

AT2, A New Time Scale Algorithm : AT1 Plus Frequency Variance

M. Weiss and T. Weissert

Abstract. The existing AT1 algorithm produces a time scale with a fractional frequency variation smaller than that of any clock in the ensemble. We are developing a modification to AT1 that includes the additional desirable features : automatic frequency-step detection, the option to run in an optimal post-processing mode and to run with minimal supervision in non-technical environments. These properties are facilitated by the inclusion of a Kalman-filter estimate of the frequency variance of each clock in the scale. Results are reported from both simulated and real clock data to demonstrate automatic frequency-step detection.

1. Introduction

By sampling an ensemble of clocks, an ideal time scale algorithm would generate time and frequency with more reliability, stability, and frequency accuracy than any one of the individual clocks in the ensemble. In this paper we study an approach to this ideal.

A time scale algorithm calculates the time offset of each clock from ensemble time at a given reference time. Ensemble time—the time of the scale—is realized by applying the appropriate offset to the time of any one clock. If there is no measurement noise, then this value is independent of the clock used. At a given reference time, the algorithm requires the time differences between pairs of clocks (one clock is chosen as the common clock for all differences) and an estimate of the characterization parameters for each clock's systematic and stochastic frequency deviations.

Because the time of an individual clock cannot be measured, one must measure time differences between clocks. Thus the ensemble time is not directly observable and it is therefore inappropriate to use an accuracy algorithm—such as a Kalman filter—to generate time by minimizing time error.

However, we can, and do in the algorithms discussed here, optimize time and time-interval stability.

The time of a clock is a derived quantity which we infer from the clock's frequency. The true physical quantity produced by a clock is its frequency ; so all the parameters which characterize clock performance necessarily describe aspects of that frequency. One can use these parameters to optimize time uniformity and frequency accuracy. An algorithm that optimizes on time accuracy allows the clock with the best long-term stability to dominate the scale while sacrificing most of the short-term performance of the other clocks.

The AT1 time scale algorithm at the National Institute of Standards and Technology (NIST) has generated a scale since 1968. This scale has a number of useful properties that include : models of white, random-walk and flicker frequency modulation (FM), a fractional frequency variation which is generally smaller than that of any clock in the scale, adaptive estimates of clock weights, reliability beyond any individual clock, the ability both to easily add clocks to or remove them from the ensemble with minimal impact on the scale and to calibrate the ensemble against a primary reference. But there is still room for improvement. It would be desirable to obtain a time scale which simultaneously : detects frequency steps automatically, can be optimized for post-processing (including running both backwards and forwards in time) and can run

M. Weiss and T. Weissert : Time and Frequency Division,
National Institute of Standards and Technology,
325 Broadway, CO 80303, USA.

with minimal supervision for use in non-technical environments.

The new algorithm we report here combines aspects of the NIST AT1 algorithm with techniques from Kalman filtering. Both AT1 and the new algorithm include estimates of the time, frequency, and frequency drift for each clock in the ensemble. We model the physical frequency of a clock as having two stochastic parameters: white FM and either flicker or random-walk FM. Whereas the time estimate in our new algorithm is made the same way as in AT1, our method of estimating frequency is new. We treat the first difference of the algorithm's time offset for a given clock as if it were a "measurement" of that clock's frequency offset from the ensemble. Once we have measurements of individual clocks, as opposed to clock differences, it is appropriate to apply Kalman techniques to filter out the white FM. The result of this calculation is not the physical frequency of the clock but a mathematical estimate of the random-walk (or flicker) FM component of frequency plus drift, denoted by y , and an estimate of the variance of y . This frequency variance provides us with a quantitative confidence of the frequency estimate, which facilitates the attainment of all three desirable improvements.

Frequency-step detection requires an examination of the data over a multiple-measurement time period because: 1) we measure time differences directly, and not frequency, thus it takes at least two measurements to determine frequency; 2) the white FM of a clock can mask frequency steps, therefore one must average the signal until the white FM drops below the size of the frequency step. In our algorithm, we compare the estimate of the average frequency offset over an interval with the filtered estimate from the beginning of that interval. If this difference is larger than the confidence estimate of the frequency—obtained from the estimate of frequency variance—we conclude that a frequency step has occurred.

The estimate of frequency variance also allows us to smooth our estimates of y in a post-processing mode, where we may combine the variances derived from both a forward and a backward filter. The formalism for this process is also derived directly from Kalman filter theory. Essentially, we use the reciprocals of the forward and backward frequency variances as weights to combine their respective estimates at a given reference time. We do not incorporate the data from both filters simultaneously, because that would not provide independent estimates. Instead, at a given reference time, we use the extrapolated estimates of frequency and variance from one filter—the backward filter for example—and combine this estimate with its counterpart from the forward filter, which has been updated with the data.

Because our algorithm includes the Kalman formalism, a new clock with an initial set of long- and short-term performance characterization parameters can be inserted into the ensemble and the algorithm will adapt from an initial estimate of frequency variance, thereby learning the new frequency. This learning curve allows us to introduce new clocks into the scale without perturbing the scale's performance.

In this paper we sketch the AT1 algorithm and describe our modifications to it, which we call AT2, concentrating on those aspects which provide automatic frequency-step detection and exemplify its use with reference to both simulation and real clock data.

2. AT1 Formalism

The AT1 algorithm estimates the time and frequency offsets of each clock from ensemble time (and frequency). An estimate of frequency drift for each clock can be entered and used for time prediction, but it is not updated in the calculations. The weight of each clock is determined by its prediction confidence, the normalized reciprocal of the squared prediction error. AT1 produces a time scale with a three-step calculation for each clock: time update, prediction confidence update, and the frequency update. Each of these steps is further sub-divided into an initial prediction and then the update using the clock data.

2.1 Time prediction

The prediction of the time offset from ensemble time, \hat{x}_i , of clock i for the current measurement time, $t + \tau$, is calculated from the previously updated time offset at time t , x_i , the filtered frequency, y_i , and the constant frequency drift, D_i , according to

$$\hat{x}_i(t + \tau) = x_i(t) + \left(y_i(t) + \frac{D_i \tau}{2} \right) \tau. \quad (1)$$

2.2 Time update

Given measurements of time differences between each pair of clocks, $x_{ji}(t + \tau)$, the time offset of clock i is updated in a weighted average, using the prediction, weight and time-difference measurement of every clock, j , that is,

$$x_i(t + \tau) = \sum_{j=1}^n w_j [\hat{x}_j(t + \tau) - x_{ji}(t + \tau)]. \quad (2)$$

The weights, w_j , are determined from the prediction confidence.

2.3 Adaptive clock weights

The clock weights used to update the time offsets are calculated from the variances of the time residuals, $\varepsilon_i^2(\tau)$, by

$$w_i = \frac{\varepsilon_x^2(\tau)}{\varepsilon_i^2(\tau)}, \quad (3)$$

where $\varepsilon_x^2(\tau)$ is the ensemble prediction error. In Appendix A we show that when the weights are calculated in this way, the resulting time updates optimize the ensemble time stability in a minimum variance sense. These weights adaptively match the relative stability of the clocks in the ensemble because $\varepsilon_i^2(\tau)$ and $\varepsilon_x^2(\tau)$ are estimated from the data.

2.4 Ensemble prediction error

The ensemble prediction error provides a numerical estimate of the stability of the ensemble over an integration time τ . It is calculated as

$$\varepsilon_x^2(\tau) = \left(\sum_{i=1}^n \frac{1}{\varepsilon_i^2(\tau)} \right)^{-1}. \quad (4)$$

From this equation we see that a poorly performing clock will not destabilize the ensemble; clocks can only improve ensemble stability.

2.5 Prediction error estimate

An estimate of the prediction error of clock i over the interval τ is calculated from the difference between the time prediction and subsequent update, plus an estimate of the average bias of this value from an absolute prediction error, according to

$$\hat{\varepsilon}_i(\tau) = |\hat{x}_i(t + \tau) - x_i(t + \tau)| + K_i, \quad (5)$$

where K_i is the expected value of the bias of clock i . Thus $\hat{\varepsilon}_i$ is an estimate of the deviation (square root of variance) of the clock based on the current measurement cycle.

2.6 Bias of the error estimate

Because ensemble time is a weighted average of each of the individual clock times, the prediction error of a clock measured against the scale will be biased low, on the average. To correct for this biasing, we include an estimate of the bias for each clock, which is:

$$K_i = \frac{2\varepsilon_x^2}{\sqrt{2\pi\varepsilon_i^2}}. \quad (6)$$

Thus the bias correction, K_i , is the difference between the average prediction error relative to the scale and the absolute average prediction error. In Appendix B we show the derivation of this factor which assumes a normal distribution of clock noise.

2.7 Prediction error update

Because of its stochastic nature, the squared prediction error of each clock needs to be averaged over some period of time. On the other hand, since the noise characteristics of a clock may not be stationary, past updates are de-weighted in an exponential filter as follows:

$$\varepsilon_i(\tau) \Big|_{t+\tau} = \frac{\varepsilon_i^2 + N_\tau \varepsilon_i^2(\tau)}{1 + N_\tau}. \quad (7)$$

The time constant for the filter is typically chosen to be $N_\tau = 20$ days for cesium clocks, representing the time one expects the white FM level to be constant. The initial value of $\varepsilon^2(\tau)$ is estimated as $\tau^2\sigma_y^2(\tau)$.

2.8 Frequency estimate

The average frequency of each clock at time $t + \tau$ over the interval τ , based on the latest two time updates, is used as an estimate of that clock's current frequency, \hat{y}_i . The calculation is

$$\hat{y}_i(t + \tau) = \frac{x_i(t + \tau) - x_i(t)}{\tau}. \quad (8)$$

2.9 Frequency update

We incorporate the current estimate of frequency and the previous frequency update into an exponentially-filtered update of the current average frequency offset of each clock, according to

$$y_i(t + \tau) = \frac{\hat{y}_i(t + \tau) + m_i y_i(t)}{1 + m_i}, \quad (9)$$

where the time constant for this filter, m_i , is calculated from the noise parameters for each clock.

2.10 Exponential time constant for the frequency update

The exponential time constant used to filter the frequency updates is determined from the relative levels of white FM and random-walk (or flicker) FM for each clock by

$$m_i = \frac{1}{2} \left(-1 + \sqrt{\frac{1}{3} + \frac{4\tau_{\text{Min},i}^2}{3\tau_0^2}} \right). \quad (10)$$

Here, $\tau_{\text{Min},i}$ is the integration time which gives the minimum value on a standard $\sigma_y(\tau)$ plot, given that the clock's stochastic deviations are characterized by white and random-walk FM and τ_0 is the minimum τ value used for computing $\sigma_y(\tau)$. This value of m_i can be shown to optimize the stability in predicting time, given these two kinds of noise in the clock (white and random-walk FM). If white

FM and flicker FM are more suitable models, then m_i can be approximated as τ_x/τ_0 , where τ_x is the intercept value of τ on a $\sigma_y(\tau)$ plot for the white and flicker FM.

3. AT2: AT1 Plus Frequency Variance

In addition to what AT1 provides, we would like to have an estimate of the variance of the residuals of the frequency offset from the ensemble. In our approach, we interpret x_i as a measurement of the time of clock i against the scale. Thus the first difference, \hat{y}_i of x_i , as in (8), is a measurement of the frequency offset of clock i from the scale. In this way, the frequency of a given clock is now measurable relative to the ensemble frequency; but still not measurable in an absolute sense. Once we have a measurement of an individual clock as opposed to only clock difference measurements, it is appropriate to apply a Kalman filter to reduce the white FM to obtain an update y .

In AT2, we model the frequency noise of a clock with two stochastic parameters: white FM, σ_α , and random-walk FM, σ_β . In estimating the parameter y , the white FM impulses, α , become measurement noise. Thus the *measurement* model is

$$\hat{y} = y + \alpha. \quad (11)$$

There is a second white-noise process σ_β — with impulse β — that drives the random-walk (integrated white noise) FM in y ; so the *system* model is

$$y(t + \tau) = y(t) + D \cdot \tau + \beta. \quad (12)$$

The parameter y is the random-walk component of the clock's frequency plus any drift, $D \cdot \tau$, and not the physical frequency produced by the clock. In contrast, the terms x_i represent the physical time offsets of each individual clock from the scale and necessarily incorporate the white FM; therefore, the first difference of x is an estimate of the physical frequency of the clock. The frequency estimate y can be used to predict the time better than the instantaneous physical frequency.

The equations implemented for the updates are as follows. We predict the frequency variance according to

$$\hat{P}(t + \tau) = P(t) + \sigma_\beta^2 \tau, \quad (13)$$

where \hat{P} is the prediction of the variance of the residuals of y that grow with τ according to σ_β^2 . We then update frequency by calculating

$$y(t + \tau) = \frac{\sigma_\alpha^2(t + \tau) \cdot y(t) + \hat{P}(t + \tau) \cdot \hat{y}(t + \tau)}{\sigma_\alpha^2(t + \tau) + \hat{P}(t + \tau)}. \quad (14)$$

Here the Kalman formalism provides us with an exponential filter on y as in AT1. In steady state AT2 does reduce to AT1, if the weights are chosen properly; thus it inherits the ability of AT1 to model flicker FM. The frequency variance update is

$$P(t + \tau) = \left(\frac{\sigma_\alpha^2 \cdot \hat{P}}{\sigma_\alpha^2 + \hat{P}} \right)_{t+\tau}. \quad (15)$$

We use the AT1 estimate of prediction error, $\varepsilon(\tau)$, as our estimate of σ_α , the white FM level. This is valid if the measurement interval, τ , is well within the range of integration times for which the white FM is dominant, as is the case with the system at NIST, which makes measurements every two hours.

Because we estimate the "measurement noise", σ_α^2 , our system constitutes an adaptive Kalman filter, which allows P , the variance of the residuals of y , to evolve with changes in σ_α^2 , from an initial value. Thus the integration time of the exponential filter on y as expressed in (14) changes with time, as compared with the parameter m_i as in (10) for AT1. This occurs when initializing a new clock and when a clock's white FM level changes.

Setting τ to unity and solving (13) and (15) for the steady-state values of P , we have

$$P = \sigma_\beta \left(\sqrt{\frac{\sigma_\beta^2}{4} + \sigma_\alpha^2} - \frac{\sigma_\beta}{2} \right). \quad (16)$$

Making the appropriate identifications between the frequency updates (9) and (14), we have

$$m = \frac{\sigma_\alpha^2}{P + \sigma_\beta^2}. \quad (17)$$

Applying (16) to this result, we obtain

$$m(m + 1) = \frac{\sigma_\alpha^2}{\sigma_\beta^2}. \quad (18)$$

These equations allow us to compare the performance of the two algorithms.

We close this section with a brief discussion of this algorithm in relation to three others. Jones and Tryon [1] have designed a time scale algorithm strictly employing a Kalman filter. The scale produced from this algorithm, called TA (NIST), has been generated at NIST in parallel with AT1 since about 1983. Their algorithm is mathematically identical with the AT1 algorithm for the time and frequency predictions and updates [2], but differs in the calculation of the clock weights for the time update and the choice of exponential filter parameters for the frequency update. These differences affect the ensemble time generated by TA (NIST).

Previous simulation has shown that the pure Kalman filter time scale sacrifices short-term

performance and simply follows the clock with the best long-term performance. This is consistent with the design of Kalman filters in general, which minimize error in estimating the state vector. Because the state vector of the Jones-Tryon filter has elements of both time and frequency, the filter minimizes both time and frequency error. This can sacrifice short-term stability. This failure is also manifest in that elements of the covariance matrix grow without bound. In practice, with a good ensemble of clocks, this growth is not large enough to cause computer overflow errors in practical applications, but it does suggest a possible opening for a failure.

The ALGOS algorithm is used at the Bureau International des Poids et Mesures (BIPM) for the generation of International Atomic Time (TAI) [3]. With it, measurements are taken every ten days and combined to optimize the long-term stability. A fundamental difference between ALGOS and AT2 is that, since ten days of integration is beyond the range of white FM of most clocks [4], short-term stability is not an issue in the ALGOS algorithm. ALGOS would be less useful at NIST because data is taken every two hours. Two other features of ALGOS are that it is a deferred time scale—so it is strictly a post-processing algorithm—and it detects abnormal behavior. Although AT2 is a real-time time scale algorithm, one of our aims is to develop a post-processing option also.

Another time scale, developed by Stein, is in use at the Naval Research Laboratory (NRL) in Washington D.C., and at CSOC in Colorado Springs, Colorado [5]. This scale was developed to run well without the scrutiny of technical staff. The algorithm is based entirely on the Kalman formalism, thus it optimally sets clock parameters when entering new clocks or starting the scale by minimizing least squares error. This algorithm has been designed to avoid the problem we discuss above with the Kalman filter's minimizing time and frequency error, thus losing short-term stability. Yet, because the Kalman filter is a least squares error algorithm, it is not clear whether this algorithm can optimize stability the way AT2 is explicitly designed to do.

4. Frequency-Step Detection

Frequency steps can be detected only some time after they occur. Since our measurements are of phase differences between clocks with inherent white FM, frequency steps appear only after the accumulation of measurements. In the presence of flicker or random-walk FM no mean frequency value may be assigned, so small frequency steps blur into the stochastic noise. We define a frequency step as a

shift in frequency greater than four times the standard deviation of the stochastic impulses. By contrast, a time step is a shift in phase of a clock that exceeds three times the current white FM characterization.

The detection of frequency steps may be obscured by the presence of time steps and the stochastic noise from white and random-walk FM. If a clock generates a time step, and that reading is used to compute an average frequency since the last measurement, then the time step can be indistinguishable from a frequency step. Only after two measurements can one conclude that the new frequency after the time step is approximately equal to that before it and so distinguish between time and frequency steps.

Whereas the possibility of time steps requires a wait of at least two measurements before checking for a frequency step, it may also be useful to look backwards beyond two measurements for the determination of small frequency steps. This procedure allows the white noise to be averaged down to the point where only random-walk FM remains. The maximum useful extent of a search backwards for frequency steps is τ_{\min} for a clock with white FM and random-walk FM, or τ_x , for a clock with white and flicker FM. In terms of a $\sigma_y(\tau)$ (Allan variance) plot, τ_{\min} is the integration time at which the white FM dominated curve gives way to the random-walk FM region, and τ_x is the integration time at which the white FM dominated curve gives way to the flicker FM region (these transitions in slope occur at the familiar "knee" of the Allan variance plot). Because we must measure clock differences, our best estimate of an individual clock frequency will be for a clock against the ensemble. But the ensemble itself must have stochastic processes, and these also need to be considered when testing for frequency steps.

The new algorithm tests for frequency steps by iterating backward for each clock a range of measurements from two before the current measurement time up to a time τ_{\min} back. Thus the search is limited to some maximum number of consecutive discrete measurement times, L_{\max} , an integer determined from τ_{\min} . Since our estimate of the white FM level, σ_x , is adaptive, τ_{\min} is always changing. We slow these changes by exponentially filtering σ_x , which is used to compute τ_{\min} , with a longer time constant than that used for the clock weights. On each measurement cycle, for each clock, we maintain a buffer of the previous L_{\max} values of all the necessary parameters for frequency-step detection. At each measurement, we search for frequency steps for each clock. We define the average frequency of a clock over the time interval from the measurement L intervals back ($L < L_{\max}$), x_{-L}

at time t_{-L} , to the most recent measurement before the current time, x_{-1} at time t_{-1} , that is,

$$y_{\text{avg}} = \frac{(x_{-1} - x_{-L})}{(t_{-1} - t_{-L})}. \quad (19)$$

We compare this frequency with the updated estimate of y , y_{-L} , saved from time t_{-L} , that is, we compare a filtered estimate of frequency from the beginning of a fixed time interval with an estimate of the average frequency over that interval. If the frequency difference exceeds a test value, we conclude that the clock generated a frequency step. The test is whether

$$|y_{\text{avg}} - y_{-L}| > 4 \sqrt{\frac{\tau_{\text{min}}}{L} \left(\frac{P_{-L} + P_{-2}}{2} + \sigma_{\alpha x}^2 \right) + \sigma_{\beta x}^2 \cdot L}, \quad (20)$$

where $\sigma_{\alpha x}^2$ and $\sigma_{\beta x}^2$ are the white and random-walk FM of the ensemble determined from

$$\sigma_{\alpha x}^2 = \left(\sum_{i=1}^n \sigma_{\alpha i}^{-2} \right)^{-1} \quad (21)$$

and

$$\sigma_{\beta x}^2 = \left(\sum_{i=1}^n \sigma_{\beta i}^{-2} \right)^{-1}. \quad (22)$$

In this way we incorporate the stochastic impulses both from the individual clocks and from the scale itself. If a clock generates a frequency step, it is possible for that same step to appear over a contiguous range of test times. We look for frequency steps over all times in the allowable range and treat that having the largest inequality in (20) as containing the actual frequency step.

When a frequency step is detected, we increase the variance, P , at the time of the step, assign the stepped clock zero weight in the scale and then re-run the scale from that time up to the most recent measurement cycle. We do not search for new frequency steps during this secondary run; thus there is no iterative search for frequency steps. The stepped clock remains deweighted until a time τ_{min} after the occurrence of the step. When the clock is re-entered into the scale, σ_{α} —which has evolved adaptively while the clock was deweighted—is increased by a fixed amount. In other words, when we find a frequency step, the clock is removed from the scale until the algorithm adaptively learns the clock's new frequency. The clock is then re-entered with an initially large doubt about its new frequency.

In a real-time algorithm, re-running the scale cannot change the values already measured against the scale. We cannot change history; but we can decide that our current estimates of clock frequency are in error, due to a frequency step in the past,

and apply steering. An alternative theoretical approach would be to use a delay larger than the maximum allowable τ_{min} for the frequency of at least one clock and to use the output of that delay to steer from, to define the ensemble time [6]. Corrections from any frequency steps could be included in this steering. Such a delay would have to be of the order of a week to be of use for current commercial cesium clocks. Practically, such a delay would have to be created digitally, and the clock driving the digitization would add further noise and unreliability to the system.

5. Simulation

We simulated an ensemble of clocks by generating ten independent sets of time series data at one point per day for seven hundred days. These clocks were generated with different values of white and random-walk FM, designed to span the noise range of typical cesium clocks. To achieve the simulation properly, we used only phase difference data between these clocks. Our aim was to establish the performance limits of the new algorithm, that is, whether:

- (1) both algorithms AT1 and AT2 produce a time scale which performs apparently better than the best clock in the scale at all integration times,
- (2) the TA (NIST) algorithm is dominated by the clock with the best long-term performance at all integration times,
- (3) the AT2 generated estimate of the confidence on the frequency offset appears reasonable and
- (4) the use of this confidence estimate to determine frequency steps improves the long-term performance of the time scale.

Figure 1 illustrates item (1). We have computed the stability of each clock using an N -cornered hat technique [7]. We determine the stability of the scale by taking the output value of clock-minus-scale and subtracting the generated value of clock-minus-truth. There are significant differences between the variances computed directly, and those estimated from the N -cornered hat, Figure 2. We conclude that these differences result from apparent correlations in the data. This correlation could come from either the finite data length, or from real correlations in the pseudo-random number generator. If the generated clocks are truly correlated, then the algorithm can only produce a variance better than the uncorrelated part. We notice that the scale seems to follow the shape of the variances from the N -corner hat. This suggests some correlation in the generated data.

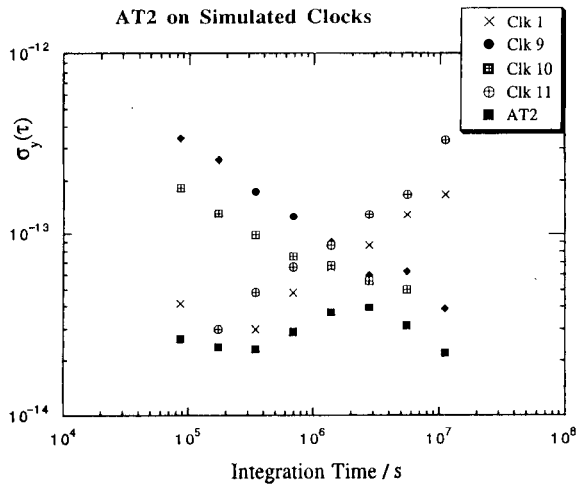


Figure 1. The stability of each simulated clock is computed using an *N*-cornered hat technique, whereas the stability of the scale is determined by taking the output value of clock-minus-scale and subtracting the generated value of clock-minus-truth. The scale produced with AT2 outperforms all clocks at all integration times.

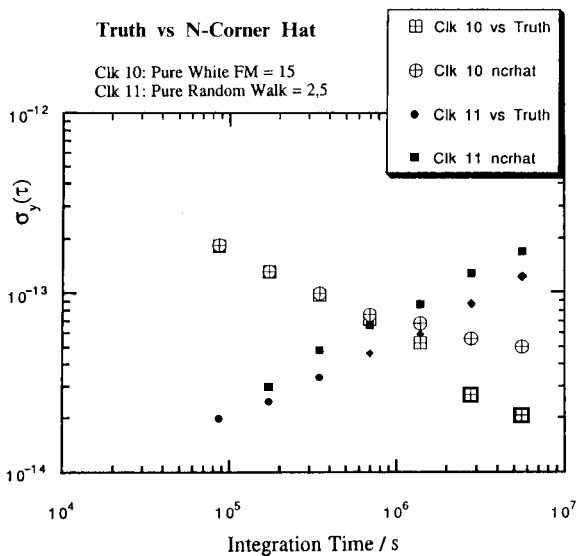


Figure 2. The significant differences between the variances computed directly and those estimated from *N*-cornered hat must be due to apparent correlations in the data.

Figure 3 shows a comparison of the output of the TA (NIST) Kalman filter with the simulated input data. As expected, the Kalman algorithm has the stability of the clock with the best long-term variations.

Figures 4 and 5 show the residuals from AT2 compared with the confidence from the variance of frequency residuals. The algorithm calculates an estimate of the random-walk component of a clock's frequency offset from the ensemble. Because these

are generated clocks, we know the true value of the random-walk component of frequency of that clock versus the true value of the time scale. The differences between these two, the estimate minus truth, are the residuals plotted. The sigma value used in the plot is the RMS of the estimated deviation of the clock plus the estimated deviation of the scale. The line plotted is the three-sigma value. This should be a 99,8 percentile. Over the 700 points plotted we should get one or two residuals crossing the lines of the sigmas. This seems to be the case.

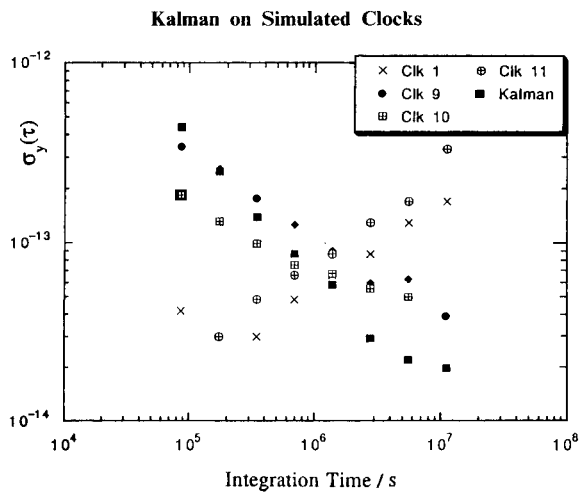


Figure 3. Clearly, in this comparison of the output of a version of the Kalman filter which defines TA (NIST) with the simulated input data, one can see that the Kalman algorithm has the stability of the clock with the best long-term variations.

Simulation Clk 1: Residuals vs. Estimated Confidence
White FM = 1 ns, Random Walk FM = 15 ns, both @ 1 day

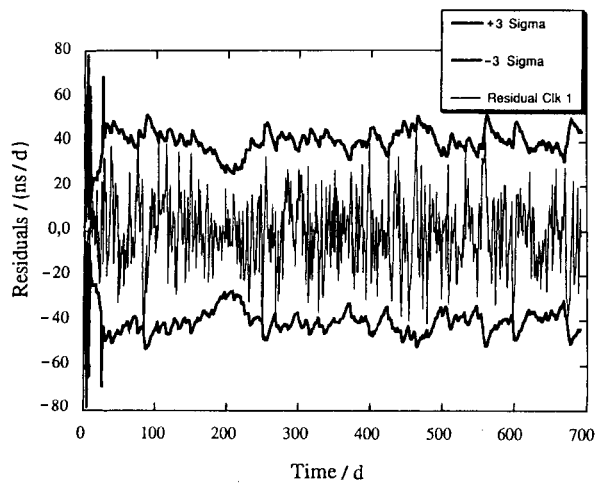


Figure 4. The frequency residuals of simulated clock No. 1 from the AT2 algorithm are compared with the three-sigma value of estimated confidence, as derived from the estimated variance of frequency residuals.

Simulation Clk 9: Residuals vs. Estimated Confidence

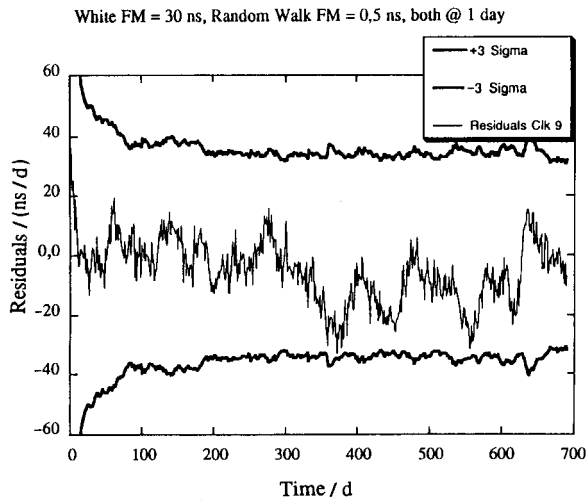


Figure 5. The frequency residuals of simulated clock No. 9 from the AT2 algorithm are compared with the three-sigma value of estimated confidence, as derived from the estimated variance of frequency residuals.

Clock 9: White FM = 30 ns, Random Walk = 0,5 ns, both @ 1 day

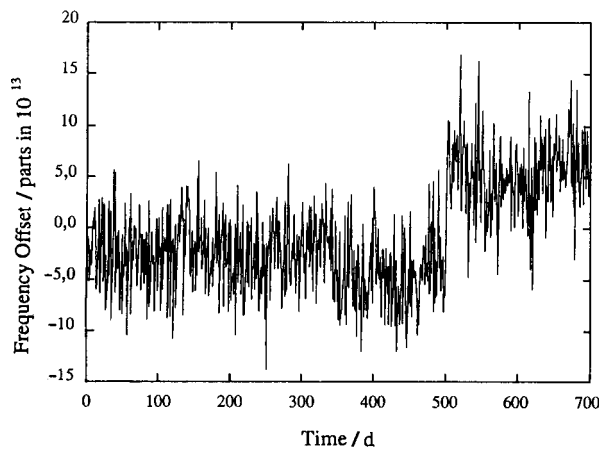


Figure 6. The frequency offset from the scale of simulated clock No. 9 clearly shows the generated frequency step of 1×10^{-12} on day 500.

Clock 9: White FM = 30 ns, Random Walk = 0,5 ns, both @ 1 day

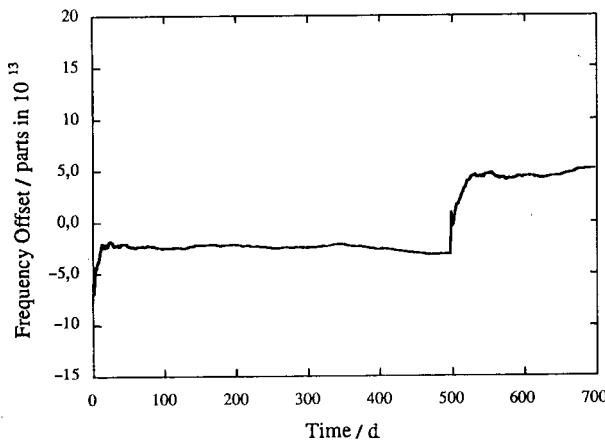


Figure 7. This plot of the AT2 estimate of the random-walk component of frequency for clock No. 9 shows that the algorithm detected the frequency step, with the step detector set at 4 sigma.

Clock 1: White FM = 1 ns, Random Walk FM = 15 ns, both @ 1 day

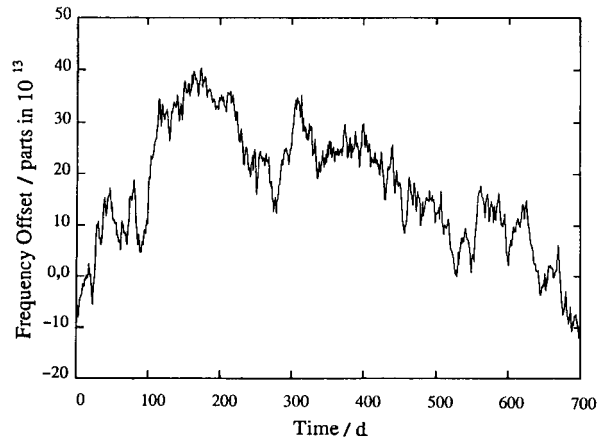


Figure 8. Although the scale shows very little smoothing, the AT2 algorithm automatically detected a frequency step of 2×10^{-12} on day 100 in simulated clock No. 1 and removed the clock from the scale.

Frequency-Step Detection Improves Long-Term Stability Simulated Clocks

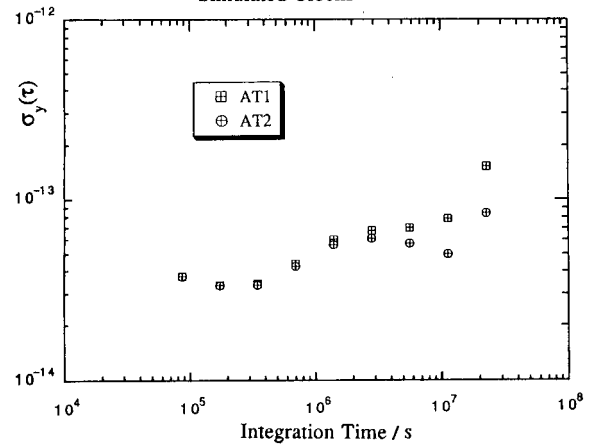


Figure 9. The benefit from having detected the frequency steps is demonstrated by the significant improvement in scale stability at an integration time of 128 d and longer.

Comparison of Ensembles

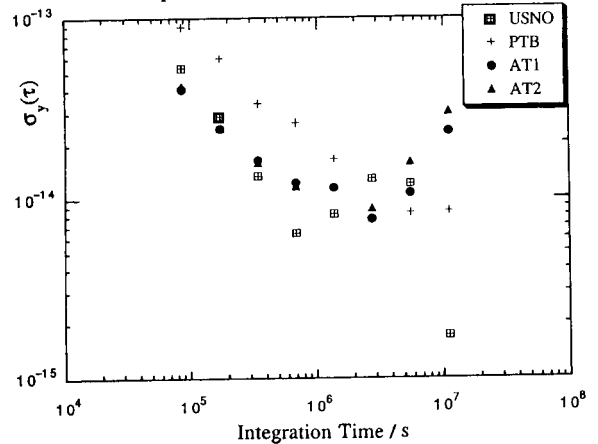


Figure 10. The results are shown for the two scales, the official AT1 and the new AT2, run on NIST clocks as well as USNO and PTB.

Lastly, we inserted frequency steps into the simulated clock data. Figure 6 shows the frequency offset from the scale of simulated clock No. 9, with a frequency step of 1×10^{-12} on day 500. This clock was given a white FM level of 30 ns, and a random-walk FM level of 0,5 ns, both at 1 d. Figure 7 shows estimates from the AT2 scale of the random-walk component of frequency. The reduction in the white FM level is apparent. The algorithm successfully located the frequency step, with the step detector set at 4 sigma, with sigma defined as above. When such a step is detected, the scale is recalculated with the stepped clock removed until the scale can "learn" the new frequency value. Figure 8 shows the frequency offset from the scale of simulated clock No. 1, with a frequency step of 2×10^{-12} on day 100. The noise of this clock is predominantly random-walk FM. The estimate from the scale shows very little smoothing. Yet, even in this case, the frequency-step detector automatically found the step and removed the clock from the scale. Figure 9 shows the benefit of frequency-step detection. There is a significant improvement at an integration time of 128 d and longer.

6. Real Data

We ran AT2 on data from real clocks at NIST over the period from December 31, 1988, to October 30, 1989. AT2 found 11 frequency steps among the 13 clocks in the ensemble over the 400-day run. The steps ranged from 3,1 to 4,2 parts in 10^{13} . We compared the scale thus created with other laboratories : PTB, USNO, TUG, and NRC, using GPS common-view measurements [8]. The GPS data is the UTC of each lab, which usually is steered to the international UTC. However, during this period no steering was applied so the scales should be independent. Using an *N*-cornered hat technique, we were able to determine the variance of each UTC scale. We compared the results with a similar analysis using the official NIST AT1 time scale. The results for the two scales run on NIST clocks are plotted with those from USNO and PTB in Figure 10. This shows that the scales of AT1 and the new AT2 are similar, but in the long term the AT1 scale is somewhat better. The AT1 scale is watched carefully and clocks are administratively checked for time and frequency steps, and for changes in general performance. Clearly human care is labor intensive, but it adds much to the performance of a time scale.

7. Conclusions

The AT2 algorithm employs the clock-against-ensemble time estimate from the AT1 algorithm as

a pseudo-measurement. The first difference of these measurements provides a pseudo-measurement of clock frequency against the scale which we use as input to a Kalman filter to estimate the random-walk component of the frequency state in the presence of white FM. In this way we avoid the usual Kalman problem with the unobservability of the ensemble frequency. Although our algorithm works similarly to AT1 when there are no frequency steps, we have demonstrated the ability to detect frequency steps automatically, and to adaptively adjust the exponential time constant for frequency averaging. Like the AT1 algorithm, the AT2 algorithm produces an ensemble time whose variance is smaller than the best clocks in the scale at all integration times. The frequency variance provides an estimate of frequency confidence which seems reasonable compared to the residuals. The quantization of this confidence enables frequency-step detection and therefore improves long-term stability in the simulated data. Because the official AT1 scale, using real clocks, is closely monitored in real time, it out-performs the new scale, as expected. The advantages of the new scale are that it automatically detects and adjusts for frequency steps, and adaptively changes the integration time of a clock's frequency as its white FM level changes.

Appendix A : The Weights in (3) Minimize the Variance of X_j

Let us define the estimate of clock *j* against the scale via the measurement using clock *i* as

$$x_{j(i)} = \hat{x}_i(t + \tau) - x_{ij}(t + \tau). \tag{A1}$$

We want to choose weights w_i to combine these different estimates into an estimate of clock *j* using the entire ensemble of clocks, that is,

$$x_j = \sum_{i=1}^n w_i x_{j(i)}, \tag{A2}$$

with the constraint that the weights are normalized, as in

$$\sum_{i=1}^n w_i = 1. \tag{A3}$$

Under the assumption of independence of the $x_{j(i)}$, if $\sigma_{j(i)}^2$ is the variance of $x_{j(i)}$, then the variance of X_j satisfies

$$\sigma_j^2 = \sum_{i=1}^n w_i \cdot \sigma_{j(i)}^2. \tag{A4}$$

We want to choose the weights, w_i , to minimize σ_j^2 for all *j*. We first fix a particular σ_j^2 . We may take the partial derivative of σ_j^2 with respect to w_i , set it equal to zero and solve to find the extremum. To aid in this we may add a constant to the sum

as the method of Lagrangian undetermined multipliers. So we minimize the quantity

$$S_j^2 = \sum_{i=1}^n w_i \cdot \sigma_{j(i)}^2 + k \sum_{i=1}^n w_i. \quad (\text{A5})$$

Taking the partial derivative of this quantity and solving for the w_i we find

$$w_i = \frac{N}{\sigma_{j(i)}^2}, \quad (\text{A6})$$

where

$$N = \sum_{i=1}^n \frac{1}{\sigma_{j(i)}^2}. \quad (\text{A7})$$

In this equation $\sigma_{j(i)}^2$ is the variance of the predictability of clock i as seen by clock j . Since we want the weights to be independent of j , we choose the weights to be inversely proportional to the variance of the predictability of clock i against the scale. This gives us the weights in (3) as defined by the variances from equations (4)-(7) with the considerations in Appendix B.

Appendix B: K_i as Defined in (6) is the Expected Value of the Bias of Clock i in (5)

For notational convenience we suppress the subscript i in what follows. This should lead to no confusion since the argument holds for any given clock. We also suppress the notation τ for the fixed time interval. Thus instead of $\varepsilon_i(\tau)$ we simply use ε .

The ensemble time is the weighted average of all the clocks in the scale. Thus, the average prediction error of a given clock relative to the ensemble time will be biased smaller than its absolute average prediction error. If the clocks are independent, the bias of a particular clock will be proportional to its total prediction error according to its weight. Thus, if we assume clock independence,

$$K = w \cdot \delta, \quad (\text{B1})$$

where δ is the average prediction error of this clock against a clock running at the theoretically perfect rate, w is the weight of this clock, as in (3), and K is the average bias of this clock's linear difference from the scale due to the fact that the scale is defined in part by this clock.

We compute δ as follows. We first assume that δ , the prediction error over a given interval, has a Gaussian distribution with mean zero and with standard deviation ε , ε as in (7).

$$p(\lambda) = \frac{1}{\sqrt{2\pi} \varepsilon} \exp\left(-\frac{\lambda^2}{2\varepsilon^2}\right). \quad (\text{B2})$$

The assumption of zero mean in the distribution is functionally correct for our purposes, but may not be true in general. We want to estimate the average linear offset of the prediction error from the mean.

The mean may not be zero since the scale will walk off in rate (and hence in time as well) from a theoretical perfect clock. Since we are concerned only with deviation from the mean, we may assume the mean is zero without loss of generality. Then

$$\begin{aligned} \bar{\delta} &= \frac{1}{\sqrt{2\pi} \varepsilon} \int_{-\infty}^{\infty} |\lambda| \exp\left(-\frac{\lambda^2}{2\varepsilon^2}\right) d\lambda \\ &= \frac{2}{\sqrt{2\pi} \varepsilon} \int_0^{\infty} \lambda \exp\left(-\frac{\lambda^2}{2\varepsilon^2}\right) d\lambda \\ &= \frac{2\varepsilon}{\sqrt{2\pi}}. \end{aligned} \quad (\text{B3})$$

Substituting this result into (B1), along with the expression for weight, w , from (3), we have

$$K = \frac{2\varepsilon_x^2}{\sqrt{2\pi}\varepsilon^3}, \quad (\text{B4})$$

where we have again suppressed the subscript i in K and in the denominator, for clock i , and the notation (τ) indicating that ε is the prediction error over the fixed interval τ . Note that ε_x is the prediction error for the ensemble as in (4).

References

1. Jones R. H., Tryon P. V., Continuous Time Series Models for Unequally Spaced Data Applied to Modeling Atomic Clocks, *SIAM J. Sci. Stat. Comput.*, 1987, **8**, 71-81.
2. Weiss M. A., Allan D. W., Peppler T. K., A Study of the NBS Time Scale Algorithm, *IEEE Trans. Instrum. Meas.*, 1989, **38**, 631-635.
3. Guinot B., Thomas C., Establishment of International Atomic Time, BIPM Annual Report of the Time Section, 1988, **1**, D1-D22.
4. Allan D. W., Time and Frequency (Time-Domain) Characterization, Estimation, and Prediction of Precision Clocks and Oscillators, IEEE UFFC, 1987, **34**, 647-654.
5. Stein S. R., Gifford G. A., Breakiron L. A., Report on the Timescale Algorithm Test Bed at USNO, Proc. 21st Annual Precise Time and Time Interval Applications and Planning Meeting, Redondo Beach CA, Nov. 28-30, 1989, 269-288.
6. Schnitger W., Personal communication: Dr. W. Schnitger, Midwaycity, CA 92655.
7. Barnes J., Time Scale Algorithms Using Kalman Filters—Insights from Simulation, Proc. 2nd Symposium on Atomic Time Scale Algorithms, NBS, Boulder, CO, June 23-25, 1982.
8. Allan D. W., Weiss M. A., Accurate Time and Frequency Transfer during Common View of a GPS Satellite, Proc. 34th Ann. Freq. Control Symp., Ft Monmouth, NJ, 1980, 334-346.

Received on 24 April 1990 and in revised form on 24 September 1990.