Addition of Random Run FM Noise to the KPW Time Scale Algorithm*

Charles A. Greenhall
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Dr.
Pasadena, CA 91109, USA
E-mail: charles.greenhall@jpl.nasa.gov

January 7, 2002

Abstract

The KPW (Kalman plus weights) time scale algorithm uses a Kalman filter to provide frequency and drift information to a basic time scale equation. The Kalman filter is the same as the one previously used for the TA(NIST) scale. This paper extends the algorithm to three-state clocks and gives results for a simulated eight-clock ensemble.

1 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, where by "date" we mean the displayed time of a clock as determined by counting its oscillations. The virtual clock is defined as an offset from one of the physical clocks, the offset being computed from the measurement data by some algorithm. The usual goal of the algorithm design is to produce a virtual clock that is more stable than any of the physical clocks in both the short term and the long term, as measured by some standard stability measure such as Allan deviation or Hadamard deviation.

*This work was performed by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

A straight Kalman filter approach to this problem has been tried at least twice [10][5]. The noise of each clock is modeled as a sum of white FM, random walk FM (RWFM), and random run FM (RRFM, random walk of drift) with known noise levels. The entire ensemble is modeled by a linear stochastic differential equation, whose state vector is estimated in a straightforward way by a Kalman filter from the clock difference measurements. Because the measurements are assumed noiseless, if we offset the tick of each clock by its Kalman phase estimate, we arrive at a single point on the time axis. It makes sense to regard this point as the estimated ensemble time, and to use the sequence of these values as a time scale. This time scale, realized as TA(NIST), was reported to follow the clock with the best long-term stability, regardless of its short-term stability [12]. The goals of the present study are to reproduce this result and to find a better way to use this Kalman filter in a time scale algorithm.

In a previous paper [7], the author carried out this program for two-state clocks, which have white FM and RWFM noise only. In the present paper, the results are extended to three-state clocks, which also contain RRFM. The modified KPW (Kalman plus weights) algorithm can be summarized as follows.

1. Initialize the Kalman filter and run it on the clock models and difference measurements.

2. Throw out the Kalman phase estimates. Their pathological behavior is the reason for the poor short-term performance of TA(NIST).

3. Use the Kalman frequency and drift estimates in a basic time scale equation (BTSE) whose weights are inversely proportional to the white FM variances of the clocks. The Kalman filter is allowed to run independently of the BTSE calculation.

In the following sections, the components of the algorithm are described in more detail. Although much of this material might be familiar, it serves to establish notation and introduce an occasional conceptual alteration. Results are shown from a simulation of an imaginary ensemble of eight clocks. In this simulation, the measured Hadamard deviation of the KPW scale is about a factor of two below the lower envelope of the Hadamard deviations of the clocks.

2 Jones-Tryon clock model

Let the ensemble have n clocks. At date t, the ith clock has state vector

$$X_i(t) = [x_i(t), y_i(t), z_i(t)]^{\mathrm{T}},$$

where $x_i(t)$ is the phase (or time) state with all three noises, $y_i(t)$ is the normalized RWFM + RRFM frequency state, and $z_i(t)$ is the RRFM drift state. These states are to be regarded as residuals from some ideal clock whose rate is constant; for example, the ideal clock could be defined by extrapolating from the time and rate of one of the physical clocks at an initial date t_0.

The evolution of the state of the ith clock from measurement date $t-\tau$ to measurement date t is described by the stochastic difference equation

$$X_i(t) = \Phi(\tau) X_i(t-\tau) + W_i(t,\tau), \tag{1}$$

where

$$\Phi(\tau) = \begin{bmatrix} 1 & \tau & \tau^2/2 \\ 0 & 1 & \tau \\ 0 & 0 & 1 \end{bmatrix},$$

and the process-noise vector $W(t,\tau)$ has covariance matrix

$$Q_i(\tau) = q_{xi} \begin{bmatrix} \tau & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + q_{yi} \begin{bmatrix} \tau^3/3 & \tau^2/2 & 0 \\ \tau^2/2 & \tau & 0 \\ 0 & 0 & 0 \end{bmatrix} + q_{zi} \begin{bmatrix} \tau^5/20 & \tau^4/8 & \tau^3/6 \\ \tau^4/8 & \tau^3/3 & \tau^2/2 \\ \tau^3/6 & \tau^2/2 & \tau \end{bmatrix}.$$

The q factors (differential variances), which specify the noise levels, are related to the Hadamard variance of the clock [9] by

$$\mathrm{H}\sigma_{yi}^2(\tau) = \frac{q_{xi}}{\tau} + \frac{1}{6} q_{yi}\tau + \frac{11}{120} q_{zi}\tau^3.$$

The clock difference measurements at date t are

$$x_{i1}(t) = x_i(t) - x_1(t), \quad i = 2, \ldots, n.$$

3 Kalman filter and natural time scale

The model and measurements determine a Kalman filter, which will not be described here [6]. For each measurement date t, the filter produces state estimates for each clock,

$$\hat{X}_i(t) = [\hat{x}_i(t), \hat{y}_i(t), \hat{z}_i(t)]^{\mathrm{T}},$$

and an error covariance matrix $\hat{P}(t) = E\left(X(t) - \hat{X}(t)\right)\left(X(t) - \hat{X}(t)\right)^{\mathrm{T}}$, where $X(t)$ is the overall state vector of length $3n$. As a general property of

Kalman filters, if the measurements are noiseless, then the estimated state also satisfies the measurement equations, i.e.,

$$x_{i1}(t) = \hat{x}_i(t) - \hat{x}_1(t), \quad i = 2, \ldots, n.$$

It follows that the quantity

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

(which is just the phase estimate error) does not depend on i. We call $x_{eK}(t)$ the *natural Kalman time scale*; it is this scale that became TA(NIST). Its instability is high in the short term because the phase estimates are poor; it seems that the white FM noise in the system is wrongly distributed among the clocks.

3.1 Initializing the Kalman filter

The Kalman filter must be given an initial state estimate and error covariance matrix. For two-state clocks, it seems that a simple method based on two measurements will start the filter smoothly enough [7]. When drift is involved, it is harder to get reasonable estimates from suboptimal schemes involving three or four measurement dates. The author finally realized that we can make the Kalman filter itself do the work of assigning an initial error covariance by running it with no data, just the covariance updates. We start the filter with a zero error covariance, and run it until the error covariance matrix $\hat{P}_{yz}$ of the y and z states settles down. It is not necessary to wait until it actually converges (if it ever does). Why just the y and z states? Because the x covariances diverge strongly; the Kalman filter seems to know that it is doing a poor job of estimating the clock phases. In fact, we now set to zero all the elements of $\hat{P}$ that involve an x state; it can be proved that doing so leaves the future y and z estimates unchanged. We can regard the initial $\hat{P}_{yz}$ as generating Type B uncertainties [1] of the initial y and z states, however they may actually be obtained.

For the simulations in this study, the initial y and z estimate errors were generated as zero-mean Gaussian random variables whose covariance is the initial $\hat{P}_{yz}$ that was determined from the above procedure.

3.2 Square-root filtering

Since the publication of Gelb's book [6], several factorization methods for Kalman filtering have been developed [3][8]. These methods, which are algebraically equivalent to the conventional Kalman mechanization, work by propagating a Choleski factor $\hat{C}$ of the error covariance matrix $\hat{P}$, not $\hat{P}$ itself. In exchange for a modest increase in complexity there are several advantages: the numerical computations are more stable, singular and nearly singular covariance matrices are not a problem, and the covariance matrices do not have to be symmetrized. Having programmed one of these methods in Matlab, the author has used it throughout this study with no problems.

4 Basic time scale equation

Most practical time scale algorithms use a form of the basic time scale equation (BTSE) [2]. To include drift estimates, we use the modification introduced by Breakiron [4]. The BTSE has several equivalent forms. One is a recursive definition of the time scale $x_e(t)$ in terms of the non-observable quantities $x_i(t)$:

$$x_e(t) = x_e(t-\tau) + \sum_{i=1}^{n} w_i(t)\left[x_i(t) - x_i(t-\tau) - \tau\hat{y}_i(t-\tau) - \tfrac{1}{2}\tau^2\hat{z}_i(t-\tau)\right]. \tag{2}$$

How the scale behaves depends on how the weights $w_i(t)$ (which add to 1) are chosen, and how the estimates $\hat{y}_i(t-\tau)$ and $\hat{z}_i(t-\tau)$ are determined from previous observations. By subtracting any $x_j(t)$ we obtain a calculation of the offset $x_{ej}(t) = x_e(t) - x_j(t)$ in terms of observed and computed quantities:

$$x_{ej}(t) = x_{ej}(t-\tau) + \sum_{i=1}^{n} w_i(t)\left[x_{ij}(t) - x_{ij}(t-\tau) - \tau\hat{y}_i(t-\tau) - \tfrac{1}{2}\tau^2\hat{z}_i(t-\tau)\right],$$

where $x_{ij}(t) = x_i(t) - x_j(t)$.

Most BTSE time scales determine $\hat{y}_i(t-\tau)$ and $\hat{z}_i(t-\tau)$ as estimates of the departure of the frequency and drift of the ith clock from the previously computed time scale. Here, we allow these quantities to represent the estimated departures from an ideal noiseless clock; this allows the Kalman filter y and z estimates to be used. It is important to observe that information from the BTSE is not fed back to the Kalman filter, which is kept as pure as possible; its only job is to produce good frequency and drift estimates from the clock models and measurements.

5 The weights

All the components of the KPW algorithm have now been specified. It remains to argue why it is a good idea to make the weights inversely proportional to the q_{xi}. Assume that $w_i(t)$ does not depend on t. Define new phase estimates $\check{x}_i(t)$ recursively by

$$\check{x}_i(t) = \check{x}_i(t-\tau) + \tau\hat{y}_i(t-\tau) + \tfrac{1}{2}\tau^2\hat{z}_i(t-\tau). \tag{3}$$

If $\hat{y}_i(t)$ and $\hat{z}_i(t)$ are good estimates of $y_i(t)$ and $z_i(t)$, then, in view of the x components of (1), it is reasonable to regard $\check{x}_i(t)$ as a good estimate of the phase of clock i without its white FM noise. Using (3) in (2) and summing over the measurement dates gives

$$x_e(t) = \sum_{i=1}^{n} w_i\left[x_i(t) - \check{x}_i(t)\right] + \text{const.} \tag{4}$$

In view of what we just said, we regard the quantity in brackets as an estimate of the white FM portion of the phase of clock i, and regard the summation as approximating a linear combination of independent random-walk processes with differential variances q_{xi}. This says that $x_e(t)$ is approximately a random walk whose differential variance is minimal when the weights are inversely proportional to the differential variances of the white FM clock noises. This approximation works best when τ is small enough that white FM dominates the observed short-term clock noises.

6 Simulation example

One of the simulation examples in the author's previous paper [7] on two-state clocks reproduces an imaginary eight-clock ensemble that was simulated by Stein [11]. The odd-numbered clocks all have the same q's; similarly for the even-numbered clocks. For the purpose of the present paper, an RRFM term was added to the odd-numbered clocks. A run of 1.8×10^8 seconds of hourly measurements was simulated. The results are shown in Fig. 1, in which H_i is an abbreviation for clock i. Figure 1(f) shows the theoretical Hadamard deviations as smooth curves and the measured Hadamard deviations as points. Figure 1(a) shows the KPW time scale along with the true clock phases. Figure 1(b) is a short-term picture of the KPW scale and the natural Kalman scale; the latter is much rougher. Figure 1(c) shows the total frequency of the KPW scale and two clocks. (Total frequency is the difference quotient of the phase.) Figure 1(d) and (e) show how well the Kalman filter estimates the frequency *states* (RWFM and RRFM only) and the drift states. Finally, Fig. 1(f) shows the measured stability of the KPW scale (connected circles) and the natural Kalman scale (crosses). Although the two scales are almost identical in long term, the natural Kalman scale is almost as noisy in short term as the even-numbered clocks. The KPW scale is about a factor of 2 below the lower envelope of all the clocks, except at the largest values of τ, where the confidence is low.

7 Conclusions

The results of [7] have successfully been extended to three-state clocks with white FM, random walk FM, and random run FM noise. The natural Kalman time scale TA(NIST), which uses the Kalman phase estimates, is almost as noisy in short term as the noisiest clock. The KPW time scale, which uses the Kalman frequency and drift estimates, seems to perform well at all averaging times, at least in a simulation playpen in which each clock is governed by its assumed model. The author has not tried the algorithm on real clock data.

The KPW algorithm could be the foundation of a practical real-time time scale algorithm, which would also have to handle clock insertion and deletion, outliers, jumps, adaptive q estimation, and adjustment of the weights. In addition, the ensemble model should be expanded to include white PM noise and measurement error.

References

[1] *Guide to the Expression of Uncertainty in Measurement*, Geneva: International Organization for Standardization, 1995.

[2] *Selection and Use of Precise Frequency and Time Systems*, Geneva: International Telecommunication Union, 1997.

[3] G. J. Bierman, *Factorization Methods for Discrete Sequential Estimation*, New York: Academic Press, 1977.

[4] L. A. Breakiron, "Timescale algorithms combining cesium clocks and hydrogen masers," *Proc. 23rd PTTI Meeting*, pp. 297–305, Pasadena, CA, 1991.

[5] K. R. Brown Jr., "The theory of the GPS Composite Clock," *Proc. ION GPS-91 Meeting*, pp. 223–241, Albuquerque, Sep. 1991.

[6] A. Gelb, ed., *Applied Optimal Estimation*, Cambridge, MA: MIT Press, 1974.

[7] C. A. Greenhall, "Kalman Plus Weights: A Time Scale Algorithm," *33rd PTTI Meeting*, Long Beach, CA, Nov. 2001.

[8] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice Using Matlab*, New York: Wiley, 2001.

[9] S. T. Hutsell, "Relating the Hadamard variance to MCS Kalman filter clock estimation," *Proc. 27th PTTI Meeting*, pp. 291–301, San Diego, CA, 1995.

[10] R. H. Jones and P. V. Tryon, "Continuous time series models for unequally spaced data applied to modeling atomic clocks," *SIAM J. Sci. Stat. Comput.*, vol. 8, no. 1, pp. 71–81, Jan. 1987.

[11] S. R. Stein, "Advances in time-scale algorithms," *Proc. 24th PTTI Meeting*, pp. 289–302, McLean, VA, Dec. 1992.

[12] M. Weiss and T. Weissert, "A new time scale algorithm: AT1 plus frequency variance," *Proc. 21st PTTI Meeting*, Redondo Beach, CA, pp. 343–357, Nov. 1989.

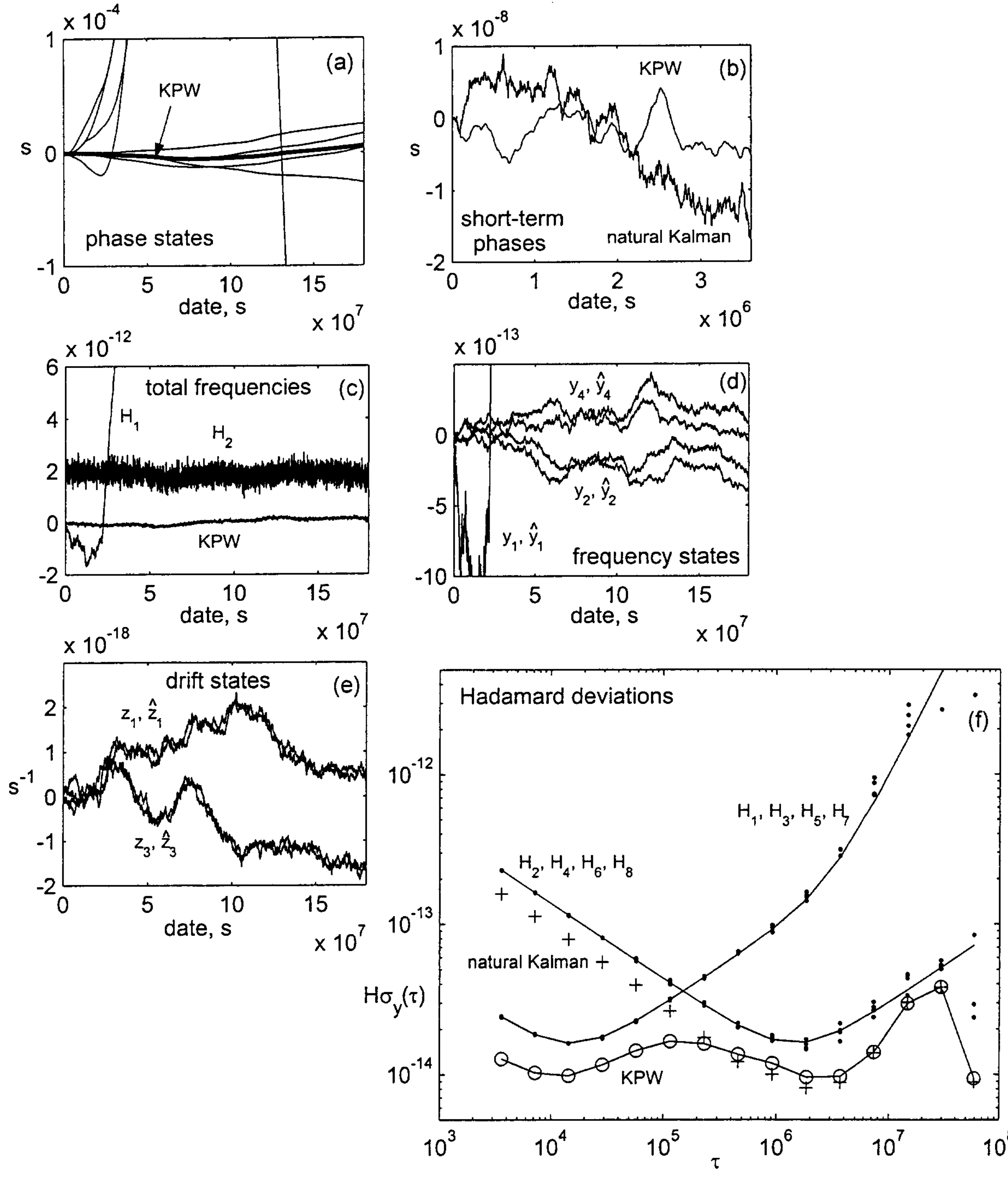

Fig. 1. Results of eight-clock simulation