A NEW TIME SCALE ALGORITHM: AT1 PLUS FREQUENCY VARIANCE

M. Weiss, T. Weissert
NIST, 576
325 Broadway, CO 80303

Abstract

The algorithm which generates the AT1 time scale at the National Institute of Standards and Technology (NIST) has generated a scale with many desirable properties since 1968. Five of these are as follows:

1. **The fractional frequency variation of the scale is smaller than any clock in the scale for all integration times.**
2. **The algorithm adaptively estimates the weights of clocks in real time.**
3. **The scale is much more reliable than any individual clock.**
4. **One can add or remove clocks from the scale easily, with a minimum impact on the scale.**
5. **One can correct the ensemble for calibrations against a primary reference.**

There are three other properties we would like to obtain:

1. **Automatic frequency step detection.**
2. **A scale optimized for post–processing, including running both forwards and backwards in time.**
3. **A scale that can run with minimal supervision for use in non–technical environments.**

It turns out that simply estimating a variance of the frequency state of the clocks facilitates all three of these new properties. We report here a new algorithm which uses techniques from Kalman filtering to estimate this variance. Results from simulation and applications to real clocks are presented also.

INTRODUCTION

Ideally, a time scale algorithm samples an ensemble of clocks to generate time and frequency with more reliability, stability, and frequency accuracy than any 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 of the ensemble clocks at a given reference time. Ensemble time, the time of the scale, is realized by applying the appropriate correction to any one clock. If there is no measurement noise this value is independent of which clock is used. The input to the algorithm at a given reference time is the time difference between each clock and a particular clock. The algorithm also requires estimates of the deterministic and stochastic parameters which characterize each clock's frequency offsets.

It is important to notice that the time of a clock is not measured. One measures only time differences between clocks. Thus the ensemble time which the algorithm generates is not observable. For this reason it is inappropriate to use an accuracy algorithm, such as a Kalman filter, to generate time by minimizing time error. We can, and do in the algorithms discussed here, optimize time and time interval stability.

It is also important to realize that a clock as a physical system produces a frequency. The time of a clock is artificially derived from the frequency, which is the true physical quantity. Because of this, all the parameters which characterize clock performance describe aspects of the frequency. One can use these parameters to optimize time uniformity and frequency accuracy. An algorithm that optimizes time accuracy should simply allow the clock with the best long term stability to dominate the scale, thus sacrificing much of the performance of other clocks, especially in short term. We will look at these things further as we go along.

The AT1 time scale algorithm at the National Institute of Standards and Technology (NIST) has generated a scale with many desirable properties since 1967. Five of these are as follows:

1. The fractional frequency variation of the scale generally appears smaller than any clock in the scale for all integration times.
2. The algorithm adaptively estimates the weights of clocks in real time.
3. The scale is much more reliable than any individual clock.
4. We can add or remove clocks from the scale easily, with a minimum impact on the scale.
5. We can correct the ensemble for calibrations against a primary reference.

There are three other properties we would like to obtain:

1. Automatic frequency step detection.
2. A scale optimized for post–processing, including running both forwards and backwards in time.
3. A scale that can run 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 to estimate clock states representing the random walk plus drift of the frequency offsets of each clock, as well as the variance of these states. For a given clock, this state, which we call "Y", is not a physical state, but a mathematical estimate of the frequency offset in the presence of white noise modulating the frequency. The variance of this state gives us a confidence of this estimate. Having this estimate facilitates attainment of all three of our goals.

Frequency step detection always requires examining the data over some time period. In nearly real time operation it is possible to compare the estimate of average frequency offset over an interval with the filtered estimate from the beginning of that interval. Using the estimate of frequency deviation as a test for outliers, we can determine if a frequency step occurred in the recent past.

We can also smooth our estimates of "Y" in post-processing by combining the forward and backward filters. The proper way to combine these is well-defined in Kalman filter theory. Essentially, we use the reciprocal of the forward and backward filter variances as weights to combine their respective state estimates at a given reference time. We must be careful not to incorporate the data at that time in both filters before combining them, else they will not be independent estimates. Thus, at a given reference time, we use the extrapolated estimates of state and variance from one direction, the backward filter for example, and combines this state estimate with the one from the forward direction which has been updated with the data.

The AT1 time scale is adaptive once we know the clock parameters characterizing the short and long term behavior of the clocks. Since our new scale uses the Kalman formalism, we can enter a new clock and the algorithm will optimally adapt its estimate of frequency offset variance. This allows us to enter new clocks without perturbing the scale.

Both results from simulation and applications to real clocks are presented.

THEORY: AT1

We first present the AT1 algorithm and then show how we have modified it. In the AT1 algorithm each clock has two states which are estimated: the time and frequency offsets of the clock from ensemble time. Frequency drift can be entered and used, but it is not estimated adaptively by the algorithm. The AT1 algorithm is a three-tiered process: the time update, the update of the variance of the time offset, and the frequency update. The weight of a clock is proportional to the reciprocal of the variance. Each of these updates can be broken down into two steps: an initial estimate and the update.

1. Time Prediction:

$$\hat{X}_i(t+\tau) = X_i(t) + (Y_i(t) + D_i\tau/2)\tau \tag{1}$$

We predict the time offset from ensemble time, $\hat{X}_i$, of clock i for the current measurement time $(t+\tau)$ based on the previous estimates at time t of time offset, X_i, and filtered frequency, Y_i, and the entered frequency drift, D_i.

2. Time Update:

$$X_j(t+\tau) = \sum_{i=1}^{n} = w_i[\hat{X}_i(t+\tau) - X_{ij}(t+\tau)] \tag{2}$$

We update the time offset of each clock j against the scale at time $t+\tau$ given the measurements $X_{ij}(t+\tau)$. Clock j's offset is estimated using the measurement and prediction of each other clock, i, then these estimates are combined in a weighted average. The weights w_i are determined adaptively in equations 6 – 10. This is the maximum likelihood estimate of X_j if the w_i's are proportional to the reciprocal of the variances of the time residuals, and the residuals have a gaussian normal distribution.[1]

3. Frequency Estimate:

$$\hat{Y}_i(t+\tau) = \frac{X_i(t+\tau) - X_i(t)}{\tau} \tag{3}$$

$\hat{Y}_i$ is the estimate at time $t+\tau$ of the average frequency of clock i over the interval τ based on the latest two time updates, X_i.

4. Frequency Update:

$$Y_i(t+\tau) = \frac{1}{m_i+1}[\hat{Y}_i(t+\tau) + m_i Y_i(t)] \tag{4}$$

We incorporate the previous frequency update into an exponentially filtered estimate of the current average frequency offset of clock i. The exponential frequency–weighting time constant (m_i) is determined from the relative levels of white noise FM and random walk (or flicker) FM for clock i (equation 5).

5. Frequency Update, Exponential Time Constant:

$$m_i = 1/2\left[-1 + \left[\frac{1}{3} + \frac{4\tau_{Mini}^2}{3_{\tau_0^2}}\right]^{1/2}\right] \tag{5}$$

We determine m_i used in equation 4 to form the filtered estimate of the frequency of clock i. Here, τ_{Mini} is the integration time which gives the minimum value on a $\sigma_y(\tau)$ plot given that the clock's stochastic deviations are characterized by white and random walk FM. 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 (equation 1) given these two kinds of noise in the clock (white and random walk FM)[1]. 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.

6. Variance Estimate:

$$\hat{\epsilon}_i(\tau) = \left|\hat{X}_i(t+\tau) - X_i(t+\tau)\right| + K_i \tag{6}$$

$\hat{\epsilon}_i$ is the lack of predictability of clock i over the interval τ, being the difference between the prediction and the update. Thus it is an estimate of the deviation (square root of variance) of the clock based on the current measurement cycle. The additive term K_i accounts for the fact that the term in brackets on the right–hand side of equation 6 is biased because clock i is part of the ensemble. See equation 10 to calculate K_i.

7. Variance Update:

$$\epsilon_i^2(\tau)|_{t+\tau} = \frac{1}{N_\tau + 1}\left[\hat{\epsilon}_i^2(\tau) + N_\tau \epsilon_i^2(\tau)|_t\right] \tag{7}$$

$\epsilon_i^2(\tau)$ estimates the mean squared time error of clock i by filtering exponentially the estimate of clock i's deviation from the current measurement cycle. Since the noise characteristics of a clock may not be stationary, past measurements are de–weighted in the filtering process. 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 $\epsilon_i^2(\tau)$ can be estimated as $\tau^2\sigma_y^2(\tau)$.

8. Ensemble Variance:

$$\epsilon_x^2(\tau) = \left[\sum_{i=1}^{n} \frac{1}{\epsilon_i^2(\tau)}\right]^{-1} \tag{8}$$

$\epsilon_x^2(\tau)$ forms an estimate of ensemble time error. Any clock can only improve this number – a poorly performing clock cannot harm the stability of the ensemble.

9. Adaptive Clock Weights:

$$w_i = \frac{\epsilon_x^2(\tau)}{\epsilon_i^2(\tau)} \tag{9}$$

w_i is the weight to be used in equation 2 for clock i. When calculated this way, the resulting ensemble time stability can be shown to be optimized in a maximum likelihood sense, assuming a normal distribution of the noise of clock i with variance $\epsilon_i^2(\tau)$.

10. Bias of the Error Estimate:

$$K_i = \frac{.8\epsilon_x^2}{(\epsilon_i^2)^{1/2}} \tag{10}$$

K_i estimates the bias in the error estimate from the first term on the right of equation 6. This error estimate is biased small, on the average, because each clock is a member of the ensemble and sees itself through its weighting factor. The larger a clock's weight, the larger is the bias. Under the assumption of a normal distribution of clock noise the size of the bias can be estimated as given by equation 10, which is added to equation 5 in order to remove the bias, on the average[1].

THEORY: AT1–PLUS–VARIANCE

What is missing here is an estimate of the variance of the residuals of the frequency offset from the ensemble, Y. 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 equation 3, is a measurement of the frequency offset of clock i from the scale. We use a simple Kalman formalism to filter this measurement to estimate Y, where we use $\epsilon_i(\tau)$ as noise of the measurement. We model Y as having a random–walk noise and a fixed drift. Thus, Y is not the physical frequency as produced by the clock, but since we are filtering down the white frequency modulation (FM), it is only the random–walk component of the frequency of the clock, plus any drift.

The X terms do reflect the physical time offset of the clock from the scale, thus incorporating the white FM. The Y term is used to better predict the X values. We substitute equations 4 and 5 with the Kalman equations:

11. System Model:

$$Y_i(t+\tau) = Y_i(t) + D_i\tau + \eta(\tau). \tag{11}$$

Y_i is the random walk component of the frequency offset of clock i from the scale plus the drift offset. The random walk is driven by the white noise process $\eta(\tau)$.

12. Measurement Model:

$$\hat{Y}_i = Y_i + \epsilon_i(\tau) \tag{12}$$

The measurement $\hat{Y}_i$ is a direct measurement of Y_i, plus white noise.

The actual equations used for update are:

13. Variance Prediction:

$$\hat{P}(t+\tau) = P(t) + \sigma_\eta^2 \star \tau. \tag{13}$$

$\hat{P}$ is the prediction of variance of the residuals of Y which grows with τ according to σ_η^2, the variance of the white noise process, η, driving the random–walk FM.

14. Frequency Update:

$$Y(t+\tau) = \frac{\epsilon_i^2(\tau) \star Y(t) + \hat{P} \star \hat{Y}}{\epsilon_i^2(\tau) + \hat{P}} \tag{14}$$

We see that the Kalman formalism also gives us an exponential filter on Y. Thus in steady state this algorithm reduces to AT1 if the weights are chosen properly. This implies that this algorithm inherits the ability that AT1 has to model flicker frequency.

15. Variance Update:

$$P = \frac{\epsilon_i^2(\tau) \star \hat{P}}{\epsilon_i^2(\tau) + \hat{P}} \tag{15}$$

In the Kalman formalism the system parameters are known in advance. This system is a modification, an adaptive Kalman filter, where we estimate the "measurement noise," $\epsilon_i^2(\tau)$. This allows the variance of the residuals of $Y$to evolve both from an initial value, as is normal for the Kalman filter, and if $\epsilon_i^2(\tau)$ changes. This allows the exponential filter parameters on Y, as expressed in 14, to change with time, both after entering a new clock, and if the white FM level of the clock changes.

It is possible to solve for the steady state form of these equations and make identifications between the AT1 algorithm and our new algorithm. We find that the steady state value of P is:

16.

$$P = \frac{\eta^2}{2}\left(\sqrt{1 + 4 \cdot \frac{\epsilon_i^2(\tau)}{\eta^2}} - 1\right) \tag{16}$$

Making the appropriate identifications between equations 4 and 14 we find:

17.

$$m = \frac{\epsilon^2(\tau)}{P + \eta^2} \tag{17}$$

Using 16 in addition yields:

18.

$$m(m+1) = \frac{\epsilon^2(\tau)}{\eta^2} \tag{18}$$

These equations allow us to compare the performance of the two algorithms with the parameters m and η set consistently.

We close this theoretical section mentioning results from elsewhere. Jones and Tryon[2] have designed a time scale algorithm which is purely a Kalman filter. This scale, called TA(NIST), has been run at NIST in parallel with AT1 since about 1983. That filter is mathematically identical with the AT1 algorithm for the time and frequency predictions and updates[3]. The difference among these algorithms is the weighting of clocks in the time update and the exponential filter parameters in the frequency update. These differences effect the ensemble time they generate, which is realized as the time offsets, X_i, of the clocks against ensemble time.

We will show in simulation and with real clocks 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. The Jones Tryon filter attempts to estimate and minimize both time and frequency error. An additional problem with this is that since time is unobservable, 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 any reasonable amount of time, though it is suggestive of an undesirable situation.

SIMULATION

In simulation we show the following:

1. Both algorithms AT1 and AT1 plus frequency variance produce a time scale 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 AT1-plus-frequency- variance estimate of the confidence on the frequency offset estimate appears to be a reasonable estimate.

4. The use of this confidence estimate to determine frequency steps improves long term performance of the time scale.

Figure 1 illustrates item (1). Here we have generated data simulating clocks with various levels of white FM and random- walk FM. We have treated the problem as we would with real clocks where we only measure clock differences. We have computed the stability of each clock using an N-cornered hat technique[4]. The stability of the scale we have determined by taking the output value of clock minus scale and subtracting the generated value of clock minus truth. If we look directly at the variances of the generated data, we see significant differences between the variances computed directly, and those estimated from N-cornered hat (Figure 2). These differences must be due to apparent correlations in the data. This should come either from 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 correlation in the generated data.

Figure 3 shows a comparison of the output of a version of the Kalman filter which defines TA(NIST) with the simulated input data. We see that the Kalman algorithm has the stability of the clock with the best long term variations.

Figures 4 and 5 show the residuals from the AT1-plus-variance algorithm compared with the estimated confidence, from the estimated variance of frequency residuals. The algorithm estimates the random-walk component of a clock's frequency offset from the ensemble time. Since 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 of these two, the estimate minus truth, are the residuals plotted. The sigma value used in the plot is the root-mean-square 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 1 or 2 residuals crossing the lines of the sigmas. This seems to be the case.

Last, we inserted frequency steps in the simulated clocks. Figure 6 shows the frequency offset from the scale of the simulated clock 9, with a frequency step of 1×10^{-12} on MJD 46500. 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 the estimate from the AT1–plus–variance scale of the random–walk component of frequency. The reduction in the white FM is apparent. The scale was able to detect the frequency step, with the step detector set at 4 sigma, with sigma defined as above. When such a step is detected, we re–run the scale, removing the clock with the step until the scale can learn the new frequency value. Figure 8 shows the frequency offset from the scale of the simulated clock 1, with a frequency step of 2×10^{-12} on MJD 46100. The noise of this clock is almost all random walk FM as compared to its level of white 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. In Figure 9 we see the benefit from having detected the frequency steps. There is a significant improvement at an integration time of 128 d and longer.

REAL DATA

Last, we took data from real clocks at NIST over the period from December 31, 1988, to October 30, 1989. We ran our AT1–plus–variance algorithm on this data, including automatic frequency step detection and recalculation. We also took data using GPS common view measurements[5] with other laboratories: PTB, USNO, TUG, and NRC. Using an N–cornered hat technique, we were able to determine the variance of each of these. We compared these results with a similar analysis using the official NIST AT1 time scale. The results for the two scales run on NIST clocks as well as USNO and PTB are plotted in Figure 10. We find that the official AT1 and the new AT1–plus–variance scale are similar, though in long term the official scale is somewhat better. The official AT1 scale is watched carefully and administratively checked for time and frequency steps, as well as changes in clock performance in general. We find that human care adds much to a time scale.

BIBLIOGRAPHY

1. Allan, D. W., Weiss, M. A., The NBS Atomic Time Scale Algorithm: AT1, NBS Technical Note 1316, 1989.

2. Jones, R. H., and Tryon, P. V., "Continuous Time Series Models for Unequally Spaced Data Applied to Modeling Atomic Clocks," SIAM J. Sci. Stat. Comput., vol 8, no. 1, Jan 87, pp 71–81.

3. Weiss, M. A., Allan, D. W., Peppler, T. K., "A Study of the NBS Time Scale Algorithm," IEEE I&M, vol. 38, no. 2, April 89, pp 631–635.

4. Barnes, J., "Time Scale Algorithms Using Kalman Filters – Insights from simulation," Proceedings of the 2nd Symposium on Atomic Time Scale Algorithms, NBS, Boulder, CO.

5. Allan, D.W., Weiss, M.A., 1980, "Accurate time and frequency transfer during common view of a GPS satellite", Proc. 34th Ann. Freq. Control Symp., Ft Monmouth, pp 334–346.

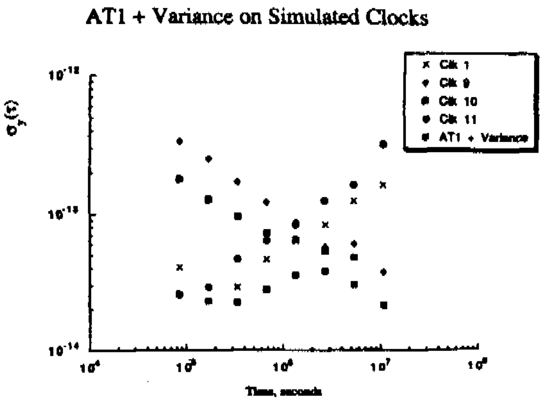

Figure 1: We have computed the stability of each simulated clock using an N-cornered hat technique. The stability of the scale we have determined by taking the output value of clock minus scale and subtracting the generated value of clock minus truth. The AT1 scale outperforms all clocks at all integration times.

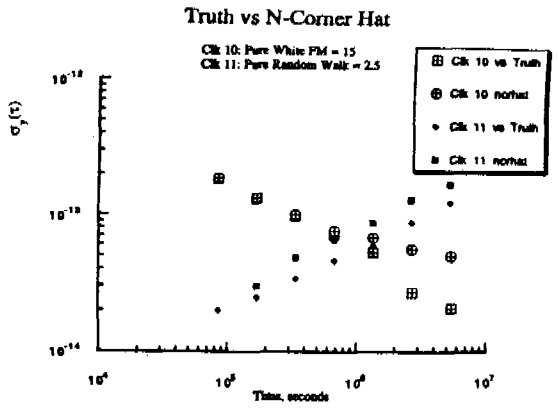

Figure 2: If we look directly at the variances of the generated data, we see significant differences between the variances computed directly, and those estimated from N-cornered hat. These differences must be due to apparent correlations in the data.

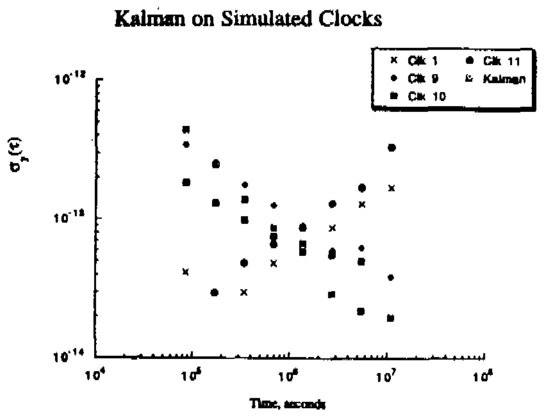

Figure 3: A comparison of the output of a version of the Kalman filter which defines TA(NIST) with the simulated input data. We see that the Kalman algorithm has the stability of the clock with the best long term variations.

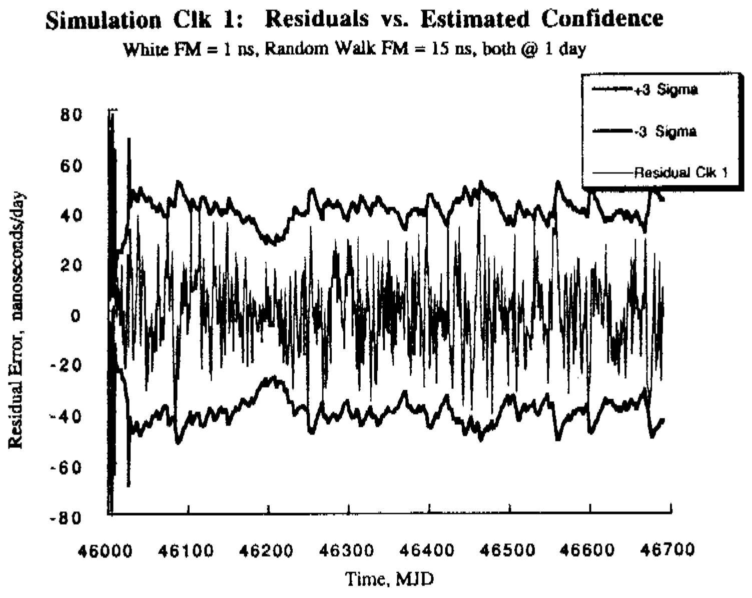

Figure 4: The residuals from the AT1-plus-variance algorithm compared with the estimated confidence, from the estimated variance of frequency residuals. The line plotted is the three-sigma value.

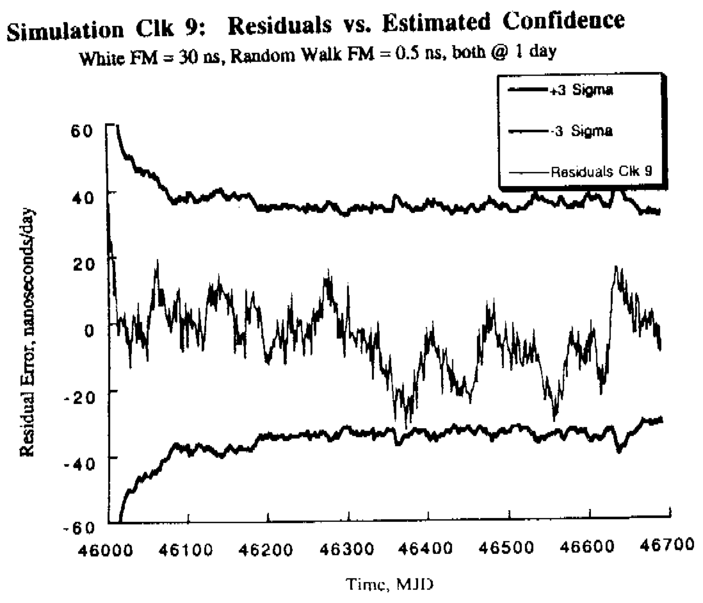

Figure 5: The residuals from the AT1-plus-variance algorithm compared with the estimated confidence, from the estimated variance of frequency residuals. The line plotted is the three-sigma value.

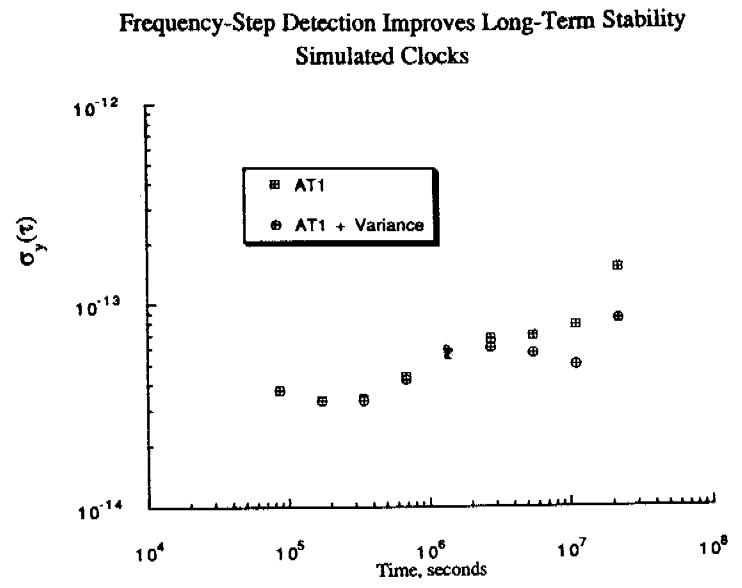

Figure 6: The frequency offset from the scale of the simulated clock 9, with a frequency step of $1*10^{-12}$ on MJD 46500.

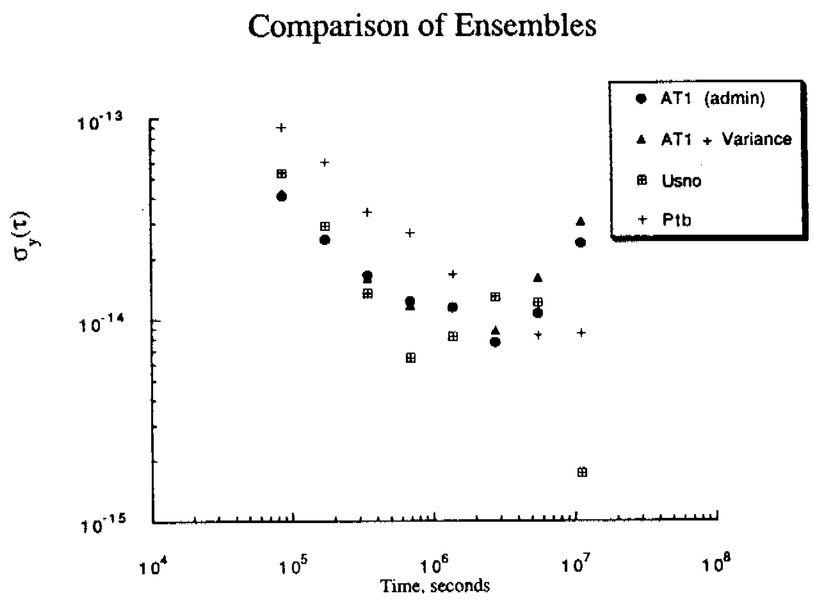

Figure 7: The estimate from the AT1-plus-variance scale of the random-walk component of frequency for clock 9. The scale was able to detect the frequency step, with the step detector set at 4 sigma.

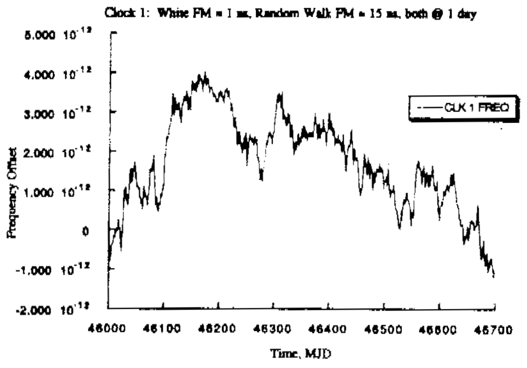

Figure 8: The frequency offset from the scale of the simulated clock 1, with a frequency step of $2*10^{-12}$ on MJD 46100. The estimate from the scale shows very little smoothing, yet the frequency step detector automatically found the step and removed the clock from the scale.

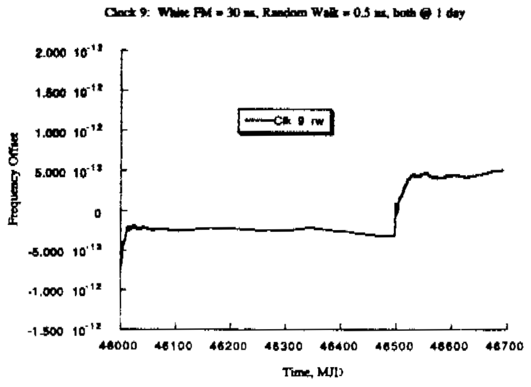

Figure 9: The benefit from having detected the frequency steps is seen in this diagram. There is a significant improvement at an integration time of 128 d and longer.

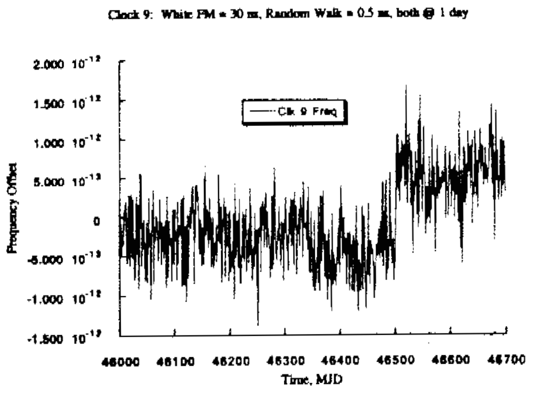

Figure 10: The results for the two scales, the official AT1 and the new AT1-plus-variance, run on NIST clocks as well as USNO and PTB.

QUESTIONS AND ANSWERS

GERNOT WINKLER, USNO: There are quite a few things that I didn't understand, but there is one thing that bothers me—at the beginning you said that, since you don't measure your time scale, but generate it, the use of a Kalman filter is inappropriate. Then you turn around and use it. Could you elaborate on that?

MR. WEISS: The use of a pure Kalman filter is inappropriate. Using Kalman techniques in part of the time scale is very useful. This is not a criticism of Sam Stein's algorithm which does a very similar thing. Kalman techniques have very important places. But to use the Kalman to generate the time itself is inappropriate. The result of that is what I have shown. You lock onto the clock with the best long term stability and you lose the short term.

MR. STEIN: I think that I would like to make a comment on that myself. The Kalman filter is merely a computational technique. It doesn't define a time scale, in the sense that we use the word. Whether or not you can use a Kalman filter, I strongly disagree with your statement, is simply whether or not you can formulate an appropriate definition of the time scale, definition of the ensemble in a way that is compatible with requirements. In fact, troubles that you pointed out, which are real, in some of the past Kalman filters were a result of the incorrect identification of the system noise covariance. An identification that did not agree with what the people that did those time scales really wanted to get out of the system. Other noise models produce different results.

JAMES BARNES, How often do you see frequency steps?

MR. WEISS: Dave Allan can probably answers that.

MR. ALLAN: It is a fairly difficult question because we don't really understand the mechanism. We do know that they occur. I have seen some clocks where they occur maybe once a year, several times a year and some clocks that seem not to have step problems. They can be fairly large and fairly significant.

MR. WEISS: We have recently discovered a possible source of the steps which was humidity, and we had never realized that before. Tracking and controlling humidity may remove some steps that we didn't understand the cause of before. As we do more research, we may understand the causes of other frequency steps.

UNIDENTIFIED QUESTIONER: How do you define 'truth'?

MR. WEISS: I don't define truth. When I generate a clock, I know what I have put into it. I generate a clock as being an offset against zero, zero being truth. If I make it up, I know what truth is.

SAME PERSON: Is it related to the Planck time?

MR. STEIN: No. You can relate it to what Einstein said, 'Time is what a clock reads'.

UNIDENTIFIED QUESTIONER: I have a question for you, Sam. You said that your model differed form the others in the process noise. As far as I can tell, the problem is one of observability. You can't change the observability of the model by changing process noise. If your H matrix is the same as everyone else's, then you have the same observability problem they do and all you are doing is de-weighting somehow past data. I think that I know how you are doing that and I can tell you.

MR. STEIN: The problem, as Mark pointed out, is not a problem of observability. We can discuss it further off-line.

DONALD PERCIVAL, U OF WASHINGTON: I just had a question to clarify what you are doing here. You are not using the (indecipherable)-Jones Kalman filtering model, is that correct? What you are doing then is taking one component of that model, the random walk component, and then using a Kalman filter by itself, a very simple one, to estimate that.

MR. WEISS: Exactly. I am using the output of AT1 time estimate against the scale as if it were a measurement of frequency against the scale. Then I filter that to obtain only the random walk component plus the variance, or confidence, on that estimate.

MR. PERCIVAL: Is there any problem, do you feel, with the observability. For example, in the time model, the random walk component is only one component of what the thing is actually doing. To separate that out, you are saying that it can be filtered out in a certain fashion.

MR. WEISS: I think that that is a subtle question, because the scale is a mathematical construct. It is something that, mathematically you are making up. The only observability of that is within the scale, there is no physical measurement of the clock against the scale. The closest that we can get to it is what the scale says is the estimate. There is no question of observability because I am looking at something that is defined.

GERNOT WINKLER, USNO: I would like to go back to the frequency step detection thing. It seems to me that it is intrinsically related to the question of post–processing or not. You can readily make a decision as to whether or not there was a frequency step, after the fact. You define as a frequency step as a systematic change in the average frequency. Whether it is systematic or random can only be decided with any kind of confidence only after the fact for small frequency steps. It is a question of the size of the frequency steps that you want to estimate *vs.* the question of post–processing or immediate detection. Here you have to solve the problem of whether you can tolerate post–processing or not. That will determine whether you can reliably detect frequency steps and account for them.

MR. WEISS: That's true. But, if you can detect that a step happened, in real time, that is, you detect now that a step happened yesterday—and the step is large enough, you can get some benefit from that now, in real time. You get the most benefit in post–processing, because then you go back over the last day as well. If the step is large enough that it is pulling the scale now, you can readjust that. You can also re–compute the scale for the last day and, even if you can't use that in real time, you can say that the frequency of the scale is now off, somewhat, from what I am actually outputting, and steer back to what the frequency should be.