DSP BASED GMSK MODEM ALGORTHM DESIGN

Prepared by:

ComFocus Corporation
San Diego, California

for:

Synetcom Digital
Hermosa Beach, California

DSP-Based GMSK Modem
Algorithm Design

From: ComFocus Corporation
Date: July 12, 1990

Scope of work:

1.Demodulation
Discuss several noncoherent detection options including:
- Synetcom's baseline approach
- Trade-off of filter-implemented versus differential delay implemented noncoherent discriminator design from the DSP point of view

2.Bit Synchronization
Discuss available options

3.Definitive Design
-Propose design for demodulator and bit synchronizer
-Assess performance in terms of bit synchronizer lock and BER versus E_b/N_o

Before launching into item 1, a couple of points arose while reading the RAM specification [1]. First of all, referring to Figure 4 of [1], bit error rate (BER) is given as a function of receiver input signal power. Without knowledge of the receiver noise figure, it is not possible to uncover just what level of performance with respect to E_b/N_o RAM is anticipating. In that regard, no real performance objectives have been levied, other than the normal "make it as good as possible." Some guestimates are in order.

The peak frequency deviation for the system is ±2 KHz corresponding to a modulation index of 0.5 since the data rate is 8 Kb/s [1]. If we arbitrarily choose a very wide receiver bandwidth, say 10 KHz, which is clearly unnecessary, with a conservative receiver noise figure of 6 dB, the effective input signal to noise ratio at -115 dBm ([1], Figure 4) is

$$SNR = -115\ dBm - [\ -174\ dBm/Hz + 10\ Log(\ 10\ kHz\ ) + 6\ dB\ ] \\ = 13\ dB \qquad (1)$$

This is far more SNR than is required to meet the $5 \cdot 10^{-2}$ BER requirement, even with the crude parameter choices used in (1). Therefore, unless the receiver noise figure is terrible by today's standards, the BER requirements ([1] Figure 4) are certainly attainable.

A second opening issue concerns frame synchronization as defined on page MTS17.2 in [1]. The block is 16 bits long and frame synch is only established if there is no more than 1 bit in error. Therefore, the probability of a successful frame acquisition (given that proper bit synchronization has occurred) is

$$P_{A1} = (1-p)^{16} + 16p(1-p)^{15} \qquad (2)$$

where p is the probability of a single bit error. It is insightful to evaluate (2) for a number of bit error probabilities as done in Table I.

Bit Error Probability p	Prob. of Successful Frame Acq. P_{A1}
20%	47.85%
10	51.47
5	81.08
2	96.01
1	98.91
0.5	99.71
0.2	99.95
0.1	99.988

Table I Successful frame synchronization probability versus bit error rate.

At the higher bit error rates, it is important to recognize that the true probability of frame acquisition given

that a synchronization signal is in fact present is given by

$$P_{ACQ1} = P_{A1}\, P_{BS1} \qquad (3)$$

where P_{BS1} is the probability of properly acquiring bit synchronization.

Although not mentioned at any time, false acquisitions when no signal is in fact present should be of some concern at the system level. For completely random data bits, the probability of erroneously obtaining frame synch (after erroneously obtaining bit synch) is

$$P_{A0} = (.5)^{16} + 16(.5)(1-0.5)^{15} = (0.5)^{16} \times 17 = 0.00026 \qquad (4)$$

At 8 kbps, this translates to a false alarm rate somewhere between once every 0.5 to 7.7 seconds. In my estimation, this is quite poor. The overall false alarm probability is in fact

$$P_{ACQ1} = P_{A0}\, P_{BS0} \qquad (5)$$

where P_{BS0} is the probability of the bit synchronizer erroneously synchronizing. If the bit synchronizer also extracts an indication of whether it has attained lock or not, then (5) indeed applies. The drawback here is that reliable determination of whether the bit synchronizer has locked or not is itself a difficult task on such a short preamble. The scheme increasingly appears targeted for only high SNR application.

Let's look more closely at the bit synchronization preamble. It is a total of 16 symbol periods long with an alternating 11 and 00 pattern (i.e. OOZZOOZZ...)[1]. Proper bit synchronization is equivalent to obtaining an accurate estimate of the symbol epoch. For this data pattern, the problem can be further equivalenced to estimating the phase of a sinusoid of known frequency in noise. Gaussian noise will be assumed for simplicity (even though the noise out of the FM demodulator is neither Gaussian nor white). Many techniques primarily based upon estimation theory have been proposed for this task (e.g. [2-4]). The variance of the epoch estimation error for a sinusoid embedded in additive white Gaussian noise (AWGN) can be bounded by using the Cramer-Rao bound [2,3,5] as

$$\frac{\sigma_T}{T} \geq \frac{1}{\pi}\sqrt{\frac{1}{2\rho n}} \qquad (6)$$

where σ_T/T Normalized time epoch estimation variance for 1010... sine wave
ρ Signal to noise ratio out of demodulator (Noise BW taken as 1/T)
n Number of symbol periods observed

The real result is 2X as bad since the aforementioned sine wave formula was based upon a OZOZ sequence whereas the present preamble pattern has a period 2 times as long.

Using n=16, and including this additional factor of 2 in (6),

$$\frac{\sigma_T}{T} \geq \frac{.1125}{\sqrt{\rho}} \qquad (7)$$

which is the estimate error standard deviation in terms of true symbol periods. Equation (7) may be expressed in terms of the input signal to noise ratio, (ρ_i), by including the so called FM improvement factor. For the 1100... case

$$\beta = \frac{\Delta f}{f_m} = \frac{2KHz}{\frac{8KHz}{4}} = 1 \qquad (8)$$

From [37], the FM improvement factor (for the sinusoidal case) is $3\beta^2$ or 3 here in which case (7) becomes

$$\frac{\sigma_T}{T} \geq \frac{0.1125}{\sqrt{3\rho_i}} \qquad (9)$$

This is, of course, a lower bound. Therefore, without further justification, the value in (9) will be increased by a factor of $\sqrt{2}$ or equivalently 3dB.

For NRZ symbols in AWGN, a σ_T/T value of 0.05 leads to roughly 1 dB from theory degradation due to timing errors whereas $\sigma_T/T = 0.07$ causes a 4.5 dB departure from theory [6]. Although these values pertain to steady-state operation with

square NRZ pulses in AWGN, I believe that they are still fairly representative. Therefore, a realistic goal for σ_T/T is 0.05. Using (9) and the $\downarrow 2$ factor, the required input signal to noise ratio is therefore $\rho_i \geq 5.3$ dB.

This is not to say that bit synchronization at fairly low SNR is taken care of because this result applies to a parallel maximum likelihood receiver, not a serial loop implementation. Limiting the implementation of the synchronizer to realistic serial loop approaches probably restricts reliable operation to considerably higher SNRs.

One alternative for improving the system performance substantially with the present waveforms is this aforementioned parallel processing approach. If we view the bit synchronization and frame synchronization frames jointly rather than separately, a number of things work in our favor. There are other complications which arise also, but I will dismiss these for now.

The basic idea is that we perform bit synchronization and frame synchronization jointly by operating the TMS320 as a matched filter receiver, matched to the bit sync and frame synchronization blocks as shown in Figure 1. Using this approach, bit and frame synchronization are achieved at the same instant (at least to within 1/6 symbol since you're sampling 6x per baud). Since many more symbols are involved, both the symbol epoch estimate and the false-alarm rate would be significantly improved. My one real concern is that you may not have adequate processor throughput for this approach.

There are other issues to address also. Ideally, the frame synchronization bit patterns would be selected from a family of low cross-correlation codes (e.g. Gold,Barker) to prevent false-alarms on correlation sidelobes. Once a detection threshold crossing has occurred, the last 16 bits could be examined per the one-bit error dismissal criterion for achieving frame synchronization. Ideally, intersymbol interference (ISI) effects would be eliminated prior to examining the actual frame synchronization bits, but here again, processor throughput is the chief concern.

Before leaving this area of discussion, a rough estimate of the bit synchronizer's transient response (serial) impact on things can be made as follows. Assume that the loop requires 4 "loop time constants" to sufficiently settle. The "loop time constant" for a classical second-order system is given by

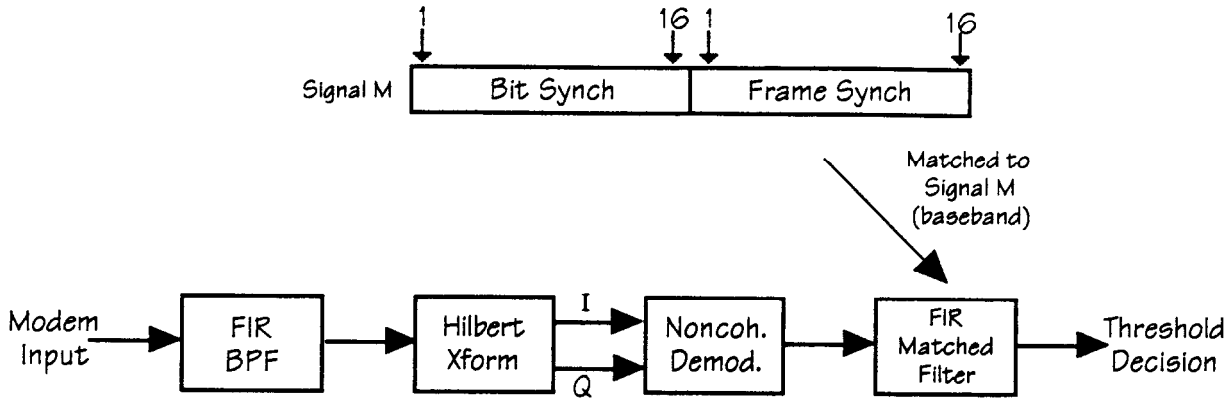


Figure 1 - Bit and Frame Synchronization

$$\tau_L = (\xi \omega_n)^{-1}$$

where ξ is the damping factor and ω_n is the loop natural frequency in radians/second. The loop bandwidth B_L is $\approx 2\xi\omega_n/2\pi$ Hz. Using this,

$$\tau_L = (\pi B_L)^{-1}$$

Given the 16 symbol periods available for bit synchronization and the $4\tau_L$ criteria above, the loop bandwidth must be $\geq$ 640 Hz. This is substantial, but still a fairly small fraction of the 8 Kb/s data rate so my guarded pessimism is probably a bit over done. The false-alarm rate remains a problem, but that will be (must be) resolved outside the (DSP portion of the) demodulator. Hence, all in all, this looks pretty good.

Demodulation

Synetcom's Baseline Approach

My understanding of the baseline approach is sketched in Figure 2 following [7]. One immediate concern with the baseline approach as described in [7] is the hard-limiter implementation. As nonlinearities go, detailed analysis is quite time consuming and will not be undertaken here. The primary point is best made with reference to Figure 3.

Admittedly, the signal ahead of the hard-limiter has been filtered and is clearly band-limited. Regardless, the sampling rate of 48 kHz compared to the 8 kHz center frequency introduces coherent and/or random phase modulation of the effective zero crossings of $\pm 2\pi/12$ or $\pm 30°$ if hardlimiting is used in the DSP algorithms. This amount of phase jitter is of concern to me unless fairly exhaustive analysis were first performed to assess its effects.

On the subject of differential versus discriminator detection, a fair amount of discussion has appeared in the literature. A fairly exhaustive treatment has been given in chapter 10 of [8]. Somewhat surprisingly, much of the applicable material from [8] is nearly word for word from [9]. There seems little point in going through many of the details aside from stating a summary position on the approach to be taken herein. Suffice it to quote:

> "The error-rate performance of differential

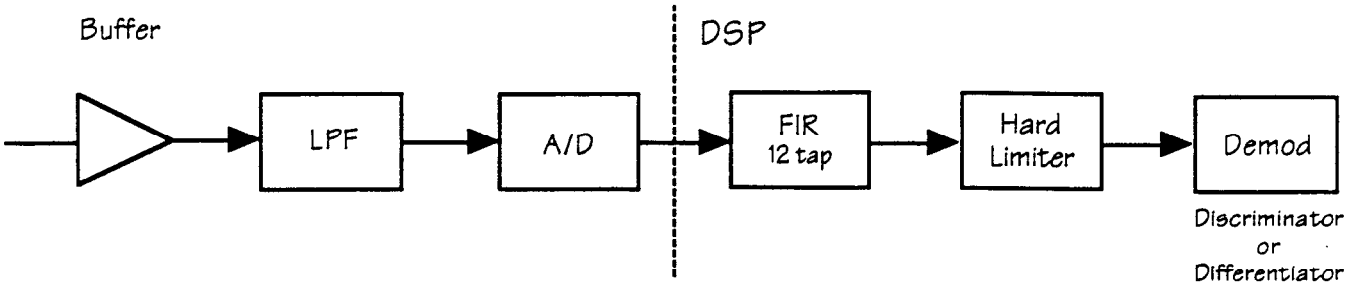


Figure 2 - Demodulator Portion of Synetcom's Baseline Approach

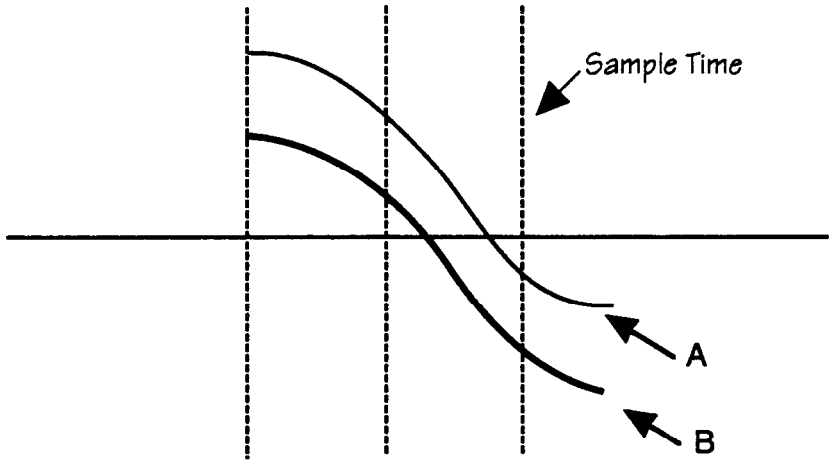


Figure 3 - Signal Paths Displaying Same Hard Limiter Output Sequence

detection is inferior to that of discriminator detection irrespective of the modulation index" in a fast Rayleigh fading environment [9].

Since the differential approach introduces memory into the demodulator by design which may be very long compared to the channel dynamics (particularly at low data rates), it is not recommended. Even for slow Rayleigh quasi-static fading, discriminator detection was found to be only ≈0.7 dB inferior to differential at 10^{-3} BER [9] and I believe that this difference may be eliminated in what I have to propose later herein.

If complex signal notation is employed, the differential and discriminator outputs may be represented by

$$v(t) = \frac{Re[-jZ^*(t)\dot{Z}(t)]}{|Z(t)|^2} \quad discriminator \tag{10}$$

$$v(t) = Re[-jZ(t)Z^*(t-T)] \quad differential \tag{11}$$

from [8]. The demodulator memory is clearly visible in (11). Although the derivative in (10) requires a time delay as well, it can be made << T.

Although the baseline I will propose is based upon a form of discriminator, several other FM demodulation techniques are worthy of mention. Some are not that suitable for implementation in our case but the references are provided just the same [10-13].

During this phase of the project, detailed fading research is a secondary issue aside from choosing a fade-resilient approach. Should this issue later be raised, a number of references have been identified to date [14-23]. No further explicit consideration will be made in the balance of this memo on fading.

Bit Synchronization

Many different methods are available for performing the bit synchronization function. Reliable determination of synchronization is an equally difficult problem given the short bit synchronization frame. Extraction of good data estimates is also a very important element of the overall system and is, or can be, viewed as a problem separate from the clock recovery problem. Due to the large amount of material on the subject, I will be brief and will only address

schemes which are suitable for DSP implementation.

A nice introduction to optimal bit synchronization can be found in [24-28] just to name a few. Gardner divides the subject into estimation theoretic (e.g. maximum likelihood), early-late gate, transition tracking, and other. The early-late gate approach is primarily intended for use with square pulses where the SNR is very high. Holmes [24] itemizes a number of suboptimal bit synchronizers which are based upon either delay-and-multiply methods or upon the use of a nonlinearity to obtain a clock component. Ad-hoc methods abound and will not be considered here. We will examine the primary options very briefly.

Estimation Theoretic

Methods grouped into this category are generally capable of near optimal performance and are primarily based upon maximum-likelihood (ML) or maximum a posteriori (MAP) criteria. The MAP-based synchronizer (no ISI) is developed in great detail in [25] and an actual detailed implementation provided in [36]. A fairly broad range of theoretic synchronizers are derived in [29-36].

As stated earlier, derivation of a MAP-based synchronizer is given in [25]. A very concise implementation of a MAP synchronizer for NRZ data is given in [36] and is shown in Figure 4. It is quite straight forward to show [36] that the log-likelihood function in this case is given by

$$\Lambda(\tau) = C_1 \sum_n \ln \cosh\left[\frac{2Ay(\tau)}{N_0}\right] \qquad (12)$$

where
- C_1 a constant
- A Peak signal amplitude, volts
- N_0 One sided noise power spectral density
- $y(\tau)$ Sampled output of matched filter matched to the symbol s(t)

and

$$y(\tau) = \int_0^{T_s} Z(t)\, h(t-nT-\tau)\, dt \qquad (13)$$

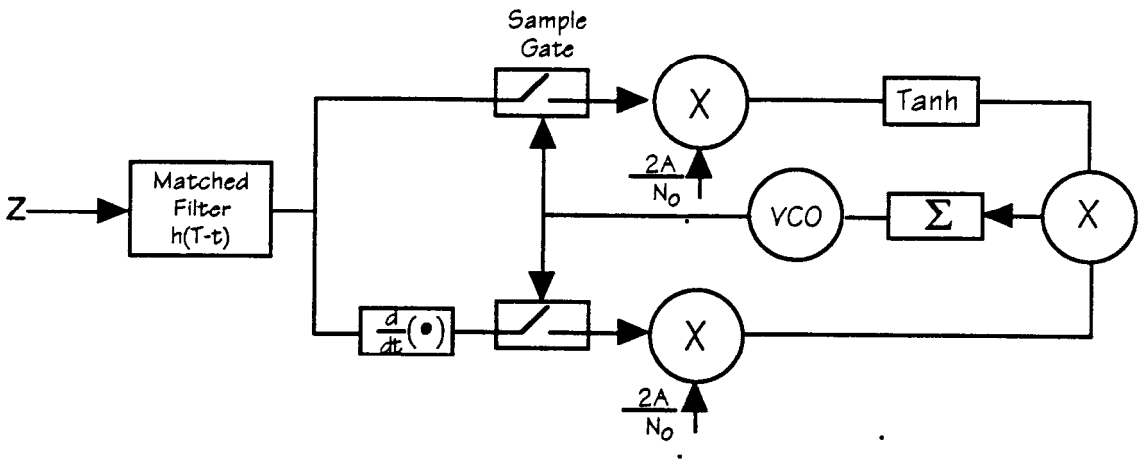


Figure 4 - Map Symbol Synchronizer

where

τ	Timing estimation error
Z(t)	Input signal
h(t)	Matched filter impulse response
T	Symbol period
T_s	Observation time

It is very simple to maximize the likelihood ratio (since ln() is a monotonic function) simply by a differentiating (12) to give

$$\frac{\partial \Lambda(\tau)}{\partial \tau} = C_1 \sum_n \frac{2A}{N_0} \dot{y}(\tau) \tanh\left(\frac{2A}{N_0} y(\tau) \right) = 0 \qquad (14)$$

The loop configuration shown in Figure 4 clearly drives $\partial \Lambda(\tau)/\partial \tau$ to zero.

The cited reference [36] used 8 samples per bit period. The derivative was implemented as

$$y \approx c[\, y(\tau) - y(\tau - T_0) \,] \qquad (15)$$

where T_0 is the sampling period (T/8).

The modified maximum likelihood (MML) synchronizer is discussed in [33] and is sketched out in Figure 5. H_{MF} represents the matched filter, whereas H(f) is a filter designed to extract the data sampling clock. In cases where the matched filter output pulse shape is Nyquist, the data reconstructor is nothing more than a comparator. Filter H(f) is given by

$$H(f) = H_{MF}(f)\left[j2\pi f - \sum_m \frac{\dot{r}(mT)}{r(0)} e^{-j2\pi fmT} \right] \qquad (16)$$

which is nothing more than the derivative of the matched filter output followed by a transversal filter. In the case where $r'(mT) \equiv 0 \ \forall \ m$, this reduces to the ML synchronizer. Time permitting, (16) will be derived in detail.

The minimum-mean-square-error synchronizer is shown in Figure 6. In this case, $H_p(f)$ eliminates ISI as well as maximizes the SNR at its output. Since the ISI has been eliminated, the data reconstruction is very simple. If the output pulse is again Nyquist, $H_p(f)$ reduces to the matched filter $H_{MF}(f)$.

Between MML and MMSE methods, the MMSE is to be much

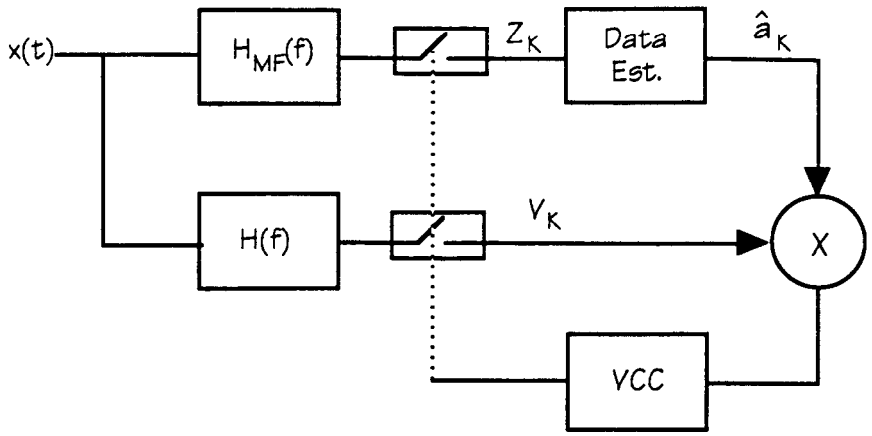


Figure 5 - The MML Synchronizer

preferred since the bandwidths involved here in our situation are considerably less than Nyquist [33].

The data-aided ML and non-data-aided ML synchronizers are analyzed in [35] and are shown in Figures 7a and 7b respectively. When there is no ISI at the output of the matched filter, the ML Data Reconstructor reduces to a simple comparator in which this case looks quite similar to Figure 6.

These previous 4 synchronizer types are compared in [34] for the case of raised-cosine symbols. Unlike in the cited references, [34] considered using each synchronizer presented to perform the timing recovery (only) and the data estimates were made using a matched filter followed by a maximum-likelihood sequence estimator [40]. As far as data estimation, this is optimal.

For the small excess bandwidth case which is applicable for us, the smallest mean-square synchronization error for low SNR occurs for the DA-ML synchronizer [34]. Although this may be true, further examination of other plots in [34] show that much larger synchronizer loop bandwidths may be used for the MML and MMSE synchronizers. Comparing Figures 5 and 6 of [34], there is not much question that the MMSE synchronizer is the best performer given our circumstances.

Selection of the MMSE type as it is presented in [33] is attractive from a DSP standpoint. All of the so-called data reconstructors (Figures 5, 7) are maximum-likelihood sequence estimators (MLSE) which would require more code to implement. Unfortunately, the bit error rate has not been evaluated in [33], only the tracking variance. Exclusion of the MLSE with the MMSE synchronizer will be somewhat suboptimal, but just how much is unknown. I'll have more to say on this in my proposed baseline section.

Performance of these various synchronizers in a fading environment is unknown and would require significant effort to assess.

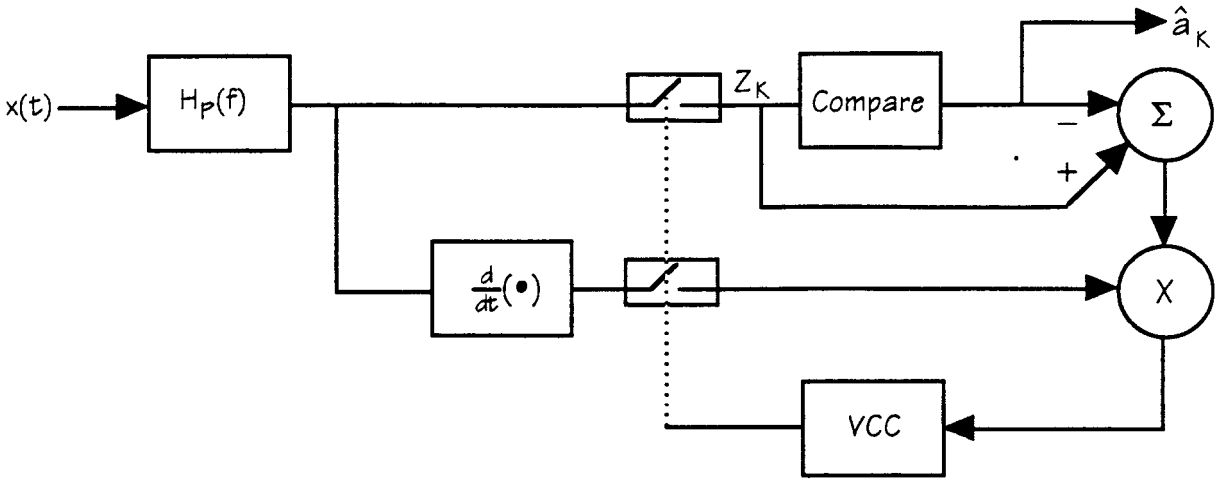


Figure 6 - The MMSE Synchronizer

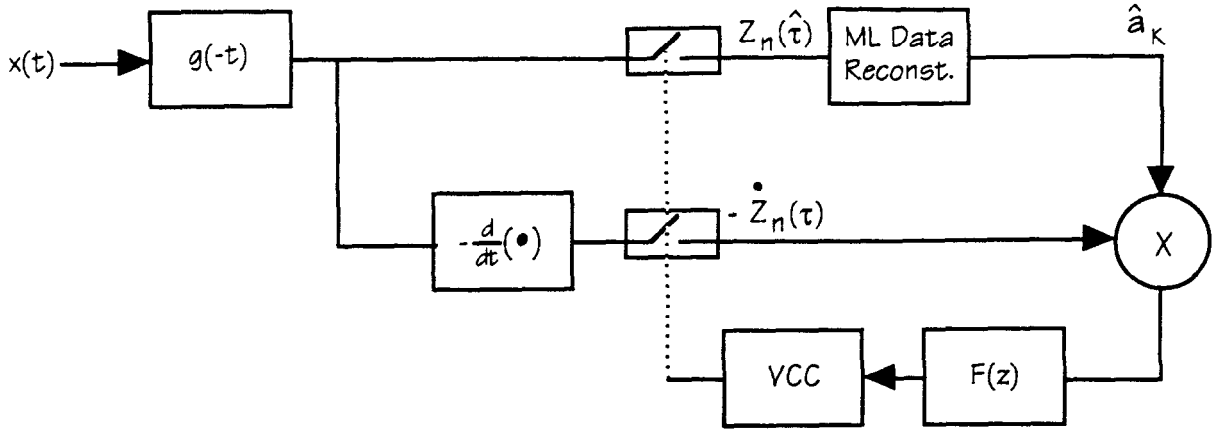


Figure 7a - Data-Aided ML Synchronizer

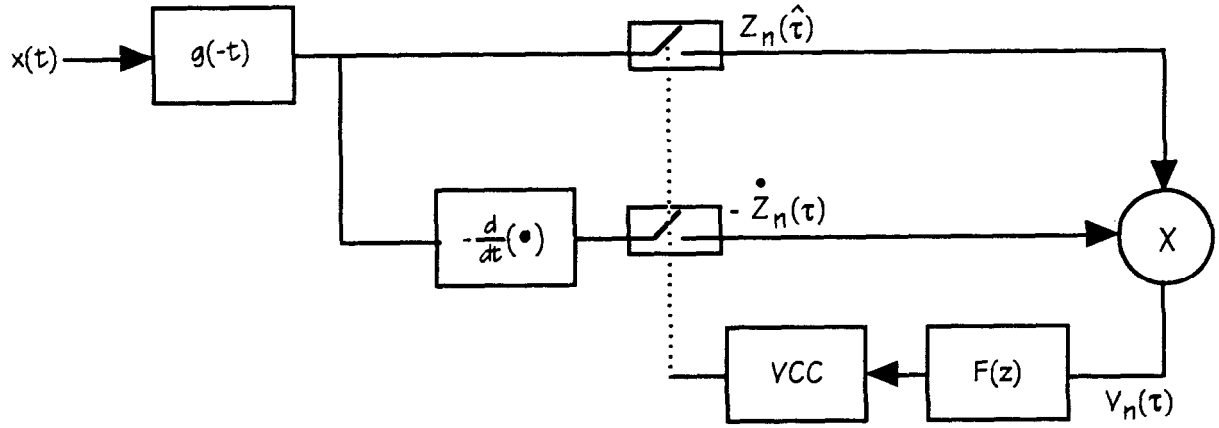


Figure 7b - Non-Data-Aided ML Synchronizer

Transition Tracking

Transition tracking symbol synchronization is discussed in nearly every basic text covering bit synchronizers including [5,6,24,26,28,27] to name a few. A particularly good reference on the subject is [39]. A block diagram for a transition tracking synchronizer (TTS) is shown in Figure 8. Note that the matched filter common to all of the previous synchronizers is missing. It is in fact present in the form of the in-phase and mid-phase integrators in the case of square NRZ pulses. In general, rigorous analysis of this method is very scarce because determination of function zeros is very difficult except for ideal square symbols. Generally, computer simulation must be used to refine the design. As intersymbol interference effects increase, this approach is less and less desirable.

Data Estimate Extraction

Once synchronization has been obtained, the data symbol estimates should be performed in an optimal manner. If there is no ISI, the procedure reduces to a simple comparator as in the MMSE synchronizer, Figure 6. If ISI is present, more elaborate methods must be employed.

If ISI is present (and caused at least partially in the receiver, not only in the transmitter), noise samples taken at each sample point will be correlated. In this case, the optimal approach involves whitening the matched filter outputs using a modified matched filter followed by a maximum likelihood sequence estimator, commonly called a Viterbi receiver. The design of the whitening filter involves a factorization of the noise power spectral density as discussed in many texts. This entire procedure is outlined in the present synchronizer context in [40].

A simple, intuitively obvious method for improved performance in the presence of ISI based upon multi-level decisioning is given in [42]. It is applied specifically to GMSK. Were it not for the digital approach afforded by the TMS320, this technique would be very good, particularly for retro-fitting existing analog radios. References [42,43] are mentioned for completeness.

Discussion at this generalized level could go on without end. I want to put as much time in on my proposed design however, so we will move on. Pertaining to this estimation topic, I omitted a number of other references [44-46]. Also [47-50] on transition tracking.

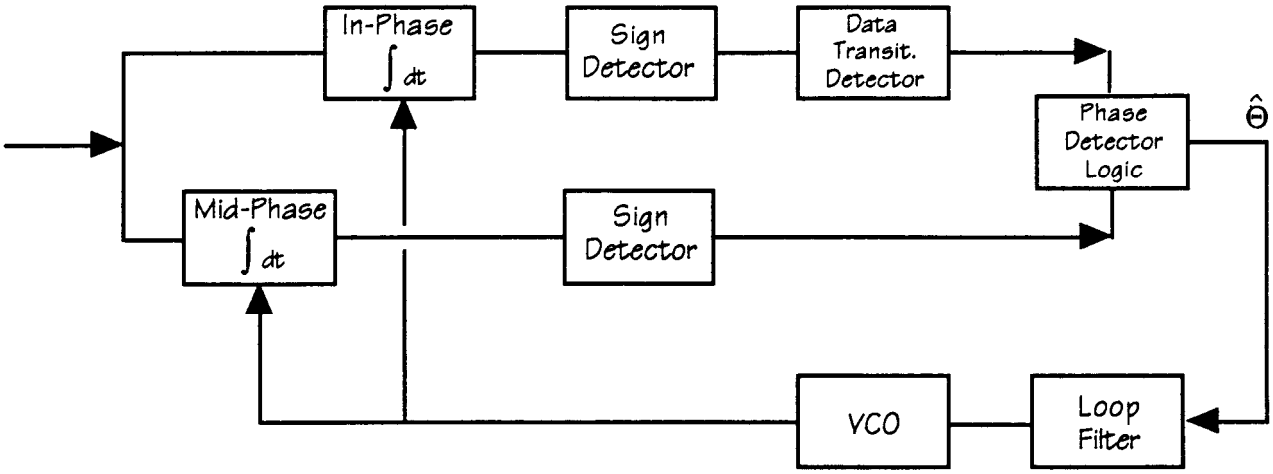


Figure 8 - TT Synchronizer

Part II

Proposed Design Approach

The proposed baseline design is shown in Figure 9. Aside from the Hilbert transform block, this approach looks very similar to the Synetcom baseline, but it is in fact, quite different. I will discuss each block with a fair amount of detail.

Input Amplifier and LPF

The intent here is identical to the Synetcom baseline, specifically, obtaining the proper level into the A/D and elimination of alias components.

One point just occurred to me. At our lunch, you said you were planning to down-convert from 10.7 MHz to 8 kHz. Unless the receiver selectivity ahead of your modem is incredibly sharp, adjacent channels only 2X f_{IF} = 16 kHz away will be down-converted along with your desired signal. This fact alone leads me to believe that you should (must) move your second IF frequency up at least some.

Not to beat a dead horse here, but when you down-convert from 10.7 MHz to whatever, some of that local oscillator energy is going to make its way back toward the receiver. This is an important reason to probably make amplifier A1 in Figure 9 fairly high in reverse isolation. There is an additional good reason for increasing the IF frequency which I'll get to in a moment.

A/D Converter

Pretty basic here. Since you have the word width in the DSP available, you might as well use as many bits as you can muster. A sample rate of 48 ksps is to be used.

Rx Bandpass Filter

At present, your baseline (Synetcom) calls for a 12-tap FIR filter. I haven't designed a lot of digital FIR filters, but I assume that this will be sufficient.

This filter needs to have linear phase in order to not distort the demodulated eye pattern or increase intersymbol interference. It needs to pass the desired signal plus frequency error (± 1.5 kHz) and no more (!!) because its noise bandwidth basically sets the predetection signal to noise

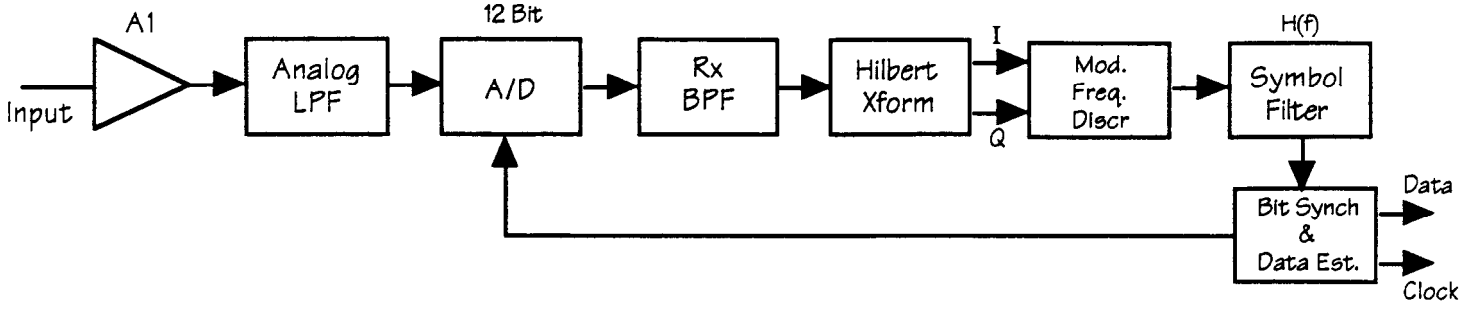


Figure 9 - Proposed Baseline Design

ratio into the demodulator.

From our earlier discussions, I believe at the transmit end the normalized 3 dB bandwidth of the Gaussian premodulation filter is $(B_bT) = 0.3$ which is slightly suboptimal [8, p. 494]. The occupied RF bandwidth is roughly [8, p. 495]

90% 0.57 x 8 kHz = 4.56 kHz
99% 0.86 x 8 kHz = 6.88 kHz

We ideally need to know the optimum bandwidth for this filter to minimize the overall system bit error rate. Unfortunately, I have no references which discuss results for this problem. At the present time, I would recommend that Rx BPF be linear phase with a bandwidth of 6.88 + 2 x 1.5 = 9.88 kHz since (i) the factor of 0.86 is actually for $B_bT = 0.25$ whereas we are using $B_bT = 0.3$ and (ii) accommodate the ±1.5 kHz frequency error. (Accommodating the ± 1.5 kHz frequency error results in a 1.6 dB predetection noise increase.)

Rx BPF Linear Phase
Bandwidth 9.9 kHz

This is one area which requires further optimization.

Hilbert Transformer

There are a number of reasons for using the Hilbert transform. Even if you were to stay with your baseline 2-filter/envelope detector discriminator, use of the Hilbert transform rather than just the square-law detector eliminates the need for any lowpass filter following the square-law detector. In short, the envelope detector using the Hilbert transformer would look like Figure 10. This approach is being used more and more in high-end instrumentation, e.g. [52].

Use of the Hilbert transform provides direct extraction of the in-phase (I) and quadrature (Q) signal phases from which a frequency discriminator can be formed well suited to this project. I'll discuss that topic shortly.

A fairly large amount of literature is available on performing and implementing Hilbert transforms. Even though I do not profess to be heavily into DSP, I have a number of references on the subject. Chapter 3 of [53] provides some high-level aspects. Reference [54] goes into much greater detail on the subject, derives the required impulse function given as

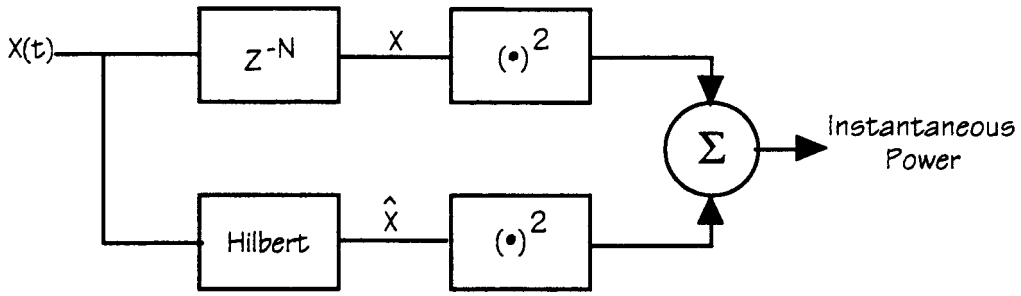


Figure 10 - Envelope Detector

$$h_d(n) = \left\{ \begin{array}{ll} \frac{2}{\pi} \sin^2(\pi n/2); & n \neq 0 \\ 0; & n=0 \end{array} \right\} \qquad (17)$$

and also discusses use of the Remez algorithm for design of a Hilbert transform filter. The ideal filter is specified a

$$H_d(\omega) = \left\{ \begin{array}{ll} -j & 0 \leq \omega \leq \pi \\ j & -\pi < \omega < 0 \end{array} \right\} \qquad (18)$$

For a bandpass signal as in our case, an alternative definition given as

$$H_d(f) = \left\{ \begin{array}{ll} -j & 2\pi f_L \leq \omega \leq 2\pi f_H \\ j & 2\pi(1-f_H) \leq \omega \leq 2\pi(1-f_L) \end{array} \right\} \qquad (19)$$

is also acceptable and would probably lead to simpler filters. As the signal bandwidth to center frequency ratio decreases, frequency sampling realizations of $H_d(f)$ are preferred compared FIR realizations. The same is true for the Rx BPF.

An alternate implementation approach for obtaining the I and Q sample streams (i.e., performing the Hilbert transform operation) which utilizes a higher IF is described in [55]. This approach utilizes bandpass sampling techniques to extract I and Q by using a sampling rate 4X the intermediate frequency. This is overly wasteful however. It can be shown that use of a sampling rate of 4/3 times the IF suffices. The "new" approach is shown in Figure 11.

In order to reinforce the underlying concepts in use, assume that the input signal x(t) may be written in standard form where

$$x(t) = A \cos[\omega_o t + \Delta\omega t + \theta(t)] \qquad (20)$$

where ω_0 Radian IF
$\Delta\omega$ Frequency error, rad/sec
$\theta(t)$ Modulation

Expanding this,

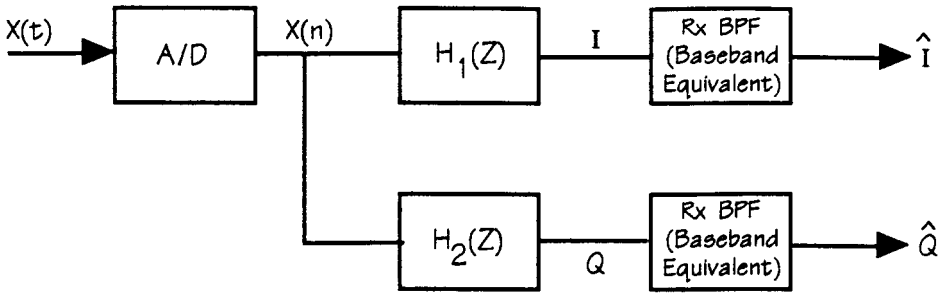


Figure 11 - Extraction of I and Q

$$x(t) = A\cos[\,\Delta\omega t + \theta(t)\,]\cos(\omega_o t) - A\sin[\Delta\omega t + \theta(t)\,]\sin(\omega_o t) \tag{21}$$

If we now admit the use of pre-envelopes (i.e., complex signals), we may write

$$\begin{aligned} x(t) &= \Re\{[\,I(t)+jQ(t)\,]\,e^{j\omega_o t}\} \\ &= \Re\left\{\begin{array}{l} I(t)\cos(\omega_o t) + jI(t)\sin(\omega_o t) + \\ jQ(t)\cos(\omega_o t) - Q(t)\sin(\omega_o t) \end{array}\right\} \\ &= I(t)\cos(\omega_o t) - Q(t)\sin(\omega_o t) \end{aligned} \tag{22}$$

where clearly then in (21)

$$I(t) = A\cos[\,\Delta\omega t + \theta(t)\,] \tag{23}$$

$$Q(t) = A\sin[\,\Delta\omega t + \theta(t)\,] \tag{24}$$

If we now go back and let

$$Z(t) = [\,I(t) + jQ(t)\,]\,e^{j\omega_o t} \tag{25}$$

and take the sampling period as $2\pi/(\omega_0 4/3)$

$$T_s = \frac{2\pi}{\omega_0}\,\frac{3}{4} = \frac{3\pi}{2\omega_0}\ \text{sec.} \tag{26}$$

then

$$
\begin{aligned}
Z(0) &= I(0) + jQ(0) \\
Z(T_s) &= [\, I(T_s) + jQ(T_s) \,]\,(-j) \\
&= -jI(T_s) + Q(T_s) \\
Z(2T_s) &= [\, I(2T_s) + jQ(2T_s) \,]\,(-1) \\
&= -I(2T_s) - jQ(2T_s) \\
Z(3T_s) &= [\, I(3T_s) + jQ(3T_s) \,]\,(j) \\
&= jI(3T_s) - Q(3T_s)
\end{aligned}
\tag{27}
$$

Taking the real parts of (27), we obtain

$$
\begin{aligned}
x(0) &= I(0) \\
x(T_s) &= Q(T_s) \\
x(2T_s) &= -I(2T_s) \\
x(3T_s) &= -Q(T_s)
\end{aligned}
\tag{28}
$$

Clearly, even samples give the in-phase sample values whereas odd samples provide the quadrature-phase sample values.

The primary reason for requiring $H_1(z)$ and $H_2(z)$ in Figure 11 is to in effect interpolate (appropriately) between I and Q samples such that I and Q samples are <u>simultaneously</u> available at the same time instant. (In (27), I and Q samples are delayed from each other by T_s.) To first-order, we could ad-hoc propose the following:

$$Q(T_s) = x(T_s)$$

$$I(T_s) \approx \frac{x(0) - x(2T_s)}{2}$$

$$Q(2T_s) \approx \frac{x(T_s) - x(3T_s)}{2}$$

$$I(2T_s) = -x(2T_s) \qquad (29)$$

$$Q(3T_s) = -x(3T_s)$$

$$I(3T_s) \approx \frac{-x(2T_s) + x(4T_s)}{2}$$

etc.

This can in fact be used quite effectively as the ratio of signal bandwidth to center frequency decreases. The approach represented by (29) could take the form of Figure 12.

At any sampling time, the transfer functions (z domain) between x(n) and the respective output are given by one of the functions

$$\pm z^{-1} \qquad (30)$$

$$\pm \frac{z^{-2} - 1}{2} \qquad (31)$$

Note that the relative phase between these is given by (ignoring leading ± signs)

$$\angle \left[\frac{z^{-2} - 1}{2} z \right] = \angle \left[\frac{z^{-1} - z}{2} \right] \qquad (32)$$

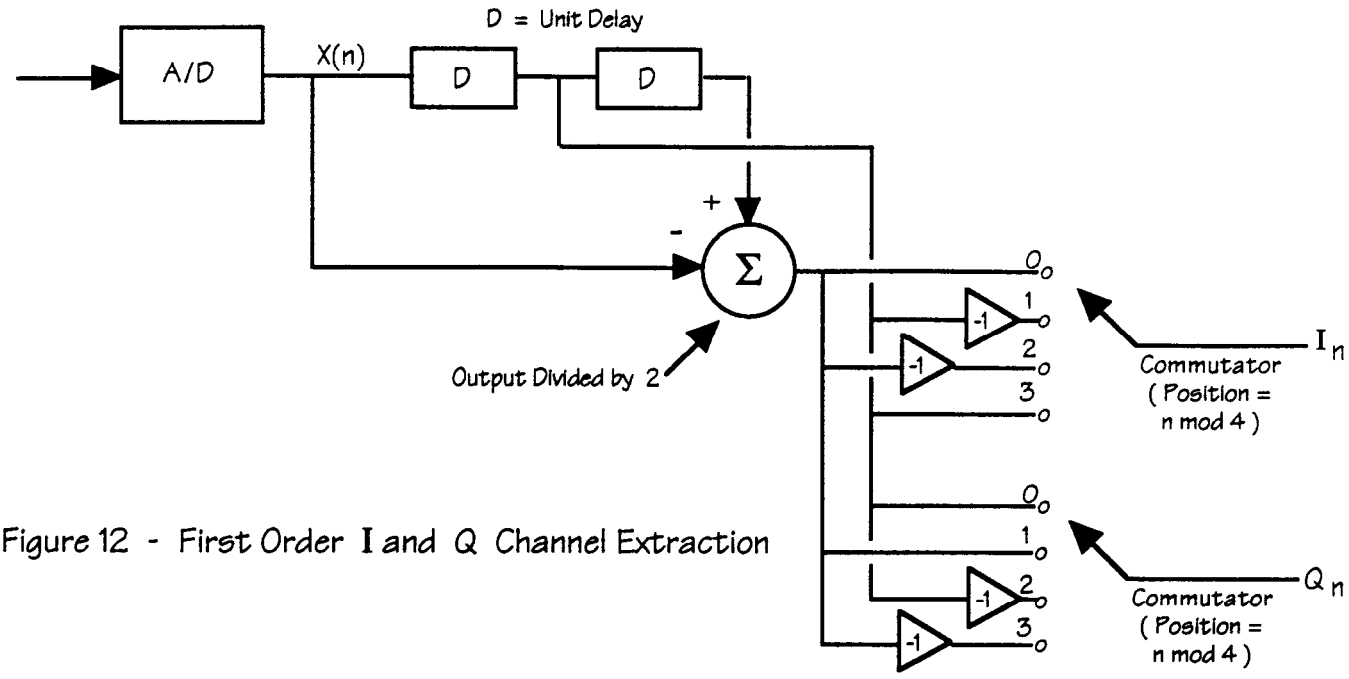


Figure 12 - First Order I and Q Channel Extraction

If the signal center frequency ω_o is instead slightly offset and is given by $\omega_o + \Delta\omega$, then (32) becomes

$$\begin{aligned} z &= e^{j(\omega_o + \Delta\omega)T_s} \\ &= e^{j(\omega_o+\Delta\omega)\frac{3\pi}{2\omega_o}} \\ &= e^{j\alpha 3\frac{\pi}{2}} \quad where\ \alpha = 1+\frac{\Delta\omega}{\omega_o} \end{aligned} \tag{33}$$

leading to

$$\begin{aligned} \angle\left[\frac{z^{-1} - z}{2}\right] &= \angle\ \frac{[e^{-j\alpha 3\pi/2} - e^{j\alpha 3\pi/2}]}{2} \\ &= \angle\,[-j\ \sin(\ 3\alpha\pi/2\ )] \approx 90\ ^\circ \end{aligned} \tag{34}$$

since $\sin(\alpha 3\pi/2) \approx -1$. Hence, this is in fact performing the desired Hilbert transformation. Note that if $\Delta\omega \neq 0$, then $\alpha \neq 1$ and there is a magnitude mismatch between (30) and (31) given by

$$\Delta_{dB} = 20\ Log\left|\sin\left(\frac{\alpha 3\pi}{2}\right)\right| \tag{35}$$

To see what kind of impact this causes, assume that $\Delta\omega$ = 1.5 kHz (2π), $\omega_o = 2\pi$ (36 kHz), and the lower deviation tone is being received. In this case,

$$\alpha = 1 + \frac{1.5 + 2}{36} = 1.09722 \tag{36}$$

from which

$$\Delta_{dB} = -0.945\ dB$$

This, in fact, is not too bad and may well be acceptable. This approach is superior to some in that although the amplitude mismatch (36) results, (i) the commutation in Figure 12 averages things out and (ii) the channels remain on the average in quadrature (34). The downside is that this much error introduces a healthy phase error (i.e. $\theta = \tan^{-1}(Q/I)$). When you add all the frequency uncertainties in the system up, (36) may be too optimistic.

It should not be too difficult to improve upon (30) and (31), i.e. $H_1(z)$ and $H_2(z)$ in Figure 11. In fact, the solution appears almost readily available in [55]. In [55], the sampling rate was again $4\omega_o/2\pi$ in which case $T_s' = (4\omega_o/2\pi)^{-1}$. Returning to (25),

$$Z(t) = \{ [\, I(t) + jQ(t) \,]\, e^{j\omega_o t} \}_{t=nT_s'}$$

$$Z(nT_s) = \left[\, I(nT_s') + jQ(nT_s') \,\right] e^{j\omega_o \frac{2\pi n}{4\omega_o}} \tag{37}$$

Performing the equivalent of (30) here,

$$\begin{aligned} x(0) &= I(0) = I(0) \\ x(T_s') &= -Q(T_s') = -Q(T_s/3) \\ x(2T_s') &= -I(2T_s') = -I(2T_s/3) \\ x(3T_s') &= Q(3T_s') = Q(T_s) \end{aligned} \tag{38}$$

Comparing (38) and (30), only two signs must be changed and we can use the results from [55] which follow:

$$H_1(z) = z^{-1}\,\frac{z^{-2} - a^2}{1 - a^2 z^{-2}}; \qquad a^2 = 0.5846832 \tag{39}$$

$$H_2(z) = -\left[\frac{z^2 - b^2}{1 - b^2 z^{-2}}\right]; \qquad b^2 = 0.1380250 \tag{40}$$

Therefore, aside from accounting for the sign differences between (30) and (38), (39) and (40) may be used directly. These filters ((39) and (40)) are more than sufficient for this task, being all-pass (no gain imbalances) with a departure from quadrature of less than 0.1145°[1] .

FM Demodulator

We've been specifically directed to use a non-coherent demodulator and that is fine. The principle paper I want to sight for FM demodulation using I and Q is [56]. This reference goes into substantial detail which we cannot repeat here.

Let the received signal be given in terms of its complex envelope form,

$$r(t) = I(t) + jQ(t) \tag{41}$$

Rewriting this in polar form we obtain

$$r(t) = \sqrt{I^2(t) + Q^2(t)} \; \angle \tan^{-1}\left[\frac{Q(t)}{I(t)}\right] \tag{42}$$

Throwing away the amplitude term (e.g., hard-limit, set $\sqrt{} = 1$), the instantaneous phase from (42) is

$$\theta(t) = \tan^{-1}\left[\frac{Q(t)}{I(t)}\right] \tag{43}$$

Since frequency is the derivative of phase, and we want to frequency demodulate, the instantaneous frequency is given by

---

[1] Had we used the more simple H_1 and H_2 from (35), it should be clear that a low IF (8 kHz) is much less attractive than the higher 36 kHz IF. This justification for going to a higher IF (36 kHz) disappears when we admit (39) and (40) into the solution.

$$\omega(t) = \frac{d\theta}{dt} = \frac{I(t)\dot{Q}(t) - Q(t)\dot{I}(t)}{I^2(t) + Q^2(t)} \tag{44}$$

In terms of discrete samples then

$$\omega(nT_s) = \frac{I(nT_s)\dot{Q}(nT_s) - Q(nT_s)\dot{I}(nT_s)}{I^2(nT_s) + Q^2(nT_s)} \tag{45}$$

If we may be so bold as to approximate the derivatives as finite differences, dropping the nT_s notation in favor of subscripts and taking $T_s = 1$,

$$\dot{Q}_n \approx \frac{Q_n - Q_{n-1}}{2} \tag{46}$$

$$\dot{I}_n \approx \frac{I_n - I_{n-1}}{2} \tag{47}$$

Substituting into (45) we obtain

$$\omega_k \approx \frac{1}{2} \frac{I_k(Q_k - Q_{k-1}) - Q_k(I_k - I_{k-1})}{I_k^2 + Q_k^2} \tag{48}$$

$$\approx \frac{1}{2} \frac{Q_k I_{k-1} - I_k Q_{k-1}}{I_k^2 + Q_k^2} \tag{49}$$

Aside from the implied division, this is a very nice form for DSP implementation. If your planned oversampling rate were higher, use of (45) would be fine. It is probably wiser to implement a better estimate of the derivative as sketched in Figure 13 since your oversampling rate is not large.

Assume that we want to estimate the derivative at t=0. A simple second-order interpolating polynomial which can be used is

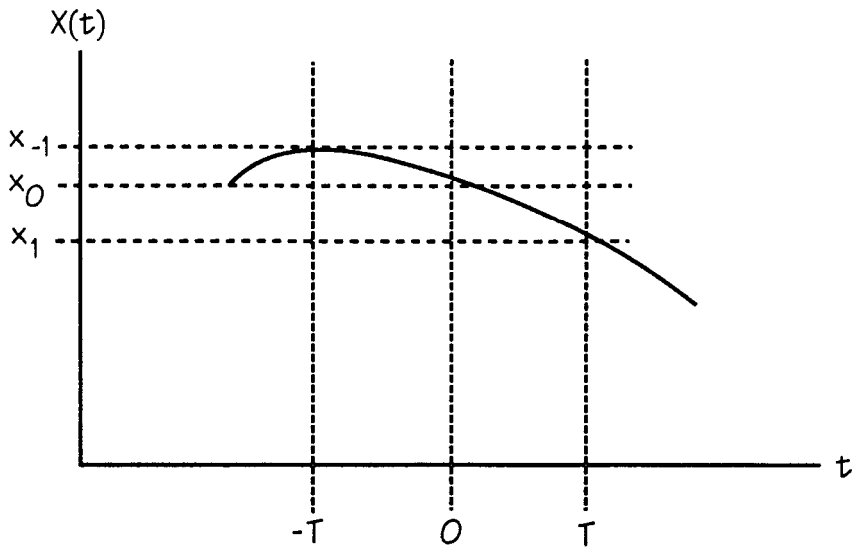


Figure 13 - Estimating the Derivative

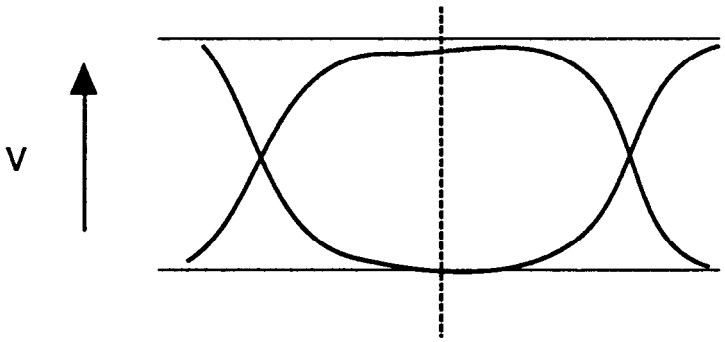


Figure 14

$$x(t) = a\,t^2 + bt + c \tag{50}$$

Evaluating (50) at each point in Figure 13 we obtain

$$\begin{aligned} x_{-1} &= a\,T^2 - bT + c \\ x_0 &= c \\ x_1 &= a\,T^2 + bT + c \end{aligned} \tag{51}$$

Solving for the polynomial coefficients,

$$\begin{aligned} c &= x_0 \\ b &= \frac{x_1 - x_{-1}}{2T} \\ a &= \frac{x_1 + x_{-1}}{2} - x_0 \end{aligned} \tag{52}$$

Differentiating (50) at t=0,

$$\dot{x}(0) = b = \frac{x_1 - x_{-1}}{2T} \tag{53}$$

Using this result in (45) leads us to (taking T=1)

$$\omega_k = \frac{I_k\left(\frac{Q_{k+1} - Q_{k-1}}{2}\right) - Q_k\left(\frac{I_{k+1} - I_{k-1}}{2}\right)}{I_k^2 + Q_k^2} \tag{54}$$

This result is clearly more complicated than (48) but it should provide significantly better results. Higher order interpolating polynomials may of course be used in (51) if desired.

Assuming that we adopt (54) as our baseline FM demodulator, let's focus on the denominator of (54). Although the click-effect is a major noise contributor in conventional digital FM, it does not contribute significantly to errors for

low frequency deviation systems [57]. Even so, since it has also been demonstrated [58] that the PLL improves the FM noise threshold by not tracking many of these "clicks" which would normally appear at the output of a limiter-discriminator, Park's paper [56] should be seriously considered. Park advocates augmenting the denominator term in (54) and (48) as

$$I_k^2 + Q_k^2 \rightarrow I_k^2 + Q_k^2 + C \tag{55}$$

where C is a positive constant. This constant prevents the denominator from ever becoming zero which is the source of the noise click problem. This modification is known as the hybrid detector in [56].

Park is less than complete in making any recommendations for the constant C. Its determination will require further work. An adaptive constant C could also prove fruitful provided that it could be rapidly updated in the presence of fast fading. I recommend setting C to a constant value which is a compromise between high and low SNR.

Bit Synchronization and Data Sequence Estimation

Bit synchronization is unquestionably the most difficult task in this project. The noncoherent demodulation assumption has fortunately simplified matters in that area considerably.

Results published in the literature are fine for initial concept development, but very rarely can they be viewed as a complete design recipe. The primary design anomalies in our case are (i) the short bit synchronization pattern and (ii) the large frequency uncertainty. The frequency error causes a substantial D.C. component out of the demodulator which must be removed (i.e., high-pass filter) or estimated before good synchronization can occur.

To examine this D.C. component problem further, consider a CW signal which is offset from the desired IF by $\Delta\omega$. It may be represented as

$$\begin{aligned} r(t) &= A\cos[\,(\omega_o+\Delta\omega)\,t + \theta_o] \\ &= A\cos[\Delta\omega t + \theta_o]\,\cos(\omega_o t) \\ &\quad - A\sin[\Delta\omega t + \theta_o]\,\sin(\omega_o t) \end{aligned} \tag{56}$$

where

$$I(t) = A\cos[\,\Delta\omega t + \theta_o\,]$$
$$Q(t) = A\sin[\,\Delta\omega t + \theta_o\,] \qquad (57)$$

Substituting into (44) we obtain

$$\omega = \frac{A^2\cos^2[\Delta\omega t + \theta_0]\;\Delta\omega + A^2\sin^2[\Delta\omega t + \theta_0]\;\Delta\omega}{A^2\cos^2[\Delta\omega t + \theta_0] + A^2\sin^2[\Delta\omega t + \theta_0]} \qquad (58)$$
$$= \Delta\omega$$

which simply shows that $\Delta\omega$ results in a D.C. output term from the demodulator. A (fixed) high-pass filter is not suitable for removal of the D.C. term because if it settles in a reasonable fraction of the bit sync frame (say 4-6 symbols), it will also erroneously track strings of ones and zeros later.

I believe that this D.C. component problem forces us to have separate acquisition and track modes for the bit synchronizer. Reliable determination of signal presence then becomes a major concern.

In reviewing the baseline design for this portion which follows, I am concerned about the complexity level to a degree. Unfortunately, short design times rarely yield elegant simple solutions. I will try to comment on the complexity aspect wherever possible.

MMSE Concept

Consider the eye pattern shown in Figure 14. The sampling instant within the eye is shown with the dotted line. The MMSE synchronizer attempts to minimize the squared-error at the sampling instant which is given by

$$\varepsilon_k = (\,v_k - a_k\overline{v}_k - v_{dc}\,)^2 \qquad (59)$$

for each k^{th} symbol.

Since the transmitter and receiver functions are to be implemented digitally, the modulation index and receiver demodulator sensitivity should be a constant making the mean value of v_k a constant v_o.

Over a block of N symbols, the mean-square error is then

$$\sigma^2 = \frac{\sum_{k=1}^{N} (v_k - a_k v_o - v_{dc})^2}{N} \qquad (60)$$

Minimizing σ^2 with respect to the sampling time error τ (data decisions a_k assumed correct),

$$\frac{\partial \sigma^2}{\partial \tau} = \frac{1}{N} \sum_{k=1}^{N} 2 (v_k - a_k v_0 - v_{dc}) \frac{dv_k}{d\tau} = 0 \qquad (61)$$

Aside from the averaging due to the summation and 1/N factor, this is precisely the structure shown in Figure 6.

A couple of points are necessary. First, v_{dc} must clearly be well known before (61) can be used, or for that matter, before the a_k can be properly estimated. On the other hand (secondly), $dv_k/d\tau$ will on the average be zero at the eye center.

Focusing upon the data pattern used during the bit synchronization frame, due to band limiting, the demodulator output during that period will be a sine wave of period 4T superimposed on an unknown D.C. component. Hence, the MSEE synchronizer can be made to operate as a sinusoidal phase-locked loop during the synchronization frame by simply disabling the comparator branch of Figure 6.

Determination of signal presence with the MMSE scheme really scares me since this operation must somehow be done with all of the others in parallel, and the synch frame is so short with really only 7 zero crossings <u>maximum</u> to work with The complexity factor is also growing which I do not like either. During acquisition in the bit synch portion, we must:

* 1) Match filter
2) Equalize matched filter output
* 3) Estimate v_{dc} on a moving block 4T long
4) Differentiate the equalizer output
5) Determine signal presence based upon a scant few 7 (max) zero crossings

and

6) Hand off to track just right.

The starred items above are unavoidable, but there is a better way I believe by returning to basics.

The baseline bit synchronizer which I will now propose is shown in Figure 15. The blocks containing * are only implemented during the initial acquisition phase.

The symbol matched filter is precisely that, matched to the received pulse shape.

The heart of the acquisition process is the preamble matched filter $H_p(f)$. It is matched to a tapered sine wave which is roughly the length of the bit synchronization frame (16 bits). Since the matched filter by definition maximizes the output SNR, this filter combined with a threshold detector is the optimal manner for determining signal presence.

Once the detection threshold has been crossed, the DSP monitors the output from $H_p(f)$ to determine precisely at what time the correlation peak occurs. From this time, the synchronizer VCO may be properly preset.

The D.C. estimation block performs a running block average of the demodulator output closely corresponding to the width of the correlator (Figure 16). Its value is taken at the peak correlation occurrence and used as the D.C. estimate for the remainder of the block reception. Analysis will verify whether this short term block average is suitably accurate.

The MLSE block is in actuality a Viterbi Receiver. I will defer comment on this until we get into the details.

The transition detector is used within the loop structure to track mid-bit transitions as described earlier. There are a number of reasons for advocating this approach over the MMSE method:

1) It is more simple.
2) It is an unbiased detector as well. Although it was not compared in [34], I believe it is one of the best.
3) First-hand experience even in the presence of heavy ISI has proven that it is a very robust method.
4) The zero-crossing statistics which may be needed for determination of lock are a natural by-product.

The zero-crossing statistics block may or may not be that necessary, but is included for completeness in Figure 15. It

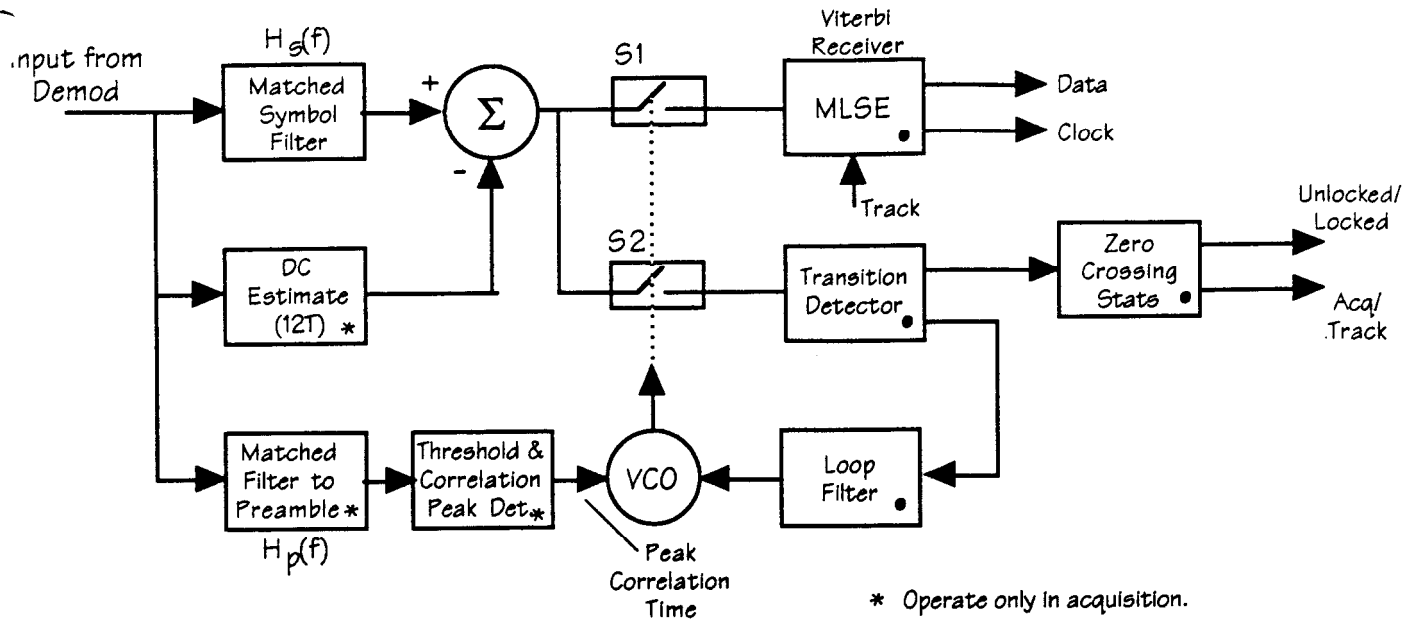


Figure 15 - Baseline Bit Synchronizer

is clear from [1] that all of the received data is block oriented and of a given length (due to the block-level interleaving being performed). Therefore, end-of-message could be determined by simply counting symbols which occur after the frame synchronization block. On the other hand, if the initial acquisition false-alarm rate is too high, or if it is desireable to determine loss-of-lock in mid-message, this function may be used.

The zero-crossing block simply keeps track of the time between zero-crossings modulo 2π. If this quantity is very dispersed over a given number of symbols, lock has been lost, and acquisition should commence again.

The loop filter and VCO will be discussed in the detailed sections which follow.

Detailed Discussion

Acquisition

The heart of the acquisition process is the preamble matched filter, $H_p(f)$, in Figure 15. Signal presence and initial bit synchronizer timing are determined from observations of the filter output.

Considering the bit synchronization frame from the MOB, we have a near sinusoidal output from the demodulator judging from the eye diagram, Figure 3 of [1]. Since the sine wave frequency is 1/4T = 2 KHz (0011 pattern), it will be well contained within any of the bandpass filters within the modem (e.g. Rx BPF Figure 9). I will assume that we basically see a sine wave at the demodulator output. This signal may be represented by

$$r(t) = A \sin(\omega_o t) \Pi(t) \tag{62}$$

where

$$\begin{aligned} \Pi(t) &= 1 \qquad 0 \le t \le 16T \\ &= 0 \qquad \textit{Otherwise} \end{aligned} \tag{63}$$

Assume now that $H_p(f)$ is equivalent to correlating r(t) in the time domain with

$$h(t) = \sin(\omega_o t) \alpha(t) \tag{64}$$

where $\alpha(t)$ is an arbitrary shaping window which is zero for $t<0$ and $t>16T$.

For the matched case, let $\alpha(t) = \sqcap(t)/(8T)$. The resulting correlator output is

$$C(\tau) = \frac{1}{8T} \int_0^{T_a} A \sin[\omega_0 (t - \tau)] \Pi(t - \tau) \sin(\omega_0 t) \, dt$$

where τ is the unknown signal timing which leads to

$$C(\tau) = \frac{A}{T_m} \int_0^{T_m} [\cos(\omega_0\tau) - \cos(2\omega_0 t - \omega_0\tau)] \prod(t - \tau)\, dt \qquad (65)$$

with $T_m = 16T$. Then for $0 \leq \tau \leq T_m$,

$$C(\tau) = \frac{A}{T_m} \int_\tau^{T_m} \cos(\omega_0\tau) - \cos(2\omega_0 t - \omega_0\tau)\, dt$$

$$C(\tau) = \frac{A}{T_m} \cos(\omega_0\tau)\,[T_m - \tau] - \frac{A}{T_m}\left[\frac{\sin(2\omega_0 t - \omega_0\tau)}{2\omega_0}\right]_\tau^{T_m} \qquad (66)$$

Since $\omega_0 T_m = N2\pi$,

$$C(\tau) = A\cos(\omega_0\tau)\left(1 - \frac{\tau}{T_m}\right) - \frac{A}{T_m}\left[\frac{\sin(-\omega_0\tau) - \sin(\omega_0\tau)}{2\omega_0}\right]$$

$$C(\tau) = A\cos(\omega_0\tau)\left(1 - \frac{\tau}{T_m}\right) + \frac{A}{T_m\omega_0}\sin(\omega_0\tau) \qquad (67)$$

$$C(\tau) = A\left[\left(1 - \frac{\tau}{T_m}\right)\cos(\omega_0\tau) + \frac{\sin(\omega_0\tau)}{\omega_0 T_m}\right] \qquad (68)$$

Given that $T_m = 16T$, $\omega_0 = 2\pi/(4T)$, with A = 1, this becomes

$$C(\tau) = \left(1 - \frac{\tau}{16T}\right) \cos\left(\frac{\pi}{2}\frac{\tau}{T}\right) + \frac{\sin\left(\frac{\pi}{2}\frac{\tau}{T}\right)}{8\pi} \tag{69}$$

and letting $\gamma = \tau/T$,

$$C(\gamma) = \left(1 - \frac{\gamma}{16}\right) \cos\left(\frac{\pi}{2}\gamma\right) + \frac{1}{8\pi}\sin\left(\frac{\pi}{2}\gamma\right) \tag{70}$$

This result (70) is shown in Figure 16, and is symmetric for $\tau < 0$. The sidelobe structure is very apparent.

When I first proposed this approach, I assumed that we would simply look for the correlation peak in order to determine course timing synchronization. But since $C'(\tau) = 0$ at the correlation peak, this is clearly not what should be done. Rather, it is much more accurate to determine course synchronization from the zero-crossing on either side of the correlation peak. Since h(t) (impulse response of $H_p(f)$) will pass no D.C. component, this transition should be easy to locate.

The noise bandwidth corresponding to this h(t) is

$$B_n = \frac{\int_{-\infty}^{\infty} |H(f)|^2 \, df}{2 \max |H(f)|^2} = \frac{\int_{-\infty}^{\infty} h^2(t) \, dt}{2 \max |H(f)|^2} \tag{71}$$

Here from (64) with $\alpha(t) = \prod(t)$,

$$H(f) = \frac{1}{jT_m}\left[\delta(f - f_0) - \delta(f + f_0) \right] \otimes \left[\frac{1 - e^{-j2\pi f T_m}}{j2\pi f}\right]$$

$$= \frac{1}{j}\left[W(f - f_0) - W(f + f_0) \right] \tag{72}$$

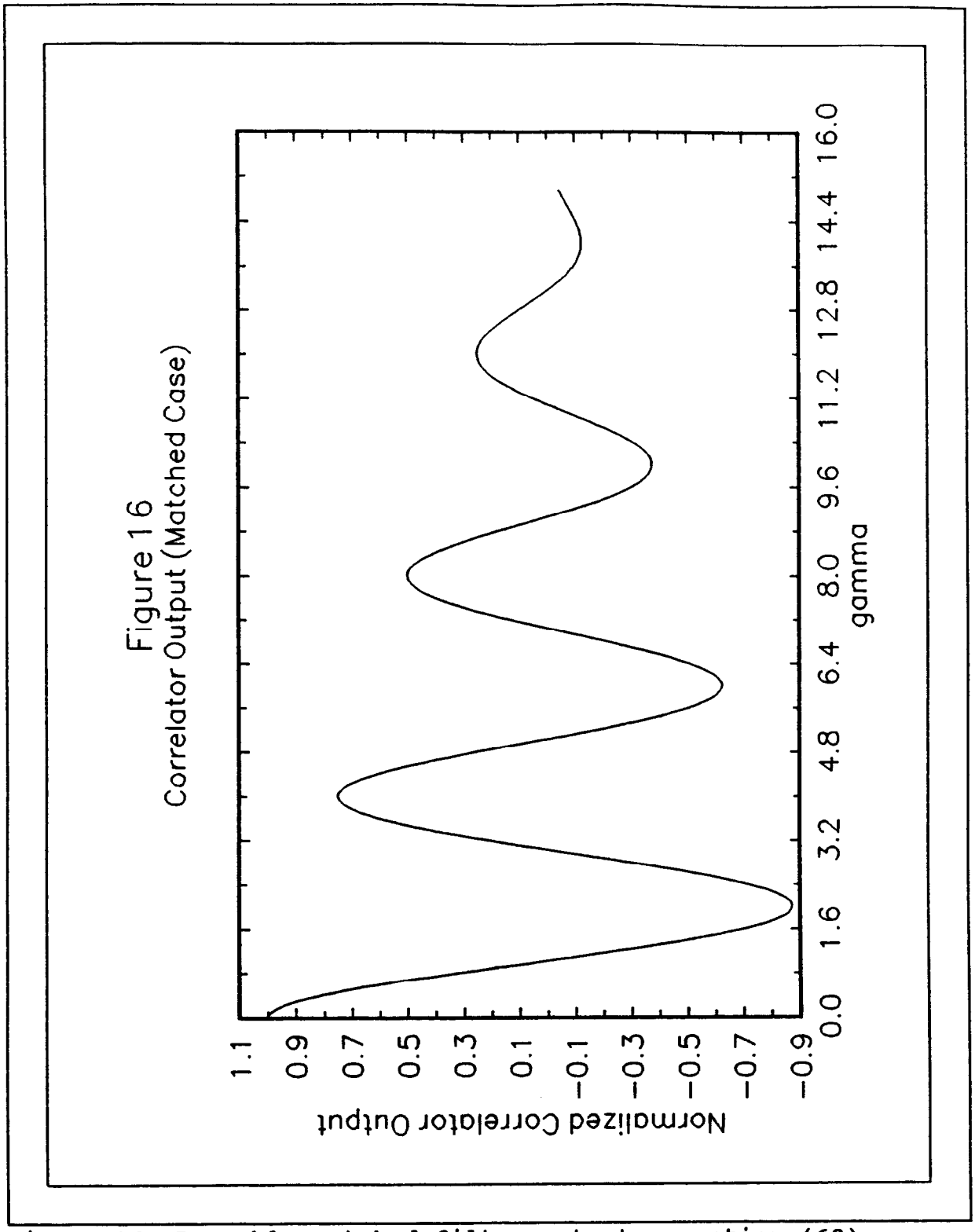


Figure 16 Preamble matched filter output, equation (68).

where

$$W(f) = \frac{\sin(\pi f T_m)}{\pi f T_m} e^{-j\pi f T_m} \tag{73}$$

Therefore, max $|H(f)|^2 = 1$. Continuing with this

$$\begin{aligned} B_n &= \frac{1}{2}\int_{-\infty}^{\infty} h^2(t)\, dt \\ &= \frac{1}{2}\int_{0}^{T_m} \sin^2(\omega_o t)\left(\frac{1}{8T}\right)^2 dt \\ &= \frac{1}{4}T_m\left(\frac{2}{T_m}\right)^2 = \frac{1}{T_m}\ Hz \end{aligned} \tag{74}$$

where this is a one-sided noise bandwidth. Therefore, assuming AWGN at the demodulator output, the variance at the correlator output is

$$\sigma_n^2 = N_o B_n \tag{75}$$

where N_o is the one-sided noise power spectral density. From (70), the correlator peak output is

$$C_p = A \tag{76}$$

and the derivative of C(t) at the zero-crossing just right of the correlation peak is

$$K = C'(t)\big|_{t=T}$$

$$= A\left[-\left(1 - \frac{t}{T_m}\right)\omega_0 \sin(\omega_0 t) - \frac{1}{T_m}\cos(\omega_0 t) + \frac{\cos(\omega_0 t)}{T_m}\right]_{t=T} \quad (77)$$

$$= -A\left(1 - \frac{1}{16}\right)\omega_o$$

$$= -A\,\omega_o \frac{15}{16} \quad V/\text{sec.}$$

Linearizing about this zero-crossing, the variance of the course synchronization timing error is then

$$(\Delta t)^2 = \frac{\sigma_n^2}{K^2} \quad \text{sec.}$$

$$= \frac{N_o B_n}{K^2} = \frac{N_o}{T_m}\frac{1}{A^2\omega_o^2}\left(\frac{16}{15}\right)^2 \quad (78)$$

$$= \frac{N_o T^2}{16\ T\ A^2 4\pi^2}\left(\frac{16}{15}\right)^2$$

$$(\Delta t)^2 = \frac{2}{SNR}\left(\frac{T}{15\pi}\right)^2 \quad \text{sec.} \quad (79)$$

In other words,

$$\frac{\Delta(t)}{T} = \frac{1}{15\pi}\sqrt{\frac{2}{SNR}} \quad \textit{symbols, rms} \quad (80)$$

where

$$SNR = \frac{A^2}{2}\left(\frac{T}{N_0}\right)$$

Hence, for a demodulator output SNR of 0 dB during the bit synchronization frame, the rms timing error for the course synchronization is only on the order of 3 degrees rms which is super.

For false alarm considerations, again the peak correlator output was A with the variance given by (75). In terms of the demodulator,

$$A = K_f \quad \Delta\omega \tag{80.1}$$

We need to know what amount of noise passes through $H_p(f)$ in order to set the detection threshold. With no signal present, assume that the input noise spectrum into the modem is given as N_I. Assuming the worst case RxBPF, assume that the worst case noise bandwidth is 10 kHz.

In terms of an equivalent noise spectrum, we have

$$S_n(f) = N_I; \qquad | f - f_o | \leq 5\ kHz \tag{81}$$

one-sided. Passing $S_n(f)$ through an FM discriminator centered at f_o then effectively results in an output spectrum given by

$$S'_n(f) = [\ 2\pi (f - f_o)\ ]^2\ N_I; \quad |f - f_o| \leq 5\ kHz \tag{82}$$

or equivalently at baseband

$$S''_n(f) = (2\pi f)^2\ N_I; \qquad | f | \leq 5kHz$$

The preamble matched filter can be approximated by

$$| H_p(f) |^2 = 1; \qquad \frac{1}{4T} - \frac{1}{2T_m} \leq f \leq \frac{1}{4T} + \frac{1}{2T_m}$$

and similarly for the negative f's. As a result, the variance of the correlator output due to noise only at the modem input is

$$\sigma_0^2 = 2 \int_{1750}^{2250} (2\pi f)^2 N_I K_f^2 \, df$$

$$= \left[2N_I \, K_f^2 \, 4\pi^2 \frac{f^3}{3} \right]_{1750\ Hz}^{2250\ Hz} \qquad (83)$$

$$= \frac{8}{3}\pi^2 \, K_f^2 \, N_I \, (6.03 * 10^9)$$

In order to put this into real terms, assume that A=1 for $\Delta\omega = 2\pi$ 2 kHz in (80.1). Then

$$K_f = \frac{1}{2\pi \ 2000}$$

Also, in order to prevent the A/D converter from being saturated, and assuming Gaussian noise peaks are < 3σ,

$$3\sqrt{N_I B_I} = 1 \quad \max \qquad (84)$$

where B_I is the bandwidth of A/D input filter in Figure 2. Taking B_I = 20 kHz, we have $N_I = 5.6 \cdot 10^{-6}$. Using these results in (83),

$$\sigma_0^2 = \frac{8}{3}\pi^2 \left(\frac{1}{2\pi (2 \ 10^3)} \right)^2 (5.6 \cdot 10^{-6}) \ (6.03 \cdot 10^9) \qquad (85)$$

which finally gives σ_0 = 0.075 volts. Therefore, negligible false alarm rate detection performance should be achievable by setting the detection threshold at $5\sigma_0$ or $6\sigma_0$. This threshold is still substantially smaller than the peak C_p which is 1.0 for the parameters selected.

This last material concerning $H_p(f)$ has not been as rigorous as I would like but the assumptions are reasonable, and support the approach taken.

The initial acquisition process is then as follows:

1) Set up $H_p(f)$ as a correlative receiver with $h(t) = K_o \sin(\omega_o t)$, $t = nT/6$.

   Unlike in the text, h(t) can only extend probably 12T rather than 16T in order to guarantee that the handoff between course acquisition and track happens before the end of the bit synch frame. One 2T must be taken off each end of h(t) such that h(t) still has a D.C. average of 0.*

2) Watch the correlator output for a sample which exceeds the detection threshold Λ_0.

3) Observe (and interpolate as discussed in the upcoming transition tracking part) where the next zero-crossing out of the correlator is with respect to the 48 K sample/s sampling clock, and adjust the phase accordingly.

4) Estimate the D.C. offset as shown next.

5) Go to track

* This problem can be circumvented by placing a M4T delay in series with the symbol matched filter in Figure 15 where M=4 ideally. The added delay would allow the acquisition algorithm to search for the global correlation peak rather than trigger on the first observed sidelobe above threshold Λ_0. Performance would be improved significantly.

D.C. Component Estimation

The D.C. estimate block shown in Figure 15 estimates the demodulator D.C. output resulting from the offset frequency errors by simply performing a block average over input samples. The average must be performed over an integral number of 4T seconds in order for the superimposed modulation term to average to zero.

The block average cannot be performed over the width of h(t) unless we can guarantee that the observed correlation peak is the main lobe rather than a sidelobe. In the sidelobe case, averaging over a large block would invariably result in some inclusion of a noise-only portion thereby impairing the D.C. estimate. Adoption of the (*) note above alleviates this concern.

Assuming that this block average is performed then over T_m=16T, and a white noise spectrum when a signal is present with one-sided noise spectrum N_o, the variance of the D.C. estimate is

$$\sigma_{dc}^2 = \boldsymbol{E}\left[\left(\int_0^{T_m} n(t)\ dt \right)^2 \right] \tag{86}$$

$$\sigma_{dc}^2 = \boldsymbol{E}\left[\int_0^{T_m} dt \int_0^{T_m} dt'\ n(t)n(t') \right]$$

$$\sigma_{dc}^2 = \int_0^{T_m} dt \int_0^{T_m} dt'\ R_n(t' - t)$$

Therefore

$$\sigma_{dc}^2 = \int_{-T_m}^{T_m} T_m \left(1 - \frac{|\tau|}{T_m}\right) R_n(\tau)\ d\tau \tag{87}$$

Assuming that the demodulator output noise is essentially white (being much more wide-band compared to $1/T_m$), effectively,

$$R_n(\tau) \rightarrow \frac{N_o}{2}\left(\frac{1}{T_m}\right)^2 \delta(\tau) \tag{88}$$

in which case

$$\sigma_{dc}^2 \rightarrow T_m \left(\frac{N_o}{2}\right)\left(\frac{1}{T_m}\right)^2 = \frac{N_o}{2T_m} \tag{89}$$

which is equivalent to having a one-sided noise bandwidth of $1/2T_m$ Hertz. Then, assuming a demodulator output signal to noise ratio of

$$SNR = \frac{A^2}{2}\frac{T}{N_0}$$

$$N_o = \left(\frac{2SNR}{A^2T}\right)^{-1} \tag{90}$$

$$\sigma_{dc}^2 = \left(\frac{2SNR}{A^2T}\right)^{-1} \frac{1}{2T_m}$$

$$= \frac{A^2T}{2\ SNR}\ \frac{1}{32\ T} = \frac{A^2}{64\ SNR} \tag{91}$$

This result shows that with K_f such that $A = K_f\Delta\omega = 1$,

$$\sigma_{dc} = \frac{A}{8\sqrt{SNR}} \tag{92}$$

$$\sigma_{dc} = \frac{1}{8\sqrt{SNR}} \tag{93}$$

which is very good. This approach is in fact optimal for a D.C. component (of specified time duration) in AWGN.

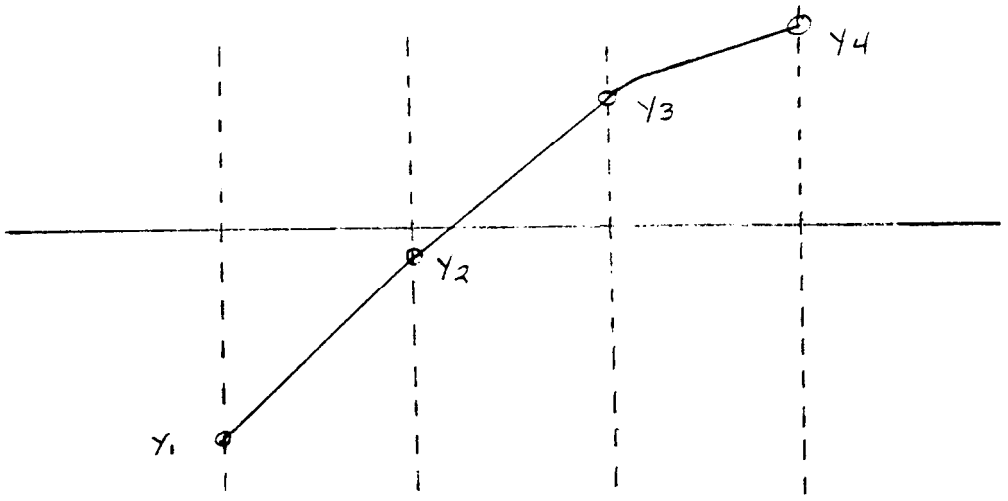
Note: Once again I avoided dealing with non-white spectra which in fact results when signal plus noise is present in the interest of time. With reasonable demodulator output SNR, the estimate for σ_{dc} should be adequate so as not to impair performance. Estimating D.C. over the known bit synch frame is also prudent since we know that the modulation term averages to zero exactly, and is not data dependent.

Symbol Matched Filter

This filter is matched to the symbol shape created at the transmitter output just prior to FM modulation of the VCO. Therefore, if the impulse response used at the premodulation filter is $h_m(t)$, the impulse response of the matched filter should be $h_s(t) = h_m(-t)$.

Transition Detector

Since the signal is very much band limited at the output of the symbol matched filter, we can very accurately estimate any zero-crossing with a simple polynomial technique as follows:



Clearly, a zero-crossing has occurred between points 2 and 3. We could use straight-line interpolation to estimate the zero-crossing instant τ_z, but with 6 samples per symbol, some finer interpolation would be better. Assuming the same kind of polynomial fit approach as before where

t = 0 @ Y_1
1 @ Y_2
2 @ Y_3
3 @ Y_4

You don't want to do divisions so this complicates doing interpolations. Even with straight-line interpolation we have

$$\tau_z \approx \left| \frac{y_z}{y_3 + y_2} \right| T_s \qquad \textit{first order} \qquad (94)$$

In lieu of division (I looked this up in the TMS320 manual), I advocate that a binary search be done between Y_2 and Y_3 in order to estimate the zero-crossing within one-fourth of your sampling clock or (T/6)/4 = T/24. This will result in a lower tracking error variance.

Loop Parameter & Loop Filter

The interrupt clock on the TMS320 will be effectively running at 20 MHz. This means that there are $20 \cdot 10^6/8 \cdot 10^3 = 2500$ clocks per symbol period. This is far more clock resolution than needed. Rather, assume that each symbol period is broken up into 50 equal portions (i.e., 50 clocks per portion).

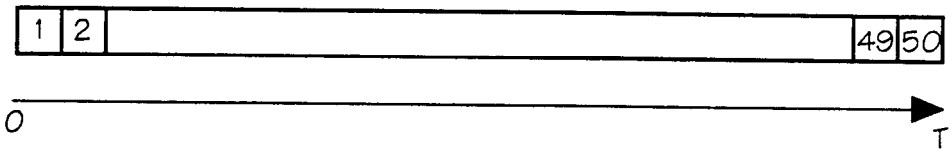


Figure 17

No stated requirements indicate that a Type II tracking loop is required so a Type I will be adopted. The true mid-bit transition (if present) occurs at an edge of one of the cells in Figure 17.

Ideally, we would like to situate the mid-bit transition at the boundary between cell 25 and 26. Therefore, if the observed (interpolated) transition occurs later than cell 26 we want to advance the clock 1/50th else if it is earlier than cell 24 retard the clock 1/50th symbol. A retard by 1/50th would be done by setting the timer count to initially (2499-50), and an advance by 1/50th, using (2499+50).

For 50% transition density, this scheme can tolerate an error in data rate up to at most

$$\frac{1}{2\pi}\frac{\Delta\phi}{T}\eta = \left(\frac{1}{50}\right) 8000\ (0.5) = \pm\ 80\ bps,\ \ which\ is\ \pm\ 1\ \% \qquad (95)$$

If larger data rate anomalies must be accommodated, matters become, shall we say, more interesting.

Assuming that random data is received in noise, at the matched filter output, the probability that a mid-bit transition occurs in cell n may be denoted by t_n. Under noise-free ideal conditions then,

$$t_{25} = t_{26} = \frac{1}{2}$$

$$t_i = 0 \qquad for\ i \neq 25,26$$

Assuming that these detailed transition probabilities have been obtained using exhaustive simulation, the probability of retarding the clock given that we believe the true transition is in cell n is

$$\begin{aligned} &\boldsymbol{P}[\ Retard\ Clock \mid Cell\ n\ ] = \\ &\qquad \boldsymbol{P}[\ Next\ Zero\ Crossing\ Appears \in Cell > n\ ] \\ &\qquad = \sum_{i=n+1}^{50} t_i \end{aligned} \tag{96}$$

$$\begin{aligned} &\boldsymbol{P}[Advance\ Clock \mid Cell\ n\ ] = \\ &\boldsymbol{P}[Next\ Zero\ Crossing\ Appears \in Cell < n] \\ &= \sum_{i=1}^{n} t_i \end{aligned} \tag{97}$$

The probabilities of retarding the clock while in cell i may be denoted by p_i, and the probability of advancing the clock while in cell i given by q_i. Note that

$$p_i + q_i = 1 \tag{98}$$

which means that this model adjusts the clock every data transition. The steady-state behavior of the tracking loop may be analyzed by modeling this as a first-order discrete Markov process. This is shown in Figure 18.

The tracking variance (steady-state) may be calculated by recognizing that the steady-state probability of being in state k (i.e., believing true mid-bit transition is in cell k)

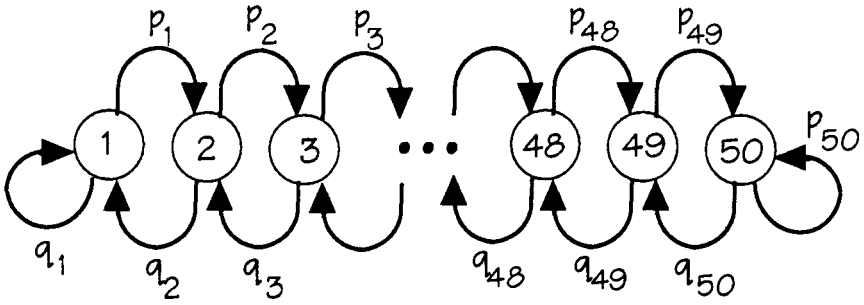


Figure 18

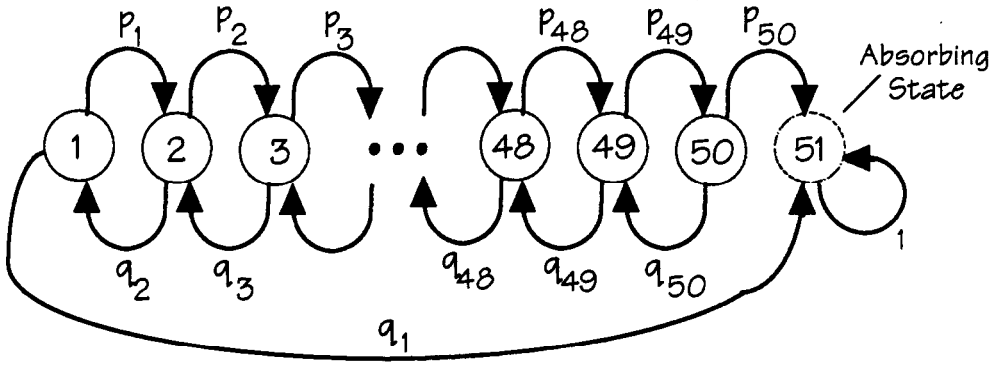


Figure 19

$$S_k = p_{k-1}S_{k-1} + q_{k+1}S_{k+1} \tag{99}$$

$$\begin{aligned} S_1 &= q_1S_1 + q_2S_2 \\ S_{50} &= p_{50}S_{50} + q_{49}S_{49} \end{aligned} \tag{100}$$

where S_k is the steady-state probability of being in state k. The first recursive equation is satisfied for

$$S_k = \frac{p_{k-1}}{q_k} S_{k-1} \tag{101}$$

Since we must be "in" one of the 50 states,

$$\sum_{k=1}^{50} S_k = 1 \tag{102}$$

A little algebra reveals that

$$S_1 = \frac{1}{1 + \sum_{k=1}^{50}\left[\prod_{i=1}^{k-1}\left(\frac{p_i}{q_{i+1}}\right)\right]} \tag{103}$$

$$S_k = S_1 \prod_{i=1}^{k-1}\left(\frac{p_i}{q_{i+1}}\right) \tag{104}$$

The mean tracking cell number is

$$\mu = \sum_{i=1}^{50} i\, S_i \tag{105}$$

and the tracking error variance in steady-state is given by

$$\sigma^2 = \sum_{i=1}^{50} (i - \mu)^2 S_i \tag{106}$$

The mean-time to cycle slip can be found in a similar manner. First of all, the Markov chain must be appropriately modified to that shown in Figure 19. Let T_k^{N+1} be the mean-time to reach state N+1 given that we begin in state k. Then

$$T_k^{N+1} = p_k T_{k+1}^{N+1} + q_k T_{k-1}^{N+1} + 1 \tag{107}$$

For the end states,

$$T_1^{N+1} = p_1\left(T_2^{N+1} + 1\right) + q_1 \tag{108}$$

$$T_1^{N+1} = p_1 T_2^{N+1} + 1$$

$$T_{N+1}^{N+1} = 0$$

$$N + 1 = 50$$

The recursive form may be rewritten as

$$T_{k+1}^{N+1} = \frac{T_k^{N+1}}{p_k} - \frac{q_k}{p_k} T_{k-1}^{N+1} - \frac{1}{p_k} \tag{109}$$

Dropping the N+1 superscript and subtracting T_k from each side,

$$T_{k+1} - T_k = T_k\left(\frac{1}{p_k} - 1\right) - \frac{q_k}{p_k}T_{k-1} - \frac{1}{p_k} \tag{110}$$

$$\therefore \quad T_{k+1} - T_k = \frac{q_k}{p_k}[T_k - T_{k-1}] - \frac{1}{p_k}$$

This recursion may be applied repeatedly starting with state 1, and after some algebra,

$$T_{k+1} - T_k = \left[\prod_{i=1}^{k}\left(\frac{q_i}{p_i}\right)\right]T_1 - \sum_{i=1}^{k}\left[\left(\frac{1}{q_i}\right)\prod_{j=1}^{k}\left(\frac{q_j}{p_j}\right)\right] \tag{111}$$

The boundary condition on T_{N+1} can be used to solve for T_1 in closed form as

$$\sum_{k=1}^{N}(T_k - T_{k+1}) = T_1 \tag{112}$$

Therefore,

$$T_1 = \frac{\sum_{k=1}^{N}\left\{\sum_{i=1}^{k}\left(\frac{1}{q_i}\prod_{j=i}^{k}\left(\frac{q_j}{p_j}\right)\right)\right\}}{1 + \sum_{k=1}^{N}\left[\prod_{i=1}^{k}\left(\frac{q_i}{p_i}\right)\right]} \tag{113}$$

Using the same approach once more for arbitrary T_k,

$$\sum_{k=1}^{\alpha}(T_k - T_{k+1}) = T_1 - T_\alpha \tag{114}$$

$$\therefore \quad T_\alpha = T_1\left\{1 + \sum_{k=1}^{\alpha}\left[\prod_{i=1}^{k}\left(\frac{q_i}{p_i}\right)\right]\right\} - \sum_{k=1}^{\alpha}\left[\sum_{i=1}^{k}\left(\frac{1}{q_i}\prod_{j=i}^{k}\left(\frac{q_j}{p_j}\right)\right)\right] \tag{115}$$

Given the transition probabilities, these preceding equations may be used to fully characterize the Markov-based bit synchronizer loop. A computer listing which performs these calculations is provided in Appendix II.

An example set of calculations utilizing these mathematical results is outlined briefly in Appendix I. Even with simulation runs of several thousand symbols for each E_b/N_o, the t_i data was fairly jagged as shown in the accompanying plots. For moderate to high SNR, the t_i data were first fitted/smoothed using a Gram-Charlier series. For very low SNR, polynomial regression with the side condition that the probability mass sum to 1 was used instead. The underlying calculations proceed as outlined in Figures A1.1 through A1.6.

Unfortunately, the detailed loop analysis for our situation cannot proceed without first running long detailed simulations. In lieu of detailed simulation results, the following approach may be adopted. The tracking loop open-loop gain function is effectively given by

$$G_{OL}(s) = \frac{K_d(SNR)\ K_v}{s} \tag{116}$$

where K_d is the phase detector gain (which is a heavy function of SNR) and in the present context

$$K_v = 2\pi\left(\frac{1}{50}\right)\left(\frac{1}{T}\right) \tag{117}$$
$$= 2\pi\ 160\ rad/sec/volt$$

Here, K_d is the slope of the phase detector curve at the tracking point. This is found by differentiating the phase detector S-curve which is obtainable from the retard/advance probabilities. Unfortunately, this information is not available. Earlier work which I have done which is probably reasonable (but still very rough) has

Eb/No, dB	Kd, v/rad	Resulting Loop BW, Hz
2	0.49	78.4
3	0.55	88
4	0.6	96
5	0.62	99
7	0.87	139
9	1.09	174.4
11	1.47	235.2

(Cited data used a modulation index of 0.7 rather than 0.3.)

Even at E_b/N_o = 5 dB, the loop noise bandwidth is ≈ 100 Hz which when compared to the 8 Kbps data rate, represents a loop SNR improvement of roughly 10 log (4000/100) ~ 16 dB. Therefore, tracking loop performance should be excellent.

For large loop SNR, the steady-state tracking performance is described by the Tikhonov probability distribution function [5] which is given by

$$p(\phi) = \frac{e^{\rho\cos\phi}}{2\pi I_0(\rho)} \qquad |\phi| \le \pi \tag{118}$$

where ρ is the loop SNR. In terms of SNR, the variance of the tracking error is given by

$$\sigma_\theta^2 = \frac{1}{2\rho} \tag{119}$$

The Tikhonov pdf (118) may be closely approximated by a Gaussian pdf for large ρ.

The average time for the tracking loop to slip is given by

$$T_{av} = \frac{2\pi\rho I_0^2(\rho)}{2B_L} \tag{120}$$

where B_L is the loop bandwidth given by

$$B_L = \frac{K_d K_v}{2\pi} \tag{121}$$

Depending upon your exposure to things like Markov processes, and Chapman-Kolmogorov equations, some of this may be somewhat esoteric. Rather than take the time to develop all the details, an example of how this applies in the case of a similar bit synch I designed is given in Appendix I. Given the detailed simulation data, the program in Appendix I could easily be modified to handle this design.

Given the steady-state tracking error pdf (118), the raw bit error rate for the complete bit synchronizer is found by

$$P_b = \int_{-\pi}^{\pi} p(e \mid \phi)\ p(\phi)\ d\phi \tag{122}$$

where $p(e|\phi)$ is the probability of a bit error given a symbol tracking error of ϕ. This conditional probability must also be evaluated via simulation at moderate to low SNRs.

I know that the tracking loop information has been less than quantitative, needing detailed simulation data before proceeding. Nonetheless, from (118), the loop SNR ρ, is going to be adequate so as to eliminate any real degradation due to tracking errors. This would not have been the case if we relied on unaided loop acquisition as described at the bottom of page 5. In that respect, this approach is much better.

As you might imagine, almost countless references are available on the subject of tracking loops. One of the nice aspects of a discrete case such as this is that first-order Markov process theory generally suffices [24] whereas with continuous systems, one is forced to deal with matters like Ito and Folker-Planck stochastic differential equations [38]. The subject of digital tracking loops is well-addressed in [61-64]. Although I have probably more than 60 papers on the subject at hand, these are some of the most readable and informative.

Data Sequence Estimation

Intersymbol interference effects are so substantial in regard to BER performance that measures must be taken to mitigate their effect on system performance. Many related

approaches deal with this subject from seismology to deterministic encryption methods. In the present context, Chapter 6 of [67] gives perhaps the best overview of available techniques. Some of these methods are

- Linear Equalization
  - Peak Distortion Criterion
  - Zero-Forcing Algorithm
  - Mean-Square Error
  - Least-Mean Square
- Decision Feedback Equalization
- Adaptive Equalization
- Maximum-Likelihood

Many varieties of algorithms under the group "linear equalization" have been studied.

Had we gone forward with MMSE synchronizer type, we would have been forced to address the equalization issue. In general, equalization approaches lead to substantially higher tracking error variance, particularly when the intersymbol interference is severe.

The maximum-likelihood approach, often called MLSE (maximum likelihood sequence estimation) is the optimal approach [67]. The intersymbol interference introduced by the transmitter Gaussian filter can be viewed as a convolutional code (analog) and can be dealt with in that manner. As such, the system performance is tightly coupled to the channel cutoff rate R_0 which has units of bits per dimension.

In my view, complete inclusion of a MLSE routine within the modem may be a question of processor throughput. In order to give you a feeling for what would be involved, an applicable example is sketched out below.

The symbol matched filter, being fairly narrow band, causes noise samples separated by one symbol period T to be correlated. This makes for very difficult (certainly impractical) algorithms so the matched filter must be followed by a whitening filter. Mathematically, at the output of the matched filter, we would have

$$y(n) = \sum_i I_{n-i}\beta_i D^i + \sum_i \nu_{n-i}\gamma^i D^i \qquad (123)$$

where

I_n	received symbol sequence
β_i	coefficients due to intersymbol interference
γ_i	"
D	Delay operator.

For no ISI and no correlation between noise samples

$$y(n) = I_n + \nu_n \tag{124}$$

The noise whitening filter transforms y(n) to

$$y'(n) = \sum_i I_{n-i}\alpha_i D_i + \nu'_n \tag{125}$$

where the ν_n' are now mutually uncorrelated. This very important step allows the joint probability functions which arise for the noise ν_n to be replaced with very simple products of marginal probability functions [67, p. 611]. From this point on, the Viterbi algorithm may be used to optimally estimate the data sequence.

In order to be reasonably optimal with the data eye patterns which apply to the present system, the Viterbi receiver would probably have to use a minimum of 8 states. A representative trellis diagram is shown below for this case.

Sample from whitening filter

	X_1	X_2			
S_0	o	o	o	o	o
S_1	o	o	o	o	o
S_2	o	o	o	o	o
S_3	o	o	o	o	o
S_4	o	o	o	o	o
S_5	o	o	o	o	o
S_6	o	o	o	o	o
S_7	o	o	o	o	o

The states can be equivalenced to voltage, e.g., S_0 corresponding to the top of the eye, S_7 the bottom. Given that the last observed sample was S_n, the probability that the true next state is S_m given that we observe the next sample X_1 is given by P_{nm} = prob ($\rightarrow$m 1 S_n, X_1).

For the scheme shown, there are 8 states, each with 8 possible next states, and given 8 possible values for X_i, 8^3 = 256 possible probabilities (which are in fact also a function of SNR). After each new X_i, all 64 of the state transition probabilities must be evaluated. Of the 64 possible paths to the next states, only 8 paths "survive", those having the highest probability or metric and the process repeated.

Let it suffice to say that although optimal, it would take considerable effort (and computation) to implement such a decoder. You may eventually wish to do this, but in the near term, there is a much simpler way.

Before leaving the MLSE topic, some additional comments may be helpful. The use of the Viterbi algorithm with systems having ISI was first discussed in [68] and is quite readable. A number of other references are worthy of note [67-76]. Generally, the surviving paths in the trellis must be calculated to a depth of 5L symbols before the first data symbol is known. Here, L is the code constraint length which for all practical purposes is 3 or 4 for our case. If the calculations are truncated earlier, some of the performance impact is discussed in [77]. One of the most excellent papers discussing performance of coding on different channels in my estimation is [78]. The underlying fundamentals can be somewhat esoteric, but are well covered in [67] and [79-81].

The principle bottleneck in implementing any Viterbi algorithm is the add/compare/select operation which determines the path survivors at each state. This issue is addressed in [82,83].

Recommended Data Estimation Approach

The approach I recommend you pursue, at least until you are certain about throughput margins, etc., is based upon [84]. Another factor is that thus far, no system noise figure has been given so discussion of E_b/N_o is meaningless. Until this information is made available and the performance measure identified, I see no point in over design.

Reference [84] is quite clear on the approach. The premodulation filter and modulation index area identical for our application. Aside from our rather large frequency uncertainty of ±1.5 kHz, we could readily adopt the

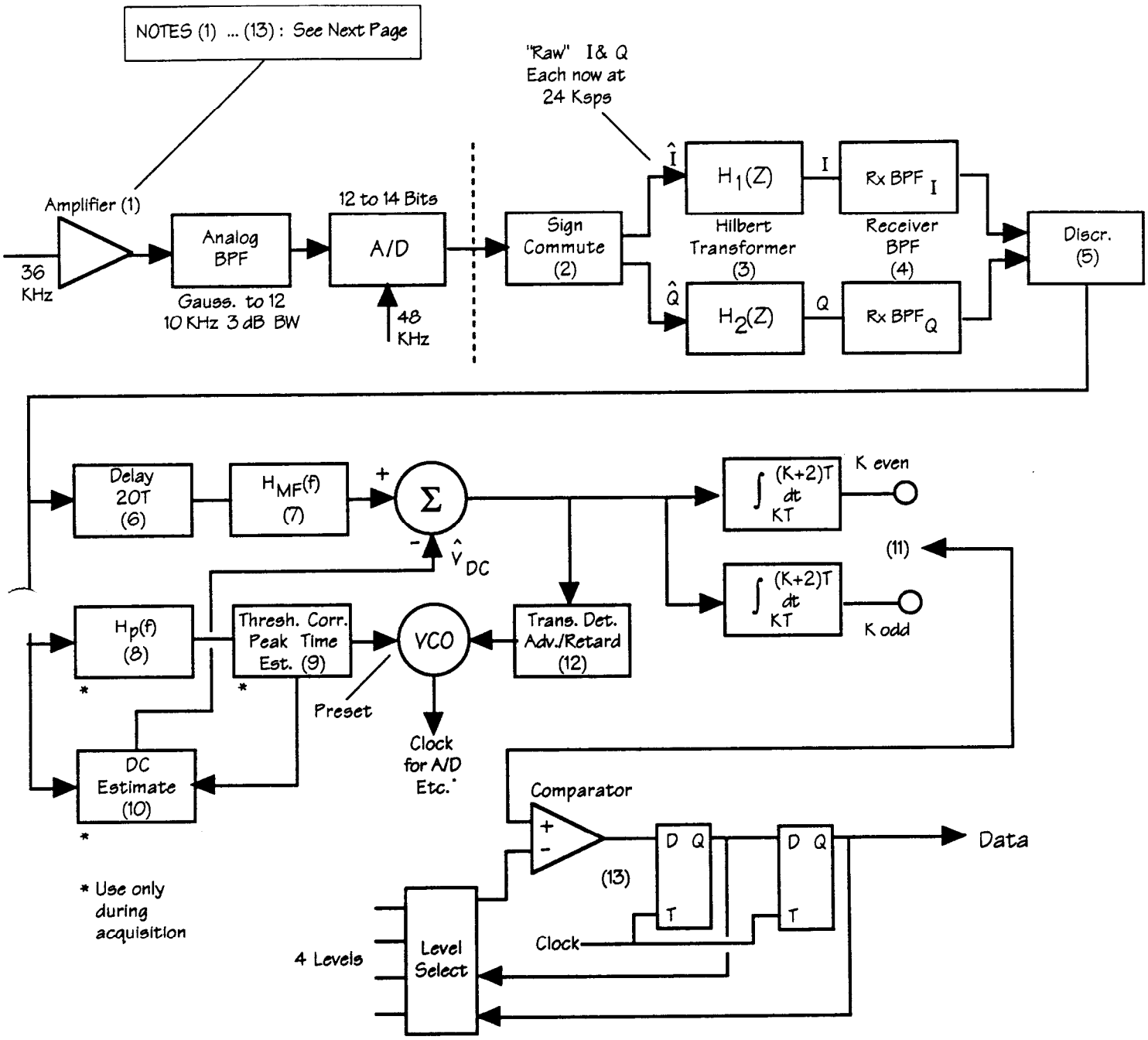


Figure 20 - Design Summary

recommended IF filter, Bessel with BT = 0.6. Including the ±1.5 kHz uncertainty, we are forced to use BT = 0.975.

This brings up a point worth mentioning. Since you are given 10.7 MHz as an input, and you are mixing down to an IF of 36 kHz, a long phase-lock receiver approach could be considered to accommodate the (rather large) frequency uncertainty and still keep BT = 0.6. The burst communication mode could on the other hand make this somewhat difficult to actually implement.

Section [B] of [84] (See Appendix II) describes the approach in good detail and I will not attempt to improve upon it here. The obtainable performance should match up very closely with Figure 7 of [84] (note that they used BT = 1.0 like our needs). The implementation is shown in Figure 4 of [84]. The so-called "detected signal" is the output from one of two integrate-and-dump filters. The integration period for both integrate-and-dump filters is 2T, but the start times (and dump times) are staggered by 1T. This is shown in greater detail in the summary design, Figure 20.

The four threshold levels discussed in [84] and shown in Figure 20 must be obtained yet either experimentally or analytically.

Further comments, Figure 20

- VCO implemented with TMS320 timer.
  Clock advance 1/50, set initial count to 2499-50
  Clock retard 1/50, set initial count to 2499+50

- Transition Phase Detector
  Unlike p. 52, no interpolation required.
  Must only determine:
  1) determine if transition occurred
  2) positive or negative transition direction
  3) decide to advance or retard clock.

- After initial frame synch, automatically go back to acquisition after N bit periods have passed.

1. Amplifier/AGC maintains average signal level into A/D, including 3σ noise loading effects.

2. Sign Commutation per Figure 12.

3. Equation 39 and 40.

4. Rx BPF implememted at baseband in terms of lowpass filters in I and Q separately. Each filter is a lowpass Bessel, BT = 1.0.

5. Equation 54.

6. Delay discussed bottom 47.

7. Matched to transmit symbol shape.

8. Matched to sine wave over period 16T.

9. After initial threshold crossing find max corr. peak. Then compute trailing zero crossing.

10. Page 47

11. Integrate and dumps over 2T each staggered by T [84].

12. Advance/Retard clock (loop filter).

13. Multi level ISI method.

References

1) RAM Specification, Excerpt
MTS 17.2 through MTS 18.2

2) Orfanidis, S.J., Optimum Signal Processing, 2nd Ed., Macmillan, 1985

3) Rife, D.C., R.R. Boorstyn, "Single-Tone Parameter Estimation from Discrete-Time Observations," IEEE IT-20, Sept. 1974, pp. 591-598

4) Abatzoglov, T.J., "A Fast Maximum Likelihood Algorithm for Frequency Estimation of a Sinusoid Based on Newton's Method," IEEE ASSP-33, Feb. 1985, pp. 77-89.

5) Ziemer, R.E., R.L. Peterson, Digital Communications and Spread Spectrum Systems, Macmillan, 1985

6) Spilker, J.J., Digital Communications by Satellite, Prentice-Hall, 1977, p. 448

7) Synetcom GMSK Modem Proposal

8) Feher, K., Advanced Digital Communications, Prentice-Hall, 1987

9) Hirade, K., et al., "Error-Rate Performance of Digital FM with Differential Detection in Land Mobile Radio Channels," IEEE VT-28, Aug. 1979, pp. 204-212

10) Klapper, J., "A New Family of Low-Delay FM Detectors," Proc. NTC, 1975, pp. 34-6 through 34-10, Patent Awarded Dec. 10, 1974, No. 3,854,099

11) Ray, S.R., "Zero-Crossing-Based Approximate Demodulation of Wide-Deviation FM," IEE Proc., Part F, Feb. 1984, pp. 47-51

12) Suzuki, H., "Optimum Gaussian Filter for Differential Detection of MSK," IEEE COM-29, June 1981, pp. 916-918

13) Scott, R.E., C.A. Halijak, "The SCEM-Phase-Lock Loop and Ideal FM Discrimination," IEEE COM-, March 1977, pp. 390-392

14) Aguirre, S., S. Hinedi, "Two Novel Automatic Frequency Tracking Loops," IEEE AES-25, Sept. 1989, pp. 749-760

15) Turner, S.E., "Multipath Interference in FM Data

Transmission," RF Design, Dec. 1988, pp. 26-47

16) Iwanami, Y., T. Ikeda, "An Adaptive Control Method of a Digital Signal Processing DPLL FM Demodulator Under a Fading Environment," Globecom 1987, pp. 34.2.1-34.2.6

17) Varanasi, S., S.C. Gupta, "Statistical Analysis of Digital Phase-Locked Loops in Fading Channels," IEEE AES-20, Nov. 1984, pp. 682-692

18) Davis, B., "FM Noise with Fading Channels and Diversity," IEEE COM-19, Dec. 1971, pp. 1189-1200

19) Dharamsi, M.T., S.C. Gupta, "Performance of Quasi=Optimum Digital FM Demodulators for Fading Channels," Comp. & Elect. Engng, Pergamon Press, 1975, pp. 175-193

20) , "Performance Analysis of Digital FM Demodulators for Fading Channels in the Threshold Region," Comput & Elect. Engrg., Vol. 3, Pergamon Press, 1976, pp. 203-208

21) Shein, N.P., "An Approximate Formula for MFSK Symbol Error Probability in Fast Rayleigh Fading," Milcom 1988, pp. 5.5.1-5.5.7

22) Varanasi, S.V., "Statistical Analysis of Zero-Crossing Sampling Digital Fading Phase-Locked Loops," Ph.D. Dissertation, SMU, 1980

23) Lac, Q.T., "Mathematical Analysis of Digital Phase Locked Loops Over Fading Communication Channels," Ph.D. Dissertation, Northwestern, 1983

24) Holmes, J.K., Coherent Spread Spectrum Systems, John Wiley, 1982

25) McBride, A.L., A.P. Sage, "Optimum Estimation of Bit Synchronization," IEEE AES-5, May 1969, pp. 21-32

26) Lindsey, W.C., Synchronization Systems in Communication and Control, Prentice-Hall, 1972

27) Gardner, F.M., Phase-lock Techniques, John Wiley, 1979

28) Franks, L.E., "Carrier and Bit Synchronization in Data Communication - A Tutorial Review," IEEE COM-28, Aug. 1980, pp. 1107-1121

29) Moeneclaey, M., "The Optimum Closed-Loop Transfer

Function of a Phase-Locked Loop Used for Synchronization Purposes," IEEE COM-31, April 1983, pp. 549-553

30) , "Synchronization Problems in PAM Systems," IEEE COM-28, Aug. 1980, pp. 1130-1136

31) , "A Simple Lower Bound on the Linearized Performance of Practical Symbol Synchronizers," IEEE COM-31, Sept. 1983, pp. 1029-1032

32) , "A Class of Phase Detector Characteristics for Symbol Synchronizers Yielding Unbiased Estimates," IEEE COM-31, Sept. 1983, pp. 1033-1036

33) , "A Comparison of Two Types of Symbol Synchronizers for Which Self-Noise is Absent," IEEE COM-31, March 1983, pp. 329-334

34) , "The Influence of Four Types of Symbol Synchronizers on the Error Probability of a PAM Receiver," IEEE COM-32, Nov. 1984, pp. 1186-1190

35) , "Two Maximum-Likelihood Symbol Synchronizers with Superior Tracking Performance," IEEE COM-32, Nov. 1984, pp. 1178-1185

36) Waggener, W.N., "A MAP Symbol Synchronizer Implemented with Charge-Coupled Devices," IEEE COM-28, Aug. 1980, pp. 1184-1189

37) Pebbles, P.Z., Communication System Principles, Addison-Wesley Publishing, 1976, p. 265

38) Meyr, H., G. Aschied, Synchronization in Digital Communications, John Wiley & Sons, 1990

39) Hurd, W.J., T.O. Anderson, "Digital Transition Tracking Symbol Synchronizer for Low SNR Coded Systems," IEEE COM-18, April 1970, pp. 141-147

40) Forney, G.D., "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," IEEE IT-18, May 1972, pp. 363-377

41) Fritchman, B.D., et al., "Optimum Sequential Detector Performance on Intersymbol Interference Channels," IEEE COM-22, June 1974, pp. 788-797

42) Takasaki, Y., "Optimizing Pulse Shaping for Baseband

Digital Transmission with Self-Bit Synchronization," IEEE COM-28, Aug. 1980, pp. 1164-1171

43) Mengali, V., "A Self Bit Synchronizer Matched to the Signal Shape," IEEE AES-7, July 1971, pp. 686-693

44) Tjhung, T.T., et al., "Error Performance Analysis for Narrow-Band Duobinary FM with Discriminator Detection and Soft Decision Decoding," IEEE COM-37, Nov. 1989, pp. 1222-1228

45) Harrold, W., "A New Approximation to the Symbol Error Probability for Coded Modulation Schemes with Maximum Likelihood Sequence Detection," IEEE COM-37, April 1989, pp. 340-352

46) Anderson, J.B., "Error Bounds for Phase Modulation with a Severe Bandwidth Constraint," IEEE 1979

47) Fultz, G.L., P.L. McAdam, "Combined (Manchester) Bit Synchronization and Convolutional Decoding," Eascon, 1975, pp. 95A to 95-K

48) Lindsey, W.C., R.C. Tausworthe, "Digital Data-Transition Tracking Loops," JPL Programs Summary, 37-50, Vol. III

49) Simon, M.K., "The Steady-State Performance of a Data-Transition Type of First-Order Digital Phase-Locked Loop," JPL Programs Summary, 37-66, Vol. III

50) Chadwick, H.D., "Estimating the Phase of a Sampled Signal with Minimum Mean-Square Error," JPL Space Programs Summary, 37-62, Vol. III

Part II References

51) Juntti, J., et al., "Digital FSK Receivers Implemented by the TMS3210 Signal Processor," IEEE 88CH 2607-0, 1988

52) Maisel, J., "Hilbert Transform Works with Fourier Transforms to Dramatically Lower Sampling Rates," Personal Engineering and Instrumentation News, Feb. 1990, pp. 83-86

53) Rabiner, L.R., B. Gold, Theory and Application of Digital Signal Processing, Prentice-Hall, 1975

54) Proakis, J.G., D.G. Manolakis, Introduction to Digital

Signal Processing, Macmillan, 1988

55) "A Simple Method for Sampling In-Phase and Quadrature Components," IEEE AES-20, Nov. 1984, pp. 821-824

56) Park, J.H., "An FM Detector for Low S/N," IEEE COM-18, April 1970, pp. 110-118

57) Schilling, D.L., R. Bozovic, "Performance of Trellis Coded Continuous Phase M-ary FSK with Incoherent Detection," IEEE CH2537-9/88, 1988

58) Hess, D.T., "Equivalence of FM Threshold Extension Receivers," IEEE Com-, Oct. 1968, pp. 746-748

59) Hirono, M., et al., "Multi-level Decision Method for Bandlimited Digital FM with Limiter-Discriminator Detection," IEEE VT-33, Aug. 1984, pp. 114-122

60) Moeneclaey, M., "A Class of Phase Detector Characteristics for Symbol Synchronizers Yielding Unbiased Estimates," IEEE COM-31, Sept. 1983, pp. 1033-1036

61) Cessna, J.R., D.M. Levy, "Phase Noise and Transient Times for a Binary Quantized Digital Phase-Locked Loop in White Gaussian Noise," IEEE COM-20, April 1972, pp. 94-104

62) Yamamoto, H., S. Mori, "Performance of a Binary Quantized All Digital Phase-Locked Loop with a Ne Class of Sequential Filter," IEEE COM-26, Jan. 1978, pp. 35-45

63) Holmes, J.K., "Performance of a First-Order Transition Sampling Digital Phase-Locked Loop Using Random-Walk Models," IEEE COM-20, April 1972, pp. 119-131

64) Garodnick, J., et al., "Response of an All Digital Phase-Locked Loop," IEEE COM-22, June 1974, pp. 751-763

65) Kumar, R., W.J. Hurd, "A Class of Optimum Digital Phase Locked Loops," Proc. 25th Conf. Decision and Control, 1986, pp. 1632-1634

66) Gupta, S.C., "On Optimum Digital Phase-Locked Lops," IEEE COM, April 1968, pp. 340-344

67) Proakis, J.G., Digital Communication, 2nd Ed., McGraw-Hill, 1989

68) Forney, G.D., "Maximum Likelihood Sequence Estimation of

Digital Sequences in the Presence of Intersymbol Interference," IEEE IT-18, May 1972, pp. 363-377

69) Fritchman, B.D., "Optimum Sequential Detector Performance on Intersymbol Interference Channels," IEEE COM-22, June 1974, pp. 799-797

70) Tjhung, T.T., et al., "Effects of Pulse Shaping and Soft Decisions on the Performance of Digital FM with Discriminator Detection," IEEE COM-34, Nov. 1986, pp. 1116-1122

71) Viterbi, A.J., "Error Bounds for Convolutional Codes and on Asymptotically Optimum Decoding Algorithm," IEEE IT-13, April 1967, pp. 260-269

72) Simon, M.K., "The Impact of Mismatch on the Performance of Coded Narrowband FM with Limiter-Discriminator Detection," IEEE COM-31, Jan. 1983, pp. 28-36

73) Anderson, J.B., "Error Bounds for Phase Modulation with a Severe Bandwidth Constraint," IEEE 1979 CH1514-9/79/0000-0243

74) Tjhung, T.T., et al., "Error Performance Analysis for Narrow-Band Duobinary FM with Discriminator Detection and Soft Decision Decoding," IEEE COM-37, Nov. 1989, pp. 1222-1228

75) Harrold, W., "A New Approximation to the Symbol Error Probability for Coded Modulation Schemes with Maximum Likelihood Sequence Detection," IEEE COM-37, April 1989, pp. 340-352

76) Glave, F.E., "An Upper Bound on the Probability of Error Due to Intersymbol Interference for Correlated Digital Signals," IEEE IT-18, May 1972, pp. 356-362

77) Hemmati, F., et al., "Truncation Error Probability in Viterbi Decoding," IEEE COM, May 1977, pp. 530-532

78) Omura, J.K., B.K. Levitt, "Coded Error Probability Evaluation for Anti Jam Communication Systems," IEEE COM-30, May 1982, pp. 896-903

79) Wozencraft, Jacobs, Principles of Communication Engineering

80) Viterbi, Omura, Principles of Digital Communication and Coding

81) Lin, Costello, Error Correction Coding

82) Fettweis, G., H. Meyr, "Parallel Viterbi Decoding by Breaking the Compare-Select Feedback Bottleneck," 1988 IEEE Intl. Conf. Comm., p. 23.5.1-23.5.5

83) , "Parallel Viterbi Algorithm Implementation: Breaking the ACS-Bottleneck," IEEE COM-37, Aug. 1989, pp. 785-7909

84) Hirono, M., T. Miki, K. Murota, "Multilevel Decision Method for Bank-Limited Digital FM with Limiter-Discriminator Detection," IEEE VT-33, Aug. 1984, pp. 114-122

Appendix I

The material here addresses a Type I DPLL used for transition tracking. The transmitter used binary FSK with a modulation index of 0.7, and there was substantial (~6dB) of eye closure due to a premodulation lowpass filter (N=3 Bessel, BT=0.625) used for spectral containment. At the receiver, a N=8 Gaussian-to-12dB filter was used with BT=1.25 (B is RF 3 dB BW).

The symbol period here was broken into 64 cells rather than 50. The zero-crossing statistics out of the matched filter were obtained by using a large workstation class simulation package called BOSS. For the 13 Eb/No cases which were run, approximately 20 hours of computer execution was involved. On an AT class computer, the same study would of course required many more hours. The zero-crossing information is shown at the beginning of the program in numerical arrays (Listing in Appendix II).

Several curves are shown here following as an example. The case shown is for E_b/N_o= 11 dB. The figure captions should be self-explanatory.

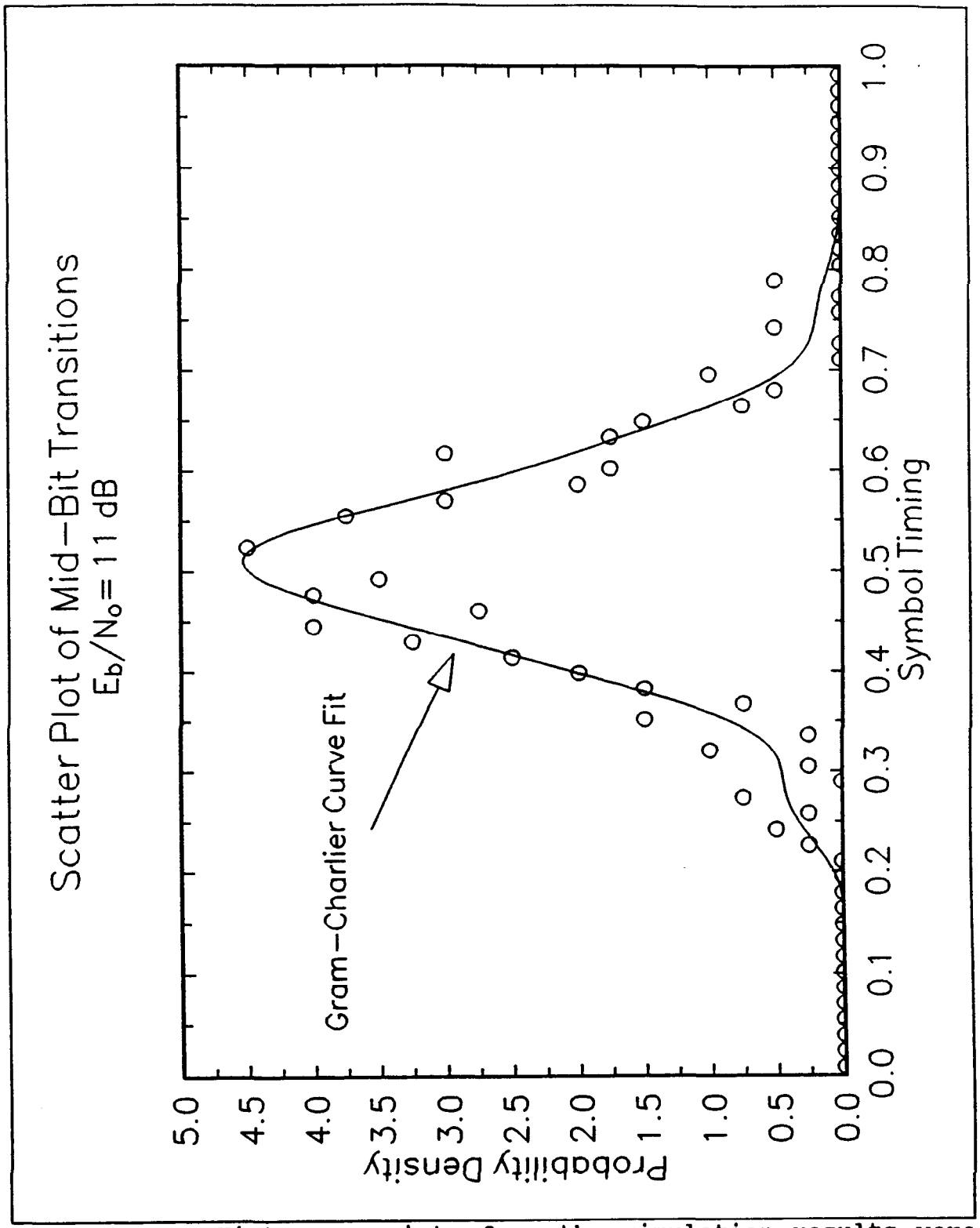


Figure A1.1 Histogram points from the simulation results were first fit with a Gram-Charlier technique which insured that (i) the data moments were preserved and (ii) the resulting fit was a valid probability density function.

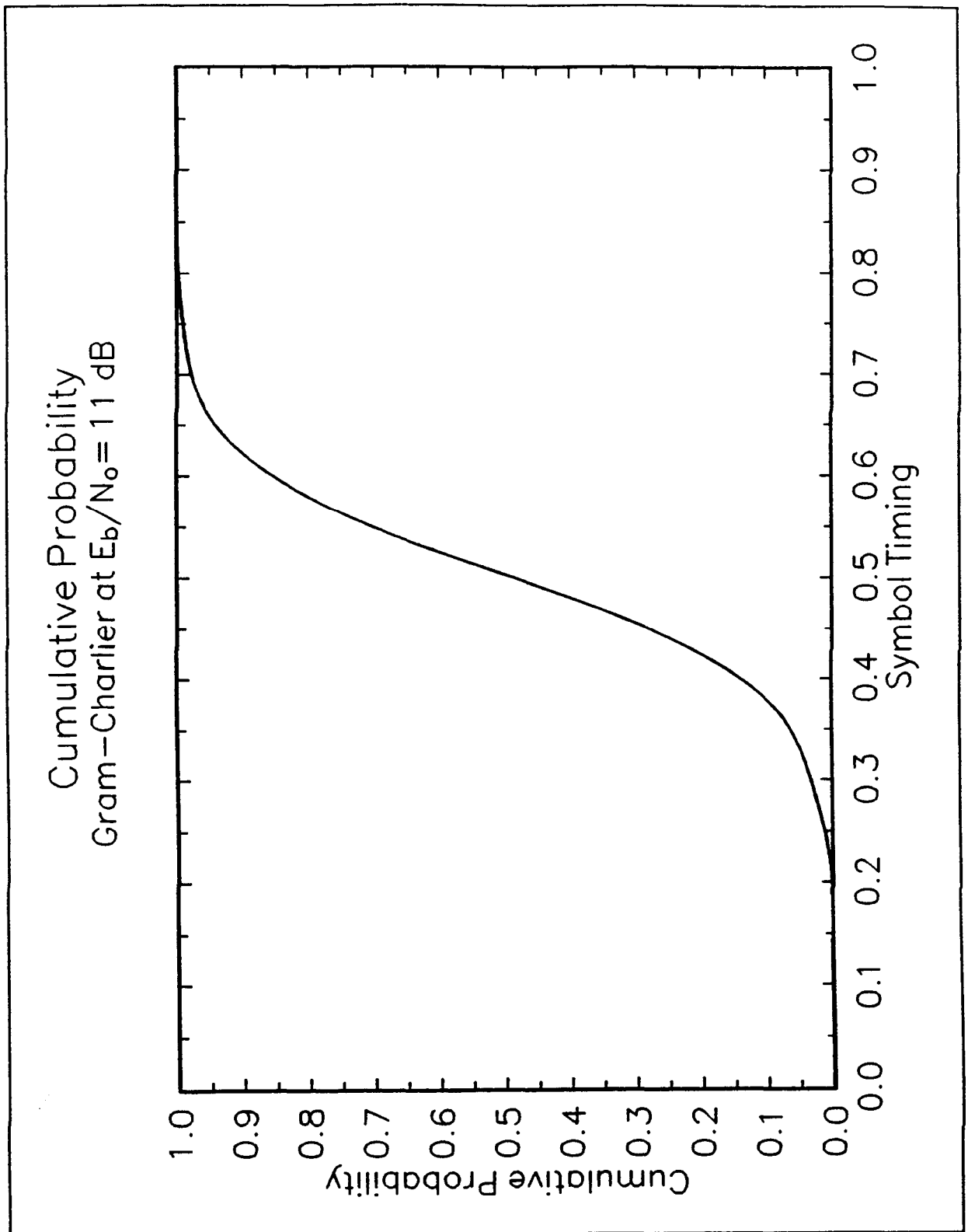


Figure A1.2 Given the valid probability density function, straight forward integration provided the cumulative probability curve.

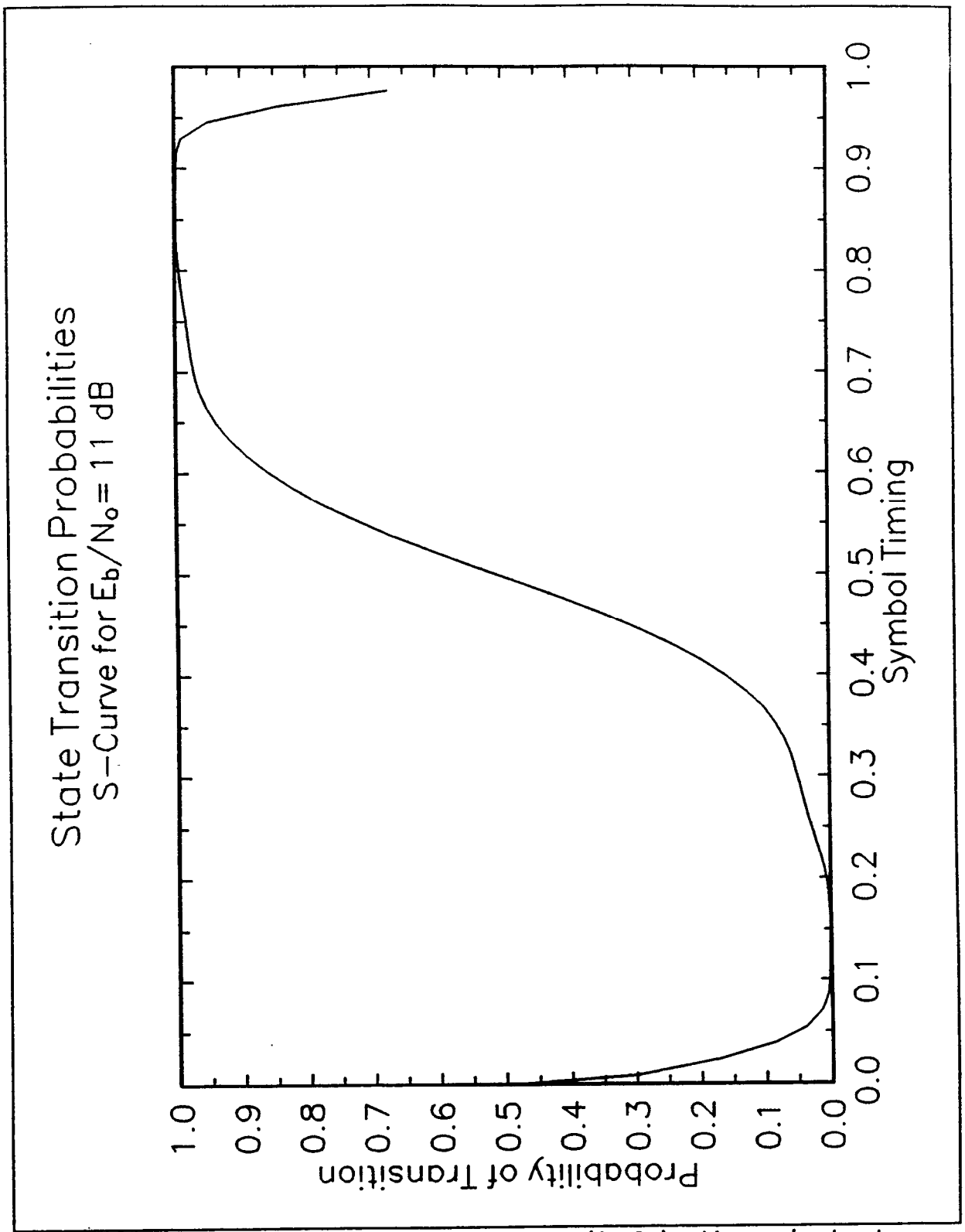


Figure A1.3 Numerical formula as described in the main text were used to extract the effective phase detector S-curve for the system.

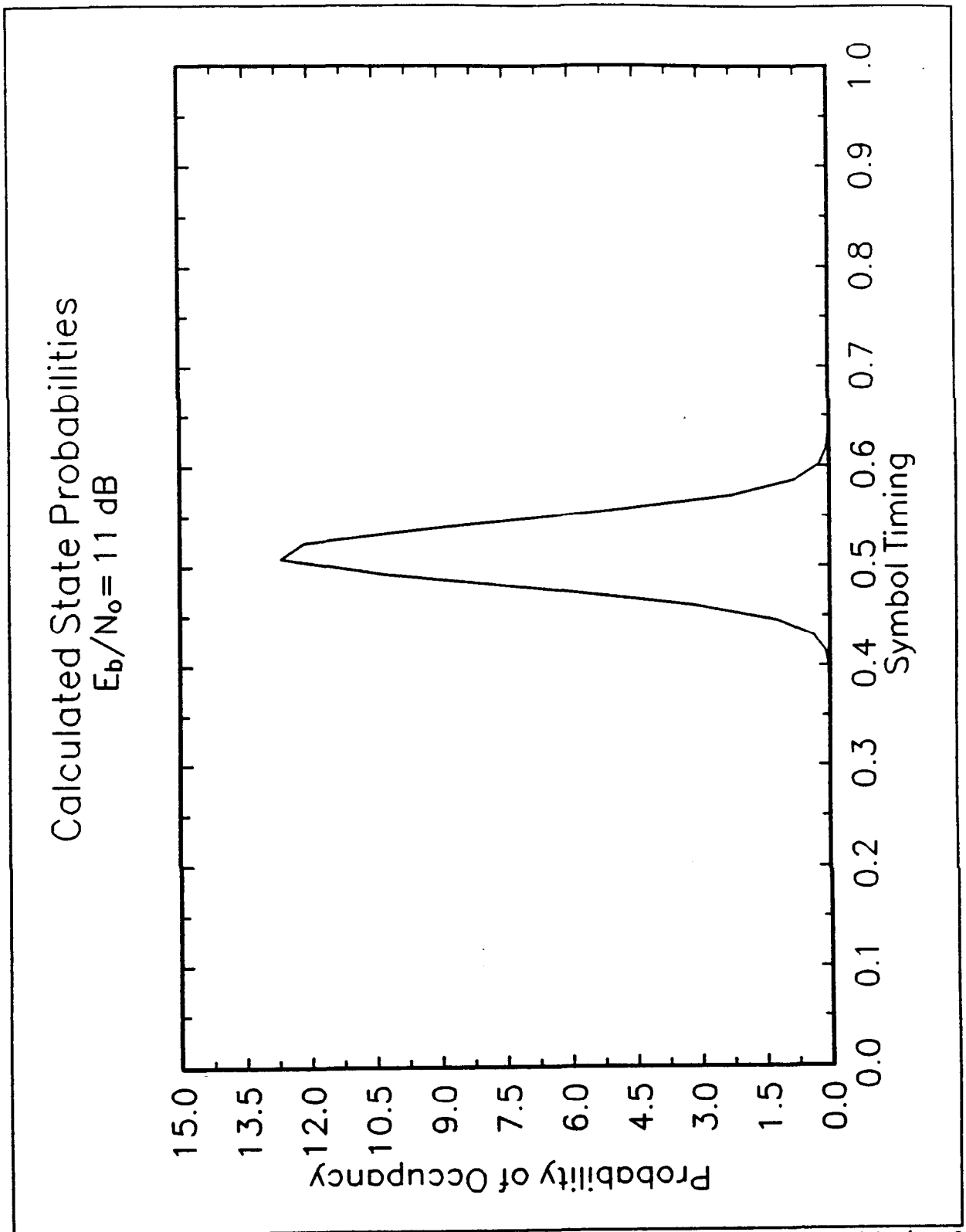


Figure A1.4 The steady-state probabilities were finally obtained as described in the main text from which the tracking variance can be easily found.