Kalman Filtering applied to Timing Recovery in Tracking Mode

Panu Chaichanavong
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
USA

Brian H. Marcus

Jorge Campello de Souza

Richard M. H. New

Bruce A. Wilson
IBM Almaden Research Center
San Jose, CA 95120
USA

Abstract

This paper investigates the performance of the Kalman filter as a timing recovery system in tracking mode, subject to a wide range of operating conditions. Our simulation compares the Kalman filter and the phase-locked loop based on the number of divergences for various values of timing disturbance and SNR. They are shown to work equally well in low SNR and disturbance. However, the Kalman filter outperforms the phase-locked loop when both SNR and disturbance are high. The simulation also suggests that the Kalman filter is more robust to variations in operating conditions.

1 Introduction

One of the most essential parts of synchronous data transmission is timing recovery. In a PRML channel employed in a high density disk drive, the analog read signal is sampled by an A/D converter and then equalized by a digital equalizer. Based on the equalized digital signal, the objective of timing recovery is to adjust the clock of the A/D converter to the correct frequency and phase.

There are many challenges in timing recovery. First the initial phase is not known in advance, and the initial frequency is known only to some degree of confidence. Second, disturbances in the system such as motor speed variation can change the phase and frequency of the read signal. Third, measurement noise can corrupt the signal, and so the timing information embedded in the signal is harder to extract. Last, timing error cannot be directly computed from the measured signal. A phase-locked loop (PLL), the most common timing recovery system, can deal with these problems to some extent. In this paper, we incorporate the well known Kalman filter to the timing system and show how it improves the performance of timing recovery. The relationship between timing recovery and Kalman filtering was discovered independently by Christiansen [1] and Driessen [2], who show that for a specific timing model, the second order PLL has the same structure as the Kalman filter. However, the Kalman filter has an advantage over the PLL in that it is time-variant in nature. In [2], Driessen demonstrates that this time-variant property of the Kalman filter makes it more favorable in acquisition mode in which the timing system adjusts the receiver clock to the initial phase and frequency. In this paper, we explain the relationship between PLL and Kalman filter in a different way from [1, 2]. Then we focus on the so called tracking mode and show that our Kalman filter works well in this mode as well.

In the next section, we briefly explain the concept and give a big picture of timing recovery. In section 3, we review the linear state space model and the state feedback and state estimation problem. The Kalman filter is also described in this section. In section 4, we derive the state space timing model used in the Kalman filter and PLL. We give a brief description of our PLL in section 5. Our simulation results are presented in section 6. Section 7 is the conclusion.

2 Timing recovery

There are many levels of synchronization, including bit, frame, and packet synchronization. In this paper, we consider only the lowest level: bit synchronization, the objective of which is to find where each bit starts. We assume that each data frame comprises three parts: preamble, frame sync word, and data (see Figure 1). Each frame begins with a preamble which is a periodic pattern of bits, usually with a short period. The preamble is used to achieve bit sync, i.e. to adjust the receiver's clock. Next is the frame sync word which is a known, fixed length word with good correlation property. After bit sync is achieved, the receiver will look for this word to determine where the data starts.

The timing recovery algorithm starts with its acquisition mode. In this mode the timing algorithm takes advantage of knowing the preamble and thus can be made aggressive. The clock must be adjusted to the correct phase and frequency as quickly as possible so that the overhead of the frame can be made small. Then it switches to tracking mode. In contrast to the acquisition mode, the tracking mode must deal with variation in the data. We shall focus only on the tracking mode.

There are many configurations for timing recovery. We shall use the structure presented in Figure 2. First the read signal is calibrated by a gain control algorithm and sampled by a clock provided by a timing loop. The samples are then equalized to a partial response target (such as PR4 $= 1 - D^2$) by an FIR equalizer and fed back to provide information to the timing and gain control loop. Thus the timing loop essentially sees the equalized signal

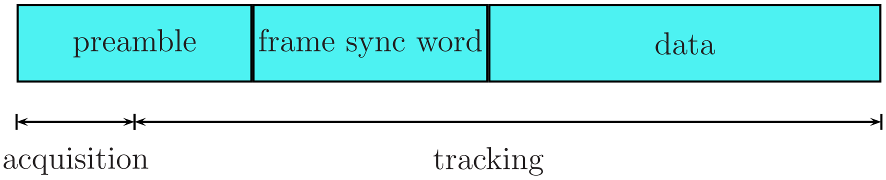

Figure 1: Frame components

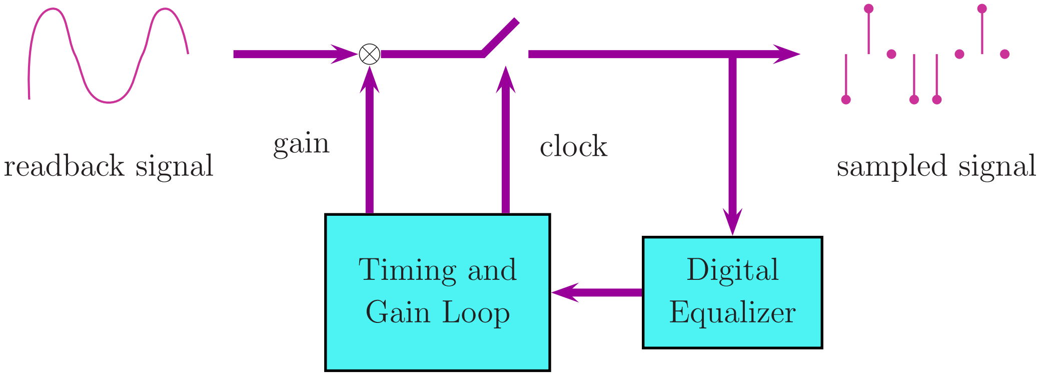

Figure 2: Timing recovery diagram

with some delay due to the equalizer. The timing loop is usually a phase-locked loop (PLL) which is described in section 5. In this paper we employ the Kalman filter in the timing loop and investigate its performance compared to the PLL.

3 Background on control and estimation

In this section, we describe the general problem of control and estimation using a state space model. For more information, the reader is referred to excellent textbooks in the field, e.g. [3, 5, 4].

The (discrete time) state space model can be described by

$$\begin{aligned} x_{i+1} &= Fx_i + Gu_i \\ y_i &= Hx_i \end{aligned}$$

where x_i is a state vector, u_i is an input vector, and y_i is an output vector. The system is characterized by three matrices: a dynamic matrix F, an input matrix G, and an output matrix H. This model leads to a powerful control law known as state feedback, in which the input is a linear function of the current state

$$u_i = -Kx_i.$$

In practice, the state is not usually known. Instead, we have some measurement y_i which conveys information on the state. This measurement is used to compute the estimate of the state by a state estimator. The estimator alternatively updates the estimate using a time update and a measurement update which can be described as follows. Let $\hat{x}_i$ and $\bar{x}_i$ denote the estimates of x_i given the measurement up to time i and $i-1$ respectively. The measurement update "corrects" the estimate by its gain L and the difference between the measurement (actual output) y_i and the expected output $H\bar{x}_i$

$$\hat{x}_i = \bar{x}_i + L(y_i - H\bar{x}_i).$$

Then the time update "propagates" the estimate by the system dynamics and the input

$$\bar{x}_{i+1} = F\hat{x}_i + Gu_i.$$

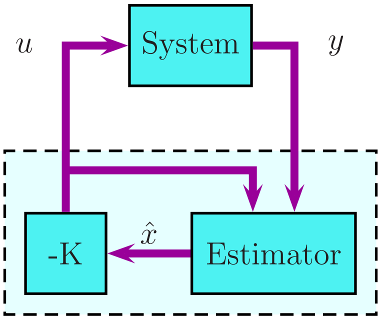

Figure 3: State feedback and state estimator

Using this approach, the control and estimation problem is to find appropriate gains K and L. Note that the system model F, G, H and both control gain K and estimator gain L can be time-variant. In fact, K and L are usually time-variant even when F, G and H are not.

Next we consider the noise in the system which is modeled as

$$\begin{aligned} x_{i+1} &= Fx_i + Gu_i + w_i \\ y_i &= Hx_i + n_i \end{aligned}$$

where w_i is the process noise or disturbance and n_i is the measurement noise. They are both assumed to be white, zero-mean with covariance matrices W and N respectively. They are also assumed to be uncorrelated with each other. The mean, (resp., covariance) of the initial state x_0 is denoted μ_0, (resp. Π_0). With these assumptions, the Kalman filter produces the gain L_i such that the error variance $E(x_i - \hat{x}_i)(x_i - \hat{x}_i)^T$ is minimized. Its algorithm can be described as follows. Let $\bar{x}_0 = \mu_0$ and $\bar{P}_0 = \Pi_0$; then for each $i \geq 0$,

$$\begin{aligned} L_i &= \bar{P}_i H^T (H\bar{P}_i H^T + N)^{-1} && (3.1)\\ \hat{x}_i &= \bar{x}_i + L_i(y_i - H\bar{x}_i) && (3.2)\\ \hat{P}_i &= (I - L_i H)\bar{P}_i && (3.3)\\ \bar{x}_{i+1} &= F\hat{x}_i + Gu_i = (F - GK)\hat{x}_i && (3.4)\\ \bar{P}_{i+1} &= F\hat{P}_i F^T + W && (3.5) \end{aligned}$$

where $\bar{P}_i$ and $\hat{P}_i$ represent the covariance of $\bar{x}_i$ and $\hat{x}_i$ respectively. Note that equations (3.1)-(3.3) characterize the measurement update; and equations (3.4) and (3.5) characterize the time update.

4 Timing model

4.1 State space model

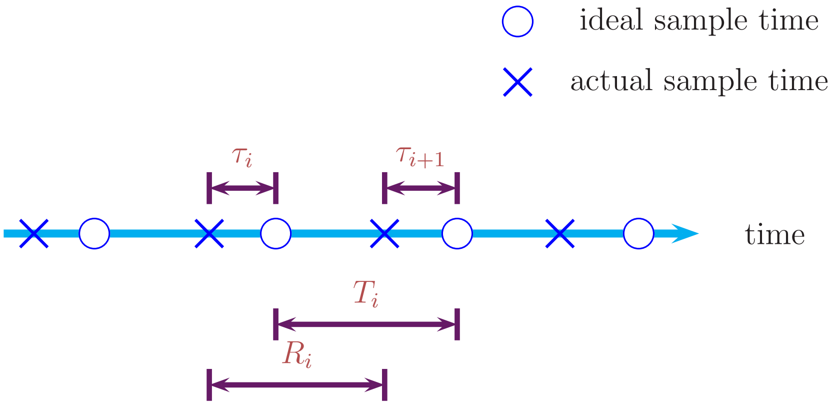

Figure 4: Time axis

Consider the time axis in Figure 4. We want to sample the signal at the ideal sample time, which can be viewed as the correct time instance that we expect to get the lowest bit error rate. The ideal inter-sample interval T_i can vary because of timing disturbances. The timing error τ_i is defined to be the difference between the actual and the ideal sample time. The clock increment, returned by a timing algorithm, is denoted by R_i. From the time axis we can see that

$$\tau_{i+1} = \tau_i + T_i - R_i. \tag{4.6}$$

Our goal is to produce the clock increment R_i which minimizes the timing error. This is done by computing estimates of τ_i and T_i, denoted by $\hat{\tau}_i$ and $\hat{T}_i$ respectively, and then setting

$$R_i = \hat{\tau}_i + \hat{T}_i - v_i \tag{4.7}$$

where v_i is a velocity disturbance. The ideal inter-sample interval is perturbed by an acceleration disturbance a_i

$$T_{i+1} = T_i + a_i. \tag{4.8}$$

Both timing disturbances represent the variation of motor speed and other mechanical units including the heads. They also include defects on the recording surface and vibration due to the movement of the disk drive. From equations (4.6)-(4.8), we obtain the timing model

$$\begin{aligned} \tau_{i+1} &= \tau_i + T_i - \hat{\tau}_i - \hat{T}_i + v_i \qquad (4.9) \\ T_{i+1} &= T_i + a_i. \qquad (4.10) \end{aligned}$$

The initial state $[\ \tau_0 \quad T_0\ ]^T$ is assumed to have mean $[\ 0 \quad \mu_T\ ]^T$ and, as in the preceding section, its covariance matrix is denoted Π_0. The disturbance $w_i = [\ v_i \quad a_i\ ]^T$ is assumed to have zero mean and its covariance matrix is denoted W.

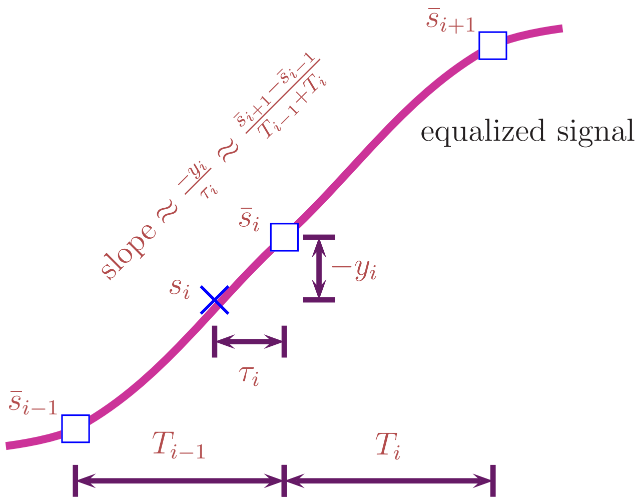
$\bar{s}_{i+1}$
slope $\approx \frac{-y_i}{\tau_i} \approx \frac{\bar{s}_{i+1}-\bar{s}_{i-1}}{T_{i-1}+T_i}$
equalized signal
$\bar{s}_i$
$-y_i$
s_i
τ_i
$\bar{s}_{i-1}$
T_{i-1}
T_i

Figure 5: Output equation computation

Next, we shall compute the output equation using Figure 5. The curve shows the signal equalized to the PR4 target; thus the ideal sample $\bar{s}_i$ can take the values -1,0,1. The actual sample that we measure is denoted by s_i. Since we have no knowledge about the data, the ideal sample values must be computed from the actual sample values. This is done by a simple threshold detector (if $s_i > 0.5$ then set $\bar{s}_i = 1$, if $s_i < -0.5$ then set $\bar{s}_i = -1$, otherwise set $\bar{s}_i = 0$). The output y_i of the system is defined to be $s_i - \bar{s}_i$. From the figure, we can see that the output can be estimated by

$$y_i \approx \left(\frac{\bar{s}_{i-1} - \bar{s}_{i+1}}{T_{i-1} + T_i} \right) \tau_i \approx \left(\frac{\bar{s}_{i-1} - \bar{s}_{i+1}}{2\mu_T} \right) \tau_i. \tag{4.11}$$

Patapoutian investigates this idea of using the slope for the output equation in [7]. He uses a more accurate nonlinear description of the measurement and then applies an extended Kalman filter to his model.

Combining equations (4.9),(4.10), and (4.11), and incorporating the measurement noise n_i into the output equation, we have the state space model

$$\begin{bmatrix} \tau_{i+1} \\ T_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tau_i \\ T_i \end{bmatrix} - \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\tau}_i \\ \hat{T}_i \end{bmatrix} + \begin{bmatrix} v_i \\ a_i \end{bmatrix} \tag{4.12}$$

$$y_i = \begin{bmatrix} \frac{\bar{s}_{i-1}-\bar{s}_{i+1}}{2\mu_T} & 0 \end{bmatrix} \begin{bmatrix} \tau_i \\ T_i \end{bmatrix} + n_i \tag{4.13}$$

In terms of the system model in section 3, we have:

$$x_i = \begin{bmatrix} \tau_i \\ T_i \end{bmatrix} \tag{4.14}$$

$$F = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \tag{4.15}$$

$$G = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \tag{4.16}$$

$$K = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.17}$$

$$H = \begin{bmatrix} \frac{\bar{s}_{i-1}-\bar{s}_{i+1}}{2\mu_T} & 0 \end{bmatrix} \tag{4.18}$$

From this model, $\bar{\tau}_i$, the estimate of τ_i given the measurement up to time $i-1$, is zero for all i (see equation (3.4)). Thus the timing error estimator becomes

$$\hat{\tau}_i = L_{1,i} y_i \tag{4.19}$$

$$\hat{T}_i = \hat{T}_{i-1} + L_{2,i} y_i \tag{4.20}$$

where $L_i = [\begin{array}{cc} L_{1,i} & L_{2,i} \end{array}]^T$.

4.2 Loop delay

The reader may notice that this model is not causal; it involves the term $\bar{s}_{i+1}$ in the output equation. Thus it cannot be directly implemented. Moreover, the A/D converter and the equalizer also introduce delay into the timing loop. To take care of this delay, we need some modification. Originally, the estimator assumes that the measurement $y_i, y_{i-1}, \ldots$ and the input $u_{i-1}, u_{i-2}, \ldots$ are known to estimate the state x_i. However, with the delay, the only measurement we have in order to estimate x_i is $y_{i-d}, y_{i-d-1}, \ldots$.

Let $\hat{x}_{i|j}$ be the estimate of x_i based on the measurement up to time j and the input up to time $i-1$. We remark that $\hat{x}_{i|i} = \hat{x}_i$ and $\hat{x}_{i|i-1} = \bar{x}_i$. We start by doing usual estimation at time $i-d$ to obtain $\hat{x}_{i-d|i-d}$. Then we propagate this estimate using the state space model and the knowledge of the input

$$\begin{aligned}
\hat{x}_{i-d+1|i-d} &= F\hat{x}_{i-d|i-d} + Gu_{i-d} \\
\hat{x}_{i-d+2|i-d} &= F\hat{x}_{i-d+1|i-d} + Gu_{i-d+1} = F^2\hat{x}_{i-d|i-d} + FGu_{i-d} + Gu_{i-d+1} \\
&\vdots \\
\hat{x}_{i|i-d} &= F^d\hat{x}_{i-d|i-d} + \begin{bmatrix} G & FG & \dots & F^{d-1}G \end{bmatrix} \begin{bmatrix} u_{i-1} \\ u_{i-2} \\ \vdots \\ u_{i-d} \end{bmatrix}.
\end{aligned}$$

This is a natural way of extending an estimator to a predictor, where we estimate the future state, not the current state. A similar approach is given by Christiansen in [1]. Patapoutian [9] studies this in a more general way by considering two different delays simultaneously: one before the timing loop and the other after the timing loop.

5 Classical phase-locked loop

In a second order classical phase-locked loop (PLL), $\hat{\tau}_i$ and $\hat{T}_i$ are computed by

$$\hat{\tau}_i = K_p\gamma_i \tag{5.21}$$

$$\hat{T}_i = \hat{T}_{i-1} + K_c\gamma_i \tag{5.22}$$

where γ_i is the timing gradient computed by [6]

$$\begin{aligned}
\gamma_i &= y_i(\bar{s}_{i-1} - \bar{s}_{i+1}) \\
&= (s_i - \bar{s}_i)(\bar{s}_{i-1} - \bar{s}_{i+1}).
\end{aligned}$$

K_p and K_c are the proportional gain and cumulative gain which adjust the effect of the timing gradient on the estimate. Finding appropriate gains is not easy and often done by trial-and-error. Note that $\gamma_i = 2\mu_T y_i h_i$ where $H_i = [\ h_i \quad 0\ ]$; thus, by comparing equations (4.19)-(4.20) and (5.21)-(5.22), we can see that the Kalman filter and the PLL indeed have the same structure. While there are many varieties of PLLs, we shall use the one described here to compare with the Kalman timing loop.

In [1, 2, 8], timing (phase) error is assumed to be measurable by a phase detector, resulting in a time-invariant output equation in contrast to equation (4.13). For the time-invariant model F, G, and H, the Kalman gain L_i and state covariance P_i will converge quickly to a steady state gain L_∞ and covariance P_∞. Thus there cannot be much improvement over PLL in tracking mode for a time-invariant model.

By comparing the Kalman and PLL algorithms, an obvious drawback of the Kalman filter is its complexity. However, Patapoutian [8] shows that for the time-invariant timing model, the Kalman gain can be explicitly given in terms of the time index and disturbance and noise variance. For our model, the output equation is time-variant. Hence, a nonrecursive expression for the Kalman gain is not possible.

6 Simulation results

There are many figures of merit which judge the performance of the timing loops, such as mean square of timing error and bit error rate. In this paper, we use the number of divergences as our criterion (we say that a divergence occurs if the timing error grows without bound; and we expect to see the bit error rate of approximately 0.5 if it occurs). In fact, the ultimate goal is to reduce the bit error rate. However, given that the timing loop does not diverge, the bit error rate of each timing loop is very close for the same value of SNR. Furthermore, the mean square timing error does not relate directly to the bit error rate. For these reasons we choose the number of divergences, as determined by bit error rate, as a measure of performance.

6.1 Performance

We are interested in a wide range of operating points; each is characterized by the SNR (for measurement noise) and the variance of the acceleration disturbance. For each operating point, we generate 1000 synthetic waveforms of length 24 sectors ($\approx 10^5$ bits) and run both the PLL and Kalman timing algorithms and count the number of divergences. A run is classified as a divergence if the number of bit errors exceeds 4000. First we fix SNR and set the acceleration disturbance to the maximum such that Kalman timing diverges on approximately 100 runs or 10%. With this disturbance, the gains K_p and K_c of the PLL are adjusted to give the best result. They are found by running the PLL over a wide range of gain on a smaller number of waveforms. The result is shown in Table 1.

SNR (dB)	Accel. Var.	Kalman	PLL	K_p	K_c
18	1×10^{-10}	88	100	4×10^{-4}	4×10^{-7}
22	2×10^{-8}	92	90	1×10^{-3}	6.7×10^{-6}
26	3.5×10^{-7}	116	555	2×10^{-3}	3×10^{-5}
30	1.5×10^{-6}	112	856	2×10^{-3}	8×10^{-5}

Table 1: Number of divergences with total of 1000 runs

Observe from the table above, that both timing algorithms appear to work well when the disturbance is low and the measurement noise is large. This can be explained by showing that in this case the Kalman filter looks very much like the PLL. To see this, first observe that it is natural to assume that, with the possible exception of the (1,1)-entry, all entries of the initial covariance matrix $\bar{\Pi}_0 = \bar{P}_0$ are very small. Now, we can use equations (3.1), (3.3) and (3.5) to iteratively pass from $\bar{P}_i$ from $\bar{P}_{i+1}$. A computation based on this reveals that in each such iteration, all entries except perhaps for $(\bar{P}_i)_{1,1}$ remain small and the (1,1)-entry is scaled by a factor of roughly $\frac{1}{1+(\bar{P}_i)_{1,1}/N}$, and will therefore decrease exponentially until the (1,1)-entry becomes much smaller than N. Indeed, simulations confirm that $\bar{P}_i$ settles very quickly to a roughly constant value much less than N. Therefore from equation (3.1), we see that L_i is approximately a constant matrix multiplied by the slope h_i. This explains why both timing loops exhibit similar performance (compare equations (4.19) and (4.20) with equations (5.21) and (5.22)).

At large disturbance and low SNR, the Kalman loop outperforms the PLL in terms of the number of divergences. In this situation, the gain L_i will be more time-variant and this will give an advantage over the PLL.

6.2 Robustness

In this section, instead of using the estimate $\hat{\tau}_i$ and $\hat{T}_i$ directly, we can reduce their effect by passing through another filter which implements the control gain similar to the one described in section 3. In this scheme, we replace $\hat{\tau}_i$ and $\hat{T}_i$ in equations (4.9) and (4.12) by τ_i' and T_i' given by

$$\begin{aligned} \tau_i' &= K_{11}\hat{\tau}_i + K_{12}(\hat{T}_i - T_i') \\ T_{i+1}' &= T_i' + K_{21}\hat{\tau}_i + K_{22}(\hat{T}_i - T_i') \end{aligned}$$

where $T_0' = \mu_T$. We have more degrees of freedom in this scheme which makes the design more difficult. However, a good choice of control gain may result in a better performance at a particular operating point. We have several assumptions for the Kalman filter to be optimal and these assumptions cannot be all true in practice. Moreover, disturbance and noise variance may not be accurately known at the design time. Thus, this scheme may be a better alternative. Its drawback is that it is more complex and a good gain is difficult to choose.

The following results demonstrate the robustness of the Kalman timing loop compared to the classical PLL. We consider two different operating points. One has an SNR of 18 dB with no disturbance. The other has an SNR of 22 dB with velocity disturbance of variance 10^{-3}. We use two different PLLs, with gains optimized for each operating condition. For the Kalman loop, the control gains are the same for both operating conditions. The parameters for the PLL and Kalman timing loops are given in Tables 2 and 3. For both operating points, each timing loop is run on 2000 synthetic waveforms. The numbers of divergences are shown in Table 4.

Timing loop	K_p	K_c
PLL1	6×10^{-4}	3×10^{-7}
PLL2	1×10^{-3}	3×10^{-7}

Table 2: Parameters of the PLLs

We can see that although the PLLs work well for a given operating condition, they are not robust to the change of the environment (in this case, SNR and disturbance). PLL1

parameter	value
disturbance variance W	$\begin{bmatrix} 10^{-3} & 0 \\ 0 & 10^{-11} \end{bmatrix}$
measurement noise variance N	0.06
loop delay	9
control gains K_{11}, K_{12}, K_{22}	0.03
control gain K_{21}	3×10^{-5}

Table 3: Parameters of the Kalman filter timing loop

SNR (dB)	Velocity Var.	Kalman	PLL1	PLL2
18	0	120	250	1198
22	10^{-3}	150	722	206

Table 4: Number of divergences with total of 2000 runs

has smaller gain and thus works well for low SNR. However, it cannot keep track of the disturbance, resulting in a large number of divergences when the disturbance is present. On the other hand, PLL2 has larger gain and the response of the timing loop is fast enough to track the actual timing even with the disturbance. This also results in larger bandwidth and the system is more sensitive to noise. For the Kalman timing loop, the numbers of divergences are small for both operating points. Since we do not change the parameters of the Kalman timing loop, this suggests that the Kalman timing may be more robust in some circumstances.

7 Conclusion

We briefly summarize our contributions as follows.

- We give a different explanation of the relationship between PLL and Kalman filter than in earlier work.

- By using the number of divergences as a criterion, we observe that:

  - the tracking mode performance of the Kalman filter is comparable to the PLL in low SNR and small disturbance; but in high SNR and large disturbance, the Kalman filter gives much lower number of divergences,

  - the Kalman filter may be more robust to variations in SNR and disturbance compared to PLL.

8 Acknowledgement

The authors would like to thank Ara Patapoutian for preprints and helpful discussion.

References

[1] Christiansen, G. S. Modeling of a PRML timing loop as a Kalman filter. *IEEE GLOBECOM Conf.* (1994), 1157–1161.

[2] Driessen, P. F. DPLL bit synchronizer with rapid acquisition using adaptive Kalman filtering techniques. *IEEE Trans. Commun. 42* (Sept. 1994), 2673–2675.

[3] Franklin, G. F., Powell, D. J., and Workman, M. L. *Digital Control of Dynamic Systems.* Addison Wesley, 1998.

[4] Kailath, T. *Linear Systems.* Prentice Hall, 1980.

[5] Kailath, T., Sayed, A. H., and Hassibi, B. *Linear Estimation.* Prentice Hall, 2000.

[6] Mueller, K. H., and Müller, M. Timing recovery in digital synchronous data receivers. *IEEE Trans. Commun. COM-24* (May 1976), 516–531.

[7] Patapoutian, A. Baseband PAM synchronization with extended Kalman filters. *submitted*.

[8] Patapoutian, A. On phase-locked loops and Kalman filters. *IEEE Trans. Commun. 47* (May 1999), 670–672.

[9] Patapoutian, A. Application of Kalman filters with a loop delay in synchronization. *IEEE Trans. Commun. 50* (May 2002).