Noncoherent Demodulators for Continuous-Phase Binary Frequency-Shift Keying Signal

Curtis Jin,* Ryan Rogalin,† Edwin Grigorian,† Matthew Klimesh,† and Hua Xie†

ABSTRACT. —Ultra-high-frequency (UHF) link for astronauts on the moon follows Space-to-Space Communications Systems (SSCS) standard using continuous-phase binary frequency-shift keying (CP-BFSK) waveform. We review the signal characteristics of CP-BFSK and cover the modulator and various types of demodulators. The optimal demodulator is derived based on the maximum likelihood criterion, which results in a bank of matched filters to find the symbol combination with the maximum correlation. An efficient implementation of the optimal demodulator is presented with low complexity and 1.9 dB gain in the bit error rate (BER) performance compared to the quadrature demodulator. The performance degradation under frequency offset has been analyzed, and the optimal demodulator shows negligible degradation for 60 parts per million (ppm) frequency error assumed for this application.

I. Introduction

We have investigated the demodulators for continuous-phase binary frequency-shift keying (CP-BFSK) waveforms, which will be used in the application of ultra-high-frequency (UHF) link for astronauts on the moon following Space-to-Space Communications Systems (SSCS) standard.[1] Advantages of imposing continuous phase on the waveform are a more compact spectrum for efficient usage of the bandwidth and an inter-symbol correlation in the waveform, which demodulators can leverage for improved bit error rate (BER) performance at a price of higher complexity [1,2,3,4].

The *coherent* receiver for CP-BFSK is the optimal design, which performs up to 3 dB better than the *noncoherent* types by maximizing the signal-to-noise ratio (SNR) [2,3]. However, coherent receivers require careful tracking of the signal phase in conjunction with the symbol detection and the symbol synchronizer, which adds complexity to the receiver structure. In our UHF astronaut application, the SNR in the link budget is large due to the proximity between users and Doppler impact is negligible due to the slow motion between

[1] NASA Johnson Space Center, Space to Space Communications System Detailed Design Document, JSC-27528, unpublished document.

* Flight Communications & Tracking Section.

† Ground Communications & Architectures Section.

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004).
© 2026 California Institute of Technology. U.S. Government sponsorship acknowledged.

users at UHF wavelength, relaxing the need for a coherent receiver. So, we focus on *noncoherent* receivers in this article, which are simpler to implement.

Nevertheless, we analyzed the degradation effect from potential frequency drifts and a possible algorithm to correct the frequency drift using short-term Fourier transform (STFT) to track the carrier. STFT applies fast Fourier transform (FFT) to overlapping windows to estimate the spectrum trend over time. The window length is chosen to match the duration of the preamble duration because the carrier is visible only during the preamble phase at the beginning of each frame in SSCS standard.

Noncoherent demodulators can be categorized by two groups: frequency discriminator type and optimal type. The frequency discriminator detects the symbols based on a single observation period. This allows the demodulator to be simple, but the demodulator's BER performance is suboptimal as it does not utilize the inter-symbol correlation introduced by the continuous-phase constraint. The optimal demodulator is based on the maximum likelihood (ML) criterion using the probability density function (PDF), which results in an optimal BER performance at the price of higher complexity. At its heart, this algorithm is searching for the most likely symbol combinations that would result in the maximum correlation with respect to the received signal. The challenge in implementing the optimal detector is the longer duration of correlation and modified Bessel function appearing in the ML detector. The longer duration of correlation inherently means needing a longer finite impulse response (FIR) filter with more multipliers, making the implementation easily infeasible. We overcome this with a recursive design that requires a correlator for *a 1-symbol duration* but still can effectively compute the correlation over multi-symbols by utilizing the CP-BFSK waveform property, and we use a block-type detector to avoid the computation of the modified Bessel function.

The next question is what the optimal observation duration would be. The performance of the optimal detector is known to plateau after a 5-symbol duration [4], and the BER degradation from frequency error also becomes stronger as the number observation increases because the received signal deviates from the expected matched signal. Taking this into account along with the complexity of implementation, the recommended choice for observation duration is 3 symbols, and the SNR of the symbol, i.e., E_s/N_0, to achieve the threshold for a BER of 1E–5 is summarized in Table 1, and the rationale for BER = 1E–5 is explained in more detail in Appendix II.D.

Table 1. E_s/N_0 thresholds for BER = 1E–5 for quadrature demodulator and 3-bit demodulator.

Frequency Error (ppm)	Frequency Error (kHz)	Quadrature Demodulator	3-Bit Demodulator
0 ppm	0 kHz	13.2 dB	11.5 dB
60 ppm	25 kHz	13.5 dB	11.6 dB

SSCS standard specifies that the system tolerates a frequency error of up to 60 ppm, which translates to a 25 kHz error with respect to the 414.2 MHz carrier frequency (for details see Appendix II). Within the possible 60 ppm frequency error for this application, the 3-bit demodulator outperforms the baseline quadrature demodulator by 1.9 dB for a BER of 1E–5 even under a frequency offset. We will show that increasing the number of

observations beyond 3 bits will start to introduce not only exponential complexity but also more sensitivity to frequency error.

In the following, we begin with presenting the signal model and the waveform characteristics in Section II.A, then the category of noncoherent demodulators and their pros and cons in Section II.B. We go into details of frequency discriminators in Section III and optimal demodulators in Section IV. The performance and recommended choice will be discussed in Section V, where the optimal noncoherent receivers perform the best even under the frequency offset for this application, 60 ppm. In Section VI, we cover a STFT-based frequency-tracking demodulator that can alleviate degradation from frequency error, and we conclude this article.

II. Problem Formulation

A. Setup

Let the encoded bits be $b_n \in \{0,1\}$, and the symbol pulses are modeled as

$$p(t) = \sum_n a_n h_T(t - nT), \tag{1}$$

where

- $a_n = -2b_n + 1$ represents the symbols that take the value between +1 and –1,
- T is the symbol period in seconds and $f_{sym} = 1/T$ is the symbol rate,
- $h_T(t)$ is a NRZ pulse, which is 1 between $0 \le t \le T$ and 0 elsewhere.

Then, a complex baseband representation of the CP-BFSK waveform is

$$s(t) = \sqrt{P} e^{j\left[2\pi f_d \int_{-\infty}^{t} p(\tau) d\tau\right]}, \tag{2}$$

where

- P is the power of the signal in watts,
- f_d is the frequency for BFSK in Hz,
  which represents the *deviation* from center frequency.

We can see from Equation (2) that the integral makes the phase continuous despite the discontinuities present in the symbol pulses $p(t)$. As a result, the bandwidth of this signal becomes narrower from the reduction of higher frequency content originating from the discontinuities in the symbol stream.

Further evaluating the integral in Equation (2), the phase term becomes

$$\theta(t) = 2\pi f_d a_n (t - nT) + 2\pi f_d \sum_{k=-\infty}^{n-1} a_k \tag{3}$$

for $nT \leq t \leq (n+1)T$. The phase is linearly increasing or decreasing at $\pm 2\pi f_d$ rad/sec depending on the sign of the symbol a_n. *The derivative of the phase* yields $2\pi f_d a_n$, a scaled symbol, which is the crux of frequency discriminator type demodulators.

The summation in Equation (3) is an accumulation of all the previous symbols, which shows that each symbol transmission will begin with a different initial phase, and it introduces memory effect in the waveform. We will exploit this property to improve the BER performance in the later sections where we develop optimal demodulators.

CP-BFSK waveforms are often parameterized by the modulation index h [1,5,6], which relates to the BFSK frequency as

$$h = 2f_d T. \tag{4}$$

We follow SSCS specification where $h = 0.7$ and the symbol rate of $1/T = 695KHz$, which yields a BFSK frequency of $f_d = 243.25\ kHz$. See Appendix II for more details on SSCS specification and link assumptions. The spectrum based on SSCS specification is plotted below.

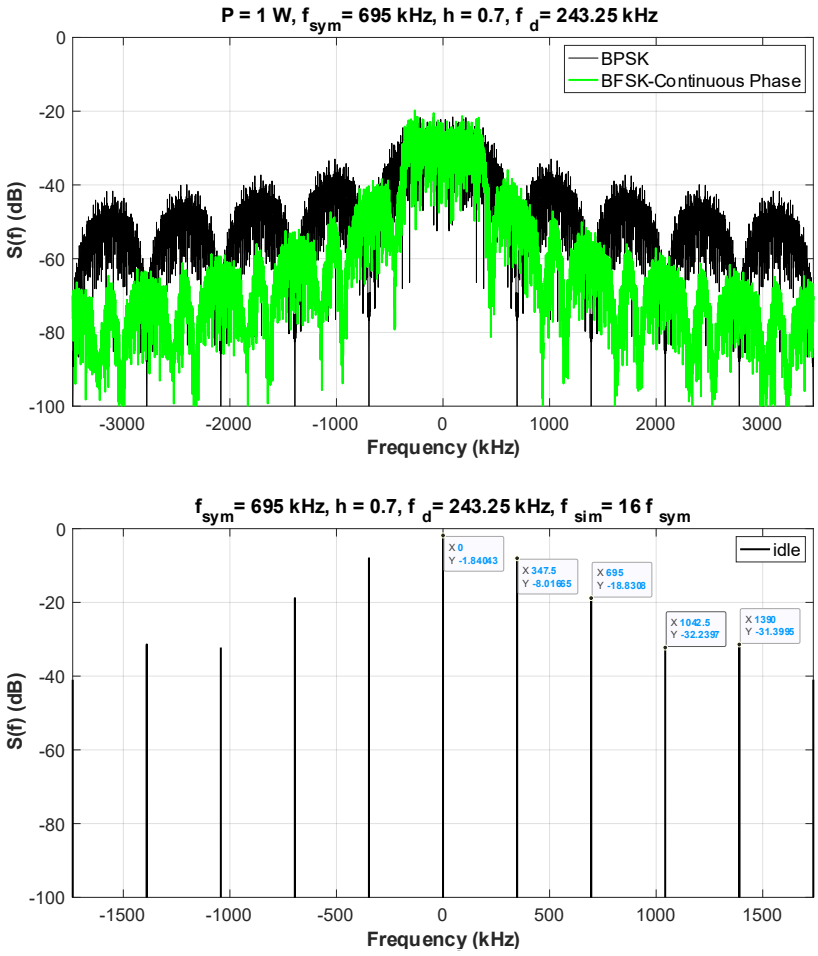


Figure 1. Spectrum modulated by random data (top) and by alternating 1 and 0 (bottom).

The top plot in Figure 1 shows that the CP-BFSK spectrum from random data is 0.7 times narrower than the binary phase-shift keying (BPSK) spectrum from identical data. Note that there is no explicit carrier when it is modulated by random data, but modulation with alternating 0 and 1 will reveal tones including the carrier in the bottom plot of Figure 1. This can be shown by a Fourier expansion over two symbols where the phase increases by $2\pi f_d$ rad/sec and then decreases by $2\pi f_d$ rad/sec. Let us define a Fourier expansion of a function $f(t)$ with period of $2T$ as

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{j\frac{2\pi n}{2T}t} \text{ and } c_n = \frac{1}{2T}\int_0^{2T} f(t)e^{-j\frac{2\pi n}{2T}t}\,dt.$$

Then, the Fourier coefficient c_n for CP-BFSK modulated by alternating symbols of 0 and 1 is

$$c_n = \frac{1}{2T}\int_0^{T} e^{j2\pi f_d t}e^{-j\frac{2\pi n}{2T}t}\,dt + \frac{1}{2T}\int_T^{2T} e^{j(4\pi f_d T - 2\pi f_d t)}e^{-j\frac{2\pi n}{2T}t}\,dt \tag{5}$$

$$= e^{j\pi(h-n)/2}\frac{2h}{\pi}\cdot\frac{\sin(\pi(h-n)/2)}{h^2-n^2}. \tag{6}$$

Evaluating this at $n = 0$ (DC), we get $|c_0|^2 = -1.8\ dB$, which reveals a nonzero carrier term. Furthermore, we can see the whole harmonics generated at multiples of half the symbol rate. This explicit carrier exists due to alternating polarity and phase continuity, where the phase does not advance over $2T$ in net. Thus, on average, the signal appears the same as a carrier-only signal, and the intermediate phase spreads the energy at multiples of half the symbol rate.

SSCS frames begin with a preamble that consists of 114 bits of alternating 1 and 0, so the carrier will be visible periodically at the beginning of each frame for 164 μsec. The STFT plot in Figure 2 shows the spectrum for five frames, and the carrier and the harmonics indeed appear at the beginning of each frame, followed by the data spectrum, which is 70% more efficient than the BPSK spectrum with rectangular pulses.

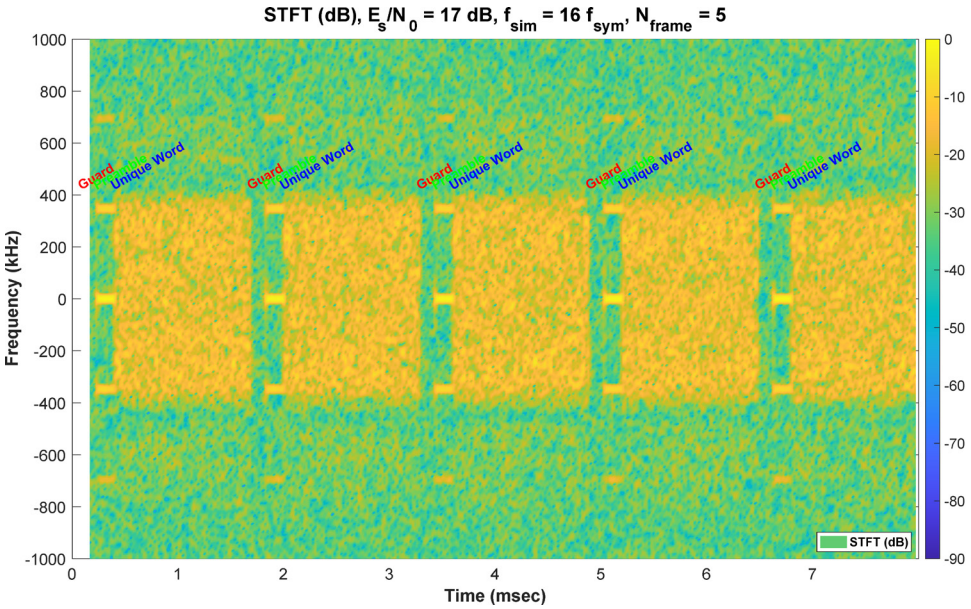


Figure 2. STFT of five SSCS frames with window length of 114 symbols with 88% overlap.

Later in this article, we will explore the option of using the explicit carrier to track frequency error at the beginning of each frame. Implementation notes for the modulator are included in Appendix I, including the design and impact when running at non-integer multiples of the symbol rate.

B. Demodulator Types

Noncoherent demodulators can be classified by two types: frequency discriminator and the optimal noncoherent demodulator. The frequency discriminator types are based on the fact that the derivative of the phase from Equation (3) yields $2\pi f_d a_n$, scaled symbols, thus transforming the demodulation problem into a binary pulse-amplitude modulation (PAM) problem. The derivative will increase the level of the high-frequency noise, so a low-pass filter (LPF) is necessary prior to the frequency discriminator to reduce noise for moderate performance. Frequency discriminators are very simple to implement, but they do not utilize the correlation introduced by the continuity in phase. In fact, the phase continuity information goes away from the derivative.

The optimal noncoherent demodulators are based on the ML criterion, where we use the PDF of the received signal to estimate the symbols. The BER performance improves at the price of complexity of implementation. See the comparison in Table 2.

Table 2. Pros and cons of various demodulators.

Type	Name	Pros	Cons
Frequency Discriminator	Differentiator	Simplest	Lowest performance
	Quadrature Demodulator	Robust performance	Higher complexity for arctangent computation
	Correlator	Optimal BER performance for 1-bit duration	Sensitivity to frequency offset
Optimal Noncoherent Demodulator	Single-symbol detection	Optimal BER performance	a) Sensitivity to frequency offset b) Higher complexity for bank of match filters, for Bessel computation, for noise density estimation
	Block-symbol detection	Optimal BER performance	a) Sensitivity to frequency offset b) Higher complexity for bank of match filters

III. Frequency Discriminator

A. Differentiator

Within a symbol period, we can simply write the received baseband signal as,

$$r(t) = \cos \omega t + n_I(t) + j\left(\sin \omega t + n_Q(t)\right) = I(t) + jQ(t), \tag{7}$$

where

- ω is the angular frequency in rad/Hz, which can be either $\pm 2\pi f_d$,
- $I(t)$ and $Q(t)$ represent the in-phase and quadrature component of the complex baseband signal, respectively,
- $n_I(t)$ and $n_Q(t)$ represent the in-phase and quadrature noise component, respectively.

Using the trigonometric identity, we can form an estimate of the angular frequency as

$$\frac{dQ(t)}{dt} \cdot I(t) - \frac{dI(t)}{dt} \cdot Q(t) \simeq \omega(\cos^2 \omega t + \sin^2 \omega t) + \tilde{n}(t) = \omega + \tilde{n}(t), \tag{8}$$

where $\tilde{n}(t)$ is the noise term, and the corresponding block diagram (BDG) is below, where the LPF is the low-pass filter with a cutoff frequency of f_{cut} Hz, AGC is the adaptive gain control for normalizing the input to the symbol tracking loop (STL) so that the STL's effective loop bandwidth (BW) remains constant.

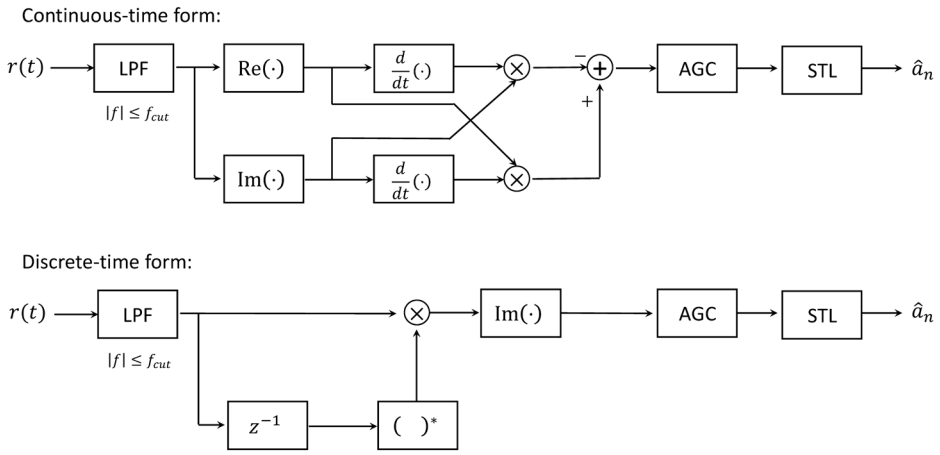


Figure 3. Differentiator: continuous-time form (top), and discrete-time form (bottom).

Replacing the differentiation operator in Equation (8) with a finite difference operator, $\frac{dQ(t)}{dt} \approx \frac{Q_n - Q_{n-1}}{T_{sim}}$ and $\frac{dI(t)}{dt} \approx \frac{I_n - I_{n-1}}{T_{sim}}$ where T_{sim} is the sampling period, we obtain the discrete-time form:

$$\begin{aligned} \frac{dQ(t)}{dt} \cdot I(t) - \frac{dI(t)}{dt} \cdot Q(t) &\simeq \frac{1}{T_{sim}} [I_{n-1} Q_n - I_n Q_{n-1}] \\ &= \frac{1}{T_{sim}} \mathrm{Im}\big[(I_n + jQ_n) \cdot \big((I_{n-1} + jQ_{n-1})\big)^*\big]. \end{aligned} \tag{9}$$

Using the fact that $I_n + jQ_n = e^{j\omega T_{sim} n}$ and $\omega = \pm 2\pi f_d$, we see

$$I_{n-1} Q_n - I_n Q_{n-1} = \mathrm{Im}[e^{j\omega T_{sim}}] = \mathrm{Im}[\cos(2\pi f_d T_{sim}) \pm j \sin(2\pi f_d T_{sim})], \tag{10}$$

which reveals an insight that *the imaginary part* of the phase-differentiated signal is the primary component in distinguishing the symbol.

B. Quadrature Demodulator

Quadrature demodulator directly differentiates the phase by taking the argument of the product between the input and a delayed-conjugated input:

$$\arg\left[e^{j\theta[n]}\left(e^{j\theta[n-1]}\right)^{*}\right] = \theta[n] - \theta[n-1] = 2\pi f_d T_{sim} a_n, \tag{11}$$

which is a scaled version of the symbol.

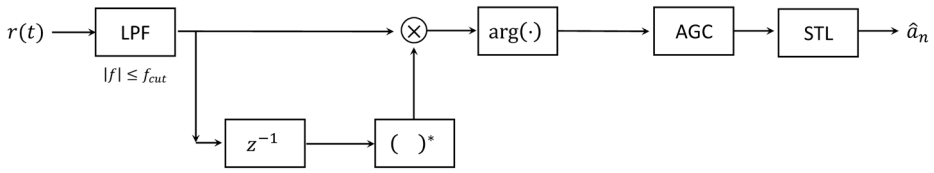


Figure 4. Quadrature demodulator.

The quadrature demodulator simply replaces the imaginary part operator with an argument operator in the differentiator in Figure 3. The argument operator can be implemented using a reverse coordinate rotation digital computer (CORDIC) algorithm in an efficient way, which does not require any multipliers [7].

The differentiation of the noise will increase the high-frequency noise content and affect the performance of symbol synchronization, and this is true for both the differentiator and quadrature demodulator. Therefore, an LPF prior to the frequency discriminator is necessary to reduce the noise impact. The BER performance versus the LPF cutoff frequency is shown in Figure 5.

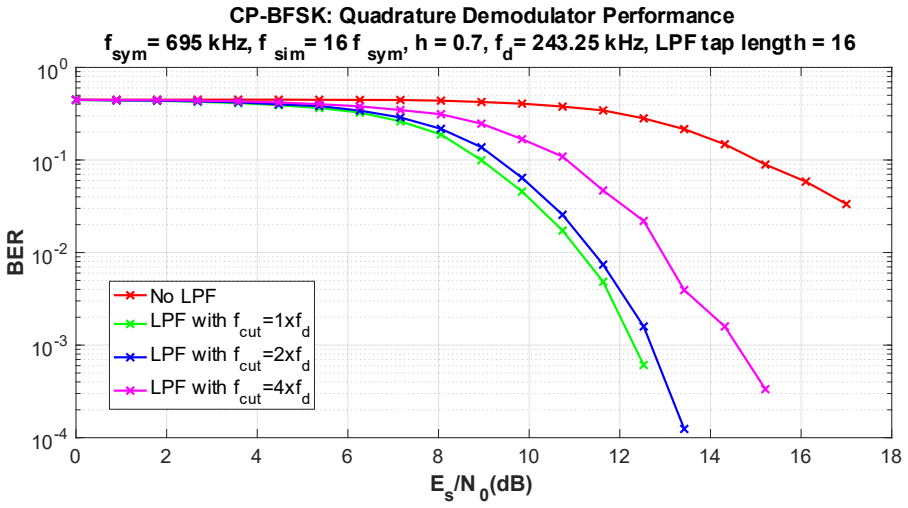


Figure 5. LPF prior to the frequency discriminator plays an important role in reducing the noise impact on the symbol detection.

C. Correlator

An optimal demodulator for an observation of 1-bit duration can be obtained by two sets of matched filters, where one is tuned at $f_1 = f_d$ (Hz) and the other at $f_2 = -f_d$ (Hz), and we form a symbol estimate by taking a difference between the two arms, as illustrated in Figure 6.

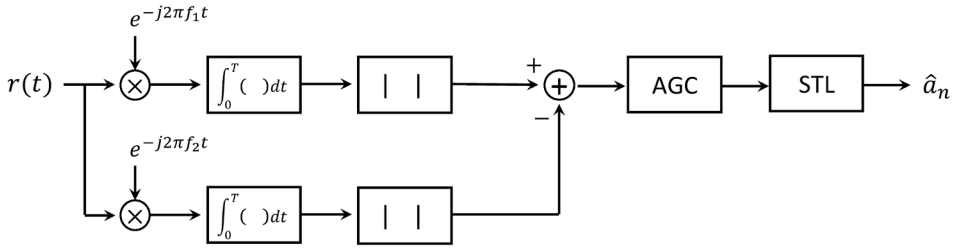


Figure 6. Correlator for 1-bit observation.

Due to the matched filter nature of this design, it is more sensitive to a frequency offset, which can push the frequency of the BFSK signal away from the matched filter frequency. However, it does not require an LPF, which was required in the former frequency discriminators. This is a special case of the optimal noncoherent demodulator, which we will explore in the next section.

IV. Optimal Noncoherent Demodulator

Given the observation duration of N symbols, we can estimate a single symbol [1,3] or a block of N symbols [1,2]. The first approach is called symbol-by-symbol detection, and it is known that estimating the middle symbol corresponds to the largest minimum-symbol-distance [1,4]. The second approach is called block detector, and the block detector is more attractive than the symbol-by-symbol detection as it does not require the estimation of the noise density and the computation of the modified Bessel function, which is required in the symbol-by-symbol detection. Here, we shall begin with symbol-by-symbol detection and extend to block detection and its implementation.

A. Symbol-by-Symbol Detector

Deriving the optimal receiver begins with the conditional PDF of the received signal. The complex baseband signal model is

$$\boldsymbol{r} = \boldsymbol{s} + \boldsymbol{v}, \tag{12}$$

where

- $\boldsymbol{s} = [s[0], s[1], s[2], \ \ldots]$ and $\boldsymbol{v} = [v[0], v[1], v[2], \ \ldots]$,

- s is the signal sampled at nT_{sim} with an initial phase of θ and modulated by symbol stream $\boldsymbol{\alpha_i} = [\alpha_1, \alpha_2, \dots, \alpha_N]$, where there can be a total of $i = 1,2, \dots, 2^N$ variations of the symbol stream.
  $s[n] \in s_i[n] = s_i(nT_{sim}; \boldsymbol{\alpha_i}, \theta) = \sqrt{P} e^{j\left(2\pi f_d \int_0^{nT_{sim}} p(\tau;\boldsymbol{\alpha_i}) d\tau + \theta\right)} = \sqrt{P} e^{j(g(nT_{sim})+\theta)}$.

- $v[n] \sim [1 \quad j] N\left(0, \sigma^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$ is an independent and identically distributed (i.i.d.) complex white Gaussian noise with zero mean and variance of $\sigma^2 = N_0 f_{sim}/2$, where N_0 is the noise density in (W/Hz) and $f_{sim} = 1/T_{sim}$ (Hz) is the sample rate, whereas $f_{sym} = 1/T$ (Hz) is the symbol rate.

- Symbol SNR (E_s/N_0) is defined as $\frac{E_s}{N_0} = \frac{P}{N_0 f_{sym}} = \frac{P}{N_0 f_{sim}} \frac{f_{sim}}{f_{sym}} = \frac{P}{N_0 f_{sim}} N_s$, where $N_s = \frac{f_{sim}}{f_{sym}}$ is the number of samples per symbol set in simulation. By using the expression for complex noise power in the simulation bandwidth $N_0 f_{sim} = 2\sigma^2$, we obtain the noise scaling factor σ for a given E_s/N_0 as $\sigma = \sqrt{\frac{P N_s}{2\, E_s/N_0}}$. This relation is used in all the simulations in this article.

For the received signal with a duration over N symbols, i.e., NT sec, the conditional PDF is proportional to

$$p(\boldsymbol{r}|\boldsymbol{\alpha_i}, \theta) \propto e^{-\frac{\sum_{k=1}^{K} (\mathrm{Re}[r[k] - s_i[k]])^2}{N_0 f_{sim}}} e^{-\frac{\sum_{k=1}^{K} (\mathrm{Im}[r[k] - s_i[k]])^2}{N_0 f_{sim}}} \tag{13}$$

$$\propto e^{\frac{2\sum_{k=1}^{K} \mathrm{Re}[r[k]]\,\mathrm{Re}[s_i[k]]}{N_0 f_{sim}}} e^{\frac{2\sum_{k=1}^{K} \mathrm{Im}[r[k]]\,\mathrm{Im}[s_i[k]]}{N_0 f_{sim}}} \tag{14}$$

$$= e^{\frac{2\sum_{k=1}^{K} \mathrm{Re}[r[k] s_i^*[k]]}{N_0 f_{sim}}} \cong e^{\frac{2}{N_0} \mathrm{Re}\left[\int_0^{NT} r(t) s_i^*(t) dt\right]}, \tag{15}$$

where we use the fact that the signal power is constant and replaced the Riemann sum with an integral by assuming $f_{sim} \to \infty$ and $KT_{sim} = NT$ for simplicity of notation.

Since the phase is unknown, averaging over the phase becomes a necessary step. In preparation for the phase average, we shall expand the conditional PDF with respect to phase θ:

$$p(\boldsymbol{r}|\boldsymbol{\alpha_i}, \theta) \propto e^{\frac{2\sqrt{P}}{N_0} \mathrm{Re}\left[\int_0^{NT} r(t) e^{-j(g(t)+\theta)} dt\right]} \tag{16}$$

$$= e^{\frac{2\sqrt{P}}{N_0}\left(\cos\theta\, \mathrm{Re}\left[\int_0^{NT} r(t) e^{-jg(t)} dt\right] + \sin\theta\, \mathrm{Im}\left[\int_0^{NT} r(t) e^{-jg(t)} dt\right]\right)} \tag{17}$$

$$= e^{\frac{2\sqrt{P}\cos(\theta+\psi)}{N_0} \sqrt{\left(\mathrm{Re}\left[\int_0^{NT} r(t) e^{-jg(t)} dt\right]\right)^2 + \left(\mathrm{Im}\left[\int_0^{NT} r(t) e^{-jg(t)} dt\right]\right)^2}} \tag{18}$$

$$= e^{\frac{2\sqrt{P}\cos(\theta+\psi)}{N_0} \left|\int_0^{NT} r(t) e^{-jg(t)} dt\right|} = e^{\frac{2\cos(\theta+\psi)}{N_0} \left|\int_0^{NT} r(t) s_i^*(t;\boldsymbol{\alpha},\theta=0) dt\right|}, \tag{19}$$

where $\tan\psi = \mathrm{Im}\left[\int_0^{NT} r(t) e^{-jg(t)} dt\right] \,/\, \mathrm{Re}\left[\int_0^{NT} r(t) e^{-jg(t)} dt\right]$. Note that the phase θ effect manifests only in the $\cos(\theta + \psi)$ term, and the integral is independent of θ and becomes a correlation with respect to the signal waveform with *an initial phase of 0 degree*.

Let us average Equation (19) with the assumptions that the phase is uniformly distributed over $-\pi$ and π, i.e., $p(\theta)\sim U[-\pi,\pi]$, and the symbol distribution is i.i.d. Then, we obtain the likelihood function of the symbol α_1 as below

$$p(\boldsymbol{r}|\alpha_1) \propto \sum_{i=1}^{2^{N-1}} p(\boldsymbol{\alpha}_i') \int_{-\pi}^{\pi} p(\boldsymbol{r}|\alpha_1, \boldsymbol{\alpha}_i', \theta)\, p(\theta) d\theta \tag{20}$$

$$\propto \sum_{i=1}^{2^{N-1}} \frac{1}{2\pi} \int_{-\pi}^{\pi} d\theta e^{\frac{2\cos(\theta+\psi)}{N_0}\left|\int_0^{NT} r(t) s_i^*(t;\alpha_1,\boldsymbol{\alpha}_i',\theta=0)dt\right|} \tag{21}$$

$$= \sum_{i=1}^{2^{N-1}} I_0\left(\frac{2}{N_0}\left|\int_0^{NT} r(t) s_i^*(t;\alpha_1,\boldsymbol{\alpha}_i',\theta=0)dt\right|\right), \tag{22}$$

where $\boldsymbol{\alpha}_i' = [\alpha_2, \ldots, \alpha_N]$, and we used the definition of the modified Bessel function for the zeroth order $I_0(x) = J_0(jx) = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{x\cos(\theta+\psi)} d\theta$. Note that the choice of the symbol to detect being α_1 can be arbitrary, and the ongoing formulation still holds as long as the symbol combinations included in the summation correspond to variations of the symbols except for the chosen symbol. It is known that choosing the symbol to detect as the middle symbol $\alpha_{\lceil N/2 \rceil}$ among $[\alpha_1, \alpha_2, \ldots, \alpha_N]$ yields the largest minimum-symbol-distance [1,4], where $\lceil N/2 \rceil$ is the ceiling of $N/2$, i.e., the least integer greater than or equal to $N/2$.

The optimal decision on whether the symbol α_1 is +1 or –1 uses the statistics from Equation (22) as

$$p(\boldsymbol{r}|\alpha_1 = +1) - p(\boldsymbol{r}|\alpha_1 = -1) \tag{23}$$

$$= \sum_{i=1}^{2^{N-1}} I_0\left(\frac{2}{N_0}\left|\int_0^{NT} r(t) s_i^*(t;\alpha_1=+1,\boldsymbol{\alpha}'_i,\theta=0)dt\right|\right) - \sum_{i=1}^{2^{N-1}} I_0\left(\frac{2}{N_0}\left|\int_0^{NT} r(t) s_i^*(t;\alpha_1=-1,\boldsymbol{\alpha}'_i,\theta=0)dt\right|\right),$$

where we declare $\alpha_1 = +1$ if $p(\boldsymbol{r}|\alpha_1 = +1) > p(\boldsymbol{r}|\alpha_1 = -1)$, and $\alpha_1 = -1$ otherwise. The corresponding block diagram is drawn in Figure 7. It essentially calculates the likelihood of $\alpha_1 = +1$ from the matched filter outputs for all possible variations of N symbols where $\alpha_1 = +1$, and calculates the likelihood of $\alpha_1 = -1$ in the same way, and subtracts each other to form an estimate of the symbol for α_1.

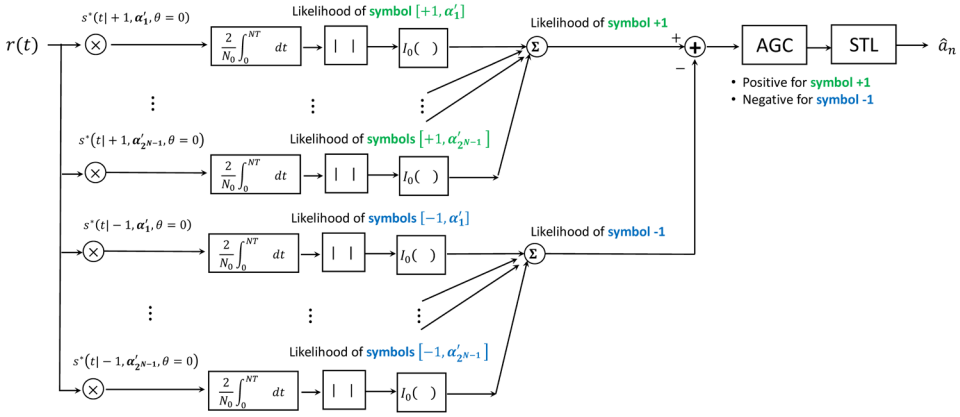


Figure 7. Noncoherent optimal demodulator for symbol-by-symbol detection.

The requirement for knowing the noise density N_0 and the computation of the modified Bessel function makes the symbol-by-symbol detector less attractive. Low SNR approximation and high SNR approximation can further simplify the design at the price of performance degradation [1,3]. In the following, we shall explore the block detector, which is both optimal and practical.

B. Block Detector

The *block detector* makes decisions for *all symbols within the observation period* $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_N] \in \{-1, +1\}^{1\times N}$, where there can be total ensembles of $i = 1, 2, \dots, 2^N$ variations of symbols. Block decision alleviates the need to average all symbols in the likelihood function, which we did in Equation (20) to Equation (22), and the decision problem simplifies to

$$\widehat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha_i}}{\operatorname{argmax}}\, I_0\left(\frac{2}{N_0}\left|\int_0^{NT} r(t)s_i^*(t;\boldsymbol{\alpha_i})dt\right|\right) = \underset{\boldsymbol{\alpha_i}}{\operatorname{argmax}}\left|\int_0^{NT} r(t)s_i^*(t;\boldsymbol{\alpha_i})dt\right|, \tag{24}$$

where we used the fact that the modified Bessel function of the zeroth order is *a monotonically increasing function.* As a result, we just need to find the symbol combination that returns the maximum correlation *without* computing the modified Bessel function. Also, it is no longer necessary to know the noise density N_0.

The corresponding block diagram is drawn in Figure 8. It is clear the decision problem becomes finding the symbol combination that returns the maximum correlation. Thus, the block detector is not only computationally more attractive but also more feasible.

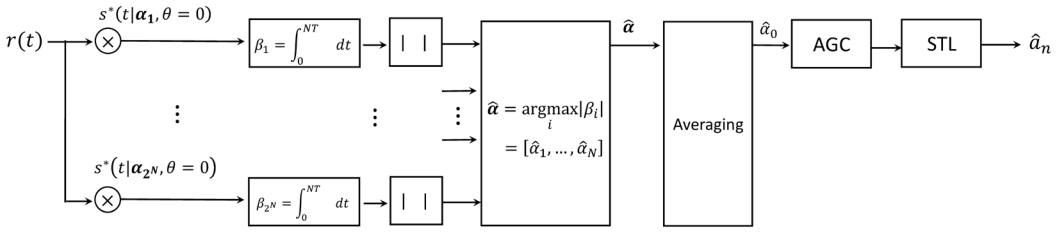


Figure 8. Noncoherent optimal demodulator for block detection.

The practical implementation requires careful consideration of the symbol synchronizer and the implementation of the 2^N matched filters, which we shall discuss next.

In general, the receiver design will have a single symbol synchronizer, or symbol tracking loop, which expects an input as a single stream of soft symbols. Either a symbol synchronizer can be integrated into the block detector to find the optimal timing for sampling, or *a conversion from block symbols to a single stream of symbols* is necessary. Here, we shall explore the latter because this decouples the symbol detection and the symbol synchronization, allowing the user to employ a symbol synchronizer of their choice.

A brute-force implementation of matched filters can be infeasible as there are total 2^N variations of matched filters, which implies a large bank of FIR filters with many multipliers. Here, we shall use a recursive form for efficient calculation of correlation [1,2],

where just two correlators with a length of a 1-symbol duration are required at the price of a filter network that combines the correlator output for all symbol variation.

Then, we will conclude this section with the implementation of max operation and the complexity of the overall block detector.

1. Block Detector: Averaging

Let us assume that the system is running at a rate of N_s samples per symbol, which is equivalent to $N_s = T/T_{sim}$, the ratio of the symbol period to the sample period. Averaging converts a block of N-symbol estimate to a single stream of symbol estimates. Suppose we are at a certain sampling instance of $t = t_n$, then there are a total of $N \cdot N_s$ estimates of the symbol for t_n because there are estimates from previous sampling times, where N is the number of symbols in the observation. The idea here is to form a symbol estimate as *an average of symbol estimates for* t_n from previous block estimates. See Figure 9 for an example of $N = 2$ and $N_s = 2$:

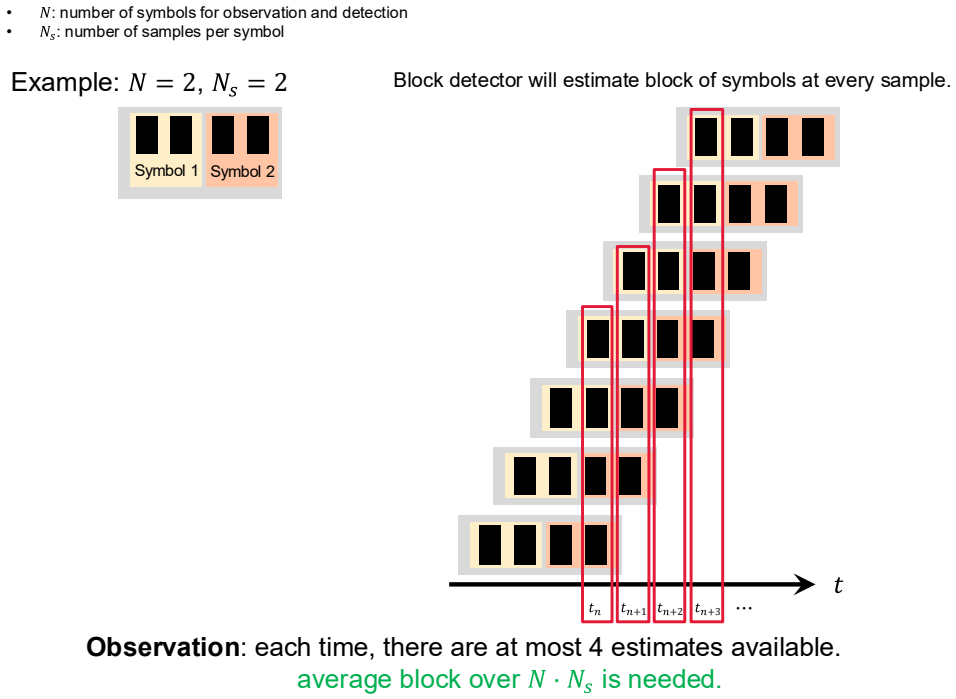


Figure 9. Rationale behind the averaging for converting a block of symbols to a single stream of symbols.

A brute-force way to implement this is to create a network of delay lines to sum the symbols from previous estimates. However, this approach is inefficient. As shown in Figure 10, we propose a daisy-chained implementation that properly pipelines the summation with the overall latency of 2 clock cycles, and it is based on the insight that the sum of the diagonal entries is the sum of the previous sub-diagonal entries plus the new input.

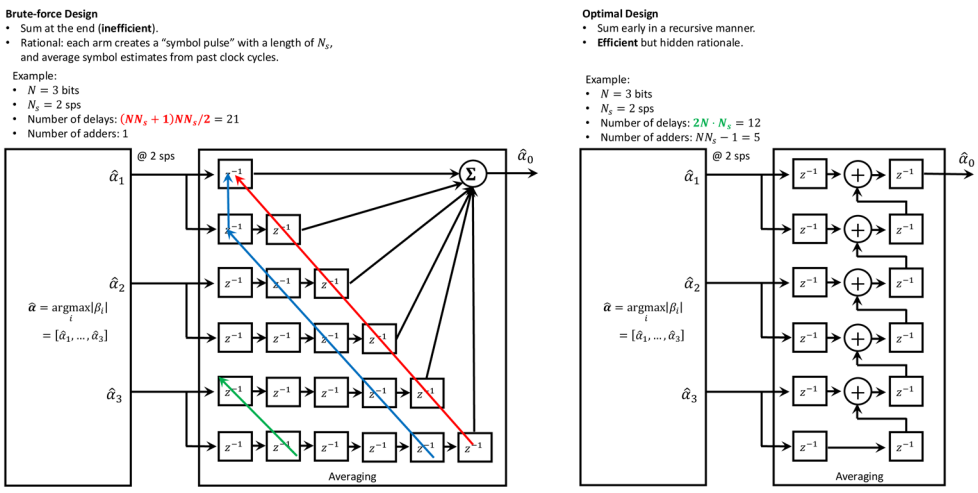


Figure 10. Optimal design for averaging. Example for $N = 3$ and $N_s = 2$.

The number of delay blocks are *reduced from quadratic to linear*, and timing is ensured within 1 clock cycle.

2. Block Detector: Recursive Correlator

Here, we explore the recursive form for implementing the 2^N variations of the matched filters. The brute-force design is 2^N numbers of matched filters of N-symbol duration. Despite the straightforward implementation, it will scale up very quickly taking up the resources of multipliers, adders, and delay blocks. The recursive design converts the brute-force design to *two* correlators of a *single*-symbol duration with a network of filters. In the end, we will compare the complexity of the brute-force design to that of the recursive design, which will reveal the efficiency of the recursive design.

Let us begin with a simple case of observation over two symbols, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2]$

$$\int_0^{2T} r(t)s^*(t;\boldsymbol{\alpha})dt = \int_0^T r(t)e^{-j2\pi f_d \alpha_1 t}dt + e^{-j2\pi f_d \alpha_1 T}\int_0^T r(t+T)e^{-j2\pi f_d \alpha_2 t}dt\,, \tag{25}$$

where we carefully included a coefficient $e^{-j2\pi f_d \alpha_1 T}$ for the second integral on the right-hand side, which represents the initial phase for the second symbol transmission and which also ensures the phase continuity.

For observation over three symbols, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \alpha_3]$

$$\int_0^{3T} r(t)s^*(t;\boldsymbol{\alpha})dt = \int_0^T r(t)e^{-j2\pi f_d \alpha_1 t}dt + e^{-j2\pi f_d \alpha_1 T}\int_0^T r(t+T)e^{-j2\pi f_d \alpha_2 t}dt$$

$$+e^{-j2\pi f_d \alpha_2 T}e^{-j2\pi f_d \alpha_1 T}\int_0^T r(t+2T)e^{-j2\pi f_d \alpha_3 t}dt\,, \tag{26}$$

which follows the same spirit of keeping the initial phase as a coefficient.

Using mathematical induction, let us generalize for $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_N]$

$$\int_0^{NT} r(t)s^*(t;\boldsymbol{\alpha})dt = \int_0^{T} r(t)e^{-j2\pi f_d \alpha_1 t}dt + e^{-j2\pi f_d \alpha_1 T}\int_0^{(N-1)T} r(t+T)s^*(t;[\alpha_2,\dots,\alpha_N])dt$$

$$= \int_0^{T} r(t)e^{-j2\pi f_d \alpha_1 t}dt + e^{-j2\pi f_d \alpha_1 T}\int_0^{T} r(t+T)e^{-j2\pi f_d \alpha_2 t}dt + \cdots$$

$$+ \prod_{n=1}^{N-1} e^{-j2\pi f_d \alpha_n T}\int_0^{T} r(t+(N-1)T)e^{-j2\pi f_d \alpha_N t}dt\,. \qquad (27)$$

The key insight from Equation (27) is that the overall matched filter output is a sum of correlations over a symbol period *with initial phase adjustments.* Also, regardless of the number of symbols for observation, only two correlators of length for one symbol are needed to calculate the overall correlation value, where one correlator is for $\alpha = +1$ and the other is for $\alpha = -1$. Figure 11 represent the Equation (25) and Equation (26) with delays to hold the previous correlator outputs and with gain blocks for initial phase adjustments.

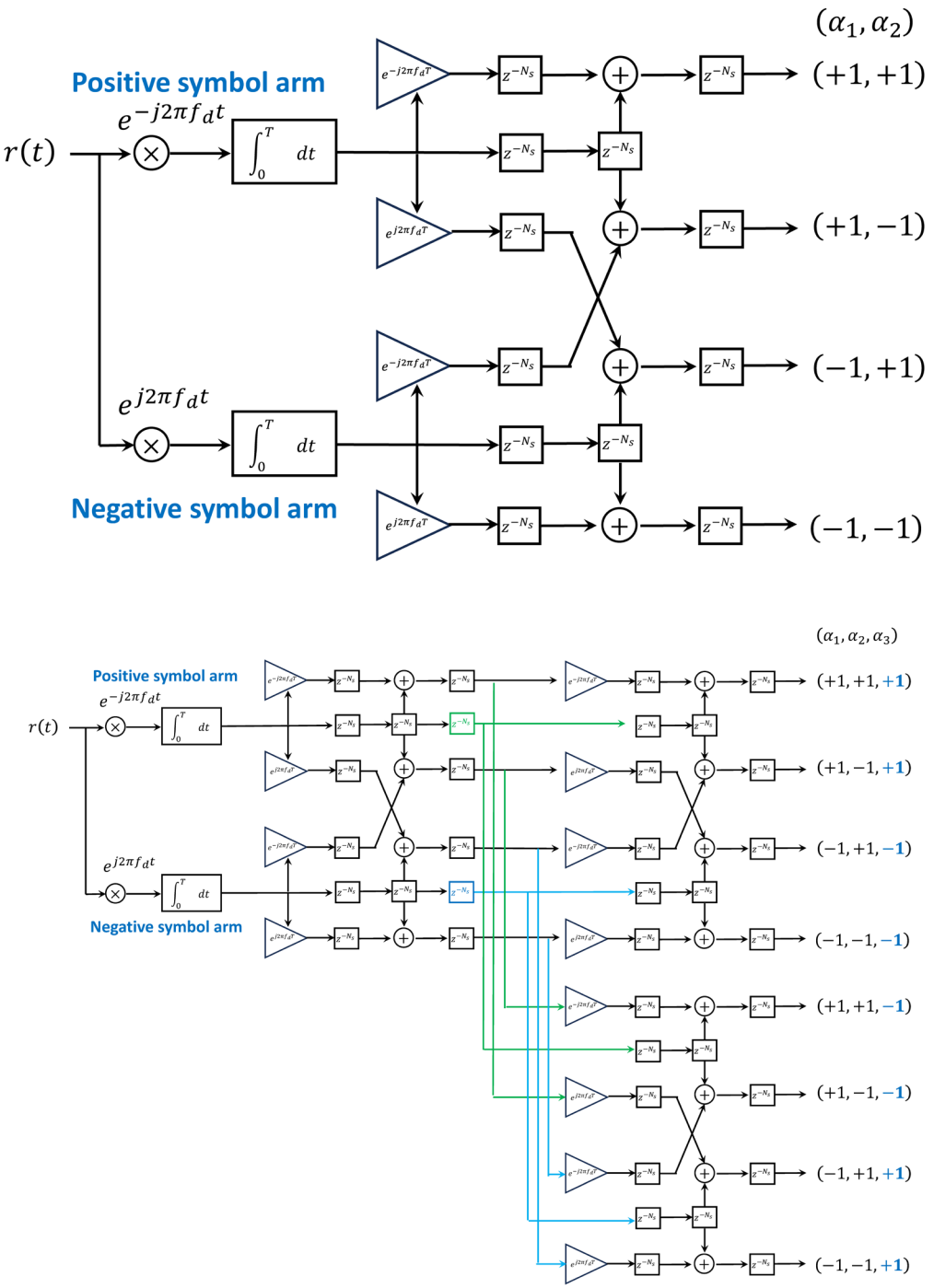
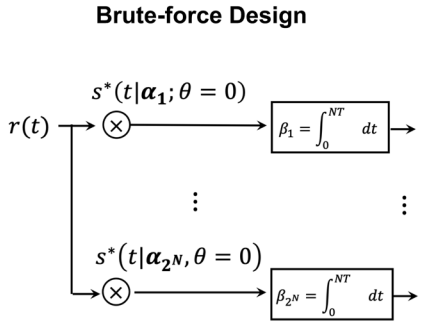
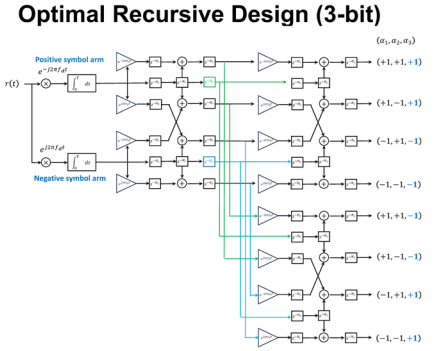


Figure 11. (Top) Recursive correlator for a 2-symbol duration. (Bottom) Recursive correlator for a 3-symbol duration. Regardless of the number of symbols for observation, only two correlators of a 1-symbol duration are needed.

We can see the *recursive nature* of the design philosophy: The 3-bit design reuses three copies of the 2-bit design with proper interconnects to represent the correct variation of symbols. Thus, the design for N-bit is simply three copies of the $(N-1)$-bit design with proper interconnects according to Equation (27).

Let us analyze the complexity of the recursive design with respect to the brute-force design. The design comparison is in Figure 12, where the resources are assumed for complex arithmetic, and N is the number of symbols for observation, N_s is the number of samples per symbol.





	Burte-force Design
# of delays	$(3NN_s)\cdot 2^N$
# of multipliers	$NN_s\cdot 2^N$
# of adders	$(NN_s-1)\cdot 2^N$
Latency (CLK)	$1+NN_s$

	Recursive Design
# of delays	$3N_s+(12N_s)\cdot 3^{(N-2)}$
# of multipliers	$N_s+4\cdot 3^{(N-2)}$
# of adders	$N_s+4\cdot 3^{(N-2)}$
Latency (CLK)	$(1+N_s)+3N_s(N-1)$

Figure 12. Correlator comparison. (Left) Brute-force design. (Right) Recursive design.

In the brute-force design, complexity grows as $O(NN_s2^N)$ because it scales as the product of the length of the correlator and the number of possible variations of the symbols 2^N. In the optimal recursive design, complexity grows as $O(N_s3^N)$ from filter network because it comprises of *two* correlators of length for a *single* symbol and filter network, which reuses three copies of previous $(N-1)$-bit design. So, the recursive design is more *scalable* as it is merely exponential whereas the brute-force design is linear times exponential, until N grows too big for the 3^{N-2} factor to become larger than the $N2^N$ factor. The improved scalability at relatively low N is evident in the example tabulated in Table 3 for $N_s=8$.

Table 3. Complexity table for $N_s=8$ samples per symbol.

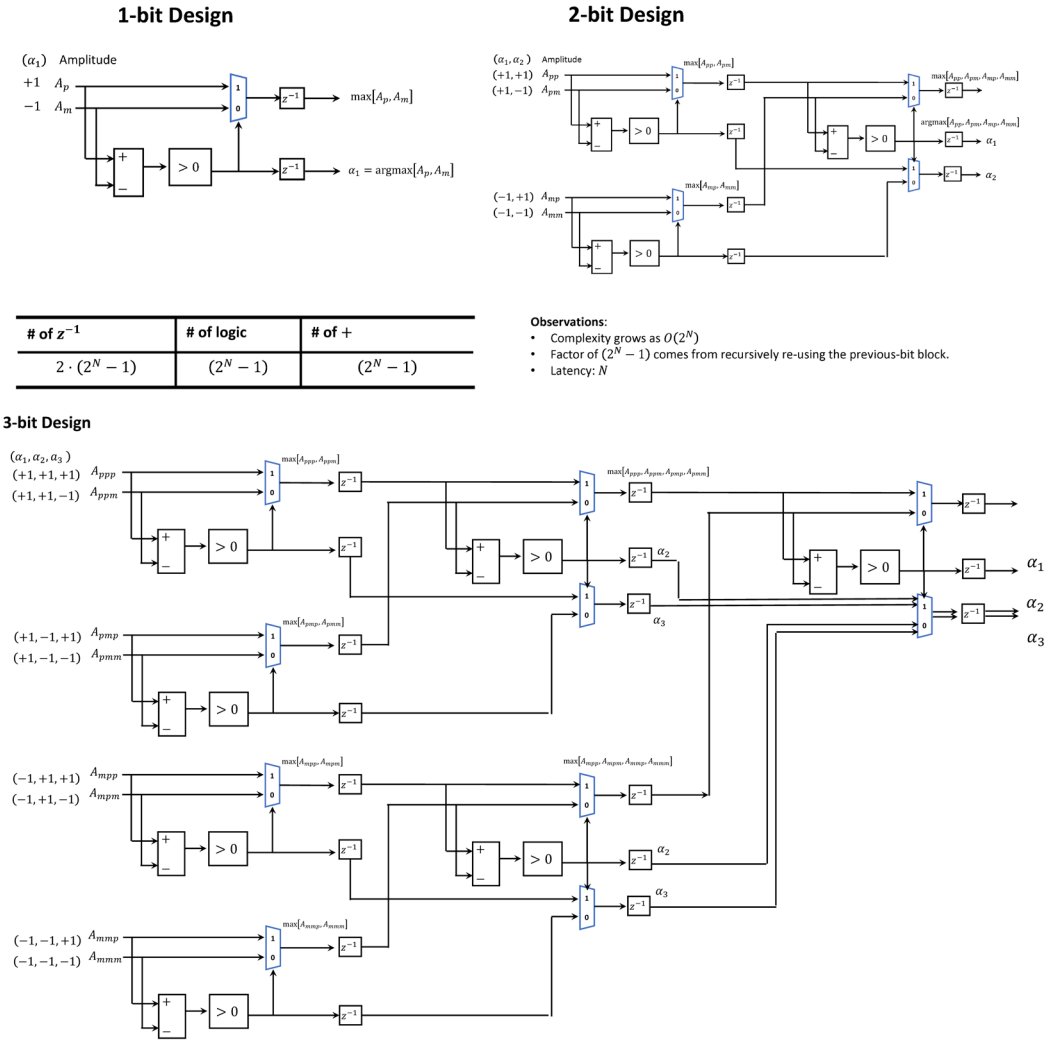
N_s=8	Brute-force Design			Recursive Design		
# of symbols (N)	2	3	5	2	3	5
# of delays	192	576	3840	120	312	2616
# of multipliers	64	192	1280	12	20	116
# of adders	60	184	1248	12	20	116
Latency (CLK)	17	25	41	33	57	105

Even though the recursive design is more efficient than the brute-force design, the number of resources will grow exponentially as the number of observations N increases. Furthermore, the BER performance improvement is known to plateau after a 5-symbol duration [4] and the performance degradation from frequency error worsens as the number of observations increases, which we shall discuss in the following sections. So, we

recommend a *3-symbol duration* for a balanced performance among resource utilization, BER performance, and robustness against frequency error.

3. Block Detector: Max

The block detector requires a max module in Equation (24), which returns the symbol combination with the maximum correlation. Figure 13 shows the block diagram for returning the max index for a 1-symbol duration, 2-symbol duration, and 3-symbol duration.



# of z^{-1}	# of logic	# of +
$2 \cdot (2^N - 1)$	$(2^N - 1)$	$(2^N - 1)$

Figure 13. (Top left) Max for a 1-symbol duration. (Top right) Max for a 2-symbol duration. (Bottom) Max for a 3-symbol duration.

The design compares the 2^N outputs from the correlators in a binary-tree fashion, while propagating the maximum index that represents the symbol combination for maximum input.

We see the recursive nature of the design where the 2-bit design reuses the 1-bit design similar to the correlator. As a result, the complexity of the max module also grows exponentially $O(2^N)$ as the number of observations N increases.

4. Block Detector: Overall

The overall block diagram for a 3-bit block detector running at 8 samples per symbol is in Figure 14, including the recursive correlators followed by max module and averaging. Figure 14 was originally conceived as a single horizontally oriented diagram, but it was separated into two parts and stacked to improve legibility.

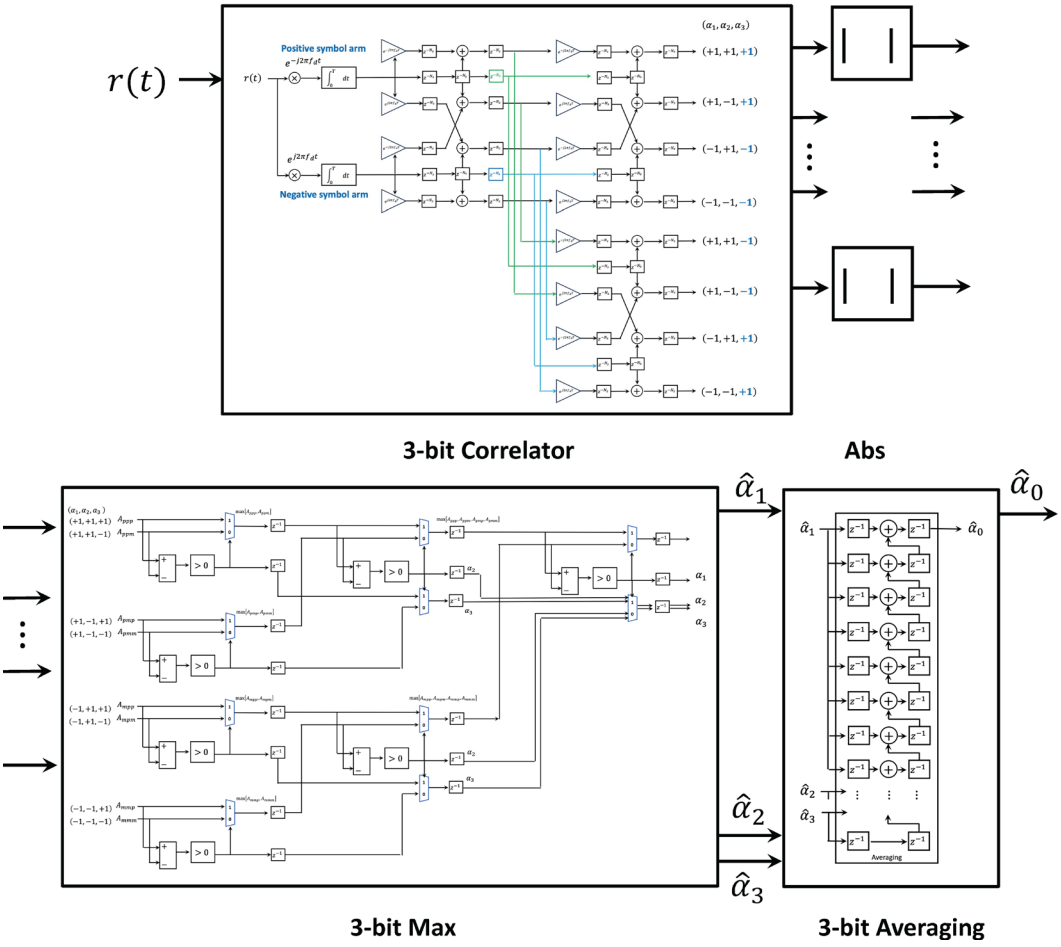


Figure 14. Overall block diagram of the 3-bit block detector for $N_s = 8$ samples per symbol.

The overall complexity is given in Table 4 below, showing a reasonable amount of resource requirement for implementation.

Table 4. Overall complexity for the 3-bit block detector for $N_s = 8$ samples per symbol.

$N_s = 8, N = 3$	Recursive Correlator	$\|x\|^2$	Max	Avg	Total
# of delays	312	8	14	48	382
# of multipliers	20	8	0	0	28
# of adders	20	0	7	23	50
# of logic	0	0	7	0	7
Latency (CLK)	47	1	3	2	53

The BER comparison between the quadrature demodulator and the 3-bit demodulator is shown in Figure 15 for a single SSCS frame. There are 1010 symbols in a single SSCS frame, and the modulated signal is fed into a quadrature demodulator and 3-bit block demodulator. The top subplot shows the in-phase and quadrature components (the Is and Qs, respectively) of the signal, and bottom subplots show the true symbols in blue, the estimated symbols in green circles after bit alignment. The red circles indicate where the erroneous bits are. The symbol SNR (E_s/N_0) is set to 5 dB, and we can clearly see that there are much fewer erroneous bits in the 3-bit block demodulator compared to the quadrature demodulator. The BER was calculated without considering the unique word in the frame, as the unique word is for frame synchronization.

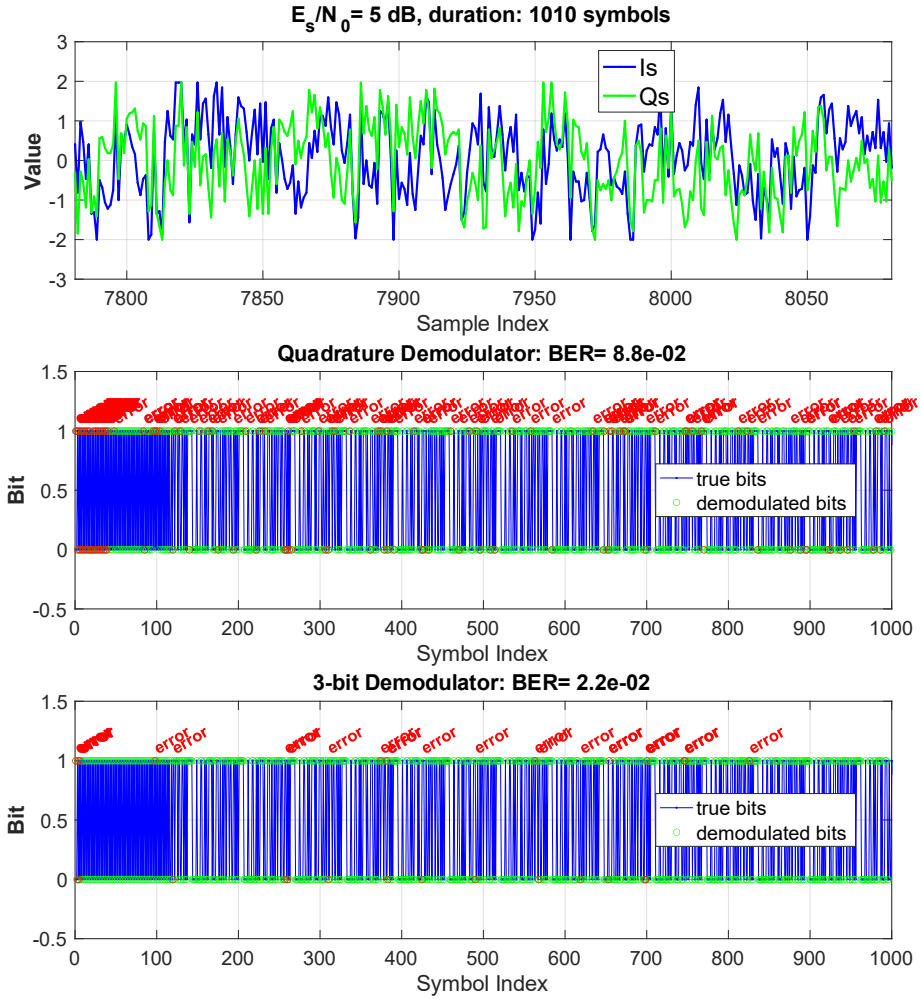


Figure 15. (Top) In-phase and quadrature components of the received signal at E_s/N_0 of 5 dB for a single SSCS frame that is 1010 symbols long. (Middle) BER of quadrature demodulator. (Bottom) BER of 3-bit block demodulator. It is evident that the block detector is outperforming the quadrature demodulator, where the quadrature demodulator shows BER of 9E−2 with 80 errors, and block demodulator shows BER of 2E−2 with 20 errors.

V. Performance

A. Performance without Frequency Error

We have run a Monte Carlo simulation of 60,000 trials where each trial contained three SSCS frames, which consist of 3030 symbols total (see Appendix II for details on frame structure). A data-transition tracking loop (DTTL) was used for symbol synchronization, and the BER was calculated after removing the unique word used for frame detection. The BER performance curves for the quadrature demodulator, block detector of 3 symbols, and 5 symbols are compared in Figure 16.

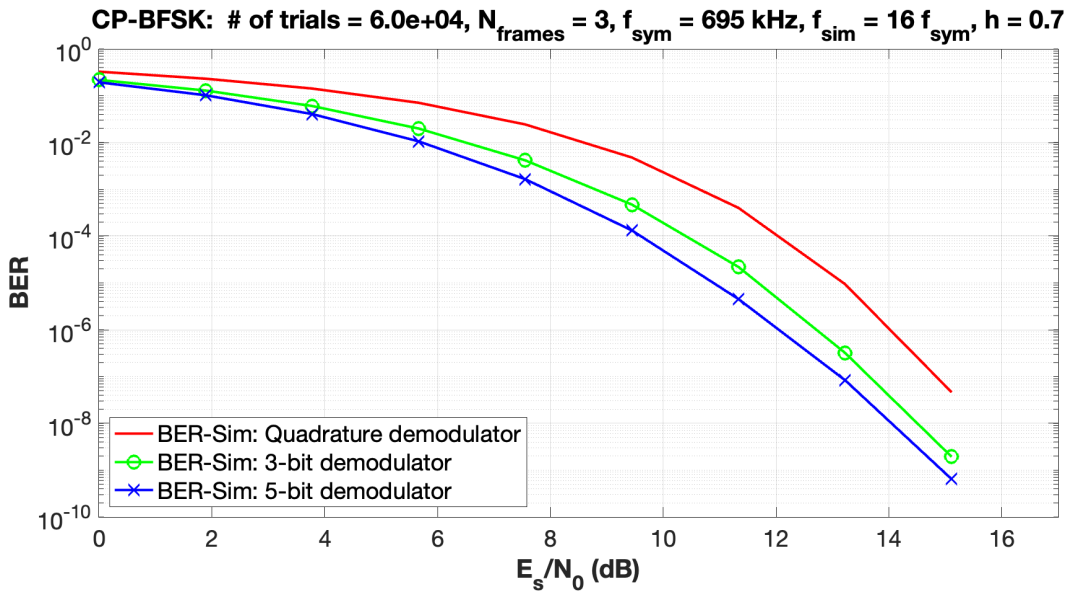


Figure 16. BER curve for quadrature demodulator in red, 3-bit detector in green, and 5-bit detector in blue.

The improved performance of the block detector is apparent, showing about 1.7 dB of performance gain compared to the quadrature demodulator, and this is within the known upper bound for the optimal noncoherent block detectors [2, 3]. We can see that the performance improvement begins to plateau from 3 to 5 bits [4], and the performance gain compared to the increased complexity makes the 3-bit detector a more attractive choice. This argument is further supported by the sensitivity of the block detector to any frequency error, which we shall see in the next section.

B. Performance with Frequency Error

The frequency discriminator, which is based on the derivatives, will show no degradation in performance from frequency error under two conditions:

1) The frequency error does not push the signal outside the input bandwidth, or LPF.

2) The estimated symbols are detrended, meaning the offset is removed prior to the symbol synchronizer.

However, the optimal demodulators, which are based on correlation, will show degraded performance due to the change in the waveform from frequency errors. Furthermore, we shall show that the sensitivity to the frequency error will increase as the observation duration increases.

The impact from frequency error on an N-bit demodulator manifests as a degradation in SNR, where we define SNR as the signal power over noise power *out of the correlator* as below

$$SNR = \frac{Signal\ Power}{Noise\ Power} = \frac{\left|\int_0^{NT} s^*(t)\, s(t)\, e^{j2\pi\tilde{f}t} dt\right|^2}{\left|\int_0^{NT} s^*(t)\, n(t)\, e^{j2\pi\tilde{f}t} dt\right|^2}, \tag{28}$$

where $e^{j2\pi\tilde{f}t}$ represents a frequency error as a fixed offset in the received signal. We clearly see that the frequency error $e^{j2\pi\tilde{f}t}$ simply rotates the complex noise and does not alter the noise power. Thus, the degradation assessment is determined by the signal power. To assess the signal power, let us assume there is no symbol timing error. The amplitude of signal correlation with frequency error would be

$$A(\tilde{f}) = \int_0^{NT} s^*(t)\, s(t)\, e^{j2\pi\tilde{f}t} dt = \int_0^{NT} e^{-j\theta(t)} \cdot e^{j(\theta(t)+2\pi\tilde{f}t)} dt = NT e^{j\pi\tilde{f}NT} \frac{\sin(\pi\tilde{f}NT)}{\pi\tilde{f}NT}, \quad (29)$$

which follows a sinc trend. Note that the amplitude converges to *the optimal amplitude* NT *as* $\tilde{f} \to 0$. Degradation factor can be defined as

$$d(\tilde{f}) = \left|\frac{A(\tilde{f})}{A(0)}\right|^2 = \left|\frac{\sin(\pi\tilde{f}NT)}{\pi\tilde{f}NT}\right|^2. \quad (30)$$

Figure 17 is a plot of the degradation factor for a frequency error from 0 kHz to 80 kHz and for a symbol rate of 695 kHz.

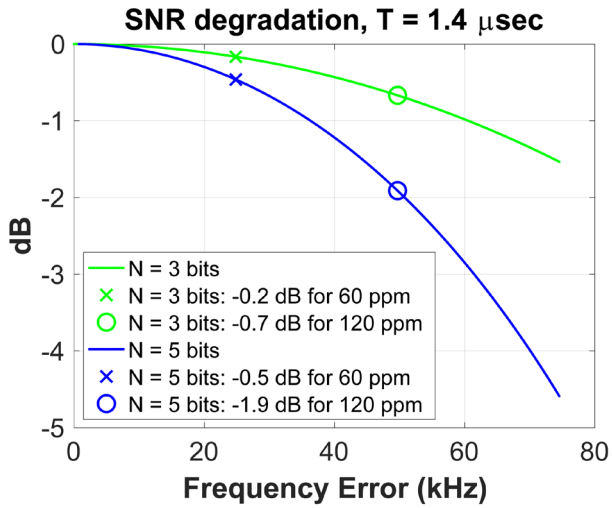


Figure 17. SNR degradation for a 3-bit demodulator in green and a 5-bit demodulator in blue.

The degradation function is the main lobe of a sinc function, and it is evident that the degradation is stronger with a 5-bit demodulator than a 3-bit demodulator for a given frequency error. This is why increasing the number of observations is not necessarily a good choice, as performance will become more sensitive to frequency drifts in addition to an increased complexity for implementation.

Per our link analysis, we have bound our worst-case frequency error to be 60 ppm with respect to the UHF carrier frequency of 414.2 MHz, which translates to 25 kHz. The 3-bit demodulator shows about –0.2 dB degradation, which is negligible compared to 1.7 dB of improved gain we saw in Figure 16. We have also placed markers for 120 ppm for reference, where we see a faster degrading trend for a 5-bit demodulator compared to that of a 3-bit demodulator.

We have run a Monte Carlo simulation of 10,000 trials where each trial contained 3 frames, which consists of 3030 symbols total. A DTTL was used for the quadrature

demodulator's symbol synchronization, and a symbol synchronizer with zero-crossing timing error detector was used for the 3-bit detector. The BER was calculated after removing the unique word used for frame detection. Figure 18 and Table 5 summarize the BER performance for the quadrature demodulator and 3-bit demodulator under a 0 kHz offset and 25 kHz offset.

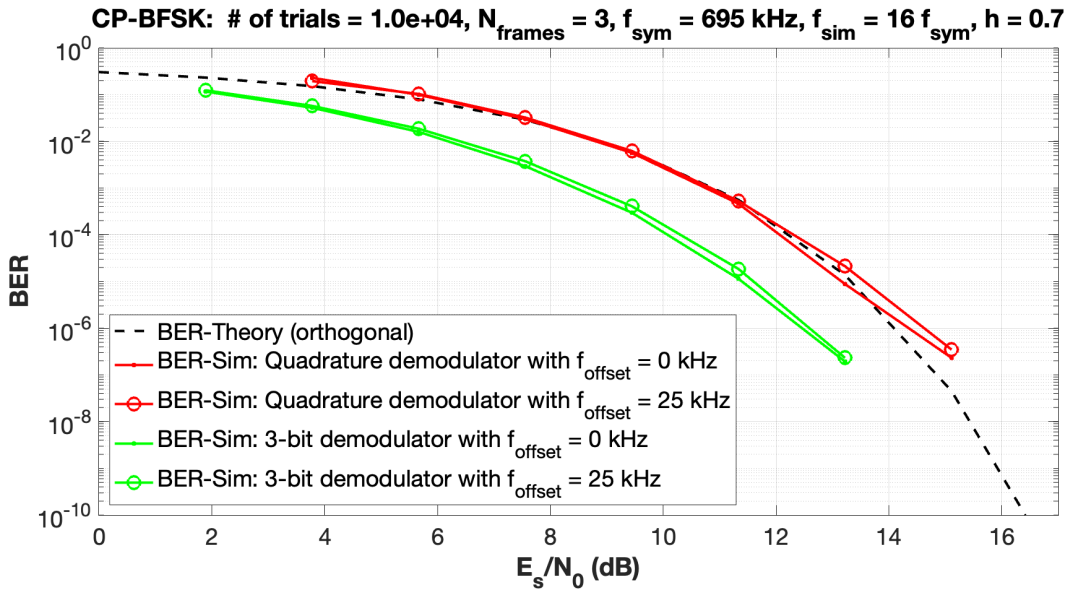


Figure 18. BER curve showing negligible degradation from a frequency offset of 25 kHz.

Table 5. E_s/N_0 thresholds for BER = 1E−5 under frequency offsets.

Frequency Error (ppm)	Frequency Error (kHz)	Quadrature demodulator	3-bit demodulator
0 ppm	0 kHz	13.2 dB	11.5 dB
60 ppm	25 kHz	13.5 dB	11.6 dB
120 ppm (for reference only)	50 kHz	14.0 dB	13.0 dB

We see that the performance degradation under a 25 kHz offset (60 ppm frequency error) is negligible for both the quadrature demodulator and the 3-bit demodulator. Under a frequency error of 0 ppm, the 3-bit demodulator performs better than the quadrature demodulator by 1.7 dB, and even under an error of 120 ppm, the 3-bit demodulator still performs better than the quadrature demodulator by 1 dB. Degradation from frequency offset is more pronounced for the 3-bit demodulator than the quadrature demodulator, –0.8 dB versus –1.5 dB, yet the 3-bit demodulator performs better within our frequency error requirement, yielding a net gain of +1.7 dB to +1.9 dB.

Thus, our recommendation based on the reduced complexity and improved BER performance under frequency offset is the 3-bit demodulator.

Note that frequency error removal was not applied to the quadrature demodulator. Although frequency error removal will keep the quadrature demodulator threshold at 13.2 dB, that is still a worse performance than that of the 3-bit demodulator under a frequency

error of 60 ppm. A possible way to remove frequency error is by using the explicit carrier revealed during the preamble phase, which we will discuss in the next section.

VI. Frequency-Tracking Demodulator

Frequency error can degrade the performance of the demodulators by introducing an offset in the symbol estimate or by changing the waveform assumed in the matched filter. An effective mitigation would be to treat the output of the frequency discriminator as a BPSK signal and use the Costas loop or spectral line method to track the frequency error [5] at the price of increased complexity. Instead, we explore the possibility of using the carrier revealed during the preamble phase of SSCS frames. The estimation of the carrier frequency can be done by STFT or a mean filter. It is important to use the frequency estimate only during the preamble phase; otherwise, the frequency estimate will be erratic. Then, we simply subtract off the frequency estimate from the symbol estimate. Note that the output of the frequency discriminator needs to be properly normalized so that its unit matches the output of the STFT.

Here we will focus on STFT-based frequency tracking for the frequency discriminator, particularly the quadrature demodulator. Figure 19 shows an overall block diagram.

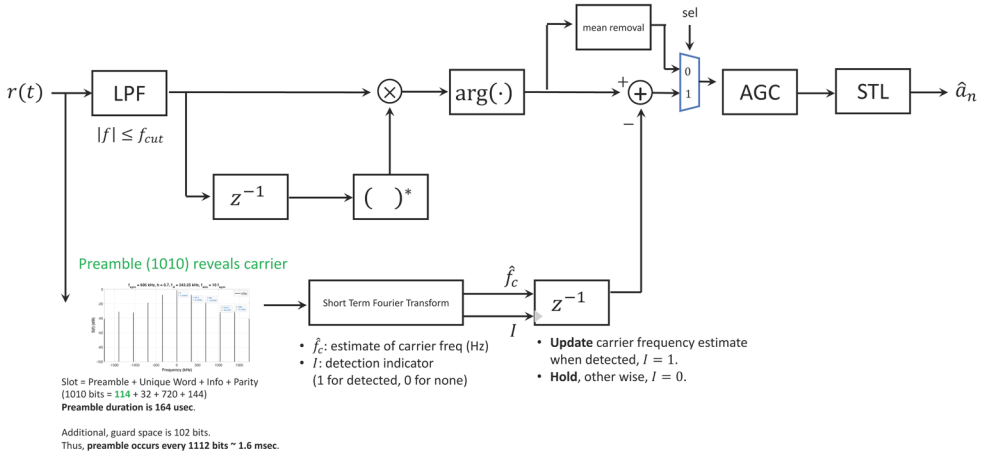


Figure 19. Frequency-tracking demodulator utilizing the carrier revealed during the preamble phase. Either a STFT or a mean-filter can be used to subtract off the frequency error.

Carrier detection using STFT is based on simple thresholding on the peak magnitude in the spectrum. Figure 20 shows a simulation of five frames under E_s/N_0 of 17 dB with a large frequency drift of 50 + 62,500 t (kHz) to illustrate the frequency drifting and tracking aspect.

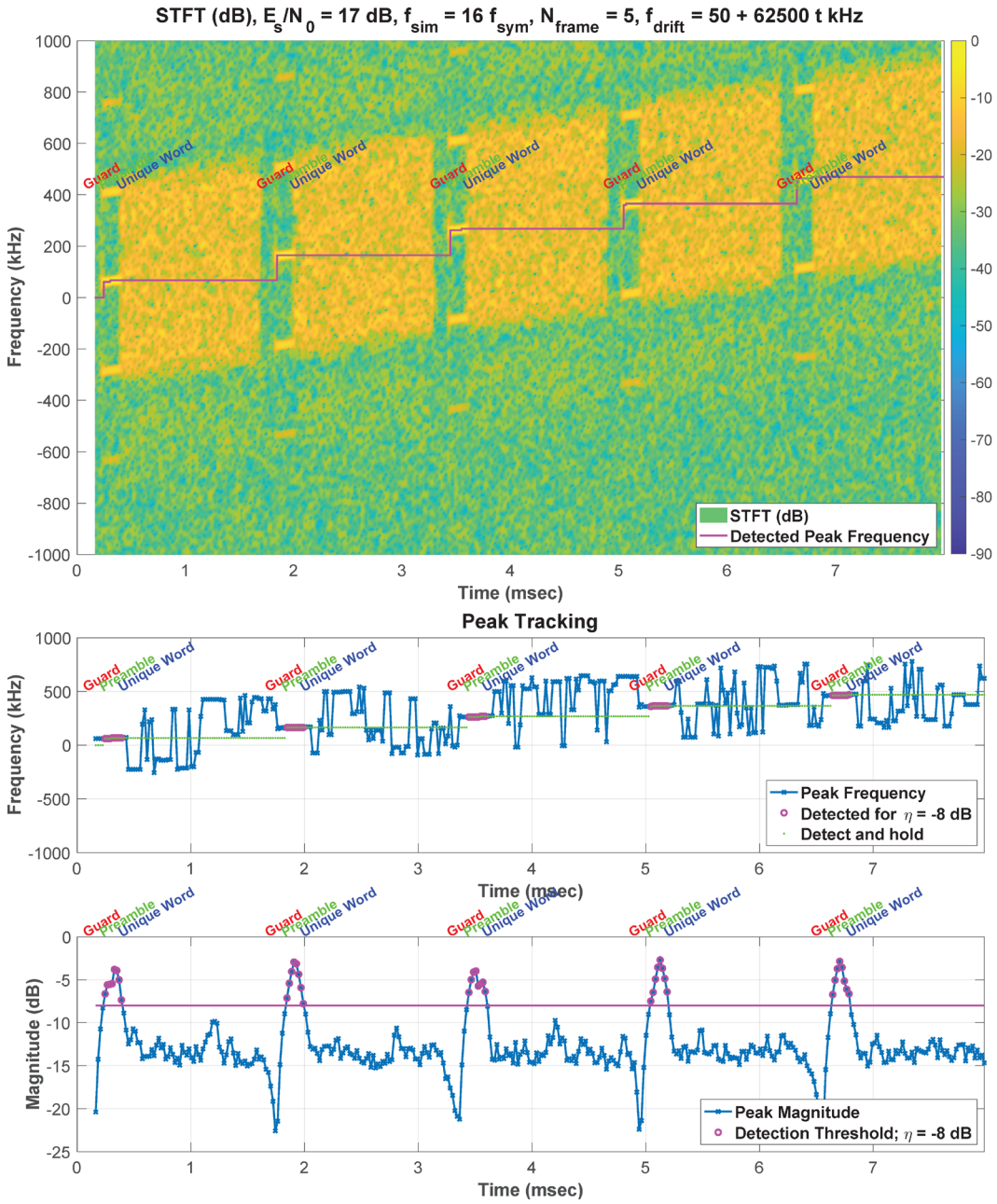


Figure 20. Simulation of five frames under E_s/N_0 of 17 dB with a large frequency drift of 50 + 62,500 t (kHz). Carrier is detected during the preamble phase with the correct frequency estimate.

The whole spectrum is drifting over time, and the carrier frequency estimate is valid during the preamble phase when the carrier is visible. As a result, peak magnitude is largest at preamble timing and tapers down when it is away from the preamble timing. Magnitude becomes the lowest during guard time when the transmitter is off.

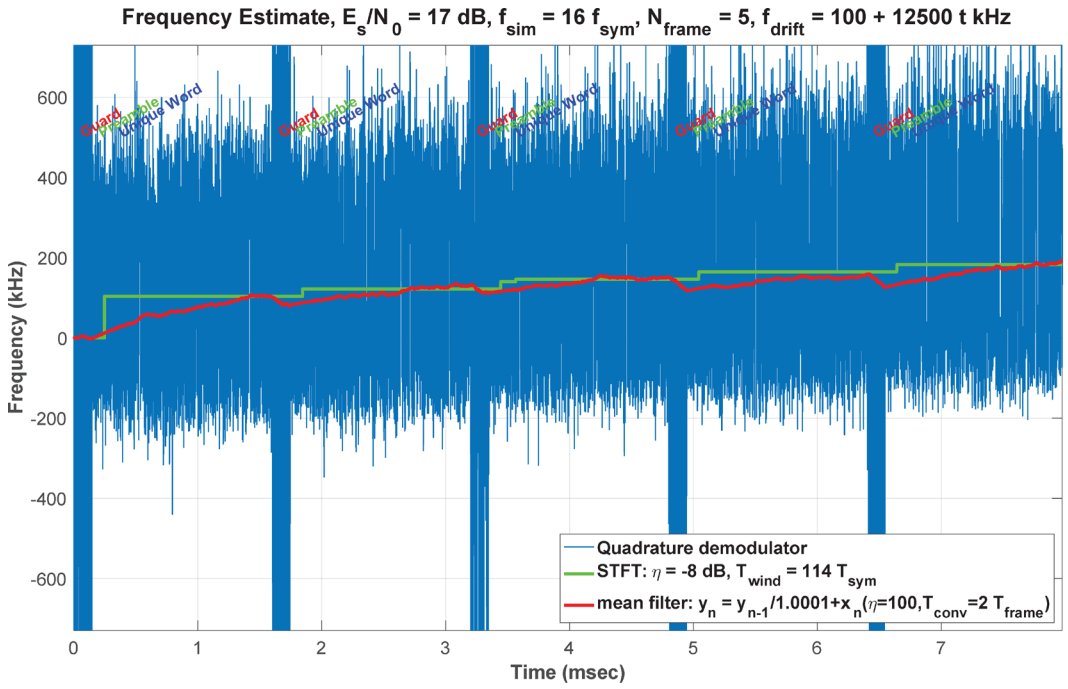


Figure 21. Quadrature demodulator output, STFT output, and mean-filter output.

The output of the quadrature demodulator with proper normalization is plotted against the STFT and mean-filter estimates in Figure 21. The STFT estimates are tracking the frequency drift in a discontinuous manner because the estimates are only available during the preamble phase. The mean-filter estimate simply tracks the average of the output of the quadrature demodulator, producing a reasonable estimate of the frequency drift. The mean-filter setting determines the convergence time and the variance of the estimate, and we have set the convergence time to be within two frames to keep the variance of the estimate low for the setup of frequency drift of 100 + 12,500 t (kHz).

Figure 22 shows the result of removing the frequency drift using STFT estimates, where the left plot (before removal) shows an increasing trend and the right plot (after removal) shows a centered signal.

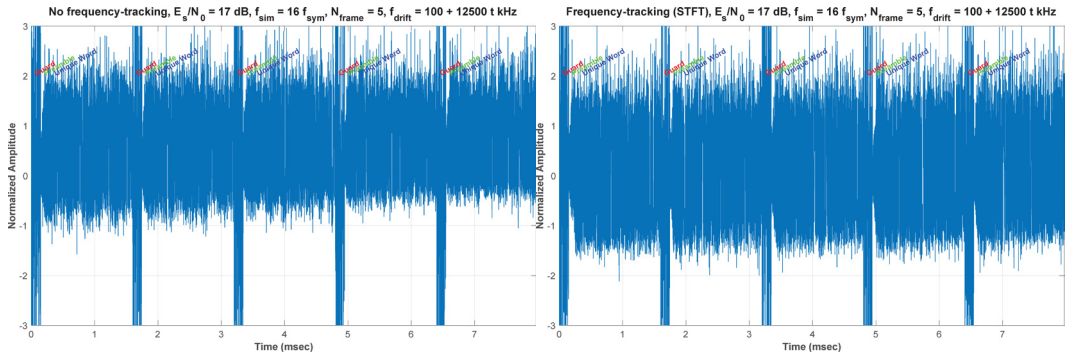


Figure 22. Before and after removal of frequency error on left and right, respectively.

We have run a Monte Carlo simulation of 10,000 trials where each trial contained 10 frames, which consists of 10100 symbols total. A DTTL was used for the quadrature

demodulator's symbol synchronization. The BER was calculated after removing the unique word used for frame detection. The BER performance for the quadrature demodulator under a frequency offset of 55.6 kHz showed about 1 dB degradation, and the STFT-based frequency-tracking quadrature demodulator shows negligible degradation in Figure 23.

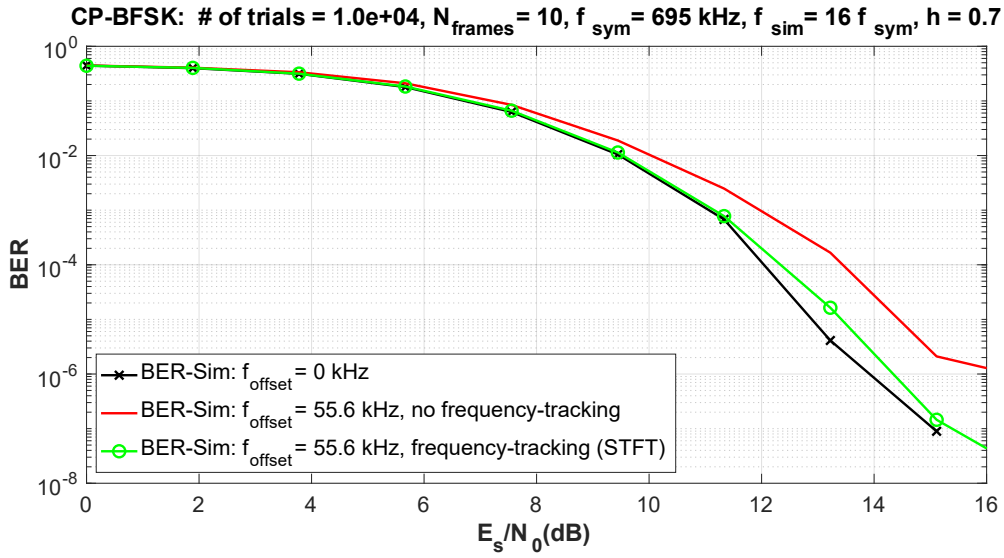


Figure 23. BER performance of the frequent-tracking demodulator via STFT showing negligible degradation from frequency offset.

VII. Conclusion

We have explored various types of demodulators for CP-BFSK waveforms. Frequency discriminators would be an ideal choice for its simplicity to implement if performance is not of a concern. For performance, we derived an optimal noncoherent demodulator using the maximum likelihood criterion. In its implementation, a block detector is more feasible as it does not require computation of the modified Bessel function and the noise density. The correlators necessary to compute the maximum likelihood are implemented using a recursive form that reduces the complexity from $O(NN_s2^N)$ to $O(N_s3^N)$. We advise setting the observation duration N to three symbols for a balanced performance in terms of complexity, BER performance, and tolerance against frequency drift. The 3-bit demodulator can yield up to 1.7 dB of gain performance within 60 ppm of frequency error. For frequency discriminators, a frequency-tracking demodulator is included in case of severe frequency drift, where the algorithm utilizes the appearance of the carrier during the preamble phase in the frame.

We leave the optimal *coherent* demodulator for future work, which not only tracks the frequency drift but also performs even better than the optimal noncoherent demodulator as all signal power rotates onto the in-phase component and sees only half of the noise on the complex plane. The development of the optimal coherent demodulator will require careful interplay between the phase-tracking component, symbol detector, and symbol synchronizer.

References

[1] M. K. Simon, S. M. Hinedi, and W. C. Lindsey, *Digital Communication Techniques: Signal Design and Detection*. Englewood Cliffs, NJ: Prentice Hall, 1995.

[2] M. K. Simon, and D. Divsalar, "Maximum-likelihood block detection of noncoherent continuous phase modulation," *IEEE Transactions on Communications*, vol. 41, no. 1, pp. 90–98, 1993.

[3] W. P. Osborne, and M.B. Luntz, "Coherent and noncoherent detection of CPFSK," *IEEE Transactions on Communications*, vol. 22, no. 8, pp. 1023–1036, August 1974.

[4] M. G. Pelchat, R. C. Davis, and M. B. Luntz, "Coherent demodulation of continuous phase binary FSK signals," *International Telemetering Conference Proceedings*, vol. 7, 1971.

[5] J. G. Proakis, and M. Salehi, *Communication Systems Engineering*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2002.

[6] S. Haykin, and M. Moher, *Communication Systems*, 5th ed. Hoboken, NJ: John Wiley & Sons, Inc., 2010.

[7] M. Rice, *Digital Communications: A Discrete-Time Approach*, 2nd ed. Seattle, WA: Kindle Direct Publishing, 2020.

APPENDICES

I. Modulator

A. Implementation

The phase term of a CP-BFSK waveform in Equation (3) shows that that phase increases linearly and its sign depends on the symbol value. Thus, the modulator is a complex exponential with a phase increment value according to the sampling period. Let us denote the rate at which the modulator is running as ℓf_{sym}, where f_{sym} is the symbol rate in Hz, and ℓ is the oversample ratio, the ratio of the sample rate to the symbol rate. Then, the phase increment per sample is

$$\Delta\theta = \frac{2\pi f_d}{\ell f_{sym}} a_m = \frac{\pi h}{\ell} a_m, \tag{31}$$

where we used the expression for modulation index $h = 2f_d T$ from Equation (4).

The modulator can be implemented with a simple switch that feeds a numerically controlled oscillator (NCO) with the phase increment $\Delta\theta$ according to the sign of the symbol a_m as below:

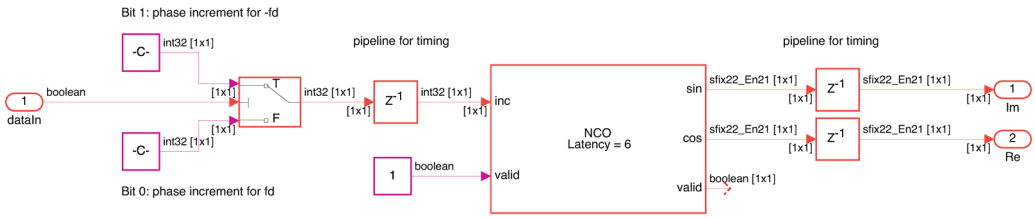


Figure 24. CP-BFSK modulator with a switch and an NCO.

B. Effect of Running Modulator at Non-integer Multiples of the Symbol Rate

Here, we analyze the effect of running the modulator at non-integer multiples of the symbol rate. The necessity of this arises when the choice of the local oscillator (LO) frequency is made for minimum spurs, and the modulator rate does not necessarily line up with the integer multiples of the symbol rate. For example, a sample rate of 200 MHz will result in an oversample ratio of

$$\ell = \frac{200\ MHz}{695\ kHz} = 287.770,$$

which is a non-integer value. Because the sample rate is non-integer multiples of the symbol rate, the data transition likely occurs *not* at the clock boundary. To understand the effect, let us imagine such a scenario. There can be two cases: The data transitions from 1 to 0, or from 0 to 1. Figure 25 describes the case of transition from 1 to 0.

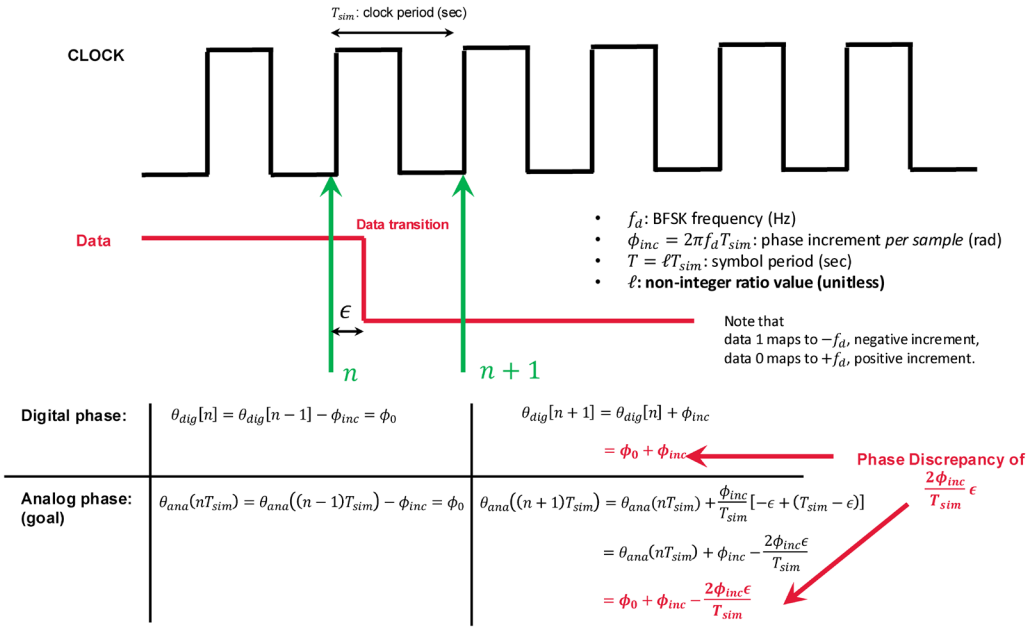


Figure 25. Description of how the phase error occurs as data transition from 1 to 0. The source of error is the digital phase not capturing the correct amount of phase change within the clock cycle. Note: bit 1 maps to symbol −1, and bit 0 maps to symbol +1.

Because the digital phase cannot capture the correct amount of phase change occurring between the clock cycles, phase error $\theta_e = +\frac{2\phi_{inc}}{T_{sim}}\epsilon$ accrues for data transition from 1 to 0, and phase error $\theta_e = -\frac{2\phi_{inc}}{T_{sim}}\epsilon$ accrues for data transition from 0 to 1, where

- T_{sim}: clock period, or sample period (sec)
- f_d: BFSK frequency (Hz)
- $\phi_{inc} = 2\pi f_d T_{sim}$: phase increment *per sample* (rad)
- $T = \ell T_{sim}$: symbol period (sec)
- ℓ: non-integer oversample ratio (unitless).

For our application, it is appropriate to assume that ϵ, the gap between symbol transition and clock boundary, is uniformly distributed between 0 and T_{sim} for the oversample ratio of $\ell = 287.770$. Figure 26 shows a numerical histogram of $\epsilon(n) = \mathrm{mod}(nT, T_{sim})$, which is close to uniform distribution, for a symbol rate of 695 kHz, sample rate of 200 MHz, and oversample ratio of $\ell = 287.770$.

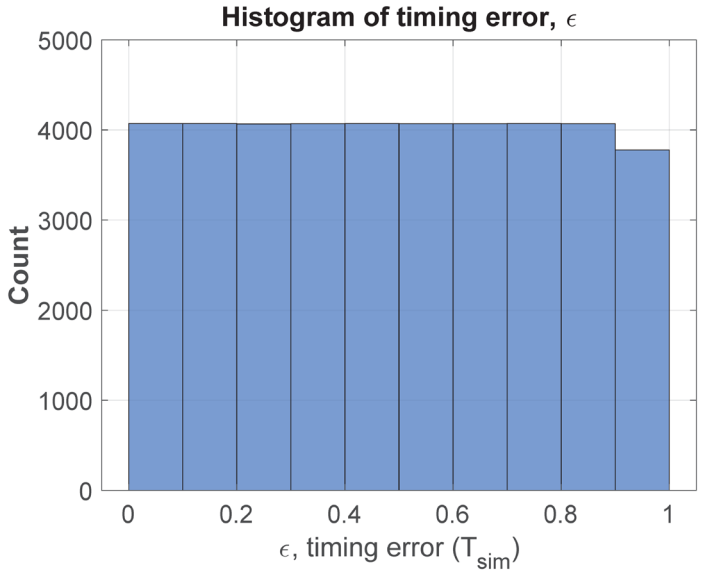


Figure 26. Distribution of the gap ϵ for oversample ratio of ℓ=287.770, showing close to uniform distribution.

Let us derive the first- and second-order statistics of the phase error. Average is

$$E(\theta_e) = \frac{1}{4}E(\theta_e|d_n = 1, d_{n+1} = 0) + \frac{1}{4}E(\theta_e|d_n = 0, d_{n+1} = 1) = \frac{2\phi_{inc}}{4T_{sim}}\frac{T_{sim}}{2} - \frac{2\phi_{inc}}{4T_{sim}}\frac{T_{sim}}{2} = 0.$$

The variance of the phase error is

$$var(\theta_e) = E(\theta_e^2) = \int_0^{T_{sim}} \theta_e^2\, p(\epsilon|d_n = 1, d_{n+1} = 0)\frac{d\epsilon}{4} + \int_0^{T_{sim}} \theta_e^2\, p(\epsilon|d_n = 0, d_{n+1} = 1)\frac{d\epsilon}{4}$$

$$= \int_0^{T_{sim}} \left(\frac{2\phi_{inc}}{T_{sim}}\epsilon\right)^2 \frac{d\epsilon}{2T_{sim}} = \frac{2}{3}\phi_{inc}^2 = \frac{8\pi^2}{3}\frac{f_d^2 T^2}{\ell^2}. \quad (32)$$

This phase error accrues every symbol period. With the i.i.d. assumption of the phase error, the error statistics would be *a random walk* with θ_e increment per symbol period. The accumulated phase error and frequency error over M symbol duration are

$$\Theta_e = \sum_{m=1}^{M} \theta_{e,m} \quad (33)$$

$$f_e = \frac{\sum_{m=1}^{M} \theta_{e,m}}{2\pi MT}, \quad (34)$$

where $\theta_{e,m}$ is an i.i.d. random variable with distribution of θ_e representing the phase error accrued at the m-th symbol. The first- and second-order statistics of the accrued error are summarized in Table 6.

Table 6. Summary of the average and standard deviation (SD) of accrual of phase error and frequency error over M symbols.

Error Type	Average	SD
Phase error	0	$\sqrt{8M/3}\pi f_d T/\ell$ (rad)
Frequency error	0	$\sqrt{2/3M}f_d/\ell$ (Hz)

Observe that choosing a clock rate with a large oversample ratio ℓ reduces the variance, making the effect negligible. Increasing the number of symbols M increases the phase error variance as $\sqrt{M}$, but reduces the frequency error variance as $1/\sqrt{M}$ because of the coefficient $1/M$ in Equation (34) for averaging. Table 7 summarizes the error for a symbol rate of 695 kHz, sample rate of 200 MHz, and oversample ratio of $\ell = 287.770$.

Table 7. Summary of phase error and frequency error for $M = 1, 3,$ and 1010. $M = 1$ represents an instantaneous error, $M = 3$ represents an accumulated error for the 3-bit demodulator, and $M = 1010$ represents an accumulated error over a single frame.

Case	Number of Symbols (M)	Phase Error SD (degree)	Frequency Error SD (Hz)
Instantaneous	1	0.36	690
	3	0.62	398
Over one frame	1010	11.36	22

For the quadrature demodulator, the instantaneous frequency error metric will determine the effect on performance, which is 690 Hz. The performance analysis in the article shows about 0.3 dB degradation from a 25 kHz offset (Table 5). Thus, a 690 Hz error will have a negligible effect on the performance of the quadrature demodulator.

For the 3-bit demodulator, the accumulated error over 3 symbols determines the effect on performance. So, both the phase error of 0.62 degree and frequency error of 398 Hz are small numbers that would result in a negligible effect.

To numerically validate the effect of running the modulator at non-integer multiples of the symbol rate, we ran simulations at *integer multiples* of the symbol rate with adding the phase error carefully as below

1. Generate a realization of the instantaneous phase error *per symbol boundary*
   a) generate a timing error distributed between $\epsilon \sim U[0, T_{sim}]$
   b) calculate the phase error: $\theta_e = \pm 4\pi f_d \epsilon$
   c) mask the symbol index that has not transitioned; –1 to –1, and +1 to +1

2. Run a cumulative sum over the instantaneous phase errors

3. Insert zeros between errors such that phase error accrues at the symbol boundary

4. Add phase error to the error-free phase *in the modulator*.

We have run a Monte Carlo simulation of 80,000 trials where each trial contained 3 frames, which consists of 3030 symbols total. A DTTL was used for symbol

synchronization. The BER was calculated after removing the unique word used for frame detection. The BER performance shows negligible degradation in Figure 27.

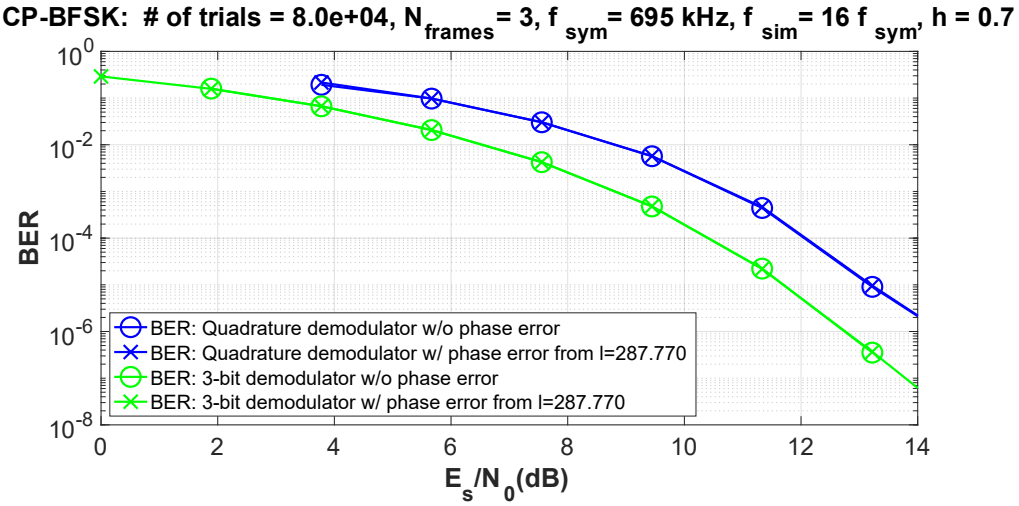


Figure 27. Negligible degradation on BER performance with phase error from running the modulator at 287.770 times the symbol rate.

II. SSCS Specification

Table 8 summarizes the key characteristics of the signal and frame, which is based on Table 3.3.1-1 from JSC-27528.[2]

Table 8. Specification table from JSC-27528.[2]

Parameters	Value
Modulation	Continuous Phase Binary-FSK: deviation of $\pm 243.25\ KHz$. carrier of 414.2 $MHz \pm 0.006\%$ (Primary UHF) carrier of 417.1 $MHz \pm 0.006\%$ (Backup UHF)
Data Rate	Burst rate: 695 kbps
Slot Length	1010 bits
Preamble	114 bits (alternating “0” and “1”)
Unique Word	32 bits (0x AAE0 7E07)
Information	720 bits
RS Parity Check	144 bits
Guard Space	102 ts

[2] NASA Johnson Space Center, Space to Space Communications System Detailed Design Document, JSC-27528, unpublished document.

A. Frame Structure

A visual representation of the frame is in Figure 28.

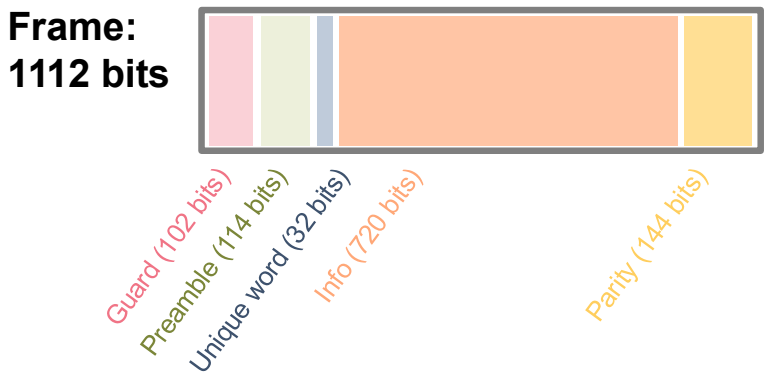


Figure 28. Frame structure.

The transmitter is off during the guard period, which corresponds to a 102-symbol duration, and the transmitter begins with alternating data between 1 and 0 during the preamble phase for a 114-symbol duration, followed by the unique word and payload data.

B. Modulation Index

The modulation index for SSCS is 0.7. This choice of $h = 0.7$ is between 0.5 and 0.715, where 0.5 is known for minimum shift keying (MSK) and 0.715 is known for the optimal choice for optimal BER performance with multi-symbol observations [2].

Table 9. Table of modulation index. SSCS choice of 0.7 is between MSK and optimum for BER performance.

IDX	Modulation Index $h = 2f_dT$	Note: Related to Frequency Spacing
01	$h = 0.5N$, or $f_d = N/(2T)$	Orthogonal set for integer N
02	$h = 0.5$	MSK
03	$h = 0.7$ (SSCS Choice)	Deviation: $\pm 243.25\ KHz$ Burst rate: 695 *kbps*
04	$h = 0.715$	Optimum BER *with observations*

C. Frequency Specification

There are two UHF frequences, the primary link and backup link. Table 10 summarizes the details, where we can see that bit 0 maps to the positive frequency and bit 1 maps to the negative frequency, which leads to mapping of bit 0 to the +1 symbol and bit 1 to the –1 symbol.

Table 10. Summary of two UHF frequencies: Primary UHF and backup UHF.

Carrier Mode	Carrier Freq. f_c	Data Input of 0 $f_0 = f_c + f_d$	Data Input of 1 $f_1 = f_c - f_d$	Freq. Difference $f_0 - f_1 = 2f_d$
Primary UHF (Lo Frequency)	414.2 MHz	414.44325 MHz	413.95675 MHz	0.4865 MHz
Backup UHF (Hi Frequency)	417.1 MHz	417.34325 MHz	416.85675 MHz	0.4865 MHz

Johnson Space Center's new requirement on frequency error is ±60 ppm, which is about 25 kHz for the primary UHF link.

Regarding the Doppler, the maximum possible motion between astronauts that we expect is 10 m/s, which translates to

$$|f_D| = \frac{v}{c} f = \frac{10\ m/s}{c} 414.2\ MHz = 13.82\ Hz,$$

where c is the speed of light. The maximum Doppler we expect is negligible compared to the UHF carrier frequency, and it will show negligible degradation in the BER performance (Table 5).

D. BER and FER Threshold

The SSCS requirement on BER[3] is to meet

uncoded BER = 1E–5 at –-110.5 dBW at the receiver input.

Assuming the noise density is $N_0 = -173$ dBm/Hz at this receiver input, the symbol SNR for a symbol rate of $R = 695\ ksps$ is $\frac{E_S}{N_0} = \frac{P}{N_0 R} = 34$ dB. This 34 dB symbol SNR bears significant conservatism considering that the required symbol SNR for BER = 1E–5 is 13.2 dB (Figure 18).

JPL missions aim for meeting a frame error rate (FER) of 1E–4 in general, and we will examine a more realistic symbol SNR threshold and ways to verify the requirement in the following.

The SSCS standard uses Reed-Solomon (RS) code (96,80) with a codeword length of 768 bits, followed by a run-length limiter (RLL) with a bandwidth expansion factor of 9/8, followed by a differential encoder, followed by a framer that adds a 114-bit preamble and a 32-bit unique word (UW), which goes into the modulator. Figure 29 below describes the overall structure, and the overall bandwidth expansion factor is $\beta = 101/64 \sim 2\ dB$, so $E_b/N_0 = E_s/N_0 + 2\ dB$. Note that we have omitted the guard period for simplicity, and it will not change our analysis result because no information is transmitted during the guard period and the symbol rate is fixed at 695 ksps.

[3] NASA Johnson Space Center, EMU Radio Detailed Design Document, JSC-28156, unpublished document.

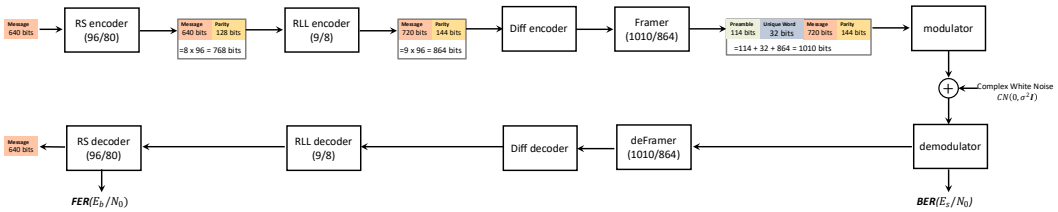


Figure 29. Overall coding and framing flow for SSCS.

We would like to emphasize that the BER of interest represents not the coded BER measured after the RS decoder but the uncoded BER measured at the demodulator output. The relationship between the FER and the uncoded BER is presented in Figure 30, which is obtained via a Monte Carlo simulation.

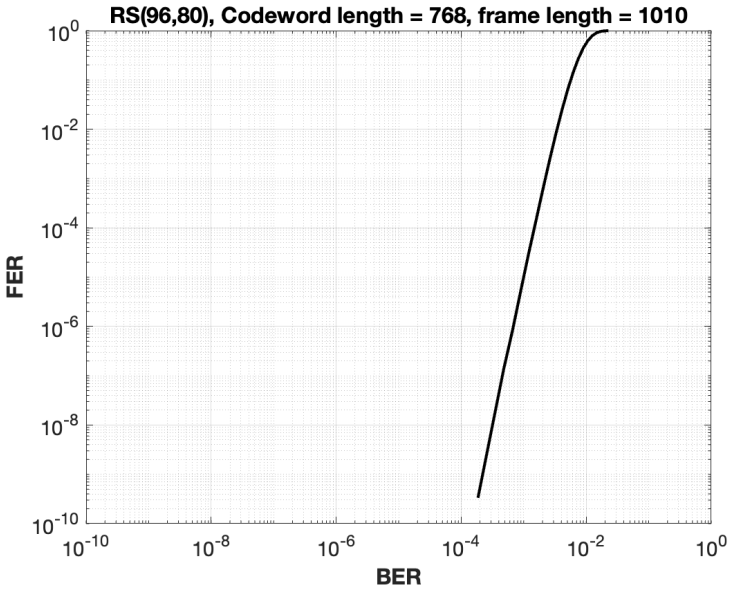


Figure 30. The relationship between the FER and the uncoded BER for SSCS coding.

We see a sharp transition from coding gain, and the FER corresponding to uncoded BER = 1E–5 will be lower than 1E–10, which would require more than 5 years just to collect 10 frame errors.

The FER calculated in Figure 30 assumes perfect knowledge of the UW location, and it can be measured by

$$FER = \frac{\#\text{ of RS decoding failure}}{\#\text{ of detected UW}}.$$

In practice, *the frame losses* can be of interest, and frame losses can occur not only when RS decoding fails but also when the UW is not detected. Thus, an additional metric of *missed detection rate of the UW* will be useful in troubleshooting. Assuming that the UW detector in the de-framer allows up to 1 erroneous bit, the missed detection rate of the UW is the probability that there is more than 1 erroneous bit in the 32-bit UW:

$$1 - [(1-p)^{32} + 32p(1-p)^{31}],$$

where p is the uncoded BER. Figure 31 shows this UW missed detection rate, FER, and BER with respect to the symbol SNR for CP-BFSK demodulated by the quadrature demodulator.

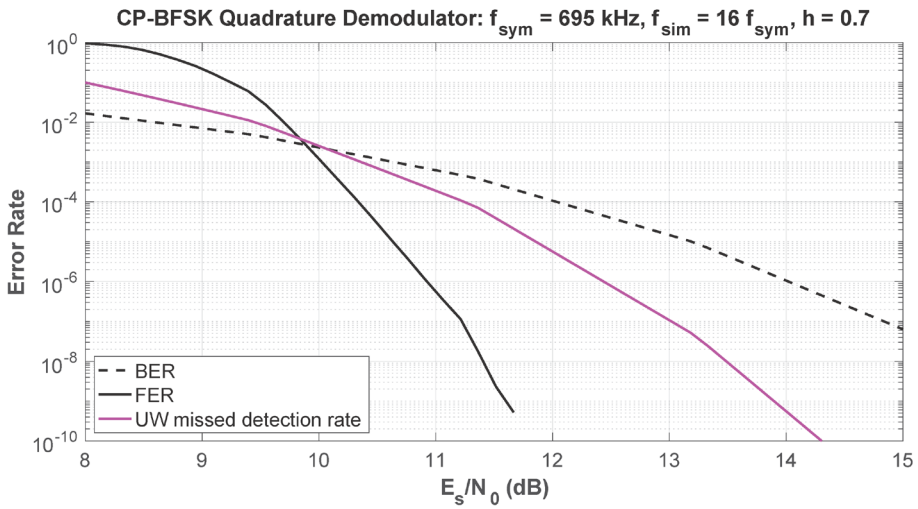


Figure 31. Uncoded BER, FER, and UW missed detection rate for CP-BFSK signal using quadrature demodulator.

We observe that when symbol SNR is higher than 10 dB, the dominant reason for frame loss is from UW missed detection, not from RS decoding failure. So, troubleshooting for missing frames at high SNR must consider the UW missed detection rate.

A summary of thresholds from Figure 31 is listed in Table 11.

Table 11. Summary of thresholds for various context.
SSCS requirement of BER = 1E–5 and JPL standard of FER = 1E–4.

BER	FER	E_s/N_0 (dB)
1E–5	<< 1E–10	13.2 dB
1E–3	1E–5	10.5 dB
2E–3	1E–4	10.3 dB

Following the JPL standard to meet FER = 1E–4, the symbol SNR threshold would be 10.3 dB, and it would take about 3 minutes to collect 10 frame errors, which is feasible.

Following SSCS requirement, we recommend that operators measure the uncoded BER = 1E–5 for practicality; otherwise, measuring FER less than 1E–10 is not feasible at all. If the operators are only limited to FER measurement, this table suggests lowering the threshold by 2.7 dB (= 13.2 dB – 10.5 dB), which puts the FER requirement to 1E–5. So, it is sufficient for the operators to set the power at the receiver input to –113.2 dBW (= –110.5 dBW – 2.7 dB) and check if the FER is below 1E–5, at which point it would take about 27 minutes to collect 10 frame errors.

Link analysis should make it explicit on the rationale behind the targeting symbol SNR. For meeting BER = 1E–5, the symbol SNR E_s/N_0 should be at least 13.2 dB for a quadrature demodulator or at least 11.5 dB for a 3-bit demodulator (Table 5).