Joint Time and Frequency Offset Estimation for Clock Synchronization

Josep M. Ferrandiz

November 8, 1995

Abstract

Timing data gathered from network traffic is fed into a Kalman filter designed to jointly estimate the clock offset and the clock frequency offset with respect to the time server. The method proposed builds a statistical framework for synchronization of computer clocks that allows a probabilistic interpretation of the quantities involved. The tests show that the filter performance is for all practical purposes optimal.

1. Introduction

Certain computer applications require clock synchronization. This happens, for instance, when several computers are networked and run applications that need a tight clock synchronization such as multimedia flows or in some secure environments. Other applications appear in distributed protocol analysis in WANs or in remote control of distributed measurement systems. The common thread of all these settings is the need for correlating distributed events. This correlation cannot be carried out without a common time base. A general review of clock synchronization can be found in [4].

The Internet Activities Board clock synchronization standard is the Network Time Protocol, Version 3 (NTPv3) developed by D. Mills [7, 5, 6, 8, 9]. The NTP is a general purpose clock distribution protocol that provides clock synchronization to arbitrary computer networks. For our purposes we can divide the NTP into 4 blocks. A transmit process, a receive process, an update procedure and a local clock process. The update process generates the best offset available with respect to a selected reference clock by performing clock selection and combination. The input to the local clock part is the data generated by the update process. The local clock part of the NTPv3 ajusts the phase offset and frequency offset of the local clock with a PLL which removes input data noise and provides a stable timescale between network updates.

The NTPv3 outputs three quantities: a clock offset with respect to a selected reference clock, an offset error (dispersion) and the current round-trip delay to this reference clock. The NTPv3 also computes the so-called synchronization distance which defines the maximum error under any operational condition. The dispersion is the error expected under nominal operating conditions and has the following components [8]:

1. measurement errors due to inherent clock reading errors, also known as timestamping errors and due to the clock resolution and the method of adjustment.
2. offset accumulated since the last offset update due to the frequency offset[1]. The NTPv3 uses the maximum nominal frequency offset to compute this quantity.
3. errors due to network delay jitter, called the peer dispersion.
4. errors due to the offset sample variance which arise when several servers are used to synchronize a client.

Normally, and especially in WANs, delay jitter errors dominate the dispersion budget. The NTPv3 uses a type II PLL to minimize the delay jitter errors. PLLs have the problem that even if they perform "well" it is hard to quantify how much "better" one could do, *i.e.*, there

[1] frequency offset refers here to the first derivative of the time offset with respect to time.

is no optimality criteria available. While the current performance of the PLL is acceptable for today's needs, it may not be good enough to reach sub-millisecond synchronization in the future. There is a need to improve the error performance of time synchronization algorithms through better timestamping accuracy and better data processing.

In this document we design a Kalman filter that provides the phase (time offset) and the frequency offset of the local clock. Kalman filters have been used to model cesium beam atomic clocks in [10]. A Kalman filter presents several advantages with respect to a PLL:

1. It is optimal in the sense that it minimizes the mean square errors.
2. It provides a probabilistic framework for the synchronization of computer clocks which allows easily used robust statistical techniques and gives a statistical meaning to some quantities involved.
3. It provides the time and frequency offset estimates in a single loop.

We have tested the filter by running it with NTP timestamps. In both cases, in a client-server model, the process is divided into 3 parts:

1. A packet transmission and reception part that generates NTP timestamps.
2. A clock filter which selects the best sample.
3. Local clock model part which is based on the Kalman filter.

When the client has timing information from several time servers, a Kalman filter has to run for every client-server pair. In this case, clock selection will be performed after the clock filters and clock and frequency combination are performed after Kalman filtering. Our tests do not perform clock compensation for inherent frequency errors between network samples however the Kalman filter provides all the data required to perform such an operation. The local clock time offset, the local clock frequency offset and their corresponding error estimates are computed with the Kalman filter. The filter also provides the time till the next packet. The local clock time offset, the local clock frequency offset and the inter-packet time can be used in a local clock slewing routine. In table 1.1 we list the main features of the PLL in the NTPv3 and the Kalman filter. The Kalman filter requires floating point arithmetic although it may be possible to avoid this requirement.

This report is divided as follows. In section 2 we present an overview of the system designed. In section 3 we model the clock offset by an evolution equation. Section 4 describe how to estimate the system parameters. Section 6 is the main section of the paper and contains the design of the Kalman filter. In section 7 we investigate some implications of the use of Kalman filters in time synchronization. Section 8 contains experimental results and section 9 the conclusions.

	NTPv3's PLL	Kalman Filter
Floating Point Arithmetic	NO	YES
Optimization Criteria	NO	MSE
Time Offset Estimation	YES	YES
Frequency Offset Estimation	NO	YES
Time Offset accumulation	YES	YES
Error accumulation	NO	YES

Table 1.1: NTPv3's PLL versus the Kalman filter

2. Overview

We adopt the following terminology. The machine that sources the packets is called the client. The machine that receives and returns the packets is called the (time) server. We assume that the client wants to be synchronized to the server.

The data collected (timestamps) is processed using a Kalman filter. The output of this processing is a time offset estimate and a frequency offset estimate between the client's clock and the server's clock as well as their corresponding error estimates.

2.1 Timestamp Generation

In this section we briefly describe the timestamp generation process. More details can be found in the NTPv3 [7] specification. Figure 2.1 shows the timestamping sequence for the n-th packet sent from the client to the server. The n-th packet is timestamped and transmitted by the client. This time stamp is denoted by T_n. When it is received at the server a time stamp R'_n is written on the packet and when the packet is sent back to the client another timestamp S'_n is written again on the packet. Finally, when the packet arrives back to the client, it is timestamped again. This time stamp is denoted by A_n. Thus, after the reception of n-th packet, four timestamps T_n, R'_n, S'_n and A_n, are available at the client. Times R'_n and S'_n at the server respectively correspond to times R_n and S_n at the client so that, if $X(t)$ is the clock offset at *client time* t between the client and the server,

$$R_n = R'_n + X(R_n) \quad (2.1)$$

$$S_n = S'_n + X(S_n) \quad (2.2)$$

Similarly, times A_n and T_n at the client correspond to times A'_n and T'_n at the server. In current UNIX workstations, kernel timestamps will be accurate to a few tens of a microsecond whereas user space timestamps will be accurate to the millisecond.

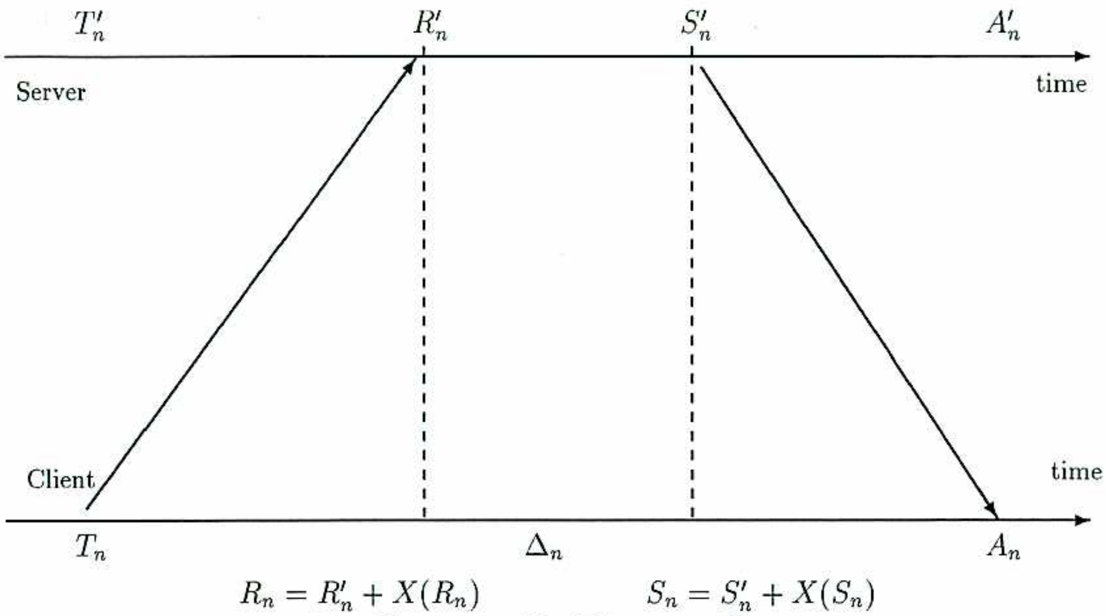

Figure 2.1: The Timestamp Sequence

For the n-th packet, we define the quantities,

$$d_n = \frac{(A_n - S'_n) + (R'_n - T_n)}{2}, \tag{2.3}$$

$$\Theta_n = \frac{(A_n - S'_n) - (R'_n - T_n)}{2}. \tag{2.4}$$

$$\Delta_n = \frac{A_n + T_n}{2}. \tag{2.5}$$

The quantities d_n, Θ_n and Δ_n can be directly computed from the timestamps and constitute the input data that a packet brings to evaluate the clock offsets. The quantity Θ_n will be called the *packet offset* and the quantity d_n will be referred to as the *half round trip delay.*

In section 3, we show that the larger the value of d_n the noisier is the offset information provided by the n-th packet. In order to reduce the noise, *i.e.*, the half round trip delay d_n, we send bursts of a few packets about one second apart. The first packet in the burst acts as a "warning shot" and has the effect of moving the server address in the cache of the routers. As a result, subsequent packets end up with a smaller half round trip delay. For every burst, we select the packet with the smallest d_n. This technique is also used in the NTPv3 (ntpdate).

2.2 The Clock Filter

The transmission part sends a burst of B packets to the destination. The clock filter selects the best sample in a burst which is the one with smallest half round-trip delay. The triple

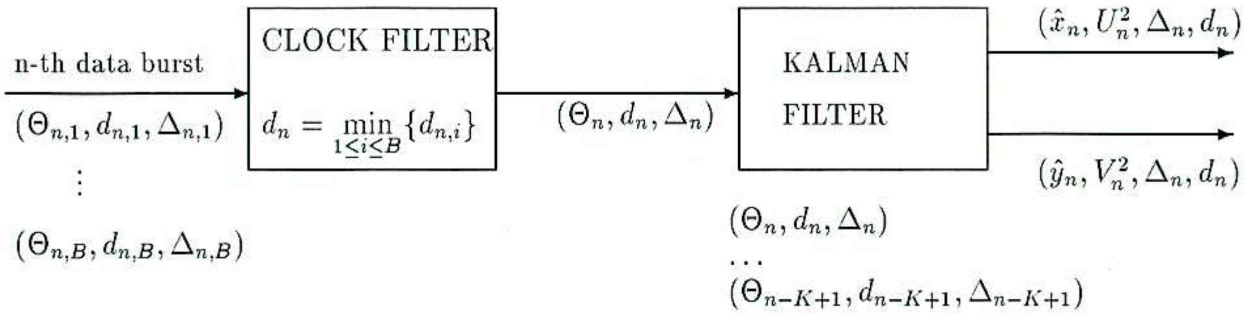

Figure 2.2: Client-Server Module

$(\Theta_n, d_n, \Delta_n)$ is then passed to the Kalman filter for time and frequency offset estimation.

The clock filter described here is very different from the NTPv3's clock filter, where the best of the last NTP.SHIFT (8) samples is selected. Frequency offset estimation is simpler if the time ordering of the input data is preseved. The NTPv3 clock filter can potentially complicate frequency offset estimation because input data ordering is not guaranteed and furthermore, one can select the same sample several times which would totally prevent frequency offset estimation.

2.3 Data Processing

The data processing block takes as input a triple $(\Theta_n, d_n, \Delta_n)$ and by means of a Kalman filter outputs,

1. A time offset estimate.
2. A frequency offset estimate.
3. The mean square error of the time offset estimate.
4. The mean square error of the frequency offset estimate.
5. The inter-burst time.

The system blocks for a client-server pair are shown in figure 2.2.

3. The Clock Offset Model

Let,

$$\Delta_n = (A_n + T_n)/2, \tag{3.1}$$
$$t_n = \Delta_{n+1} - \Delta_n. \tag{3.2}$$

In modern computers, the frequency offset y_n between two clocks increases at a constant speed D over time. The constant D is called the frequency offset drift. The frequency offset evolution equation is completed by a noise component W'_n which captures the frequency noise due to short term frequency offset fluctuations between times Δ_n and Δ_{n+1}. The linear increase over time in the frequency offset implies a quadratic increase over time for the time offset x_n between two clocks. At time Δ_n the frequency offset y_n and the time offset x_n satisfy the evolution equations,

$$y_{n+1} = y_n + t_n D + W'_n, \tag{3.3}$$

$$x_{n+1} = x_n + t_n y_n + \frac{t_n^2}{2} D + t_n W'_n. \tag{3.4}$$

The initial values of these evolution equations will be specified in section 6. We assume that the sequence $(W'_n)_{n\geq 0}$ is independent, with zero mean and $\text{Var}(W'_n) = \epsilon^2 + t_n \nu^2$. The term ϵ^2 describes the flicker frequency noise and the term ν^2 the random walk frequency noise [1].

Assumptions:

- *frequency offset:* The frequency offset per unit time is assumed at most of the order of $10^{(-3)}$, *i.e.*, ,

$$|y_n| \ll 1, \quad t \geq 0.$$

- *frequency drift:* The frequency drift D is constant.
- *symmetry assumptions:*
  - *symmetric protocol stack processing:* The value of the assymetry can only be evaluated if detailed information on the hardware architecture of the systems involved is available.
  - *routing symmetry:* In the average, the client-server delay and the server-client delay are equal, *i.e.*,

$$\lim_{m\to\infty} \frac{\sum_{n=0}^{m} (A_n - S_n) - (R_n - T_n)}{m} = 0.$$

  In practice the symmetry assumptions do not hold. They introduce a systematic bias in the offset estimates.

After some elementary algebra, we have,

$$\Theta_n = x_n + W_n, \tag{3.5}$$

where, taking the above assumptions into account, the random quantity W_n is given by,

$$W_n = \frac{(A_n - S_n) - (R_n - T_n)}{2}, \quad n \geq 1. \tag{3.6}$$

The packet offset Θ_n is a noisy time offset at time Δ_n, the noise level being equal to W_n. The quantity W_n will be referred to as the measurement or phase noise. As a result of the symmetry assumption, the random variable W_n has zero mean. Its variance is denoted by σ^2, *i.e.*,

$$\sigma^2 = \text{Var}(W_n). \tag{3.7}$$

If the symmetry assumptions fail, then W_n is not a zero mean random variable and the offset estimate will be systematically biased. The systematic bias due to processing assymetries is often below the accuracy requirements and can be quantified if the characteristics of the hosts involved are known or can be avoided by timestamping at the MAC level. The systematic bias due to routing assymetries is beyond our scope and may be important.

4. System Parameter Estimation

The system parameters estimation algorithms and the filter design have been tested on six data sets, covering both WAN and LAN environments. In both cases (LAN and WAN), we collected NTP timestamps by running at fixed intervals `ntpdate` between a standalone machine, `viper.hpl`, running as a client and known NTPv3 time servers. In the WAN environment (HP internet) the time servers were `andover.ntp`, `paloalto.ntp` and `cupertino.ntp` and in the LAN environment `netman.hpl` and `sysman.hpl`. Since the client remains unchanged, each data set is named after the server.

The estimation of the noise variances and drift is based on the Allan variance. The Allan variance is described in [8, 9, 1]. We fix the interburst time t_n to a constant value, say t and define,

$$\Theta'_n = \frac{\Theta_{n+1} - \Theta_n}{t} = y_n + \frac{t}{2}D + W'_n + \frac{W_{n+1} - W_n}{t}, \tag{4.1}$$

and

$$\Theta'_{n+1} - \Theta'_n = tD + W'_{n+1} + \frac{W_{n+2} - 2W_{n+1} + W_n}{t}. \tag{4.2}$$

The Allan variance is defined by,

$$\Sigma^2(t) = \frac{\text{Var}(\Theta'_{n+1} - \Theta'_n)}{2} \tag{4.3}$$

$$= \frac{\epsilon^2}{2} + \frac{t\nu^2}{2} + \frac{3\sigma^2}{t^2}, \tag{4.4}$$

and the Allan deviation is $\Sigma(t)$. We have observed in all data sets that the empirical Allan deviation starts with a slope of -1 and eventually ends with a slope of 0.5 in a log-log plot. This behavior is in accordance with equation (4.4).

Time Server	`netman`	`sysman`	`andover`	`colorado`	`cupertino`
duration (days)	38	38	35	38	38
σ (msecs)	0.33	0.42	6.72	6.68	6.82
ν (ppm)	0.002	0.002	0.002	0.002	0.002
ϵ (ppm)	0.52	0.49	0.56	0.60	0.57
D (ppb)	-0.001	-0.001	0.000	-0.000	-0.000
$\mathsf{E}[d_n]$ (msecs)	0.39	0.38	135.47	110.44	97.22
hops	0	0	18	16	14
empirical σ (msecs)	0.39	0.38	7.13	6.50	6.48

Table 4.2: Parameter Estimates

4.1 Algorithms

From equation (4.2), we see that the drift is estimated by,

$$\hat{D} = \frac{\mathsf{E}[\Theta'_{n+1} - \Theta'_n]}{t}, \tag{4.5}$$

and

$$\mathsf{Var}(\hat{D}) = \frac{2\Sigma^2(t)}{t^2}.$$

The estimation algorithm for σ, ν and ϵ starts by estimating $\Sigma^2()$ as a function of the interpacket time t from the empirical values of $(\Theta_n)_{n\geq 1}$. The first point is obtained by setting t equal to the minimum interburst time, currently 16secs. Further points are obtained for $2t, 4t, \ldots$.

The value of the phase or measurement noise deviation σ is obtained by best fitting the function $3\sigma^2/t^2$ to the empirical Allan variance. The value of the frequency noise components ν and ϵ are obtained by best fitting the function $\sigma^2/2+t\nu^2/2$ to the empirical Allan variance. which provides the minimum variance for $\hat{D}$. Table 4.2 shows the parameters for each trace, and figures 4.3, 4.4, 4.6, 4.5 and 4.7 show the plots of the empirical Allan deviation and the estimated Allan deviation for all the traces considered.

4.2 Convergence

We have checked the convergence of the algorithms described in section 4.1 for each trace. For a trace of duration T we have computed estimates of σ, ν and ϵ for estimation periods of length $T, T/2, T/4, \ldots$. Thus, when the estimation period is T/k the algorithm of section 4.1 provides k estimates for σ, ν and ϵ

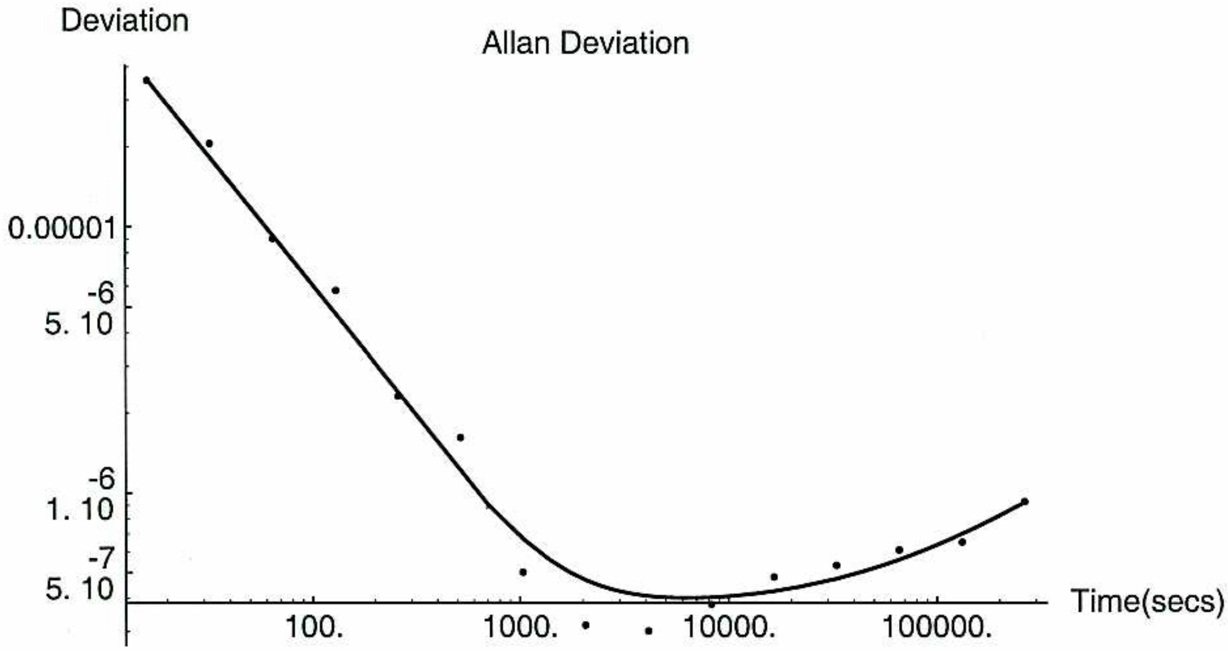

Figure 4.3: Allan Deviation for netman.hpl

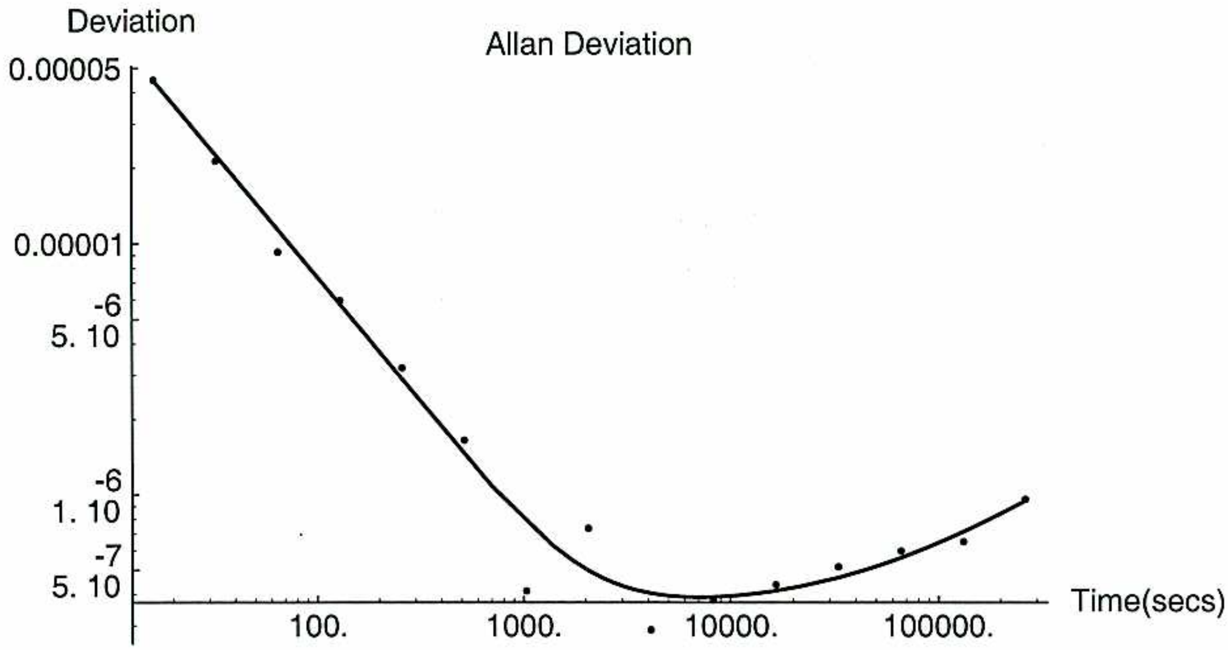

Figure 4.4: Allan Deviation for `sysman.hpl`

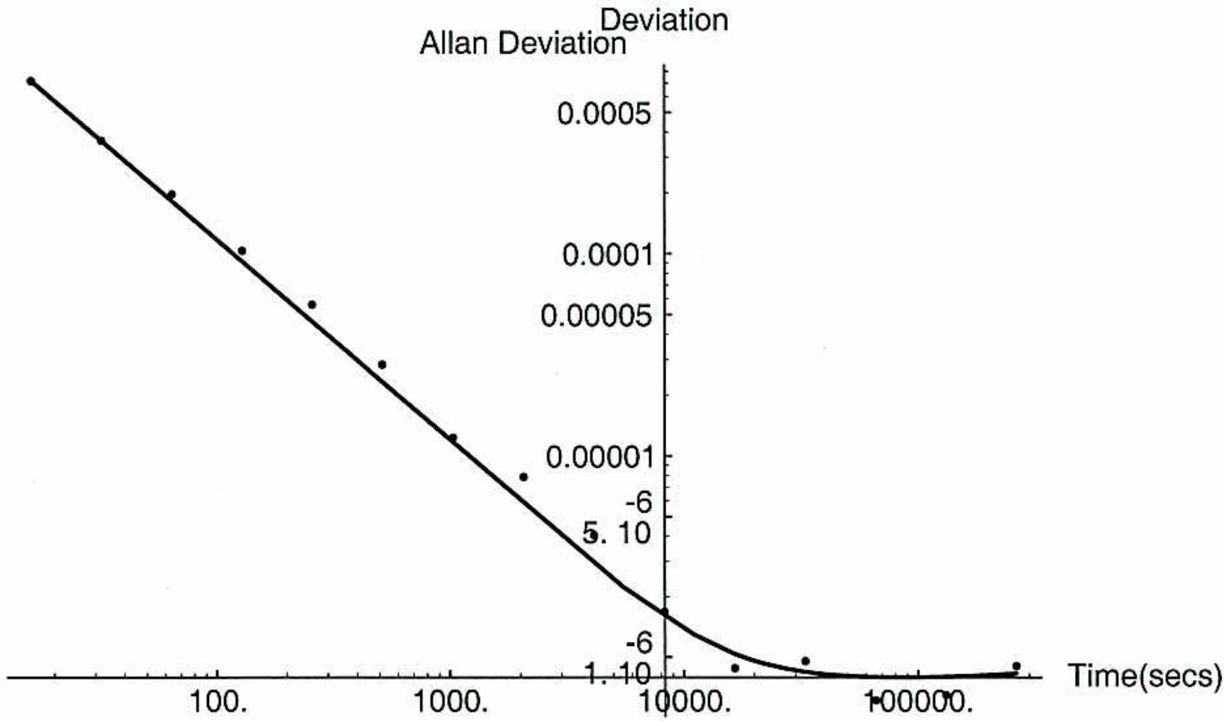

Figure 4.5: Allan Deviation for `colorado.ntp`

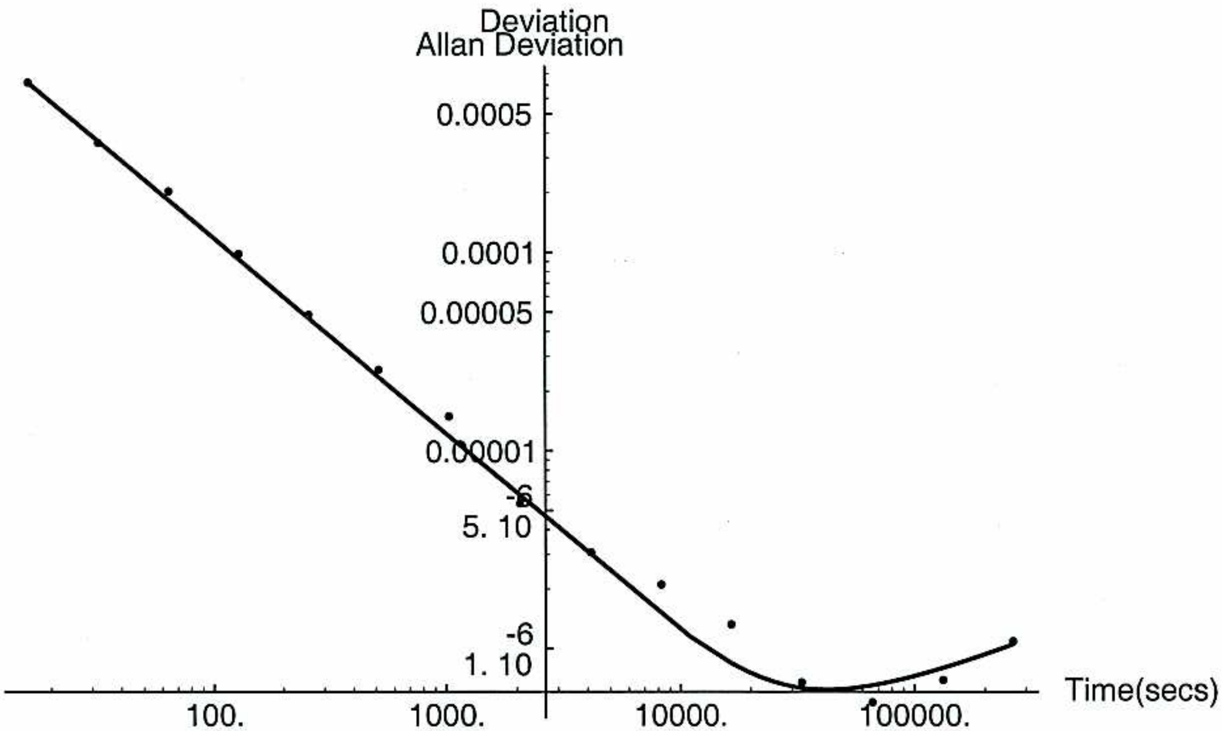

Figure 4.6: Allan Deviation for `andover.ntp`

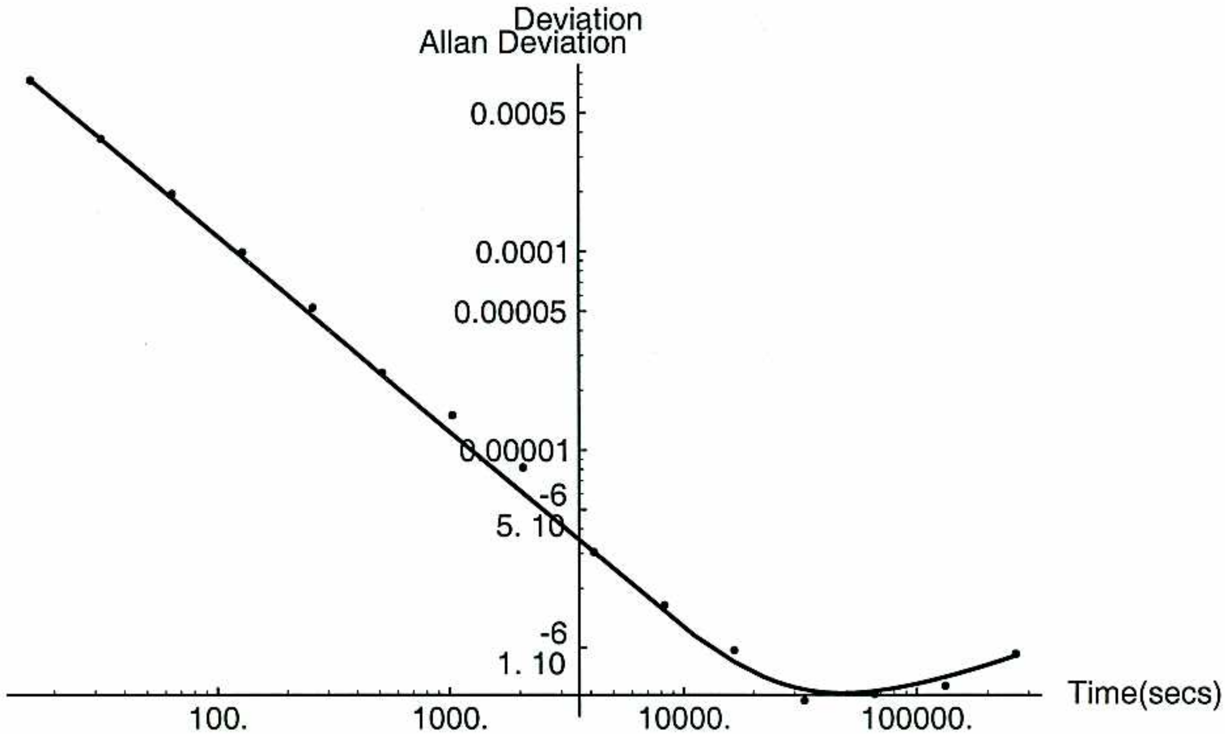

Figure 4.7: Allan Deviation for `cupertino.ntp`

We decided that the algorithm had converged for the estimation period of length T/k when the following two conditions were met:

1. The distance between the estimate obtained for the trace of length T and the mean of the k estimates obtained for estimation periods of length T/k was less than the standard deviation of the k estimates obtained for estimation periods of length T/k.

2. The standard deviation of the k estimates obtained for estimation periods of length T/k is an order of magnitude below the mean of the k estimates obtained for estimation periods of length T/k.

The convergence results show that:

1. Convergence in LAN seems slower. This may be due to the fact that timestamping errors are probably comparable to the values to be estimated.

2. Even when convergence is achieved with the algorithms proposed, the time it takes to achieve convergence is impractical for real time implementations. So, while the algorithm in section 4.1 and the convergence criteria can be used to compute the true values of σ, ν and ϵ, empirical approaches are needed to estimate σ, ν and ϵ in real time.

3. Convergence for the phase noise variance occurs earlier than convergence for the frequency noise variance.

In figures 4.8, 4.8 we show the Allan deviation convergence for traces of T, T/2 and T/4 where T is the overall trace duration for `sysman.hpl` and `colorado.ntp`. Thus, there are 4 plots that last till time T/4, 2 plots that last till time T/2 and one plot that lasts till time T. We can see in all cases how the deviation plots remain close until about 100000secs.

5. Empirical Estimates

If X_n and Y_n respectively denote the inbound and the outbound delays of the n-th packet, we have,

$$\begin{aligned} d_n &= \frac{X_n + Y_n}{2}, \\ W_n &= \frac{X_n - Y_n}{2}. \end{aligned}$$

Simple computations show that if X_n and Y_n are uncorrelated and if both $\mathsf{Var}(d_n)$ and $\mathsf{Var}(W_n)$ exist, then,

$$\mathsf{Var}(W_n) = \mathsf{Var}(d_n).$$

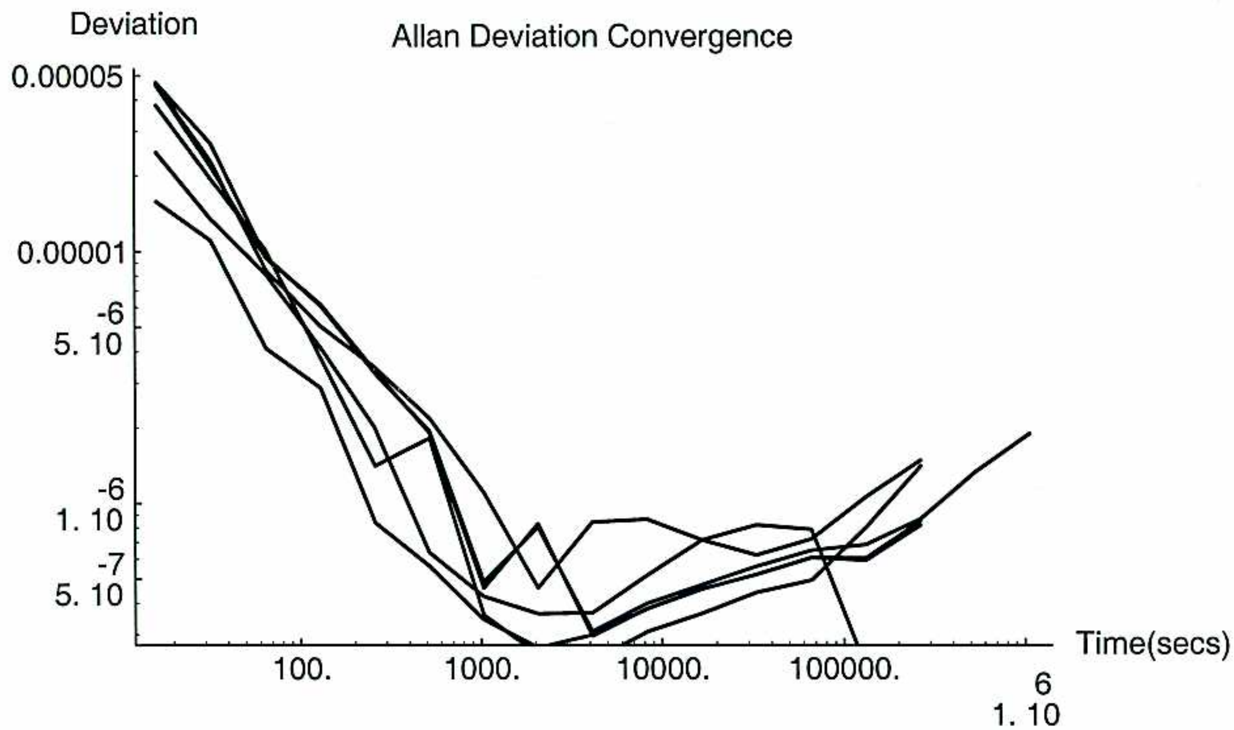

Figure 4.8: Allan Deviation convergence for `sysman.hpl`

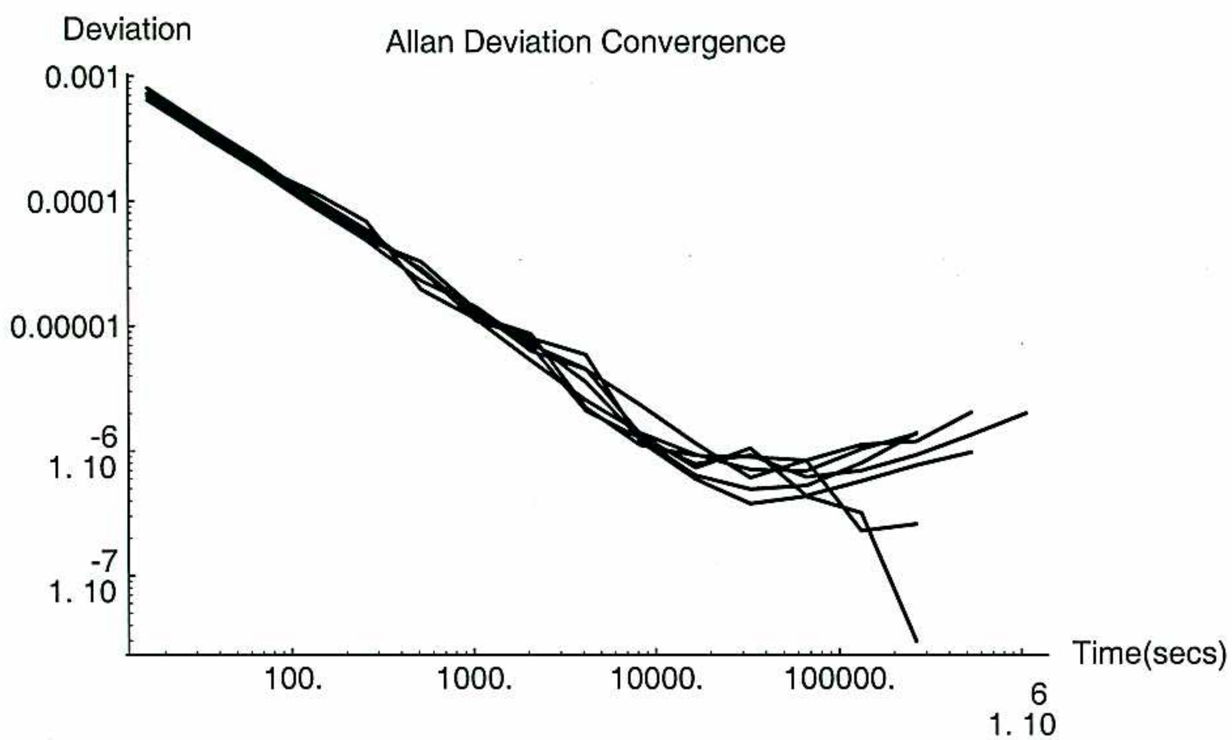

Figure 4.9: Allan Deviation convergence for `colorado.ntp`

Unfortunately, the empirical data seems to indicate that $\mathrm{Var}(d_n)$ does not exist, and thus this method cannot be used.

5.1 Phase Noise Empirical Characterization

We have observed that when σ is well approximated by the average half-round trip delay in LAN cases and in WAN cases, we have,

$$\sigma \approx \frac{d}{1+h} \tag{5.1}$$

where d is the average half-round trip delay and h is the number of routers the sync packets cross from the client and back. These empirical estimates need further testing.

5.2 Frequency Noise Empirical Characterization

The noise frequency parameters seem to be independent of the time server considered. The flicker noise standard deviation seems to be about 0.55ppm and the random walk noise standard deviation about 0.002ppm. We do not know how general this situation is and further tests are needed to verify this situation. Another alternative is to develop algorithms that can estimate in real time these parameters. This, however seems to be not trivial because the noise frequency parameters only become observable at times higher than 1000secs.

6. Kalman Filtering

In this section we design a Kalman filter to estimate the time and frequency offset and their errors. The system equations are,

$$\begin{aligned} \Theta_n &= x_n + W_n, \quad n \geq 0, & (6.1)\\ y_{n+1} &= y_n + t_n D + W'_n, & (6.2)\\ x_{n+1} &= x_n + t_n y_n + \frac{t_n^2}{2} D + t_n W'. & (6.3) \end{aligned}$$

The Kalman filter will provide a recursive method to compute the time offset estimates,

$$\hat{x}_n = \mathbb{E}[x_n | \Theta_0, \ldots, \Theta_n], \quad n \geq 0,$$

the clock frequency offset estimate,

$$\hat{y}_n = \mathsf{E}[y_n | \Theta_0, \ldots, \Theta_n], \quad n \geq 0,$$

and their respective mean square errors and cross-correlations,

$$\begin{aligned} U_n^2 &= \mathsf{E}[(x_n - \hat{x}_n)^2], \quad n \geq 0 \\ V_n^2 &= \mathsf{E}[(y_n - \hat{y}_n)^2], \quad n \geq 0 \\ \gamma_n^2 &= \mathsf{E}[(y_n - \hat{y}_n)(x_n - \hat{x}_n)] \quad n \geq 0. \end{aligned}$$

Kalman filters provide the minimum mean square error estimates when the observations are perturbed by white noise and the system evolution is also driven by white noise. If the white noise is gaussian, Kalman filters are optimum in the mean square error sense. When the white noise is not gaussian, Kalman filters are the best linear filters in the mean square error sense. For further details on Kalman filters, the reader is referred to [3, 2].

6.1 Filter Design

Unfortunately, a straightforward application of Kalman filtering results to the design of a Kalman filter for time and frequency offset estimation is not possible because the system parameters ν, ϵ, σ and D are not known. This is circumvented by estimating the parameters using the empirical estimates proposed in section 5. To summarize we will implement a Kalman filter under the following assumptions,

1. $(W_n)_{n\geq 0}$ is a sequence of independent zero mean random variables and variance σ^2 where $\sigma \approx d/(1+h)$ with $d = \mathsf{E}[d_n]$ and h number of routers crossed.
2. $(W'_n)_{n\geq 0}$ is a sequence of independent zero mean random variables and $\mathsf{Var}(W'_n) = \epsilon^2 + t_n\nu^2$ where $\nu \approx 0.002$ppm and $\epsilon \approx 0.55$ ppm.
3. x_0 is a random variable of mean Θ_0 and variance σ^2.
4. The random variable sequences $(W_n)_{n\geq 0}$ and $(W'_n)_{n\geq 0}$ are independent of each other.

6.2 Kalman Equations

We will design a filter that simultaneously estimates time and frequency offset. Let,

$$\boldsymbol{Z}_n = (x_n, y_n)^T,$$

where T is the transposition operator, The system equations (6.1), (6.2) and (6.3) become,

$$\Theta_n = \boldsymbol{h}_1^T \boldsymbol{Z}_n + W_n, \quad n \geq 0, \tag{6.4}$$

$$\boldsymbol{Z}_{n+1} = \boldsymbol{F}_n \boldsymbol{Z}_n + \boldsymbol{\Gamma}_n + \boldsymbol{W}'_n \quad n \geq 1. \tag{6.5}$$

where, $\boldsymbol{h}_1$ is the 2 column vector,

$$\boldsymbol{h}_1 = (1,0)^T,$$

$\boldsymbol{F}_n$ is the 2×2 matrix,

$$\boldsymbol{F}_n = \begin{pmatrix} 1 & t_n \\ 0 & 1 \end{pmatrix},$$

$\boldsymbol{\Gamma}_n$ is the vector,

$$\boldsymbol{\Gamma}_n = (D\frac{t_n^2}{2}, Dt_n)^T,$$

and $\boldsymbol{W}'_n = (t_n W'_n, W'_n)^T$.

If $\hat{\boldsymbol{Z}}_n = \mathsf{E}[\boldsymbol{Z}_n|\Theta_0,\ldots,\Theta_n]$, denotes the estimate of $\boldsymbol{Z}_n$ given $\Theta_0,\ldots,\Theta_n$, and the matrix $\boldsymbol{C}_n$ denotes the 2×2 covariance matrix of $\boldsymbol{Z}_n$ *i.e.*,

$$\boldsymbol{C}_n = \mathsf{E}[(\boldsymbol{Z}_n - \hat{\boldsymbol{Z}}_n)(\boldsymbol{Z}_n - \hat{\boldsymbol{Z}}_n)^T],$$

then the theory of Kalman filtering provides formulas for the updates of $\boldsymbol{C}_n$ and $\hat{\boldsymbol{Z}}_n$ at time Δ_{n+1}. These updates are as follows:

$$\hat{\boldsymbol{Z}}_{n+1} = \boldsymbol{F}_n\hat{\boldsymbol{Z}}_n + \boldsymbol{\Gamma}_n + \frac{\left(\boldsymbol{F}_n\boldsymbol{C}_n\boldsymbol{F}_n^T + \boldsymbol{Q}_n\right)\boldsymbol{h}_1}{t_n^3\nu^2 + t_n^2\epsilon^2 + t_n^2V_n^2 + 2t_n\gamma_n + U_n^2 + \sigma^2}\left(\Theta_{n+1} - \boldsymbol{h}_1^T(\boldsymbol{F}_n\hat{\boldsymbol{Z}}_n + \boldsymbol{\Gamma}_n)\right), \tag{6.6}$$

$$\boldsymbol{C}_{n+1} = \left(\boldsymbol{F}_n\boldsymbol{C}_n\boldsymbol{F}_n^T + \boldsymbol{Q}_n\right) - \frac{\left(\boldsymbol{F}_n\boldsymbol{C}_n\boldsymbol{F}_n^T + \boldsymbol{Q}_n\right)\boldsymbol{h}_1\boldsymbol{h}_1^T\left(\boldsymbol{F}_n\boldsymbol{C}_n\boldsymbol{F}_n^T + \boldsymbol{Q}_n\right)}{t_n^3\nu^2 + t_n^2\epsilon^2 + t_n^2V_n^2 + 2t_n\gamma_n + U_n^2 + \sigma^2}, \tag{6.7}$$

where,

$$\boldsymbol{Q}_n = (\epsilon^2 + t_n\nu^2)\begin{pmatrix} t_n^2 & t_n \\ t_n & 1 \end{pmatrix}, \tag{6.8}$$

$$U_n^2 = \boldsymbol{C}_n(1,1), \tag{6.9}$$

$$V_n^2 = \boldsymbol{C}_n(2,2), \tag{6.10}$$

$$\gamma_n = \boldsymbol{C}_n(2,1) = \boldsymbol{C}_n(1,2), \tag{6.11}$$

and the initial values are set after sending the first 2 packets as,

$$\hat{\boldsymbol{Z}}_1 = (\Theta_1, \Theta'_0)^T, \tag{6.12}$$

$$\boldsymbol{C}_0 = \sigma^2\begin{pmatrix} t_0^2 & t_0 \\ t_0 & 1 \end{pmatrix}, \tag{6.13}$$

$$D = 0. \tag{6.14}$$

The Kalman recursions in scalar form are in appendix A.

6.3 Predicted Offset Errors

The theory of Kalman filtering provides estimates for the time and frequency offset errors t seconds after the n-th packet burst received. The correlation matrix of the time and frequency offset system t seconds after the n-th packet burst received is,

$$\boldsymbol{C}(t) = \boldsymbol{F}(t)\boldsymbol{C}_n\boldsymbol{F}(t)^T + \boldsymbol{Q}(t),$$

where,

$$\begin{aligned} \boldsymbol{F}(t) &= \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix}, \\ \boldsymbol{Q}(t) &= (\epsilon^2 + t\nu^2)\begin{pmatrix} t^2 & t \\ t & 1 \end{pmatrix}. \end{aligned}$$

In particular, the time offset error is given by,

$$U^2(t) = t^3\nu^2 + t^2\epsilon^2 + t^2V_n^2 + 2t\gamma_n + U_n^2.$$

and the frequency offset error is given by,

$$V^2(t) = \nu^2 t + \epsilon^2 + V_n^2.$$

6.4 Inter Burst Time

The Allan variance plots show that the time offset error will be minimized for small inter burst times whereas the frequency offset error will be minimized for times close to the time that minimizes the Allan variance, *i.e.*,

$$t = \Big(\frac{12\sigma^2}{\nu^2}\Big)^{1/3}.$$

These two goals are incompatible. If one is interested in providing a stable time scale, large interpacket times should be used. If one is interested in very accurate synchronization at a given instant, small interpacket times should be used.

6.4.1 Bounded Time Offset Error

The predicted time offset error can be used to select the inter burst time to bound the time offset error. By choosing $t_n = \Delta_{n+1} - \Delta_n$ as the minimal positive solution of,

$$t^3\nu^2 + t^2\epsilon^2 + t^2V_n^2 + 2t\gamma_n - \alpha^2U_n^2 = 0,$$

we guarantee that the squared time offset error will only increase to $(1+\alpha^2)U_n^2$ between two packets. This inter burst time selection method will be called the *error accumulation criteria.*

6.4.2 Bounded Time Offset

One can also select the inter burst time to bound the time offset increase between packet bursts by sending the next burst when the estimated time offset increases by a preset value τ. The interpacket time is then,

$$t_n = \frac{\tau}{|y_n|}.$$

This inter burst time selection method will be called the *time offset accumulation criteria.* This criteria is only well defined when the effects of local clock ajustments between packet bursts are ignored.

7. Clock Synchronization with Kalman Filtering

In this section we explore how some of the algorithms used by the NTPv3 to perform time synchronization could be modified if Kalman filters where used.

7.1 Clock Selection and Combination

Until now we have only addressed a client-server situation. When a client receives timing information with respect to a reference clock from several time servers, a filter must exist for every client-server pair. The issue of clock selection and combination arises.

The first step in clock selection is to identify a set of reliable and consistent clocks. The NTPv3 provides efficient and widely tested algorithms that perform that task. The second step is to rank the reliable clocks from better to worse. The NTPv3 uses the so-called synchronization distance to rank the clocks [6]. Our previous analysis shows that, in a Kalman filter setting, the natural metric to rank clock is the euclidean distance based on the estimated variances.

Clock combination consists in taking as the client time offset with respect to the reference clock some average of the selected servers' offsets with respect to the reference clock. The Kalman filtering setting provides a natural way of selecting this average.

If a client has receives timing information from S servers, all synchronized with respect to the same reference clock and all equally consistent and reliable, it is natural to try to use all the information available to derive the time offset estimate of the client with respect to this reference clock.

The time offset[2] x_i with respect to the ith server is a random variable of mean $\hat{x}_i$ and variance U_i, where $\hat{x}_i$ and U_i are respectively the current time offset estimate and the current time offset error obtained from the Kalman filter associated to the ith server.

The client's offset x with respect to the reference clock is estimated by $\hat{x} = \mathsf{E}[x]$ and its error by $U^2 = \mathsf{Var}(x)$. The time offset x is obtained as an average of the time offsets of each active client-server pair. The client time offset x and frequency offset with respect to the reference clock are of the form,

$$\begin{aligned} x &= p_1 x_1 + \ldots + p_S x_S, \\ y &= q_1 y_1 + \ldots + q_S y_S, \end{aligned}$$

with $p_i \geq 0$, $1 \leq i \leq S$, and $p_1 + \ldots + p_S = 1$, and $q_i \geq 0$, $1 \leq i \leq S$, and $q_1 + \ldots + q_S = 1$. Then,

$$\begin{aligned} \hat{x} &= p_1 \hat{x}_1 + \ldots + p_S \hat{x}_S, \\ \hat{y} &= q_1 \hat{y}_1 + \ldots + q_S \hat{y}_S, \end{aligned}$$

and, since x_i and x_j are independent for $i \neq j$, and similarly for the frequencies,

$$\begin{aligned} U^2 &= p_1^2 U_1^2 + \ldots + p_S^2 U_S^2, \\ V^2 &= q_1^2 V_1^2 + \ldots + q_S^2 V_S^2, \end{aligned}$$

The values $(p_i)_{i \leq i \leq S}$ are selected to minimize the time offset error U^2. They are given by[3],

$$p_i = \frac{\prod_{k \neq i} U_k^2}{\sum_j \prod_{k \neq j} U_k^2}, \quad 1 \leq i \leq S,$$

so that, the variance is,

$$\frac{1}{U^2} = \sum_i \frac{1}{U_i^2}.$$

[2] In this section indices refer to client-server pairs rather than to time.

[3] S. Crouch, private communication

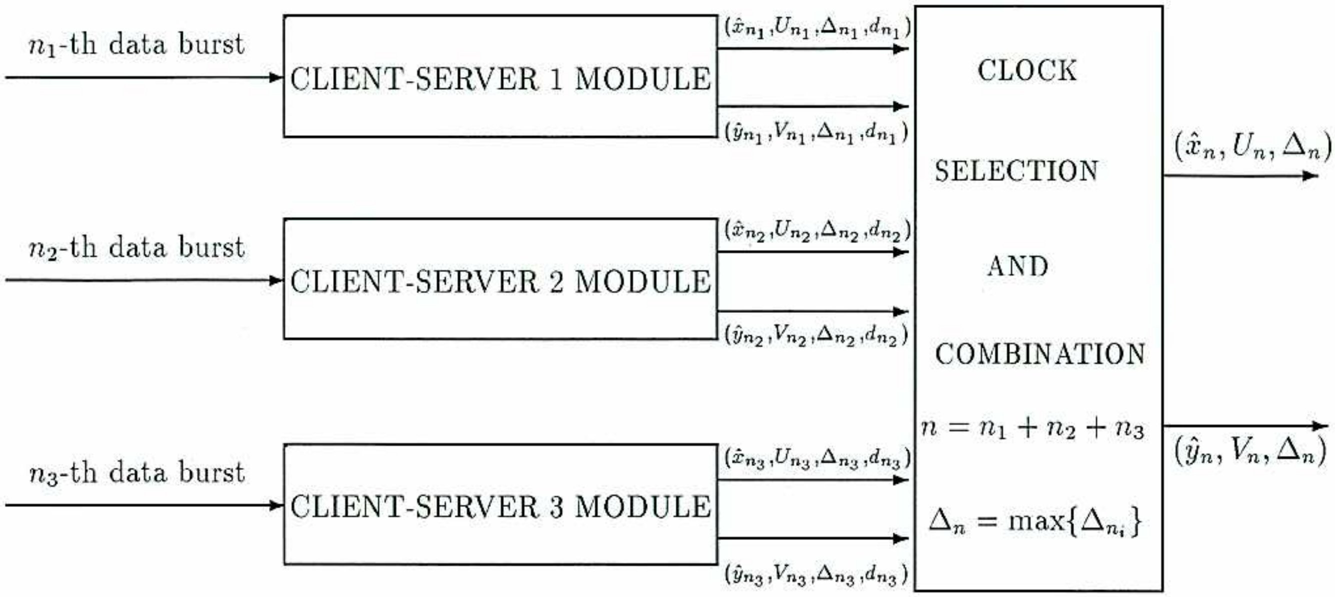

Figure 7.10: Clock Combination Architecture

Similarly, we obtain,

$$q_i = \frac{\prod_{k \neq i} V_k^2}{\sum_j \prod_{k \neq j} V_k^2}, \quad 1 \leq i \leq S,$$

so that, the variance is,

$$\frac{1}{V^2} = \sum_i \frac{1}{V_i^2}.$$

This shows that clock combination can achieve error reductions by factors of $1/\sqrt{S}$. In particular, $U \leq \min\{U_1, \ldots, U_S\}$, *i.e.*, there is a gain in using several time servers to estimate the time offset. Also, clock combination stabilizes both the time offset and its error by protecting against sudden error surges in one server.

The clock combination architecture is shown in figure 7.10. Clock selection occurs at the output of the client-server modules. The survivors are used for clock combination. Even if a server fails the clock selection test at some point, it is worth keeping the client-server exchange alive because past history will always help to improve time offset and frequency offset estimates.

8. Results

We have estimated the time and frequency offset for all the data sets using the Kalman filter designed in section 6.1 with the values for σ, ν and ϵ obtained in section 4. The filter ran

Parameter	Value	Description
MAX_SECS	4096	Maximum inter-burst time (secs)
MIN_SECS	16	Minimum inter-burst time (secs)
BSZ	3	Burst size

Table 8.3: List of Parameters

using both inter burst time criteria described in section 6.4. We analyze the data obtained from two angles. Firstly, we assess the filter error performance and stability and secondly we study the filter optimality. Table 8.3 gives a list of the more relevant parameters used in the tests.

8.1 Filter Performance and Stability

In order to assess the filter performance and stability we provide for each trace the following values:

1. The empirical mean interburst time $\overline{t}$.
2. The empirical mean of the time offset error $\overline{U}$ and its empirical standard deviation $\sqrt{\overline{(U-\overline{U})^2}}$.
3. The empirical mean of the frequency offset error $\overline{V}$ and its empirical standard deviation $\sqrt{\overline{(V-\overline{V})^2}}$.
4. The empirical mean frequency offset $\overline{y}$ and its empirical standard deviation $\sqrt{\overline{(y-\overline{y})^2}}$.

The filter performance is assessed by comparing the empirical mean errors obtained to the inherent data noise. Thus, the empirical time offset mean error $\overline{U}$ is compared to the corresponding noise quantity, σ and the empirical frequency offset mean error $\overline{V}$ is compared to the corresponding noise quantity, $\sqrt{\epsilon^2 + t\nu^2 + 2\sigma^2/t^2}$.

The filter stability is studied by comparing mean values with their standard deviations for relevant parameters. We will say that a quantity is stable when its empirical standard deviation is an order of magnitude below its empirical mean.

Finally, since the frequency drift should be very small, the frequency offset estimate should remain roughly constant between packet bursts, *i.e.*, the short term fluctuations of the frequency offset estimate should be very small. A simple way of testing this is by looking at the values of the correlation coefficients of the frequency offset estimate time series, their values should be very close to 1.

8.2 Filter Optimality

For the Kalman filter, it is known [2] that the random variables $(M_n)_{n\geq 1}$, where

$$M_{n+1} = \frac{\Theta_{n+1} - \left(\boldsymbol{h}_1^T(\boldsymbol{F}_n\hat{\boldsymbol{Z}}_n + \boldsymbol{\Gamma}_n)\right)}{\sqrt{t_n^3\nu^2 + t_n^2\epsilon^2 + t_n^2V_n^2 + 2t_n\gamma_n + U_n^2}}, \quad n \geq 0, \tag{8.1}$$

should be uncorrelated with zero mean and unit variance. These properties can be tested from the empirical data. The random sequence $(M_n)_{n\geq 1}$ can be evaluated from the measurements and hence one can compute empirical means and correlation parameters. If the empirical mean, variance and correlation of $(M_n)_{n\geq 1}$ are close to their theoretical values, one should expect that the Kalman filter design will be close to the best[4] linear filter to estimate the time and frequency offset[5].

Intuitively, an empirical mean close to zero and an empirical variance close to one show that the filter tracks well. Furthermore, a variance much less (resp. more) than the unity can be a sign of overestimated (resp. underestimated) the noise power. Finally, the correlation coefficients give a measure of how good is the filter at extracting the real information from the data.

The *a posteriory* checks are done by computing,

1. The empirical mean $\overline{M}$ of the sequence $(M_n)_{n\geq 1}$.
2. The empirical standard deviation $\sqrt{\overline{(M-\overline{M})^2}}$ of the sequence $(M_n)_{n\geq 1}$.
3. The first five autocorrelation coefficients $\rho_1, \ldots, \rho_5$ of the sequence $(M_n)_{n\geq 1}$,

8.3 Data Analysis

The results obtained are gathered in tables 8.5, 8.7, 8.4 and 8.6. Both in LAN and WAN environments the numerical results obtained show that the filter performs very close to its expected theoretical behavior. The numerical data also shows that the filter performance in WAN environments is better since the aposteriory checks are closer to the expected theoretical values. It is possible that this difference is due to the fact that in LAN environments we are within the limits of the timestamp accuracy since we are dealing with sub-millisecond errors.

[4] in the mean square error sense.

[5] If furthermore it ever turns out that the random variables M_n are normal then they will be independent and one should expect that the Kalman filter design will be close to the optimal filter to estimate the time and frequency offset. We have not found $(M_n)_{n\geq 1}$ to be normal.

server	netman			sysman		
α	1	3	5	1	3	5
$\bar{t}$(secs)	142	1066	2302	205	1432	3082
$\overline{U}$(msec)	0.23	0.31	0.32	0.30	0.39	0.41
$\sqrt{\overline{(U-\overline{U})^2}}$(msecs)	0.01	0.00	0.01	0.02	0.00	0.00
$\overline{V}$(ppm)	0.75	0.30	0.18	0.69	0.30	0.17
$\sqrt{\overline{(V-\overline{V})^2}}$(ppm)	0.33	0.00	0.00	0.16	0.41	0.00
$\overline{y}$(ppm)	-9.21	-9.20	-9.20	-9.21	-9.87	-9.21
$\sqrt{\overline{(y-\overline{y})^2}}$(ppm)	2.70	1.76	1.75	1.94	31.59	1.74
$\overline{M}$	0.00	-0.00	-0.00	0.00	-0.00	-0.00
$\sqrt{\overline{(M-\overline{M})^2}}$	0.62	0.62	0.77	0.58	0.66	0.71
ρ_1	0.11	-0.15	-0.05	-0.17	-0.02	0.22
ρ_2	-0.01	-0.03	-0.03	-0.01	-0.08	-0.02
ρ_3	-0.19	0.02	0.04	-0.05	-0.00	-0.14
ρ_4	-0.20	-0.01	0.01	-0.07	0.01	-0.10
ρ_5	-0.16	-0.01	-0.03	-0.06	-0.02	-0.06

Table 8.4: Average Values for LAN environment - Error accumulation

server	colorado			andover			cupertino		
α	0.5	0.75	1	0.5	0.75	1	0.5	0.75	1
$\bar{t}$(secs)	683	1499	2567	698	1831	2585	744	1607	2757
$\overline{U}$(msec)	3.64	4.27	4.72	3.67	4.29	4.75	3.73	4.36	4.82
$\sqrt{\overline{(U-\overline{U})^2}}$(msecs)	0.03	0.01	0.02	0.04	0.02	0.02	0.03	0.02	0.01
$\overline{V}$(ppm)	1.32	1.04	0.90	1.31	0.87	0.88	1.25	1.00	0.84
$\sqrt{\overline{(V-\overline{V})^2}}$(ppm)	0.27	0.32	0.73	0.31	0.39	0.29	0.32	0.46	0.41
$\overline{y}$(ppm)	-9.03	-9.28	-8.80	-9.18	-9.13	-9.17	-9.40	-9.51	-9.60
$\sqrt{\overline{(y-\overline{y})^2}}$(ppm)	6.07	2.36	7.88	2.54	2.55	2.17	4.04	4.92	5.92
$\overline{M}$	0.01	-0.01	0.01	0.00	0.01	0.01	-0.02	-0.03	-0.05
$\sqrt{\overline{(M-\overline{M})^2}}$	1.01	0.95	1.13	1.00	1.01	1.00	1.002	1.07	1.09
ρ_1	0.07	-0.04	-0.01	-0.01	-0.03	-0.06	0.06	-0.00	0.05
ρ_2	0.04	-0.10	-0.01	-0.07	-0.09	-0.07	-0.02	0.01	0.12
ρ_3	-0.01	-0.09	-0.09	-0.04	0.04	-0.08	-0.00	0.00	0.08
ρ_4	-0.03	-0.14	-0.01	-0.06	0.00	-0.05	-0.04	0.02	0.19
ρ_5	-0.02	-0.04	-0.04	-0.07	-0.09	-0.01	-0.00	0.02	0.09

Table 8.5: Average Values for WAN environment - Error accumulation

server	netman			sysman		
τ(msecs)	1	10	20	1	10	20
$\bar{t}$(secs)	107	1085	2173	107	1085	2071
$\overline{U}$(msec)	0.22	0.31	0.32	0.26	0.37	0.40
$\sqrt{\overline{(U-\overline{U})^2}}$(msecs)	0.01	0.02	0.00	0.01	0.00	0.00
$\overline{V}$(ppm)	0.85	0.37	0.19	0.88	0.34	0.22
$\sqrt{\overline{(V-\overline{V})^2}}$(ppm)	0.50	2.03	0.03	0.59	0.03	0.03
$\overline{y}$(ppm)	-9.54	-9.52	-9.52	-9.55	-9.54	-9.53
$\sqrt{\overline{(y-\overline{y})^2}}$(ppm)	2.08	2.61	1.69	2.75	1.68	1.68
$\overline{M}$	-0.00	-0.01	-0.01	-0.00	-0.00	-0.00
$\sqrt{\overline{(M-\overline{M})^2}}$	0.62	0.85	0.75	0.53	0.62	0.67
ρ_1	-0.21	-0.19	-0.10	-0.08	-0.09	0.04
ρ_2	-0.11	-0.15	0.03	-0.06	-0.11	0.01
ρ_3	0.03	-0.02	0.04	-0.01	-0.04	-0.05
ρ_4	0.13	0.01	0.00	0.14	0.03	-0.03
ρ_5	0.11	0.00	-0.01	0.05	0.01	-0.04

Table 8.6: Average Values for LAN environment - Time Offset accumulation

server	colorado			andover			cupertino		
τ(msecs)	1	10	20	1	10	20	1	10	20
$\bar{t}$(secs)	108	1088	2165	108	1085	2167	108	1080	2155
$\overline{U}$(msec)	2.40	3.99	4.56	2.40	4.00	4.58	2.40	4.01	4.59
$\sqrt{\overline{(U-\overline{U})^2}}$(msecs)	0.15	0.18	0.17	0.17	0.19	0.18	0.16	0.18	0.17
$\overline{V}$(ppm)	2.22	1.17	0.97	2.23	1.16	0.92	2.14	1.17	0.95
$\sqrt{\overline{(V-\overline{V})^2}}$(ppm)	0.29	0.65	0.96	0.35	0.23	0.01	0.34	0.95	0.79
$\overline{y}$(ppm)	-9.61	-9.44	-8.76	-9.65	-9.57	-9.59	-9.90	-10.60	-10.50
$\sqrt{\overline{(y-\overline{y})^2}}$(ppm)	3.41	5.01	12.97	3.05	2.07	1.58	6.70	15.87	12.86
$\overline{M}$	-0.01	-0.00	0.03	-0.00	-0.00	-0.00	-0.01	-0.05	-0.07
$\sqrt{\overline{(M-\overline{M})^2}}$	0.92	0.99	1.12	0.93	0.92	0.95	0.91	1.04	1.13
ρ_1	0.07	0.00	0.14	0.06	-0.04	-0.11	0.09	0.12	0.19
ρ_2	0.03	-0.06	0.12	0.04	-0.08	-0.09	0.05	0.07	0.20
ρ_3	0.04	-0.08	0.07	-0.02	-0.03	0.00	0.02	0.03	0.17
ρ_4	0.00	-0.07	0.10	-0.02	-0.08	-0.03	0.03	0.05	0.17
ρ_5	-0.02	-0.03	0.06	-0.02	-0.08	-0.09	0.00	0.14	0.24

Table 8.7: Average Values for WAN environment - Time Offset accumulation

In general, time offset estimates are better than frequency offset estimates. The filter time offset error estimate is very small and stable.

On the other hand, both the filter frequency offset estimate and its error are more less stable.

The estimated frequency drift is negligible for the length of traces available but its standard deviation comparatively large. Thus, the influence and importance of the frequency drift cannot be fully assessed in this study.

We have also found that the frequency offset estimate can fluctuate between packet bursts. Thus, there is a high frequency noise component in the frequency offset estimate that the Kalman filter cannot remove. Since the effect of the frequency drift is negligible for the time scales considered, we should expect a frequency offset estimate with no short term fluctuations. We can remove the frequency offset estimate short term fluctuations by performing a moving average or a regression over consecutive frequency offset estimates. In practice we have found that the combination of 5 consecutive values normally brings the first few correlation coefficients above 0.95.

Finally, the time offset accumulation criteria for inter burst time seems to provide a more consistent performance accross all the traces. Its values would probably be less variable if the frequency estimates were improved.

8.3.1 LAN Performance and Optimality

1. The time offset error is in all cases smaller than the noise in $(\Theta_n)_{n\geq 1}$. It is also very stable. However, for $\bar{t}$ larger than 1000 secs, $\overline{U} \leq \bar{d}$. Thus, for average inter burst times above 1000 secs, the usage of the filter solely for time offset estimation cannot be justified. In this case the usefulness of the filter relies on the frequency offset estimate which provides the means for generating accurate time between packet bursts.

2. The frequency offset error becomes more stable as the interpacket times grows. It is always below the noise in $(\Theta'_n)_{n\geq 1}$.

3. According to our stability criteria, the frequency offset is unstable and fluctuates more than expected.

4. Optimality: The filter shows excellent tracking in all cases ($\overline{M} = 0$). The empirical standard deviation of $(M_n)_{n\geq 1}$ seems to be a bit low, indicating a possible noise parameter overestimation. The correlation coefficients of $(M_n)_{n\geq 1}$ are for the most part close to 0, indicating that the filter is very efficient at extracting information from the noise.

8.3.2 WAN Performance and Optimality

1. The time offset error is in all cases smaller than the noise in $(\Theta_n)_{n\geq 1}$. It is also very

stable. The average time offset error $\overline{U}$ is well below a tenth the half round trip delay $\overline{d}$.

2. The frequency offset error becomes more stable as the interpacket times grows. It is always below the noise in $(\Theta'_n)_{n\geq 1}$.
3. According to our stability criteria, the frequency offset is unstable and fluctuates more than expected.
4. Optimality: The filter shows excellent tracking in all cases ($\overline{M} = 0$). The empirical standard deviation of $(M_n)_{n\geq 1}$ is also very close to unity, indicating good noise parameter estimates. The correlation coefficients of $(M_n)_{n\geq 1}$ are for the most part close to 0, indicating that the filter is very efficient at extracting information from the noise.

9. Conclusions and Future Work

The Kalman filter provides time offset estimates with very low errors. The frequency offset estimates have also very low errors. However, they fluctuate more than expected. This fluctuation can be reduced by choosing interpacket times large enough and combining consecutive estimates with a moving average or a linear regression. The a posteriory checks performed on the data show that the Kalman filter performs very close to optimality in practically all the cases.

There are some areas of future work.

1. Perform further checks on the assumptions on ν and ϵ.
2. Is the drift needed?
3. Filter performance on longer runs.
4. Filter performance in real time.
5. Refinement and further checks on the empirical algorithm for σ.
6. Test the filter performance with more precise timestamping, especially in LAN environments.
7. Full integration with clock selection and combination.

Acknowledgements

I would like to thank Neil McKee for providing parts of the code used in early tests, Dave Mills for a very illuminating discussion and especially Norman Chang for all his efforts in investigating the possibilities of the approach proposed in this paper and for his assistance and help in testing the Kalman filter with NTP timestamps.

A. Appendix: Kalman Recursions in Scalar Form

At time Δ_{n+1} we have,

$$\begin{aligned}
\hat{x}_{n+1} &= \boldsymbol{h}_1^T \hat{\boldsymbol{Z}}_{n+1}, & \text{(A.1)}\\
U_{n+1}^2 &= \boldsymbol{h}_1^T \boldsymbol{C}_{n+1} \boldsymbol{h}_1, & \text{(A.2)}\\
\hat{y}_{n+1} &= \boldsymbol{h}_2^T \hat{\boldsymbol{Z}}_{n+1}, & \text{(A.3)}\\
V_{n+1}^2 &= \boldsymbol{h}_2^T \boldsymbol{C}_{n+1} \boldsymbol{h}_2, & \text{(A.4)}\\
\gamma_{n+1} &= \boldsymbol{h}_1^T \boldsymbol{C}_{n+1} \boldsymbol{h}_2 = \boldsymbol{h}_2^T \boldsymbol{C}_{n+1} \boldsymbol{h}_1, & \text{(A.5)}
\end{aligned}$$

where $\boldsymbol{h}_2 = (0,1)^T$.

A.1 Scalar Recursions

If $z_n^2 = \nu^2 t_n + \epsilon^2$, the offset updates in scalar form are,

$$\begin{aligned}
\hat{x}_{n+1} &= \hat{x}_n + \hat{y}_n t_n + \frac{t_n^2}{2} D + G_{n+1}^2 \Big(\Theta_{n+1} - (\hat{x}_n + \hat{y}_n t_n + \frac{t_n^2}{2} D)\Big), \quad n \ge 0, & \text{(A.6)}\\
U_{n+1}^2 &= \sigma^2 G_{n+1}^2, \quad n \ge 0, & \text{(A.7)}\\
G_{n+1}^2 &= \frac{U_n^2 + 2t_n\gamma_n + t_n^2(V_n^2 + z_n^2)}{\sigma^2 + U_n^2 + 2t_n\gamma_n + t_n^2(V_n^2 + z_n^2)}, \quad n \ge 0, & \text{(A.8)}\\
\hat{y}_{n+1} &= \hat{y}_n + t_n D + H_{n+1} \Big(\Theta_{n+1} - (\hat{x}_n + \hat{y}_n t_n + \frac{t_n^2}{2} D)\Big), \quad n \ge 0, & \text{(A.9)}\\
V_{n+1}^2 &= \frac{(V_n^2 + z_n^2)(\sigma^2 + U_n^2) - \gamma_n^2}{\sigma^2 + U_n^2 + 2t_n\gamma_n + t_n^2(V_n^2 + z_n^2)} \quad n \ge 0, & \text{(A.10)}\\
H_{n+1} &= \frac{\gamma_n + t_n(V_n^2 + z_n^2)}{\sigma^2 + U_n^2 + 2t_n\gamma_n + t_n^2(V_n^2 + z_n^2)}, \quad n \ge 0, & \text{(A.11)}\\
\gamma_{n+1} &= \sigma^2 H_{n+1}, \quad n \ge 0. & \text{(A.12)}
\end{aligned}$$

References

[1] D. W. Allan. Time and frequency (time-domain) characterization, estimation and prediction of precision clocks and oscillators. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, UFFC-34(6):647–654, November 1987.

[2] B. D. O. Anderson and J. B. Moore. *Optimal Filtering.* Prentice-Hall, Englewood Cliffs, NJ, 1979.

[3] M. H. A. Davis and R. B. Vinter. *Stochastic Modelling and Control.* Monographs on Statistics and Applied Probability. Chapman and Hall, London- New York, 1985.

[4] W. C. Lindsey, F. Ghazvinian, W. C. Hangmann, and K. Dessouky. Network synchronization. *Proceedings of the IEEE*, 73(10):1445–1467, October 1985.

[5] D. L. Mills. Internet time synchronization: The network time protocol. *IEEE Transactions on Communications*, Com-39(10):1482–1493, October 1991.

[6] D. L. Mills. Modeling and analysis of computer netwrok clocks. *Technical Report 92-5-2, Electrical Engineering Department, University of Delaware*, May, 1992.

[7] D. L. Mills. Network time protocol (version 3). specification, implementation and analysis. *RFC 1305, University of Delaware*, March, 1992.

[8] D. L. Mills. Precision synchronization of computer network clocks. *Technical Report 93-11-1, Electrical Engineering Department, University of Delaware*, November, 1993.

[9] D. L. Mills and A. Thyagarajan. Network time protocol (version 4). proposed changes. *Technical Report 94-10-2, Electrical Engineering Department, University of Delaware*, October, 1994.

[10] P. V. Tryon and R. H. Jones. Estimation of parameters in models for cesium beam atomic clocks. *Journal of Research of the Nationa Bureau of Standards*, 88(1):3–15, January-February 1983.