Enhanced Weak Signal Detection Using SVM Based Correlation Algorithm

Samuel L. Kramer

Thesis submitted to the Faculty of

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

Alfred Wicks, Chair

Chris Fuller

Steve Southward

May 3rd, 2024

Blacksburg, Virginia

Keywords: Digital Signal Processing, Signal Detection, Engineering Acoustics

Enhanced Weak Signal Detection Using SVM Based Correlation Algorithm

Samuel L. Kramer

(ABSTRACT)

Traditional signal detection algorithms are often robust and are typically sufficient for high SNR data. However, the assumptions behind these methods begin to fall apart when signal period becomes either very short, or small in amplitude compared to any corruptive noise. To address this a kernel transform based cross-correlation algorithm is proposed for the application of weak signal detection. The algorithm leverages kernel methods to inflate SNR of the data and enhance the noise rejection capabilities of the traditional cross-correlation. The goal of the algorithm is to achieve detection for signals past the limits of those of the matched filter and the cross-correlation in the presence of white and colored noise. To evaluate the effectiveness of the correlation algorithm, Monte Carlo simulations are performed to determine the performance in the context of different types of noise. The performance of the algorithm will be compared against the cross-correlation and the matched filter.

Enhanced Weak Signal Detection Using SVM Based Correlation Algorithm

Samuel L. Kramer

(GENERAL AUDIENCE ABSTRACT)

As society advances, the tools we rely on become increasingly more intricate. Pivotal to the development of these systems is the algorithms used to process the data they collect. Particularly crucial to the field of signal processing, weak signal detection is focused on the processing of barely comprehensible data in the context of powerful noise. In recent years, advancements in weak signal detection have focused on pushing the theoretical limits of signal discernibility, especially when heavily obscured by noise. Leveraging the power of machine learning, certain AI algorithms have showcased promise in the detection of weak signals. It has yet to be seen if a foundational principle of AI called a kernel transform can be applied to classic signal detection theory to increase detection performance. This thesis will propose a kernel based detection algorithm for weak signal detection and the performance of the algorithm will be compared against previously established theory. New breakthroughs in detection algorithms facilitate improvements in active and passive sonar, medical devices and even the finance sector.

Dedication

To my parents, who always supported me

Contents

List of Figures

List of Abbreviations

σ Gaussian width hyperparameter

τ The time delay of a correlation algorithm. Is referred to as lags and is a function of the sample rate of a signal where the max lag of a correlation will be $2N - 1$.

AI Artificial Intelligence

autocorrelation Abbreviation for the autocorrelation function.

DFT Discrete Fourier Transform

FFT Fast Fourier Transform

FIR Finite Impulse Response; A digital filter with a finite duration response to an impulse input

KCC Kernel Cross-Correlation algorithm

L-Abs Linear Absolute Value Kernel

PSD Power Spectral Density; the frequency decomposition of the autocorrelation of a signal.

RBF Radial Basis Function, synonym for Gaussian kernel

SNR Signal-to-Noise Ratio

SSE Sum of Squared Errors

STFT Short-Time-Fourier-Transform

SVM Support Vector Machine

TFD Time-Frequency Distribution

XCF Abbreviation for the cross-correlation function.

Chapter 1

Introduction

The field of signal processing has become increasingly important with a continually modernizing society and is deeply intertwined in many of the electronic products used today. Our phones and computers are laden with image compression algorithms and audio filtering protocols to remove background noise. Signal processing presents its own set of challenges that remain a large focus of academic research.

Signal detection theory had its infancy in World War II radar technology that specified optimal observation for detecting electronic signals with random white noise [1]. The field took even further strides in the mid 20th century amid the backdrop of the Cold War. Naval sonar systems positioned to spy on enemy ship movements relied on the ability to capture and detect noise caused by submarines. Modern passive sonar arrays are able to detect ships, reliably determine direction of travel, and the country of origin based off the noise generated from the rotor. Similarly, the discovery of the Fast Fourier Transform during the Cold War was motivated by the American desire to accurately detect Soviet underground nuclear tests from across the world.

The natural progression of physics and technology has demanded the advancement in signal detection theory, and over time various techniques have evolved for different applications. However, issues still arise when the signal is very short, incoherent, or "weak" compared to noise. In these situations, the underlying assumptions foundational to detection theory begin to fall apart and many techniques begin to fail [2].

Over the past three decades, significant claims regarding detection performance have been attached to new advancements in linear and non-linear methods [3] [4]. More recently, revolutions in Artificial Intelligence (AI) models have proved promising for the application of signal detection. It remains to be proven that an application of AI and data science principles to a classic linear algorithm could extend the detection performance of the algorithm.

1.1 Motivation

Most signal processing methods begin to lose effectiveness as the power of noise in the data increases. Data with signal-to-noise power ratios close to unity typically constitutes a weak signal. This can be achieved through a very short signal period or small signal amplitude. Regardless, the smaller the power ratio becomes, the more difficult the signal is to process. There also exists a sensitivity to types of noise and signals of interest. Some linear and AI driven methods fail disproportionately based on the signal and noise models used.

Support Vector Machine (SVM) based AI algorithms have been used to separate out data points for classification, which can be leveraged to elevate the useful data out of the noise for detection. The borrowing of transformation functions from SVM, and implementation into a correlation algorithm could have a "best-of-both-worlds" approach where the benefits of kernel functions can be combined with the cross-correlations ability to template match. This could prove to be a practical algorithm with an increased ability to detect weak signals in high power, white and colored noise.

1.2 Thesis Objective and Outline

The goal of this thesis is to combine a principle of AI with a classic detection algorithm through the formulation of a kernel cross-correlation, and evaluate its performance against other methods.

First, foundational theory is established which is ultimately followed by the formulation of a kernel cross-correlation and a polynomial kernel function. During the derivation, connections to other forms of detection methods based upon the cross-correlation will be made. The algorithm will then be experimentally tested for a performance review and comparison to other techniques.

Chapter 2 is the Review of Literature section and surveys the field of signal processing related to signal detection. Basic signal and noise models are introduced, as well as a definition for weak signals is clarified. This chapter mainly focuses on building the understanding of modern signal detection philosophy and the derivation of the kernel cross-correlation algorithm.

Chapter 3 outlines experimental data sets used, and presents all experimental results in the form of detection probability densities distributions.

Chapter 4 is a discussion of the results presented in the previous chapter and draw conclusions on the performance. A comparison of the kernel cross-correlation to other detection methods is also presented.

Chapter 5 provides conclusions and final thoughts and offers further suggestions for continued research.

Chapter 2

Review of Literature and Background

2.1 Classic Detection Theory

2.1.1 Signal Models

Signals are a phenomena of a observation, a recording of some sort of a process that has occurred. Throughout this paper, the term signal is used to refer to a typically oscillatory wave that contains information that is collected by a receiver. A signal can be periodic, or that it repeats at a certain time interval T. A signal $y = f(x)$ is considered periodic if $f(x+T) = f(x)$. It is understood that for any m interval of T this rule will also hold, where m is any real integers.

$$f(x + mT) = f(x + T) = f(x), \quad \forall m = \mathbb{Z}$$

Periodic signals by implication are not unique, in that the signal at location a will be identical to signal b when $a + T = b$ [5]. Periodic signals have many applications in mathematics and engineering, however, the signals used in this thesis should be assumed non-periodic unless otherwise specified.

The standard mathematical model for an analog, continuous signal from the perspective of the receiver is described as a sine with an amplitude of A, a frequency of ω, and a phase

shift of ϕ.

$$s(t) = Asin(\omega t + \phi) \tag{2.1}$$

Signals can also be represented in complex exponential or phasor models where the natural number is raised to the signal frequency. This is common in electrical and acoustic engineering where AC electricity and acoustic pressure waves can conveniently be represented as phasors. The value of A in phasor notation can be a constant or a complex number.

$$s(t) = Ae^{j(\omega t+\phi)} \tag{2.2}$$

Both of these signal models are continuous time models, which are not realistic to applications in real-world electronics. Most electronics use a digital data acquisition system which reads data like voltage at a given sample rate. This causes our data to become discrete, therefore a discrete data model is required. Given a discrete signal s with K total samples we can say that our data then becomes a vector of data $s[k] \in \mathbb{R}^K$ where k denotes the index of the k^{th} discrete data point. The discrete signal model is given as

$$s[k] = A_k sin(2\pi f_k \frac{k}{K} + \phi_k) \tag{2.3}$$

where ω now can be described in Hertz as $2\pi f$. The amplitude and phase now must be described as the discrete vectors A_k, f_k, ϕ_k. Given they remain constant throughout the sample period the notation can then be simplified back into A, f, ϕ. Equation 2.3 is considered the sampled version of the continuous data in 2.1. The sampled time period T can be found through the division of the length of the data set K by the sample frequency f_s, or more

simply $T = f_s/K$.

Signals can be classified into two different categories: stationary and non-stationary. A stationary signal will have data that remains statistically constant in time. In other words, the statistical understandings of $A, f,$ and ϕ do not change through the sampled period of $0 \leq k \leq K$. A basic test to determine stationarity can be done by inspecting the spectral content of the data for within a small time window, say from $0 \leq k \leq K/5$, and comparing it to the spectral content later in the sampled period, $4K/5 \leq k \leq K$. For any given stationary process, the spectral content of the two windowed periods will appear the same.

The converse is true for a given non-stationary signal, more specifically, the statistical properties of a non-stationary signal will vary in time. Important examples of a non-stationary signal are swept-frequency cosine functions, otherwise known as chirps which have a changing frequency f in time. Thus arises the need to have vector descriptions of the signal amplitude, frequency, and phase and therefore A_k, f_k, ϕ_k cannot be simplified.

Most data will take the form of a non-stationary signal, because that is how information within the wave is carried. Take a radio communications for example, FM and AM radio both contain a modulation of the data with time. The modulation in the wave is what stores the useful information that is of interest to the user.

2.1.2 Noise Models

Most noise models are assumed to be additive, or they are simply values added to the pure data. There are also multiplicative noise models, however, these mostly occur in image processing, and can easily be converted into additive noise. The underlying assumptions for this thesis are that the noise is additive and is stationary in time.

Noise within the signal can be categorized in may ways, but typically comes in two

statistically different types: stochastic and deterministic noise. Stochastic noise can be divided into different types based on its spectral content. White noise is a type of Additive Gaussian Noise because its statistical properties will follow a Gaussian Probability Density centered at mean zero ($\sigma = 0$). The power of the noise is connected to the standard deviation or width of the Gaussian distribution. The larger the standard deviation, the more powerful the noise will be. White noise follows a flat power spectral density (PSD) throughout the frequency band.

Colored noise is the name given for all other forms of stochastic noise that do not present as white, particularly by inspection of the spectral content. Colored noise follows a power law distribution in the frequency domain where the order of the power law changes the type of noise. The PSD characteristically has a slope that can be used to simplify the classification of the type of colored noise [6]. The time-history waveform differs from white noise, hence the need to classify it differently.

Common types of colored noise are pink noise and Brownian noise. Pink noise appears similar to white noise, but has a slope of -10 dB/decade on a logarithmic frequency scale and has equal power in bands that are proportionally wide. In other words, every octave contains the same amount of energy, making it an ideal reference for audio engineering. Brownian or red noise, has a slope of -20 dB/decade. The power law definition for Brownian noise is the inverse of the squared frequency $1/f^2$, which is the time integral of white noise [7].

Real applications of stochastic noise appear in electronics as phenomena referred to as Johnson noise, a result of thermal motion within electronic conductors. Johnson noise was first theorized by Einstein in 1905 as Brownian motion within materials [8] and was subsequently observed by Johnson in 1928. By considering the voltage fluctuation, which is proportional to the resistance, the noise content is independent of the shape and is only linked to temperature of the resistor [9]. For electrical systems, the noise presents as volt-

age measurements, commonly characterized by mean-square voltage $\bar{V}^2$ and is related to temperature by Nyquist's law,

$$\bar{V}^2 = 4kT\mathrm{Re}(Z)\Delta f \tag{2.4}$$

where Δf is the bandwidth over which the noise is observed. Assuming the conductor is a pure resistor, then the power spectral density is independent of frequency, the noise appears approximately white [10] [11]. For this reason, white noise rejection is important in digital signal processing.

However, not all real world noise is white. For example most underwater acoustics noise is caused by living organisms. This noise is distinctly colored and changes with time as organisms move through space and time [12]. The noise created by sea life is incoherent and therefore is categorized as stochastic, rather than deterministic. The noise falls somewhere in between the Brownian and white noise ranges making Brownian noise a good approximation for simulating real-world, "worst-case" acoustic noise [2]. Colored noise can obstruct detection techniques and significantly worsen the performance of detection algorithms more-so than white noise. For this reason, signal detection with additive colored noise is of particular interest.

Deterministic noise is defined as noise with similar mathematical characteristics to the signal of interest but does not contain the desired data from the signal. Because it appears similar to the signal of interest, it can be difficult to discern the difference between deterministic noise and useful data. This form of noise can be introduced to the data in a number of ways, such as background dialog during audio recordings, or electromagnetic leakage. Typically, this type of noise can be filtered out, however, when the noise appears within the same frequency band as the signal of interest, it becomes challenging to remove.

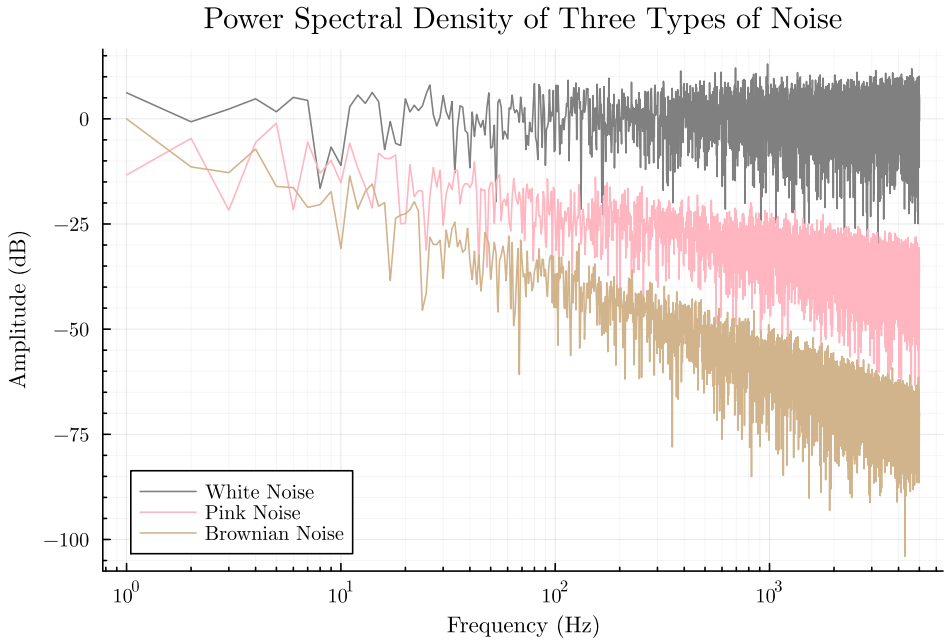


Figure 2.1: The PSD of three types of noise. White noise is identifiable by a 0 slope on a logarithmic x-axis, where Brownian noise is identifiable by a -20 dB/decade slope, and pink noise contains a slope of -10 dB/decade slope.

2.1.3 Signal-to-Noise Ratio & Weak Signals

Signal-to-Noise Ratio (SNR) is a widely used measurement for determining the quality of a signal. There are many definitions based on the type of signal being processed, however the most common is the ratio of powers definition.

$$SNR = \frac{P_s}{P_n} \tag{2.5}$$

Where P_s is the power of the signal and P_n is the power of the noise. The signal power over a time period T with K samples can be defined as the absolute squares of the signal normalized by the signal length [13].

$$P = \frac{1}{K}\sum_{k=1}^{K} A^2[k] \tag{2.6}$$

where $A^2[k]$ denotes the amplitude of the k^{th} discrete data point of the signal. Which can be expressed as an absolute value of the amplitude squared $\mathbb{E}[A^2[k]]$. The power of the noise can be found by subtracting out the signal from the entire data set and then using equation 2.6. For measuring data without a signal present, the noise power is the preferred metric.

It is common practice to define SNR in terms of decibels because of the broad scales that the ratio can take [14]. The value of the SNR in dB is found by taking the logarithm of base 10 of the ratio of powers and multiplying by 10 [15].

$$SNR = 10log(\frac{P_s}{P_n}) \tag{2.7}$$

The decibel definition of the SNR approaches 0 dB as the value of the ratio of signal to noise power approaches unity. Because of the definition of signal power, signals with short periods or small amplitudes can have relatively low SNR values.

An alternate definition of SNR can be found through a more statistical approach which uses the variance of the data set as the denominator in the ratio. Because power can be expressed as an expected value definition, we can break the data down into signal and noise as $A^2[k] \simeq S^2[k] + N^2[k]$. Then the expected value of the amplitude squared of the noise can be written as the sample variance σ^2. SNR can then be expressed as a ratio of the signal power to the sample variance [16].

$$SNR = 10log(\frac{P_s}{\sigma^2})$$

Most electronic applications require high SNR, for example wireless networks recommend a SNR upwards of 25 dB for streaming any sort of audio and video [17]. However the benchmark for a weak signal is far below that at around the 0dB point [3]. In the context of this thesis, only signals with negative decibel SNR values will be explored.

2.1.4 Window Functions

Given discrete data collected is of a finite length of K data points which can be used to construct the observed signal. Practically, there will be some element of noise which can corrupt the data particularly when finding the discrete Fourier transform (DFT) of the data. The application of windows to sampled data can be used to smooth out the spectral content and reduce leakage when computing the DFT. Windows are weighting functions applied to discrete data in order to diminish the effects of edge discontinuities on the spectral content of the data [18].

There are many different types of window functions which are quite common to apply to discrete data. The equations for the windows and their effect on the spectral content of the data will be discussed further.

Rectangular Window

A rectangular window is simply a weighting of unity across all K data points in the vector. In essence, it is a gating sequence applied to the data to control the edge behaviour and ensure a finite length. A mathematical description for the rectangular window function is

$$w(k) = 1, \quad k = 0, 1, ..., K-1 \tag{2.8}$$

The transform of this window is a Dirichlet which has a DFT main-lobe, and sidelobes which peak at -13 dB and fall off at -6 dB per octave. Because of the nature of the even weighting and its definition as a sum of an infinite sum, the DFT of the rectangular window cannot capture all terms leading to inaccuracies near the edges of the rectangular window. These inaccuracies are oscillations called Gibbs phenomenon. This oscillatory behaviour is what windowing functions seek to suppress. For that reason, it is desired to explore other windows.

Triangular Window

A triangular window is another simple window like the rectangular window, but has a maximum weighting at the center and a weighting of 0 at the edges. For a window centered at $k = 0$ with a length of K, the triangular window can be defined as

$$w(k) = 1.0 - \frac{|k|}{K/2}, \quad k = -\frac{K}{2}, ..., -1, 0, 1, ...\frac{K}{2} \tag{2.9}$$

The transform of this window is the squared Dirichlet kernel which has a main-lobe and sidelobes which are twice the width of the rectangular window. The sidelobes are approximately -26 dB and fall off at twice the rate per octave as the rectangular window.

$\cos^{\alpha}(x)$ *Windows*

Cosine windows are a family of windows which are dependent on an alpha parameter and are used for cosine smoothing of the data. This window function is relatively easy to compute because of its dependence on a simple cosine function. The cosine window for any value of α can be defined as

$$w(k) \;=\; \cos^{\alpha}\left[\frac{k}{K}\pi\right], \quad k = -\frac{K}{2}, ..., -1, 0, 1, ...\frac{K}{2} \tag{2.10}$$

and for data where the DFT will be performed, must be in a DFT even distribution, starting at $k = 0$. In this case it can be expressed as a sine relationship

$$w(k) \;=\; \sin^{\alpha}\left[\frac{k}{K}\pi\right], \quad k = 0, 1, ..., K-1$$

The most common versions of the cosine window have α values of 1 to 4, with $\alpha = 2$ being the most important of the family. This is known as the Hann window and was named after Austrian meteorologist Julius von Hann.

$$w(k) \;=\; \sin^{2}\left[\frac{k}{K}\pi\right] \;=\; 0.5\left[1.0 - \cos\left[\frac{2k}{K}\pi\right]\right], \quad k = 0, 1, ..., K-1 \tag{2.11}$$

The Hann window functions are much better than the rectangular and triangular at rejecting problematic frequencies at the edges of the data. The larger the value of α the larger the difference between the main lobe and the initial side lobes will be.

Constructed Windows: Poisson

A Poisson window is a two sided exponential window that is constructed through a piece-wise equation rather than a single function. This is also a family of functions like the $\cos^{\alpha}(x)$ functions which varies based on a parameter of α. The Poisson window can be described by the following relation

$$w(k) \;=\; \begin{cases} \exp\left[-\alpha\frac{|k|}{K/2}\right] & 0 \le k \le \frac{K}{2} \\ \exp\left[\alpha\frac{|k|}{K/2}\right] & -\frac{K}{2} \le k \le 0 \end{cases} \tag{2.12}$$

where there is a discontinuity at $k = 0$. This causes the main lobe of the window in the

frequency domain to be much wider than something like the Hann window or the triangle window. This also has implications on the fall-off of the side-lobes as it is much slower than the Hann window.

Different types of windows can be used for different types of signal content. For example, a Hann window would be preferable for data with sine waves or a combination of sine waves. Whereas a response signal may benefit from an exponential-constructed window like the Poisson window function.

2.1.5 Correlation Algorithms

The correlation is a classic form of feature detection algorithm in discrete signal processing and is often a component of a more complicated techniques. Correlations are a moving average process which takes two classic forms in the autocorrelation and the cross-correlation. Broadly, we are able to define the autocorrelation function as the auto-covariance of the signal divided by the sample variance [16]. Autocorrelations measure the correlation between a data set and a delayed copy of itself as a function of the delay [19]. The delay is commonly known as the lags τ. Moreover, the result of the autocorrelation will show the similarity of random variables and any repeated patterns in the data set. The autocorrelation can be represented as the convolution of the data with its time-reversed version. The autocorrelation for a discrete data set $x[k] \in \mathbb{R}^K$ is not the true autocorrelation R_{xx}, but instead is an estimate of the autocorrelation $\hat{R}_{xx}$, and can be defined as

$$\hat{R}_{xx}(\tau) = \frac{1}{K}\sum_{k=1}^{K}(x[k]-\mu)(x[k+\tau]-\mu), \;\; -K \leq \tau \leq K \tag{2.13}$$

which is equivalent to the expected value of the signal with the time delayed version of itself.

$$\hat{R}_{xx}(\tau) = \mathbb{E}[(x[k] - \mu)(x[k + \tau] - \mu)]$$

Convolution in the time domain may be represented as multiplication in the frequency domain by the convolution theorem [18]. This vastly speeds up computation time having an $\mathcal{O}(Klog(K))$ time dependency versus $\mathcal{O}(K^2)$ in the time domain. Thus, the autocorrelation can be represented in the frequency domain as a product of two vectors.

$$\hat{R}_{xx}(\tau) = F^{-1}\{X[\omega] \odot X^*[\omega]\} \tag{2.14}$$

where $X[\omega]$ is the frequency domain representation of $x[k]$ and the complex conjugate is denoted by the star.

$\hat{R}_{xx}(\tau)$ is equivalent to the expected value of the power of the signal. The above definition in equation 2.14 is the non-normalized version of the autocorrelation. Normalization in the frequency domain is difficult to represent and often increases complexity.

Assuming that our data contains stochastic noise, the autocorrelation can be used to reject noise from the data. The larger the number of data points K the more accurate the autocorrelation will be and the better the noise rejection. This takes advantage of random content's tendency to trend to an expected value of 0 as the set becomes infinitely large.

Where the autocorrelation uses a time delayed version of the original signal, the cross-correlation uses a time delayed reference signal $y[k] \in \mathbb{R}^K$. The cross-correlation is used to check for similarities to $y[k]$ within $x[k]$.

$$\hat{R}_{xy}(\tau) = \frac{1}{K}\sum_{k=1}^{K}(x[k] - \mu)(y[k + \tau] - \mu), \quad -K \leq \tau \leq K \tag{2.15}$$

Similarly to the autocorrelation, the cross-correlation for the finite data set is an estimator for the continuous-infinite cross-correlation. The cross-correlation $R_{xy}(\tau = 0)$ will be the same as the expected value of the convolution of the two signals reversed. Similarly to the autocorrelation, the cross-correlation can be represented as an expected value of the two data sets [20].

$$\hat{R}_{xy}(\tau) = \mathbb{E}[(x[k] - \mu)(y[k + \tau] - \mu)]$$

The cross-correlation also has a frequency domain representation as was proved for equation 2.14. This can be found by performing element-wise multiplication of the Fourier transform of the original data and the complex conjugate of the frequency domain of the reference signal [16][18].

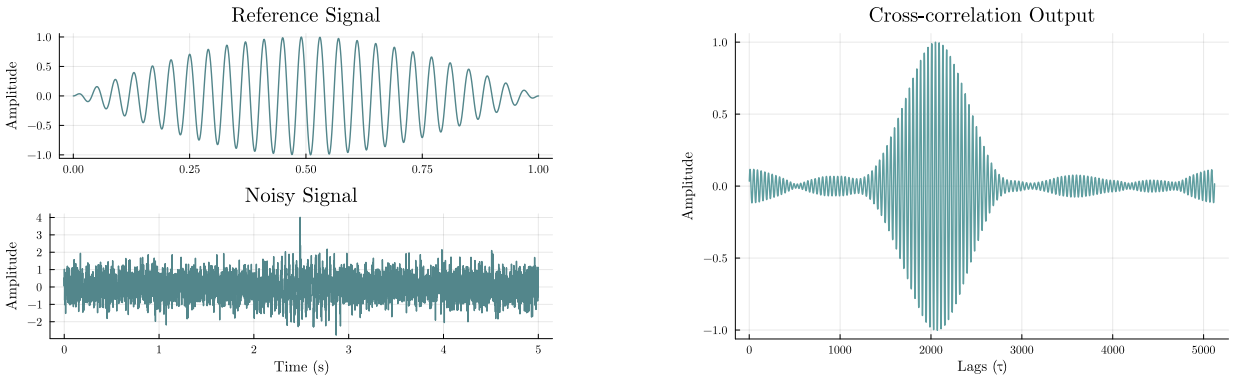
$$\hat{R}_{xy}(\tau) = F^{-1}\{X[\omega] \odot Y^*[\omega]\} \tag{2.16}$$

In equation 2.16, F^{-1} denotes the inverse Fast Fourier Transform (FFT) operator, $Y^*[\omega]$ denotes the complex conjugate of $y[k]$ in the frequency domain, and $\odot$ represents element wise multiplication. The use of this principle can be used to drastically speed up calculations for correlation functions of large data sets. In the frequency domain, the XCF follows $\mathcal{O}(Klog(K))$ complexity as the samples size increases, the same as the autocorrelation. The normalized version of the correlation in the frequency domain takes a much more complicated form and is difficult to calculate because of the transformation of the normalization term into the frequency domain.

The time domain definition of the cross-correlation will yield an output of length $[2 * K - 1]$. The frequency domain definition will have a length of K because of the element

wise definition of multiplication. Therefore, in both domains, the signals $x[k]$ and $y[k]$ must contain the same number of discrete data points.

The benefit of the cross correlation function is its usage in template matching within data. By checking against a reference, the local maxima within the resultant correlation will occur at index j wherein the original signal $x[k]$ best matches the selected reference set $y[k]$.



(a) Original signal and reference signal

(b) Cross-correlation output

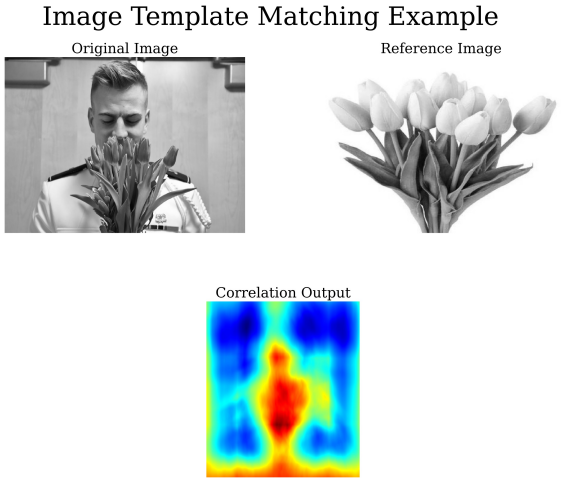
Figure 2.2: Simple example case of a cross-correlation of a sinusoidal signal with white noise. The correlation local maxima occur at where the beginning of the reference pulse lines up with the beginning of the sine pulse. In this case, that location would be at lag $\tau = 2.0 \times 10^3$.

Cross-correlations have been adapted into other formats, particularly for array noise rejection, known as a cross-channel correlation algorithm. The cross-channel correlation can observe similarities between p number of channels and remove redundant information. The correlated data is assumed to be noise and the remaining parts of the signal are considered "new" information [21]. Multi-channel cross correlations have found use in beam forming applications [22].

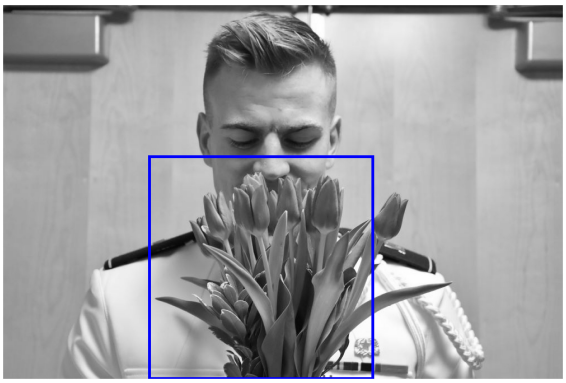
The cross-correlation's ability to detect patterns in data are not just used in a signal time-history context. The sizes of x and y can be any arbitrary size [k x m] so long as both have the same size. This is another benefit of the frequency domain representation. Therefore, the correlation can be used to calculate similarities between two images, known

as template matching.

This style of cross-correlation is mathematically very similar to the single dimensional case. For this application, the normalized version of the cross-correlation is more accurate and thus preferred. In this case the use of the frequency domain convolution theorem is still used, however the normalization term of is reduced to an integral of search area image and image squared approach. The computation requirements for the image correlations thus evolve at a rate of $\mathcal{O}(3K^2)$ as opposed to $\mathcal{O}(3K^2(M-K+1)^2)$ [23]. Modern applications of the cross-correlation include face verification algorithms and feature detection in computer vision systems [24] [25] [26].



(a) Original image, reference image used for template matching, and cross-correlation output



(b) Original Image with box placed around location where template matching identified presence of the reference image

Figure 2.3: A simple example case of an image cross-correlation example for template matching able to detect the bouquet of tulips within the original photo using fast normalized cross-correlator proposed by Lewis [23].

2.1.6 Correlations as Building Blocks

The correlation algorithm can be used as the building blocks for other signal processing algorithms. The power spectral density (PSD), or autospectrum, is defined as the Fourier Transform of the autocorrelation by the Wiener-Khinchin theorem. If the discrete Fast Fourier Transform for some dataset $x[k] \in \mathbb{R}^K$ is defined as

$$S(\omega) = \sum_{k=0}^{K-1} x[k]\ e^{-i\omega tk/K}$$

then the discrete-time PSD of $x[k]$ can be written as

$$\hat{S}_{xx}(\omega) = \sum_{k=0}^{K-1} \hat{R}_{xx}(\tau) e^{-i\omega tk/K} \tag{2.17}$$

where $\hat{R}_{xx}(\tau)$ is the definition of the autocorrelation as given by equation 2.13. By expansion of the definition of the autocorrelation function $\hat{R}_{xx}(\tau)$, the formula for the PSD becomes, the Fourier transform of the convolution of the signal with the time reversed version of itself.

$$\hat{S}_{xx}(\omega) = \sum_{k=0}^{K-1} \sum_{k=1}^{K} (x[k] - \mu)(x[k+\tau] - \mu) e^{-i\omega tk/K}$$

$$\hat{S}_{xx}(\omega) = \frac{1}{K} \sum_{k=0}^{K-1} \mathbb{E}[(x[k] - \mu)(x[k+\tau] - \mu)] e^{-i\omega tk/K}$$

Applying the convolution theorem [18] to the definition of the autocorrelation, $\mathbb{E}[(x[k] - \mu)(x[k+\tau] - \mu)]$ can be expressed as the inverse Fourier Transform of the multiplication of the frequency domain expression of the data set $X[\omega]$ and its complex conjugate.

$$\hat{S}_{xx}(\omega) = \sum_{k=0}^{K-1} \sum_{k=0}^{K-1} [X[\omega] \odot X^*[\omega]] \; e^{i\omega tk/K} e^{-i\omega tk/K}$$

The Fourier transform and the inverse Fourier Transform can be cancelled and the PSD takes the form of the absolute value of the element wise multiplication in the frequency domain [20].

$$\hat{S}_{xx}(\omega) = |X[\omega] \odot X^*[\omega]| \tag{2.18}$$

Mathematically the autospectrum is very similar to the frequency domain definition of the autocorrelation, the difference being the expression is left in the frequency domain.

The use of the autospectrum is widespread and has applications in neurophysics research as well as tracking climate patterns [27] [28]. In terms of signal detection, the PSD can help identify the most powerful frequency components within a noisy signal. This has limitations and typically requires a high SNR to work properly.

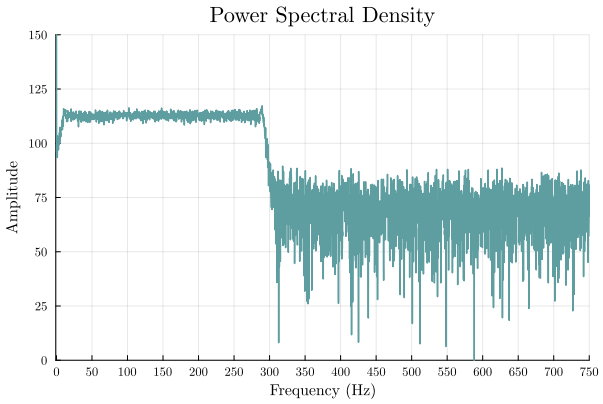
In terms of signal detection, for a given data set, if a signal is present the PSD should reveal the more powerful frequency content which can be used to confirm the presence of useful data. However, it distinctly lacks any temporal resolution, and does not reveal any information regarding the stationarity of the data without some sort of sliding window.

The cross-spectrum is similar to the autospectrum, in that it is the frequency domain decomposition of the cross-correlation [20]. The cross-spectral density of a signal can be found in a similar manner to the cross-correlation definition in equation 2.16, and is defined as absolute value of the data in frequency domain, times the complex conjugate of the reference [29]. Like the cross-correlation, the cross-spectrum evolves at an $\mathcal{O}(Klog(K))$ time complexity.

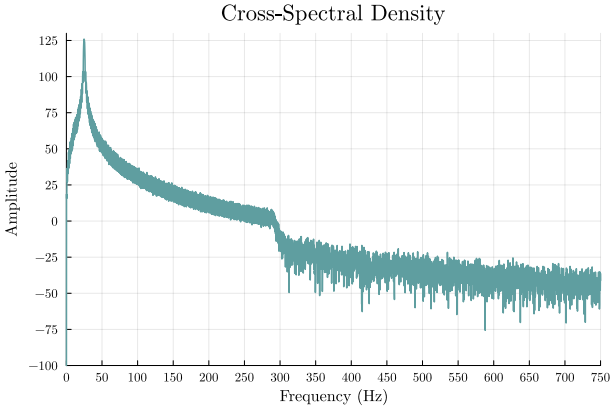
$$\hat{S}_{xy}(\omega) = |X[\omega] \odot Y^*[\omega]| \tag{2.19}$$

The cross-spectrum is used to show the power of the shared frequency content between two signals and is commonly used to identify the frequencies present of data in a system. The cross-spectrum was created as an alternative to other frequency decomposition algorithms as the cross-spectrum is able to accurately measure the frequencies of short time, low signal-to-noise ratio (SNR) signals. Cross-spectral analysis is also quick comparative to other frequency decomposition methods of signals [30].

The cross-spectrum also distinctly lacks any sort of temporal resolution when analyzing signals. For that reason, both the PSD and the cross-spectral density are only able to determine presence of data with specific frequency content and not its location in time.



(a) Power Spectral Density of a linear swept frequency cosine (chirp) from 10 to 300 Hz



(b) Cross-spectral density of a chirp signal in noise using a sine wave of 25 Hz as reference

Figure 2.4: Power spectral density of a chirp from 10-300 Hz in (a) has a sharp drop off at 300 Hz indicating a signal is present in the data. The cross-spectrum shown in (b) where the data from (a) is checked against a sine wave of 25 Hz for the presence of shared frequencies. The peak appears at the expected shared frequency strongly indicating presence of a deterministic signal.

2.1.7 Matched Filtering

Filtering data is a common method to reduce noise present in a signal. The standard linear filter is the convolution of the data with a windowing function that can reject redundant frequencies that do not carry data. In short, filtering acts as a partial suppression of unwanted components of a signal.

Digital filters can be used for signal detection, but common types like Butterworth and Chebyshev filters are more optimized for noise rejection. A matched filter is a form digital filters used for optimizing the SNR of a signal in the presence of stochastic additive noise. Matched filters are commonly used in radar and digital communications for the special purpose of signal detection.

A comprehensive derivation of the matched filter can be found using the Schwarz's inequality to maximize output signal energy to noise energy in received signal [31]. Assuming the noise is stationary, the theorem holds and any signal $s(t)$ with additive noise $n(t)$ has a filter that is matched to its impulse response:

$$h(t) = g(T - t) \tag{2.20}$$

where g is an arbitrary constant and T is the filter output time. Equation 2.20 implies the maximum SNR is achieved when this filter has an impulse response that is the same as the complex conjugate, time-reversed input signal [32]. Therefore it is common to compute a matched filter as a digital, finite impulse response (FIR) filter with the time reversed signal of interest or reference signal as the coefficients.

The transfer function of the matched filter corresponds to the complex conjugate of the spectrum of the received signal $s(t)$ of interest [33].

$$H(\omega) = gS^*(\omega)e^{-j\omega T} \tag{2.21}$$

Note the similarity to the definition of the cross-correlation in the frequency domain. Both consist of some variant of the reference signal that has been time-manipulated convolved with the pure data. In the matched filter's case, it is the FIR filter created by the time reversed reference. Both the correlation and the matched filter are categorized as ideal receivers.

2.1.8 Time-Frequency Distributions

Often, data may change with time, these are known as non-stationary signals. Standard Fourier analysis like PSD may be able to describe the frequency versus intensity relationship of the data, however it fails to relate to variations in time domain. Time frequency distributions (TFD) are a powerful signal processing technique used to analyze the spectral content of a given data set by decomposing it into time, frequency, and intensity [34] [16]. Thus allowing relationships to be made between the three.

The most simple of the time frequency distribution algorithms is the is the spectrogram, which makes use of the Short-Time-Fourier-Transform (STFT). The STFT slides a window over the data, and subsequently calculates the strongest frequencies within that time window to determine a frequency relationship with time [35] [36]. Given a data set $x[k]$ with K data points, a window of size m where $m < K$ is taken. The data is then convolved with a windowing function $h[k-m]$, and the Fourier transform of the data is taken. It was seen previously that the windowing function will have an effect on the frequency content of the data [18]. Ideally, for better temporal resolution, the size of the window should be much smaller than K, such that it captures any non-stationary behaviour of the data accurately.

Overlap between windows will determine resolution in the frequency axis.

$$\mathrm{STFT}(m,\omega) = \frac{1}{\sqrt{2\pi}} \sum_{k=-\infty}^{\infty} x[k]h[k-m]e^{-j\omega t} \tag{2.22}$$

the window function $h[k-m]$ is assumed to be discretized to match the discrete definition of the STFT. There are many choices of windowing function that are used for different reasons [37] [18].

The STFT of the data can be used to compute the spectrogram of the data, which shows the PSD of the data versus time. This can be done by simply squaring the STFT.

$$S(m,\omega) = STFT^2(m,\omega) = [\frac{1}{\sqrt{2\pi}} \sum_{k=-\infty}^{\infty} x[k]h[k-m]e^{-j\omega t}]^2 \tag{2.23}$$

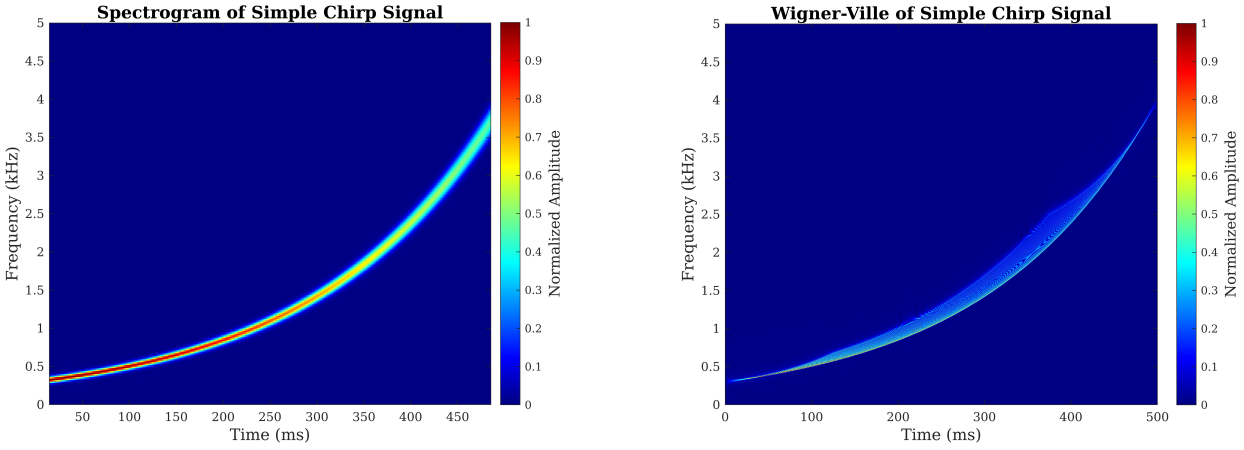
Ultimately the windowing is the largest flaw of the spectrogram as it is unable to capture signals with short periods that occur within one time window [36]. When this occurs the signal may may be drowned out by surrounding spectra which does persist over multiple windows. Window size must strike a balance between time and frequency resolution, therefore tuning of the window size is needed.

Where the spectrogram fails, the Wigner-Ville distribution succeeds in calculating high resolution time-frequency distributions. The Wigner-Ville distribution is a modern TFD that is used to estimate signals with incredibly short time periods and instantaneous frequency estimations [38]. The Wigner-Ville distribution generalizes a relationship to the PSD through the use of autocorrelation [37] [39]. In its discrete form, the Wigner-Ville distribution can be defined by equation 2.24 [40] [41].

$$W_{xx}(k,\omega) = \sum_{m=-K}^{K} x[k+m/2]x^*[k-m/2]e^{-im\omega/K} \tag{2.24}$$

For signal detection, and instantaneous frequency estimation of non-stationary signals, the cross Wigner-Ville distribution may also be used. Similarly to the cross-spectral density of a signal, the cross Wigner-Ville is the TFD of the cross-correlation of the data. It is useful due to its ability to handle noise at much lower SNR values [42].

$$W_{xy}(k,\omega) = \sum_{m=-K}^{K} x[k+m/2]y^*[k-m/2]e^{-im\omega/K} \tag{2.25}$$



(a) Spectrogram (b) Wigner-Ville distribution

Figure 2.5: A simple example case of a spectrogram (a) and Wigner-Ville distribution (b) of a logarithmic chirp from 300 Hz to 4 kHz. The width of the spectrogram appears much thicker than the width of the Wigner-Ville distribution, and will detect shorter period signals because of the higher resolution scale.

2.2 Support Vector Machines

Support Vector Machines are a type of supervised machine learning used for classification of data problems. The principles are based on solid and well understood optimization theory and mathematics, making SVM one of the most widely used types of machine learning. SVM has applications in fraud detection, speech and handwriting recognition, and wireless communications [43][44].

The goal of SVM is to create a boundary in a higher dimension space between data points that otherwise would not be linearly separable. Discrete data with N dimensions and K data points will be transformed through algorithms, known as kernels, into additional dimensions called a feature space [45]. For data $x[k] \in \mathbb{R}^{K\times N}$ the kernel transform of the data will add an additional dimension where $\phi_t(x[k]) \in \mathbb{R}^{K\times M}$ such that $M > N$. The feature space projection of the data in the M dimension is the convolution of the data $x[k]$ with the kernel function $\phi_t(x[k])$. This is known as the "Kernel Trick" in data science [46].

Considering two linearly-independent data sets $A, B \in \mathbb{R}^{K\times N}$ that do not intersect and are not linearly separable, the data may be convolved with any kernel function to create a projection into an M dimensional higher space. For each data point, there exists a location on the lower dimensional space, and will now be assigned a new coordinate on the higher dimension space. Our data sets A and B now exist in the vector space $\mathbb{R}^{K\times M}$. Data sets $A[k,n]$ and $B[k,n]$ may now become,

$$A[k,n] = [A[k]_1, A[k]_2, ...A[k]_N] \rightarrow \phi_t(A[k,m]) = [A[k]_1, A[k]_2, ...A[k]_N, \phi_t(A[k,n])_M]$$

and,

$$B[k,n] = [B[k]_1, B[k]_2, ...B[k]_N] \rightarrow \phi_t(B[k,m]) = [B[k]_1, B[k]_2, ...B[k]_N, \phi_t(B[k,n])_M]$$

Notice how the data from dimensions $1 \rightarrow N$ remains the same, however an additional dimension is added when the transformation is taken.

For a given two-dimensional data set, say an acoustic recording waveform, $p[\tau]$ it can be plotted as a two-dimensional data set where pressure depends on discrete time variable τ. The kernel transformation of this data adds an additional dimension which is the convolution of ϕ_t and p, and the two-dimensional data set will become a three-dimensional data set $\langle \tau, p, \phi_t \rangle$, and can be plotted as such.

The data in the feature space projection, may then be linearly separated by a plane where $\Phi_t(A[k,n])$ is above the plane and $\Phi_t(B[k,n])$ lies below. This plane is called a linear hyperplane and is the decision boundary for the data, provided that the optimal kernel has been chosen. Data on one side of the plane belongs to one class, and above to another class [47]. Hyperplane equation is generated through a supervised training process where regression is used to find the optimal margin between the data points.

Given j training sets in our training data $T = (T_1, T_2, ...T_j)$, that exist in the dimension space N, the traditional hyperplane is

$$\langle \beta, T \rangle + c = 0 \tag{2.26}$$

where β is the normal vector to the hyperplane, $T \in \mathbb{R}^M$, c is the scalar offset of the hyperplane. The M dimension space is a higher feature space projection of the data within the N dimension space that our training data set T existed in. The optimal solution to the hyperplane can be found by maximizing the distance between the two closest points. This can be done by minimizing the normal vectors, and introducing an error term [48] [49].

$$\min \frac{|\beta|^2}{2}\kappa(\sum_{i=1}^{m}\tau_i) \tag{2.27}$$

κ is a regularity parameter, and balances the maximization of the distance, and the classification error "slack variable" τ.

Alike many of the other applications, SVM has shown promise in the detection of signals in noise. The use of SVM in signal detection translates the detection problem to a classification problem, *i.e.* noise vs. signal [50]. However, for the creation of the algorithm that will be done in the formulation section the training of a hyperplane will not be used. Instead, the concept of most importance from SVM is the kernel function, for that reason kernel functions will be the focus for the remainder of this section.

Transformation is highly dependent on the kernel function that is chosen. There are many different kernel functions used for different applications. Kernel combinations are also valid which may be customized for a unique application. Transformations may be linear or non-linear in nature.

$$\phi_t(x[k], y[k]) = exp\{\frac{-(x[k]^2 + y[k]^2)}{2\sigma}\} \tag{2.28}$$

$$\phi_t(x[k], y[k]) = |x[k] + y[k]| \tag{2.29}$$

$$\phi_t(x[k], y[k]) = (x[k] \times y[k] + 1)^3 \tag{2.30}$$

The three kernel transformation functions above are the Radial Basis Function (RBF), linear absolute value (L-Abs), and cubic respectively. The L-Abs kernel is the only transformation function of the three that is linear. The kernels above are functions of two discrete variables being $x[k]$ and $y[k]$, but this can vary based on dimension of the data. The RBF

kernel introduces an additional hyperparameter σ which controls the width of the Gaussian distribution. The Width parameter σ has inverse effect on the distance between points in the feature space. Additionally, the cubic kernel is a variation of a polynomial kernel which has a specific degree of 3.

The ability to separate otherwise non-linearly separable data, lends kernel transforms the ability to have discriminatory power against stochastic components in the signal. Given a signal corrupted by noise, transformations of the data can artificially enlarge the SNR. This is predicated on the assumption that the noise is mean-zero and additive in nature.

Given implementation into established signal processing techniques, the transformation power of kernel functions to increase SNR could increase detection probability in noisy data. For that reason, the design and validation of a kernel function based cross-correlation is of particular interest.

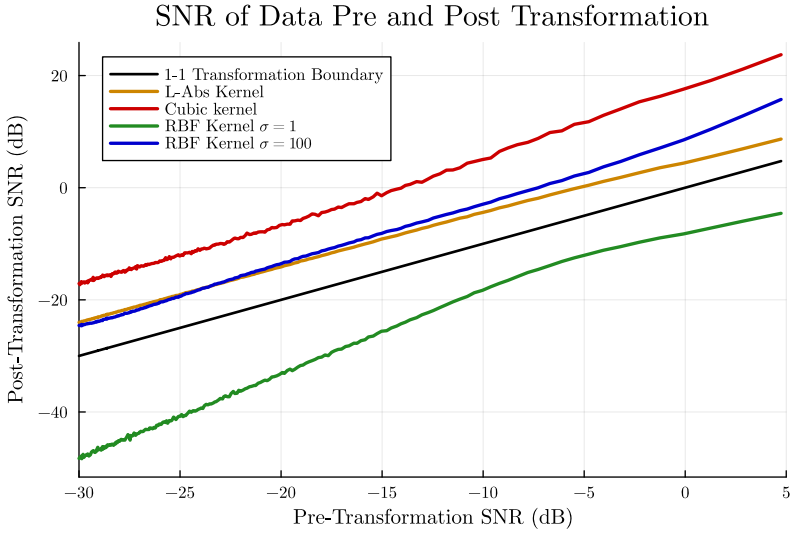


Figure 2.6: A graphical analysis comparing the original SNR of the data set to SNR after kernel transformation. The three functions examined are the cubic kernel, L-Abs kernel, and RBF kernel with varying width parameters. The 1 to 1 transformation boundary line, meaning if a kernel lies above this line it is considered to have a positive SNR boost when transforming data. The Gaussian (RBF) kernel with $\sigma = 1$ has a negative boost for SNR where the larger sigma has a positive boost similar to the L-Abs kernel.

2.3 Formulation

In order to evaluate the efficacy of the algorithm a derivation must first be presented. This section will review the derivation of the kernel cross-correlation (KCC) algorithm [51] and propose a special polynomial kernel function purpose-built for weak signal detection.

2.3.1 Formulation of KCC

In order to order to derive the KCC, the frequency domain definition of the cross-correlation must first be considered. Given two data sets described by $x(t)$ and $y(t)$ in the time domain, their cross-correlation can be found by equation 2.16.

$$\hat{R}_{xy}(\tau) = F^{-1}\{X[\omega] \odot Y^*[\omega]\} \tag{2.16}$$

The first term can become the time domain kernel transform of the data in the frequency domain, denoted by $\Phi_t(X)$. The same can be done to the reference signal term, becoming $\Phi_t(Y)$. The complex conjugate of this value is still used in the correlator. The equation then becomes

$$\hat{R}_{xy}(\tau) = F^{-1}\{\Phi_t(X) \odot \Phi_t^*(Y)\} \tag{2.31}$$

which is the simplest definition of the KCC. This is simply the element-wise multiplication of the time-domain kernel transformed data in the frequency domain with the complex conjugate of the time-domain kernel transformation of the reference in the frequency domain. To reiterate, the size of the data elements must both match because of the element wise multiplication present.

One key feature of the KCC is the implementation of an output mapping selection designated as $G(z)$. $G(z)$ may have up to p mapping sets of discrete data therefore making $G = [z_1, z_2, ...z_p]^T$. To implement $G(z)$ we must redefine the term for the reference signal as the correlator term $H(Y)$ and the equation for the KCC becomes

$$\hat{R}_{xy}(\tau) = F^{-1}\{\Phi_t(X) \odot H^*(Y)\} \tag{2.32}$$

The introduction of $H(Y)$ as the correlator term now rises a training problem where the selection of $G(z)$ is meant to minimize an error function. The optimum value will be centered at the global minima of the sum of squared errors (SSE) between the desired output and the actual output. The sum of squared errors is selected to be minimized because of the quadratic nature and its single minimum value. This guarantees that an optimum value of $G(z)$ can be found.

$$\frac{H(Y)}{\Phi_t(Y)} - G(z_p) = e \rightarrow \sum_{k=1}^{K}\{\frac{H(Y_k)}{\Phi_t(Y_k)} - G(z_{p_k})\}^2$$

The optimum value of $H(Y)$ can then be found at the minima of the SSE equation by taking the derivative.

$$\frac{d}{dH(Y)}\{\frac{H(Y_k)}{\Phi_t(Y_k)} - G(z_{p_k})\}^2 = 0$$

Which can then be simplified and solved for the final expression for $H(Y)$, which yields equation 2.34.

$$\hat{H}(Y) = G(z_p) \odot \Phi_t(Y) \tag{2.33}$$

This is our correlator which is used to check the data for patterns based on the reference

signal $y[k]$. There is no restriction to size of our reference signal and $G(z_p)$, however, $H(Y)$ must be the same size as the data $x[k]$. For a vector-time history signal, $H(Y)$ must be of size K. Our final equation for the cross-correlator then becomes the combination of equations 2.32 and 2.34.

$$R_{xy}(\tau) = F^{-1}\{\Phi_t(X) \odot [\hat{G}(z_p) \odot \Phi_t(Y)]^*\} \tag{2.34}$$

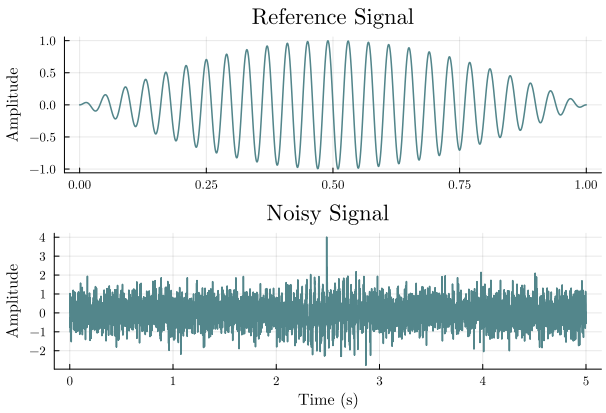
Examining $G(z_p)$ more closely, the mapping selection will be convolved with the kernel transformation of the reference signal. The selection for $G(z_p)$ is entirely arbitrary and may be selected based on user selection. The purpose for the mapping term is to control the output to make the output more predictable for automation. Mapping can also be used to simplify the threshold process for event detection in a sensor. For computer vision tasks, a mapping set can be selected to smooth the image first to help increase accuracy of the correlation. The mapping selection should be an iterative and supervised training process and is dependent on problem set.

Selections for $G(z_p)$ could be impulses, Bessel functions, sombrero hat distributions, triangular distributions, and Gaussians. Impulse functions within the frequency domain introduce energy into all frequencies, essentially creating an "unmapped" version of $H(Y)$ and will resemble 2.31. Some mapping may act as filters to the output, as they have only low frequency content which removes high frequencies present in the output. In essence, a window function could be selected as the mapping set in the correlator to allow for windowing of the data in a single step.

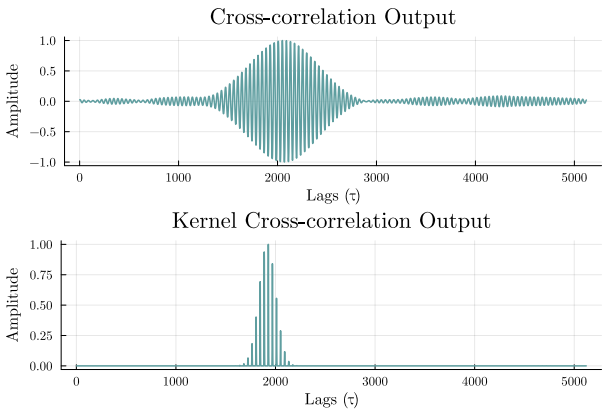
For a multi-kernel case, we can extend the kernel transformation $\Phi_t(x[k], x[k])$ may have up to M different kernels. The multi-kernel case may then become $\Phi_t(x[k], x[k]) = [\phi_1(x[k], x[k]), \phi_2(x[k], x[k]), ...\phi_M(x[k], x[k])]^T$ where all terms become size $[M \times K]$. Using

J different reference signals follows suit where the terms will all be of size $[J \times K]$. This allows for $J \times M$ multiple different correlations to be done successively, and the output will be $[(J^*M) \times K]$.

The classic correlation can be compared to the output of the kernel cross-correlation using the L-Abs kernel. Both correlation peaks will occur in the same location. At this point, the beginning of the reference signal is aligned with the beginning of the signal, and a maximum correlation is returned. In both correlations, the peak value of the output is used to communicate the location where the highest similarity between the data sets was detected.



(a) Original signal and reference data

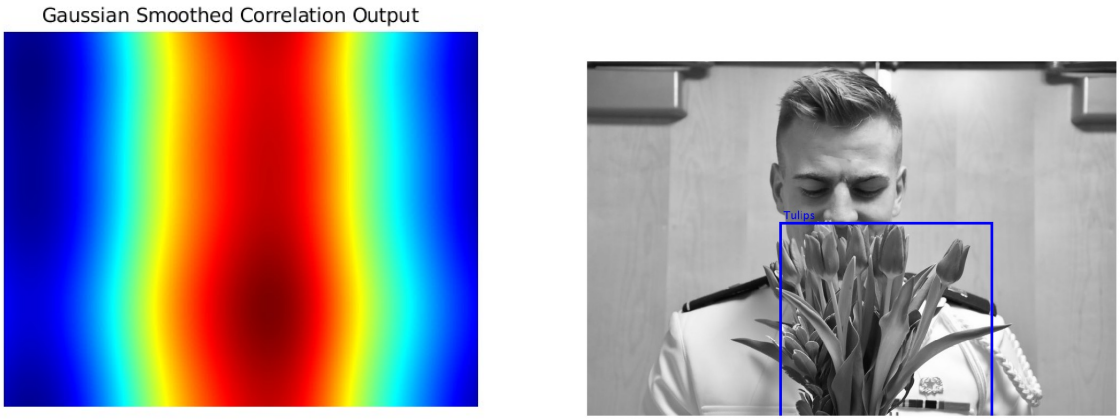


(b) Correlation output and kernel cross-correlation output

Figure 2.7: Comparison of a traditional cross-correlation to the kernelized version for the two data sets given in (a). The original signal contains a sine wavelet, beginning at 2 seconds, and additive white Gaussian noise.

The benefit of the frequency domain definition of the kernel cross-correlation is the ability to scale up the datasets $x[k]$ and $y[k]$ from a vector form $x[k], y[k] \in \mathbb{R}^K$ into a matrix form $x[m,k], y[m,k] \in \mathbb{R}^{M \times K}$. This flexibility allows for fast template matching in images as proposed by Wang et al. [51]. The canonical, non-normalized version of the cross-correlation lacks the accuracy needed for template matching and therefore the normalized

version proposed by Lewis [23] is used as shown in figure 2.3. The introduction of image kernel transformations and the mapping set $G(z_p)$, greatly increases template matching accuracy in the non-normalized version of the correlation. Thus eliminating the need to compute the integral of the image and image power of the original image. Gaussians, which are commonly used to smooth images, can be easily programmed into the mapping set to reject noise and increase template matching accuracy.



(a) The template matching correlation output yielded by kernel cross-correlator

(b) Original image with box placed where tulips are detected

Figure 2.8: The kernel cross-correlation used in template matching for the same example photo as used in 2.3. (a) Shows the correlation output yielded by the correlation algorithm when a cubic kernel and Gaussian mapping set are used. Multiple different reference images were used in the correlation calculations. (b) Shows the location of where the template matching placed the location of the flowers in the image.

2.3.2 Kernel Selection

The selection of the kernel function used in the correlator has large implications on detection performance. There are two important criteria for the selection and formulation of the kernel.

The first being the effects of the transformation on the frequency content of the data and reference sets. The second being the kernel function's effect on SNR values.

The preservation of the distinct frequency content is of major concern with the selection of kernel function. The transformation of the reference is done to counteract any negative effects on the frequency content that transformation of the data may have. However to achieve the best results, dominant frequencies should avoid being altered during transformation. For time-history data sets, polynomial kernels like the cubic in equation 2.30 should only depend on a single variable of $x[k]$. This ensures frequency content is preserved through transformation.

$$\Phi_t(x) = (x[k] + 1)^n \tag{2.35}$$

The selection of the degree of the polynomial n will depend on assumptions made about our signal and noise models. If we assume additive noise, our data $x[k]$ then can be broken up into the signal and the noise $S[k]+N[k]$. The kernel transformation seeks to maximize the SNR value by "boosting" the amplitude of the signal while minimizing the relative increase in the noise amplitude. The theoretical limit where there is no noise in our signal is SNR $= \infty$, either implies that $||N[k]|| = 0$ or the modulus of the noise amplitude is much smaller than the modulus of the signal amplitude and therefore negligible. Thus $||S[k]|| >> ||N[k]||$ or $||S[k]||/||N[k]|| >> 1$.

This can be achieved very simply by a polynomial transformation with infinite degree ($n = \infty$), and will achieve the theoretical SNR maximum. For a discrete data set, we can

approximate this with a very large value of $n >> 1$

$$(x[k]+1)^n = ((S[k]+N[k])+1)^n, \quad \forall n >> 1 \text{ and } n \in \mathbb{Z}$$

Which can then be expanded out into polynomial form

$$(S[k]+N[k])^n + n(S[k]+N[k])^{n-1} + ...1^n \approx S[k]^n + N[k]^n, \quad \forall n >> 1 \text{ and } n \in \mathbb{Z}$$

Because the noise is additive, the data described by $S[k]+N[k]$ prior to transformation will have a larger amplitude than the areas where signal is not present, provided the value of the signal is non-zero. Therefore, after transformation, the areas with signal present will have a much larger amplitude than that where no signal is present.

$$S[k]^n + N[k]^n >> N[k]^n \text{ when } n >> 1 \text{ and } n \in \mathbb{Z}$$

At the limit $n \to \infty$, the amplitude of noise surrounding our data set becomes much smaller compared to the signal, artificially removing noise. The relationship between positive SNR boost and polynomial degree is linear. For computational purposes, polynomials ranging from degree $n = 20$ to 150 are sufficiently large.

Polynomial kernels suffer from numerical instabilities during their calculations. If the polynomial to the power of 1 is less than unity, then it can be generalized that it will trend towards zero as degree increases. Conversely, if it is larger than unity, it will trend towards infinity when degree is increased. Therefore, the addition of a constant in the kernel function is important. For this case, 1 was chosen because it was theorized that the areas where signal was present to be pushed above the threshold causing the amplitude trend towards infinity, whereas pure noise would fall below the threshold and trend towards zero.

This theory falls apart as the noise power becomes so much larger than signal power that it is almost indiscernible and the modulus of the amplitude of the data when $x[k] = S[k] + N[k]$ is comparable to modulus of the pure noise. This means that the transformation would appear as if it was a transformation of pure noise.

Increasing the the order of n, does not change the dominant frequency value, it simply increases the number of harmonics linearly. Given the dominant frequency of the signal, f_0 the location of those harmonics can be predicted by the following relation,

$$f_n = kf_0, \qquad \forall k = 1, 2, ...n \tag{2.36}$$

where k is a constant that varies from 1 to n. Given a 10 Hz signal, the transformation using a 5th degree polynomial will have a base frequency at 10 Hz and then harmonics located at 20, 30, 40, and 50 Hz. 0 Hz denotes the DC bias of the transformation. It is also important to understand how the selection of training set $G(z)$ can affect the frequency of the transformation.

The use of large degree polynomial kernel functions is not needed for data with large SNR values and may increase computation time when it is not needed. Other kernels like functions 2.28, 2.29, and 2.30 suffice, and should be explored instead.

2.3.3 Walk through

Because of the complexity of the algorithm, it may be helpful to describe step-by-step the process by which the KCC can be computed for a very simple example. Given a noisy data set $x[k]$ with a short period of 3 seconds (3072 samples with a sample frequency of 1024 Hz), and a known signal which appears somewhere in the data set, it is desired to find

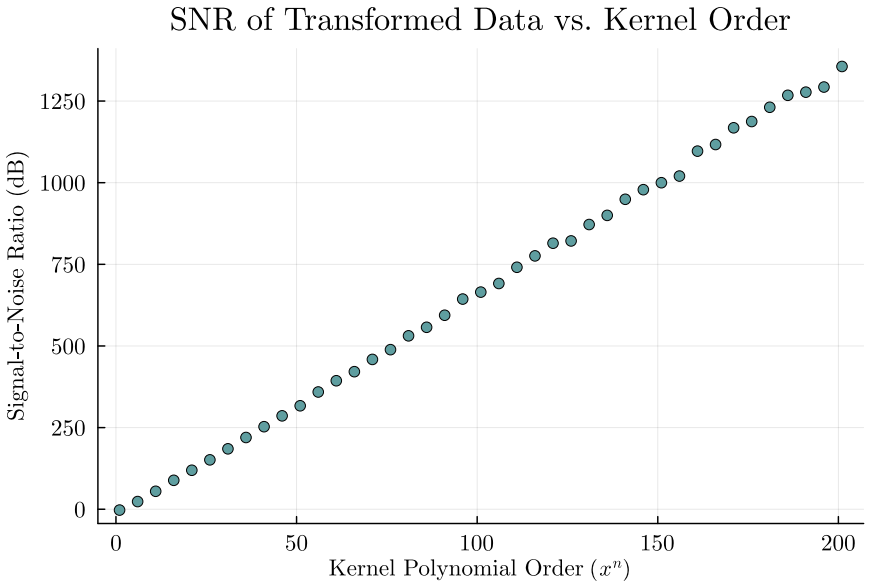


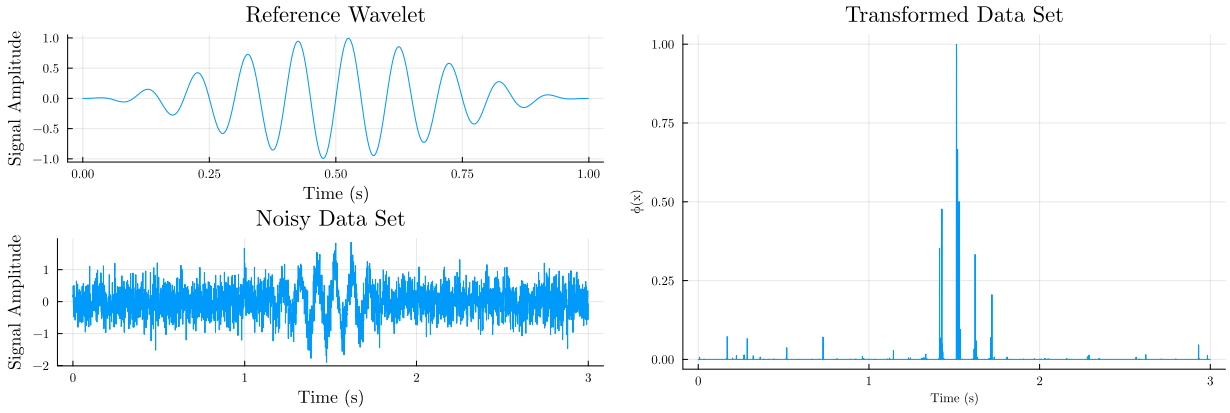
Figure 2.9: SNR of the transformed data versus the order of the kernel polynomial degree for a signal with 0 dB SNR pre-transformation. At $n = 1$ the transformation does not change the amplitude of the data, and therefore SNR remains at 0 dB. However, SNR increases linearly with the polynomial degree.

the signal's location in time. For this example, the SNR values of our data is going to be relatively high, however the signal will still be classified as weak. The reference signal will be a Hann windowed sine wave with an amplitude of unity.

The definition of the KCC requires the kernel transformations of the data $\phi(x[k])$ must be computed in the time domain before the transformation of the data into the frequency domain. A new dimension of the data will be added, it can now be represented as a three-dimensional data set. For this example, a polynomial degree of $n = 15$ will be chosen with a constant of 1.

$$\phi(x[k]) = (x[k] + 1)^{15}$$

The above function is the exact kernel equation used for the transformation, which is sufficient to achieve the desired outcome while minimizing computational cost. A depiction of the data as well as the kernel transformation of the data is shown below in figure 2.10.



(a) (b)

Figure 2.10: (a) The original noisy data set with a Hann windowed sine wavelet with a frequency of 10 Hz. Sub figure (b) shows the feature space transformation $\phi(x)$ of the noisy data using the kernel function described in the previous paragraph.

Once the kernel transformation of the data has been found, the reference signal must also be transformed. However, it first must be zero-padded. Because of the element-wise multiplication of the two sets in the frequency domain, the data must be the same length. Once this has been done, the reference may then be transformed. The data may then be transformed into the frequency domain from the time domain, notice how the kernel transformations were all performed on the time-domain data. The two frequency domain data sets may be then be used in the correlation equation given by 2.31. However, the training set has yet to be selected and encoded into the KCC equation to makeup the $H(Y)$ term in 2.32. For this simple problem, a simple impulse function will be chosen. This inputs equal energy into all frequencies, essentially keeping the data untrained by unaltering the reference $\Phi_t(Y)$ term.

From this, the complex conjugate of the $\Phi_t(Y)$ term can be performed and multiplied by the data set. This will return the correlation in the frequency domain, which can then be transformed back into the time domain via an inverse Fourier transform. The global maximum of the correlation output will then be the approximate location of the useful signal

in the noisy data as found by the KCC. For our simple problem, the correlation output will return a graph like shown in figure 2.11.

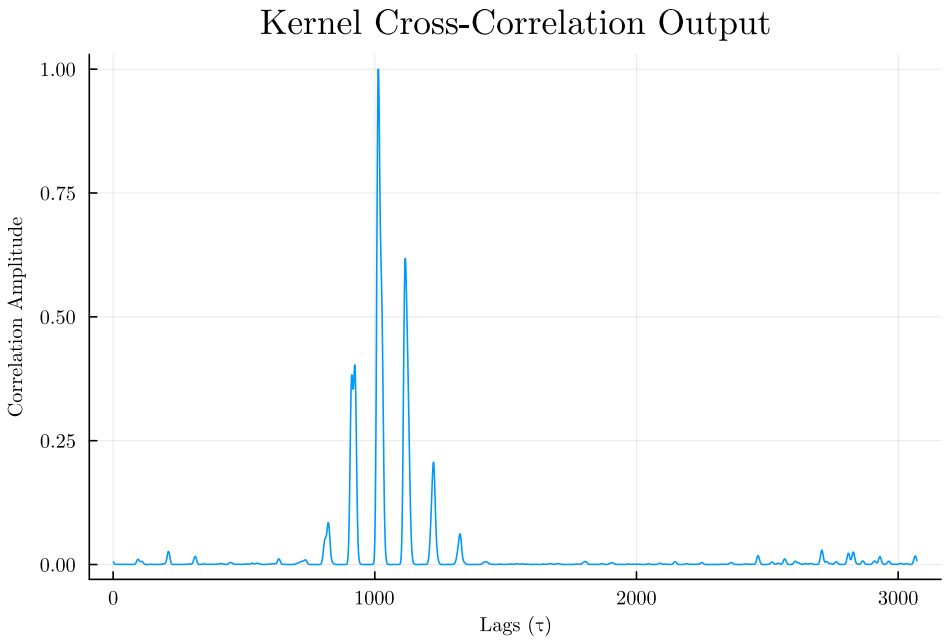


Figure 2.11: Kernel cross-correlation output from the simple case. The global maximum occurs at the beginning of the location of where the signal appears in the noisy data, which in this case would be at lag $\tau = 1024$.

In this simple example, the location of the data is situated in the middle of the time period, with the signal beginning at 1 second. Therefore, when the correlation is taken the peak should appear at lag $\tau = 1^{*}fs$, which it does, indicating that a positive detection result was returned.

2.3.4 Spectral Analysis

Much like the use of the classic correlation, the kernelized version of the cross-correlation can also be used to determine spectral content within a signal. This could provide use for power spectral analyses and frequency identification for signals with low SNR values.

Beginning with the autospectrum or power spectral density, it can be constructed via the frequency domain definition of the autospectrum. This allows a kernelized version of the PSD to be calculated using a combination of equations 2.18 and 2.31. This yields the power spectral density of the kernel transformation of the data.

$$\hat{S}_{xx}(\omega) = \Phi_t(X) \odot \Phi_t^*(X) \tag{2.37}$$

The kernel function transformation rejects the noise within the data set, which in turn will allow for a clearer picture of the frequency content inside of a noisy data set. This definition of the PSD is an effective way to discern the presence, and analyze stationary signals within data. Non-stationary signals that vary in frequency like a chirp can be detected, and will present as multiple frequencies or a frequency "bar". However, no conclusion regarding temporal or phase information can be made because it returns a purely real output. Therefore it is not ideal for analyzing non-stationary signals.

The kernelized version of the cross-spectrum follows a similar definition of the PSD, and again would be a combination of equations 2.18 and 2.31.

$$\hat{S}_{xy}(\omega) = \Phi_t(X) \odot \Phi_t^*(Y) \tag{2.38}$$

It is a further evolution that modifies the second term and uses the reference data set $y(t)$ to check for specific frequency content. However unlike equation 2.31, the use of a transformed reference is not desired. This ensures that the reference maintains the same frequency content and it is not changed by transformation. By checking for shared frequency content between two time-series, the kernel cross-spectral density can strongly indicate the presence of a signal within data, provided there is a shared frequency.

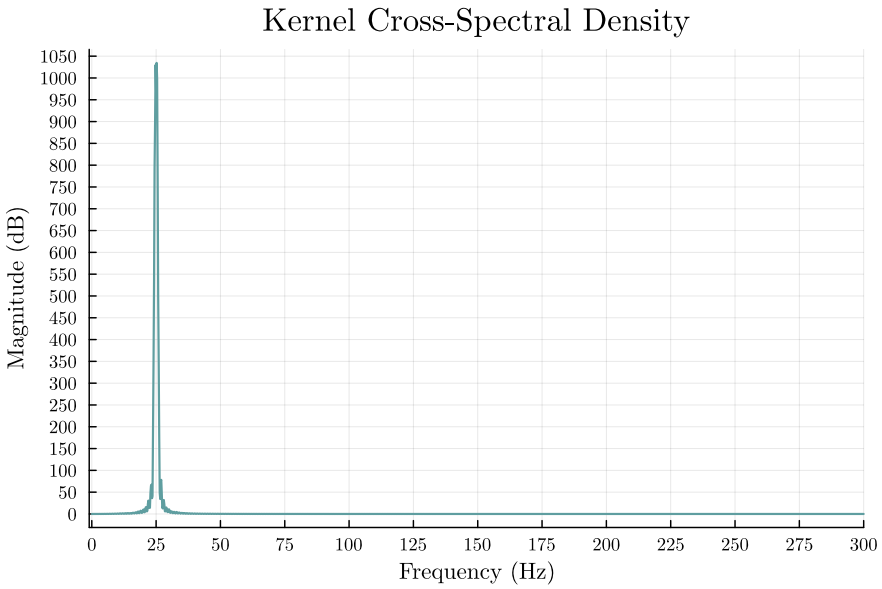


Figure 2.12: The kernel cross-spectrum of the same chirp used in figure 2.4a checked against the same wavelet in figure 2.2a. The spike appears at the shared frequencies between the two, which is 25 Hz. The kernel cross-spectrums peak appears larger than the classical form, however, unlike the classical form, there isn't the same frequency band from 10-300 Hz showing dominant frequencies in the noisy chirp data. This is the kernelized version of the CSD example shown in figure 2.4b

Chapter 3

Results

This section is a presentation of the performance of the Kernel Cross Correlator. In order to properly assess the detection capabilities of the algorithm, Monte Carlo simulations were used to generate detection probability curves for various kernels. These detection capabilities will be compared against classic detection theory like the ideal matched filter and the cross-correlation. Detection simulations will be performed in the context of mean-zero, additive, Gaussian white and Brownian noise. Additionally, different types of signals will be used including stationary waves and non-stationary chirps. Through these simulations, an advancement in signal detection theory will be proved.

3.1 Detection Performance and Simulation Results

Prior to conducting any simulations, it was desired to know the effects of different degree polynomials on transformation SNR. To get a baseline understanding a similar graph as the one shown in figure 2.6 was produced, however using polynomial kernels. Degrees 20 through 120 were used to generate the curves.

Two different Monte Carlo simulations were conducted to analyze the detection performance of two signal types in both white and colored noise environments. The difference between the two being the signals under consideration. The first simulation comprised of a non-stationary linear chirp function, where the second used a sinusoidal wavelet that had a

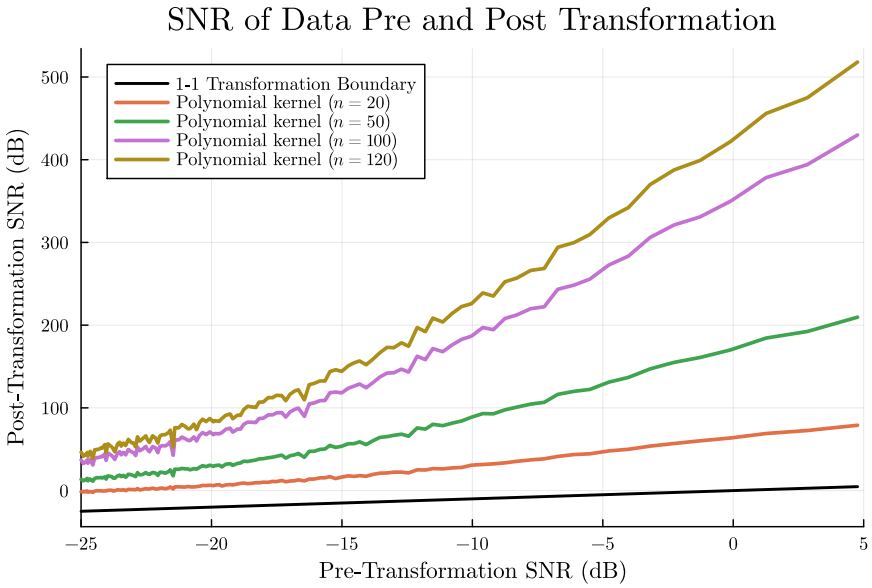


Figure 3.1: Data post-transformation SNR versus pre-transformation SNR for the polynomial kernel derived in equation 2.3.2. The 1-1 transformation line seen in black indicates there is no increase in SNR by transformation.

fixed frequency. Each signal possessed identical signal periods and equivalent signal power.

Signal-to-Noise Ratio was systematically adjusted across a range from 5 dB to -60 dB, as this was the range of interest, and was assumed that past -60 dB detection would be highly unlikely. The detection performance at a given SNR was evaluated through 500 independent simulation trials. In each simulation repetition, the signal power remained constant, while different noise datasets, with approximately equivalent noise power, were employed to maintain consistent SNR levels. The average SNR was calculated across each individual trial. Notably, the signal's location within the data remained fixed throughout the tests and was known prior to the simulation.

The correlation was then performed using the variation of the correlation algorithm in question. Given the location of the signal was known, an approximate location for the spike in the correlation was assumed. Binary hypothesis testing was used in order to verify proper detection of the signal. If the spike did not occur within one tenth of the sample

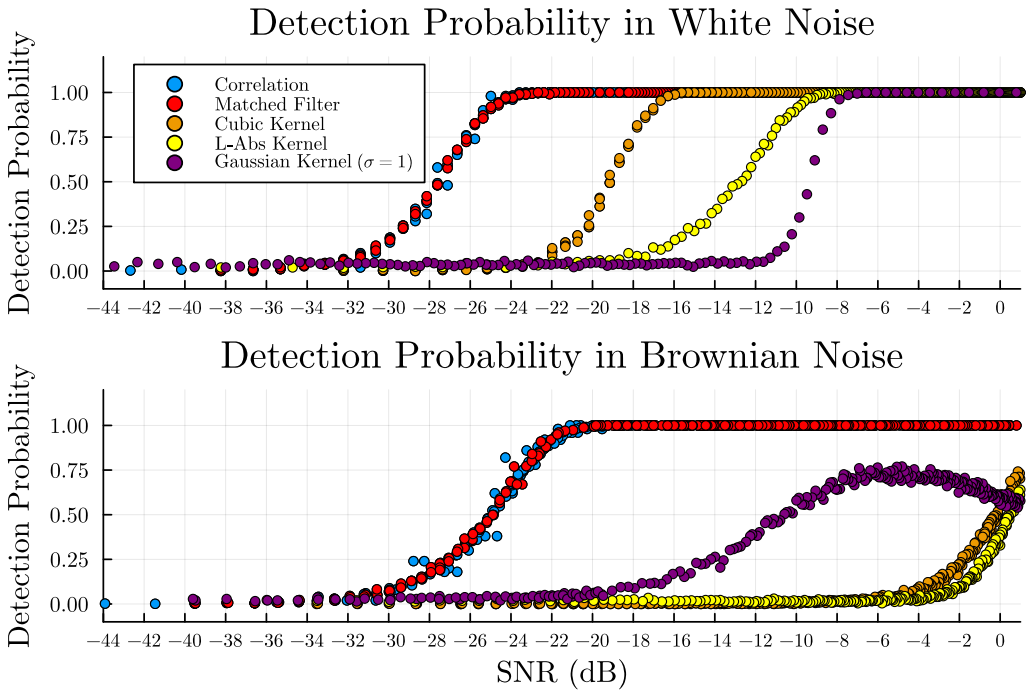


Figure 3.2: First Monte Carlo simulation conducted with the three classic SVM kernels and a linear chirp signal. The same simulations were conducted in both white and colored noise to understand the performance differences between the two.

frequency lags accuracy of the presumed location (0.1 s given a 10 kHz sample frequency), then the null hypothesis would be rejected, hence a failure to detect the signal. For a positive detection, a value of boolean value of 1 was assigned to the output and 0 for a failure. These outcomes were recorded and then averaged over the 500 individual simulations to determine a probability value at that specific SNR. These simulations were conducted in both white and Brownian noise, with two different signals. Simulations were done in the Julia programming language and results were then used to generate SNR versus detection performance curves.

Of the detection algorithms tests, the first was the classical version of the cross-correlation followed by the matched filter. These initial simulations were used as control baselines for the remainder of the tests. The goal was to achieve a higher detection performance at lower SNR values than the correlation and matched filter. The first Monte Carlo simulation that

was conducted only tested the three classic SVM kernels presented in equations 2.28, 2.29, and 2.30. This test used non-stationary chirp signals, which were chosen for its common use in radar and sonar systems. Examples of biological sonar echolocation also frequently use chirp signals. The results from this simulation is presented in figure 3.2.

In order to assess the performance for the large degree polynomial kernel described in equation 2.3.2 a similar Monte Carlo simulation was performed. This simulation used the same type of signal and can be used to draw conclusions regarding the relative performance to the kernel functions defined in equations 2.28, 2.29, and 2.30. Two different polynomial degrees were used for this portion of the performance simulation, being $n = 50$ and $n = 100$. These two were chosen arbitrarily. The results from the polynomial kernel functions can be seen in figure 3.3.

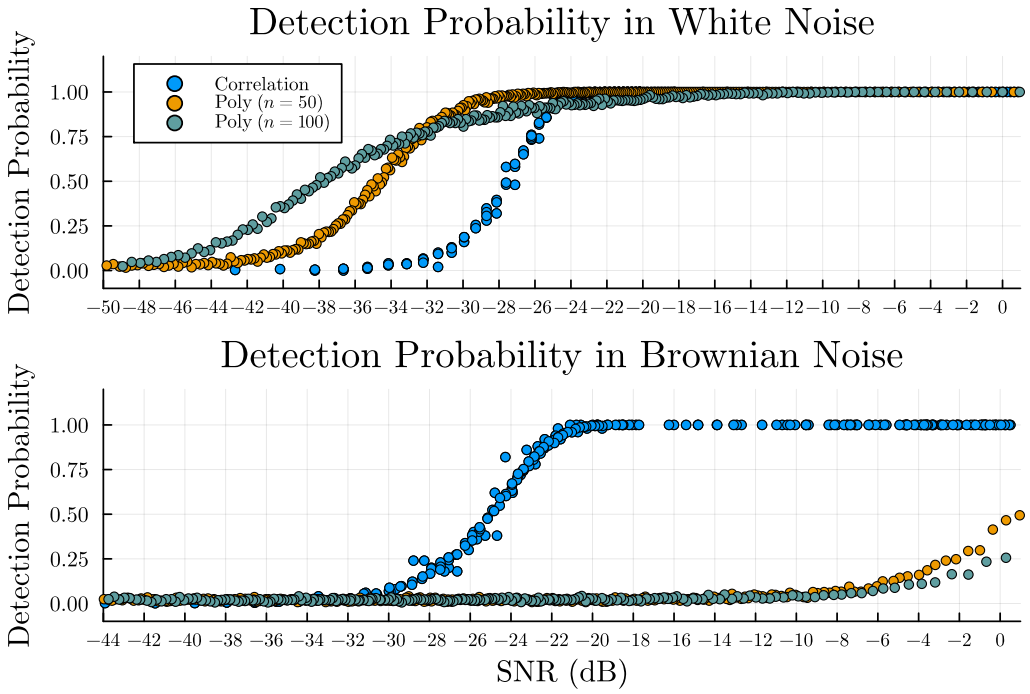


Figure 3.3: First Monte Carlo simulation conducted with the two polynomial kernel functions of different degrees.

A second Monte Carlo simulation was done using the same set of kernels however, the

signal in question was changed from a chirp to a sinusoidal wavelet of the same signal period and comparable signal power. This was done to investigate any dependencies on performance related to the signal used. This simulation used a 50 Hz sine wave pulse as its data content. Like the above simulation, both white and Brownian noise were examined.

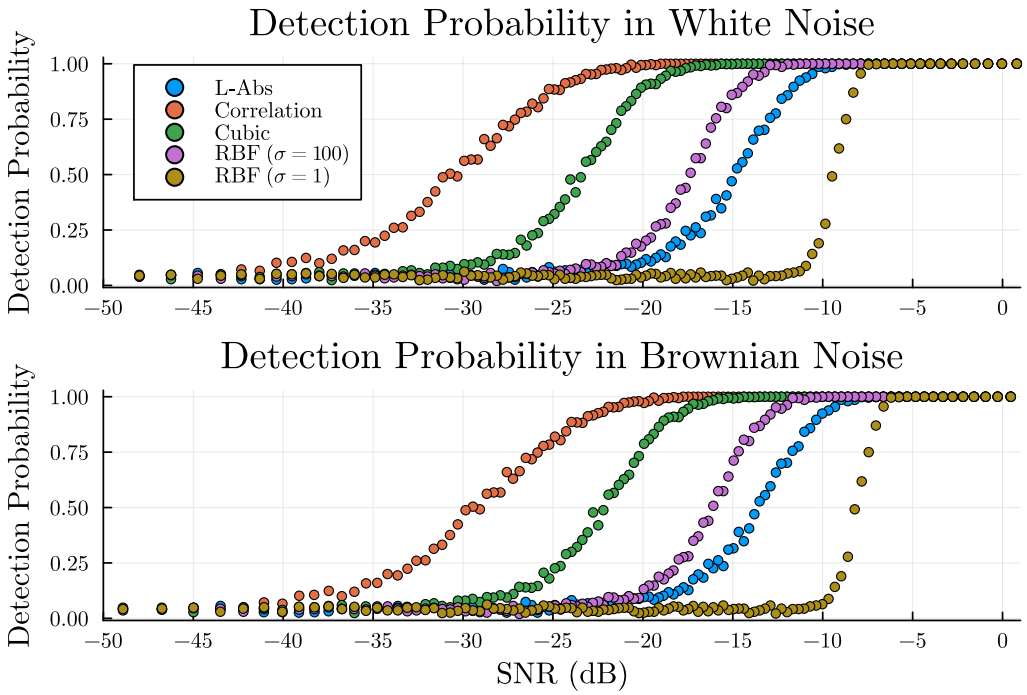


Figure 3.4: Second Monte Carlo simulation conducted with the same three kernel functions as used in figure 3.2, however with the sinusoidal wavelet.

For this iteration, different degree polynomials varying from $n = 20$ up to $n = 120$ were used. The different polynomial degrees were used in order to validate theory regarding the trend of increasing polynomial order. It was also desired to find the cross-off point, *i.e.* what degree was required in order to achieve a similar performance to the cross-correlation/matched filter.

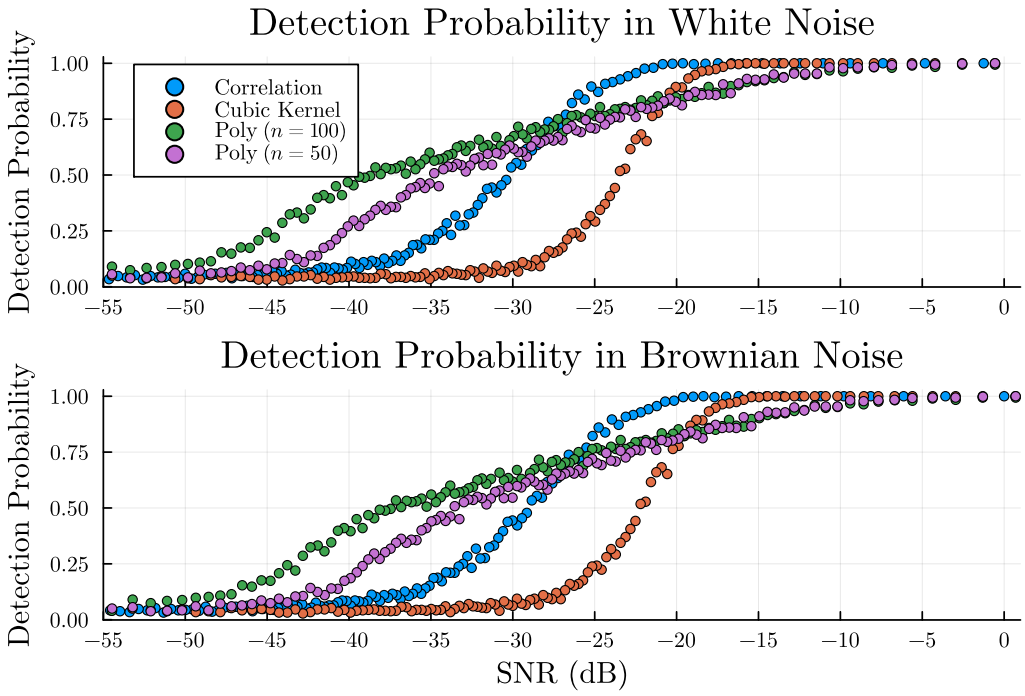


Figure 3.5: Second Monte Carlo simulation conducted with the polynomial kernel correlation. Multiple different degree polynomials are shown to draw conclusions to performance. The cubic kernel from the first simulation is also shown to show the relative performance to the other classic kernels.

Chapter 4

Discussion

Most signal processing techniques are more than adequate for data with relatively large signal-to-noise-ratios. A challenge and limiting factor to many important engineering systems is the detection of signals in the presence of powerful noise. As a result, technology that would advance the performance of radar and sonar systems would be deemed extremely valuable, and as such are an important field of research. This thesis seeks to determine the usability of a new kernel based algorithm to advance the field of weak signal detection.

4.1 Kernel Transformations and SNR Boost

Beginning with the analysis of the kernel transformations, it was noted in chapter 2 that traditional kernels were beneficial for inflating the SNR of signals. All of the kernels proved they could boost the SNR provided that the proper parameters were selected. The RBF kernel with a width parameter of $\sigma = 1$ under-performed as the entire curve fell under the 1-1 transformation line. It was also deemed to be a worse alternative than the RBF kernel where $\sigma = 100$ where there was a proven boost in SNR. There is a proportional relationship between width parameter and SNR boost for the Gaussian kernel. All other transformation functions, were in fact a positive boost on SNR and fell above the 1-1 transformation line, albeit offering only a small advantage.

The same simulation was done with the polynomial kernels and were presented in figure

3.1. Similarly to the SVM kernels, the polynomial transformation functions resulted in a far greater boost in SNR. There also appeared to be a linear relationship between the degree and the SNR boost. This can be seen in both figures 2.9 and 3.1 where the larger the degree of function, the higher above the 1-1 transformation line the curve appeared.

$$\text{SNR} \propto n$$

Interestingly, the curves appeared to return an increasingly larger SNR value post-transformation the larger the pre-transformation SNR became. The rate at which the return increased appeared exponential and proportional to the degree polynomial. At the lower SNR range, the transformation curves did grow tighter together, indicating that for data with very low SNR, the benefits of increasing polynomial degree may outweigh computational costs.

4.2 First Monte Carlo Simulation

Of the first Monte Carlo simulation performed, the first algorithms tested were the matched filter and the cross-correlation. These were mainly used as a benchmark for performance and a control for following simulations. It was expected that the ideal matched filter would outperform the cross-correlation; however, they appeared to have the exact same detection probability across both types of noise. In the first simulation, where the signals in question was a chirp, the detection roll-off began around -25 to -22 decibels, with the 0.50 detection probability hitting at around -3 dB from the initial roll-off point. This was relatively consistent across the two types of signals tested.

The second simulation where the sine wavelet was used, the roll off began around the same place, however the slope was not as steep and the 0.50 probability of detection mark

landed at -30 dB, 2 to 5 decibels less than the first simulation. It was expected that a linear or logarithmic trend with a very sharp drop off would be observed. The detection probability curves appeared like sigmoid functions which have connections to convolutional neural networks and saturation problems.

Notably, all kernel cross-correlation functions from the first simulation performed worse than the cross-correlation and the matched filter. This was consistent across both types of noise, however, all three kernel functions tested performed much better in white noise than in colored noise, which was to be expected. The cubic kernel correlator performed about 8 dB worse than the cross-correlation, while in white noise, and barely was able to detect weak chirps in Brownian noise with the 0.50 probability mark landing at the 0 dB mark. The linear-absolute value kernel performed very similarly, and was only slightly worse in white noise than the cubic kernel.

While the cubic and the L-Abs kernels behaved similarly, the Gaussian kernel, was the outlier. In white noise, it performed the worst of the three, however, in Brownian noise it outperformed its white noise detection. Oddly, it never achieved a max probability 1.0, instead peaking at around -5 dB at a probability of 0.75. Perhaps paradoxically, all of the kernel cross-correlators under performed relative to what was expected based on the SNR boost created from transformation.

The polynomial kernels were also tested in the same simulation and are presented on a second figure. This was mainly done for figure cleanliness as many of the curves laid on top of each other. It can be seen from figure 3.3 that both kernel functions out performed the cross-correlation in white noise. The performance gap was as wide as 10 dB difference at some points along the curves. Interestingly, the kernel that used a degree of 50, maintained the probability of 1 for a longer range than its larger degree counterpart. The distributions also appeared slightly different where the polynomial that used a degree of 50 appeared very

similar to the sigmoid function distribution exhibited by the cross-correlation whereas the degree 100 appeared to present a more logarithmic growth.

In colored noise, the polynomial kernel transformation functions were distinctly poor. In fact, they performed the worst of all kernels tested. Perhaps, the KCC is particularly unsuited for detection in data with a combination of the chirp signal and colored noise. However, the KCC does seem to provide an advantage over the cross-correlation in terms of detection performance in white noise.

4.3 Second Monte Carlo Simulation

Performance was not purely a function of signal used. It can be seen in figure 3.4 that the cubic, linear-absolute value, and RBF kernels still fall short of the cross-correlation curve in the second simulation. Performance was slightly better in white noise with the sinusoidal wavelet, having shifted -3 dB to the left, somewhat similar to what was seen with the cross-correlation. However, the outcome of this simulation is very comparable to the white noise simulation in simulation 1.

The shift between simulation 1 and simulation 2 for data in Brownian noise was much larger at -20 dB. All probability curves appeared the same regardless of the type of noise present. Thus, it can be reasonably concluded that the gap in performance between the white and Brownian noise was tightened using the sine wavelet as the signal. However, the kernel functions still under performed relative to the cross-correlation.

Lastly, the polynomial kernel functions were tested for performance. For simplicity, two different polynomial degrees from the selected range were shown on figure 3.5, those being $n = 50$ and $n = 100$. It was determined that kernels with degrees above $n = 30$ all had

larger probability densities than the cross-correlation, and therefore performed better.

Interestingly, the polynomial kernel seems to deviate from the trend of a sigmoid distribution. Instead, it appears more like logarithmic growth. It begins to fall off more gradually, much earlier than the cross-correlation at about -10 dB. However, the cross-over begins at the 0.75 detection probability mark where there is a higher rate of detection from the polynomial kernel than the cross-correlation for signals less than -25 dB SNR. In fact, at the point where both the cross-correlation had a detection probability of 0.5 was 10 dB higher than the polynomial KCC with degree $n = 100$. Indicating for signals with -40 dB, a polynomial kernel has a probability of detection of 0.50 where the cross-correlation has that less than 0.10. This performance gap was carried over across the two noise types.

For degrees higher than $n = 130$, numerical calculation issues were encountered limiting the range of degrees. These errors were the result of memory overflows because of the extremely large amplitude transformations being computed. For data with increasingly worse SNR, the noise power and thus the amplitude of the noise must increase, therefore the feature space amplitude increases as well. For very high degrees, the transformations become too large for memory allocation, and therefore will return an "NaN" due to memory overflow. This issue can be alleviated easily by dividing the noisy data by a constant to keep the average absolute value of the amplitude close to unity. This also maintains SNR as the signal and the noise power are both divided by the same constant.

As a whole, the polynomial kernel correlator provided an increase in the detection capabilities over the cross-correlation, as it increased the detection probability by as much as 0.50 for some SNR values. It was also also found that there was a greater than zero probability for signals out past -50 dB SNR. While there are admitted shortcomings of the algorithm, there are significant advantages of the KCC over previous theory.

Chapter 5

Conclusion

As the pace of society is pushed further into the future, there is a demand for increasingly higher performing technology to power evermore complex systems. As militaries evolve and electronics improve, physics and engineering are pushed to further innovate. Digital signal processing is a foundational principal for many important everyday tools heavily entangled in modern society. A subset of digital signal focused on pushing theoretical limits when noise power becomes larger than the power of the useful signal is called weak signal detection.

Advancements in weak signal detection facilitate breakthroughs in active and passive sonar, medical devices, and even the finance sector. For this reason it is of particular interest to develop new signal processing techniques which provide a clear advantage in noise rejection. In this thesis, foundational signal detection theory was presented, combined with a simple AI classification model to produce a kernel cross-correlation algorithm using a large degree polynomial kernel specifically designed for weak signal detection.

Through a number of numerical simulations, it was found that the polynomial kernel function was able to increase SNR of a dataset as the kernel degree increased. Thus allowing a typically weak signal to be treated as if it were data with a more acceptable SNR. The relationship between SNR and degree was found to be approximately linear as seen in figure 2.9. Rather than using the original data, it was theorized that implementing kernel transformations into the algorithm would increase detection performance and provide distinct advantages for some applications.

To validate this theory, two independent Monte Carlo simulations were conducted to generate detection probability densities for a number of different kernel functions in the context of two different types of noise. The two tests only differed in the type of signal of interest used. In both simulations, polynomial kernel functions improved signal detection probability for degrees larger than $n = 30$ while in the presence of white noise. There was a marked improvement in probability density observed for polynomials larger than $n = 50$. In some cases, the KCC outperformed the cross-correlation by around a 10 dB difference past -40 dB SNR.

While specific signals in colored noise failed dramatically to detect weak signals, the same increase in performance was observed for others. Which leads to the conclusion that there is a dependence on the signal that is of interest. However, the KCC does appear to be an improvement over previously established theory, particularly in the presence of white-passing noise.

Furthermore, the KCC can be built upon to form other algorithms for frequency domain analysis of signals. Not unlike the correlation algorithm, there is the potential for expansion of the algorithm past this single application. The kernel cross-spectrum does hold some promise and its resistance to noise corruption could be of interest in the future.

The KCC and its advantages has implications for a wide variety of applications. In resource restricted environments, the superior performance could allow for the reduction of equipment and other algorithms in parallel to the kernel cross-correlator. There may also be a reduce false alarm rate as noise does not corrupt the correlation algorithm as easily. Regardless of future application, the advantages of the KCC and the advancements in signal processing it provides cannot be denied.

Bibliography

[1] J. A. Swets, "History of Signal Detection Theory," *International Encyclopedia of Social & Behaviour Sciences*, pp. 14 075–14 078, 2001.

[2] J. Deeks, "Nonlinear Signal Processing Techniques for Signal Detection," English, Ph.D. dissertation, Univeristy of Southampton, Institute of Sound and Vibration Research, Nov. 2017.

[3] J. Wang, L. Yang, L. Gao, and Q. Miao, "Current progress on weak signal detection," en, in *2013 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE)*, Chengdu, China: IEEE, Jul. 2013, pp. 1812–1818, ISBN: 978-1-4799-1016-8 978-1-4799-1014-4. DOI: 10.1109/QR2MSE.2013.6625929. [Online]. Available: http://ieeexplore.ieee.org/document/6625929/ (visited on 09/23/2023).

[4] M. Li-xin, "Weak Signal Detection Based on Duffing Oscillator," en, in *2008 International Conference on Information Management, Innovation Management and Industrial Engineering*, Taipei, Taiwan: IEEE, Dec. 2008, pp. 430–433, ISBN: 978-0-7695-3435-0. DOI: 10.1109/ICIII.2008.226. [Online]. Available: http://ieeexplore.ieee.org/document/4737578/ (visited on 09/23/2023).

[5] G. Tolstov, *Fourier Series*. Dover Publications, 1962.

[6] N. Kasdin, "Discrete simulation of colored noise and stochastic processes and 1/f/sup / power law noise generation," en, *Proceedings of the IEEE*, vol. 83, no. 5, pp. 802–827, May 1995, ISSN: 00189219. DOI: 10.1109/5.381848. [Online]. Available: http://ieeexplore.ieee.org/document/381848/ (visited on 12/22/2023).

[7] J. Barnes and D. Allan, "A statistical model of flicker noise," en, *Proceedings of the IEEE*, vol. 54, no. 2, pp. 176–178, 1966, ISSN: 0018-9219. DOI: 10.1109/PROC.1966.4630. [Online]. Available: http://ieeexplore.ieee.org/document/1446560/ (visited on 01/09/2024).

[8] A. Einstein, "-INVESTIGATIONS O N THE THEORY ,THE BROWNIAN MOVEMENT," en,

[9] J. Johnson, *Thermal Agitation of Electricity in Conductors*, 1928.

[10] J. F. Qu, S. P. Benz, H. Rogalla, W. L. Tew, D. R. White, and K. L. Zhou, "Johnson noise thermometry," en, *Measurement Science and Technology*, vol. 30, no. 11, p. 112 001, Nov. 2019, ISSN: 0957-0233, 1361-6501. DOI: 10.1088/1361-6501/ab3526. [Online]. Available: https://iopscience.iop.org/article/10.1088/1361-6501/ab3526 (visited on 02/20/2024).

[11] D. V. Perepelitsa, "Johnson Noise and Shot Noise," en,

[12] K. J. Benoit-Bird and C. M. Waluk, "Remote acoustic detection and characterization of fish schooling behavior," en, *The Journal of the Acoustical Society of America*, vol. 150, no. 6, pp. 4329–4342, Dec. 2021, ISSN: 0001-4966, 1520-8524. DOI: 10.1121/10.0007485. [Online]. Available: https://pubs.aip.org/jasa/article/150/6/4329/994395/Remote-acoustic-detection-and-characterization-of (visited on 03/14/2024).

[13] G. L. A. De Sousa and G. C. Cardoso, "A battery-resistor analogy for further insights on measurement uncertainties," en, *Physics Education*, vol. 53, no. 5, p. 055 001, Sep. 2018, ISSN: 0031-9120, 1361-6552. DOI: 10.1088/1361-6552/aac84b. [Online]. Available: https://iopscience.iop.org/article/10.1088/1361-6552/aac84b (visited on 12/22/2023).

[14] C. H. Sherman and J. L. Butler, *Transducers and Arrays for Underwater Sound*, en. New York, NY: Springer New York, 2007, ISBN: 978-0-387-32940-6 978-0-387-33139-3. DOI: 10.1007/978-0-387-33139-3. [Online]. Available: http://link.springer.com/10.1007/978-0-387-33139-3 (visited on 12/22/2023).

[15] S. Kassam and H. Poor, "Robust techniques for signal processing: A survey," en, *Proceedings of the IEEE*, vol. 73, no. 3, pp. 433–481, 1985, ISSN: 0018-9219. DOI: 10.1109/PROC.1985.13167. [Online]. Available: http://ieeexplore.ieee.org/document/1457435/ (visited on 09/23/2023).

[16] J. D. Hamilton, *Time series analysis*, en. Princeton, N.J: Princeton University Press, 1994, ISBN: 978-0-691-04289-3.

[17] C. Systems, "Signal-to-Noise Ratio (SNR) and Wireless Signal Strength," Oct. 2023.

[18] F. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," en, *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, 1978, ISSN: 0018-9219. DOI: 10.1109/PROC.1978.10837. [Online]. Available: http://ieeexplore.ieee.org/document/1455106/ (visited on 09/29/2023).

[19] I. S. Lin and A. M. Weiner, "Selective Correlation Detection of Photonically Generated Ultrawideband RF Signals," en, *Journal of Lightwave Technology*, vol. 26, no. 15, pp. 2692–2699, Aug. 2008, ISSN: 0733-8724. DOI: 10.1109/JLT.2008.927174. [Online]. Available: http://ieeexplore.ieee.org/document/4652287/ (visited on 12/23/2023).

[20] J. O. Smith, *Mathematics of the discrete Fourier transform (DFT): with audio applications*, eng, 2. ed. North Charleston: BookSurge, 2007, ISBN: 978-0-9745607-4-8.

[21] J. Michels, P. Varshney, and D. Weiner, "Multichannel signal detection involving temporal and cross-channel correlation," en, *IEEE Transactions on Aerospace and Elec-*

tronic Systems, vol. 31, no. 3, pp. 866–880, Jul. 1995, ISSN: 00189251. DOI: 10.1109/7.395251. [Online]. Available: http://ieeexplore.ieee.org/document/395251/ (visited on 10/23/2023).

[22] K. Kumatani, J. McDonough, J. F. Lehman, and B. Raj, "Channel selection based on multichannel cross-correlation coefficients for distant speech recognition," en, in *2011 Joint Workshop on Hands-free Speech Communication and Microphone Arrays*, Edinburgh, United Kingdom: IEEE, May 2011, pp. 1–6, ISBN: 978-1-4577-0997-5. DOI: 10.1109/HSCMA.2011.5942398. [Online]. Available: http://ieeexplore.ieee.org/document/5942398/ (visited on 10/23/2023).

[23] J. P. Lewis, "Fast Normalized Cross-Correlation," en, *Industrial Light and Magic*, 1995.

[24] M. Savvides and P. Khosla, "Face Verification using Correlation Filters," en,

[25] B. Kumar, M. Savvides, and Chunyan Xie, "Correlation Pattern Recognition for Face Recognition," en, *Proceedings of the IEEE*, vol. 94, no. 11, pp. 1963–1976, Nov. 2006, ISSN: 0018-9219, 1558-2256. DOI: 10.1109/JPROC.2006.884094. [Online]. Available: http://ieeexplore.ieee.org/document/4052477/ (visited on 12/23/2023).

[26] H. W. Hsu, T.-Y. Wu, W. H. Wong, and C. Y. Lee, "Correlation-Based Face Detection for Recognizing Faces in Videos," en, in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB: IEEE, Apr. 2018, pp. 3101–3105, ISBN: 978-1-5386-4658-8. DOI: 10.1109/ICASSP.2018.8461485. [Online]. Available: https://ieeexplore.ieee.org/document/8461485/ (visited on 12/23/2023).

[27] H. Wen and Z. Liu, "Separating Fractal and Oscillatory Components in the Power Spectrum of Neurophysiological Signal," en, *Brain Topography*, vol. 29, no. 1, pp. 13–26, Jan. 2016, ISSN: 0896-0267, 1573-6792. DOI: 10.1007/s10548-015-0448-0. [Online].

Available: http://link.springer.com/10.1007/s10548-015-0448-0 (visited on 09/29/2023).

[28] H. v. Storch and F. W. Zwiers, *Statistical analysis in climate research*, en. Cambridge ; New York: Cambridge University Press, 1999, ISBN: 978-0-521-45071-3.

[29] F. S. Racz, A. Czoch, Z. Kaposzta, O. Stylianou, P. Mukli, and A. Eke, "Multiple-Resampling Cross-Spectral Analysis: An Unbiased Tool for Estimating Fractal Connectivity With an Application to Neurophysiological Signals," *Frontiers in Physiology*, vol. 13, p. 817 239, Mar. 2022, ISSN: 1664-042X. DOI: 10.3389/fphys.2022.817239. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8936508/ (visited on 09/29/2023).

[30] Y. Liu, J. Liu, and R. Kennel, "Optimization of the Processing Time of Cross-Correlation Spectra for Frequency Measurements of Noisy Signals," en, *Metrology*, vol. 2, no. 2, pp. 293–310, Jun. 2022, ISSN: 2673-8244. DOI: 10.3390/metrology2020018. [Online]. Available: https://www.mdpi.com/2673-8244/2/2/18 (visited on 10/23/2023).

[31] R. E. Ziemer and W. H. Tranter, *Principles of communications: systems, modulation, and noise*, en, Seventh edition. Danvers, MA: Wiley, 2015, ISBN: 978-1-118-07891-4.

[32] J. C. Bancroft, "Introduction to matched filters," en, vol. 14, 2002.

[33] G. Turin, "An introduction to matched filters," en, *IEEE Transactions on Information Theory*, vol. 6, no. 3, pp. 311–329, Jun. 1960, ISSN: 0018-9448. DOI: 10.1109/TIT.1960.1057571. [Online]. Available: http://ieeexplore.ieee.org/document/1057571/ (visited on 01/24/2024).

[34] D. Gabor, "Theory of Communication," *J. IEE (London)*, vol. 93, pp. 429–457, 1946.

[35] L. Cohen, "Time-frequency distributions-a review," en, *Proceedings of the IEEE*, vol. 77, no. 7, pp. 941–981, Jul. 1989, ISSN: 00189219. DOI: 10.1109/5.30749. [Online]. Available: http://ieeexplore.ieee.org/document/30749/ (visited on 10/04/2023).

[36] E. Sejdić, I. Djurović, and J. Jiang, "Time–frequency feature representation using energy concentration: An overview of recent advances," en, *Digital Signal Processing*, vol. 19, no. 1, pp. 153–183, Jan. 2009, ISSN: 10512004. DOI: 10.1016/j.dsp.2007.12.004. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S105120040800002X (visited on 12/23/2023).

[37] M. Varanis, J. P. C. V. Norenberg, R. T. Rocha, C. Oliveira, J. M. Balthazar, and Â. M. Tusset, "A Comparison of Time-Frequency Methods for Nonlinear Dynamics and Chaos Analysis in an Energy Harvesting Model," en, *Brazilian Journal of Physics*, vol. 50, no. 3, pp. 235–244, Jun. 2020, ISSN: 0103-9733, 1678-4448. DOI: 10.1007/s13538-019-00733-x. [Online]. Available: http://link.springer.com/10.1007/s13538-019-00733-x (visited on 09/23/2023).

[38] B. Boashash and P. O'Shea, "Use of the cross Wigner-Ville distribution for estimation of instantaneous frequency," en, *IEEE Transactions on Signal Processing*, vol. 41, no. 3, pp. 1439–1445, Mar. 1993, ISSN: 1053587X. DOI: 10.1109/78.205752. [Online]. Available: http://ieeexplore.ieee.org/document/205752/ (visited on 09/28/2023).

[39] T. Claasen and W. Mecklenbrauker, "The Wigner Distribution – A Tool For Time-Frequency Signal Analysis," *Phillips Journal of Research*, vol. 35, no. 4, pp. 276–300, Jan. 1980.

[40] A. Kumar and A. Prasad, "Wigner-Ville distribution function in the framework of linear canonical transform," en, *Journal of Pseudo-Differential Operators and Applications*, vol. 13, no. 3, p. 38, Sep. 2022, ISSN: 1662-9981, 1662-999X. DOI: 10.1007/

s11868-022-00471-w. [Online]. Available: https://link.springer.com/10.1007/s11868-022-00471-w (visited on 09/28/2023).

[41] S. Haykin and T. Bhattacharya, "Wigner-Ville distribution: An important functional block for radar target detection in clutter," en, in *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*, vol. 1, Pacific Grove, CA, USA: IEEE Comput. Soc. Press, 1994, pp. 68–72, ISBN: 978-0-8186-6405-2. DOI: 10.1109/ACSSC.1994.471419. [Online]. Available: http://ieeexplore.ieee.org/document/471419/ (visited on 09/28/2023).

[42] C. Guanghua, M. Shiwei, Q. Tinghao, W. Jian, and C. Jialin, "The Wigner-Ville Distribution and the Cross Wigner-Ville Distribution of Noisy Signals," en, in *2006 8th international Conference on Signal Processing*, Guilin, China: IEEE, 2006, p. 4 128 821, ISBN: 978-0-7803-9736-1. DOI: 10.1109/ICOSP.2006.344485. [Online]. Available: http://ieeexplore.ieee.org/document/4128821/ (visited on 09/28/2023).

[43] F. Camastra and A. Vinciarelli, *Machine learning for audio, image and video analysis: theory and applications* (Advanced information and knowledge processing), en. London: Springer, 2008, OCLC: ocn166357875, ISBN: 978-1-84800-006-3 978-1-84800-007-0.

[44] A. Zaki, A. Métwalli, M. H. Aly, and W. K. Badawi, "Enhanced feature selection method based on regularization and kernel trick for 5G applications and beyond," en, *Alexandria Engineering Journal*, vol. 61, no. 12, pp. 11 589–11 600, Dec. 2022, ISSN: 11100168. DOI: 10.1016/j.aej.2022.05.024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S111001682200343X (visited on 10/23/2023).

[45] W. S. Noble, "What is a support vector machine?" en, *Nature Biotechnology*, vol. 24, no. 12, pp. 1565–1567, Dec. 2006, ISSN: 1087-0156, 1546-1696. DOI: 10.1038/nbt1206-

1565. [Online]. Available: https://www.nature.com/articles/nbt1206-1565 (visited on 11/29/2023).

[46] M. A. Aizerman, E. M. Braverman, and L. I. Rozonoer, "Theoretical foundations of the potential function method in pattern recognition learning," *Automation and Telecommunication*, vol. 25, no. 6, pp. 917–936, Jun. 1964.

[47] P. J. Phillips, "Support vector machines applied to face recognition," en, National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep. NIST IR 6241, 1998, NIST IR 6241. DOI: 10.6028/NIST.IR.6241. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir6241.pdf (visited on 11/29/2023).

[48] A. Elen, S. Baş, and C. Közkurt, "An Adaptive Gaussian Kernel for Support Vector Machine," en, *Arabian Journal for Science and Engineering*, vol. 47, no. 8, pp. 10 579–10 588, Aug. 2022, ISSN: 2193-567X, 2191-4281. DOI: 10.1007/s13369-022-06654-3. [Online]. Available: https://link.springer.com/10.1007/s13369-022-06654-3 (visited on 12/24/2023).

[49] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," en, *SUPPORT VECTOR MACHINES*,

[50] B. Tan, J. Guo, and G. Chang, "Detection of weak monocycle sinusoidal signals with a low constant false alarm rate based on the support vector machine," en, *The Journal of Engineering*, vol. 2019, no. 16, pp. 2255–2260, Mar. 2019, ISSN: 2051-3305, 2051-3305. DOI: 10.1049/joe.2018.8584. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1049/joe.2018.8584 (visited on 01/09/2024).

[51] C. Wang, L. Zhang, L. Xie, and J. Yuan, "Kernel Cross-Correlator," en, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018, ISSN: 2374-3468, 2159-5399. DOI: 10.1609/aaai.v32i1.11710. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11710 (visited on 09/28/2023).