Draft version April 22, 2024
Typeset using LaTeX **preprint2** style in AASTeX631

Radio Frequency Interference Mitigation Via Cyclostationary Signal Processing: Simulations and Performance Metrics

Ryan S. Lynch
Green Bank Observatory, PO Box 2, Green Bank, WV, 24944-0002, USA

Evan T. Smith
Green Bank Observatory, PO Box 2, Green Bank, WV, 24944-0002, USA

Matthew F. Harrison
Green Bank Observatory, PO Box 2, Green Bank, WV, 24944-0002, USA

(Received April 22, 2024)

Submitted to AJ

ABSTRACT

We describe an algorithm for identifying radio frequency interference in astronomical data by detecting cyclostationarity using the strip spectral correlation analyzer. Cyclostationarity is a property common to many sources of interference but rare in astrophysical sources. We test our algorithm using simulated interfering signals with a variety of modulation processes, symbol durations, numbers of bits-per-symbol, and signal-to-noise ratios, and compare the performance for different algorithmic parameters and thresholds for flagging a signal as interference. We also include a simulated astrophysical spectral line. Our algorithm performs reasonably well for most simulation parameters, with an average area under the resulting receiver operating characteristic curve of 0.90 and ϕ coefficient value of 0.61 when averaged over all signal properties and when using optimal algorithmic parameters. However, we find better performance for subsets of the simulated signals, especially when the signals have relatively narrow bandwidth compared to a spectrometer channel. Our approach does not perform as well for wide-bandwidth signals and frequency-switched signals with large frequency deviations. We discuss potential strategies for improving performance for these types of interferers. We believe cyclostationary signal processing is a promising approach to interference mitigation that can complement other methods.

rlynch@nrao.edu

1. INTRODUCTION

Radio frequency interference (RFI) is a ubiquitous problem in radio astronomy, analogous to light pollution at optical wavelengths. Sources of RFI are legion, including (but certainly not limited to) telecommunications, wireless Internet, navigational aides such as radar and Global Positioning System (GPS), high-speed electronics, and electrical generators and transmission lines. RFI degrades the quality of astronomical data by raising the effective noise floor, sometimes making it impossible to detect weak astrophysical sources, and in extreme cases can damage the sensitive electronics used in modern radio telescopes. Radio astronomy observatories are often been built in remote locations, taking advantage of terrain to shield telescopes, and are sometimes protected by regulatory restrictions on the types and strength of nearby transmitters. However, the growing number of satellite transmitters and mobile electronic devices, coupled with ever more sensitive astronomical instruments, make it impossible for any observatory to completely escape the effects of RFI. There is an urgent need for strategies that will allow radio astronomers to share the spectrum with other users.

Ideally, one would subtract an interfering signal, leaving behind only the astronomical signal of interest and instrumental noise, with no loss of data. In practice, it is difficult to estimate and remove the interfering signal without biasing the underlying astronomical signal. It is, therefore, more common to identify and "flag" samples contaminated by RFI so that they can be ignored at some stage of processing, at the expense of losing a (potentially large) fraction of the data. The challenge then becomes robustly detecting RFI on short timescales, so as to maximize the fraction of usable data.

A number of RFI identification techniques have been developed. Some of these assume that signals from astrophysical sources can be closely approximated as Gaussian random processes, calculate moments of the observed data, and flag non-Gaussian outliers as RFI (e.g. Nita et al. 2007; Nita & Gary 2010a; Purver et al. 2022). Others use principal component analysis to identify bases in which RFI stand out from sources of interest (Yuan et al. 2022). Machine learning offers another approach, in which algorithms are trained to recognize the same characteristics that humans use to manually identify RFI (e.g. Akeret et al. 2017; Vafaei Sadr et al. 2020; Pinchuk & Margot 2022). Each of these approaches has advantages and drawbacks. Statistical tests are straightforward and can be computationally inexpensive, but may also accidentally flag strong, impulsive astronomical sources. Principal component analysis and machine learning can use a rich, multi-dimensional representation of the data to identify RFI, but can fail when confronted with novel sources not in the training data set, though unsupervised learning methods may be able to overcome this weakness. Because RFI can take on many forms, and can have different impacts in different observing modes, it is important to explore new mitigation techniques that can complement and, in some cases, improve upon existing methods.

In this paper, we explore the use of cyclostationary signal processing (CSP) to identify RFI. A cyclostationary process is one with a statistical moment, such as mean or variance, that changes periodically or quasi-periodically (Gardner et al. 2006), as opposed to a wide-sense stationary process whose statistical moments are constant in time. Many sources of RFI are cyclostationary, with alternating current being a simple example. Cyclostationarity also arises from digital information encoding schemes in which the amplitude, frequency, and/or phase of a carrier wave switches between some finite number of possible states. Each state represents a symbol and the total number of possible states determines the number of bits that can be transmitted by each symbol. The signal will be cyclostationary at modulation frequencies related to the symbol rate, also known as the Baud rate. Since most[1] astrophysical processes are approximately wide-sense stationary, evidence of cyclostationarity could be a powerful way of distinguishing between RFI and astronomical sources.

Cyclostationarity has been discussed as an RFI mitigation technique in radio astronomy by Hellbourg et al. (2012) and Cucho-Padin et al. (2019), but has not yet been widely adopted. We have developed an algorithm for identifying and flagging RFI in astronomical data when there is significant evidence of cyclostationarity. Our long-term goal is to develop a system that can be integrated into modern radio astronomy digital spectrometers, but before doing so it is important that we determine the optimal algorithmic parameters and rigorously characterize its efficacy. As a first step in this process, we simulated a large number of human-generated signals using amplitude, phase, and frequency shift keying, and pre-processed them in a way that emulates the digital spectrometer used by the Robert C. Byrd Green Bank Telescope (GBT). Using this simulated data, we defined a "ground truth" that we then compared to the output of our algorithm. We simulated different symbol rates, numbers of bits per symbol, and signal-to-noise (S/N) ratios, in addition to the different keying techniques. This allowed us to explore the impact of different algorithmic parameters within a large parameter space. In §2 and §3 we provide some theoretical background and define our algorithm in detail. In §4 we describe our simulations, including the parameter space of the various signals and algorithmic parameters, and the metrics we use to judge performance. We present results in §5 and discuss future avenues of research in §6, before concluding in §7.

[1] Pulsars and potential extraterrestrial techno-signatures are important exceptions.

2. OVERVIEW OF CYCLOSTATIONARY SIGNAL PROCESSING

Let $x(t)$ a be a radio-frequency signal described by

$$x(t) = s(t)e^{2\pi i f_c t + i\phi} + s^*(t)e^{-(2\pi i f_c t + i\phi)} \quad (1)$$

where $s(t)$ is a signal of bandwidth B, $f_c \gg B$ is the carrier frequency, t is time, and ϕ is phase. $s(t)$ can itself be represented by in-phase and quadrature components:

$$s(t) = \frac{s_I(t) - i s_Q(t)}{2} \quad (2)$$

If $s(t)$ is periodic on a timescale T_0, then $x(t)$ will be cyclostationary, and we can extract several quantities of interest from $x(t)$. The first, known as the *non-conjugate cyclic autocorrelation function* (CAF), is a Fourier series representation of the traditional auto-correlation function given by

$$R^\alpha_{xx^*}(\tau) = \frac{1}{T_0}\int_{-T_0/2}^{T_0/2} E\left\{x\left(t+\frac{\tau}{2}\right)x^*\left(t-\frac{\tau}{2}\right)\right\}e^{-2\pi i\alpha t}dt \quad (3)$$

where E is the expectation operator, * denotes complex conjugation, t is time, τ is a time offset known as the lag, and α is the *cycle frequency* (Gardner 1991). A second quantity of interest is the *conjugate* CAF[2]

$$R^{\alpha}_{xx}(\tau) = \frac{1}{T_0}\int_{-T_0/2}^{T_0/2} E\left\{x\left(t+\frac{\tau}{2}\right)x\left(t-\frac{\tau}{2}\right)\right\}e^{-2\pi i\alpha t}dt \quad (4)$$

In a cyclostationary analog to the Wiener–Khinchin theorem, the Fourier transform of $R^{\alpha}_{xx^*}$ and R^{α}_{xx} with respect to τ yields the *non-conjugate* and *conjugate spectral correlation functions*, respectively (SCF; also known as the cyclic spectrum; Gardner 1991):

$$S^{\alpha}_{xx^*}(\nu) = \int_{-\infty}^{\infty} R^{\alpha}_{xx^*}(\tau)e^{-2\pi i\nu\tau}d\tau \quad (5)$$

$$S^{\alpha}_{xx}(\nu) = \int_{-\infty}^{\infty} R^{\alpha}_{xx}(\tau)e^{-2\pi i\nu\tau}d\tau. \quad (6)$$

We will refer to ν as the *spectral frequency* to differentiate it from the cycle frequency. The non-conjugate CAF and SCF will be non-zero only for $\alpha_n = n/T_0$, while the conjugate CAF and SCF will be non-zero only for $\alpha_n = n/T_0 \pm 2f_{\rm c}$, where $n = 0, 1, 2, \ldots$ is an integer. Note that when $\alpha = 0$, the non-conjugate SCF reduces to the usual definition of the power spectral density (PSD).

The non-conjugate and conjugate *spectral coherence* are normalized versions of the non-conjugate and conjugate SCF, defined as

$$\rho^{\alpha}_{xx^*}(\nu) = \frac{S^{\alpha}_{xx*}(\nu)}{\sqrt{S^0_{xx*}(\nu+\alpha/2)S^0_{xx*}(\nu-\alpha/2)}} \quad (7)$$

$$\rho^{\alpha}_{xx}(\nu) = \frac{S^{\alpha}_{xx}(\nu)}{\sqrt{S^0_{xx*}(\nu+\alpha/2)S^0_{xx*}(\nu-\alpha/2)}} \quad (8)$$

where $S^0_{xx*}(\nu \pm \alpha/2)$ is a frequency-shifted version of the PSD. Note that when $\alpha = 0$ the non-conjugate spectral coherence function is unity for all values of ν, regardless of the properties of the input signal.

Our algorithm exploits the fact that the SCF[3] of a stationary process only has significant power when $\alpha = 0$, whereas the SCF of a cyclostationary process also has significant power at higher cycle frequencies. Since the magnitude of the spectral coherence function is ≤ 1, it is especially useful for setting detection thresholds for data with arbitrary mean and variance.

3. AN ALGORITHM FOR DETECTING RFI USING CYCLOSTATIONARY SIGNAL PROCESSING

In general, the data collected by a radio telescope may contain a large number of cyclostationary sources of RFI whose properties (e.g. carrier frequency, modulation frequency, encoding scheme, etc.) will not be known *a priori*. To blindly find evidence of cyclostationarity we need to have some way of efficiently estimating the SCF for a large number of discrete α. We make use of the *strip spectral correlation analyzer* (SSCA; Roberts et al. 1991), which works by time-averaging frequency-domain correlations (see Equations 9 and 10). Given a signal discretely sampled at a rate $f_{\rm s}$ with N total points, the SSCA estimates the SCF at N discrete values of α. The number of spectral frequencies, M, is controlled via a first-stage channelizer. In words, the steps in the SSCA are

1. Take a data set, denoted as $x[n]$, of length N points and duration T.

[2] The nomenclature here can be confusing, since the non-conjugate CAF is calculated using the traditional definition of the autocorrelation function in which $x(t)$ is multiplied by a lagged version of its complex conjugate, while the conjugate CAF is calculated without using the conjugate of $x(t)$. We use this nomenclature to be consistent with other CSP literature.

[3] In the remainder of this paper we will use SCF as an abbreviation for the non-conjugate and conjugate spectral correlation and coherence functions in contexts where these are interchangeable.

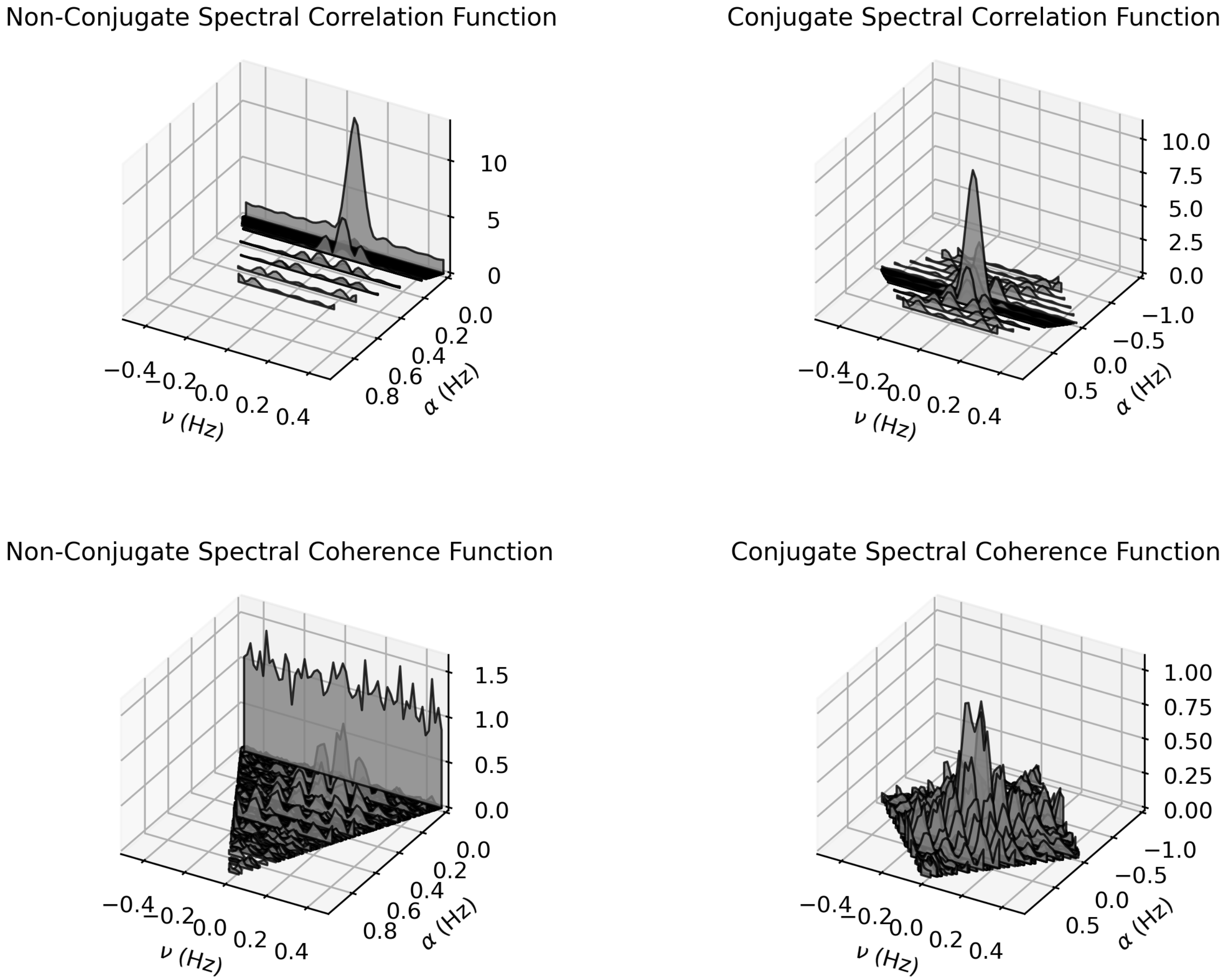

Figure 1. An example visualization of various forms of the SCF for a rectangular-pulse binary phase-shift keyed signal with a Baud rate of 0.1 Hz and carrier frequency of 0.05 Hz. The signal was 32,768 samples long and the SCFs were generated via our implementation of the strip spectral correlation analyzer using $M = 64$ (see text for details). For clarity, we have only plotted α corresponding to the top 200 values of the SCFs.

2. Use a windowing function and sliding Fourier transform to channelize subsets of $x[n]$, each of length M, yielding $X[\nu_k, r]$, where ν_k are the channelizer frequencies (*not* the final spectral frequencies that appear in Equations 5 and 6) and r is the time index.

3. Multiply $X[\nu_k, r]$ by $x^*[r]$ (for the non-conjugate SCF) or $x[r]$ (for the conjugate SCF).

4. Take a discrete Fourier transform of the result of step 3 along the time axis.

5. If desired, compute the spectral coherence using an over-sampled estimate of the PSD.

Table 1. Generalized Extreme Value Distribution Shape Parameters (ξ,μ,σ) for Various SSCA Parameters

Method	N \ M	32	64	128	256	512	1024
Non-Conjugate	1024	(0.06,0.35,0.02)	···	···	···	···	···
	2048	(0.06,0.27,0.01)	(0.10,0.36,0.02)	···	···	···	···
	4096	(0.06,0.20,0.01)	(0.03,0.28,0.01)	(0.03,0.38,0.02)	···	···	···
	8192	(0.03,0.15,0.01)	(0.06,0.21,0.01)	(0.05,0.29,0.01)	(0.05,0.40,0.01)	···	···
	16384	(0.02,0.11,0.01)	(0.01,0.16,0.01)	(0.02,0.22,0.01)	(0.03,0.30,0.01)	(0.05,0.41,0.01)	···
	32768	(0.04,0.08,0.00)	(0.03,0.11,0.00)	(0.01,0.16,0.01)	(0.00,0.23,0.01)	(0.00,0.32,0.01)	(0.05,0.43,0.01)
Conjugate	1024	(0.06,0.42,0.03)	···	···	···	···	···
	2048	(0.00,0.32,0.02)	(0.06,0.43,0.02)	···	···	···	···
	4096	(0.02,0.25,0.01)	(0.06,0.33,0.02)	(0.03,0.44,0.02)	···	···	···
	8192	(0.00,0.18,0.01)	(0.00,0.25,0.01)	(0.04,0.34,0.02)	(0.06,0.45,0.02)	···	···
	16384	(0.06,0.14,0.01)	(0.00,0.19,0.01)	(0.00,0.26,0.01)	(0.00,0.35,0.02)	(0.04,0.46,0.02)	···
	32768	(0.00,0.10,0.01)	(0.02,0.14,0.01)	(0.02,0.19,0.01)	(0.00,0.27,0.01)	(0.05,0.36,0.02)	(0.03,0.48,0.02)

Mathematically, the SSCA for the non-conjugate SCF can be written as

$$\hat{S}_{xx^*}^{\nu_k+q\Delta\alpha}\left[n, \frac{\nu_k}{2} - \frac{q\Delta\alpha}{2}\right]_T = \sum_r X[\nu_k, r]\, x^*[r] w[n-r] e^{-2\pi i q r/N} \quad (9)$$

and for the conjugate SCF

$$\hat{S}_{xx}^{\nu_k+q\Delta\alpha}\left[n, \frac{\nu_k}{2} - \frac{q\Delta\alpha}{2}\right]_T = \sum_r X[\nu_k, r]\, x[r] w[n-r] e^{-2\pi i q r/N} \quad (10)$$

where the T subscript indicates time averaging, $\Delta\alpha = T^{-1}$ is the cycle frequency resolution, q is an integer index running from $-N/2$ to $N/2$, and w is a windowing function. The cycle and spectral frequencies are

$$\begin{aligned} \nu &= \frac{\nu_k}{2} - q\frac{\Delta\alpha}{2} \\ \alpha &= \nu_k + q\Delta\alpha \ . \end{aligned} \quad (11)$$

For the SSCA to provide an accurate estimate of the SCF it must satisfy the condition $N/M \gg 1$. In this work, we only considered cases where $N/M \geq 8$.

We implemented the SSCA in Python, closely following the approach described by Carter (1992). We use a Hann window and short-time Fourier transform implemented as part of the cuSignal package (Thompson & Nicely 2021) for the first-stage channelization, overlapping each window by $M-4$ samples. We use the CuPy (Okuta et al. 2017) fast Fourier transform routines for the second-stage transform. One critical and difficult aspect of using the SSCA to estimate the spectral coherence is estimating the PSD at the appropriate frequencies, particularly when the PSD estimate takes on small values, in which case small errors in the denominator of Eqs. 7 and 8 can lead to numerical artifacts. We use a time-averaged estimate of the PSD calculated from the input to the SSCA. Specifically, we use the SciPy implementation of Welch's method with 32 time domain segments (i.e. each segment has a length of $N/32$ points), 50% overlap between segments, and a Hann window. Empirically, this leads to a robust estimate of the spectral coherence (see Fig. 1).

To use the SSCA to find evidence of cyclostationarity, we must define a robust detection statistic. We experimented with using the mean, median, and maximum energy of both $\hat{S}_{xx^{(*)}}^{\alpha}$ and $\hat{\rho}_{xx^{(*)}}^{\alpha}$ (in the remainder of this paper we use $(*)$ in the subscripts of S and ρ to mean both the non-conjugate and conjugate SCF). Recall that for all signal types, including stationary ones, the non-conjugate spectral correlation function reduces to the PSD and the non-conjugate spectral coherence function reduces to unity for all ν. Since we are only interested in cyclostationary signals, it is therefore sufficient to consider only $\alpha \neq 0$. We find that it is not ideal to use $\hat{S}^{\alpha}_{xx^{(*)}}$ for detection because the observed values depend on the input mean and variance of the data, which may not always be known in advance. We tried accounting for this by normalizing our input data to have zero mean and unit variance, but this biased $\hat{S}^{\alpha}_{xx^{(*)}}$ in the presence of strong signals. We had much better results using the $\hat{\rho}^{\alpha}_{xx^{(*)}}$ and so adopted this approach for all the results presented here. Furthermore, as explained in §5.1, our algorithm works best when based on the maximum amplitude of the spectral coherence, as opposed to the mean or median.

The maximum amplitude of $\hat{\rho}^{\alpha}_{xx^{(*)}}$ follows a generalized extreme value (GEV) distribution, whose probability density function is

$$f(x;\mu,\sigma,\xi) = \frac{1}{\sigma} t(x)^{\xi+1} e^{-t(x)} \tag{12}$$

where

$$t(x) = \begin{cases} e^{-\frac{x-\mu}{\sigma}} & \text{if } \xi = 0, \\ \left[1+\xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-\frac{1}{\xi}} & \text{if } \xi \neq 0 \end{cases} \tag{13}$$

The quantile function for the GEV distribution is

$$Q(p) = \begin{cases} \mu - \sigma \ln\left[-\ln(p)\right] & \text{if } \xi = 0, \\ \mu + \sigma \frac{[-\ln(p)]^{-\xi} - 1}{\xi} & \text{if } \xi \neq 0 \end{cases} \tag{14}$$

We can thus set a detection threshold, p_{thresh} such that we consider the data set under analysis to show significant evidence of cyclostationarity when

$$\max\left\{|\rho^{\alpha\neq 0}_{xx^{(*)}}(\nu)|\right\}_{\text{observed}} > Q(p_{\text{thresh}}) \, . \tag{15}$$

In principle the shape parameters should be independent of the implementation details of the SSCA, but in practice we find a small but complicated dependence on the choice of M and N, and especially on the windowing function. We determined the shape parameters empirically for a Hann window for various combinations of M and N, and for the non-conjugate and conjugate spectral coherence function, by generating normally distributed complex random values, passing the data through our SSCA implementation, and recording $\max\left\{|\rho^{\alpha\neq 0}_{xx^{(*)}}(\nu)|\right\}_{\text{observed}}$. We repeated this procedure 10^3 times and fit a GEV distribution to the results using the `stats` module in SciPy, recording the best-fit values of μ and β. The results are shown in Table 1. Recall that we only considered cases where $M/N \geq 8$. We use these shape parameters to determine $Q(p_{\text{thresh}})$ for any given combination of M, N, and conjugate/non-conjugate spectral coherence.

We also explored using the mean and median values of the SCF as a detection statistic, which follow normal distributions in the presence of noise. Since they do not perform as well as the maximum value of the SCF, we do not report the distribution parameters here.

4. SIMULATIONS

We wish to measure the efficacy of our algorithm for various types of RFI and to determine the optimal values of M, N, and p_{thresh}. We are especially interested in emulating the data stream of modern radio telescope instruments so that our findings can be readily applied in real-world contexts. To do so, we simulated a large number of data sets and applied our algorithm for different parameter combinations. The steps in our simulations were

1. Define the signal parameters: symbol duration (t_{sym}, the inverse of the Baud rate), bits per symbol (n_{bit}), and energy per symbol (E_{sym}).

Table 2. Simulation and Algorithmic Parameters[a]

Modulation Type	t_{sym} (samples)	n_{bit}	E_{sym}/N_0 (dB)	M	N	p_{thresh}
ASK	30	1	3	32	1024	0.001
OOK	32	2	5	64	2048	0.01
QAM	100	4	10	128	4096	0.05
FSK	128	6	20		8192	0.3
PSK	300	8			16384	0.6
	512				32768	0.9
	1000					0.95
	1024					0.99
						0.999
						0.9999

[a]This table is meant to be read down each column, and not across each row. Note that OOK signals are by definition limited to 1-bit.

2. Generate a symbol sequence, s, in the form of random integers in the interval $[0, 2^{n_{\mathrm{bit}}})$, and use this to modulate some property of a complex exponential carrier wave. We simulated signals with seven different types of modulation: amplitude shift keying (ASK), on-off keying (OOK; a special case of ASK), quadrature amplitude modulation (QAM; also a special case of ASK), phase shift keying (PSK), and frequency shift keying (FSK). .

3. Add a simulated astrophysical spectral line with a Gaussian profile.

4. Include additive white Gaussian noise (AWGN) with some noise power spectral density (N_0).

5. Pass the final time series through a simulated astronomical spectrometer, producing a number of narrow-band, Nyquist-sampled complex voltage time series corresponding to different frequency channels.

6. Define a "ground truth" of which spectrometer channels and time samples contain the simulated RFI.

7. Independently analyze the output of each spectrometer channel using our SSCA-based algorithm using both the non-conjugate and conjugate spectral coherence function for various combinations of M, N, and p_{thresh}.

8. Compare the output of our algorithm with the ground truth record and characterize the performance of the algorithm using various metrics.

9. Repeat this process ten times for each signal parameter and algorithmic combination in order to better characterize the distribution of the various performance metrics.

In all cases we worked in normalized units, i.e. with a sampling rate $f_{\mathrm{s}} = 1$ Hz. The carrier frequency of the simulated RFI was $f_{\mathrm{c}} = 0.3$ Hz. We always used a noise power of

$N_0 = 1\ \mathrm{W\,Hz^{-1}}$, so that the S/N is equivalent to the value E_{sym}. The full parameter space of our simulations is shown in Table 2. In the following sections we describe the above steps in more detail.

4.1. *Simulated Interference Signals*

There are many different modulation processes in use with telecommunications signals, some of which are quite complex. While we are interested in eventually characterizing our algorithm with as many encoding schemes as possible, as a first step we limit our simulations to a simplified and somewhat idealized parameter space using basic amplitude, phase, and frequency shift keying processes. Each symbol sequence, denoted as s, was a pulse train that consisted of n_{sym} symbols that were each t_{sym} in length, so that s was a total of $n_{\mathrm{sym}} \times t_{\mathrm{sym}}$ samples long. The symbols themselves were simply random integers in the interval $[0, 2^{n_{\mathrm{bits}}})$. This symbol sequence was convolved with a Hann window to reduce spectral leakage. The carrier wave for each signal was

$$x(t) = \sqrt{\frac{E_{\mathrm{sym}}}{t_{\mathrm{sym}}}} e^{-2\pi i f_c t}. \tag{16}$$

Using this definition the integrated energy of $x(t)$ is $n_{\mathrm{sym}} E_{\mathrm{sym}}$. The modulation schemes are described below.

4.1.1. *Amplitude Shift Keyed Signals*

For generic ASK signals the modulated amplitude is related to the symbol sequence by

$$a = \frac{2s}{2^{n_{\mathrm{bit}}} - 1} - 1. \tag{17}$$

This normalization ensures that the amplitude modulation is defined on the interval $[-1, +1]$. To ensure that the integrated energy is $n_{\mathrm{sym}} E_{\mathrm{sym}}$ we divided the final signal by the standard deviation of a. For the special case of an OOK signal, $n_{\mathrm{bit}} = 1$ and $a = s$ without any normalization, i.e. a is either 0 or 1.

We also simulated signals using QAM, which consists of two carrier waves, known as the in-phase (I) and quadrature (Q) components, which have the same frequency while being 90° out of phase. The amplitude of I and Q are modulated independently according to Equation 17 using different symbol sequences. The total number of bits is split evenly between the two sequences. When $n_{\mathrm{bit}} = 1$, $Q = 0$ and only I is used.

4.1.2. *Phase Shift Keyed Signals*

For PSK signals with $n_{\mathrm{bit}} \geq 2$, the phase modulation is given by

$$\phi = \frac{2s + 1}{2^{n_{\mathrm{bit}}}} \pi. \tag{18}$$

Using this definition the discrete phases are bounded on $[\pi/2^{n_{\mathrm{bit}}}, \pi(2 - 1/2^{n_{\mathrm{bit}}})]$. However, when $n_{\mathrm{bit}} = 2$, we instead follow the typical convention that ϕ switches between 0 and π.

4.1.3. *Frequency Shift Keyed Signals*

We simulated a voltage controlled oscillator to generate FSK signals. The oscillator frequency was defined as

$$f = f_0 + sK_0 \tag{19}$$

where f_0 is the quiescent oscillator frequency (in our case, the frequency of the carrier wave) and K_0 is the oscillator gain in units of $\mathrm{Hz\,V^{-1}}$. We defined the phase of the carrier by integrating over f, thus ensuring that the phase was continuous across frequency shifts. In our simulations we used $K_0 = 0.01$ Hz.

4.2. *Simulated Spectral Line*

Our algorithm should be insensitive to stationary astronomical sources. We confirmed this by adding a voltage time series corresponding to a spectral line with a Gaussian profile.

In all our simulations the line had an amplitude of 20 V, was centered at a frequency of 0.1 Hz, and had a full-width at half-maximum of 0.01 Hz. We first created the line with the relevant parameters in the frequency domain but with random phases, mimicking an incoherent astrophysical source. We then took an inverse Fourier transform to create the corresponding voltage time series.

4.3. *Simulated Spectrometer*

We passed the input data stream through a 64-channel, 24-tap polyphase filterbank (PFB) spectrometer (Price 2021). This architecture is similar to that of the Versatile Green Bank Astronomical Spectrometer (VEGAS), the primary backend for the GBT (Prestage et al. 2015). In our implementation, we read $64 \times 24 = 1536$ complex samples, multiplied this time series by a windowing function of the same length, reshaped the data set into a 64×24 array, took a fast Fourier transform along the first axis, and then summed the result. This created an amplitude spectrum with 64 Nyquist-sampled channels. The window that we used was the product of a sinc function and Hann window. We did not form a power spectrum by taking the square modulus of the PFB output, but instead retained the full phase information. We repeated this channelization step until we accumulated $10N$ amplitude spectra.

4.4. *Ground Truth Determination*

The Hann window that we used to taper the symbol sequence and our PFB implementation both greatly reduce spectral leakage, but do not eliminate it completely. Therefore, the RFI signal is present at some level across all PFB channels, but usually at a level that is not expected to corrupt astronomical data. For the purposes of defining the ground truth comparison record, we passed both a noise-free version of the signal and the realization of AWGN through our PFB and formed the resulting power spectra. We considered the signal to be present at a significant level when its power was greater than or equal to the corresponding noise power.

4.5. *SCF Estimation and Flagging*

The output of the PFB was 64 narrow-band times series, each $10N$ points long. We analyzed each channel independently in segments that were each N points long (recall that, in the SSCA, N is equal to the number of discrete α at which the SCF is estimated), resulting in ten SCF estimates for each PFB channel across our full data set. Note that there is a trade-off in the choice of N between cycle frequency resolution and the time resolution with which we can flag data as being contaminated with RFI.

4.6. *Performance Metrics*

We computed several binary classification metrics. First, we compared the output of our algorithm for both the non-conjugate and conjugate SCF to our ground truth definition and counted the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). We also computed these for the union of the non-conjugate and conjugate outputs. From these we calculated the following metrics:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{20}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{21}$$

$$\phi = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}} \tag{22}$$

where TPR is the true positive rate, FPR is the false positive rate, and ϕ, also known as the Matthews correlation coefficient, is a widely-used binary classification metric that performs well for imbalanced classes. We also plot receiver operating characteristic (ROC) curves,

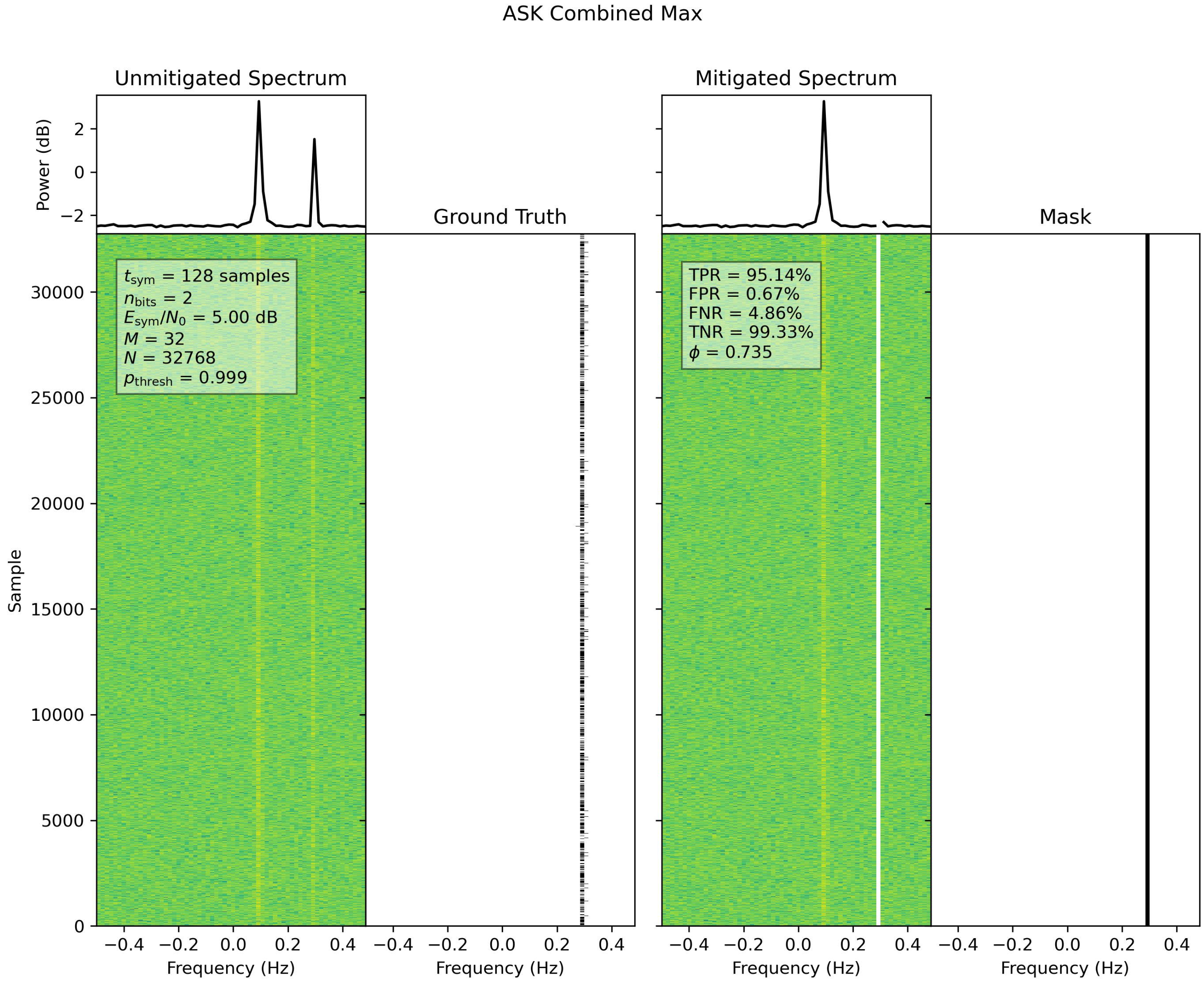

Figure 2. A summary plot for an ASK signal with $t_{\mathrm{sym}} = 128$ samples, $n_{\mathrm{bits}} = 2$, and $E_{\mathrm{sym}}/N_0 = 5$ dB, processed using $(M, N) = (32, 32768)$ and $p_{\mathrm{thresh}} = 0.999$. We show here the results of combining our algorithmic output for both the non-conjugate and conjugate SCF. This is one of the best performing combination of parameters in our simulations. In the mitigated spectrum we remove samples that are flagged by our algorithm (indicated in the mask panel), which completely removes the simulated signal. The simulated astrophysical spectral line appearing at 0.1 Hz is unaffected.

i.e. FPR vs TPR for different values of p_{thresh}, and from these estimate the area-under-curve (AUC) value using a trapezoidal integration method as implemented in SciPy. A perfect classifier will have an ROC curve that immediately rises to a TPR of 1.0 and an FPR of 0.0, and will maintain a TPR of 1.0 while the FPR rises as lower thresholds are used. The corresponding AUC would be 1.0. An uninformative classifier has an ROC curve with a slope of one and an AUC of 0.5. Values of ϕ and AUC in excess of 0.7 are generally considered to be good, and values in excess of 0.8 are generally considered to be very good.

5. RESULTS

Figure 2 shows an example summary plot from one of our simulations. We used 423,360 com-

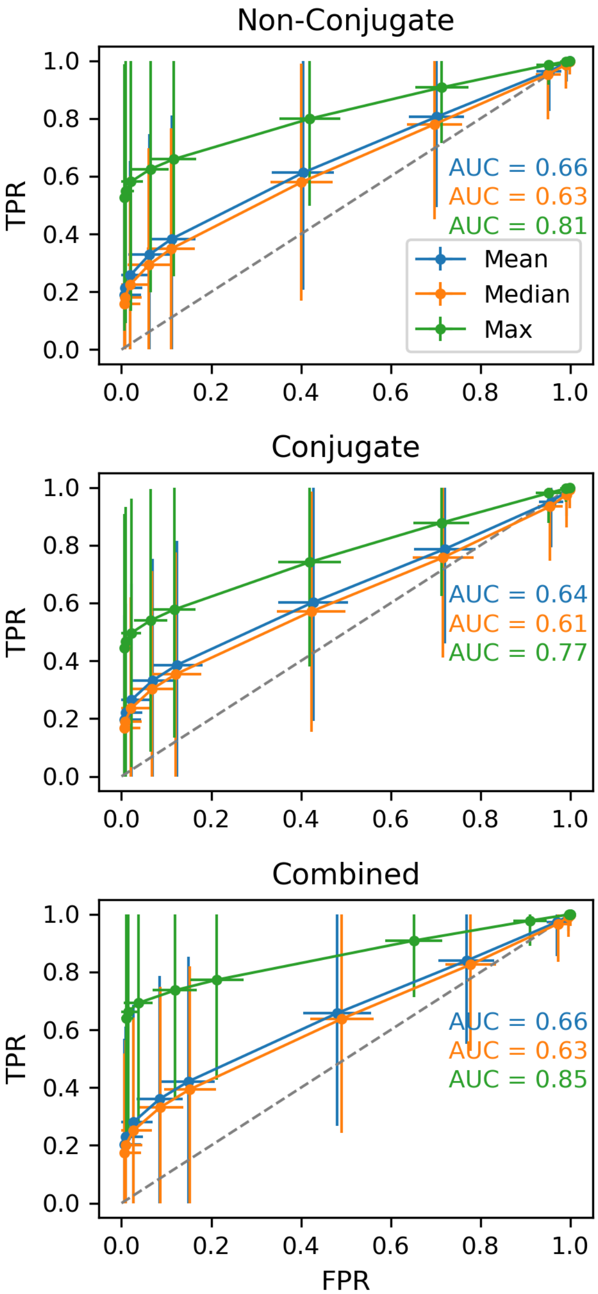

Figure 3. ROC curves with associated AUC values for different detection metrics using the non-conjugate (top) and conjugate (middle) SCF, and the combination of the two (bottom). The combined results using the maximum value of the SCF yields the highest AUC.

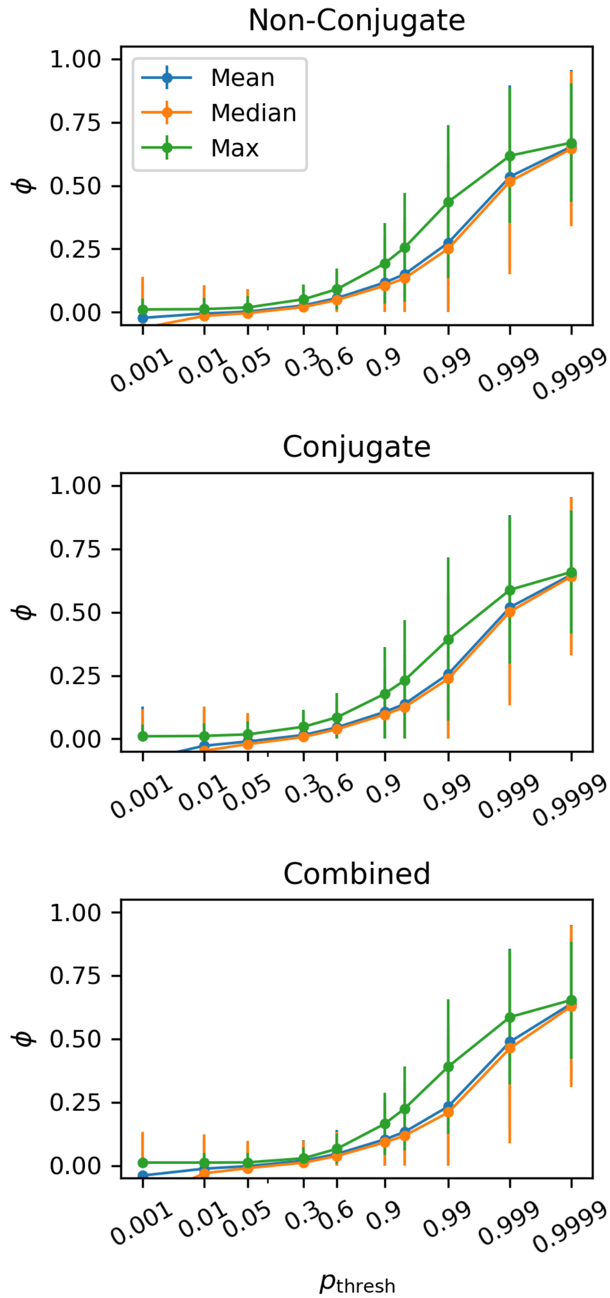

Figure 4. ϕ as a function of $p_{\rm thresh}$ for different detection metrics. Line colors are the same as in Fig. 3 The combined results using the maximum value of the SCF yields the highest /*phi* at a value of 0.65 for $p_{\rm thresh} = 0.9999$.

binations, and exploring this large parameter space is challenging. We begin by determining whether it is most effective to flag based on the mean, median, or maximum value of the SCF. Next, we find the optimal values of M, N, and $p_{\rm thresh}$, and then investigate how the performance varies with different signal properties.

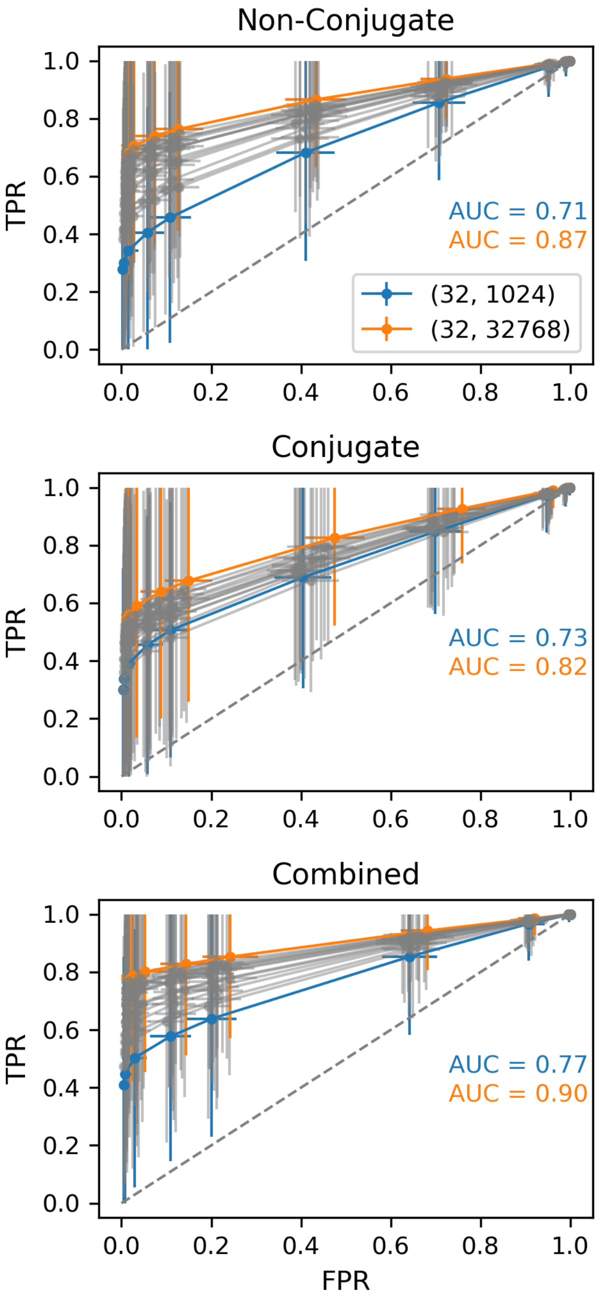

Figure 5. ROC curves for different combinations of M, and N. Curves for all combinations are plotted to provide a complete picture of the performance of our algorithm, but we highlight two combinations of interest. See text for details.

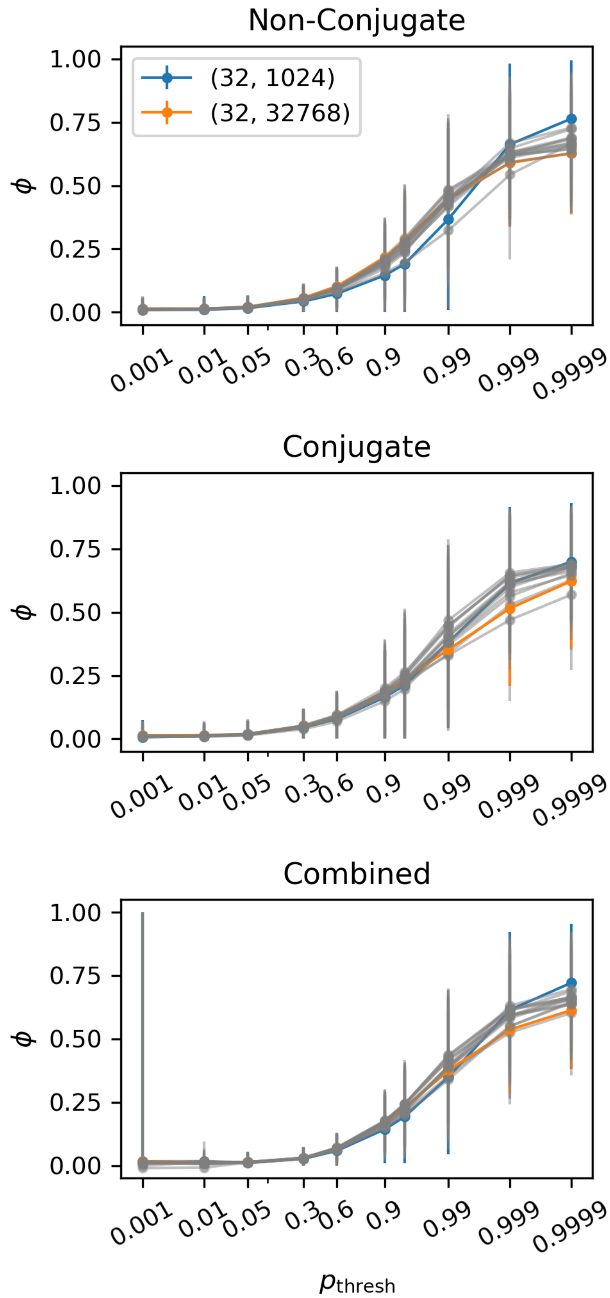

Figure 6. ϕ coefficients for different combinations of M, and N. As with Fig. 5, we plot all combinations but highlight two of interest.

5.1. *Optimal Detection Metric*

Figures 3 and 4 show ROC curves and ϕ for different detection metrics when aggregating the results over all other simulation parameters. As noted previously, using the maximum value of the SCF significantly outperforms the mean or median, with an AUC of 0.85 and maximum ϕ value of 0.65 when combining results for the non-conjugate and conjugate SCF. We consider these to be fairly good results, especially considering that they cover a wide range of values of M and N, signal types, t_{sym}, n_{bits}, and E_{sym}/N_0. In the remainder of this paper we will only consider results when using the maximum value of the SCF as a detection metric.

5.2. *Optimal* (M, N) *Pair*

In Figure 5 we show ROC curves and in Figure 6 we show ϕ for all combinations of M and N. In both figures we averaged over all signal properties (i.e., modulation type, t_{sym}, n_{bits}, and E_{sym}/N_0). This provides the most complete measure of the performance of our algorithm but, as we will see, it includes signal properties for which the algorithm has weaknesses. We highlight two (M, N) pairs of particular interest, representing the highest AUC and ϕ.

The highest AUC is 0.90, which is obtained when using $(M, N) = (32, 32768)$ and the combination of the non-conjugate and conjugate SCF. We consider this an excellent score. The lowest AUC is 0.71, which is obtained for $(M, N) = (32, 1024)$ when using only the non-conjugate SCF, which is still a good AUC score. However, the situation is reversed when considering ϕ, i.e. the highest value is $\phi = 0.72$ for $(M, N) = (32, 1024)$ at $p_{\mathrm{thresh}} = 0.9999$, while for $(M, N) = (32, 32768)$ and p_{thresh}, $\phi = 0.61$.

The discrepancy between AUC and ϕ can be understood by examining Table 3, which shows the TPR, FPR, TNR, and FNR for the two cases discussed above. As N increases from 1024 to 32768, the TPR increases by a factor of 1.9, but the FPR increases by an even larger factor of 3.9. The ϕ coefficient punishes the algorithm for this larger relative increase in FPR. However, we note that the absolute improvement in TPR is 0.368, while the absolute deterioration in FPR is only 0.0129, and remains quite low. In a real-world context, the question of which parameters are "better" will depend on the scientific goals of the observation. In some cases the large absolute improvement in TPR will make the slightly higher FPR tolerable, while other cases may require a lower FPR. For completeness we will report performance for both $(M, N) = (32, 1024)$ and $(32, 32768)$ in the remainder of this paper.

Table 3. Performance for $p_{\mathrm{thresh}} = 0.9999$

(M, N)	TPR	FPR	TNR	FNR
$(32, 1024)$	0.409	0.00451	0.996	0.591
$(32, 32768)$	0.777	0.0174	0.983	0.223

It is worth asking why the FPR increases with N? From first principles, we would expect the SSCA to be more accurate as N increases because it is a time-averaging technique for estimating the SCF, and by analyzing more data the signal-to-noise ratio of an interfering signal should go up. The observed behavior likely results from our method for defining the ground truth comparison. Recall that we mark a sample as truly containing RFI when the amplitude of the simulated signal is equal to the noise level. In the low signal-to-noise regime this will be sensitive to the exact realization of the noise. Such a situation can occur when RFI spills over with reduced amplitude into nearby PFB channels. Since we identify RFI in segments of length N, the algorithm may flag data that technically falls just below the threshold for being included in our ground truth mask, leading to those samples being marked as false positives. An analogous situation could arise in the presence of transient RFI because good data will be flagged along with bad. We discuss potential ways to mitigate this shortcoming in §6.

We further note that, for any given value of N, there is a preference for smaller values of M,

Table 4. Performance for Various p_{thresh}

N		0.898^{a}	0.95	0.984^{b}	0.99	0.998^{c}	0.9999	$1 - 4.61 \times 10^{-5}$ [d]
1024	TPR	0.641	0.577	0.519	0.502	0.456	0.409	0.239
	FPR	0.205	0.109	0.0410	0.0291	0.0100	0.00451	0.00254
32768	TPR	0.854	0.828	0.807	0.801	0.788	0.777	0.457
	FPR	0.247	0.143	0.0671	0.0526	0.0276	0.0174	0.0100

[a] Minimizes Eq. 23 for $N = 1024$

[b] Minimizes Eq. 23 for $N = 32768$

[c] Yields FPR $= 0.01$ for $N = 1024$

[d] Yields FPR $= 0.01$ for $N = 32768$

but the dependence on M is fairly weak. For the sake of simplicity we will only report results for $M = 32$ going forward. This is fortuitous because smaller M reduce the computational complexity of the SSCA.

5.3. *Optimal* p_{thresh}

The optimal parameters for any classification algorithm will depend on the tolerance for false positives and false negatives, with ϕ being one commonly used measure. Figure 6 shows that ϕ is maximized for $p_{\text{thresh}} = 0.9999$ when aggregating over all simulation parameters, but there are diminishing returns for $p_{\text{thresh}} > 0.999$.

Another approach is to select a p_{thresh} based upon the ROC curves. A perfect classifier will always have TPR $= 1$ and FPR $= 0$, so we could choose the p_{thresh} that comes closest to this point. This is equivalent to finding the minimum of

$$\sqrt{\text{FPR}^2(p_{\text{thresh}}) + [1 - \text{TPR}(p_{\text{thresh}})]^2}. \quad (23)$$

We used the SciPy `PchipInterpolator` routine, which implements a piecewise cubic Hermite interpolating polynomial, to interpolate FPR and TPR as a function of p_{thresh}, and then used Brent's method to find the minimum of Equation 23. For $N = 1024$ this results in $p_{\text{thresh}}^{\text{opt}} = 0.898$ and for $N = 32768$ the optimal result is $p_{\text{thresh}}^{\text{opt}} = 0.984$ (recall that we only consider the case of $M = 32$).

Yet another approach is to choose a sensible false alarm probability for the maximum value of the SCF to exceed what is expected for AWGN, e.g. $p_{\text{thresh}} = 0.95$ or 0.99. While this may be attractive because it is motivated by the statistics for the SCF, we stress that it is *not* equivalent to the FPR of our algorithm, because we flag data in segments of length N (see §5.2 and §6).

In Table 4 we show TPRs and FPRs for various choices of p_{thresh} for our two representative values of N. For the remainder of this paper we will present results for $p_{\text{thresh}} = 0.99$ unless otherwise noted. We have chosen this value for three reasons: 1) it results in FPR $\lesssim 0.02$ for $N = 1024$ and FPR $\lesssim 0.05$ for $N = 32768$; 2) it is very close to the optimal value for $N = 32768$ when using Eq. 23; and 3) it is one of the values we directly simulated, avoiding the need to interpolate other results.

5.4. *Performance for Different Modulation Types*

In Figures 7 and 8 we show ROC curves and ϕ separated by modulation type for both $N = 1024$ and $N = 32768$ (recall that we only con-

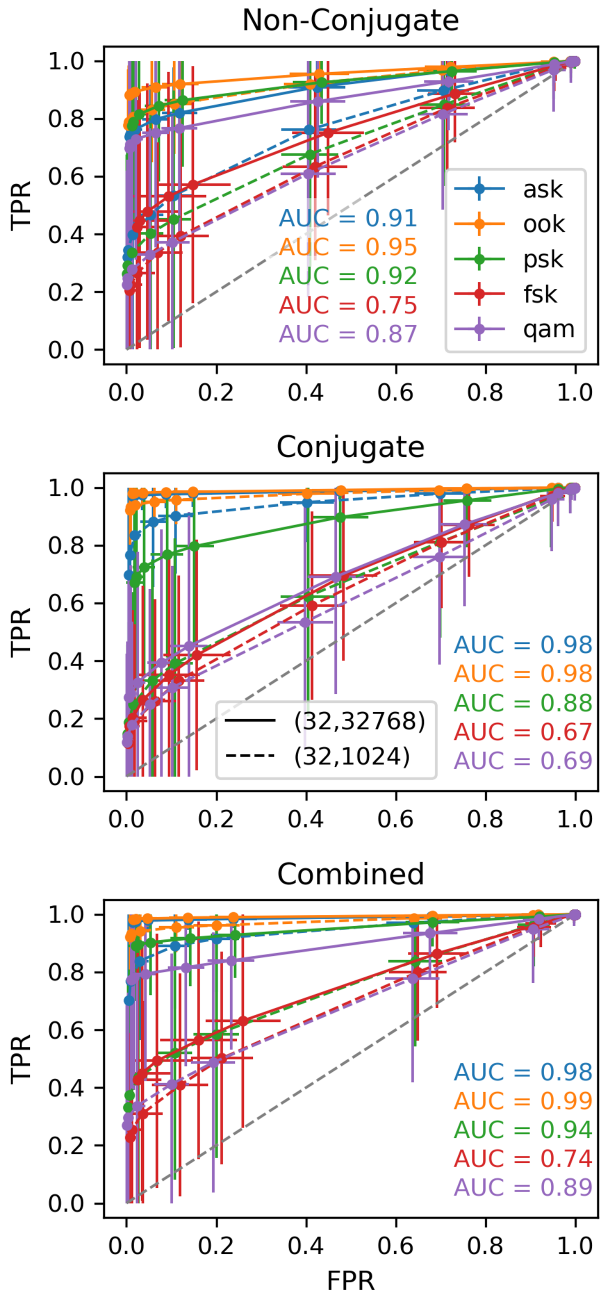

Figure 7. ROC curves and AUC values for different modulation types. Different color curves represent different modulation types, and different line styles indicate the two representative values of N that we consider. We see excellent performance for ASK, OOK, and PSK modulation, with good performance for QAM modulation, but a weakness to FSK signals.

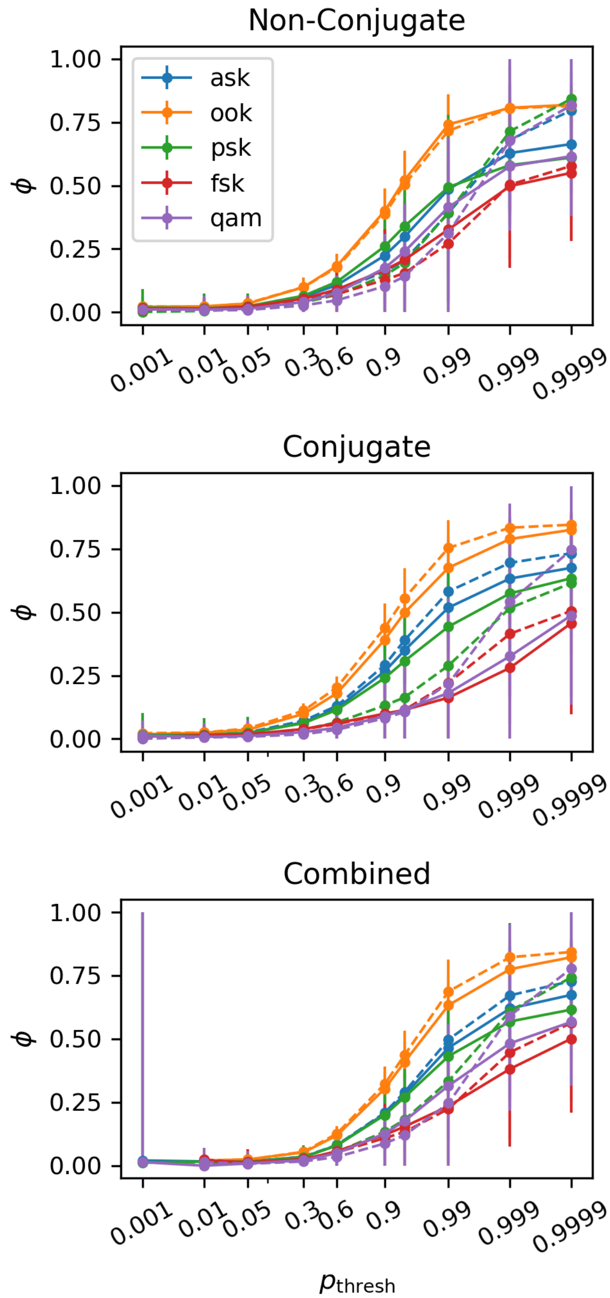

Figure 8. ϕ for different modulation types and values of N. The colors and line styles are the same as in Fig. 7. As with AUC, the results are best for ASK, OOK, and PSK modulation, and reasonably good for QAM modulation. However, the algorithm does not perform well for FSK modulation.

sider $M = 32$). Our algorithm works extremely well for OOK signals, as well as ASK, PSK and QAM signals, with maximum AUC $\gtrsim 0.9$. ϕ

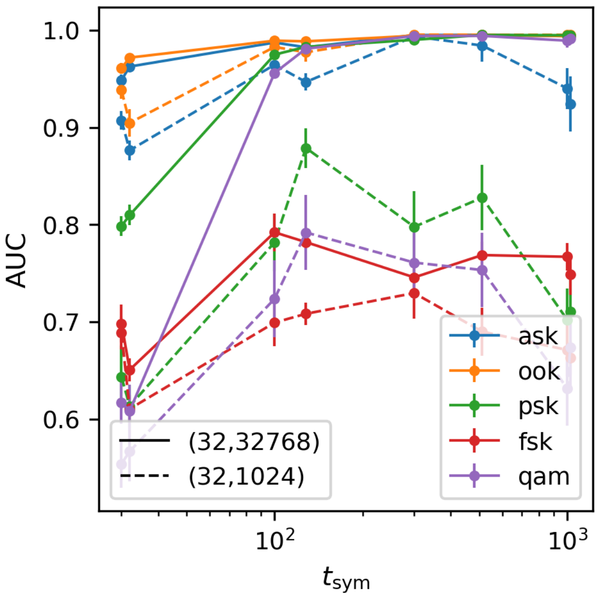

Figure 9. AUC values for different values of t_{sym}. Colors and line styles have the same meaning as in Fig. 7. AUC values increase quickly with larger t_{sym}, up to ~ 128 samples, and then either remain approximately constant or decrease slightly.

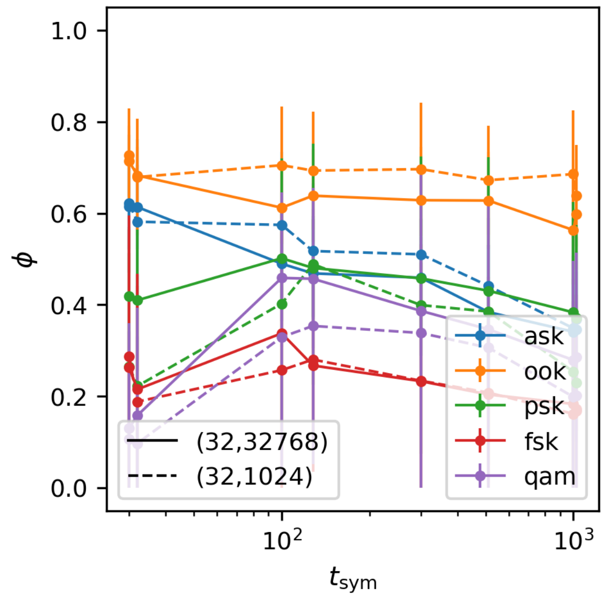

Figure 10. ϕ for different values of t_{sym}. Colors and line styles are the same as in Fig. 7. All values use $p_{\mathrm{thresh}} = 0.99$. As with AUC values, ϕ increases quickly with larger t_{sym}, up to ~ 128 samples, and then either remain approximately constant or decrease slightly.

values span a wider range but are ≈ 0.75 for OOK signals at $p_{\mathrm{thesh}} = 0.99$, and 0.6–0.75 for all other signal types (except FSK), albeit at higher p_{thresh}. The performance is not as good for FSK signals, with a maximum AUC $\simeq 0.74$ and $\phi = 0.56$. As discussed in §6, the relative weakness to FSK signals most likely is a consequence of the way we independently analyze different PFB channels. The frequency shift that is used could exceed the width of a PFB channel, and in some cases the signal may not return to the original PFB channel within N samples, obscuring its cyclostationary nature. Nevertheless, the performance for FSK signals is still reasonably good when aggregating over all other simulation parameters.

ASK and OOK signals are better detected using the conjugate SCF, while PSK, FSK, and QAM signals are better detected using the non-conjugate SCF. The results for the combination of the two conjugation strategies are usually as good, and in some cases slightly better, than the best individual results. This highlights the importance of using both conjugation strategies. For the sake of clarity, in the remainder of this paper we will only present the combined non-conjugate/conjugate results, but we will still separate results by modulation type since it has a significant impact on the performance of the the algorithm.

5.5. *Performance for Different Symbol Durations*

In Figures 9 and 10 we show AUC values and ϕ for different values of t_{sym}, separated by modulation type for our two representative combinations of N (we use $M = 32$ for both). ϕ coefficients are calculated for $p_{\mathrm{thresh}} = 0.99$. As already noted in §5.4, the performance is best for OOK and ASK signals, followed by PSK and QAM, while performance is worst for FSK signals. However, we can now see that results

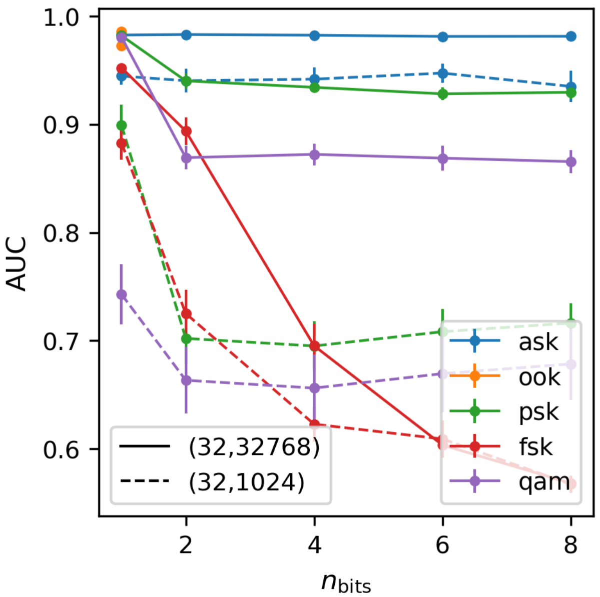

Figure 11. AUC values for different values of n_{bits}. Colors and line styles are the same as in Fig. 7. For most modulation types there is a small drop in performance between $n_{\text{bits}} = 1$ and 2, and relatively constant performance thereafter, though this does depend on the choice of N. However, the algorithm performs successively worse for FSK signals as n_{bits} increases, becoming nearly uninformative when $n_{\text{bits}} = 8$.

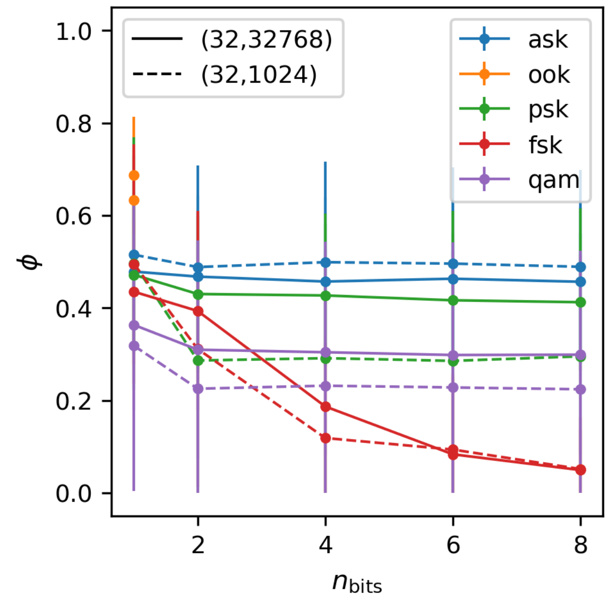

Figure 12. ϕ for different values of n_{bits}. Colors and line styles are the same as in Fig. 7. All values were calculated for $p_{\text{thresh}} = 0.99$. The drop in performance when going from $n_{\text{bits}} = 1$ to 2, is smaller than implied by AUC values. However, as with AUC values, ϕ coefficients imply that the algorithm is nearly uninformative when $n_{\text{bits}} = 8$.

also improve, sometimes significantly, when t_{sym} increases from $\sim$ 30 samples to $\sim$ 100 samples, especially when measuring performance via AUC. We can understand these trends by recalling that we pass our data through a first-stage 64-channel PFB, and analyze each channel independently. When $t_{\text{sym}} \lesssim 64$, signals are spread across multiple PFB channels, reducing the signal-to-noise ratio. Once t_{sym} is greater than the width of a PFB channel, the signal is fully contained within one channel and the performance of the algorithm does not change very much, until reaching the highest values of t_{sym}.

At the highest values of t_{sym} we do see a significant drop in performance when using $N = 1024$, because as t_{sym} increases there are fewer symbols over which we can average to obtain an accurate estimate of the SCF. This is an argument against using small values of N when trying to detect narrow-bandwidth signals.

The algorithm continues to perform poorly for FSK signals because the frequency shift can still exceed the width of a PFB channel.

We tested t_{sym} that are and are not evenly divisible by N, i.e. that have or do not have Baud rates that align precisely with the SSCA cycle frequency bins (see Table 2 for the complete list of t_{sym}). As expected, the performance for Baud rates that are not equal to a cycle frequency bin are somewhat lower than similar Baud rates that do align with a cycle frequency bin. We return to this point in §6.

5.6. *Performance for Different Numbers of Bits*

Figures 11 and 12 show AUC values and ϕ as function of n_{bits} per symbol, separated by mod-

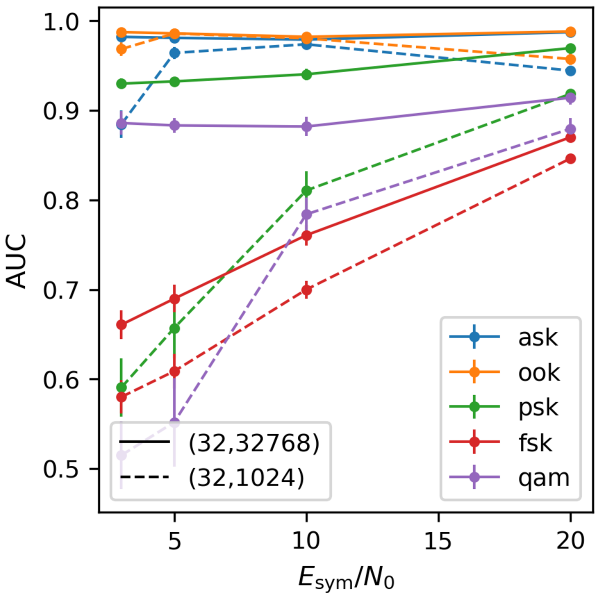

Figure 13. AUC values for different E_{sym}/N_0. Colors and line styles are the same as in Fig. 9. As expected, the algorithm performs better as E_{sym}/N_0 increases for all modulation types.

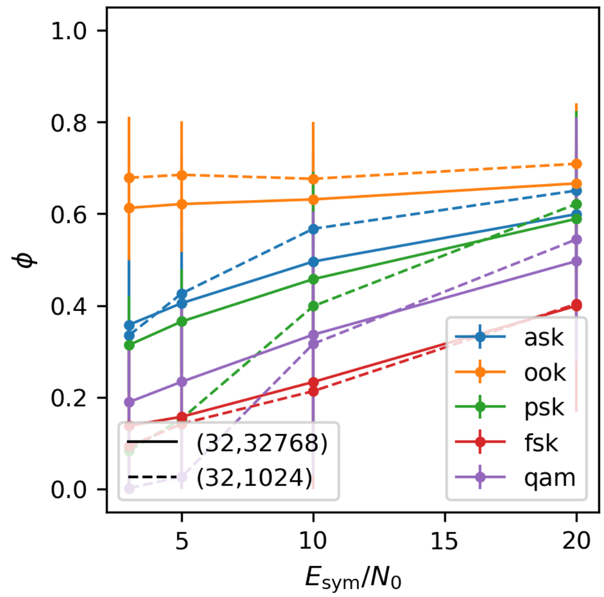

Figure 14. ϕ for different E_{sym}/N_0. Colors and line styles are the same as in Fig. 9. All values were calculated for $p_{\mathrm{thresh}} = 0.99$. As with AUC values, the ϕ coefficients show that the algorithm performs better as E_{sym}/N_0 increases for all modulation types.

ulation type, for our two representative combinations of N (both using $M = 32$). ϕ coefficients are calculated for $p_{\mathrm{thresh}} = 0.99$. There is a slight drop in performance when going from one bit to two for ASK, PSK, and QAM signals, but no significant dependence on n_{bits} at higher values (OOK signals are only 1-bit). However, there is a strong dependence on n_{bits} for FSK signals, with more bits per symbol leading to steadily worse performance. Once again, this is related to our approach of analyzing PFB channels independently. In our implementation, FSK-like signals with more bits per symbol will be spread over a wider range of frequencies. In the extreme case, a signal may not return to a given frequency channel within N samples, in which case its cyclostationary nature will not be detected at all by our algorithm. This does indeed seem to be the case, as can be seen by AUC values approach 0.5 and ϕ coefficients approach zero as n_{bits} increases. However, we can also see that the algorithm performs well for FSK signals when $n_{\mathrm{bits}} = 1$, and its performance remains acceptable up to $n_{\mathrm{bits}} = 2$–4.

5.7. *Performance for Different* E_{sym}/N_0

Figure 14 shows ϕ as a function of E_{sym}/N_0 for different modulation types, for both $N = 1024$ and $N = 32768$, and using $p_{\mathrm{thresh}} = 0.99$. As expected, higher E_{sym}/N_0 leads to better performance. The relative improvement is not as high for ASK, OOK, PSK, and QAM signals since the algorithm already detects these signals well, even at low E_{sym}/N_0, but there is a large relative improvement for FSK signals. However, as discussed in §5.5 and 5.6, there is a strong dependence on other parameters for FSK signals. The improvements seen here are due to those few cases where the algorithm works reasonably well for FSK signals (e.g. $n_{\mathrm{bits}} = 1$). For others, such as very high values of n_{bits}, the algorithm does not work well for FSK signals even at very high E_{sym}/N_0.

5.8. *Performance for Simulated Spectral Line*

As noted above, our algorithm should not identify astrophysical spectral lines as potential sources of RFI because they are not cyclostationary. All of the results discussed in the preceeding sections include a simulated spectral line in the data, and so any false positives would include samples containing signal from this line. To further verify that this simulated line is not mistakenly being identified as RFI, we also simulated data sets containing only the line and analyzed them using all algorithmic parameter combinations and recorded the FPR (since there is no true source of RFI, the TPR is undefined). We did the same for pure AWGN. We then performed a two-sided Kolmogorov-Smirnov test using SciPy's `kstest` routine. We find KS test statistics of 0.0078, 0.0064, and 0.0096 for the non-conjugate SCF, conjugate SCF, and combined results, respectively. These correspond to p-values of ≥ 0.99. As expected, we thus find no evidence for rejecting the null hypothesis that the FPRs of the data sets containing the simulated spectral line and pure noise come from the same distribution.

6. DISCUSSION

These results show that cyclostationary tests are a promising approach to RFI mitigation. Aggregating our results across different signal properties provides a more complete picture of how our algorithm performs, but for particularly favorable combinations of signal properties the performance can be much better than the aggregate results imply. As an example, when using the combined non-conjugate/conjugate SCF, $(M, N) = (32, 32768)$ and $p_{\rm thresh} = 0.99$, for $t_{\rm sym} = 128$ samples, $n_{\rm bits} = 1$, and $E_{\rm sym}/N_0 = 10$ dB, we find TPR > 0.97 and FPR < 0.06 for all modulation types except FSK (which has TPR $= 0.94$ and FPR $= 0.13$). If we choose $(M, N) = (32, 1024)$ and $p_{\rm thresh} = 0.999$ we can achieve $\phi > 0.78$ for all modulation types except QAM ($\phi = 0.72$) and FSK ($\phi = 0.69$). Furthermore, we find no evidence that our algorithm systemically flags the simulated astrophysical spectral line that we included in our simulations. Obviously, we cannot optimize the properties of real-world RFI to maximize the effectiveness of mitigation techniques, but these results do suggest that cyclostationary tests can perform extremely well and potentially complement other approaches. For example, spectral Kurtosis (Nita & Gary 2010a,b; Smith et al. 2022) is a computationally simple statistical method that distinguishes normally-distributed data and from RFI, though it has weaknesses to sidelobe spillover as well as weaker signals and those that have a 50% duty cycle. Smith et al. (2022) measured the performance of single- and multi-scale SK using many of the same simulated sources of RFI as we use here. ϕ scores varied substantially depending on the characteristics of the signal and SK parameters, but could be as high as ~ 0.75 for ASK signals with high data rates, and were usually ~ 0.5–0.7, which is broadly similar to our results (E. Smith, private communication). AOFLAGGER is used on low-frequency arrays such as the Low Frequency Array (LOFAR) and the Murchison Widefield Array (MWA; Offringa et al. 2010a,b, 2012), and flags the post-correlation visibilities with the highly optimized `SumThreshold` method. AOFLAGGER does very well at flagging most RFI in the dataset, but operates on the power values, which means its performance may be hindered by uneven bandpass responses or strong periodic astronomical signals such as pulsars or FRBs.

Nevertheless, the current implementation of our algorithm does have a weakness to signals that are spread across multiple PFB channels, whether because of the modulation technique being used or intrinsic bandwidth of the signal itself. As noted previously, this stems from analyzing each PFB channel independently. When a signal is spread across multiple channels the effective E_{sym}/N_0 decreases. In the worst-case scenario, a frequency-switched signal may not return to a given PFB channel within the block of data that we analyze, completely obscuring its cyclostationary nature. We chose to analyze PFB channels independently in order to more closely match the architecture of the digital spectrometer used at the GBT. In this case, PFB channels are formed on a field programmable gate array prior to being transmitted to computers where a real-time RFI mitigation algorithm might be implemented. However, there are alternative architectures or approaches. For example, RFI mitigation could be implemented prior to the PFB. The PFB operation could also be inverted in software, and groups of PFB channels could be analyzed in groups covering sufficient bandwidth to capture even relatively broad-band RFI. These groups could also be made to overlap by using an oversampled PFB, to avoid missing signals that cross over group boundaries.

In §5.2 we showed that while using larger values of N leads to a higher TPR, it also leads to a higher FPR. We attribute this to our method of defining a ground truth comparison, but we also expect it to be true when data contain transient RFI. Our algorithm operates on data in segments of length N, so there is a chance that samples that are free of RFI will be incorrectly flagged when RFI only contaminates some of the samples. By analyzing data that contains both cyclostationary and non-cyclostationary signals, we would also lower the sensitivity of the algorithm. We could avoid these pitfalls by using multiple values of N and selecting the best value for any given segment of data by choosing the value that maximizes the signal-to-noise ratio of the SCF. Using multiple values of N would also lead to different cycle frequency resolutions, which could help detect signals at different Baud rates. However, this would increase computational cost, which is already high to begin with (see below).

Quantization errors may also impact the performance of our algorithm. In our simulations we generated signals with floating point precision, but modern analog-to-digital converters (ADCs) use much lower quantization depth, e.g. the VEGAS spectrometer used at the GBT outputs 8-bit values. This may be at least partially ameliorated by using higher bit-depth ADCs — commercial models are now available that output 12-bit values and that can sample bandwidths of several GHz.

However, each of these approaches does come with challenges. Sampling with more bits increases data rates, requiring new network topologies. Analyzing the full bandwidth with different values of N would be computationally expensive and may exceed the resources available with modern hardware for all but relatively narrow observing bands. Inverting the PFB operation and using overlapping groups also adds computational cost. The computational complexity of the SSCA algorithm is $O \sim NM \log_2 N$ (Roberts et al. 1991). Processing a bandwidth of 1 GHZ in two separate polarization channels with our optimal algorithmic parameters of $M = 32$ and $N = 32768$ in real-time would thus require a computing system capable of $\sim$ 30 PFLOPS. This is well beyond the capability of current commercial graphics card, but the current generation of GPUs designed for artificial intelligence training offer theoretical maximum computational power of $\sim$ 300–600 TFLOPS, depending on the numerical precision being used. Over the next several years it may become feasible to adopt a hybrid approach, wherein wide observing bandwidths are split into a modest number of overlapping sub-bands and processed independently before being combined to record the full bandwidth. A similar approach has already been developed to enable real-time coherent dedispersion of pulsars using the GBT's 0.7 – 4 GHz ultrawideband receiver, for which the GBT's VEGAS spectrometer will use 24 compute nodes to process 3.3 GHz of instantaneous bandwidth, as well as to enable cyclostationary techniques for studying pulsars (Demorest 2011). Another approach is to eschew real-time RFI mitigation in favor of temporarily recording Nyquist-sampled voltages to disk and processing them offline with some reasonable turnaround time. This approach is used by the Breakthrough Listen project (MacMahon et al. 2018) to process several GHz of instantaneous bandwidth. We leave a detailed analysis of these approaches to future work.

We chose a limited number of idealized signal types to illustrate a CSP-based approach to RFI mitigation, but real-world telecommunications signals can be much more complex. In future work we plan on simulating additional modulation strategies and windowing functions, including more complex astrophysical sources, and adding multiple sources of RFI within the frequency range of interest. More complex strategies for improving our algorithm could also include using multiple PFBs to channelize the data with different numbers of channels. Finally, as an alternative to our blind identification algorithm, we could study the local RFI environment and use cyclostationary detectors that are tuned to sources of RFI with known properties, which would greatly reduce the computational cost. We also plan to apply our algorithm using the optimal parameters derived here to archived astronomical data collected with the GBT.

7. CONCLUSIONS

We have developed an approach to identifying and mitigating RFI by testing whether data contain significant evidence of cyclostationarity, and tested its performance using a range of simulated signals. We find good performance for most simulated signals, with some weaknesses to broad-band and frequency-switched signals. Specifically, when using optimal algorithmic parameters we find AUC scores > 0.90 and ϕ scores $\gtrsim 0.61$, aggregated over all modulation schemes, symbol durations, bits-per-symbol, and signal-to-noise ratios that we simulated. The algorithm performs best for OOK signals and reasonably well for more generic ASK and PSK signals. We find no systemic tendency for our algorithm to incorrectly identify a simulated astrophysical spectral line. We believe that tests of cyclostationarity are a promising technique for RFI mitigation that can complement other approaches.

This work is supported by the National Science Foundation through Advanced Technologies and Instrumentation grant #1910302. We are grateful to an anonymous referee for providing comments that improved the quality of this manuscript, and to Chad Spooner for helpful discussions and for maintaining cyclotationary.blog.

REFERENCES

Akeret, J., Chang, C., Lucchi, A., & Refregier, A. 2017, Astronomy and Computing, 18, 35, doi: 10.1016/j.ascom.2017.01.002

Carter, N. J. 1992, Master's thesis, Naval Postgraduate School, Monterey, CA.

Cucho-Padin, G., Wang, Y., Li, E., et al. 2019, Radio Science, 54, 986, doi: 10.1029/2019RS006902

Demorest, P. B. 2011, MNRAS, 416, 2821, doi: 10.1111/j.1365-2966.2011.19230.x

Gardner, W. A. 1991, IEEE Signal Processing Magazine, 8, 14, doi: 10.1109/79.81007

Gardner, W. A., Napolitano, A., & Paura, L. 2006, Signal Processing, 86, 639, doi: https://doi.org/10.1016/j.sigpro.2005.06.016

Hellbourg, G., Weber, R., Capdessus, C., & Boonstra, A.-J. 2012, Comptes Rendus Physique, 13, 71, doi: 10.1016/j.crhy.2011.10.010

MacMahon, D. H. E., Price, D. C., Lebofsky, M., et al. 2018, PASP, 130, 044502, doi: 10.1088/1538-3873/aa80d2

Nita, G. M., & Gary, D. E. 2010a, MNRAS, 406, L60, doi: 10.1111/j.1745-3933.2010.00882.x

—. 2010b, PASP, 122, 595, doi: 10.1086/652409

Nita, G. M., Gary, D. E., Liu, Z., Hurford, G. J., & White, S. M. 2007, PASP, 119, 805, doi: 10.1086/520938

Offringa, A. R., de Bruyn, A. G., Biehl, M., et al. 2010a, MNRAS, 405, 155, doi: 10.1111/j.1365-2966.2010.16471.x

Offringa, A. R., de Bruyn, A. G., Zaroubi, S., & Biehl, M. 2010b, arXiv e-prints, arXiv:1007.2089. https://arxiv.org/abs/1007.2089

Offringa, A. R., van de Gronde, J. J., & Roerdink, J. B. T. M. 2012, A&A, 539, A95, doi: 10.1051/0004-6361/201118497

Okuta, R., Unno, Y., Nishino, D., Hido, S., & Loomis, C. 2017, in Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS). http://learningsys.org/nips17/assets/papers/paper_16.pdf

Pinchuk, P., & Margot, J.-L. 2022, AJ, 163, 76, doi: 10.3847/1538-3881/ac426f

Prestage, R. M., Bloss, M., Brandt, J., et al. 2015, in 2015 URSI-USNC Radio Science Meeting, 4, doi: 10.1109/USNC-URSI.2015.7303578

Price, D. C. 2021, in The WSPC Handbook of Astronomical Instrumentation, Volume 1: Radio Astronomic al Instrumentation, ed. A. Wolszczan, 159–179, doi: 10.1142/9789811203770_0007

Purver, M., Bassa, C. G., Cognard, I., et al. 2022, MNRAS, 510, 1597, doi: 10.1093/mnras/stab3434

Roberts, R. S., Brown, W. A., & Loomis, Herschel H., J. 1991, IEEE Signal Processing Magazine, 8, 38, doi: 10.1109/79.81008

Smith, E., Lynch, R. S., & Pisano, D. J. 2022, AJ, 164, 123, doi: 10.3847/1538-3881/ac7e47

Thompson, A., & Nicely, M. 2021, cuSignal: The GPU-Accelerated Signal Processing Library, 0.19.0. https://github.com/rapidsai/cusignal

Vafaei Sadr, A., Bassett, B. A., Oozeer, N., Fantaye, Y., & Finlay, C. 2020, MNRAS, 499, 379, doi: 10.1093/mnras/staa2724

Yuan, M., Zhu, W., Zhang, H., et al. 2022, MNRAS, 513, 4787, doi: 10.1093/mnras/stac963