning of the scale that is justified on pragmatic grounds and will therefore always be outside of any purely statistical discussion. Jones and Tryon (op. cit) include a reset procedure in their version of the Kalman time scale that is based on the innovation and that is similar in spirit to the AT1 method based on the prediction error that is described above. However, the implementation is much more complicated, since a clock that is reset must also be removed from the covariance matrix. A reset of the reference clock also requires special handling in their formulation. No special handling is needed for this situation in AT1.

The reset procedure in AT1 modeled a large prediction error as a single time step and most other real-time algorithms do this as well. The algorithm drops the clock from the ensemble average, resets its time to match the predicted time offset and does not update its other parameters (frequency offset, etc.). The result is unambiguous, but might not be correct; the assumption that the large prediction error was due to a single time step is only one of the many possibilities of what might have happened. It is somewhat more difficult to do the same thing in a Kalman scale implementation. It is straightforward to adjust the time state of the clock to reduce the prediction of the time error to zero, and to set the appropriate parameters in the Kalman gain matrix to zero to prevent the measurements from updating the other components of the state vector for that clock. It is less clear how to deal with the covariance matrices, which describe properties of the ensemble as a whole. Jones and Tryon (op. cit.) allow for the possibility that the reset was due to a frequency step by increasing the covariance matrix elements for the frequency state so that the clock will be less likely to be reset on a subsequent computation cycle. The intent is for this adjustment to allow the algorithm to "learn" the new frequency over the next cycles.

XVI. THE KAS-2 KALMAN ALGORITHM

Since both the AT1 family of time scales and the Kalman family operate on time differences between the member clocks, the absolute times and frequencies of the clocks and, therefore, the time of the ensemble itself are free parameters. The AT1 algorithm provides an unambiguous, unique definition of the paper time of the ensemble as the time that makes the weighted average of the predictions of each of the clocks equal to zero on the average (apart from random noise). Neither the generic Kalman algorithm nor the Jones and Tryon variant have anything equivalent to this definition of the ensemble. Each clock is modeled with respect to the ensemble, but there is no prescription for combining the estimates to form a single ensemble time. I will discuss this point in more detail in Sec. XVII.

The deficiencies of the default Kalman algorithm are addressed in the KAS-2 (Ref. 25) implementation by adding additional constraints to the Kalman solution so as to provide an unambiguous definition of the ensemble parameters. The constraints are derived from the principle that the deviations of the times of the clock from a "perfect" device are random and uncorrelated and that the mean of the noise estimates is zero. This principle is applied to each component of the state of the clocks, so that there are three sets of weights, for the time state, the frequency state, and the frequency aging state. Since the mean of the deviations of the real ensemble states are not equal to zero in general, imposing this condition introduces correlations among the clock parameters.

The KAS-2 weights are chosen so as to minimize the noise of the time scale and subject to the additional condition that the sum of the weights must be unity. The details are similar to the weighting procedure in AT1: the weight of each component of the state of each clock in the ensemble mean is proportional to the inverse of the expectation of its corresponding noise parameter. The normalization requirement scales these weights by the sum of all of them as in AT1.

The KAS-2 algorithm includes a limit on the maximum weight that is implemented in very much the same way as AT1: if the calculated weight exceeds the maximum, it is limited to the maximum value and the other weights are recomputed with this limit. The use of a limit on the weights has the same advantages and disadvantages as in AT1: the method prevents the positive feedback loop in which a good clock eventually takes over the scale, but the performance is degraded because limiting the weight of a good clock implicitly gives poorer clocks more weight than they deserve.

XVII. COMPARING AT1 AND KALMAN ALGORITHMS

The long-term stability of the AT1 algorithm (and related algorithms such as ALGOS) is degraded by a clock with very good short-term stability and significant long-term frequency aging because the weight of the clock is largely determined by the short-term stability and because it is difficult to estimate an accurate value for the aging parameter in the presence of Gaussian and random-walk frequency noise. The Kalman formalism can address this possibility by explicitly including a noise parameter in the model for the frequency aging, which adds a consideration of the longer term noise performance to the model. The Kalman algorithm of Jones and Tryon (op. cit.) and the Kalman-based composite clock algorithm[26] used to estimate the clocks in the global positioning system do not use this parameter, but KAS-2 does (op. cit.). However, estimating the variance of the frequency aging is a difficult business because it requires long averaging times and because the aging usually is not stationary, so that the cure does not work as well in practice as we might expect.

A basic parameter in both the AT1 and Kalman algorithms is the innovation – the difference between the predictions of the model of the clocks in the ensemble and the measurements of the actual clock time differences. For example, Eq. (46), which scales the innovation by the Kalman gain factor, is formally very similar to Eq. (24) for the update of the frequency state in AT1.

This formal similarity can be extended by identifying the relationships between the weights used in AT1 and the weights used in a Kalman algorithm.[27] However, this formal similarity hides some significant differences. At the most

basic level, the weights in AT1 for the time state of each clock must satisfy a sum rule (Eq. (14)), and the weights in AT1 for the frequency update (Eqs. (24) and (25)) are constants that are determined administratively and are not adjusted by the algorithm. However, the definition of the ensemble is also different in the two time-scale algorithms.

To see this difference, we first compute the difference in the time states between two clocks with indices m and n in an AT1 ensemble. Using Eq. (22) twice, we obtain

$$x_{me}(k) - x_{ne}(k) = \{x_{re}(k) - X_{rm}(k)\} - \{x_{re}(k) - X_{rn}(k)\}$$
$$= X_{rm}(k) - X_{rn}(k), \tag{57}$$

where the upper case symbols are the measured hardware time differences and the lower case symbols are the times with respect to the ensemble, e. The hardware time differences are given by the difference of the physical times

$$X_{rm} = X_r - X_m,$$
$$X_{rn} = X_r - X_n, \tag{58}$$

so that the difference of the ensemble time states is exactly equal in magnitude to the difference of the hardware time measurements

$$x_{me}(k) - x_{ne}(k) = -\{X_n(k) - X_m(k)\}. \tag{59}$$

Since Eq. (21) is the same as Eq. (22), this equivalence is true even if a clock was reset because of a large prediction error. However, if we use Eq. (46) to compute the difference in the time states between two clocks in the Kalman scale, it is clear that the difference in the two time states will not, in general, be the same as the measured hardware time difference between the two clocks.

The reason that the difference in the time states in the AT1 time scale is the same as the difference in the physical time difference is that Eqs. (21) and (22) contain an implicit definition of the time of the ensemble with respect to the reference clock, x_{re}, that is the same for every one of the member clocks. Therefore, it cancels in the time difference in Eq. (57). The generic Kalman formalism does not contain an equivalent definition, which implies that the ensemble time implicitly defined by one clock is not necessarily identical to the ensemble time implicitly defined by another one. The composite clock description of Brown (op. cit.) discusses this point as well. These differences in the implicit definitions of the time of the ensemble are distinct from the overall arbitrary constant that can be added to all of the times of the ensemble clocks without changing the dynamics of the calculation. This arbitrary constant contributes to the "unobservable covariance" of the composite clock, and Brown provides a method for separating this contribution from the portion that is used to define the implicit ensemble time scale. This "implicit ensemble mean" uses weights that are derived from the variance of the clock states. As we discussed above the KAS-2 algorithm also defines a procedure for uniquely defining the time of the ensemble.

However, although AT1 defines an ensemble time uniquely, its definition of ensemble frequency is less clear. To illustrate this in a simple case, consider clock m and clock n in an AT1 ensemble with no frequency aging, so that d = 0 for both of them. If we observed the physical times of the clocks (by using a time-interval counter, for example), the average frequency over the averaging time τ would be given by

$$Y_{mn} = \frac{X_{mn}(k) - X_{mn}(k-1)}{\tau}$$
$$= \frac{\{x_{me}(k) - x_{ne}(k)\} - \{x_{me}(k-1) - x_{ne}(k-1)\}}{\tau}$$
$$= f_{me} - f_{ne}, \tag{60}$$

where X_{mn} is the measured time difference between clocks m and n. Thus, the average frequency difference measured by the hardware is the same as the difference of the two frequencies with respect to the ensemble. However, if we compute the difference in the frequencies of clocks m and n with respect to the ensemble by using Eq. (24) twice, we find that the difference in the frequency states of the two clocks is not the same,

$$y_{me}(k) - y_{ne}(k) \neq f_{me}(k) - f_{ne}(k). \tag{61}$$

In fact, the difference in the frequency states at time t_k is a function of the current frequency estimate combined with the previous frequency states, and the result is similar in form to what Eq. (46) would have predicted – the difference in the frequency states is a combination of the previous frequency states and the current measurements.

The differences between the two algorithms are not as surprising as they seem at first, and are actually simply a result of the assumptions that were used to construct them. The AT1 algorithm is designed on the basis that the measurement noise of the time differences is negligible, so that the measurements represent the best estimate of the time differences of the clocks. This is also true even for a clock whose prediction error is large enough to cause a reset – the algorithm assumes that the problem is internal to the clock, and difference of the time states is set to the measured physical time difference parameters. The AT1 algorithm will not do the right thing if the noise is really in the measurement system, which is time noise without any associated frequency fluctuation.

The BIPM ALGOS algorithm, which is used to compute EAL and TAI is similar in concept. It uses a longer averaging time, so that the underlying noise type is assumed to be closer to flicker or random walk frequency noise. In either case, the measurement noise of the time differences is assumed to be very small compared to the contribution of the frequency variance to the measurements. Therefore, the measured hardware time differences are assumed to represent the true time differences between any pair of clocks, and the time states of the clocks are set to agree with the hardware measurements. Likewise, the time of the ensemble is uniquely defined based on the same considerations.

On the other hand, the AT1 estimate of the frequency of the clock with respect to the ensemble is modeled as having white frequency noise, and so the frequency state of the clock is not (and should not be) set to the frequency over the last measurement interval but to an average of those estimates and the prior values, with a time constant defined by the range of

validity of the assumption that the frequency noise is a random variable. Therefore, it is natural that the difference in the frequency states of two clocks is not the same as what would be calculated from the measured time differences over any single averaging time. The ALGOS algorithm uses longer averaging times and therefore assumes that the frequency noise is closer to flicker or random walk. The optimum estimate in this case is the value at the end of the last interval rather than the average value used by AT1. In order for this assumption to be valid, the ALGOS algorithm must have already averaged the white frequency noise in the 5-day measurements, which are the basis for the computation. Again, the frequency estimated over any single averaging time is not going to be the same as the frequency state of the clock in the ensemble. The details in AT1 and ALGOS are different, but this conclusion is the same in both procedures.

The Kalman algorithm, on the other hand, makes no such *a priori* assumptions about the underlying noise sources or types. The hardware measurements provide information on the time differences between the clocks, but, in addition to the effects of the frequency noise, those measurements have noise both from the measurement process and from the white phase noise of the clocks themselves. Therefore, the time difference measurements should have a vote in specifying the difference in the time states of the clocks, but not a veto. The previous difference in the time states also has information about the current difference in those states, so that the current time difference should be a combination of the current measurements and the previous state. That combination will be more appropriate for averaging the white phase noise of the clocks and the measurement system.

The Kalman algorithm has the ability to model random frequency aging, but the noise parameter to implement this ability is not used in the composite clock used in the global positioning system or in the algorithm of Jones and Tryon. It is used in KAS-2, but this advantage is mitigated to some extent by the difficulty of specifying a robust value for this parameter.

The problems of dealing with frequency aging would be less serious in a time scale that did not have to run in real time, and TAI is not computed in real time for that reason. Any algorithm that can "know the future" is better able to cope with the present, especially with the long-period variations in the frequency that are characteristic of hydrogen masers with frequency aging. This luxury of post-processing is not available to timing laboratories or to any real-time application. The usefulness of a retrospective time scale, such as TAI (or UTC, which is derived from it), decreases as real-time applications become more common. On the other hand, a retrospective time scale will always have advantages in being able to cope with data anomalies and frequency aging, and there will continue to be a tension between ultimate stability and accuracy on one hand and real-time performance on the other.

The assumptions that are the basis for the AT1 algorithm limit the range of time intervals between computations, but this is not a significant effect for the NIST implementation of AT1. The measurement noise is small enough that the noise due to the frequency dispersion is likely to be larger than the measurement noise for almost any reasonable time interval longer than a few seconds (Eq. (7)). However, the interval between measurements cannot be increased arbitrarily without running into the problem that the frequency noise is no longer well characterized as a random variable. The upper limit depends on the type of clocks used in the ensemble, but is probably at least 24 h for typical cesium devices. As I discussed above, the BIPM uses a measurement interval of five days in the computation of EAL and International Atomic Time, and this requirement would permit even longer intervals – perhaps as long as 3 months, for typical cesium devices.

However, an AT1 algorithm that used data from conventional time interval counters would have a more significant lower bound to the interval between computations because the measurement noise would be larger than for dual-mixer systems, and the requirement that this contribution be small compared to the clock noise is not so easy to realize for averaging times less than 1 or 2 h.

The ALGOS algorithm has the same sort of problem, since the time difference data are often derived at least partially from measurements that use the global positioning satellites. For example, if the time transfer noise was of order 1–2 ns and if the frequency fluctuations of the clock were of order 10^{-14}, the minimum interval between computations would be 1–2 days, and a larger interval would probably be desirable. Therefore, the existing 5-day computation cycle can be shortened only to some extent if the time differences are to be observed by the use of navigation satellites. However, two-way satellite time transfer and more sophisticated use of data from navigation satellites could reduce the measurement noise by a factor of 10 or more, which could be translated into a correspondingly more rapid computation.

XVIII. INCORPORATING FREQUENCY DATA INTO THE TIME SCALE

The Kalman algorithm can support incorporating frequency measurements in addition to the time-difference measurements that are the basis for most time scales. The observation matrix (Eq. (34)) would be modified to include these data, and the rest of the algorithm would be unchanged. This is not a common practice, at least to some extent because frequency difference data are not generally available – almost all systems measure time differences and infer frequency from them.

The data from a primary frequency standard could be an exception to this principle, since most primary frequency standards operate only occasionally and do not run as clocks. Therefore, they are a source of frequency but not of time. However, NIST no longer uses a Kalman scale, so that this method of incorporating data from a primary frequency standard is not available. A simpler and more direct method is used instead.

The frequency estimates from the primary frequency standard at NIST are used to characterize the frequency of one of the masers, which is also a member of the time scale ensemble. Consecutive estimates of the frequency of the maser with this method provide information both on the frequency of the maser in terms of the definition of the second and also

on the frequency aging of the maser. The frequency of the maser determined in this way is used to transmit the primary frequency data to the BIPM, and the frequency aging estimate is used to adjust the frequency aging parameters of the masers in the AT1 time scale. (As discussed above, the AT1 software uses these parameters but treats them as constants.)

Since the stability of the masers and the ensemble is comparable to the accuracy of the primary frequency standard, a running average of three determinations is typically used to provide an estimate of the aging parameter. A typical value for the aging would be about 10^{-21} s^{-1}, but there is considerable variation both with time and from one maser to another.

This process can also be used to estimate the frequency aging of a cesium device, but the value of the aging parameter is so small that the averaging time needed to estimate it in the presence of measurement noise and white frequency noise is too long to be useful – on the order of years. Not only is this too long to be useful from the point of view of time-scale operations, but it is not much shorter than the life of a cesium tube, so that the determination would not be very useful in practice. Some preliminary data suggest that the magnitude of the aging might be of order 2×10^{-22} s^{-1}. The long-period frequency variations of a cesium clock are generally not less than 2×10^{-14}, so that the impact of the frequency aging on the frequency is comparable to the frequency noise only after about 1200 days. This value can be shortened if the ensemble is operated in a temperature-controlled environment. The frequency noise of a cesium clock will often be less than 1×10^{-14} in this environment, so that averaging time needed to estimate the frequency aging can be shortened by a factor of 2–5.

As we noted above, the AT1 algorithm is designed as a *time* scale – the ensemble time and the time of each of the member clocks are well defined on each measurement cycle. On the other hand, the frequency of the ensemble is a more complicated function of the frequencies of the member clocks, and the definition of the ensemble frequency is more complicated.

It is possible to re-cast the AT1 algorithm to make it more useful for incorporating frequency calibrations. The AF1 algorithm[28] defines an ensemble frequency and frequency aging based on the weighted average of these parameters derived from the frequency data of the members of the ensemble. The algorithm facilitates a comparison between the ensemble frequency and the frequency of a primary frequency standard. It starts from the same time-difference data that are used in the AT1 calculation, and it provides a different perspective on these measurements. For example, it is better able to detect frequency steps in the ensemble clocks and is better able to estimate frequency aging, at least to some extent because it does not have to run in real time. However, it has many of the same weaknesses of AT1, and cannot function without administrative weight limits or a reset algorithm, both of which are very similar to the AT1 equivalents.

Another approach to modeling the frequency of the NIST clock ensemble is the TA2 algorithm.[29] The algorithm estimates both frequency and time steps by running in both directions over the data. The bi-directional analysis provides a more robust estimate of time and frequency steps, since the algorithm can see the future at every analysis epoch. It has been used at NIST on an experimental basis, but it cannot run in real time and is not a substitute for AT1.

XIX. OTHER CLOCK MODELS AND MODELING A SINGLE CLOCK

The algorithms we have described above are intended specifically for estimating the performance of an ensemble of clocks. However, the same clock models used to define ensembles can also be used to describe the performance of a single clock with respect to some other timing reference. In the simplest case, the timing reference is modeled as being much more stable than the device under test, so that all of the variance in the data is attributed to the clock being characterized. In addition to the clock models used in ensembles, there are other clock models that are limited to estimating the performance of a single clock.

Some of these models are purely phenomenological. For example, the current time difference of the clock with respect to the reference can be modeled as a linear combination of the current time-difference measurement and N previous ones[30]

$$\hat{x}_k = \sum_{i=0}^{N} a_i x_{k-i}. \tag{62}$$

The coefficients of the terms in the summation are chosen so as to minimize the RMS difference between the predicted values and the measurements. (Note that the time state of the clock after the current measurement defined in this way is generally not equal to the value that was reported by the measurement hardware on this cycle. Thus, the difference in the time states of two clocks measured with respect to the same reference and estimated in this way will not, in general, be equal to the time difference that would be measured between these two clocks by a direct hardware connection. This is the same issue as we discussed with Kalman methods and arises for basically the same reason – the current measurement has a vote in determining the time state but the previous measurements do too.)

These models have a "finite impulse response" in the sense that the model "remembers" only the N most recent results and ignores older values. Since the frequency of a clock is based on the first difference of the time differences and the frequency aging is based on the second differences, Eq. (62) with $N = 2$ can be considered to be a different formulation of the models we presented in Eqs. (1)–(3). If the coefficients are determined by standard least squares then this will result in stationary, unbiased estimates if (*and only if*) the variance in the measurements is at least approximately a random process. The coefficients will have a bias if the noise process has a non-zero mean, but this mean will be absorbed into the constants if it is really unchanging, and this does not degrade the determination of the coefficients. As we have discussed several times, this is likely to be true for short time intervals and not adequate in the longer term.

If the algorithm is used as is, then a time step or measurement error will persist for N cycles and then be forgotten. However, it will modify the determination of the coefficients

during those N cycles, and it will take some time for the coefficients to return to the steady-state values. Alternatively, a "reset" algorithm could be included in which the prediction error on every cycle was compared against some average prediction error based on previous cycles, and a measurement that was considered to be an outlier could be corrected to match the expected value.

Two more complicated situations are common. In the first case, the channel used to support the time-difference measurements has noise processes that both make a significant contribution to the variance of the data and are also very far from a white random process. None of the methods we have described is adequate for this situation, which arises routinely in comparing clocks over a public network such as the Internet. A number of methods have been developed for dealing with this situation, but none of them is really adequate to the job because the fluctuations in the network delay are not easily described by stationary statistical models or by RMS noise parameters.[31,32]

The second complication arises when time difference measurements become expensive in terms of a scarce resource, such as network bandwidth, computer cycles or some other parameter. This complication introduces the requirement for a cost/benefit analysis, where the accuracy of a clock synchronization algorithm must be balanced against the cost of realizing it. This issue is important in two-way satellite time transfer, which is used by many national laboratories to compare national time scales and primary frequency standards. These comparisons form an important part of the realization of the SI second, and the increasing cost of satellite time has necessitated experiments to evaluate the impact of reducing the bandwidth of the transmissions on the accuracy and stability of the frequency comparisons.[33]

The cost/benefit analysis is also a consideration for Internet-based time services, which estimate the one-way transmission delay as one-half of the measured two-way value. Both stations must maintain intermediate results while each connection is in progress so that there is a direct relationship between the number of simultaneous clients that can be served and the size and speed of the time server.

One method of reducing the load on Internet time servers is to increase the time between queries from any one client. If the variance in the data can be modeled as white phase noise, the cost of the algorithm decreases linearly with the polling interval while the accuracy degrades only as the square root of this interval. There is a smaller advantage even when the noise process is characterized as flicker phase noise. This is an important result, since the accuracy required by a client application is often much less than the accuracy that can be realized by the algorithm, so that the degradation of performance that accompanies the increased polling interval has no significant impact.[34]

XX. TIME SCALE STEERING

I now turn to a discussion of ensemble steering – adjusting the parameters of an ensemble in response to external data. I have already discussed in a general way the use of data from primary frequency standards to generate TAI from EAL, the free-running ensemble of commercial cesium standards and hydrogen masers. Other steered scales are commonly used: the composite clock, the time scale used by the global positioning system, is steered to Coordinated Universal Time as realized by the U.S. Naval Observatory, or UTC(USNO). The time scale distributed by all NIST time services, UTC(NIST), is steered towards UTC as computed by the BIPM. Many timing laboratories and National Metrology Institutes also maintain a local realization of UTC, which is steered by the use of UTC data published by the BIPM in Circular T.[13] This document, which is issued monthly, lists the time difference every five days between UTC as computed by the BIPM from EAL and TAI and the realization of UTC by each laboratory, which is designated as UTC(lab).

All steering methods are based on the fact that adding a value to the time of every member of an ensemble modifies the time states of the clocks but does not have any effect on the ensemble computation, which depends only on time differences. Since the times of the member clocks are always expressed with respect to the ensemble time, adding a value to these state variables is equivalent to subtracting the same value from the ensemble time state. If the value that we added to the time states at time t_k is T_s and if T_s is defined by

$$T_s = T(t_0) + Y(t_k - t_0) + \frac{1}{2}D(t_k - t_0)^2, \tag{63}$$

then the effect will be to steer the time, frequency, and frequency aging of the ensemble by -T, -Y, and -D, respectively, starting from the origin time t_0. There is nothing special about implementing the steering by adjusting the time state – steering in frequency or in frequency aging could also be implemented by adjusting the corresponding state variable for every clock by the negative of the adjustment to the ensemble.

The process of applying a steering correction (or modifying the parameters of an existing steering operation), always introduces a discontinuity in the state parameter of the ensemble that is being steered. Pure time steering is almost never used for this reason, because many user processes cannot cope with discrete time steps, especially if the adjustment is negative so that time appears to run backwards. (Forward time steps are not much better, since a forward time step implies that some clock readings may never exist, and a process that is waiting for a specific time to arrive may wait forever.)

The only application that routinely uses steering of frequency aging is the composite clock algorithm used by the global positioning system. The GPS controllers steer the time scale using discrete steps of $\pm 10^{-19}$ s^{-1} in the frequency aging parameter. (so called "bang-bang" steering). A steer in the frequency aging of this magnitude introduces a frequency change of about 10^{-14} and a time steer of about 1 ns after 1 day. These changes are small enough to be ignored by almost all users of the system. This is a steer of the *ensemble* time scale; since the navigation solution depends on differences between signals from the various satellites in view, it is not affected by this steer in first order. There may be an effect in second order, since different satellites in the constellation can learn of the steer at different times. Therefore, the steer can introduce a temporary inconsistency in the ensemble parameters broadcast by the different satellites, which would

have an impact on the accuracy of the navigation solution. It can also affect users who use simultaneous observations of several satellites to compute the time of a station clock with respect to the average of all of the measurements, since this calculation links the data from all of the satellites to the ensemble time scale. All of these effects would be larger if the same steering were implemented as a single frequency steer.

The steer of the composite clock is completely transparent to an application that uses the physical signal from a single GPS satellite in common-view, where several stations observe the same satellite at the same time, and compute the difference of the station clocks by subtracting the arrival times of the signal at each site as measured by a local clock. The offset of the satellite clock from the composite clock time scale cancels in this difference. The situation with regard to the clock on the satellite is somewhat more complicated if the transit times of the signals from the satellite to each of the ground stations are not the same. (This is almost always the situation, since the satellites move across the sky in many different directions as seen from the ground stations.) If the ground stations subtract signals that left the satellite at the same instant then the properties of the satellite clock are irrelevant, but the signals arrive at the ground stations at different times, and this complicates the subtraction process, since the difference in the arrival times varies as the satellite moves across the sky. On the other hand, if the ground stations subtract signals that arrived at the same time as measured by the ground clocks, then they left the satellite at different times and the stability of the satellite clock (and possibly the motion of the satellite) during this time difference will become important.

When the frequency of the ensemble is an important parameter (as with TAI), it is more common to apply the steering correction to the frequency itself rather than to the frequency aging. Although frequency steering introduces a discontinuity in the frequency when the steering parameters are modified, the frequency between changes has the full stability of the ensemble. This is not true for steering by modifying the frequency aging – the average frequency over every interval is different. This is why frequency steering is used by the BIPM to compute TAI from EAL. Frequency steering is also used by NIST (and other timing laboratories) to compute UTC(lab), a local realization of Coordinated Universal Time, based on data from a local ensemble time scale.

At NIST, UTC(NIST) is defined as an offset from the atomic time scale, AT1, by Eq. (63) with the frequency aging term set to zero. The parameters are defined so that the time offset defined by the equation is continuous whenever the frequency steering parameter is modified. That is, $T(t_0)$, the origin time offset for any steering equation, is exactly equal to the last value of T_s of the preceding equation. The UTC(NIST) time scale is therefore realized as a piecewise-linear offset from AT1 that is continuous in time with discrete steps in frequency.

However, there is a fundamental difference between UTC(NIST) and paper time scales such as TAI or the GPS composite clock. The UTC(NIST) time scale is used as the reference time for all of the NIST services, and it therefore must have a physical realization – there must be physical signal whose time realizes the definition. This is not true of the other two scales, where it is enough to compute simply the offset between contributions to the time scale and the ensemble mean and to list or broadcast these offsets. No clock need actually realize the ensemble time. Because of this difference, the NIST steering equations are actually applied to a physical clock rather than to the parameters of a paper ensemble.

The physical hardware used at NIST is typical of the configuration used at other timing laboratories. One of the physical clocks in the ensemble is used as the hardware reference for a phase stepper – a device that produces an output signal that can be offset from its input signal in phase or in frequency in response to external commands. (It is possible, although less common, to steer a physical clock rather than a separate phase stepper, but this does usually does not work as well because the physical clocks are typically not designed for this application.) The output of the phase stepper is processed by the time scale as if it were a real clock, except that its weight in the ensemble definition is set to zero for all time. The prediction error of the steered clock is the difference between its time with respect to the ensemble and the value predicted by Eq. (63) for that value of t_k. After each computation of the time scale (every 12 min at NIST), a steering command is sent to the phase stepper to drive the prediction error to zero. The magnitude of the correction is discussed below.

The stability of the steered clock at NIST is defined by different considerations in different time domains. For times less than the cycle time of the scale (12 min at NIST), the steered clock is free-running and its stability depends on the stability of the clock used as the reference for the phase stepper and the stability of the phase stepper itself. A hydrogen maser is used at NIST as the reference for the phase stepper because of its very good short-term stability, which is about $2\text{-}3 \times 10^{-15}$ for these averaging times. The free-running stability of the frequency of the steered clock combined with the phase noise of the phase stepper and of the measurement system produces a time dispersion of a few picoseconds between measurement cycles.

For times much longer than the cycle time of the scale, the control loop drives the steering error to zero, and the stability of the steered clock is identical to the free-running stability of the scale itself. Over the last few years, the free-running stability of the scale has been about $1\text{–}2 \times 10^{-15}$ for averaging times out to 30–40 days; this value varies somewhat from month to month. The situation at intermediate averaging times depends on the details of the steering control loop, and we now discuss the considerations that are important to its design.

All of the frequency-steering loops we have discussed must try to satisfy two conflicting requirements. If the uses of the steered output depend on the accuracy of the time with respect to some external reference, then errors in the time of the steered output should be removed aggressively by the use of frequent steering changes with large frequency offsets, if necessary. The frequency stability of the steered output will be degraded by these frequent steers, but that is a secondary consideration in this model. On the other hand, if frequency stability is the primary goal, then steering commands should be infrequent with only small changes from the previous command. The time accuracy will be degraded in this model, since

time errors will be removed slowly, and large time offsets may persist for an appreciable period. (Steering by small changes in the frequency aging parameter, as is done for the GPS composite clock, also produce slow changes in the steered time output, since the time depends on the square of the interval between cycles.)

As a practical matter, this choice is not too significant in designing the control loop that constructs UTC(NIST) from the AT1 time scale, because the free-running stability of the steered clock results in a time offset from the steering equation of only a few picoseconds over 12 min, so that the steering commands are normally very small and the choice between frequency stability and time accuracy does not arise. However, this conclusion may not be true for other clock ensembles.

The implementation of the NIST steering system described in the previous paragraphs implicitly assumed that the time scale ensemble, which is used as the reference for the steering equation, is much more stable than the steered clock, so that the deviation of the time of the steered clock from the prediction of the steering equation can be attributed completely to the steered clock. This is a reasonable assumption at NIST, since the clock ensemble has a number of masers as members, and an ensemble of masers is generally more stable in short term than any one of them. However, this would probably not be true for an ensemble that was composed primarily of cesium standards with only one maser as the reference for the phase stepper that implemented the steered clock. The free-running stability of the steered clock could be better than the free running stability of the entire ensemble for short averaging times (on the order of the 12 min cycle time of the algorithm), so that the steering method described above would degrade the short-term stability of the steered clock by adding ensemble noise. A steering algorithm that averaged the steering corrections over a longer period could provide better short-term stability, although it would degrade the ability of the algorithm to detect and respond to errors in the steered clock. There are additional issues that must be considered in implementing the steering needed to construct TAI from EAL or UTC(NIST) from AT1.

In addition to the compromise that is needed between time accuracy and frequency stability discussed above, there is also a consideration of frequency accuracy. For example, based on data from primary frequency standards, the BIPM estimates that the frequency aging of EAL was about 6×10^{-15} during 2009, and TAI would have to be steered so as to remove this aging. If the frequency adjustments needed to remove this aging were applied in 12 equal monthly installments (which is about as often as is practical based on administrative considerations), the monthly frequency adjustments would be 5×10^{-16}, which is roughly the free-running stability of EAL for an averaging time of 1 month. Thus, the steering would have made a significant contribution to the free-running stability of the scale. The problem would be worse if the steering corrections were applied less frequently because they would have to be larger to keep up with the aging of the input scale.

This discussion highlights a relationship between the long-term frequency aging that the steering is intended to remove and the short-term frequency stability of the scale. If frequency stability is a consideration, as is true for TAI, the maximum frequency adjustment that can be used at any time is limited so as not to degrade the frequency stability of the scale unduly. Since there is no physical realization of either EAL or TAI, this problem can be avoided by tabulating some fraction of the frequency corrections without actually applying the full correction to the scale. Applications that needed frequency stability would not use the tabulated corrections, whereas applications that needed accuracy would do so.

This solution cannot be used at a timing laboratory such as NIST, because the steered clock has a physical realization. It would be possible, in principle, to provide a physical realization with an ancillary table of corrections, but this is not a very practical solution for most users. Accurate modeling of the frequency aging of the hydrogen masers in the NIST ensemble is very important for this reason – we cannot simultaneously provide time accuracy, frequency stability, and frequency accuracy without it.

To further complicate matters, the information used to compute the steering equation used to steer UTC(NIST) to UTC is available only with a delay of several weeks, so that the steering involves an extrapolation into the future by up to 30 days. The free-running stability of the NIST time scale, which is the reference for the steering equation, is about 1–2 $\times 10^{-15}$ for averaging times of this order, so we would expect that the time dispersion of UTC(NIST)-UTC could be as large as $30 \times 86\,400 \times 2 \times 10^{-15} = 5$ ns RMS. This time dispersion is limited primarily by the 30 day delay in the information about the difference UTC(NIST) – UTC. The previous estimate used a linear relationship between time dispersion and averaging time; the actual relationship would depend on the characteristics of the frequency stability of the time scale. A more optimistic estimate would use a dependence proportional to the square root of the averaging time, which would predict a time dispersion of about 1 ns RMS. Figure 4 shows the values of UTC – UTC(NIST) for the last 10 years. The RMS performance is somewhat better than the conservative estimate above because the frequency stability varies from month to month because of environmental perturbations and other effects.

The considerations that govern the maximum frequency adjustment in the NIST steering algorithm are limited by the frequency stability of the NIST free-running scale, AT1. The frequency used in the steering equation should not change by more than about 2×10^{-15} (0.17 ns/day) in any month to preserve the frequency stability of the NIST time scale, UTC(NIST). The frequency aging of the NIST time scale has been smaller than this limit in recent years, although this was not always true. Figure 5 shows the frequency steers that have been applied to the NIST atomic time scale to realize UTC(NIST). The rate is shown in ns/day, where 1 ns/day $= 1.16 \times 10^{-14}$.

In summary, steering a time scale requires a balance among competing requirements, and different laboratories will adopt different strategies as a result. The steering used by a timing laboratory to construct UTC(lab) has additional complexity because the data are available only after a significant delay, so that the effects of frequency and frequency aging

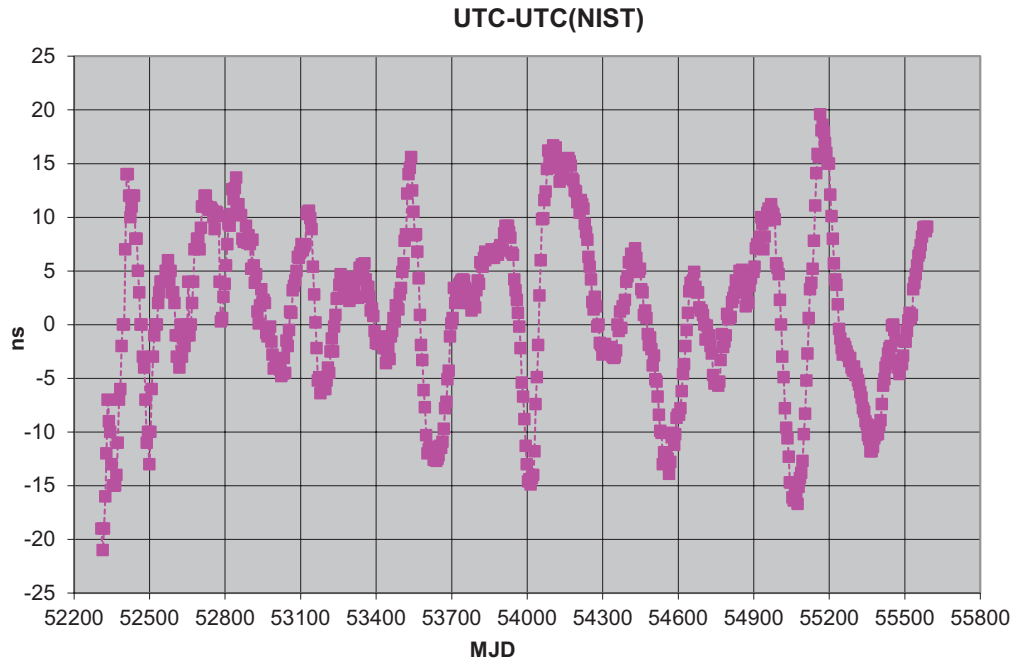

FIG. 4. (Color online) The difference between UTC as computed by the BIPM and UTC(NIST), the realization of that time scale at NIST. The UTC(NIST) time scale is constructed by applying a steering correction to the free-running atomic-clock time scale AT1 as described in the text. The modified Julian day number (MJD) is an integer count of days and is commonly used in time scale work since it is easy to compute time intervals using it. The limits of the plot, MJD values 52 200 and 55 800, correspond to 18 October 2001 and 27 August 2011, respectively.

fluctuations are not detected until well after they have happened. Therefore, the steering correction will always lag the variation that it is intended to remove. This time lag is important because the stochastic frequency aging of the time scales mean that the correction is not stationary.

The time lag discussed in the previous paragraph is equivalent to a phase shift in the steering control loop, and the loop can oscillate at the period where the time delay is equivalent to a phase shift of 180°. To prevent this oscillation, the gain of the control loop must be less than unity at this

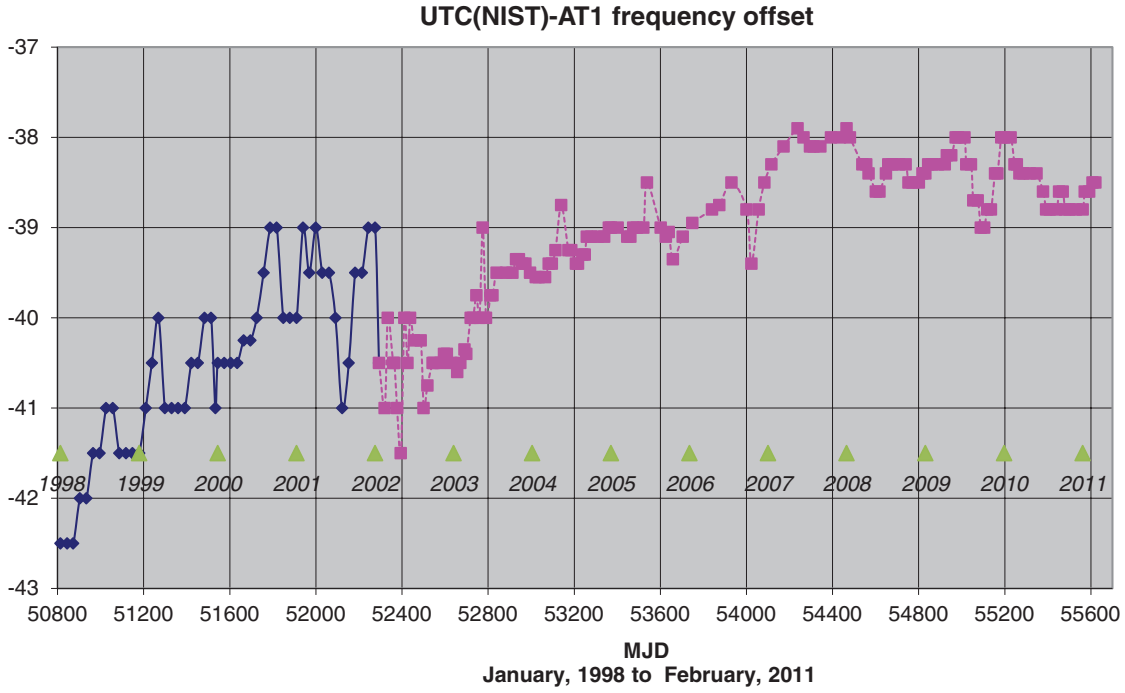

FIG. 5. (Color online) The frequency steering adjustment applied to AT1 to produce UTC(NIST). The frequency adjustments are applied with no time steps so that the UTC(NIST) – AT1 is a piecewise-linear function as described in the text. The vertical scale is in units of ns/day, where 1 ns/day = 1.16×10^{-14}. The modified Julian day number (MJD) is an integer count of days and is commonly used in time scale work since it is easy to compute time intervals using it. The limits of the plot, MJD values 50 800 and 55 600, correspond to 18 December 1997 and 8 February 2011, respectively.

period, and this generally further limits the maximum steering changes that can be used. The steering control loop at NIST operates close to this limit, so that some oscillations in the values of UTC(NIST)-UTC are present in Fig. 4. The steering corrections in Fig. 5 before the middle of 2002 show the oscillation in the steering corrections when the gain in the control loop was too high for the time delay in reporting the values of UTC–UTC(NIST). The loop delay was shortened in 2002, which removed the oscillations. The variations in UTC–UTC(NIST) after this time are comparable to the frequency stability of the time scale. It is also possible that they may also be due to long-period random walk of the frequency of AT1, which only appears to be quasi-periodic over the relatively short interval of a few years.

XXI. DISCUSSION AND CONCLUSIONS

We have presented the design considerations that define the time scale algorithms that are currently used to realize the national and international standards of time and frequency, and we have discussed the AT1 algorithm and the Kalman algorithm in some detail. These two algorithms form the basis for most of the time scale algorithms that are in current use. Both of the algorithms have strengths and weaknesses, and neither is obviously superior for every application.

The weaknesses of either algorithm can be mitigated, at least to some extent, if the analyses can be done in a retrospective mode, since many problems are much more easily detected when an analysis can see the future as well as the past at each epoch. The BIPM computes International Atomic Time in this way, and NIST also uses a retrospective version of AT1, which includes administrative tuning to model and remove both frequency and time steps, to assist in evaluating the NIST primary frequency standard.

Although retrospective time scales have many theoretical and practical advantages, the applications that depend on time and frequency information (such as telecommunications and navigation) and the time services of national laboratories, depend on ensemble estimates computed in real-time, with no possibility for going back to re-write history. Finding better real-time algorithms will continue to be an important research area, especially in modeling clocks and channels that have non-Gaussian noise characteristics. Many timing systems are sensitive to ambient temperature variations, so that methods of modeling the admittance to diurnal or nearly diurnal fluctuations will become increasingly important in the future. One way of doing this is to include periodic terms into the covariance matrix of a Kalman algorithm.[35] These models are complicated because the amplitude and the phase of the admittance are not constant values and the variations in these parameters may not be stationary. The lack of a stationary admittance is a serious limitation for Kalman methods.

The development of the next generation of primary frequency standards is unlikely to eliminate the need for time scale algorithms. In fact, the reverse is likely to be true, for a number of reasons. Many of these standards do not operate continuously as clocks so that a time scale is needed for supporting time and frequency distribution methods, for providing a short-term reference that is needed to evaluate the residual systematic errors in these primary standards, and for comparing primary standards that do not operate at the same time.

Note Added in Proof: The approximate frequency aging (discussed in Sec. VII) continued during the remainder of 2011, and the frequency steering of EAL-TAI was 6.526×10^{-13} in January, 2012.

[1] *Explanatory Supplement to the Astronomical Ephemeris*, edited by P. Kenneth Seidelmann (HM Stationery Office, London, 1961), Chap. 3; D. D. McCarthy and P. Kenneth Seidelmann, *Time: From Earth Rotation to Atomic Physics* (Wiley-VCH, Weinheim, Germany, 2009), Chaps. 1 and 2; See also J. Jesperson and J. Fitz-Randolph, *From Sundials to Atomic Clocks—Understanding Time and Frequency* (Dover, Mineola, New York, 1999).

[2] E. M. Reingold and N. Dershowitz (Cambridge University Press, Cambridge, England, 2001).

[3] T. J. Quinn, Proc. IEEE **79**, 894 (1991).

[4] R. A. Nelson, D. D. McCarthy, S. Malys, J. Levine, B. Guinot, H. F. Fliegel, R. L. Beard, and T. R. Bartholomew, Metrologia **38**, 509 (2001).

[5] W. Markowitz, R. G. Hall, L. Essen, and J. V. L. Perry, Phys Rev. Lett. **1**, 105 (1958).

[6] *Resolution 9 of the 13th Coférence Générale des Poids et Mesures (CGPM)*, 1960. Sèvres, France, The International Bureau of Weights and Measures (BIPM).

[7] W. M. Itano, J. C. Bergquist, T. Rosenband, D. J. Wineland, D. Hume, C. W. Chou, S. R. Jefferts, T. P. Heavner, T. E. Parker, S. A. Diddams, and T. Fortier, in *Proceedings 2009 ICOLS Conference* (World Scientific, Hackensack, New Jersey, 2009), pp. 117–124; Also available on the web from publications database at tf.nist.gov, paper 2391.

[8] See http://www.ngs.noaa.gov/GEOID for information about the definition and realization of the geoid and its relationship to mean sea level.

[9] S. R. Stein, *Chapter 12 in Precision Frequency Control* (Academic, New York, 1985); Reprinted in NIST Technical Note 1337, edited by D. B. Sullivan, D. W. Allan, D. A. Howe, and F. L. Walls, Boulder, Colorado, NIST, 1990.

[10] Judah Levine, Rev. Sci. Instrum. **70**, 2567 (1999), especially Sec. VIII.

[11] G. E. P. Box and G. M. Jenkins, *Time series Analysis: forecasting and control* (Holden-Day, Inc., San Francisco, 1970), see especially Chap. 3.

[12] S. Stein, D. Glaze, J. Levine, J. Gray, D. Hilliard, and D. Howe, in *Proceedings of 36th Annual Frequency Control Symposium* (IEEE, Piscataway, NJ, 1982); Reprinted in NIST Technical Note 1337, S. Stein, D. Glaze, J. Levine, J. Gray, D. Hilliard, and D. Howe, edited by D. B. Sullivan, D. W. Allan, D. A. Howe, and F. L. Walls, Boulder, Colorado, NIST, 1990. Available on the web from publications database at tf.nist.gov, paper 610.; See also S. Stein, D. Glaze, J. Levine, J. Gray, D. Hilliard, and D. Howe, IEEE Trans. Instrum. Meas. **32**, 227 (1983); NIST publications database paper 599.

[13] S. Romisch, T. E. Parker, and S. R. Jefferts, in *Proceedings of 2009 Precise Time and Time Interval Planning and Applications Meeting* (US Naval Observatory, Washington, DC, 2009), pp. 397–408; Also available on the web from NIST publications database as paper 2442.

[14] The Circular T is published monthly and is available from www.bipm.org.

[15] BIPM Annual Report on Time Activities, Vol. 4, 2009. Available from www.bipm.org.

[16] The bulletin is published every month. See http://tf.nist.gov/pubs/bulletin/timescaleindex.htm

[17] B. Guinot and C. Thomas, "Establishment of International Atomic Time," Annual Report of the BIPM Time, Sec. **1** (1988). Available at www.bipm.org. See also the next reference.

[18] G. Panfilo, A. Harmegnies, and L. Tisserand, in *Proceedings of 2011 Joint European Time and Frequency Forum and International Frequency Control Symposium* (IEEE, Piscataway, NJ, 2011), pp. 850–855. G. Panfilo and E. F. Arias, in *Proceedings of 2009 Joint European Time and Frequency Forum and International Frequency Control Symposium, Besancon, France, 2009*, pp. 110–115.

[19] P. Tavella and C. Thomas, Metrologia **28**, 57 (1991).

[20] G. Petit, Metrologia **40**, S252 (2003).; See also J. Azoubib, in *Proceedings of 32nd Annual Precise Time and Time Interval Planning and Applications Meeting, Reston, Virginia, 28-30 November 2000*, (US Naval Observatory, Washington, DC, 2001), pp. 195–210, available online at www.pttimeeting.org.

[21]P. Tavella, J. Azoubib, and C. Thomas, in *Proceedings of 5th European Frequency and Time Forum, Besancon, France*, 1991 (Swiss Foundation for Research, Neuchatel, Switzerland, 2001).
[22]A. Gelb, *Applied Optimal Estimation* (MIT, Cambridge, MA, 1974), see especially Chap. 4.
[23]R. H. Jones and P. V. Tryon, J. Res. NBS **88**, 17 (1983).
[24]L Galleani and P Tavella, IEEE Control Syst. Mag. **30**, 44 (2010). See also the many references in this paper.
[25]S. R. Stein, in *Proceedings of the 24th Precise Time and Time Interval Planning and Applications Meeting*, 1992, pp. 289–302.; NASA Conference publication 3218, Goddard Space Flight Center, Greenbelt, Maryland 20771. See also U.S. patent 5,155,695 and 5,315,566.
[26]K. R. Brown, Jr., in *Proc. of the 4th International Technical Meeting of the Satellite Division of The Institute of Navigation*, 1991 (Institute of Navigation, Manassas, Virginia, 1991), pp. 223–242.
[27]M. A. Weiss, D. A. Allan, and T. K. Peppler, IEEE Control Syst. Mag. **38**, 631 (1989).
[28]J. Levine, IEEE Trans. Ultrason. Ferroelectr. Freq. Control **44**, 629 (1997).
[29]M. A. Weiss and T. P. Weissert, in *Proceedings of 7th European Time and Frequency Forum, Neuchatel, Switzerland*, 1993 (Swiss Foundation for Research, Neuchatel, Switzerland). Also available on the web from NIST publications data base at tf.nist.gov, paper 1022.
[30]Y. S. Shmaily, Metrologia **45**, 571 (2008); Y. S. Shmaily, IEEE Signal Process. Lett. **15**, 517 (2008).
[31]D. L. Mills, *Computer Network Time Synchronization: The Network Time Protocol* (CRC, New York, 2011).
[32]J. Levine, IEEE/ACM Trans. Netw. **3**, 42 (1995).
[33]V. Zhang, T. E. Parker, J. Achkar, A. Bauch, L. Lorini, D. Matsakis, D. Piester, and D. G. Rovera, in *Proceedings of 41st Annual Precise Time and Time Interval Planning and Applications Meeting*, 2009 (US Naval Observatory, Washington, DC), available online at www.pttimeeting.org. Also available on the web from the NIST publications data base at tf.nist.gov, paper 2432.
[34]J. Levine, IEEE Trans. UFFC **46**, 888 (1999).
[35]A. Gelb, op. cit., Eqs. (3.8)–(21), p. 82.