KALMAN PLUS WEIGHTS: 
A TIME SCALE ALGORITHM*

Charles A. Greenhall
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, MS 298-100
Pasadena, CA 91109, USA
E-mail: Charles.Greenhall@jpl.nasa.gov

Abstract

KPW is a time scale algorithm that combines Kalman filtering with the basic time scale equation (BTSE). A single Kalman filter that estimates all clocks simultaneously is used to generate the BTSE frequency estimates, while the BTSE weights are inversely proportional to the white FM variances of the clocks. Results from simulated clock ensembles are compared to previous simulation results from other algorithms.

INTRODUCTION

The purpose of a time scale is to create a virtual clock from an ensemble of physical clocks whose differences from each other are measured at a sequence of dates (a date being the displayed time of a clock). The virtual clock is defined as an offset from one of the clocks, computed from the measurement data by some algorithm. We usually want the virtual clock to be quieter than any of the real clocks in both the short term and the long term.

One approach, which was tried in the early 1980s [1], is to run a Kalman filter on the clock difference measurements, the noise of each clock having previously been modeled by a stochastic linear system. The filter produces an estimate, unbiased and with minimum error variance, of the phase and frequency of each clock; moreover, if we offset the tick of each clock by its phase estimate, we arrive at a single point on the time axis (if the measurements are noiseless). It makes sense, then, to regard this point as the estimated origin of the ensemble, and to use the sequence of these values as a time scale. This time scale, which was realized as TA (NIST), was reported to follow the clock with the best long-term stability, regardless of its short-term stability [2]. My goals have been to reproduce this finding, understand it, and improve the method. I seem to have achieved the first and third goals, but not the second.

It turns out that a good time scale algorithm can be constructed by injecting some of the Kalman-filter information into the traditional "basic time scale equation" (BTSE), which requires frequency

*This work was performed by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.
estimates and a set of weights. In brief, here is the new time scale algorithm, called “Kalman plus weights (KPW).”

1. Initialize the Kalman filter properly, and run it on the clock models and difference measurements.
2. Throw out the Kalman phase estimates.
3. Use the Kalman frequency estimates in the BTSE, whose weights are made inversely proportional to the white FM variances of the clocks.

After describing the algorithm in detail, I show results from simulations of clocks with independent white FM and random-walk FM (RWFM) components. Included are comparisons with previous simulation results for two other time scales that make heavy use of Kalman filtering: the Barnes–Allan “frequency Kalman” [3] and Stein’s KAS-21 [4].

TERMINOLOGY AND NOTATION

I shall try to introduce a consistent and suggestive notation in which to work. The ensemble has \( n \) clocks \( H_1, \ldots, H_n \). A date is the displayed time of a clock, determined by counting its oscillations. At date \( t \), the following quantities are defined:

- \( h_i(t) \): time coordinate (on some time scale) of \( H_i \)'s tick when it shows date \( t \); not directly observable.
- \( h_0(t) \): time coordinate of an ideal clock \( H_0 \); \( h_0(t) = a + bt \) for some constants \( a, b \). This is not an unattainable concept; an ideal clock can and will be defined by extrapolating the initial state of a physical clock; see the section on startup.
- \( h_e(t) \): time scale or ensemble time, the time coordinate of a virtual clock \( H_e \), to be determined by its computed offsets from the physical clocks.
- \( x_e(t) = h_e(t) - h_0(t) \), offset of \( H_e \) from \( H_0 \), also called a time scale here.
- \( x_i(t) = h_i(t) - h_0(t) \), offset of \( H_i \) from \( H_0 \).
- \( x_{ij}(t) = x_i(t) - x_j(t) = h_i(t) - h_j(t) \), clock difference measurements, taken at an increasing sequence of dates \( t_0, t_1, \ldots \). This study assumes noiseless measurements.
- \( x_{i\epsilon}(t) = x_i(t) - x_e(t) = h_i(t) - h_e(t) \), offset of \( H_i \) from \( H_e \). The \( x_{i\epsilon}(t) \) are to be computed as statistics of the measurements through date \( t \), perhaps with some initial conditions. A time scale \( x_e(t) \) is determined by \( x_i(t) \) and \( x_{i\epsilon}(t) \) for some \( i \). When the measurements are noiseless, it usually turns out that any \( i \) can be used, that is, \( x_i(t) - x_{i\epsilon}(t) \) gives the same value \( x_e(t) \) for all \( i \). An equivalent condition is

\[
x_{i\epsilon}(t) - x_{j\epsilon}(t) = x_{ij}(t), \quad i, j = 1, \ldots, n. \tag{1}
\]

If (1) is fulfilled, I shall say that the offsets \( x_{i\epsilon}(t) \) are consistent with the measurements. This just means that the set of points \( \{x_{i\epsilon}(t) : i = 1, \ldots, n\} \) is a rigid translation of the set \( \{x_i(t) : i = 1, \ldots, n\} \).

\(^1\text{A tradename of Timing Solutions Corporation}\)
AVERAGE TIME SCALES

This category includes many of the time scales in actual use [5]. An average time scale is defined recursively at measurement date \( t \) from measurement date \( t - \tau \) by an equation of form

\[
x_e(t) = x_e(t - \tau) + \sum_{i=1}^{n} w_i(t) [x_i(t) - x_i(t - \tau) - \tau \dot{y}_i(t - \tau)]. \tag{2}
\]

The weights \( w_i(t) \) (which add to 1) and the frequency estimates \( \dot{y}_i(t) \) depend on the measurements through date \( t \). From (2) and the previous definitions, there follows the basic time scale equation (BTSE),

\[
x_{je}(t) = \sum_{i=1}^{n} w_i(t) [x_{ji}(t) + x_{ie}(t - \tau) + \tau \dot{y}_i(t - \tau)], \quad j = 1, \ldots, n, \tag{3}
\]

a recursive equation for the offsets \( x_{je}(t) \). This is the computation that is actually performed to obtain the offsets of the virtual clock from the physical clocks. These offsets are consistent with the measurements.

Average time scales usually calculate \( \dot{y}_i(t) \) as an estimate of the frequency of \( H_i \) relative to the scale \( H_e \) as calculated through date \( t \). In the present formulation, \( \dot{y}_i(t) \) is an estimate of the frequency of \( H_i \) relative to \( H_0 \), not \( H_e \). This is the case for the Kalman-based estimate discussed below; the Kalman filter knows nothing about \( H_e \).

CLOCK MODEL AND KALMAN FILTER

At this stage of development, I am using a two-state clock model: white FM plus RWFM. Jones and Tryon [1] showed how to integrate the differential-equation model to a stochastic difference-equation model for discrete measurement dates, which may be unequally spaced. For the \( i \)th clock, the equations taking the state \([x_i, y_i]\) from date \( t - \tau \) to date \( t \) can be stated as

\[
x_i(t) = x_i(t - \tau) + \tau y_i(t - \tau) + w_{xi}(t, \tau) \tag{4}
\]

\[
y_i(t) = y_i(t - \tau) + w_{yi}(t, \tau).
\]

The process noise vector \([w_{xi}(t, \tau), w_{yi}(t, \tau)]\) (uncorrelated over dates and clocks) has covariance matrix

\[
Q_i(\tau) = q_{xi} \begin{bmatrix} \tau & 0 \\ 0 & 0 \end{bmatrix} + q_{yi} \begin{bmatrix} \tau^3/3 & \tau^2/2 \\ \tau^2/2 & \tau \end{bmatrix}, \tag{5}
\]
in which the \( q \)'s, which we assume to be known, specify the white FM and RWFM noise levels. In terms of them, the Allan variance of the \( i \)th clock is

\[
\sigma_{yi}^2 (\tau) = \frac{q_{yi}}{\tau} + \frac{q_{yi}^2 \tau}{3}.
\]

The overall state vector is \( X (t) = [x_1(t), y_1(t), \ldots, x_n(t), y_n(t)] \). In a standard way [6], which will not be repeated here, the Kalman filter uses the model (4)-(5) and the measurements \( x_{ij}(t) \) to obtain a recursive estimate \( \hat{X}(t) = [\hat{x}_1(t), \hat{y}_1(t), \ldots, \hat{x}_n(t), \hat{y}_n(t)] \) and its error covariance matrix \( \hat{P}(t) \) from the same quantities at date \( t - \tau \).

It turns out that the Kalman phase estimates \( \hat{x}_i(t) \) are consistent with the clock measurements in the sense that \( \hat{x}_i(t) - \hat{x}_j(t) = x_{ij}(t) \); consequently, it makes sense to define a natural Kalman time scale by

\[
x_{eK}(t) = x_i(t) - \hat{x}_i(t)
\]

(the same for all \( i \)). It is this scale that was used for TA (NIST) and found wanting.

**Startup**

The Kalman filter must be initialized at a starting date \( t_1 \) by providing a state estimate \( \hat{X}(t_1), \hat{P}(t_1) \). By taking care with this task, we can establish a reference for the ensemble and make the filter settle down quickly [7]. Let us take noiseless clock difference measurements \( x_{ij}(t_0), x_{ij}(t_1) \), where \( t_1 = t_0 + \tau \). Without loss of generality, we can assume that the \( x_i(t_0) \) are known exactly. Some initial information about the random walk frequency states is needed, too. For this purpose, let us regard \( H_1 \) as a master clock whose initial frequency state, relative to some ideal clock \( H_0 \), can be defined or estimated. Thus, let \( \hat{y}_1(t_0) \) be some unbiased prior estimate of \( y_1(t_0) \), with error variance \( p_1 \). We can always set:

\[
x_1(t_0) = 0, \ y_1(t_0) = 0, \ p_1 = 0.
\]

The implication of (8) is that \( H_0 \) is defined as the noiseless extrapolation of the initial state of \( H_1 \), which, though unknown, can still act as a reference. This convention will be called the **Master Clock 1 Startup**. This is not the same thing as using \( H_1 \) as a master clock during the run [8]; after startup, its state is estimated (relative to \( H_0 \)) on a par with the states of the other clocks.

Here are the initialization equations, whose derivation is omitted:

\[
\hat{y}_i(t_0) = \hat{y}_1(t_0) + \frac{1}{\tau} [x_{i1}(t_1) - x_{i1}(t_0)], \quad i = 1, \ldots, n
\]

\[
\hat{x}_i(t_1) = x_i(t_0) + \tau \hat{y}_i(t_0), \quad \hat{y}_i(t_1) = \hat{y}_i(t_0), \quad i = 1, \ldots, n.
\]

These give \( \hat{X}(t_0) \) and \( \hat{X}(t_1) \). For the Master Clock 1 Startup (assumed for simplicity), we have \( \hat{P}(t_1) = AQ(\tau)A^T \), where \( Q(\tau) \) is the overall process covariance matrix with diagonal blocks
\[ Q_t(\tau), \quad \text{and} \]

\[
A = \begin{bmatrix}
1 & 0 \\
0 & 1 \\
1 & 0 & 0 & 0 \\
\frac{1}{\tau} & 0 & -\frac{1}{\tau} & 1 \\
\vdots & \vdots & \ddots & \ddots \\
1 & 0 & 0 & \cdots & 0 & 0 \\
\frac{1}{\tau} & 0 & 0 & \cdots & -\frac{1}{\tau} & 1
\end{bmatrix}.
\]

**KPW ALGORITHM**

The KPW time scale algorithm can now be quickly given. Set \( x_{ie}(t_0) = x_i(t_0) \) (that is, \( x_e(t_0) = 0 \)). Initialize the Kalman filter from the clock difference measurements at \( t_0 \) and \( t_1 \). Run the Kalman filter on the measurements at dates \( t_2, t_3, \ldots \) to produce state estimates \( \hat{x}_i(t_k), \hat{y}_i(t_k) \). At measurement date \( t = t_k, k \geq 1 \), apply the BTSE (3), where \( t - \tau = t_{k-1} \). The Kalman phase estimates \( \hat{x}_i \) are not used at all.

It remains to specify the BTSE weights. To argue towards a reasonable choice, define the estimated RWFM component of \( x_i(t) \) as the discrete time integral of \( \hat{y}_i(t) \),

\[
\hat{x}_{i,RWF}(t) = \hat{x}_{i,RWF}(t-\tau) + \tau \hat{y}_i(t-\tau) \\
\hat{x}_{i,RWF}(t_0) = x_i(t_0).
\]

Assume constant weights \( w_i \). Summing (2) over the measurement dates from \( t_1 \) to \( t \), we obtain

\[
x_e(t) = \sum_{i=1}^{n} w_i [x_i(t) - \hat{x}_{i,RWF}(t)].
\]

We can regard the quantity in brackets as an estimate of the white FM component of \( H_i \). Thus, the implicitly defined time scale is a weighted average of approximate white FM components. To try to minimize the instability of \( x_e(t) \), we make \( w_i \) proportional to \( 1/q_{x_i} \). This method has been used for all the simulations. In practice, if the \( q \)'s are revised in the middle of a run, then so would the \( w \)'s.

**SIMULATIONS**

These were run for various sets of clocks as determined by their \( q \)'s. For convenience, the data were generated at equally spaced dates. The model equations (4) were used to generate the true clocks. The Kalman filters were initialized by the Master Clock 1 Startup condition (8), and were mechanized by a covariance square root method [9,10] to avoid numerical instability and problems with singular covariance matrices. Because the true clocks were available, the time scales were computed from (9) and (10) instead of the BTSE (3).
Two Opposite Clocks

The simplest and most revealing example is two clocks that are as opposite as they can be: $H_1$ is pure white FM, while $H_2$ is pure RWFM. The results are shown in Fig. 1 (arbitrary scaling). The upper plot shows the phases of the true clocks and the KPW time scale $H_e$. The natural Kalman time scale (7) exhibits an extreme form of its well-known behavior: it is exactly equal to $x_1(t)$. This means that the Kalman phase estimate $\hat{x}_1(t)$ is identically zero; all of $H_1$’s white FM is thrown onto $\hat{x}_2(t)$.

The middle plot shows the total frequency (difference quotient of the phase) for the KPW scale $H_e$ and the true clocks; the curves are offset for clarity. For $H_2$, the total frequency is the same as the RWFM state $y_2(t)$. The two upper intertwined curves, which are $y_2(t)$ and its estimate $\hat{y}_2(t)$, show that the Kalman filter does an excellent job of estimating this frequency state.

In the lower plot, the straight lines are the theoretical Allan deviations of the true clocks; the dots are the measured Allan deviations. The KPW scale $H_e$ is only moderately noiser than $H_2$ for short $\tau$, and about the same as $H_1$ for long $\tau$. All the weight is on $H_2$; this means that $x_e(t) = x_2(t) - x_{2,\text{RWFM}}(t)$, with $H_1$ playing no role in the BTSE. Nevertheless, the long-term behavior of the scale seems to be governed by $H_1$.

Eleven NIST Cs Clocks

This example comes from a study by Barnes and Allan [3] based on simulations of a set of cesium clocks whose $q$’s had previously been measured. Figure 2 shows the result of a simulation using the same $q$’s. The phases of all the true clocks and the total frequencies of three of them are shown. The crosses in the lower plot (data from Fig. 9 of [3]) show the Allan deviation of a time scale derived from the “frequency Kalman” filter, which uses pseudo-measurements of frequency differences. That scale dips a little lower than the KPW scale at $\tau = 10^5$ s. At $\tau = 10^4$ s, though, the measured KPW Allan deviation is $1.30 \times 10^{-14}$, close to the minimum value $1.27 \times 10^{-14}$ that can be achieved by a weighted average of the white FM components of these clocks.

Eight Imaginary Clocks

This example reproduces a simulation that was carried out by Stein [4] on an imaginary eight-clock ensemble to demonstrate the KAS-2 time scale algorithm. The odd-numbered clocks all have the same $q$’s, as do the even-numbered clocks. Figure 3 shows the results of simulating this ensemble; the crosses in the lower plot show the KAS-2 stability from Stein’s Fig. 1. For each $\tau$, the measured KPW Allan deviation is less than 60% of the theoretical Allan deviation of the best clock for that $\tau$.

CONCLUSIONS

The KPW time scale algorithm has been demonstrated in a simulation playpen with perfect knowledge of the stochastic clock models and their noise levels. Under these conditions, KPW seems to be competitive with other Kalman-based time scale algorithms. The natural Kalman time scale, which does not use the basic time scale equation, has again been shown to be noisy in the short
term because the Kalman filter attributes the white FM noises to the wrong clocks. I do not yet understand why this happens.

A related symptom is the unbounded growth of portions of the covariance matrix. This growth does not harm the frequency estimates, especially if the Kalman filter is mechanized by a square-root method. Nevertheless, as Weiss and Weissert wrote [2], "it is suggestive of an undesirable situation." In view of Brown's [8] work, it might be possible to reduce this matrix transparently.

The KPW algorithm might serve as the foundation of a time scale that gives real-time results. Of course, a practical time scale must provide for clock insertion and removal, outlier detection and rejection, jumps in phase and frequency, steering, adaptive estimation of the $q$'s, and so on. In addition, the clock and measurement models should be expanded to include random run FM, white PM, and measurement noise.

REFERENCES


Fig. 1. Two opposite clocks. Clock 1 = white FM, clock 2 = RWFM
Fig. 2. Eleven simulated NIST Cs clocks
Fig. 3. KPW and KAS-2 on eight simulated clocks