Continuous-Phase Frequency Shift Keying (FSK)

Contents

Continuous-Phase Frequency Shift Keying (FSK)

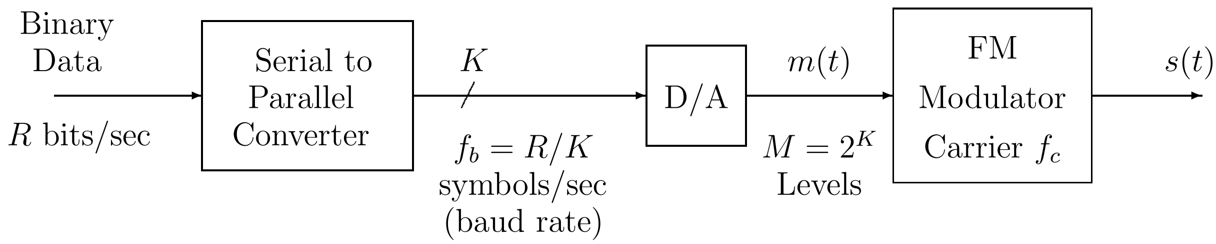
Continuous-phase frequency shift keying (FSK) is often used to transmit digital data reliably over wireline and wireless links at low data rates. Simple receivers with low error probability can be built. Binary FSK is used in most applications, often to send important control information.

- Early voice-band telephone line modems used binary FSK to transmit data at 300 bits per second or less and were acoustically coupled to the telephone handset. Teletype machines used these modems.

- Some HF radio systems use FSK to transmit digital data at low data rates.

- The 3GPP Cellular Text Telephone Modem (CTM) for use by the hearing impaired over regular cellular speech channels uses four level FSK.

The FSK Transmitter



At the FSK transmitter input, bits from a binary data source with a bit-rate of R bits per second are grouped into successive blocks of K bits by the "Serial to Parallel Converter." These block occur at the rate of $f_b = R/K$ symbols per second which is called the baud rate.

Each block is used to select one of $M = 2^K$ radian frequencies from the set

$$\begin{aligned} \Lambda_k &= \omega_c + \omega_d[2k - (M-1)] \\ &= 2\pi\{f_c + f_d[2k - (M-1)]\} \\ &\quad \text{for} \;\; k = 0, 1, \ldots, M-1 \end{aligned} \tag{1}$$

The frequency $\omega_c = 2\pi f_c$ is called the carrier frequency.

The FSK Transmitter (cont. 1)

The radian frequencies

$$\begin{aligned}\Omega_k &= \omega_d\,[2k-(M-1)] = 2\pi f_d[2k-(M-1)] \\ &\quad \text{for} \;\; k = 0,1,\ldots,M-1 \end{aligned} \tag{2}$$

are the possible frequency deviations from the carrier frequency during each symbol.

• The deviations range from $-\omega_d(M-1)$ to $\omega_d(M-1)$ in steps of $\Delta\omega = 2\omega_d$.

• Each selected frequency is sent for $T_b = 1/f_b$ seconds. The sinusoid transmitted during a block is called the FSK symbol specified by the block.

During the symbol period $nT_b \le t < (n+1)T_b$ the "D/A" box uniquely maps each possible input block to a possible frequency deviation

$$\Omega(n) = \omega_d\,[2k_n-(M-1)] \tag{3}$$

where k_n is the decimal value of the binary input block.

The FSK Transmitter (cont. 2)

The "D/A" block then forms the signal $\Omega(n)p(t-nT_b)$ where $p(t)$ is the unit height pulse of duration T_b defined as

$$p(t) = \begin{cases} 1 & \text{for } \ 0 \le t < T_b \\ 0 & \text{elsewhere} \end{cases} \tag{4}$$

Assuming transmission starts at $t = 0$, the complete "D/A" converter output is the staircase signal

$$m(t) = \sum_{n=0}^{\infty} \Omega(n)p(t-nT_b) \tag{5}$$

This baseband signal is applied to an FM modulator with carrier frequency ω_c and frequency sensitivity $k_\omega = 1$ to generate the FSK signal

$$s(t) = A_c \cos\left(\omega_c t + \int_0^t m(\tau)\, d\tau + \phi_0\right) \tag{6}$$

where A_c is a positive constant and ϕ_0 is a random initial phase angle of the modulator.

The FSK Transmitter (cont. 3)

The pre-envelope of $s(t)$ is

$$s_+(t) = A_c e^{j\omega_c t} e^{j\int_0^t m(\tau)\,d\tau} e^{j\phi_0} \tag{7}$$

and the complex envelope is

$$x(t) = A_c e^{j\int_0^t m(\tau)\,d\tau} e^{j\phi_0} \tag{8}$$

The phase contributed by the baseband message is

$$\begin{aligned} \theta_m(t) &= \int_0^t m(\tau)\,d\tau = \int_0^t \sum_{n=0}^{\infty} \Omega(n) p(\tau - nT_b)\,d\tau \\ &= \sum_{n=0}^{\infty} \Omega(n) \int_0^t p(\tau - nT_b)\,d\tau \qquad (9) \end{aligned}$$

Now consider the case when $iT_b \le t < (i+1)T_b$. Then

$$\theta_m(t) = \sum_{n=0}^{i-1} \Omega(n) T_b + \Omega(i) \int_{iT_b}^{t} d\tau$$

The FSK Transmitter (cont. 4)

$$\begin{aligned}
\theta_m(t) &= T_b\,\omega_d \sum_{n=0}^{i-1}[2k_n - (M-1)] + \\
&\quad T_b\,\omega_d[2k_i - (M-1)]\frac{(t - iT_b)}{T_b} \\
&= \pi\frac{2\omega_d}{\omega_b}\sum_{n=0}^{i-1}[2k_n - (M-1)] + \\
&\quad \pi\frac{2\omega_d}{\omega_b}[2k_i - (M-1)]\frac{(t - iT_b)}{T_b} \qquad (10)
\end{aligned}$$

• The *modulation index* for an FSK signal is defined to be

$$h = \frac{2\omega_d}{\omega_b} = \frac{\Delta\omega}{\omega_b} = \frac{\Delta f}{f_b} \qquad (11)$$

The phase at the start of the ith symbol is

$$\theta_m(iT_b) = \pi h \sum_{n=0}^{i-1}[2k_n - (M-1)] \qquad (12)$$

The FSK Transmitter (cont. 5)

Therefore,

$$\begin{aligned} \theta_m(t) &= \theta_m(iT_b) + \pi h\left[2k_i - (M-1)\right]\frac{(t-iT_b)}{T_b} \\ &\qquad \text{for } \; iT_b \le t < (i+1)T_b \end{aligned} \tag{13}$$

The phase function $\theta_m(t)$ is continuous and consists of straight line segments whose slopes are proportional to the frequency deviations.

Discrete-Time Transmitter Implementation

The FSK signal sample at time nT is

$$s(nT) = \cos\left(\omega_c nT + \int_0^{nT} m(\tau)\,d\tau + \phi_0\right) \tag{14}$$

The angle at time nT is

$$\theta(nT) = \omega_c nT + \int_0^{nT} m(\tau)\,d\tau + \phi_0 \tag{15}$$

Transmitter Implementation (cont.)

and the angle one sample in the future is

$$\begin{aligned}
\theta((n+1)T) &= \omega_c(n+1)T + \int\limits_0^{(n+1)T} m(\tau)\,d\tau + \phi_0 \\
&= \left[\omega_c nT + \int\limits_0^{nT} m(\tau)\,d\tau + \phi_0\right] \\
&\quad + \omega_c T + \int\limits_{nT}^{(n+1)T} m(\tau)\,d\tau \\
&= \theta(nT) + \omega_c T + Tm(nT) \qquad (16) \\
&= \theta(nT) + T\Lambda(nT) \qquad (17)
\end{aligned}$$

where $\Lambda(nT) = \omega_c + m(nT)$ is the total tone frequency at time nT. The M possible values for $T\Lambda(nT)$ can be pre-computed. Thus, the new phase angle can be computed recursively. In your transmitter program, make sure that $|\theta(nT)|$ does not get large.

Switched Oscillator FSK

Another approach to FSk would be to switch between independent tone oscillators. This switched oscillator approach could cause discontinuities in the phase function depending on the tone frequencies and symbol rate. The discontinuities would cause the resulting FSK signal to have a wider bandwidth than continuous phase FSK. It could also cause problems for a PLL demodulator.

Power Spectral Density for FSK

The term "power spectrum" will be used for "power spectral density" from here on for simplicity.

Lucky, Salz, and Weldon[1] present the solution for a slightly more generalized form of FSK than described above. Let the pulse $p(t)$ to have an arbitrary shape but still be confined to be zero outside the interval $[0, T_b)$. The power spectrum, $S_{xx}(\omega)$, of a random process $x(t)$ can be defined as:

$$S_{xx}(\omega) = \lim_{\lambda \to \infty} \frac{1}{\lambda} \mathrm{E}\left\{ \left| X_\lambda(\omega) \right|^2 \right\} \tag{18}$$

where E{ } denotes statistical expectation and

$$X_\lambda(\omega) = \int_0^\lambda x(t) e^{-j\omega t}\, dt \tag{19}$$

[1] R.W. Lucky, J. Salz, and E.J. Weldon, *Principles of Data Communications*, McGraw-Hill, 1968, pp. 203–207 and 242–245.

Power Spectral Density for FSK (cont. 1)

Only formulas for the power spectrum of the complex envelope will be presented here since the power spectrum for the complete FSK signal can be easily computed as

$$S_{ss}(\omega) = \frac{1}{4}S_{xx}(\omega - \omega_c) + \frac{1}{4}S_{xx}(-\omega - \omega_c) \quad (20)$$

• The frequency deviation in the complex envelope during the interval $[nT_b, (n+1)T_b)$ is

$$s_n(t - nT_b) = \Omega(n)p(t - nT_b) \quad (21)$$

• The phase change caused by this frequency deviation during the baud when time is taken relative to the start of the baud is

$$b_n(t) = \Omega(n)\int_0^t p(\tau)\,d\tau \quad \text{for} \;\; 0 \le t < T_b \quad (22)$$

• The total phase change over a baud is

$$B_n = b_n(T_b) = \Omega(n)\int_0^{T_b} p(\tau)\,d\tau \quad (23)$$

Power Spectral Density for FSK (cont. 2)

• The Fourier transform of a typical modulated pulse is

$$F_n(\omega) = \int_0^{T_b} e^{jb_n(t)} e^{-j\omega t}\, dt \tag{24}$$

It is convenient to define the following functions:

1. The characteristic function of $b_n(t)$

$$C(\alpha; t) = \mathrm{E}\left\{e^{j\alpha b_n(t)}\right\} \tag{25}$$

2. The average transform of a modulated pulse

$$F(\omega) = \mathrm{E}\left\{F_n(\omega)\right\} \tag{26}$$

3.

$$G(\omega) = \mathrm{E}\left\{\overline{F_n(\omega)} e^{jB_n}\right\} \tag{27}$$

4. The average squared magnitude of a pulse transform

$$P(\omega) = \mathrm{E}\left\{|F_n(\omega)|^2\right\} \tag{28}$$

Power Spectral Density for FSK (cont. 3)

5.

$$\gamma = \frac{1}{T_b} \arg C(1; T_b) \tag{29}$$

In terms of these quantities, the power spectrum is

$$\frac{T_b}{A_c^2}\, S_{xx}(\omega) = \begin{cases} P(\omega) + 2\Re e\left[F(\omega)G(\omega)\times \right. \\ \left. \frac{e^{-j\omega T_b}}{1 - C(1;T_b)e^{-j\omega T_b}}\right]; \quad |C(1;T_b)| < 1 \\ \\ P(\omega) - |F(\omega)|^2 + \omega_b \times \\ \displaystyle\sum_{n=-\infty}^{\infty} |F(\gamma + n\omega_b|^2\, \delta\left(\omega - \gamma - n\omega_b\right) \\ \text{for} \;\; C(1;T_b) = e^{j\gamma T_b} \end{cases} \tag{30}$$

Notice that the spectrum has discrete spectral lines as well as a distributed part when the characteristic function has unity magnitude.

Spectrum for Rectangular Frequency Pulses

The power spectrum for the case where $p(t)$ is the rectangular pulse given by (4) and the frequency deviations are equally likely reduces to

$$\frac{T_b}{A_c^2}\; S_{xx}(\omega) = \begin{cases} P(\omega) + 2\Re e\left[\dfrac{F^2(\omega)}{1 - C(1;T_b)e^{-j\omega T_b}}\right] \\ \quad \text{for} \;\; h = \dfrac{2\omega_d}{\omega_b} \;\; \text{not an integer} \\ \\ P(\omega) - |F(\omega)|^2 + \omega_b \times \\ \displaystyle\sum_{n=-\infty}^{\infty} \left|F(\gamma + n\omega_b)\right|^2 \delta\left(\omega - \gamma - n\omega_b\right) \\ \quad \text{for } h = \text{an integer } k \end{cases} \tag{31}$$

where

$$\gamma = \begin{cases} 0 & \text{for } k \text{ even} \\ \omega_b/2 & \text{for } k \text{ odd} \end{cases} \tag{32}$$

Spectrum for Rectangular Frequency Pulses (cont. 1)

$$F_n(\omega) = T_b \; \frac{\sin \dfrac{(\omega - \Omega_k)T_b}{2}}{\dfrac{(\omega - \Omega_k)T_b}{2}} \; e^{-j(\omega - \Omega_k)T_b/2} \tag{33}$$

$$P(\omega) = \frac{T_b^2}{M} \sum_{k=0}^{M-1} \left[\frac{\sin \dfrac{(\omega - \Omega_k)T_b}{2}}{\dfrac{(\omega - \Omega_k)T_b}{2}} \right]^2 \tag{34}$$

$$F(\omega) = \frac{T_b}{M} \sum_{k=0}^{M-1} \; \frac{\sin \dfrac{(\omega - \Omega_k)T_b}{2}}{\dfrac{(\omega - \Omega_k)T_b}{2}} \; e^{-j(\omega - \Omega_k)T_b/2} \tag{35}$$

and

$$C(1; T_b) = \frac{2}{M} \sum_{k=1}^{M/2} \cos[\omega_d T_b (2k-1)] = \frac{\sin(M\pi h)}{M \sin(\pi h)} \tag{36}$$

Properties of the Spectrum for Rectangular Frequency Pulses

- $F_n(\omega)$ has its peak magnitude at the tone frequency $\Omega_n = \omega_d[2n - (M-1)]$ and zeros at multiples of the symbol rate, ω_b, away from the tone frequency. This is exactly what would be expected for a burst of duration T_b of a sinusoid at the tone frequency.

- The term $P(\omega)$ is what would result for the switched oscillator case when the phases of the oscillators are independent random variables uniformly distributed over $[0, 2\pi)$.

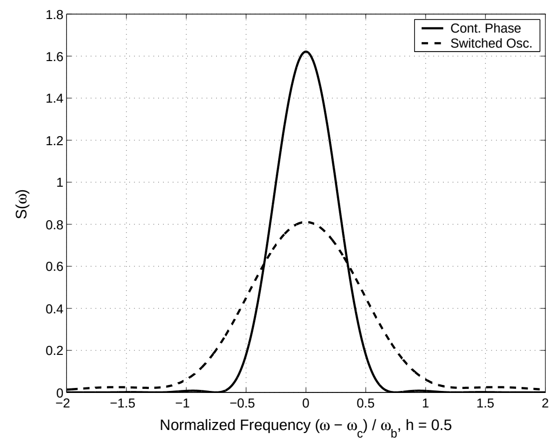
- The remaining terms account for the continuous phase property and give a narrower spectrum than if the the phase were discontinuous.

- The power spectrum has impulses at the M tone frequencies when h is an integer. However, the impulses at other frequencies disappear because they are multiplied by the nulls of $F(\gamma + n\omega_b)$.

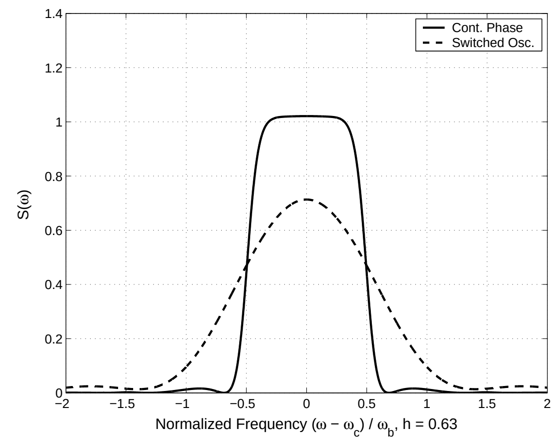
Specific Examples of FSK Spectra

Examples of the power spectral densities for binary continuous phase and switched oscillator FSK are shown in Figure 1 for $h = 0.5, 0.63, 1$ and 1.5. The spectra become more peaked near the origin for smaller values of h. They become more and more peaked near $-\omega_d$ and ω_d as h approaches 1 and include impulses at these frequencies when $h = 1$. Bell Labs designed its Bell 103 binary FSK modem, released in 1962, with $h = 2/3$ to avoid impulses in the spectrum that could cause cross talk in the cables.

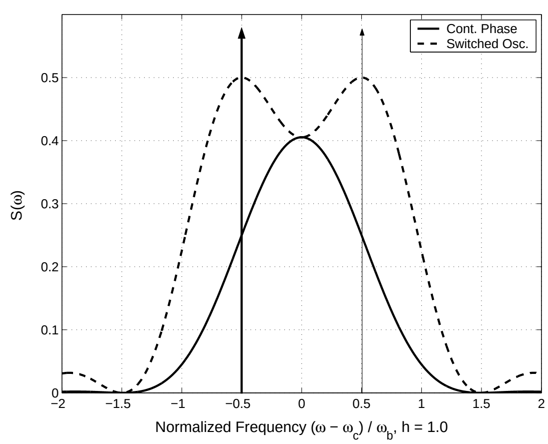
The spectra for $M = 4$ continuous phase and switched oscillator FSK are shown in Figure 2 for $h = 0.5, 0.63, 0.9$, and 1.5. The CTM with $M = 4$ uses a symbol rate of 200 baud with a tone separation of 200 Hz and, thus, has the modulation index $h = 1$.



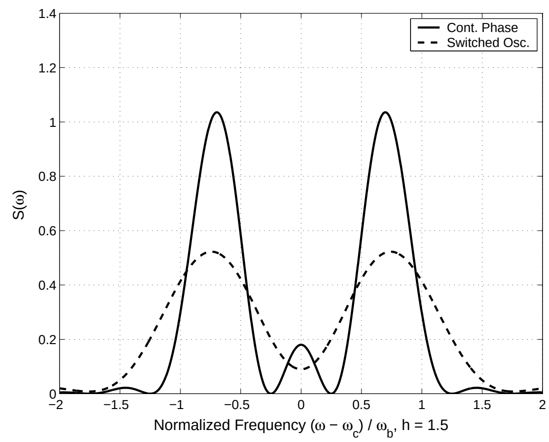
(a) $M = 2,\ h = 0.5$



(b) $M = 2,\ h = 0.63$

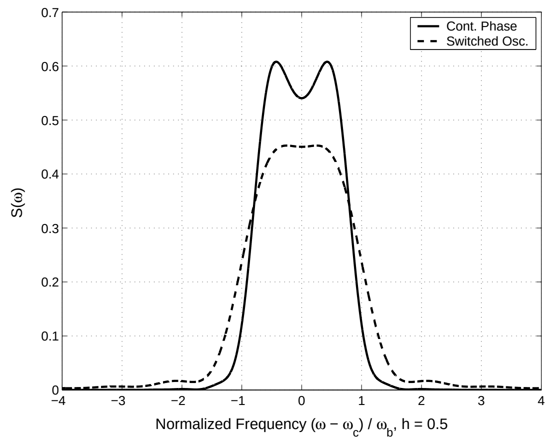


(c) $M = 2,\ h = 1$

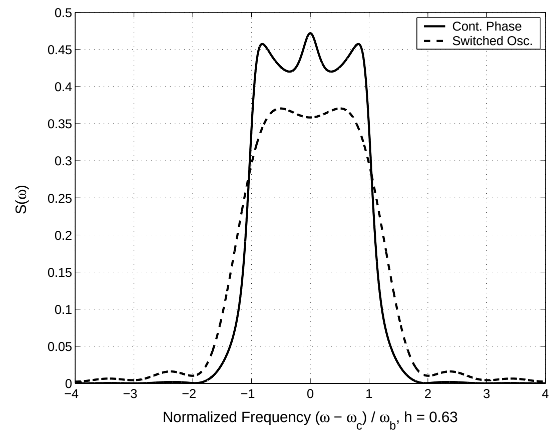


(d) $M = 2,\ h = 1.5$

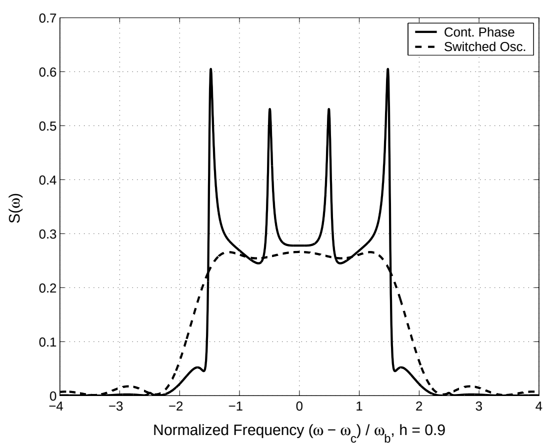
Figure 1: Normalized Power Spectral Densities $T_b S_{xx}(\omega)/A_c^2$ for Continuous Phase and Switched Oscillator Binary FSK for Several Values of h



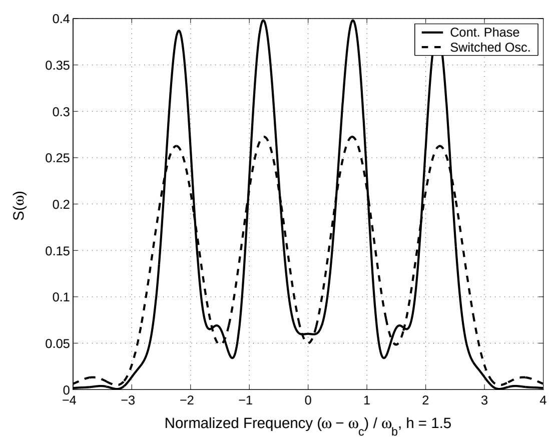
(a) $M = 4,\ h = 0.5$



(b) $M = 4,\ h = 0.63$



(c) $M = 4,\ h = 0.9$



(d) $M = 4,\ h = 1.5$

Figure 2: Normalized Power Spectral Densities $T_b S_{xx}(\omega)/A_c^2$ for Continuous Phase and Switched Oscillator $M = 4$ FSK for Several Values of h

FSK Demodulation

Continuous phase FSK signals can be demodulated using a variety of methods including:

1. **A frequency discriminator.**
   A frequency discriminator works well when signal-to-noise ratio (SNR) is high but performs poorly when the SNR is low.

2. **A phase-locked loop.**
   A phase-locked loop performs better at lower SNR but is not good when the FSK signal is present for short time intervals because a narrow-band loop takes a long time to acquire lock.

3. **Tone filters with envelope detectors.**
   Tone filters with envelope detection is theoretically the optimum noncoherent detection method when the FSK signal is corrupted by additive white Gaussian noise in terms of minimizing the symbol error probability.

The Frequency Discriminator

The complex envelope of the FM signal is

$$\begin{aligned} x(t) &= s_+(t)e^{-j\omega_c t} = A_c e^{j\int_0^t m(\tau)\,d\tau} e^{j\phi_0} \\ &= s_I(t) + j\, s_Q(t) \end{aligned} \tag{37}$$

The angle of the complex envelope is

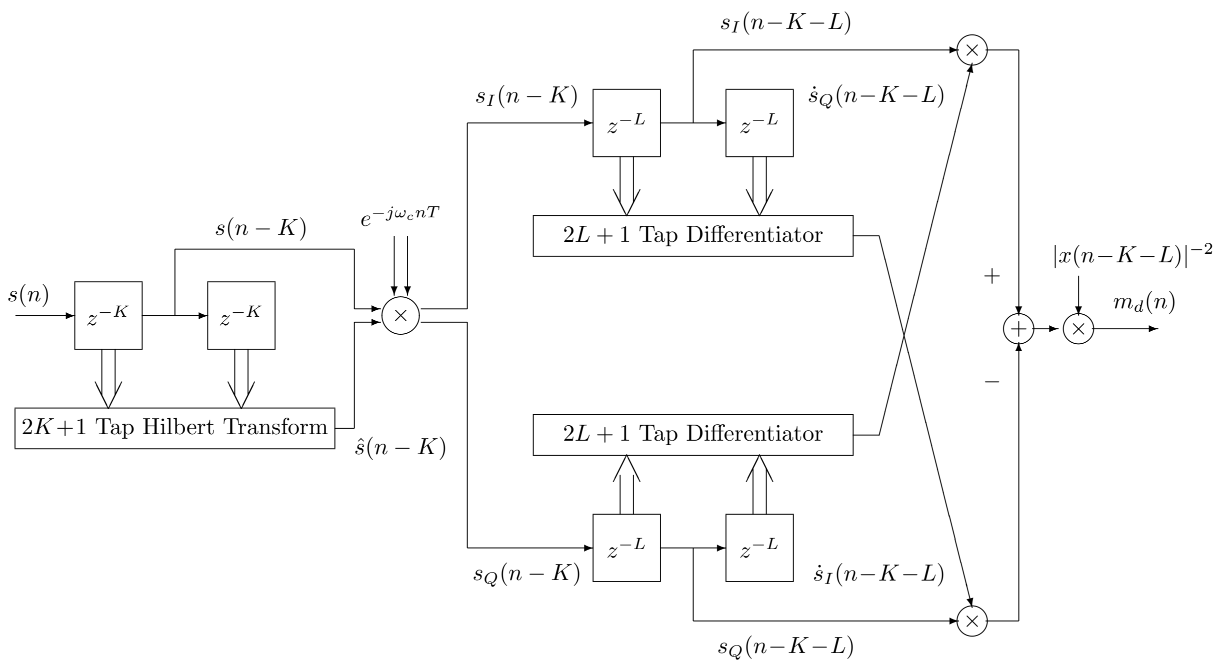
$$\tilde{\varphi}(t) = \arctan[s_Q(t)/s_I(t)] = \int_0^t m(\tau)\,d\tau + \phi_0 \tag{38}$$

and the derivative of this angle is

$$\frac{d}{dt}\,\tilde{\varphi}(t) = \frac{s_I(t)\dfrac{d}{dt}\,s_Q(t) - s_Q(t)\dfrac{d}{dt}\,s_I(t)}{s_I^2(t) + s_Q^2(t)} = m(t) \tag{39}$$

which is the desired message signal. A block diagram for implementing this discriminator is shown in Figure 3.

Figure 3: Discrete-Time Frequency Discriminator Realization Using the Complex Envelope



• First the pre-envelope is formed and demodulated to get the complex envelope whose real part is the inphase (I) component and imaginary part is the quadrature (Q) component.

• The frequency response of the differentiators must approximate $j\omega$ from $\omega = 0$ out to the cut-off frequency for the I and Q components which will be somewhat greater than the maximum frequency deviation $\omega_d(M-1)$.

Discriminator Implementation (cont.)

- The differentiator amplitude response should fall to a small value beyond the cut-off frequency because differentiation emphasizes high frequency noise which can cause a significant performance degradation.

- Notice how the delays through the FIR Hilbert transform filter and differentiation filters are matched by taking signals out of the center taps.

An example of the discriminator output is shown in Figure 4 when $f_c = 4000$ Hz, $f_d = 200$ Hz, and $f_b = 400$ Hz, so the modulation index is $h = 1$. The tone frequency deviations alternate between 200 and -200 Hz for eight symbols followed by two symbols with -200 Hz deviation.

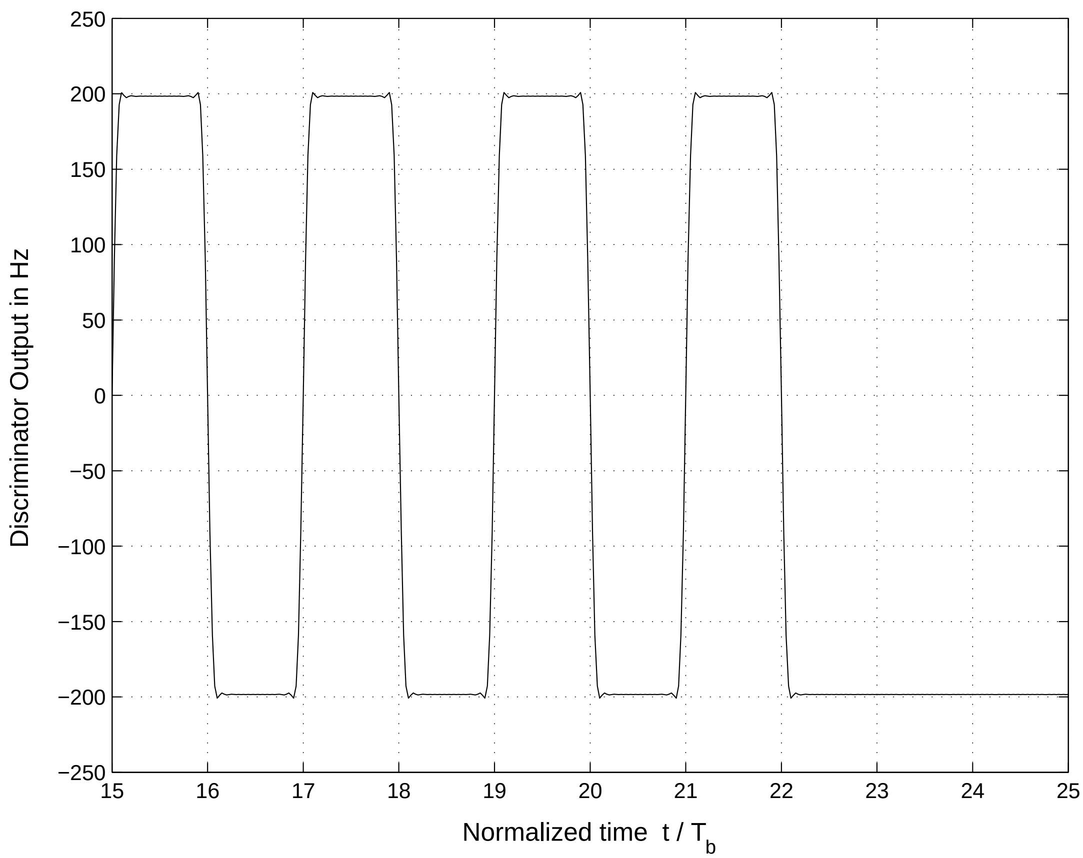


Figure 4: Discriminator Output for $h = 1$

A Simple Approximate Frequency Discriminator

A simpler approximate discriminator will be derived in this subsection. Let $1/T = f_s$ be the sampling rate. Usually there will be multiple samples per symbol so $T << T_b$. Using the complex envelope the following product can be formed:

$$\begin{aligned}
c(nT) &= \frac{1}{A_c^2 T}\,\mathrm{Im}\left\{x(nT)\overline{x(nT-T)}\right\} \\
&= \frac{1}{T}\,\mathrm{Im}\left\{e^{j\left[\int_0^{nT} m(\tau)\,d\tau+\phi_0\right]}e^{-j\left[\int_0^{nT-T} m(\tau)\,d\tau+\phi_0\right]}\right\} \\
&= \frac{1}{T}\mathrm{Im}\left\{e^{j\int_{nT-T}^{nT} m(\tau)\,d\tau}\right\} = \frac{1}{T}\sin\int_{nT-T}^{nT} m(\tau)\,d\tau \\
&\approx \frac{1}{T}\sin[Tm(nT-T)] = \frac{1}{T}\sin[m(nT-T)/f_s] \\
&\approx \quad m(nT-T)
\end{aligned} \tag{40}$$

To get the final result, the approximation $\sin x \simeq x$ for $|x| << 1$ was used.

Approximate Discriminator (cont.)

In terms of the inphase and quadrature components

$$c(nT) = \frac{1}{A_c^2 T}\left[s_Q(nT)s_I(nT-T) - s_I(nT)s_Q(nT-T)\right] \tag{41}$$

and this is the discriminator equation that would be implemented in a DSP.

As another approach, suppose the derivatives in (39) are approximated at time nT by

$$\frac{d}{dt}\, s_I(t)|_{t=nT} \simeq \frac{s_I(nT) - s_I(nT-T)}{T} \tag{42}$$

and

$$\frac{d}{dt}\, s_Q(t)|_{t=nT} \simeq \frac{s_Q(nT) - s_Q(nT-T)}{T} \tag{43}$$

Subsituting these approximate derivatives into (39) gives $\frac{d}{dt}\,\tilde{\varphi}(t)|_{t=nT} \simeq c(nT)$ exactly as in the previous approach.

Symbol Clock Acquisition and Tracking

The discriminator output will look like an M-level PAM signal with rapid changes at the symbol boundaries where the frequency deviation has changed. The discriminator output must be sampled once per symbol at the correct time to estimate the transmitted frequency deviation and, hence, the input data bit sequence.

When the signal-to-noise ratio is large at the receiver, the sharp transitions in the discriminator output can be detected.

- A method for doing this is to form the absolute value of the derivative of the discriminator output. This will generate a positive pulse whenever the output level changes.
- A pulse location can be determined by looking for a positive threshold crossing.

Symbol Clock Acquisition (cont. 1)

- Then the symbol can be sampled in its middle by waiting for half the symbol period, $T_b/2$, after the pulse detection before sampling the discriminator output level.

- The absolute value of the derivative will be very small in the middle of the symbol and a search for the next peak can be started.

- The derivative will be zero at the symbol boundaries where the levels do not change. Therefore, the search for a new peak should only extend for slightly more than $T_b/2$. If no new peak is found by that time then successive symbol levels are the same and the start of the next symbol should be estimated as the sampling time in the middle of the last symbol plus $T_b/2$. This process can then be repeated for each successive symbol.

Symbol Clock Acquisition (cont. 2)

In lower SNR environments, the method for generating a symbol clock signal for PAM signals discussed in Chapter 11 can be used. This involves

- passing the discriminator output through a bandpass filter with center frequency at $f_b/2$,
- squaring the filter output,
- and passing the result through a bandpass filter with a center frequency at the symbol rate f_b.
- The receiver can then lock to the positive zero crossings of the resulting clock signal and sample the discriminator output with an appropriate delay from the zero crossings.

The Phase-Locked Loop

The block diagram of a phase-locked loop (PLL) that can be used to demodulate a continuous phase FSK signal is shown in Figure 5. The theory for this PLL is discussed extensively in Chapter 8 and the main points are summarized in this subsection.

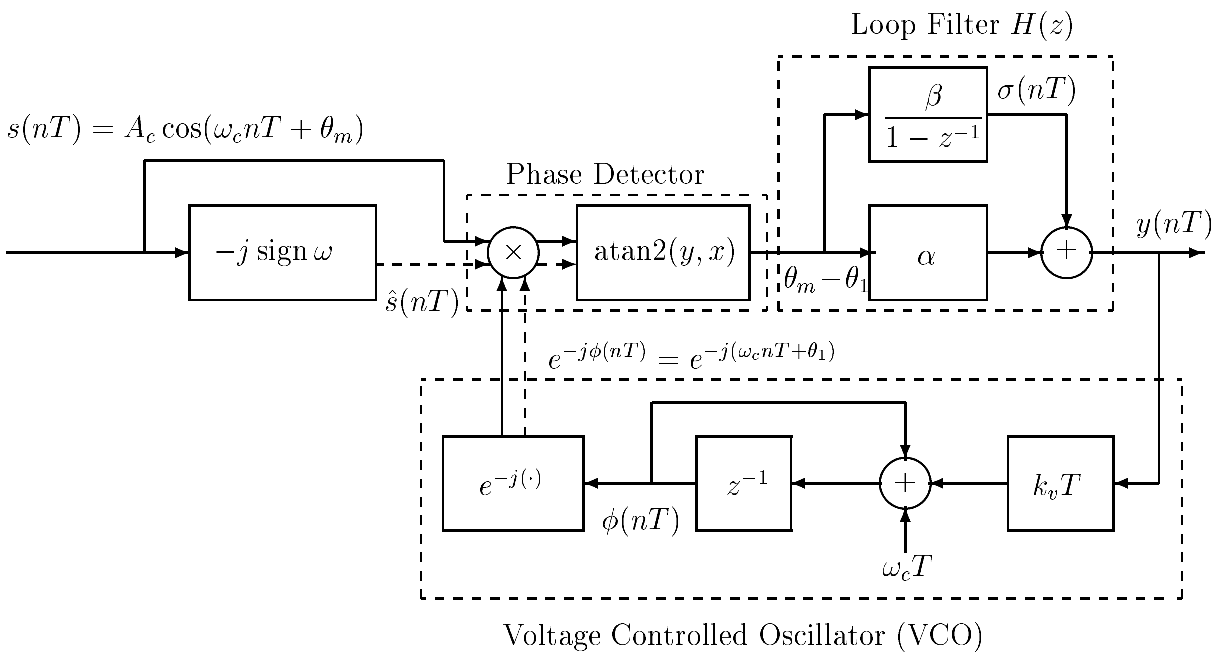


Figure 5: Phase-Locked Loop for FSK Demodulation

The Phase-Locked Loop (cont. 1)

First, the received FSK signal is sampled with period T and passed through a discrete-time Hilbert transform filter to form the pre-envelope $s_+(nT) = s(nT) + j\hat{s}(nT)$.

Suppose there are L samples per baud so that $T_b = LT$. Then for $iT_b \leq nT < (i+1)T_b$, $n = iL + \ell$ for some integer ℓ with $0 \leq \ell \leq L-1$. From (6) and (10) it follows that the total phase angle of the pre-envelope during baud i is

$$\Theta(nT) = \omega_c nT + T_b \sum_{k=0}^{i-1} \Omega(k) + \Omega(i)\ell T + \phi_0$$

$$\text{for } \; 0 \leq \ell < L-1 \qquad (44)$$

The PLL contains a voltage controlled oscillator (VCO) which generates a complex exponential sinusoid at the carrier frequency ω_c when its input is zero.

The Phase-Locked Loop (cont. 2)

The PLL acts to make the VCO total angle $\phi(nT) = \omega_c nT + \theta_1(nT)$ equal to the angle of the pre-envelope.

The multiplier in the Phase Detector box demodulates the pre-envelope using the replica complex exponential carrier generated by the VCO.

The phase error between the angles of the pre-envelope and replica carrier is computed by the C arctangent function $\text{atan2}(y, x)$ where y is the imaginary part of the multiplier output and x is its real part.

The parameters α and β in the Loop Filter are positive constants. Typically, $\beta < \alpha/50$ to make the loop have a tranient response to a phase step without excessive overshoot.

The Phase-Locked Loop (cont. 3)

The accumulator generating $\sigma(nT)$ is included so that the loop will track a carrier frequency offset.

The parameter k_v is also a positive constant. The product, $\alpha k_v T$, controls the tracking speed of the loop. It should be large enough so the loop tracks the input phase changes, but small enough so the loop is stable and not strongly influenced by additive input noise.

The VCO generates its phase angle by the following recursion:

$$\phi(nT+T) = \phi(nT) + \omega_c T + k_v T y(nT) \qquad (45)$$

Therefore

$$k_v y(nT) = \frac{\phi(nT+T) - \phi(nT)}{T} - \omega_c \qquad (46)$$

During baud i and assuming the loop is perfectly in lock so that $\Theta(nT) = \phi(nT)$, substituting $\Theta(nT)$ given by (44) for $\phi(nT)$ into (46) gives

The Phase-Locked Loop (cont. 4)

$$k_v y(nT) = \frac{\Theta(nT+T) - \Theta(nT)}{T} - \omega_c = \Omega(i) \quad (47)$$

Therefore, the PLL is an FSK demodulator.

When the loop is in lock and the phase error is small, atan2(x, y) can be closely approximated by the imaginary part of the complex multiplier output divided by A_c. The multiplier output is

$$\begin{aligned}
&[s(nT) + j\hat{s}(nT)]e^{-j\phi(nT)} = \\
&\qquad A_c e^{j[\omega_c nT + \theta_m(nT)]} e^{-j\phi(nT)} \\
&\qquad\qquad = A_c e^{j[\phi_m(nT) - \theta_1(nT)]} \qquad (48)
\end{aligned}$$

and its imaginary part is

$$\begin{aligned}
&\hat{s}(nT)\cos\phi(nT) - s(nT)\sin\phi(nT) \\
&= A_c \sin[\phi_m(nT) - \theta_1(nT)] \simeq A_c[\phi_m(nT) - \theta_1(nT)] \qquad (49)
\end{aligned}$$

where $A_c = |s(nT) + j\hat{s}(nT)|$.

The Phase-Locked Loop (cont. 5)

The imaginary part can be divided by the computed A_c or this scaling can be accomplished by an automatic gain control (AGC) in the receiver or by adjusting the loop parameters. The loop gain in the PLL and, hence, its transient response depend on A_c if the approximation (49) is used, so this normalization by A_c is important. The atan2(y, x) function automatically does the normalization.

An example of the PLL behavior is shown in Figure 6 for a binary FSK input signal. The binary data input to the modulator was a PN sequence generated by a 23-stage feedback shift register. The carrier frequency was 4 kHz, the frequency deviation was 200 Hz, and the baud rate was 400 Hz. The output shows a segment where the input alternated between 0 and 1 followed by a string of 1's.

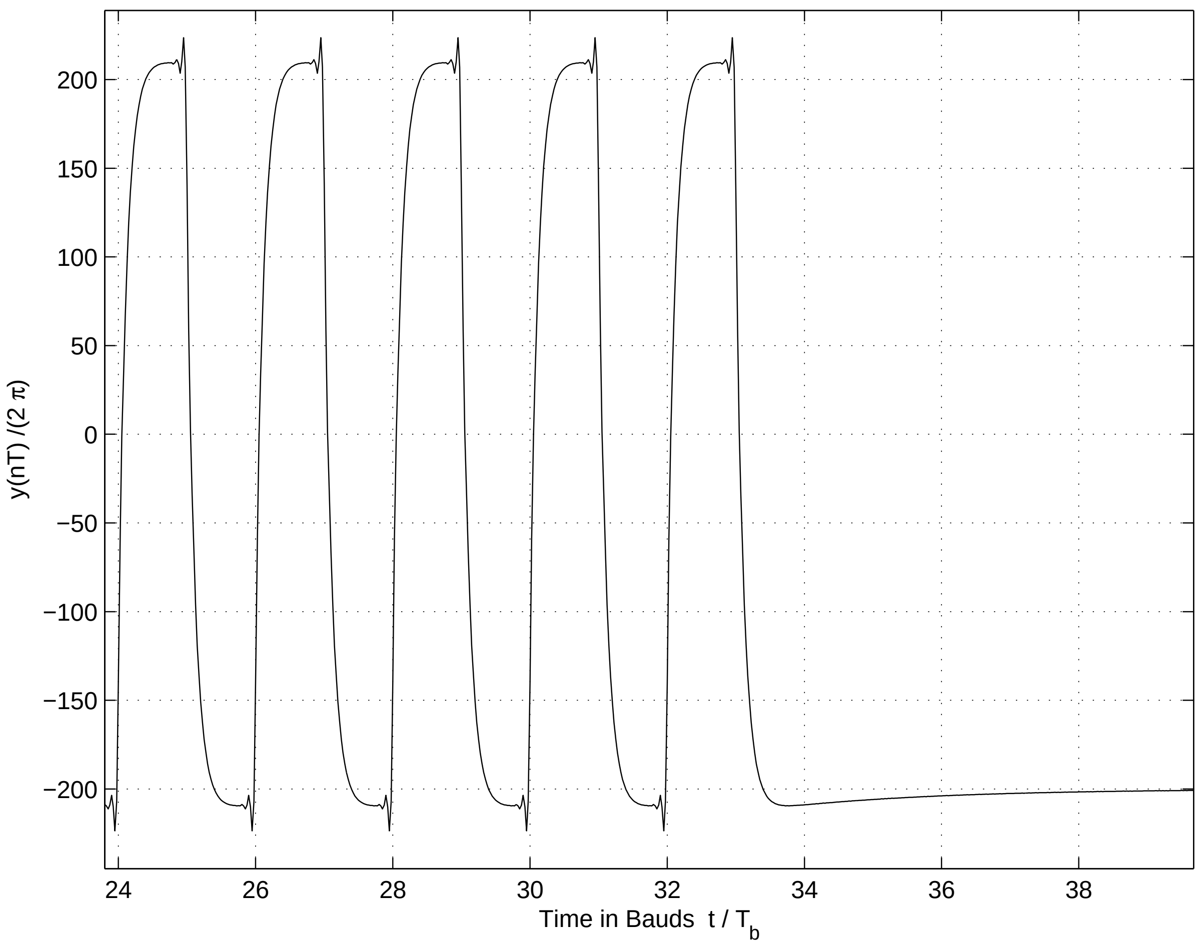


Figure 6: PLL Output with $1/T = 16000$ Hz, $k_v = 1$, $\alpha k_v T = 0.2$, and $\beta = \alpha/100$

Optimum Noncoherent Detection by Tone Filters

• The FM discriminator performs very poorly when the SNR is low or the FSK signal is distorted, for example, by a speech compression codec in a cell phone, because differentiation emphasizes noise.

• The phase-locked loop demodulator performs better than the discriminator at low SNR but can have difficulty locking on to FSK signals that are present in short bursts.

• A better detector for these cases that uses "tone filters" will now be described . This approach does not use knowledge of the carrier phase and is called noncoherent detection.

• A result in detection theory is that in the presence of additive white Gaussian noise the detection strategy that is optimum in the sense of minimizing the symbol error probability for symbol interval N is to compute the following statistics for the symbol interval and decide that the frequency that was transmitted corresponds to the largest statistic[2]:

$$
\begin{aligned}
I_k(N) &= \left[\int_{NT_b}^{(N+1)T_b} s(t)\cos(\Lambda_k t + \epsilon)\, dt \right]^2 \\
&\quad + \left[\int_{NT_b}^{(N+1)T_b} s(t)\sin(\Lambda_k t + \epsilon)\, dt \right]^2 \qquad (50) \\
&= \left| \int_{NT_b}^{(N+1)T_b} s(t) e^{-j(\Lambda_k t + \epsilon)}\, dt \right|^2 \\
&= \left| \int_{NT_b}^{(N+1)T_b} s(t) e^{-j\Lambda_k t}\, dt \right|^2 \\
&\qquad \text{for } k = 0, \ldots, M-1 \qquad (51)
\end{aligned}
$$

where $\Lambda_k = \omega_c + \Omega_k = \omega_c + \omega_d[2k - M - 1]$ is the

[2] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, John Wiley & Sons, Inc., 1965, pp. 511–523.

total tone frequency, $s(t)$ is the noise corrupted received signal, and ϵ is a conveniently selected phase angle. Notice that the statistics have the same value for every choice of ϵ.

Computing $I_k(N)$ Using Oscillators and Integrators

The receiver could implement (50) in the obvious way. It could have a set of M oscillators, each generating an inphase sine wave $\cos \Lambda_k t$ and a quadrature sine wave $\sin \Lambda_k t$. Then it would multiply the input, $s(t)$, by the sine waves, integrate the products over each baud, and form the sum of the squares of the inphase and quadrature integrator outputs for each tone frequency. The receiver would then decide that the tone frequency corresponding to the largest statistic was the one that was transmitted for that baud.

Computing $I_k(N)$ by a Bank of Filters

The statistics can also be generated using a bank of "tone" filters. Let the impulse response of the kth tone filter be

$$h_k(t) = \begin{cases} e^{j\Lambda_k t} & \text{for } \ 0 \leq t < T_b \\ 0 & \text{elsewhere} \end{cases} \tag{52}$$

The output of this filter when the input is $s(t)$ is

$$\begin{aligned} y_k(t) &= \int_{t-T_b}^{t} s(\tau) e^{j\Lambda_k (t-\tau)} \, d\tau \\ &= \int_{t-T_b}^{t} s(\tau) e^{-j\Lambda_k \tau} \, d\tau \; e^{j\Lambda_k t} \qquad (53) \\ &= \int_{t-T_b}^{t} s(\tau) \cos[\Lambda_k (t-\tau)] \, d\tau \\ &\quad + j \int_{t-T_b}^{t} s(\tau) \sin[\Lambda_k (t-\tau)] \, d\tau \qquad (54) \end{aligned}$$

Using Tone Filters (cont. 1)

Let $t = (N+1)T_b$ which is at the start of symbol interval $N+1$ or the end of symbol interval N. Then

$$|y_k(NT_b + T_b)|^2 = \left| \int_{NT_b}^{(N+1)T_b} s(\tau) e^{-j\Lambda_k \tau} \, d\tau \right|^2 \tag{55}$$

which is the desired statistic $I_k(N)$ given by (51).

The frequency response of a tone filter is

$$\begin{aligned} H_k(\omega) &= \int_0^{T_b} e^{j\Lambda_k t} e^{-j\omega t} \, dt = \frac{1 - e^{-j(\omega - \Lambda_k)T_b}}{j(\omega - \Lambda_k)} \\ &= e^{-j(\omega - \Lambda_k)T_b/2} \, T_b \, \frac{\sin[(\omega - \Lambda_k)T_b/2]}{(\omega - \Lambda_k)T_b/2} \end{aligned} \tag{56}$$

The magnitude of this function has a peak at the tone frequency Λ_k and zeros spaced at distances that are integer multiples of $\omega_b = 2\pi/T_b$ away from the peak. Thus, the tone filters are bandpass filters with center frequencies equal to the M tone frequencies.

Discrete-Time Tone Filter Implementation

The integrals can be approximated by sums. Suppose there are L samples per symbol so that $T_b = LT$. Then the last term on the right of (51) can be approximated by

$$I_k(N)/T^2 \simeq D_k(N) = \left| \sum_{\ell=NL}^{(N+1)L-1} s(\ell T) e^{-j\Lambda_k \ell T} \right|^2 \tag{57}$$

A discrete-time approximation to the tone filter is

$$h_k(nT) = \begin{cases} e^{j\Lambda_k nT} & \text{for } n = 0, 1, \ldots, L-1 \\ 0 & \text{elsewhere} \end{cases} \tag{58}$$

and the output of this filter is

$$\begin{aligned} y_k(nT) &= \sum_{\ell=n-L+1}^{n} s(\ell T) e^{j\Lambda_k (n-\ell)T} \\ &= e^{j\Lambda_k nT} \sum_{\ell=n-L+1}^{n} s(\ell T) e^{-j\Lambda_k \ell T} \end{aligned} \tag{59}$$

Discrete-Time Tone Filter Implementation (cont. 1)

Notice that each of the M tone filter impulse responses is convolved with the same set of L samples $\{s(\ell T)\}_{\ell=n-L+1}^{n}$. Therefore, an efficient implementation in terms of minimum memory usage should have only one delay line containing these L samples.

The decision statistics for symbol interval N are obtained at the end of this interval by letting $n = (N+1)L - 1$ in (59) to get

$$\begin{aligned} D_k(N) &= \left| y_k[(N+1)LT - T] \right|^2 \\ &= \left| \sum_{\ell=NL}^{(N+1)L-1} s(\ell T) e^{-j\Lambda_k \ell T} \right|^2 \end{aligned} \tag{60}$$

Frequency Response of the Discrete-Time Tone Filter

The frequency response for tone filter k is

$$
\begin{aligned}
H_k(\omega) &= \sum_{n=0}^{L-1} e^{j\Lambda_k nT} e^{-j\omega nT} \\
&= e^{-j(\omega-\Lambda_k)(L-1)T/2} \, \frac{\sin[(\omega-\Lambda_k)LT/2]}{\sin[(\omega-\Lambda_k)T/2]} \qquad (61)
\end{aligned}
$$

• The amplitude response of this filter has a peak of value L at the tone frequency Λ_k.

• It has zeros at frequencies $\Lambda_k + p\,\omega_b$ in the interval $0 \leq \omega < \omega_s$ where p is an integer and $\omega_b = 2\pi/T_b$.

• It repeats periodically outside this interval as would be expected for the transform of a sampled signal.

• This tone filter is a bandpass filter centered at the tone frequency Λ_k.

Block Diagram of Receiver Using Tone Filters

The block diagram of a receiver using tone filters is shown in Figure 7 for $M = 4$.

• The boxes labelled "Complex BPF" are the tone filters.

• The solid line at the output of a box is the real part of the output and the dotted line is the imaginary part.

• The boxes labelled $|\cdot|$ form the squared complex magnitudes of their inputs. These squared magnitudes are the squared envelopes of the tone filter outputs.

• The squared envelopes are sampled at the end of each symbol period, the largest is found, and the corresponding frequency deviation is assumed to be the one that was actually transmitted. This decision is then mapped back to the corresponding bit pattern.

Efficient Computation with FIR Tone Filters

If the receiver has locked its local symbol clock frequency to that of the received signal and the phase for sampling at the end of a symbol has been determined, then

• the convolution sum in (59) only has to be computed at the sampling times.

• In between sampling times, new samples must be shifted into the filter delay line but the output does not have to be computed.

• In practice the clocks will continually drift and must be tracked. The block diagram indicates that the tone filter outputs are computed for each new input sample. It will be shown below how a signal for clock tracking can be derived from these signals.

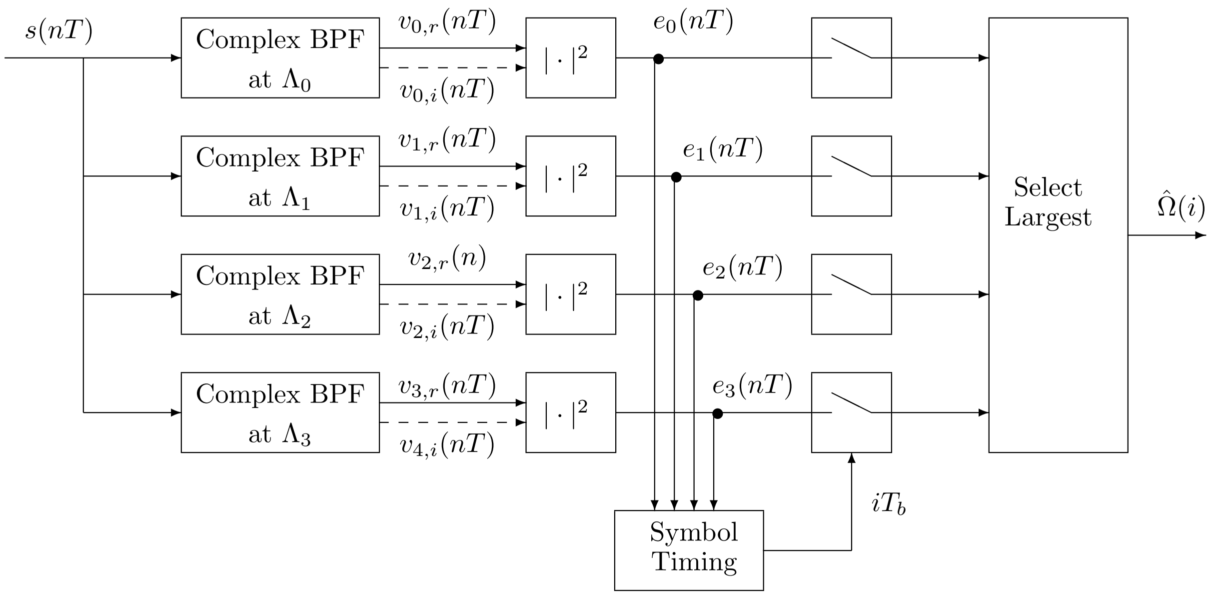


Figure 7: FSK Demodulator Using Tone Filters for $M = 4$

Recursive Implementation of the Tone Filters

The tone filters can be efficiently implemented recursively when their outputs must be computed at every sampling time. To ensure stability of the recursion, the tone filter impulse responses will be slightly modified to $g_k(nT) = r^n h_k(nT)$ where r is slightly less than 1. The z-transform of a modified tone filter impulse response is

$$G_k(z) = \sum_{n=0}^{L-1} r^n e^{j\Lambda_k nT} z^{-n} = \frac{1 - r^L e^{j\Lambda_k LT} z^{-L}}{1 - re^{j\Lambda_k T} z^{-1}} \tag{62}$$

The output of this modified tone filter can be computed as

$$\begin{aligned} y_k(nT) = s(nT) - r^L e^{j\Lambda_k LT} s(nT - LT) \\ + re^{j\Lambda_k T} y_k(nT - T) \end{aligned} \tag{63}$$

Recursive Implementation of the Tone Filters (cont. 1)

The real part of $y_k(nT)$ is

$$\begin{aligned} v_{k,r}(nT) &= \Re e\{y_k(nT)\} \\ &= s(nT) - r^L \cos(\Lambda_k LT) s(nT - LT) \\ &\quad + r\cos(\Lambda_k T) v_{k,r}(nT-T) \\ &\quad - r\sin(\Lambda_k T) v_{k,i}(nT-T) \end{aligned} \tag{64}$$

and the imaginary part is

$$\begin{aligned} v_{k,i}(nT) &= \Im m\{y_k(nT)\} \\ &= -r^L \sin(\Lambda_k LT) s(nT - LT) \\ &\quad + r\cos(\Lambda_k T) v_{k,i}(nT-T) \\ &\quad + r\sin(\Lambda_k T) v_{k,r}(nT-T) \end{aligned} \tag{65}$$

These last two equations are what could actually be implemented in a DSP since additions and multiplications must operate on real quantities in a DSP. This filter structure is sometimes called a "cross-coupled" implementation.

Computation for the Recursive Implementation

The quantities $r^L \cos(\Lambda_k LT)$, $r^L \sin(\Lambda_k LT)$, $r \cos(\Lambda_k T)$, and $r \sin(\Lambda_k T)$ can be precomputed.

• Then, computation of the real and imaginary outputs for the cross-coupled form requires six real multiplications and five real additions for each n.

• Computation by direct convolution requires $2(L-1)$ real multiplications and $2(L-1)$ real additions for each n since $h_k(0) = g_k(0) = 1$ and this is usually much larger than the computation required for the cross-coupled form.

• The signal memory required for the cross-coupled form is an $L+1$ word buffer to store the real values $\{s(\ell T)\}_{\ell=n-L}^{n}$ plus two locations to store $v_{k,r}(nT-T)$ and $v_{k,i}(nT-T)$. This is just slightly more than required by the direct convolution method.

Simplified Demodulator for Binary FSK

The demodulator structure can be simplified for binary ($M = 2$) FSK. A block diagram of the simplified demodulator is shown in Figure 8.

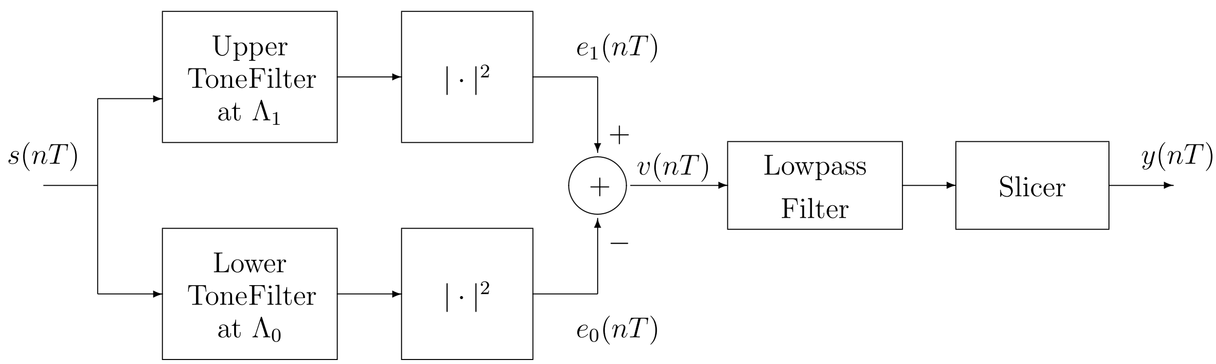


Figure 8: Simplified FSK Demodulator for $M = 2$

- The squared envelopes of the two tone filter outputs are computed as before but now one is subtracted from the other.
- This difference is passed through a lowpass filter to smooth it and eliminate some noise.

Simplified Demodulator for Binary FSK (cont.)

• The slicer hard limits its input to a positive voltage A if its input is positive and to a negative voltage $-A$ if its input is negative.

• When no noise is present on the transmission channel, a slicer output of A indicates that the frequency deviation Ω_1 was transmitted and an output of $-A$ indicates that Ω_0 was transmitted during the symbol interval.

Generating a Symbol Clock Timing Signal

In a low noise environment and when the system filters are wideband, the symbol clock can be tracked by locking to the sharp transitions in the demodulator output.

This will not work in a high noise environment and when the system filters cause gradual transitions.

Generating a Symbol Clock Timing Signal (cont. 1)

One way to generate a signal for clock tracking is to form the sum, $c(nT)$, of the $\binom{M}{2} = M(M-1)/2$ absolute values of the differences of the pairs of different tone filter output squared envelopes. In equation form

$$c(nT) = \sum_{0 \le i < j \le M-1} |\, e_i(nT) - e_j(nT)| \qquad (66)$$

The idea behind this is that during each symbol where the tone frequency changes from the one in the previous symbol, the tone filter output for the previous tone will ring down and the tone filter output for the new tone will ring up, so the absolute value of the difference will show a transition. The tone filter envelopes can then be sampled at the peaks of $c(nT)$, the largest envelope determined, and the result mapped back to a data bit sequence.

Adding a Bandpass Filter to the Clock Tracker

When the tone frequency is the same in adjacent symbols, the tone filter output envelopes will not change and $c(nT)$ will not have a transition between the symbols. A symbol clock tracking algorithm based on $c(nT)$ would have to "flywheel" through the symbol intervals where $c(nT)$ has no transitions.

A solution to this problem is to pass $c(nT)$ through a bandpass filter centered at the symbol rate f_b. A simple 2nd order bandpass filter with nulls at 0 and $f_s/2$ Hz and a peak near f_b Hz has the transfer function

$$H(z) = (1-r)\,\frac{1-z^{-2}}{1-2z^{-1}r\cos(2\pi f_b/f_s)+r^2z^{-2}} \tag{67}$$

where f_s is the sampling rate and r is a number close to but slightly less than 1.

Adding a Bandpass Filter to the Clock Tracker (cont. 1)

The closer r is to 1, the more narrow the filter bandwidth.

Let $c(nT)$ be the filter input, $y(nT)$ the filter output, and $v(nT)$ an internal filter signal. Then the filter output can be computed recursively by the equations

$$\begin{aligned} v(nT) &= (1-r)c(nT) + 2r\cos(2\pi f_b/f_s)v(nT-T) \\ &\quad -r^2 v(nT-2T) \qquad (68) \\ y(nT) &= v(nT) - v(nT-2T) \qquad (69) \end{aligned}$$

This filter will "ring" at the symbol clock frequency. The receiver's clock tracker can lock to the positive zero crossings of this signal. The slope of the filter output $y(nT)$ is a maximum at the zero crossings. Therefore, the zero crossings can be determined with significantly higher accuracy than the peaks where the slope is zero.

Adding a Bandpass Filter to the Clock Tracker (cont. 2)

The peaks of the tone filter envelopes will occur with some delay from these zero crossings depending on the filter parameters. The tone filter squared envelopes should be sampled with this delay from the zero crossings.

The bandpass filter output will continue to oscillate at the symbol rate but will decay exponentially through intervals where the input is constant because of no tone frequency changes.

By choosing r close enough to 1, the output will remain large enough during the intervals with no transitions to still detect the positive zero crossings and allow the clock tracker to automatically flywheel through these intervals.

Detecting the Presence of an FSK Signal

The presence of an FSK signal can be detected by monitoring the sum of the M squared envelopes

$$\rho(nT) = \sum_{k=0}^{M-1} e_k(nT) \tag{70}$$

This sum indicates the power received in the tone filter pass bands. Detection of an FSK signal can be declared when $\rho(nT)$ exceeds a threshold for one or more samples.

The termination of an FSK signal can be declared when the sum falls below a threshold. The termination threshold can be set below the detection threshold to allow hysteresis and reduce false loss of signal detection.

An $M = 4$ FSK Example Using Tone Filters

Typical signals for an $M = 4$ FSK signal with $f_d = 200$ Hz, $f_c = 4000$ Hz, $f_b = 400$ Hz, $f_s = 16000$ Hz, and tone filter detection are shown in Figures 9, 10, 11, 12, and 13. The tone frequencies are 3400, 3800, 4200, and 4600 Hz. For the tone filters $r = 0.999$ and for the clock bandpass filter $r = 0.998$.

Figure 9 shows a small segment of the FSK signal. The tone frequency for symbols during normalized times 10 to 11 and 12 to 13 is 4600 Hz. The tone frequency during times 11 to 12 and 13 to 14 is 3400 Hz. The varying amplitudes is an optical illusion created by connecting samples of the signal taken at a 16000 Hz rate with straight lines.

Figure 9: Segment of the FSK Signal

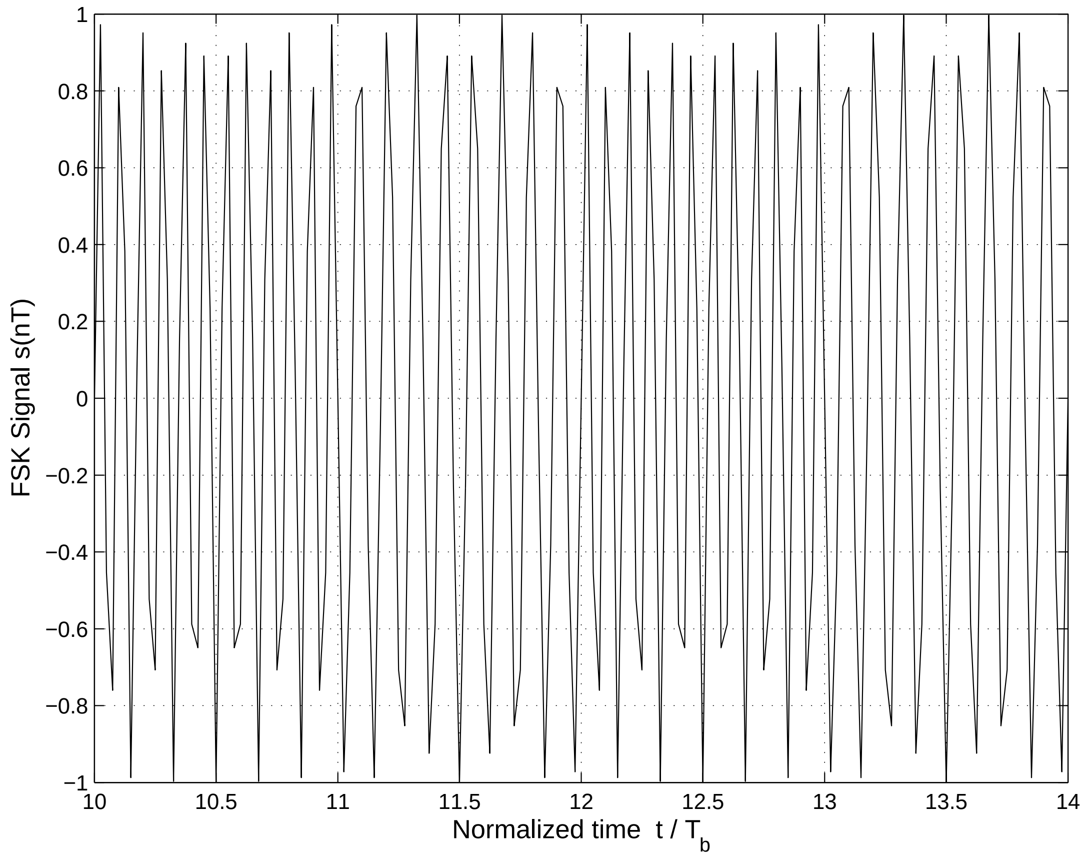


Figure 10 shows the squared envelope $e_0(nT)$ at the output of the 3400 Hz tone filter. Notice that the peaks occur at the integer normalized times.

Figure 10: 3400 Hz Tone Filter Output Squared Envelope

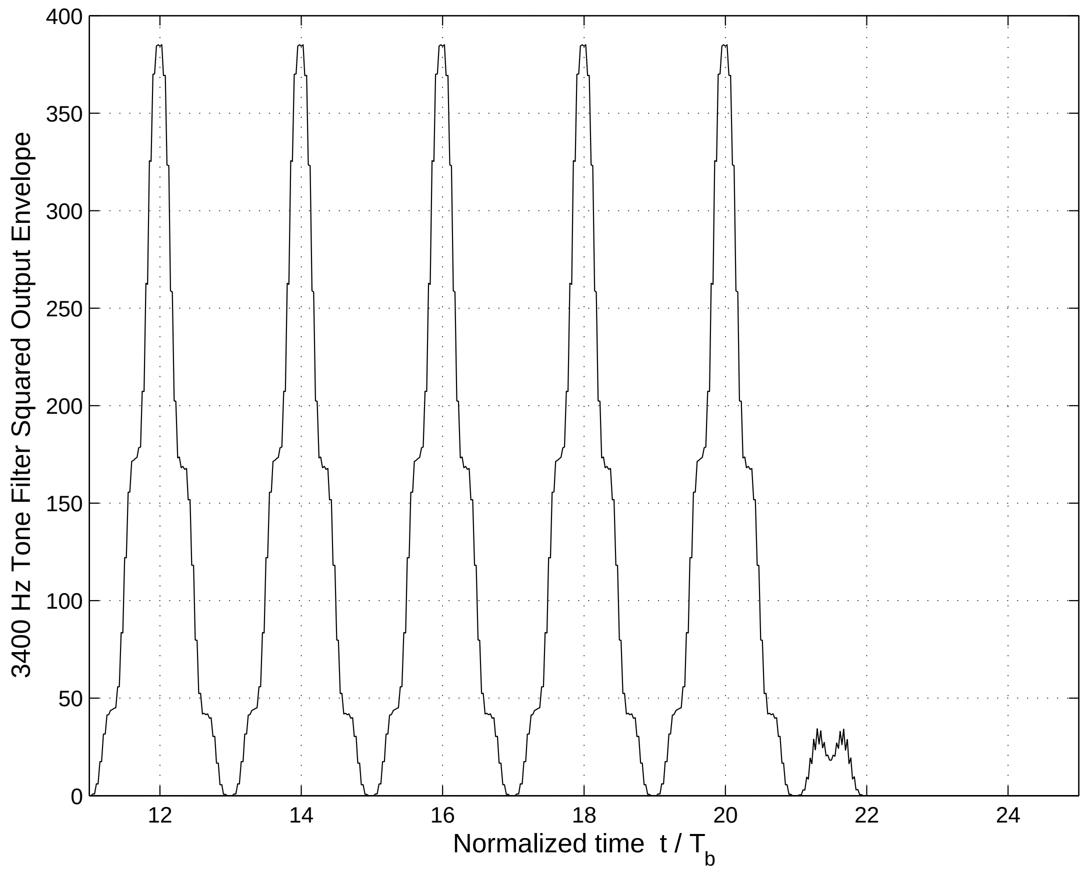


Figure 11 shows a segment of the preliminary clock signal $c(nT)$ computed as

$$
\begin{aligned}
c(nT) = &\,|\,e_0(nT) - e_1(nT)| + |\,e_0(nT) - e_2(nT)| \\
&+ |\,e_0(nT) - e_3(nT)| + |\,e_1(nT) - e_2(nT)| \\
&+ |\,e_1(nT) - e_3(nT)| + |\,e_2(nT) - e_3(nT)| \quad (71)
\end{aligned}
$$

$M = 4$ Example (cont.)

In this example, the tone frequencies for symbols 10 through 21 alternate between 3400 and 4600 Hz creating peaks in $c(nT)$ each symbol as the envelopes of the corresponding two tone filters charge up and down.

The tone frequency remains constant during symbols 22, 23, and 24, so there is no change in the tone filter output squared envelopes and $c(nT)$ had no transitions.

Observe that the peaks occur at the integer normalized times which are exactly where the peaks in $e_0(nT)$ occur in Figure 10. One could lock to the peaks in $c(nT)$ and sample the envelopes at the peak times but would have to "flywheel" through intervals when the tone frequency does not change.

Figure 11: The Preliminary Symbol Clock Tracking Signal $c(nT)$

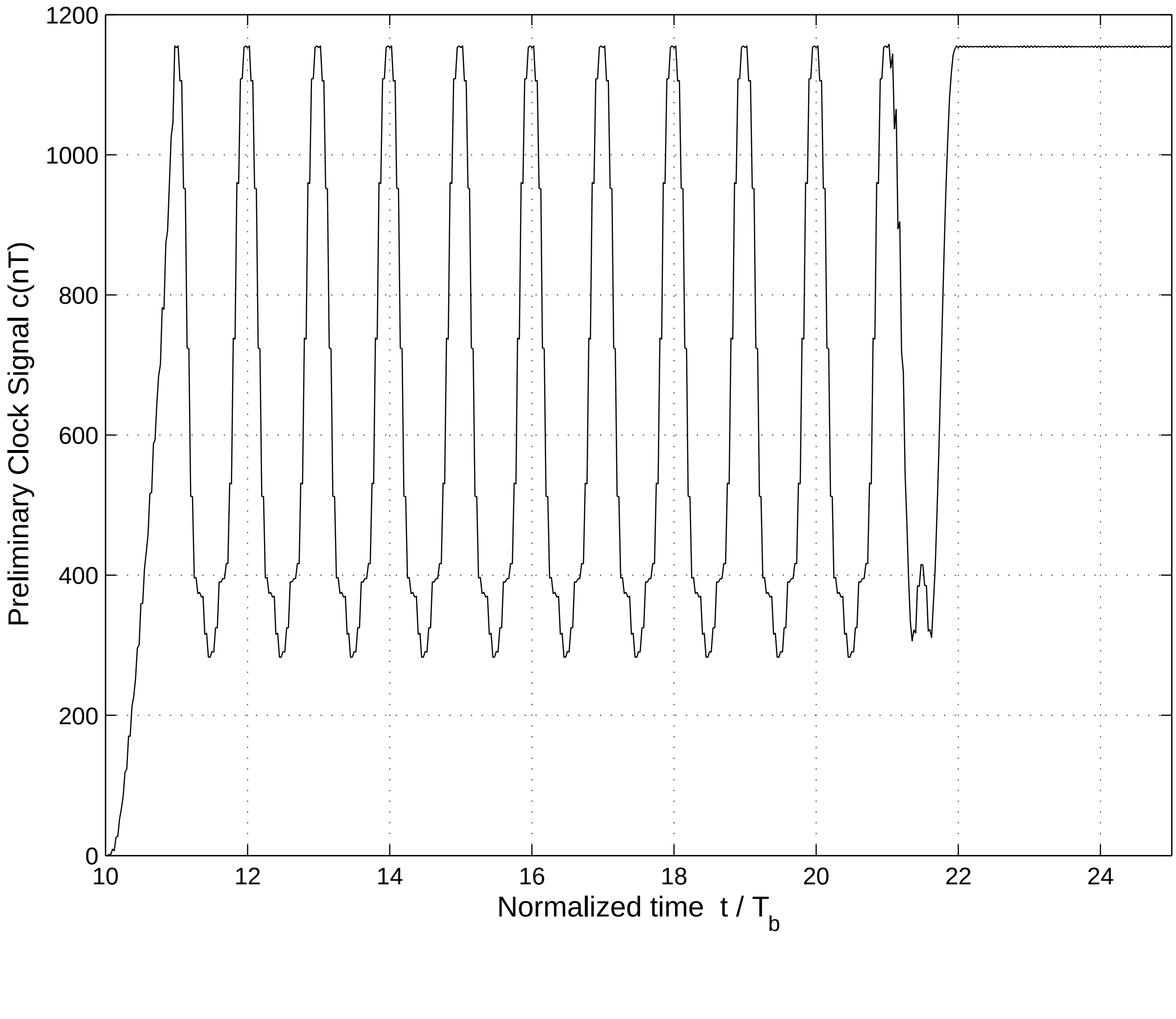


Figure 12: The Signal $c(nT)$ Passed through a 2nd Order Bandpass Filter

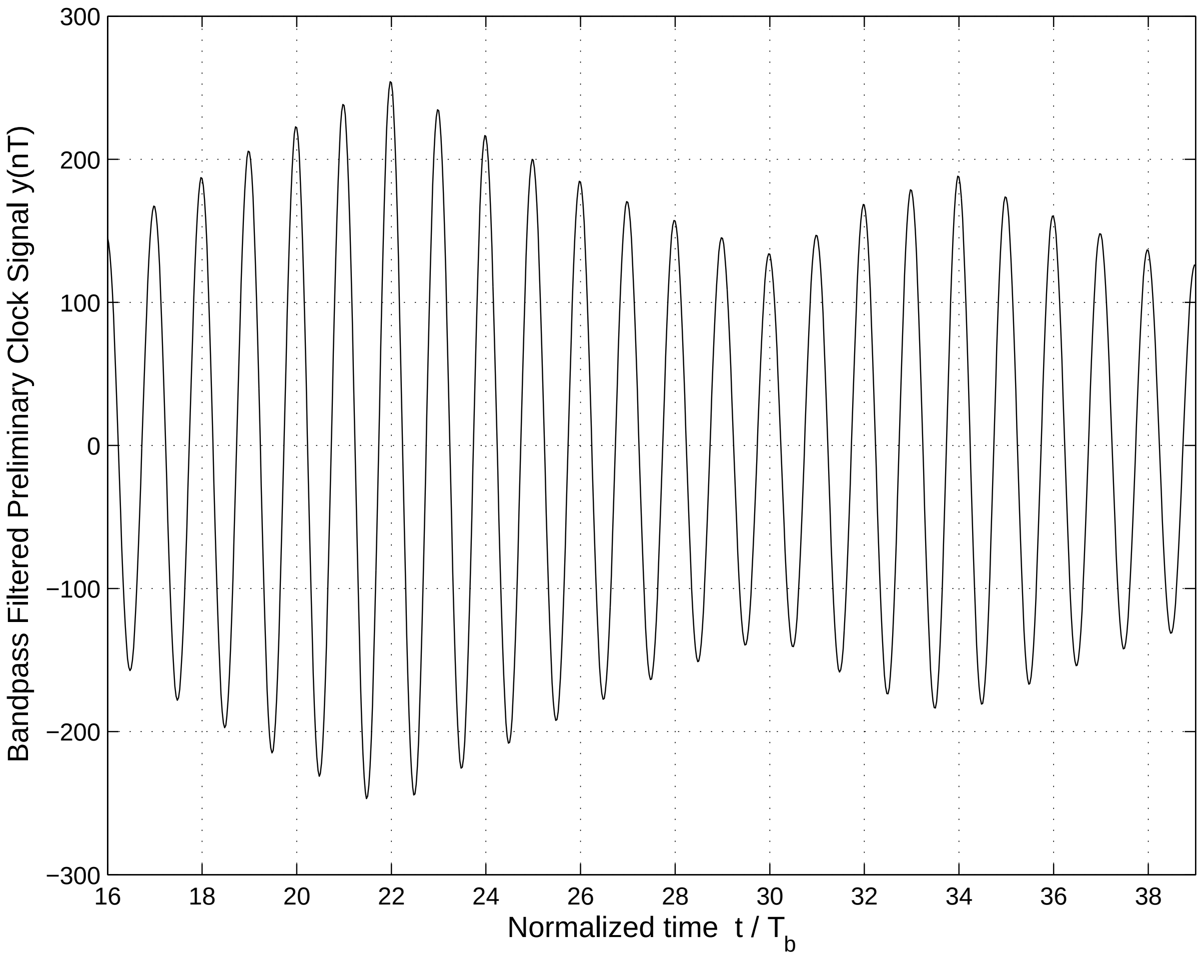


Figure 12 shows the result when $c(nT)$ is passed through the bandpass filter centered at the symbol clock frequency. Notice that the signal is exponentially damped between normalized times 22 and 30 when $c(nT)$ has no transitions.

Figure 13: Superimposed Preliminary and Bandpass Filtered Clock Signals

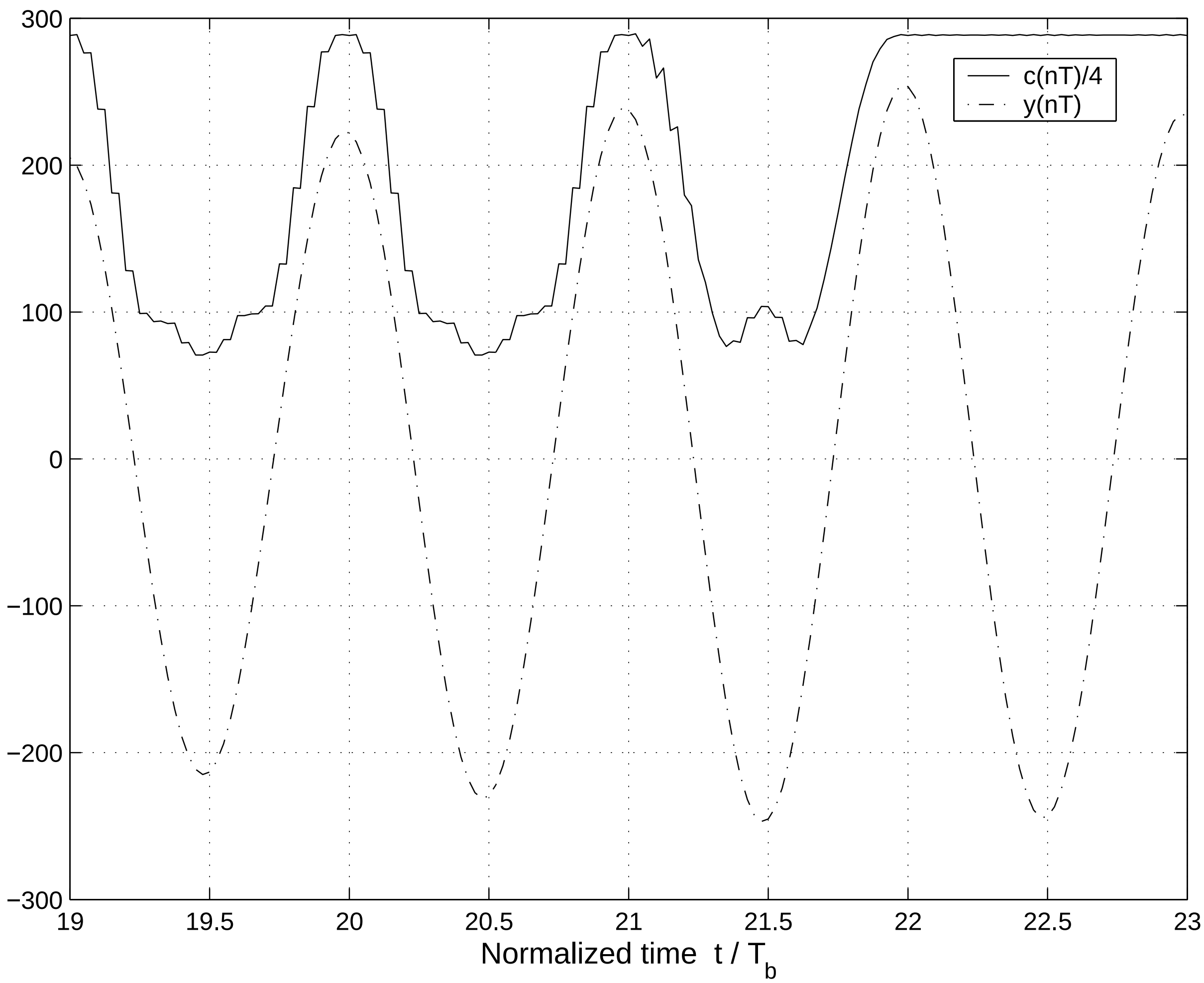


Figure 13 shows a segment of the preliminary clock signal, $c(nT)$, and the bandpass filtered clock signal, $y(nT)$, superimposed on the same graph. The peaks of $y(nT)$ occur at almost the same times as the peaks in $c(nT)$.

The positive zero crossings of $y(nT)$ occur 1/4 of a symbol before the peaks of $c(nT)$. A good clock tracker would lock to these zero crossing and the receiver would then sample the tone filter squared envelopes with a delay of 1/4 of a symbol which corresponds to the peaks of $c(nT)$. The exact delay necessary depends on the filter parameters.

Symbol Error Probabilities for FSK Systems

The problem of computing the symbol error probability for different types of FSK receivers is discussed extensively in Chapter 8 of Lucky, Salz, and Weldon[3]. The problem is very difficult because of the nonlinear natures of the modulator and various receivers. Many of the results are approximations or require evaluation of complicated integrals by numerical integration.

[3] R.W. Lucky, J. Salz, and E.J. Weldon, *Principles of Data Communication*, McGraw-Hill Book Company, 1968

Orthogonal Signal Sets

One case where exact closed form results are know is when the transmitted symbols are orthogonal, they are corrupted by additive white Gaussian noise, and optimum noncoherent detection by tone filters is used.

Two continuous-time signals over the interval $[t_1, t_2)$ with complex envelopes $x_1(t)$ and $x_2(t)$ are said to be orthogonal if

$$\rho = \int_{t_1}^{t_2} x_1(t)\overline{x_2(t)}\,dt = 0 \tag{72}$$

From (13) it follows that the complex envelopes of the FSK signal set during symbol period i where $iT_b \le t < (i+1)T_b$ are

$$x_k(t) = A_c e^{j\theta_m(iT_b)} e^{j\omega_d[2k-(M-1)](t-iT_b)}$$
$$\text{for } k = 0, \ldots, M-1 \tag{73}$$

Orthogonal Signal Sets (cont. 1)

For two distinct integers k_1 and k_2 in this set

$$\begin{aligned}
\rho &= \int_{iT_b}^{(i+1)T_b} x_{k_1}(t)\overline{x_{k_2}(t)}\,dt \\
&= A_c^2 \int_{iT_b}^{(i+1)T_b} e^{j2\omega_d(k_1-k_2)(t-iT_b)}\,dt] \\
&= A_c^2 \int_0^{T_b} e^{j2\omega_d(k_1-k_2)t}\,dt \\
&= A_c^2\,\frac{e^{j2\omega_d(k_1-k_2)T_b}-1}{j2\omega_d(k_1-k_2)} \qquad (74)
\end{aligned}$$

This integral will be zero if

$$2\omega_d(k_1-k_2)T_b = 2\pi\ell \qquad (75)$$

or

$$h(k_1-k_2) = \ell \qquad (76)$$

where ℓ is an integer. This will be satisfied for all pairs of signals in the FSK signal set when the modulation index, h, is an integer.

Orthogonal Signal Sets (cont. 2)

An analogous property holds for the discrete-time FSK approximation. Assume there are L samples per symbol so that $T_b = LT$. The complex discrete-time envelopes during symbol interval i where $iT_b \le nT < (i+1)T_b$ are

$$x_k(nT) = A_c e^{j\theta_m(iT_b)} e^{j\omega_d[2k-(M-1)](nT-iLT)} \quad \text{for } k = 0, \ldots, M-1 \tag{77}$$

Then for two distinct integers k_1 and k_2 the correlation is

$$\begin{aligned} \rho &= \sum_{n=iL}^{(i+1)L-1} x_{k_1}(nT)\overline{x_{k_2}(nT)} \\ &= A_c^2 \sum_{n=0}^{L-1} e^{j2\omega_d(k_1-k_2)nT} \\ &= A_c^2 \frac{1 - e^{j2\omega_d(k_1-k_2)LT}}{1 - e^{j2\omega_d(k_1-k_2)T}} \end{aligned} \tag{78}$$

The correlation ρ will be zero if $2\omega_d(k_1 - k_2)LT = 2\omega_d(k_1 - k_2)T_b = 2\pi\ell$ where ℓ is an integer just as in the continuous-time case.

Orthogonal Signal Sets (cont. 3)

The energy transmitted during symbol period i is

$$\begin{aligned}\mathcal{E} &= \int_{iT_b}^{(i+1)T_b} s^2(t)\,dt = \frac{1}{2}\int_{iT_b}^{(i+1)T_b} |x_k(t)|^2\,dt \\ &= A_c^2 T_b/2 \qquad (79)\end{aligned}$$

and the average power transmitted during this interval is $S = \mathcal{E}/T_b$. Let the two-sided noise power spectral density be $N_0/2$. Then the symbol error probability is[4]

$$P_e = \frac{\exp\left(-\frac{\mathcal{E}}{N_0}\right)}{M} \sum_{i=2}^{M} (-1)^i \binom{M}{j} \exp\left(\frac{\mathcal{E}}{iN_0}\right) \qquad (80)$$

For binary FSK, *i.e.*, $M = 2$, the symbol error probability is

$$P_e = \frac{1}{2}\exp\left(-\frac{\mathcal{E}}{2N_0}\right) \qquad (81)$$

[4] Andrew J. Viterbi, *Principles of Coherent Communication*, McGraw-Hill, 1966, p. 247.

Orthogonal Signal Sets (cont. 4)

An upper bound for the symbol error probability for arbitrary M is

$$P_e \leq \frac{M-1}{2} \exp\left(-\frac{\mathcal{E}}{2N_0}\right) \qquad (82)$$

There are $k = \log_2 M$ bits per symbol. For orthogonal signal sets, all symbol errors are equally likely, so all bit-error patterns in a block of k transmitted bits assigned to a symbol are equally likely. Based on this observation, Viterbi[5] shows that the bit error probability is related to the symbol error probability by the formula

$$P_b = \frac{2^{k-1}}{2^k - 1} P_e \qquad (83)$$

---

[5] Andrew J. Viterbi, *Principles of Coherent Communication*, McGraw-Hill, 1966, p. 226.

Experiments for Continuous-Phase FSK

For these experiments you will explore $M = 2$ and $M = 4$ continuous-phase FSK transmitters and receivers. For all these experiments use the following parameters: carrier frequency $f_c = 4000$ Hz, frequency deviation $f_d = 200$ Hz, symbol rate $f_b = 400$ Hz, sampling frequency $f_s = 16000$ samples per second, and $p(t)$ is the rectangular pulse given by (4). Initialize the TMS320C6713 DSK as usual.

EXP 1. Theoretical FSK Spectra

Write a MATLAB program or use any other favorite programming language to compute the power spectral density for an FSK signal with arbitrary M, f_c, f_d, and f_b using (31). Then plot the spectra for $M = 2$ and $M = 4$ *vs.* the normalized frequency $(\omega - \omega_c)/\omega_b$ for the parameters specified for these experiments. Experiment with other parameters also.

EXP 2. Making FSK Transmitters

Write programs for the TMS320C6713 DSK to implement continuous-phase FSK transmitters for $M = 2$ and $M = 4$. Write the output samples to the left codec output channel. You will be using these transmitters as FSK signal sources for your receivers.

EXP 2.1 Initial Handshaking Sequence To help the receivers detect the presence of an FSK signal and lock to the transmitter's symbol clock, make your transmitter send the following signal sequence:

1. First send 0.25 seconds of silence, that is, send 0 volts for 0.25 seconds. This will allow your receiver to skip over any initial transient that occurs when the transmitter program is loaded and started.

2. Then for $M = 2$ send 25 symbols alternating each symbol between $f_0 = 3800$ and $f_1 = 4200$ Hz tones. This will allow the receivers to detect the FSK signal and lock on to the symbol clock. For $M = 4$ send 25 symbols alternating each symbol between $f_0 = 3400$ and $f_3 = 4600$ Hz.

3. Suppose the frequencies of the last few symbols of the alternating sequence for $M = 2$ were $\cdots, f_0, f_1, f_0$. Next send an alternating frequency sequence for 10 symbols but with the alternation reversed. That is send $f_0, f_1, f_0, f_1, f_0, f_1, f_0, f_1, f_0, f_1$. Your receiver can detect this change in the alternations and use it as a timing mark to determine when actual data will start.

   For the $M = 4$ transmitter change to alternating between $f_1 = 3800$ and $f_2 = 4200$ Hz for 10 symbols. Again, this change can be used as a timing mark.

EXP 2.2 Simulating Random Customer Data

After the alternations, begin transmitting "customer" data continuously.

Simulate this data by using a 23 stage PN sequence generator as discussed in Chapter 9.

Use the connection polynomial

$$h(D) = 1 + D^{18} + D^{23}$$

so the data bit sequence, $d(n)$, is generated by the recursion

$$d(n) = d(n-18) \oplus d(n-23) \tag{84}$$

where "$\oplus$" in the recursion is modulo 2 addition, that is, the exclusive-or operation. Initialize the PN sequence generator shift register to some non-zero state.

For $M = 2$, shift the PN generator once to get a new data bit $d(n)$ which will be a 0 or 1. Map this bit to the tone frequency $\Lambda(n) = \omega_c + \omega_d[2d(n) - 1]$.

For $M = 4$, shift the register twice to get a pair of bits $[d_1(n), d_0(n)]$. Consider this bit pair to be the integer $k(n) = 2d_1(n) + d_0(n)$ which can be 0, 1, 2, or 3. Map this bit pair to the tone frequency $\Lambda(n) = \omega_c + \omega_d[2k(n) - 3]$.

EXP 2.3 Experimentally Measure the FSK Power Spectral Density

Measure the power spectral density of the transmitted FSK signals for $M = 2$ and 4 after the initial handshaking sequence when random customer data is being transmitted.

• If you made the spectrum analyzer for Chapter 4, run it on one station and connect your transmitter to it.

• You can use a commercial spectrum analyzer if it is available.

• Otherwise collect an array of transmitted samples, write them to a file on the PC, and use MATLAB's Signal Processing Toolbox function pwelch( ). Compare your measured spectra with the theoretical ones you computed.

EXP 3. Making a Receiver Using an Exact Frequency Discriminator

Make a receiver using the exact frequency discriminator shown in Figure 3 for $M = 2$.

Connect the transmitter from another station to your receiver. There are RCA-to-RCA barrel connectors in the cabinet to connect RCA to mini-stereo cables together.

First leave your transmitter off and turn on your receiver. When the receiver is running, turn on your transmitter. Your receiver program should do the following:

1. The receiver should detect the absence or presence of an input FSK signal by monitoring the received signal power. The power can be calculated by doing a running average of the squared input samples over several symbols. You can also try a single pole exponential averager.

EXP 3. Making an Exact Discriminator (cont. 1)

The receiver should assume that no FSK input signal is present when this power is small and sit in a loop checking for the presence of an input signal. When the measured power crosses a threshold, the receiver can start the discriminator and symbol clock tracking algorithm.

You should predetermine a good threshold based on your knowledge of the transmitter amplitude and system gains. You can do this experimentally by observing the received power when the transmitter is running.

2. The receiver should continue to monitor the input signal power and detect when the signal is gone and then go into a loop looking for the return of a signal.

EXP 3. Making an Exact Discriminator (cont. 2)

3. Start your discriminator and symbol clock tracker once an input signal is detected. Monitor the tone frequency alternations and look for the alternation switch. Count for 10 symbols after the switch and begin detecting the tone frequencies resulting from the input customer data.

4. Send the output samples of the discriminator to the left codec channel.

   Send a signal to the right codec output channel that is a square wave at the symbol clock frequency to use for synching the oscilloscope. You can do this by sending a positive value for 20 samples at the start of a symbol followed by its negative value for the next 20 samples.

EXP 3. Making an Exact Discriminator (cont. 3)

Observe the discriminator output on the oscilloscope and take a picture of a single trace to show a typical output of the discriminator.

Alternatively, you can capture an array of discriminator output samples with Code Composer or use fprintf(·) to write the array to a PC file and plot the output file with your favorite plotting program.

If you allow the oscilloscope to run freely, you will see multiple traces synchronized with the symbol clock overlapped on the screen. This type of display is called an "eye pattern" in the communications industry. At the end of each symbol you should see two distinct equal and opposite levels and the eye is said to be open. The eye pattern can be used as a diagnostic tool. Noise and system problems cause the eye to be less open.

EXP 3. Making an Exact Discriminator (cont. 4)

5. Map the detected tone frequency sequence back into a bit sequence $\hat{d}(n)$.

6. Check that the received bit stream is the same as the transmitted one. Your receiver should have a 24-stage shift register that contains $\hat{d}(n), \hat{d}(n-1), \ldots, \hat{d}(n-23)$. You can check for errors by checking that

$$\hat{d}(n) \oplus \hat{d}(n-18) \oplus \hat{d}(n-23) = 0$$

for all n except for an initial burst of 1's when the shift register is filling up. If you initialize the state of the register to the initial state of the transmitter register the result should be all 0's if you have detected the starting time of the customer data correctly.

EXP 4. Running a Bit-Error Rate Test (BERT)

A measure of the quality of a digital transmission scheme is its bit-error rate performance in the presence of additive noise. Once your frequency discriminator receiver is working correctly with noiseless received FSK signals, perform a bit-error rate test as follows:

1. Generate zero-mean Gaussian noise samples in the DSP with some variance σ^2 and add them to the received signal samples. Implement a power meter in your receiver program to measure the power of the FSK input samples, say P. Compute the SNR = $10\log_{10}(P/\sigma^2)$ dB.
2. Your receiver should have a replica of the PN sequence generator in the transmitter. You should synchronize the state of the local PN generator to that of the transmitter so it's output sequence will be in phase with the received one.

EXP 4. BERT Test (cont. 1)

3. Start your BERT test with a very high SNR so few errors will occur. Check if each bit estimated by the receiver is the same as the transmitted one and count any bit errors for a number of bits sufficient to give a good estimate of the bit-error probability.

   The estimated bit-error rate is:

   BER = (the number of errors in the observed sequence of detected bits)/(the number of observed detected bits).

   The number of observed bits should be at least 10 times the bit-error rate to get an estimate with good accuracy. The variance of this estimator decreases inversely with the number of observed bits.

EXP 4. BERT Test (cont. 2)

4. Decrease the SNR in steps of 0.25 dB and measure the bit-error rate for each SNR. Continue decreasing the SNR until you can no longer synchronize the replica PN generator with the transmitter's generator.

5. Plot the bit-error rate *vs.* SNR. The bit-error rate should be plotted on a logarithmic scale and SNR in dB on a linear scale. This kind of plot is known as a "waterfall curve" in the communications industry.

EXP 5. Making a Receiver Using an Approximate Frequency Discriminator

Repeat the tasks for the exact frequency discriminator in EXP 3 for the approximate frequency discriminator presented starting on slide FSK-25.

EXP 6. Making a Receiver Using a Phase-Locked Loop

Make a receiver using the phase-locked loop described starting on Slide FSK-30 for $M = 2$. Test your receiver by following the steps starting on Slide FSK-76. Compare the bit-error rate performance of this receiver with the discriminator receiver.

EXP 7. Making a Receiver Using Tone Filters

EXP 7.1 $M = 4$ Tone Filter Receiver

1. Make a receiver using tone filters for $M = 4$ FSK. Implement the method for generating a symbol clock tracking signal described starting on Slide FSK-52.

2. Use the sum of the tone filter output squared envelopes, $\rho(nT)$ given by (70), to detect the presence or absence of a received FSK signal.

EXP 7.1 (cont.)

3. Test your receiver by following the steps for the exact discriminator starting on Slide FSK-76.

4. In addition, send the squared envelope at the output of one of the tone filters to one output channel of the codec, send the output of the 2nd order bandpass symbol clock tone generation filter to the other channel to use as a synch signal, observe them on the oscilloscope, and record the display. Also send the preliminary clock tracking signal to the oscilloscope and record the result.

5. Compare your measured bit-error rate curve with the theoretical bit error probability curve given by (83). Also compare the bit-error rate performance of this receiver to the others.

EXP 7.2 Simplified $M = 2$ Tone Filter Receiver

Make and test the simplified binary tone filter receiver described starting on Slide FSK-51. Measure and plot the bit-error rate *vs.* SNR and compare it with the theoretical curve.