Constrained Clustering for Frequency Hopping Spread Spectrum Signal Separation

Parker D. White

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

R. Michael Buehrer, Co-chair
William C. Headley, Co-chair
Jeffrey H. Reed
Ryan K. Williams

August 12, 2019
Blacksburg, Virginia

Keywords: Frequency Hopping Spread Spectrum, Signal Separation, Constrained Clustering, Semi-Supervised Learning

Constrained Clustering for Frequency Hopping Spread Spectrum Signal Separation

Parker D. White

(ABSTRACT)

Frequency Hopping Spread Spectrum (FHSS) signaling is used across many devices operating in both regulated and unregulated bands. In either situation, if there is a malicious device operating within these bands, or more simply a user operating out of the required specifications, the identification this user important to insure communication link integrity and interference mitigation. The identification of a user involves the grouping of that users signal transmissions, and the separation of those transmission from transmissions of other users in a shared frequency band. Traditional signal separation methods often require difficult to obtain hardware fingerprinting characteristics or approximate geo-location estimates. This work will consider the characteristics of FHSS signals that can be extracted directly from signal detection. From estimates of these hopping characteristics, novel source separation with classic clustering algorithms can be performed. Background knowledge derived from the time domain representation of received waveforms can improve these clustering methods with the novel application of cannot-link pairwise constraints to signal separation. For equivalent clustering accuracy, constraint-based clustering tolerates higher parameter estimation error, caused by diminishing received signal-to-noise ratio (SNR), for example. Additionally, prior work does not fully cover the implications of detecting and estimating FHSS signals using image segmentation on a Time-Frequency (TF) waterfall. This work will compare several methods of FHSS signal detection, and discuss how each method may effect estimation accuracy and signal separation quality. The use of constraint-based clustering is shown to provide higher clustering accuracy, resulting in more reliable separation and identification of active users in comparison to traditional clustering methods.

Constrained Clustering for Frequency Hopping Spread Spectrum Signal Separation

Parker D. White

(GENERAL AUDIENCE ABSTRACT)

The expansion of technology in areas such as smart homes and appliances, personal devices, smart vehicles, and many others, leads to more and more devices using common wireless communication techniques such as WiFi and Bluetooth. While the number of wirelessly connected users expands, the range of frequencies that support wireless communications does not. It is therefore essential that each of these devices unselfishly share the available wireless resources. If a device is using more resources than the required limits, or causing interference with other's communications, this device will impact many others negatively and therefore preventative action must be taken to prevent further disruption in the wireless environment. Before action can be taken however, the device must first be identified in a mixture of other wireless activity. To identify a specific device, first, a wireless receiver must be in close enough proximity to detect the power that the malicious device is emitting through its wireless communication. This thesis provides a method that can be used to identify a problem user based only off of its wireless transmission behavior. The performance of this identification is shown with respect to the received signal power which represents the necessary range that a listening device must be to identify and separate a problem user from other cooperative users that are communicating wirelessly.

Contents

List of Figures

List of Tables

List of Abbreviations

AWGN Additive White Gaussian Noise

DFT Discrete Fourier Transform

DoA Direction of Arrival

FHSS Frequency Hopping Spread Spectrum

FN False Negative

FP False Positive

IOU Intersection Over Union

ISM Industrial, Scientific, and Medical Radio Band

NMI Normalized Mutual Information

QPSK Quadrature Phase Shift Keying

RSS Received Signal Strength

RV Random Variable

SEI Specific Emitter Identification

SNR Signal-to-Noise Ratio

STFT Short Time Fourier Transform

TF Time-Frequency

ToA Time of Arrival

TP True Positive

V2X Vehicle-to-X

Chapter 1

Introduction

1.1 Motivation

Areas such as healthcare, Vehicle-to-X, industrial automation, smart homes, personal devices and emergency response are all experiencing unprecedented growth in the number of physical devices that are active in wireless communication systems [1]. As the number of these devices continues to grow, and the availability of the finite spectrum becomes more and more limited, maintaining cooperative use of the spectrum becomes crucial for the interoperability of these multi-user systems. While there are additional means of communication tools and frequency bands being developed and implemented to support this increasing spectrum activity, many of these future and current devices will rely on traditional multiple access techniques, such as FHSS, and will operate in unlicensed frequency bands [2]. While many devices, as in the areas of V2X and healthcare, are cooperative, not every device will be communicating it's intent on using the shared spectrum. This leads to a mixture of known and unknown devices all operating within a limited frequency band. This opens the possibility of unknown and uncooperative devices operating outside of the regulations enforced by the FCC in [3] with restriction on Bluetooth in [4], for example. This malicious activity can result in a reduction in communication possibilities for other cooperative users. External monitoring of a shared spectrum over time is therefore necessary in maintaining cooperative and collective use of multi-user frequency bands.

There is also military interest in a mix of users operating within a frequency band of interest. This knowledge can provide advantageous intelligence of the number of active communicators in an otherwise unknown environment. Among these communicators, spectral analysis can lead to the identification of users in an environment which can aid in friend or foe determination. Also, if there is interest in a specific type of user, the identification of this user could lead to activity tracking once a communication profile has been established.

In either commercial or military applications, the identification of a user of interest first involves the separation of signals received in a time and frequency observation window, followed by the relation of these signals to their respective transmitters. However, a commonly used multiple access technique such as Frequency Hopping Spread Spectrum (FHSS) can make this grouping of signals difficult, since in this communication technique, the frequency content is spread across a bandwidth much wider than the bandwidth of the information bearing signal. FHSS communication spreads the frequency content by pseudo-randomly changing the frequency at which data is being modulated. In the commercial application, this spreading of information can reduce the interference caused by a source operating on a single, stationary, frequency. Changing frequency of the information bearing signal also reduces the ability to detect and track the transmitter over a period of time, which is desirable in a military application.

The detection and separation of FHSS signals is still actively being researched and has previously been approached in several ways. Hardware fingerprinting, or SEI, uses RF transmitting chain imperfections such as a transient behavior, quadrature errors [5], amplitude clipping, and/or self-interference [6] to separate sources. While this hardware specific information may be effective in user identification, these hardware imperfections are difficult to calculate and obtain [7]. Geolocation techniques, such as TDOA, DOA, or RSS location involves the cooperation of multiple receiving elements, or the use of a directional antenna,

to determine possible source locations in 3-D space as shown in [8],[9], and [10] respectively. However, nearby spectrum users and multi-path effects complicate source separation based strictly on location. Another approach to separate FHSS sources is based on the use of generalized parameter sets that can be estimated with just the reception of the source signals. Prior works that make use of these parameter sets are typically modeled as a classification problem where there are a set number of known sources that can be recognized from previous learned behavior. Other works require precise knowledge of transmission times or center frequency [11], which are typically not available in a reception environment with a number of unknown sources.

With the clear limitations of current approaches to separating mixtures of FHSS signals in a band of interest, the work in this thesis provides a solution to this problem by:

- Presenting a novel signal separation clustering solution utilizing constraint based clustering on a mixture of unknown sources without needing prior transmission information.

- Demonstrating the implications of signal detection on raw data using several image segmentation techniques, and showing how this detection can influence the performance of constraint-based clustering for FHSS signal separation.

As a result, FHSS signal parameters derived from signal detection are successfully used in signal separation, which is supported with the following simulation that covers the generation of FHSS signals, the detection and estimation of these signals, and finally the separation and identification of previously unknown sources operating within a bandwidth of interest over time.

1.2 Outline and Contribution

Chapter 2 first provides necessary background covering traditional signal separation techniques applied to stationary signals. Two methods including RF fingerprinting and localization techniques will be discussed, along with the difficulty in applying these techniques to FHSS signals. The next section covers the background describing the relevant properties of FHSS signals, providing both the modulation characteristics as well parameters that will be used for signal separation in Chapters 3 and 4. Previous methods used to separate FHSS sources and the limitations of each will also be provided in this section to further demonstrate the need of a unique solution to this problem.

Chapter 3 presents the novel application of a constraint based clustering algorithm on FHSS signal separation, and compares this method to classic clustering techniques. To do so, first, realistic parameters sets for a range of FHSS sources are generated, and then using a modeled detection stage, an estimation error is applied to each signal's parameter set. Next, separation performance of each clustering algorithm is compared as a function of decreased signal SNR, which is represented by an increase in parameter estimation error. Several conclusions can be drawn from the analysis in this chapter. First, the use of detection based parameters to separate FHSS signals shows promise with the modeled estimation error outlined in Chapter 2. Second, the use of background knowledge in the semi-supervised constrained clustering algorithm provides a more accurate separation solution compared to traditional clustering algorithms.

Chapter 4 develops an end-to-end simulation that begins with FHSS synthetic signal generation and concludes with the separation of FHSS signals sources based on actual detected and estimated signal parameters. First, raw data is generated for a number of FHSS sources operating in a time-frequency observation window. Next, a realistic detection stage using image

processing techniques is performed on a time-frequency waterfall image that was generated from the raw data. Finally, signal separation is performed with the constrained clustering algorithm presented in Chapter 3. The result of this analysis leads to three primary contributions. First, three unique image segmentation techniques are compared based on their ability to detect FHSS signals while reducing false detections of background noise. Next, the implications of each stage of detection is discussed in terms of the impact it may have on signal parameter estimation error and therefore signal separation accuracy. Then, the performance advantage of using background knowledge in the constrained clustering algorithm is further validated with the use of realistically detected data.

Finally, in Chapter 5, concluding remarks and a discussion of future work is provided. In this chapter, major takeaways will be discussed covering both the advantages and limitations of the FHSS signal separation solution developed in this work. Further, a discussion of future steps to improve this approach will be considered.

1.3 Publications

Conference Papers

- Parker D. White, R. Michael Buehrer, William C. Headley, "FHSS Signal Separation Using Constrained Clustering." *(In review MILCOM, 2019)*

Chapter 2

Background

2.1 Signal Separation

Signal separation involves dividing a set of wirelessly received signals into groups based on similarity, and then relating each group to a specific transmitter. This process first involves identifying separable characteristics that are similar for one transmitter and different from other transmitters, then using these properties to relate unknown signals to the transmitter that emitted these signals. These characteristics commonly stem from modulation type or radio frequency hardware imperfections that are consistent for subsequent transmissions from a single user. Geographic properties such as DoA and ToA can also be used as a means of differentiating sources. The following subsections discuss these separation techniques, as well as the benefits and difficulties in using each for both stationary and FHSS signals.

2.1.1 RF Fingerprinting

RF fingerprinting techniques used in the works [5, 6, 7, 12, 13] use I/Q representation of signals to identify transmission hardware imperfections caused by manufacturing inconsistencies. These imperfections include phase error, frequency offset, and I/Q imbalance. Phase error occurs when a transmitter and receiver are not perfectly synchronized, which causes symbols to arrive at shifted locations in the I/Q plane. Frequency offset can occur

when a transmitter and cooperative receiver have minor differences in their local oscillators, which continuously shift the I/Q symbols from their original optimal carrier frequency. I/Q offset is caused by differences in both In-Phase and Quadrature oscillators which cause non-orthogonality in the I and Q channels. While all of these modulation imperfections can be used separately or collectively to accurately identify and track a user [7], they require expert selection of features that are specifically chosen for a single set of transmitters. This expert dependent selection criteria, as well as the difficulty in estimating these parameters [7], makes this type of signal separation difficult when applied to a variety of unknown sources which are likely to be using an unlicensed frequency band.

An alternative approach to signal separation with hardware fingerprinting is use of transient features. These features, used in the works [6, 14, 15, 16, 17, 18, 19] are derived from the transition periods when an amplifier on the RF transmitting chain increases or decreases output power that occurs during data modulation. Due to manufacturing inconsistencies, every amplifier may produce different transients features that can be used to identify a specific transmitter. Examples of transient features, as listed in [17], are transient length, normalized amplitude variance, number of peaks in transition, discrete wavelet transform properties, and difference between normalized mean and normalized maximum value. The use of these features has shown to be a very accurate way of identifying a *known* device [20]. With a fingerprint tied to a single device, it can reduce the effectiveness of impersonation attacks, as discussed in [18, 19]. However, without this prior knowledge, unknown users that appear to these classification algorithms cannot be successfully separated, especially if the transceiver belongs to the same manufacturer [21, 22]. In addition, the transient features chosen are non trivial to calculate in a noisy environment [7], and detecting the start and stop transients alone has proven to be a difficult task [23, 24].

Overall, this RF fingerprinting process is treated much like a classification problem, which

requires a set of known devices, and hand picked characteristics that can be used to successfully separate and identify these known signal sources. In a scenario where there are many unknown users in a shared band, this information will most likely not be available. Also, calculating RF specific characteristics for some types of digital modulation techniques, like spread spectrum, is often not realizable. The widespread of frequency content of FHSS signaling contributes to two major difficulties in applying RF fingerprinting to FHSS signals. First, some of these RF features require high resolution in either time, frequency, or both domains. Since FHSS signals are often detected with a Time-Frequency (TF) waterfall over a wide band, this resolution is restricted by the Discrete Fourier Transform (DFT) calculation, which presents a trade off in both time and frequency resolution. Another difficulty in calculating these features arises from the large observation interval required for adequate feature estimation [7]. The potentially short duration of hopping signals may not allow for such lengthy calculations. For these reasons, different characteristics must be observed and used as a means to separate FHSS signals. The next section will discuss what FHSS communication is, and steps that can be taken to separate a mixture of unknown frequency hopping sources.

2.1.2 Localization

Another method that can be used to separate signals involves determining the location where transmission occurred. There are several means of measurement that can help in location determination including ToA, DoA, and RSS [25, 26, 27]. Measurements from multiple cooperative nodes at known locations canbe used in coordination to approximate transmission locations. A couple of common methods used in this location approximation is Trilateration and Triangulation.

Trilateration, which is the main method for GPS location, involves using ToA, or RSS measurements, to approximate a transmitter location. If ToA is used in trilateration [28, 29], the receiving nodes must know precisely when the original signal was transmitted. With this information, by measuring the time it takes to receive the signal, a circumference of the possible distances can be drawn around every receiving node. Alternatively, in a more blind scenario, RSS can be used to estimate sensor locations [30, 31, 32, 33]. When trilateration is used, knowledge of Friis transmission attenuation is used to create a power based location circumferences around every node. When each node creates this circumference, intersections of the circumferences can be used to estimate the transmitter location. For two dimensional localization at least three non-collinear nodes must be used. If three dimensional location is required, four non-collinear nodes must be used to determine an approximate location. However, the use of ToA and RSS are both subject to noise and estimation error [34]. ToA measurements are subject to clock drift [35], propagation speed variability due to adverse weather conditions [36], and multi-path effects [37]. Every one of these effects, as well as the combination of them, causes shifts in travel time estimates that are used in ToA triangulation. This leads to regions of uncertainty in terms of initial transmitter location. Similarly, RSS measurements are sensitive to channel noise including multi-path, shadow fading, and interference. All of these effects can collectively result in signal amplitude variation that can change based on the time, and location, of the transmission. Without accounting for these effects by using only free space attenuation, localization precision will suffer, resulting in large regions of uncertainty that may not be able to distinguish sources located within this error region.

Another method of localization is the use of Triangulation. Triangulation relies on the measurement of DoA to approximate transmission locations [38, 39, 40, 41]. As discussed in [38], DoA is typically measured with the phase difference of two or more antennas on

an array. Another method to estimate DoA uses beamforming or a directional antenna to measure the angle of highest received power. With an estimated DoA, each receiving node will have a bearing line in the direction of the received signal. The intersection of multiples node bearing lines can then be used to localize the transmission location. In a two dimensional case, only two nodes are required to approximate transmission location. With increased numbers of receiving elements, the accuracy of this localization can increase as well, at the expense of receiver cost. DoA localization alone is subject to multiple errors due to imperfection in determining the direction of arrival. These errors are typically caused by phase errors, multi-path reflections, and a restricted beam resolution. Triangulation with each receiving nodes direction of arrival is dependent on the precision of these estimates. Also, this method requires several nodes to be in receiving range of a transmitter of interest. If one node fails to receive signals from a transmitter of interest, the use of bearing line intersection to approximate location would not be possible.

Overall, triangulation and trilateration can be used with estimations of ToA, DoA, and RSS. Multiple nodes are required to make these measurements and each can have variable estimation error. This estimation error leads to localization error which may only be able to provide a possible region of location, rather than a precise pin point location of the transmitter. With this in mind, multiple users of the spectrum that are operating in close proximity may be difficult to distinguish using these localization methods alone. In addition, ToA measurement requires cooperative sharing of transmission timing information, which may not be available in non-cooperative or blind scenarios.

2.2 FHSS

The previous section discussed the difficulties in applying RF fingerprinting and localization separation techniques to sources using FHSS multiple access communication. The following section will describe FHSS communication to provide sufficient background in further understanding the difficulties of separating this multiple access technique. Next, the parameters that can be used to separate FHSS signals, derived strictly from signal detection, will be discussed in Section 2.2.1. The estimation error associated with each of these parameters will then be covered in Section 2.2.2. Finally, signal separation techniques using the parameters of FHSS signals are discussed in Section 2.2.3, along with the areas in which prior works lack in solving this problem.

2.2.1 Modulation Characteristics

Frequency Hopping spread spectrum is a spread spectrum digital communication technique used commonly in multiple-access communication systems. Multiple-access is a technique that allows multiple spectrum users to operate in the same band efficiently. With an increasing number of users sharing a band, interference avoidance becomes a priority in this shared bandwidth. Frequency hopping spread spectrum signaling is used with this interference avoidance in mind. Specifically, FHSS can avoid interference due to jamming, other users, and even self interference caused by multipath propagation. FHSS achieves this by spreading the information signal over a band much wider than the signal's bandwidth. This spreading is achieved by pseudo-randomly varying the center frequency of the information bearing signal. This spreading is created by the combination of the pseudonoise sequence generator and frequency synthesizer, as can be seen in Fig. 2.1. These pseudo-random center frequencies are used to modulate the information signal. This information begins as a

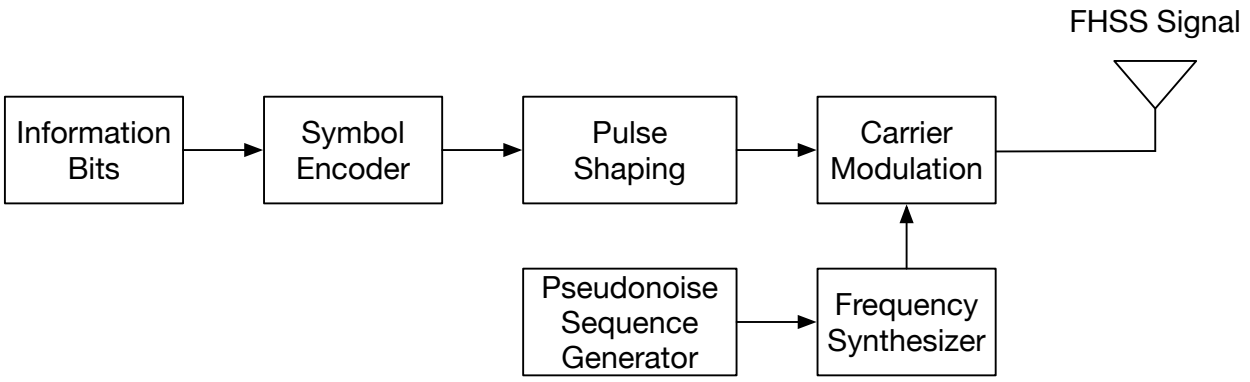

Figure 2.1: Block diagram showing the necessary steps to transmit a FHSS signal.

sequence of information bearing bits, which are then encoded to symbols by mapping short sequences of bits to symbol locations on the IQ plane. Then, a pulse shaping function is used to generate a stream of pulses with magnitude based on the input stream of symbols. Finally this signal is then modulated with the previously discussed random carrier frequency. As defined in the standard in [4], there are typically a set number of channels or frequency slots that this center frequency can take. The center frequency can also be hopped from one slot to another at a range of speeds. For a single user, however, this swapping usually occurs at a constant rate. This means that the dwell time, or time one information signal spends in a certain frequency slot, is typically consistent per user. An example of single user activity can be seen in Fig. 2.2. Another property of Frequency hopping signals is their use of the spectrum. Bandwidth is a commonly used metric in stating the amount of the frequency band an information signal is currently occupying. This bandwidth is caused by the speed at which information in transmitted. In typical digital communication techniques, bits are mapped to symbols which then are modulated with the carrier frequency to transmit information. The rate at which the symbols change, due to the bit rate, determines how fast the signal pulses must vary which causes a higher use of bandwidth. For a given user, the bit rate and therefore bandwidth is typically consistent between hops. The choice of pulse shape used in frequency hopping also determines the spectral characteristics of the signal.

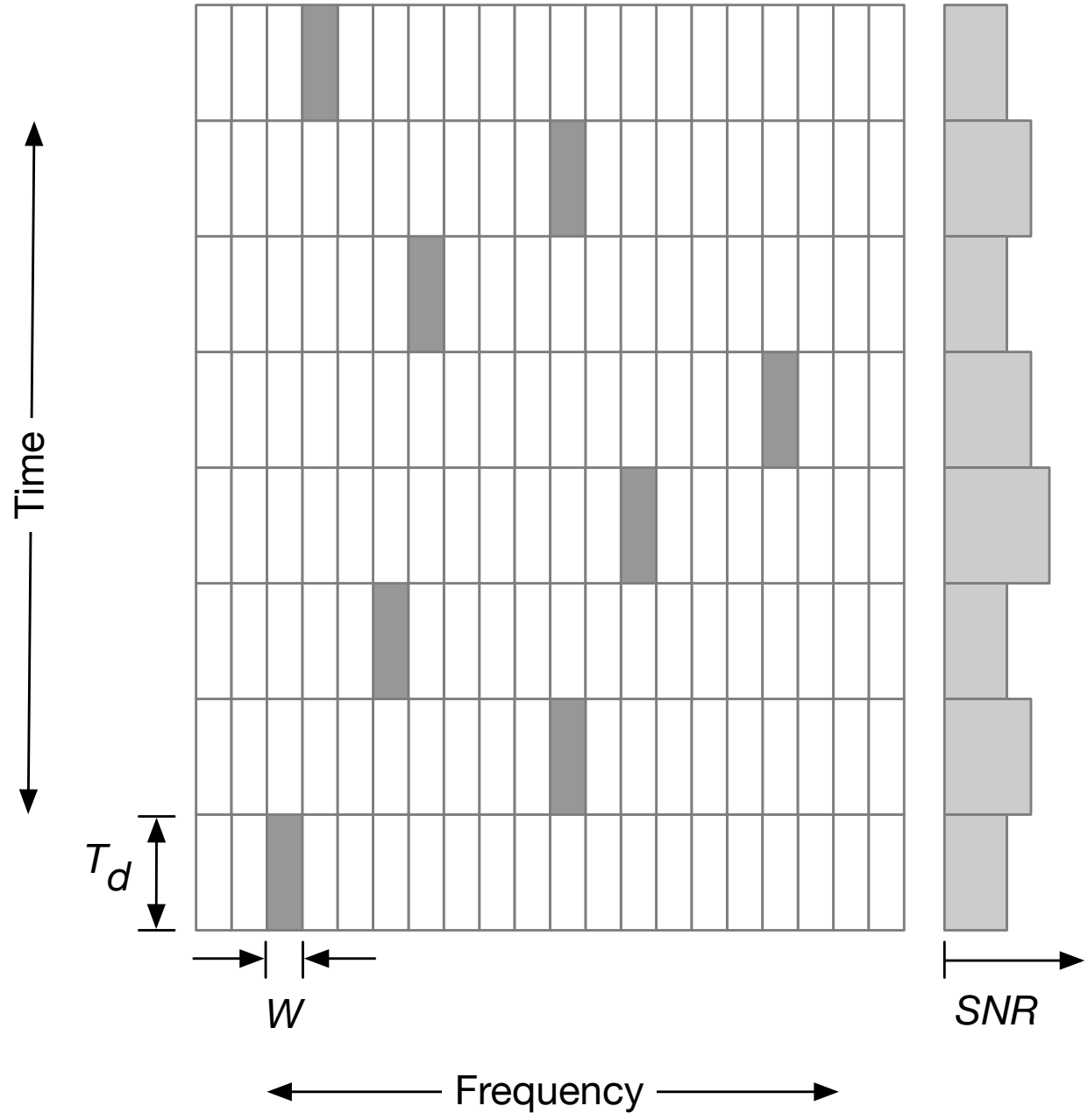

Figure 2.2: Example of the consecutive hopping characteristics of a single user displaying the random shifts in carrier frequency over time. In this example, T_d, W and SNR represent the dwell time, bandwidth, and signal to noise ratio of each signal.

Depending on this choice, the sharpness of the pulse can vary, and therefore give the signal a defined characteristic when observed in the frequency domain. Another factor derived from modulation characteristics is the transmit power. At a receiver, this transmit power is seen as the RSS. While channel effects can vary this transmit power, in a fairly constant environment, such as an AWGN channel, the transmit power can be assumed to be fairly consistent between signals sent by the same user. In a shared band, users operating both cooperatively and non-cooperatively will have variable modulation characteristics, and will also be transmitting at various times. Since the carrier frequencies are varied pseudo-randomly, collisions in time, frequency, or both, are likely to occur.

2.2.2 Detection-Based Parameter Estimation

Actual estimation techniques of FHSS signals vary, some techniques covered in the following works have been used in order to estimate FHSS characteristics such as hopping instants, bandwidth, symbol rate, center frequency and received power. The works [42, 43, 44] show estimation of hop timing and center frequency when periodic hops are present. Timing estimates of aperiodic hoppers are presented in [45], where antenna arrays are used as the receiving elements. This work is expanded to consider collisions in [46], and a blind approach in [47] with an expectation-maximization algorithm used to estimate hop timing and center frequency. Multiple measurements of bandwidth can describe spectral properties, like pulse shaping, of a received signal. One type of bandwidth is total bandwidth or occupied bandwidth, which includes the entire range of frequency content of the received signal, typically measured by -20dB of signal energy [3]. There is also necessary bandwidth, B_n, or 3dB bandwidth which describes the bandwidth of the signal at 3dB below the peak energy. Bandwidth is estimated in the work [48] from the cyclic spectrum of signals, where sub-Nyquist sampling rates are considered. The works [49] and [50] estimate symbol rate, however considering the

high correlation between symbol rate and bandwidth for the considered signal types, the use of both parameters is unnecessary. While power estimation is at risk of many channel effects such as multi-path and shadow fading, it may still be helpful in a relatively constant environment. Estimates of power can be improved with an increased number of receivers, as well as a closer proximity to the receiving elements. [27] discusses this power estimate and gives a typical range based on [51].

The parameters discussed in this subsection covered properties of FHSS signals that are derived strictly from detection. The properties including hop rate, spectral shape, and power, are typically consistent for each transmitter across several transmitted hopping bursts. Although there is error in estimating these parameters, the similarity of these feature sets can be used to first group received signals, and then assign each group of signals to a suspected transmitter. The following section will discuss previous methods that have performed this, and the issues faced with each method.

2.2.3 Signal Separation

Previous works have used mixtures of similar hopping parameters to solve different tasks. The work in [52] uses center frequency, bandwidth, power, and duration to classify the type of hopping device currently using the unlicensed ISM band. The method in [52] requires prior knowledge of typical hopping characteristics and is treated much like a classification problem, where there are a set number of previously learned source classes. In high traffic unregulated frequency bands, the knowledge of specific device characteristics is typically unavailable. The work in this thesis will not rely on previously seen and learned hopping patterns of specific devices, but will utilize the similarity of signal transmissions based on the previously discussed signal properties. Using this method not only removes the requirement of prior

hopping characteristics for each source, but also does not bound the number of recognizable signal sources. The work in [53] detects non-WiFi interfering devices using parameters such as duration, bandwidth, center frequency, spectral signature, timing signature, duty cycle, and pulse spread. This work strictly uses common WiFi hardware for signal detection, but does not account for the possibility of higher quality signal detectors providing more accurate parameter estimation at equal or lower received power compared to more cheaper, and common equipment. The current work will measure the impact of device capability on source separation by clustering across a range of receive SNRs. The work in [11] uses frequency offset measurements as well as precise start and stop times to gain information about hopping signals in a band of interest. While the current work also assumes high fidelity in timing estimation, the current work does not require a reference to specific or ideal center frequencies that were attempted to be transmitted. The work [11] also requires precise knowledge of transmission frame period or dwell time. They require that every source has equal dwell times, an assumption that for non-Bluetooth devices will most likely not hold. This work also assumes every frequency hopping signal is received at 20dB SNR. Due to the fast and large spreading properties of FHSS signals, a high SNR may be required to achieve accurate estimation, however the authors of [11] do not evaluate their system on degrading estimation quality. The current work explores how this estimation quality may impact separation quality. The work in [54] uses multiple receiving elements to detect hop instants and performs blind localization of hops. While this work does perform parameter estimation of frequency hopping signals, and proceeds to track a single user, it does not consider multiple user separation and characterization. Similar to prior work, the current work utilizes the estimated parameters listed in Table 2.1 to perform source separation. The current work does not require prior knowledge of device specific hopping behaviors, and can also separate a variable number of sources and signals. These advantages come in the novel application of pairwise constraint-based clustering. Unlike prior work, the effectiveness of

Table 2.1: FHSS Parameters Used for Signal Separation

	L-Bound	U-Bound	Clustered	Refs
Start (s)	0.0	-	N	[14-19]
Stop (s)	-	0.5	N	[14-19]
Dwell (s)	3.125E-4	0.4	Y	[14-19]
B_n (kHz)	50	250	Y	[20-22]
B (kHz)	250	500	Y	[20-22]
RSS (dBm)	-10	10	Y	[23,24]

this clustering will be demonstrated with various receive SNRs to demonstrate the impact of variable detector capabilities on source separation.

2.3 Summary

This chapter first covered traditional means of separating signals that includes the use of RF characteristics and location information. The use of RF fingerprinting to separate signals when hardware features sets are known *a priori*, can provide excellent identification of these known sources. However, if sources are not known, the features that differentiate observed unknown users becomes reliant on expect picked features. Often times, especially in a widely used, unlicensed band, the prior knowledge of these RF imperfections is not realistic. In addition, the observation interval required to estimate these features may not be available when signals have a short dwell time, which is often the case for FHSS signals. Signal separation based on localization is dependent on estimation of DoA, ToA, and RSS. These estimations can provide regions of locations, but for a number of sources that are in close proximity, additional information about the signal would be required to distinguish sources. The use of ToA also requires transmission timing information that would typically not be available in a band of unlicensed spectrum. These difficulties in both fingerprinting and localization

techniques make the application of these methods to FHSS signals impractical. This chapter discussed the properties of FHSS signals to further demonstrate why the separation of these signals is a difficult with traditional techniques. Alternatively, the separation of FHSS signals using parameters such as hop rate, spectral shape, and RSS has shown promise in prior work. Prior works however require specific knowledge of hop instants, are restricted to a set number of known sources, or do not account for the impact that estimation error may have on signal separation quality. The approach in this thesis is not dependent on a set number of known sources, and can therefore separate a range of sources that have not been previously seen. This work also does not depend FHSS hopping information such as hopping instants or ideal center frequencies that must be known *a prior* in works discussed in Section 2.2.3. However, there are limitation is only using the detection based parameters of FHSS signals listed in Table 2.1. First, a possible issue in strictly using FHSS detection-based parameters in signal separation is the difficulty in separating sources that have very similar parameter sets. For example, if two sources have similar hop rate and bandwidth, the only parameter that may differentiate them is the RSS. Every parameter, or no parameters could be similar for a number of sources, and in the case where all parameters are similar, the estimation process may not provide enough resolution to distinguish sources.

Although there are some limitations in using the approach presented in this thesis, the benefits will be demonstrated with a novel separation technique based on constrained clustering, which does not rely on prior knowledge of signal sources. In addition, the performance of this algorithm will be compared with classical clustering method with respect to increasing estimation error that would be caused by lower receive SNR.

Chapter 3

Separating Signals with Clustering Algorithms

3.1 Introduction

Clustering is the unsupervised (or unassisted) sorting of data points into groups based on similarity. Each data point is typically described with a set of features, and the similarity of the features of one data point and the features of another data point describes in general how similar these points are. Cluster analysis is most beneficial when points within each group of data are highly similar and the groups themselves are very different. As discussed in the previous chapter, the signal separation approach presented in this thesis utilizes the similarities of signals emitted from one transmitter, as well as the differences from signals of other transmitters. Several features of signals including bandwidth, dwell time, and power will be used in the separation approach presented in this chapter. Applying clustering algorithms to FHSS signal separation has several unique benefits. First, the bounds on dwell time, bandwidth, and power, listed in Table 2.1 provide a wide range of possible parameter sets for each transmitter using the unlicensed frequency band. This range provides variation in source characteristics that are beneficial in clustering process. Another benefit of using clustering algorithms is the robustness to the number of input data points, as well as the number of output clusters. Considering this current work is based on shared and unlicensed

spectrum, the number of users at one time, as well as the number of signals each user transmits can vary widely.

Before clustering can be used to separate signals, the FHSS signals must first be detected, and the parameters must be estimated. As discussed in Section 2.2.2, the quality of estimation is largely dependent on SNR and receiver capability. Prior works fail to consider receiver estimation quality on signal separation performance [53], [11]. In this application, higher receiver cost leads to better estimation capabilities, but without measuring the impact that a receiver can have on separation performance, a receiver cannot be properly selected which may lead to higher costs in a real world implementation.

This chapter will consider the impact of parameter estimation quality on signal separation capabilities of several clustering algorithms. In an attempt to combat an increase in parameter estimation error, the novel application of signal background knowledge into a constrained clustering algorithm will be demonstrated. Since FHSS sources transmit one signal at a time, any other signals occurring in the same time window can be constrained to a different group and therefore source. The performance gain of this algorithm can provide better clustering accuracy at lower SNR, which could allow for the use of cheaper receivers in a real application.

To demonstrate the benefits of this constrained clustering approach, an environment containing a range of users, each with random hopping parameter sets will be simulated and discussed in Section 3.2. Next, estimation error based off the works in Section 2.2.2 will be applied to each parameter set, in an attempt to provide realistic estimation error to the signal source analyzed in this current work. Finally, signal separation with traditional and constraint-based clustering methods will be discussed in Section 3.3 and the performance benefits in using constrained clustering will be presented in Section 3.4.

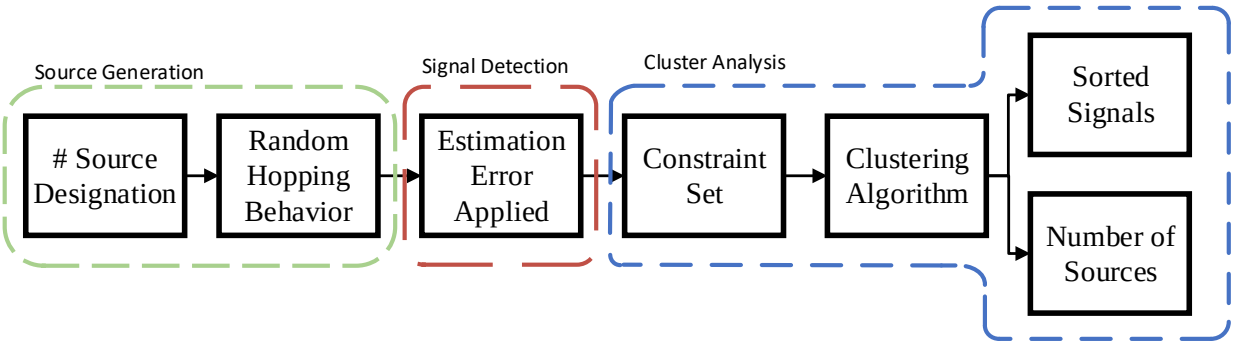

Figure 3.1: The method used to test the novel separation algorithms presented in this work. Starting with random source generation, a signal detection stage with a modeled distribution of parameter estimation error, and finally the clustering algorithm resulting in sorted signals and an estimated number of sources.

3.2 Environment Simulation

The simulation of a mixture of signals first involves the specification of an observation widow. Since there is an available 80 MHz in the unlicensed band discussed in Section 1.1, the observation window used in this work will be 80 MHz. A time window of 0.5 seconds is used to limit computation time. With the observation window specified, the next stage of the simulation involves generating sources that are actively operating within this window. As can be seen in Fig. 3.1, this stage requires the number of sources to be specified and the hopping pattern for each source be randomly generated. This stage of simulation is further discussed in Subsection 3.2.1. Next, a signal detection stage is modeled by assigning an estimation error to each set of signal parameters previously generated and is discussed in detail in Subsection 3.2.2. Finally, the clustering algorithms are described in more detail in Section 3.3, and the application of these algorithms, as well as the results in using them for signal separation is discussed in Section 3.4.

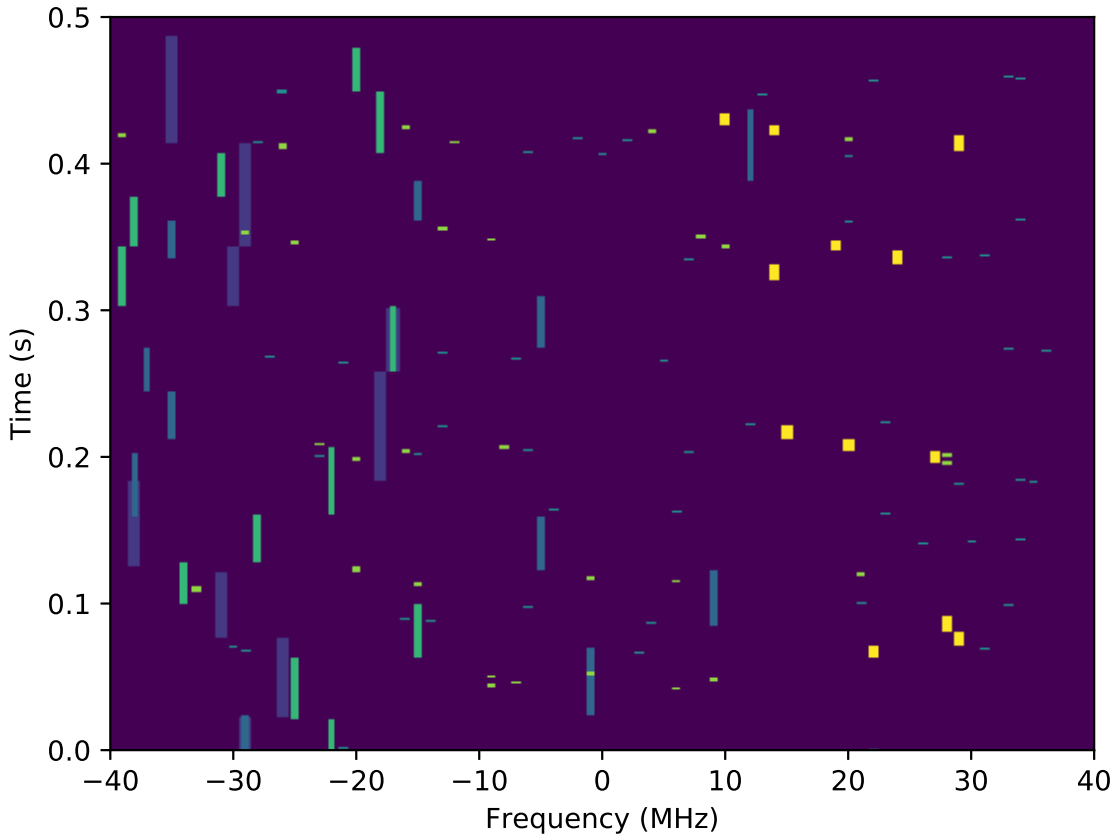

Figure 3.2: A visual representation of the randomly generated parameter sets of 6 active sources, each represented by a different color. The parameter sets are shown prior to any amount of estimation error being applied.

3.2.1 Hopping Behavior

In this scenario, a number of hopping sources will be active at a time, each with randomly selected characteristic parameter sets as defined in Table 2.1. These transmit parameters will be selected within the Bluetooth specifications, [3] and [4], and other than dwell time, will not change over consecutive hops or packet arrivals. The dwell time for each hop will vary randomly between consecutive hops. The hopping bounds are defined and generated within the following categories shown in Table 2.1: bandwidth, number of hopping channels, center frequency, dwell time, packet size, and maximum hops/second. Given these bounds, a range of concurrent users will be active in a time and frequency observation window. The number of channels, dwell time, packet size, and bandwidth are all selected uniformly within the bounds stated. The initial arrival time will be modeled as a Poisson process with an independent arrival modeled with an exponential distribution [55]. The inter-arrival, or arrival time after a source has stopped transmitting, will also be modeled with an exponential random variable. The center frequency will be pseudo-randomly chosen in accordance with the number of allotted channels selected previously. While collisions in this simulation may occur, the collided signals are assumed to be separable with no interference between them. An example spectrum with FHSS sources defined by the previous properties can be seen in Fig. 3.2

3.2.2 Parameter Collection

Prior work discussed in Section 2.2.2 demonstrates that parameter estimation accuracy is highly dependent on the receive signal SNR. Signal properties like start time, end time, and bandwidth become more difficult to accurately estimate. It was shown in prior work that this estimation error scales inversely with received signal power. In the current work,

the magnitude of parameter estimation error will start at no error and be increased, to represent this degrading reception environment. The distribution of this error is modeled as a Gaussian distribution, which serves as a general distribution for estimation error. [56]. The Gaussian distributed estimation error applied to parameters in this chapter is implemented with a coefficient of variation, defined as $E = \sigma/\mu$. This coefficient of variation, E, applies error relative to the mean μ by increasing the variance σ of the Gaussian distribution. This ensures that error is applied uniformly across all estimation parameters, listed in Table 2.1. Realistically, the estimation error for each parameter will largely depend on the receiving device, and estimation algorithm used [53], but for ease of analysis, a uniform error is applied to all parameters.

3.3 Clustering Algorithms

3.3.1 Traditional Algorithms

Many different clustering algorithms have been used in a wide variety of applications and fields. The work [57], discusses many of these algorithms and their history, and also discusses the determination of an appropriate clustering algorithm for a given problem. With this is mind, the clustering algorithms investigated in the current work have been chosen to cover a variety of clustering methods, including density-based DBSCAN [58], graph-based Spectral [59], hierarchical [60], and centroid based [61] clustering. These algorithms will be applied in the typical unsupervised domain where there is an unspecified number of inputs (individual received hops), and a determined number of clusters (signal sources). The performance of each can provide not only an indication of best algorithm for this application but also provides a baseline for further improvement.

DBSCAN

Density based clustering algorithms, like DBSCAN, starts at a data point, and absorbs all neighboring points within some distance threshold specified by the user. The absorbed points will then recursively absorb their neighboring points until no neighboring points exist within the distance threshold. At this stage a new cluster will be created elsewhere, and the process will repeat. Since DBSCAN is dependent on the selection of a distance threshold, often times, a statics threshold value will not provide a clustering solution comparable to other algorithms. Although this threshold must be specified, DBSCAN does not require a set number of clusters to be selected prior to clustering.

Spectral

Spectral clustering is a multi-step clustering algorithm. [59] First, an ϵ-neighborhood, K Nearest Neighbor, or a fully connected graph is computed according to similarity. The next step is to project the data into a lower dimensional space by computing the Graph Laplacian which can be used to find eigenvalues and eigenvectors of the original graph. The last step is to compute cluster assignment with a specified K by assigning points based on eigenvector values. Unlike other clustering approaches like K-Means, spectral clustering does not depend on the shape of the data to be spherical about the mean. This algorithm does implement K-Means after a the data points are projected into lower dimensional space, this means that the number of clusters must be specified prior to clustering.

Agglomerative

Agglomerative clustering is a form of hierarchical clustering in which every point gets assigned to its own group, and similar points are combined until a number of specified clusters

is reached. This hierarchical clustering method is known as the bottom up approach and will provide a benchmark for the use of hierarchical clustering methods in this application. Similar to Spectral clustering this algorithm requires a specific number of clusters to determine when the algorithms should stop grouping points. The determination of the number of clusters is not always available prior to clustering.

K-means

Alternatively, centroid based clustering assigns points based on distance to a cluster centroid, or cluster mean. Algorithms like K-means, group by initializing a specified number of cluster centroids to random data points, and then drawing decision regions between the K centroids or cluster centers. Points are then assigned to the nearest cluster centroid. After assignment, the centroid locations are moved to the cluster mean and decision regions are drawn again and points are reassigned. This process typically minimizes point to cluster distance or inertia. K-means is fairly simple to implement and, in comparison to the previously discussed clustering methods provides the quickest solution. However, the number of K clusters must be specified by this algorithm before clustering. Also, K-means is purely a distance based algorithm, which means distance boundaries work best when data point are distributed spherically around the cluster center.

In addition to classic clustering algorithms, there have been some modifications that make clustering algorithms slightly more intelligent. An example of this is the introduction of constraints to the K-Means algorithm in [62].

3.3.2 Constraint-Based Algorithms

In the signal separation domain, background knowledge may come in the form of a signals time domain properties, direction of arrival, modulation class, among others. Incorporating this information in the form of constraints in the clustering process can improve results when parameter sets become more erroneous. There are several variations of constraint-based clustering, where the majority of these algorithms apply constraints during the clustering process in the form of cannot-link or must-link pairwise constraints. Among the previously listed classical clustering algorithms, K-means provides the most elegant transition to incorporating these constraints into the clustering process [62].

Probabilistic Cannot-Link Constraints

The constraints in the current work are assigned on a pairwise, or point to point, basis. These constraints can require two points to be assigned to the same cluster, as in must-link, or restrict assignment of one point to another cluster, as in cannot-link. In this work, must-link constraint assignment requires absolute certainty that two signals are from the same source. This level of certainty is not available due to the erroneous parameter estimation error used in this work. Cannot-link constraints however, can be assigned more freely based on the assumption that frequency hopping sources only transmit one hopping signal at a time. If two hopping signals occur in the observation bandwidth at the same time and/or frequency, these signals can be strictly assigned to separate sources with a cannot-link constraint. A hard requirement on constraints is enforced in this work, so no two individual signals can be assigned to the same cluster if there is a cannot-link constraint between them. As a result, as estimation error of parameters increases, points may not be validly assigned due to a constraint existing in every other group. If this happens, the point will be assigned into a

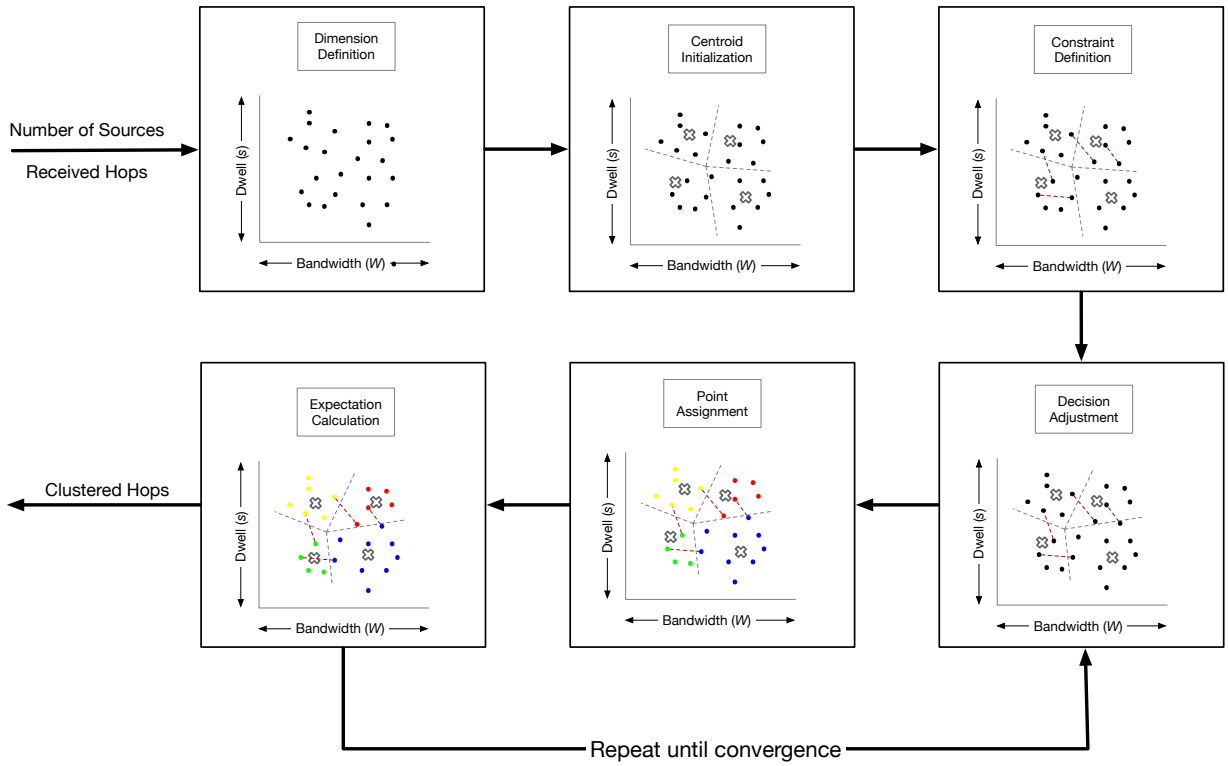

Figure 3.3: A visual representation of the steps of the constraint partitioning clustering algorithm. The progression of the third step to the fourth step demonstrate the effect that constraints can have on decision boundaries.

null set and will be re-included in the next clustering iteration.

3.3.3 COP-Kmeans Clustering

The COP-Kmeans (Constraint Partitioning K-Means) algorithm was first introduced in [62], and was applied on three general University of California Irvine machine learning data sets [63], and in a real world GPS lane finding application. The current work makes novel modifications to the K-means algorithm to incorporate instance level pairwise constraints in the application of signal separation. First, these modifications will be outlined and discussed. Then, a detailed description of each stage of the algorithm will be provided. A visual representation of this algorithm can be seen in Fig. 3.3.

Modifications

While the original COP-Kmeans algorithm in [62] shows obvious benefits, there are some additional considerations that can improve the classification performance and reduce the number of unassigned points. First, the order in which the points get assigned can impact the quality of the assignment. This is due to the lack of decision-making in terms of which point should be assigned to an alternate cluster when two points are cannot-link constrained. To handle this issue, in the current work the data points are shuffled prior to the assignment stage shown by step 2 in Algorithm 1. Cluster center initialization techniques can also improve clustering convergence time and overall performance. Kmeans++ initialization, introduced in the work in [64], is an initialization method which assigns other cluster centers with probability proportional to $P(d)^2$ where $P(d)$ is the euclidean distance from each point d, to the current cluster center. This means that there is a high probability that cluster centers will be initially assigned further distances from each other. Random initialization used in the original work relies on fairly balanced cluster points, since data points are picked as random center locations. For FHSS sources, the hop rate can vary widely, resulting in a range of transmitted source signals. Kmeans++ initialization proposed in [64] does not depend on similar source output sizes, and will be used in the current work instead. The work [62] also does not consider the number of points that are in violation with every other cluster and therefore cannot be validly assigned. In this case, the work states that these points are assigned to a null set. If a large percentage of the data points end up in this null set, the lack of proper cluster assignment may be undesirable, and could also skew performance results with this lower number of considered point assignments. In contrast, the current work includes the null set back into the assignment stage in step 5 of Algorithm 1. This inclusion provides more chances for every point to be validly assigned to a cluster, and thus reduces the number of points unassigned prior to algorithm convergence. In the case of convergence

with points remaining in the null set, these points can either be forcefully assigned to the nearest cluster center, or left out for further evaluation. In later sections, the use of this unassigned point number can actually improve cluster results by running the algorithm on the same data several times, and choosing the best result in terms of minimum unassigned points. This method of clustering evaluation is often used in performance analysis.

Execution

COP-Kmeans functions with inputs D, cannot link constraints Con, and number of clusters K. Each input d_i is a signal hop observed in the time-frequency window. Each pairwise cannot-link constraint is derived from temporal overlaps in the time-frequency observation interval. In this application, very accurate GPS clock timing estimates of start and stop time are assumed, which are used to define these cannot-link constraints between hops. This assumption will be further discussed and validated in Chapter 4. Now, with input signals D, constraints Con, and number of clusters C defined, clustering as in Algorithm 1 and Fig. 3.3 is executed. Once the cluster centers do not change, or no points are assigned to alternate clusters, the algorithm has converged with K clusters containing all points not including the null set. The K cluster assignment represents separate sources in this application, and can be used as a means of identifying typical hopping characteristics of every independent source. As will be shown, accurate cluster assignment increases the uncertainty about an individual source's hopping behavior which is improved overall with the addition of time based constraints.

Algorithm 1 Modified COP-Kmeans Algorithm

procedure COP-KMEANS(D, Con, K) ▷ Data points D, cannot-link constraints $Con_{\not=} \subseteq D$xD, Clusters K

1: Initialize Cluster Centers $C_1 \dots C_k$ with K-means++.
2: Randomly shuffle all $d \in D$
3: **for all points** $d \in D$:
    Assign d_i to nearest cluster c_j s.t. VIOLATE-CONSTRAINTS is **false**.
    If all clusters fail, assign point to left out set.
4: **for each** cluster $c \in C$:
    compute mean of $D \in c_j$ and assign cluster center to new location μ_j.
5: Add left out set to D and repeat steps (4) and (5) until convergence: $\sum_{n=1}^{j} \Delta\mu_n < threshold$

procedure VIOLATE-CONSTRAINTS($d_i, C_j, Con_{\not=}$) ▷ Data point d_i, cluster C_j, cannot-link constraints $Con_{\not=} \subseteq D$xD

1: **for each** $(d_i, d_{\not=}) \in Con_{\not=}$:
2: **if** $d_{\not=} \in C$, **return** true.
3: **else return** false.

3.4 Results and Discussion

The analysis in this section will be performed with respect to the system model outlined in Section 3.2, and outlined in Fig. 3.1. This model simulates what an observable time and frequency window may look like in a realistic scenario. With this in mind, a variable amount of hopping sources may be active at a time, each with independent hopping behaviors. Signals from each of these hopping sources are assumed to be detected, and the spectrum sensing based characteristics defined in Table 2.1 are modeled with some error outlined in Section 3.2.2. After this parameter estimation stage, each hop will be clustered by this set of hopping parameters. This section will demonstrate the effectiveness of clustering these hopping signals based on the similarity and differences of each set of hopping parameters. Classic clustering techniques and modified constraint based clustering are the methods used for this signal separation, as outlined in Section 3.3.

3.4.1 Performance Metrics

The following metrics will be used to quantify clustering quality and outline artifacts that are present as a result of classic and modified constraint based clustering.

Normalized Mutual Information

Normalized mutual information (NMI) [65] is commonly used for clustering performance evaluation [66]. This measure is defined as

$$\text{NMI} = \frac{MI(L, C)}{E[H(L), H(C)]} \tag{3.1}$$

where L is the vector of true labels of the data set, C is a vector of assigned labels as a result of cluster groupings, MI is the mutual information [65] between L and C and is defined as

$$\text{MI}(L, C) = \sum_{i=1}^{|L|} \sum_{j=1}^{|C|} P(i, j) \log \left(\frac{P(i, j)}{P(i)P'(j)} \right) \tag{3.2}$$

where $P(i, j) = |C_i \cap L_j|/N$ is the probability that a randomly selected data point resides in both clusters C_i and L_j and H is the entropy or the amount of uncertainty [65] of either set and is defined as

$$\text{H}(C) = -\sum_{i=1}^{|C|} P(i) \log(P(i)) \tag{3.3}$$

where $P(i) = |C_i|/N$ is the probability that a data point randomly selected resides in cluster C_i. Mutual information can be described as the measure of information, the clustering labels, contains about the true labels. While this measure can mathematically evaluate clustering performance, an additional measure of accuracy is included in this work to give a relatable percentage value in terms of the percent of correctly assigned labels.

Accuracy

A measure of raw accuracy, or purity [67], is used as another performance benchmark. With a set of clusters C, classes L, and N total data points D, accuracy is defined as follows.

$$\text{Accuracy}(C, D, L) = \frac{1}{N} \sum_{c \in C} \max_{l \in L} |c \cap d| \tag{3.4}$$

For every cluster, the maximum number of one source's signals in that cluster, is assigned to be that clusters true label. With this assignment, a percentage can be derived by the number of correct labels and total number of labels.

Inertia

In many clustering algorithms, convergence and performance are measured in terms of inertia. Inertia is defined as the sum of squared distances of N total points D to their cluster's, C, centers μ as

$$\text{Inertia}(C, D) = \sum_{i=0}^{n} \min_{\mu_j \in C} (||d_i - \mu_j||^2). \tag{3.5}$$

A best result in terms of K-means iterations on a data set is defined by a minimum value of this inertia. K-means algorithms therefore seek to minimize this inertia. For constraint based clustering however, the following analysis will show that inertia may not always describe the best algorithm outcome. The addition of the assigned point metric provides another means of determining the best algorithm outcome.

Assigned Points

Some clustering algorithms account for outliers and pass them to a null set while some assign all points regardless of their dissimilarity to any other data points. Classic K-means

will assign all points by the end of the algorithm. Applying hard constraints to K-means, as in COP-Kmeans, can lead to points that cannot be assigned. If there is no cluster that does not violate a constraint, the point will be passed to a null set and will be attempted to be assigned at the next algorithm iteration. Once the algorithm has converged, there may be points that remain in the null set.

3.4.2 Clustering Results

The results of this section are all simulated with a Monte Carlo Simulation. The constants and variable parameters for each Figure will be discussed in their respective sections. The following sections will discuss each result in detail to demonstrate the performance benefit in using constraint-based clustering.

Algorithm Comparison

The clustering NMI of algorithms discussed in section 3.3 are shown in Fig. 3.4. As can be seen, the best performing algorithms without constraints were agglomerative and K-means. For this reason, the K-Means algorithm was chosen as the baseline in clustering performance. For this current work the addition of pairwise constraints, shown with the COP-Kmeans result demonstrates the benefit of incorporating accurate timing information. A closer comparison of the performance benefit is shown in Fig. 3.6, and is discussed in the next section.

Cluster Iterations

For the result in Fig. 3.5, the number of sources is set at 5, and the parameter estimation error is set with a coefficient of variation of 5 as described in Section 3.2.2. Since most clustering

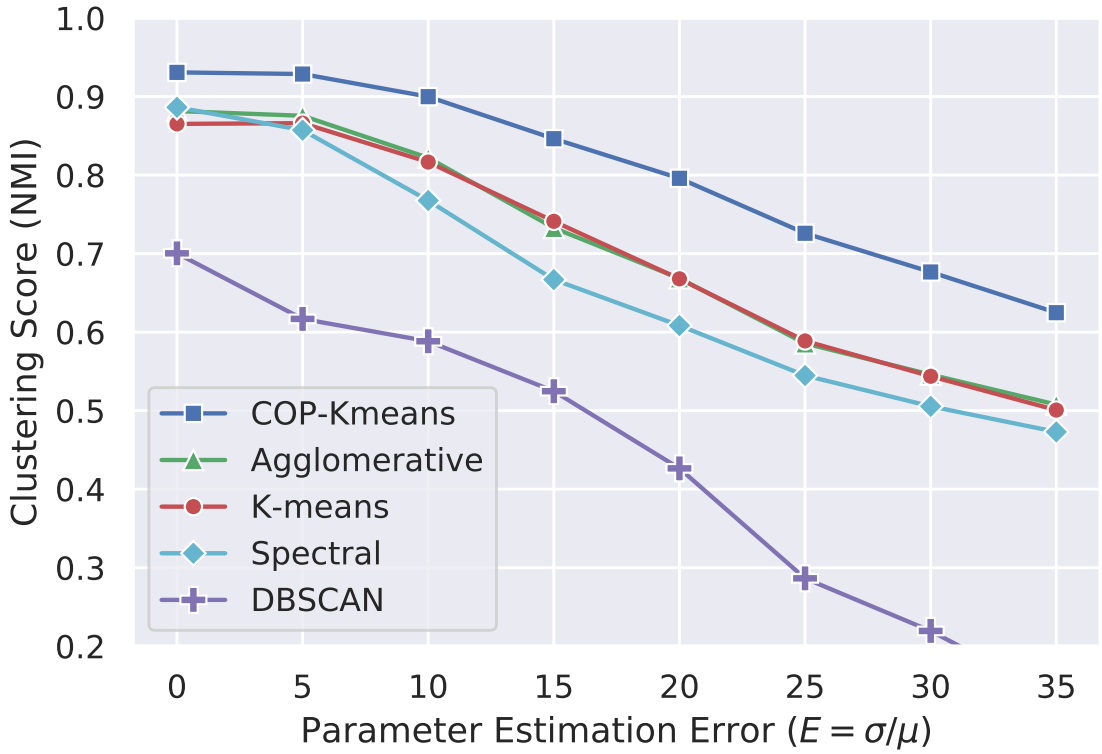

Figure 3.4: Comparison of Normalized Mutual Information across clustering algorithms as a function of increasing coefficient of variation to represent degrading FHSS signal receive SNR.

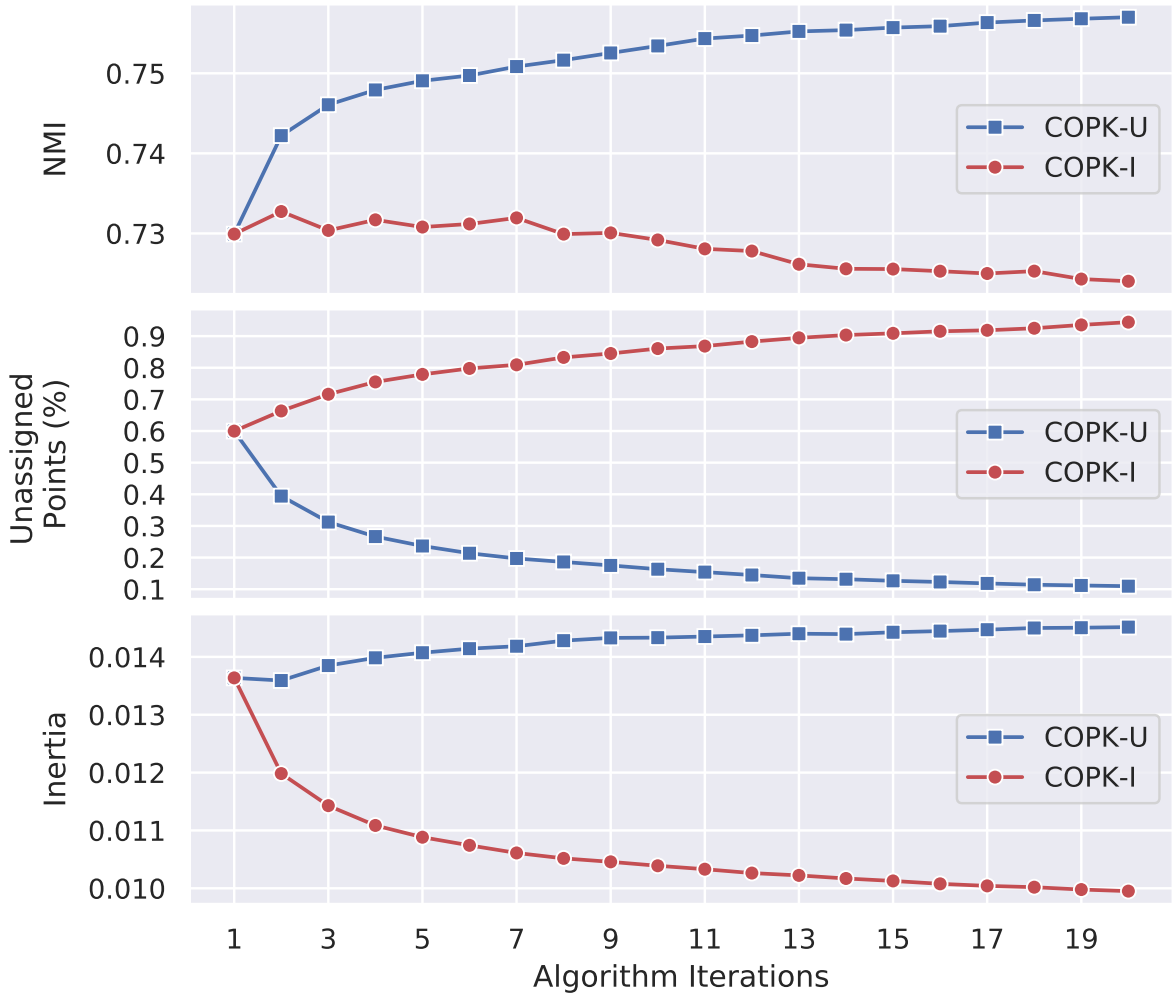

Figure 3.5: Performance difference based on NMI in choosing the best COP-Kmeans clustering iteration with Inertia (COPK-I) vs minimum Unassigned points (COPK-U). In choosing the best algorithm run based on the number of unassigned points, the NMI is shown to increase in the top plot, which represents better clustering accuracy.

algorithms are influenced by initial randomization of cluster locations, these algorithms are subject to reaching non-optimal solutions with few iterations. With more iterations, the best clustering result can be chosen from every clustering run, based on a performance metric such as inertia as in (3.5). The addition of constraints in the COP-Kmeans algorithm allows for additional criteria in selecting the best clustering outcome. In this application, this criteria is in the form of number of unassigned points, or points remaining in the null set. The use of this metric and the impact it has on choosing the best clustering outcome is shown in Fig. 3.5. The line labeled "COPK-U" is the Monte Carlo average clustering result when the best clustering iteration is kept with the minimum points unassigned metric. The line labeled "COPK-I" is the Monte Carlo average clustering result when the minimum value of inertia is kept as the best clustering outcome across 20 iterations. The top NMI plot shows how well the clustering performs based on choosing the algorithm run where either unassigned points or inertia were at a minimum. The performance while choosing the minimum number of unassigned points results in higher clustering accuracy, in comparison to the prior method of using cluster inertia. This clustering artifact was largely ignored in the work by [62], but has proven to have a small but beneficial impact on clustering performance. In the application where a number of clustering iterations are ran for a single set of a data points, the use of this minimum null set metric can aid in determining the best clustering outcome in terms of NMI, and also reduces the number of points that were not successfully assigned to a valid cluster. Although the performance gain in terms of NMI was small, in some instances where there are a large number of signals sources that have similar hopping characteristics, the metric to minimize the number of unassigned points may be a priority. In this application, the number of unassigned point were fairly minimal, and so the performance gained by this selection was apparent, but slight.

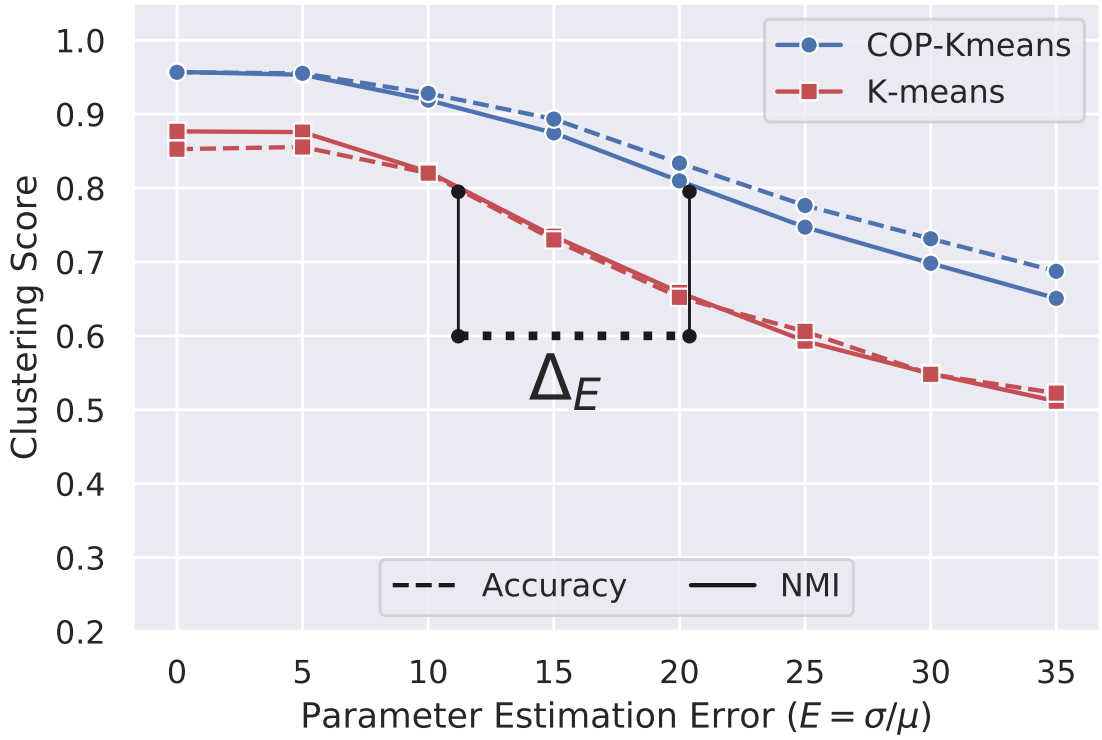

Figure 3.6: The increase in estimation error tolerance of the K-means algorithm with pairwise constraints applied in the COP-Kmeans algorithm.

Accuracy and NMI vs Error

For Fig. 3.4, and 3.6, K was varied from 3 to 8 and the modeled estimation error was swept from 0 to 20. For every random hopping scenario in the Monte Carlo simulation, 20 runs of each algorithm was performed, and the best grouping of signals determined by minimum inertia for traditional clustering methods, and the minimum number of signals in the null for constrained clustering, was selected for comparison. The performance gain in this analysis is expressed in terms of NMI and Accuracy as previously stated in Section 3.4.1. For this analysis, the correct number of K clusters will be correctly assigned prior to clustering. The results in Fig. 3.6 demonstrate the improvement in using pairwise cannot-link constraints in clustering rather than strictly a classic clustering algorithm. The separation distance Δ_E demonstrates the tolerance of parameter estimation error with equivalent performance in signal clustering. Using constraints in clustering increases the error tolerance of nearly 10 (normalized by mean) standard deviations. An increased error tolerance will allow for lower quality measurements that are typically available in cheaper receivers, as well as lower receive SNRs for equivalent clustering performance. Although constrained clustering provides higher signals separation accuracy in comparison to traditional algorithms, the additional computation time that this algorithm requires may not desirable, depending on the application. Additional computation in the COP-Kmeans algorithm may result from the violate constraint check in the assignment stage 3 in Algorithm 1. With a sorted list of clusters based on distance, this stage requires a crosscheck of every other point to verify that no constraints are being violated. If the signal has a long dwell time, and therefore overlaps in time with many other signals, the ”VIOLATE CONSTRAINT” step of this algorithm can become cumbersome.

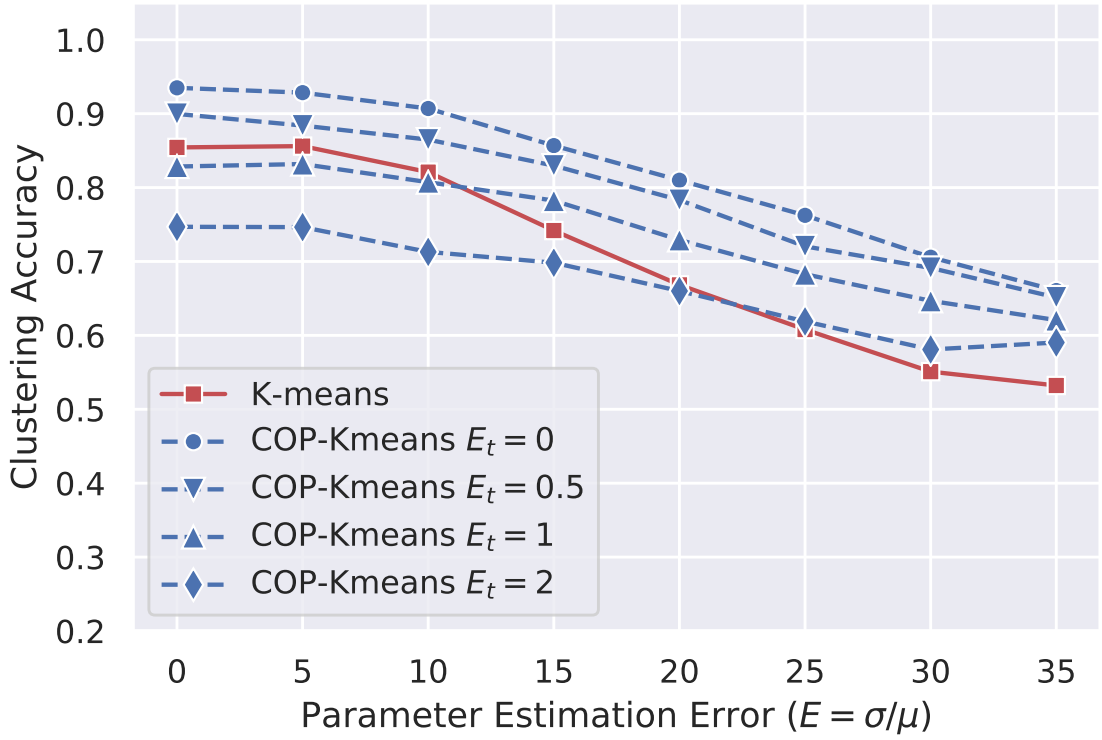

Figure 3.7: The impact of erroneous cannot-link constraints on clustering performance (NMI), caused by hop timing estimation error E_t, as defined in Section 3.2.2

.

Impact of Erroneous Constraints

Up to this point, the results in this current work considered the pairwise constraints as hard and completely accurate. These hard constraints are dependent on very accurate hop timing estimation, which will not be perfect in a realistic scenario. For Fig. 3.7, K was varied from 3 to 8 and the modeled estimation error was swept from 0 to 20. Again, for every random hopping scenario in the Monte Carlo simulation, 20 runs of each algorithm was performed, and the best grouping of signals determined by minimum inertia for traditional clustering methods, and the minimum number of signals in the null for constrained clustering, was selected for comparison. Fig. 3.7 shows the impact of increased start and stop time estimation error which causes false cannot-link constraints assignments. As seen in the figure, the approach in this current work can tolerate some amount of timing errors but with increased amounts of timing error, K-means becomes the better performing algorithm as expected. In scenarios where GPS level accuracy is available for timing estimates, the COP-Kmeans algorithm can tolerate higher parameter estimation error with equal clustering performance compared to classic clustering algorithms.

3.4.3 Choice of K

A drawback of the K-means and COP-Kmeans algorithms are their required specification of the number of clusters prior to clustering. In a blind scenario, where information about the number of sources is possibly unavailable, an accurate estimate of the number of sources is essential in providing accurate signal separation. An inaccurate choice of K will result in wrongly clustered signals, as well as a false estimate of users in the observation spectrum. For the result in Fig. 3.8, parameter estimation error was held constant at 10, and the determination of K was for the best run of 20 algorithm runs. The following sections will

discuss the takeaways of these results.

Silhouette method

First, the Silhouette method [68] is used as a baseline in determining the correct number of clusters or signal sources. The silhouette coefficient is calculated using mean intra-cluster distance and the mean point-to-nearest cluster distance. To use this metric, an algorithm should be swept with a range of K, and the highest silhouette coefficient value is chosen as the estimate of K. This method is applied in both the K-means and COP-Kmeans algorithm runs to determine a K value, as shown in Fig. 3.8. The determination of K using the silhouette coefficient for K-means does not perform well with this data set, most likely due to the small cluster densities with high variance. Using this result alone is not adequate in properly determining the number of clusters for this application.

Minimum K from constraints

The use of constraints in this current work can define a lower bound in terms of concurrent users in a time and bandwidth observation window. Specifically, the minimum number of signal sources can be defined as the number of hoppers occurring at one instant of time during this observation. In Fig. 3.8, the line labeled "Constraint Only" demonstrates the accuracy of choosing the number of K based on these concurrent temporal occurrences alone. As the number of K increases, the probability that every source transmits at the same time decreases. This decrease could be potentially offset with longer observation times, but based on the activity of users, this method may not provide a consistently reliable result.

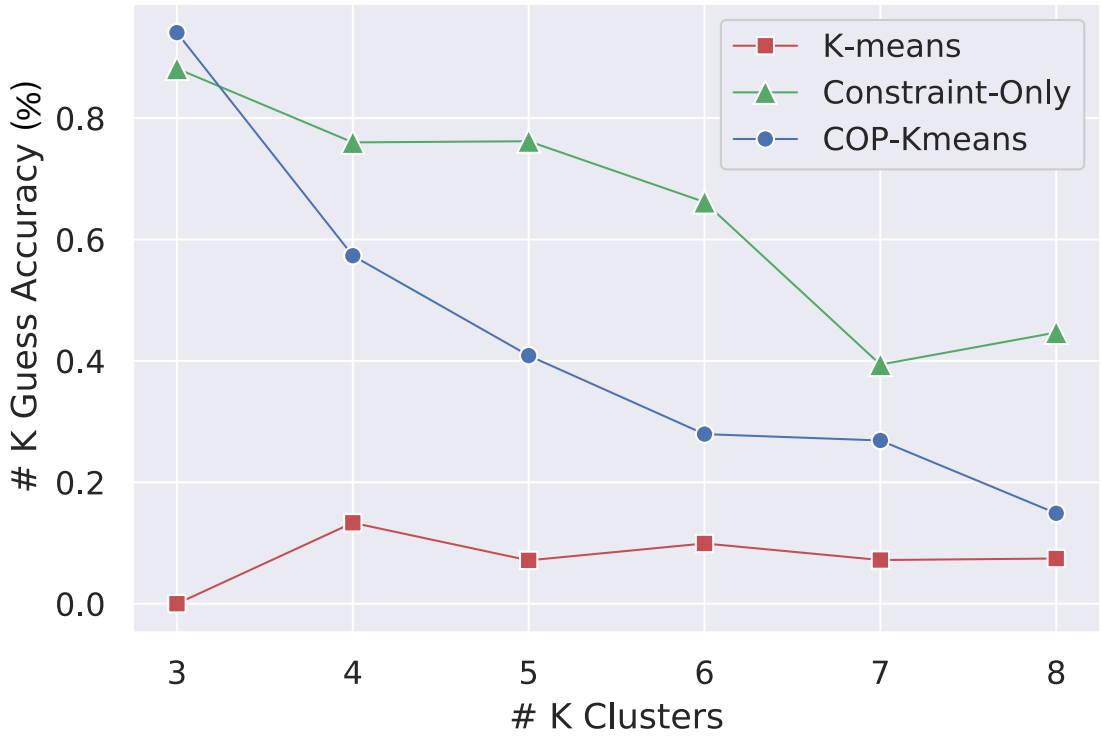

Figure 3.8: Accuracy comparison for several methods of estimating the number of K sources. Where the accuracy of the correctly guessed K value is shown as a function of number of true sources.

Determination of K with COP-Kmeans

Using COP-Kmeans to determine a minimum number of K is shown in Fig. 3.8. Where the accuracy in this determination is defined as Guess Accuracy = (# Correct predictions of K)/(# Total predictions of K). The "COP-Kmeans" line demonstrates the accuracy of choosing the correct K using the constraint derived lower bound along with the silhouette method. Considering the poor performance of the silhouette method, using constraints in the algorithm provides increased accuracy compared to K-means alone, but is ultimately worse than using strictly constraints to determine a minimum value for K.

Summary

Overall the selection of K in this problem has proven to be a difficult task. The common silhouette method of determining K for this application has not given promising results. Selecting K based on the number of overlaps at one instant of time relies heavily on how consistent each source is transmitting data. While the purpose of this chapter was not primarily focused on determining the number of sources in the environment, rather grouping a set of known sources, this topic still needs further research prior to a realistic application. The determination of this K value with further be explored in the next chapter.

3.5 Summary and Future Work

3.5.1 Summary

The current work has highlighted detection based FHSS signal characteristics that can be estimated and used to separate signals. These characteristics include bandwidth, dwell time, hop instants, ToA, and RSS. Other works have estimated and used these parameters for signal classification, but have been limited to recognizing previously seen sources. The current work also removed some prior knowledge in signal characteristics, like transmission time, center frequency and modulation class, and demonstrated that classic clustering methods can be used to separate signals. DBSCAN, Spectral, agglomerative, and K-means clustering were compared for this problem data. K-Means was the best performing traditional clustering algorithm, and was chosen as a baseline and further improved with the novel application of pairwise cannot-link constraints. Since FHSS sources only transmit a single signal at a time, when two signals overlapped in time, they could be assigned to different clusters with the use of cannot-link constraints. Improvement in clustering accuracy allows for higher parameter

estimation error and more consistent grouping of signals to the correct transmitter. In addition, the use of constraints can create clustering artifacts, like number of unassigned, violating points, which can be exploited to better choose a clustering outcome over many iterations. Determining the number of sources operating in an environment was explored, and proved to be difficult in this application. Preliminary work involving the estimation of K based on the maximum number of concurrent signals overlapping at any instant of time was demonstrated. This method was not as successful when the number of sources increased. This is due to the lower probability that every source would overlap at any one time instant. Overall, with accurate measurements of hop timing and erroneous measurements of other parameters listed in Table, 2.1, clustering methods can be used to separate sources, and improved with the additional of background knowledge.

3.5.2 Future Work

The constraints used in the current work are dependent on accurate hop timing estimates. As these estimates become more erroneous, hard constraints can get falsely assigned and decrease clustering performance as shown in Fig. 3.7. Rather than removing constraints altogether when accurate hop timing estimates are unavailable, soft constraints could be used instead. Unlike the strict requirement that constraints must be satisfied in the COP-Kmeans algorithm, soft constraints could be violated with a weighted penalty. This penalty would be applied in the point assignment stage in Algorithm 1, and would be balanced with point to point distance, or similarity. The work [62] is expanded in [69] which incorporates soft constraints to the previous COP-Kmeans algorithm.

Chapter 4

Real World Considerations for Signal Separation

4.1 Introduction

The previous chapter first outlined parameters that can be used to group similar FHSS signals then relate each group to a specific transmitter. To show this analysis, parameter sets for FHSS signals were generated and a modeled Gaussian error was applied to simulate the error that a detection and estimation stage may add. Even with this modeled detection error, signals from a specific source were similar enough to be separated with the use of traditional clustering algorithms. An improvement in this separation process was shown with the addition of constraints based on temporal overlaps of the generated hopping signal parameter sets. This constrained clustering algorithm provided an increased separation accuracy, at the cost of timing estimation sensitivity. The true impact of this sensitivity, as well as realistic bounds on estimation error will be shown with the actual modulation and detection of generated FHSS signals. The process used to perform this analysis can be seen in Fig. 4.1. First, the raw data generation results in time series I/Q data that will be converted into a time-frequency waterfall, which then enables the use of image segmentation methods to detect, and estimate signals. While some image operations have been used in prior work to detect signals within a TF window, the affects that mathematical image oper-

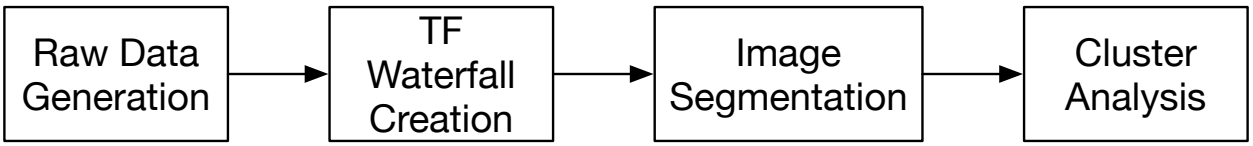

Figure 4.1: The steps of the simulation used in this chapter.

ation may have on signal detection has not been previously discussed. In addition, several methods of detection with image segmentation algorithms will be compared. Further, the novel application of the constrained clustering algorithm will be shown with a full simulation from signal generation, simulated reception, detection and estimation leading up to source separation with the algorithm developed in Chapter 3. The performance advantages of this algorithm will again be compared to traditional clustering methods. The following chapter is organized as follows.

Section 4.2 will discuss the signal generation stage, and the additional properties required to generate these signals with signal processing software. Section 4.3 will discuss the generation of the TF waterfall used to detect signals in this work, as well as the implications this has on signal parameter estimation error and therefore signal separation quality. Section 4.4 will demonstrate several possible detection methods and the implications associated with each. Finally, Section 4.5 uses the best signal detection method from the previous section and shows clustering performance of traditional clustering methods and the improvement gained by using a constrained algorithm on the detected and estimated signals.

4.2 FHSS Signal Generation

In the previous chapter, source properties such as signal bandwidth, dwell time, and power were randomly selected, and assigned to every source operating within a band of interest. For every source, these parameters did not change for consecutive hops, and were defined within

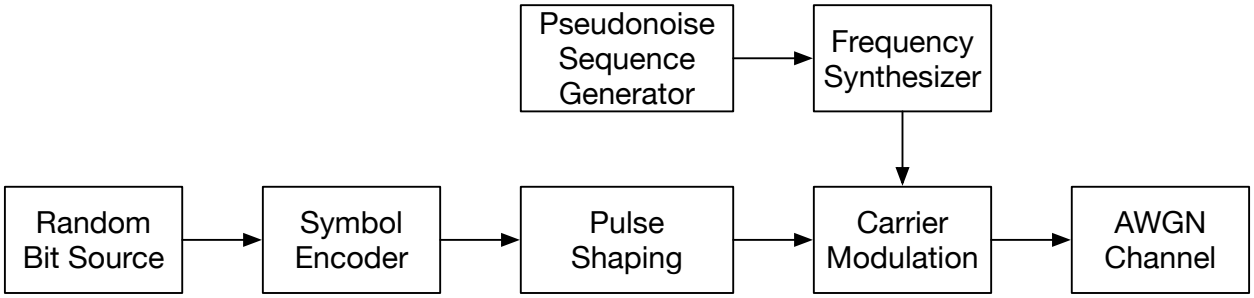

Figure 4.2: The steps used in this work to generate complex data for every hopping source.

the bounds in Table 2.1. Rather than generating real signals, the source separation problem was performed on these randomly selected parameter sets with the addition of modeled error. Alternatively, the generation of FHSS signals in this chapter involves the specification of source properties such as modulation class and order, pulse shaping function, roll-off factor, symbol rate, hop rate, signal power, center frequency, and hop timing, which will be used to generate raw complex I/Q data. Although there are additional properties that must be specified in order to generate raw data, the generation tool that determines the bandwidth, power, and dwell time is the same tool that was used in Chapter 3. The signal processing chain used to generate signals in this chapter can be seen in Fig. 4.2. First a pseudo-random sequence generator creates a stream of bits that is used as the information signal to be transmitted. The next stage involves mapping this stream of bits to symbols residing on the I/Q plane. The number of bits mapped to each symbol is dependent on the order of the modulation scheme. In this work, QPSK modulation was used, where the symbols are evenly spread at different angles on the I/Q plane. The modulation scheme and order primarily effects how efficiently the spectrum is used and since this work is concerned about total bandwidth and dwell times, order and modulation scheme are not prime contributors to signal separation. The next stage of signal generation is pulse shaping, which requires the specification of the filtering function, power, as well as the roll off factor. There is a wide variety of pulse shaping functions that can be used for digital modulation, however, a

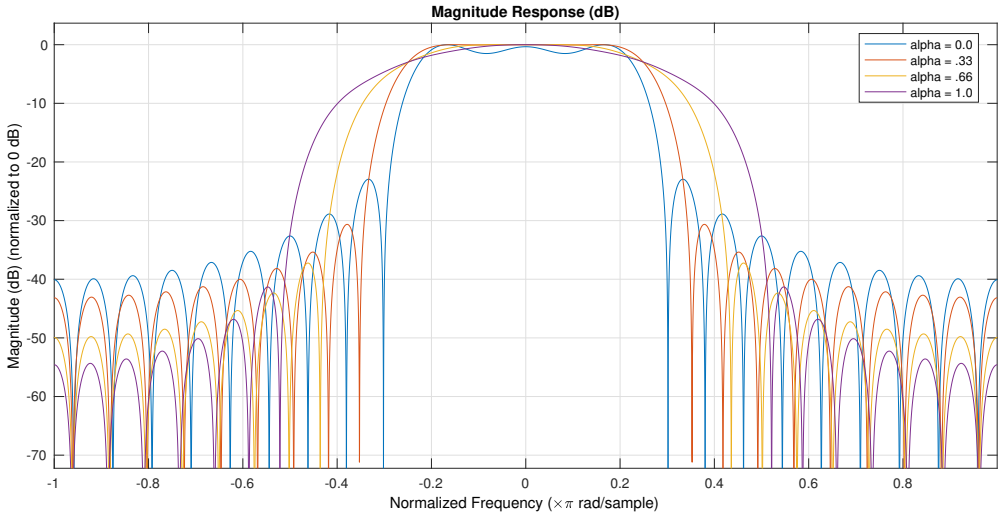

Figure 4.3: Root Raised Cosine filter with various roll off factors to show the possible variations in spectral shape.

common filtering function used is a square root raised cosine filter [70]. For a RRC filter, the roll off factor can affect the steepness of the main spectral lobe as well as the side lobe height, shown in Fig. 4.3. While this roll off factor may have an effect on detection accuracy, the impact of this factor will be considered future work. The next stage of signal generation involves the modulation of the information signal with the carrier wave. Since the signals in this application change carrier frequencies over time, a pseudo-noise sequence generator is used to randomly select center frequencies, which then determines the carrier frequency used to modulate the signals. The rate at which the center frequency changes is determined by the hop rate, which controls how long a signal remains at one frequency, also known as the dwell time. This hop rate is specified by the user and can vary within the bounds stated in Table 2.1. The bit rate controls the symbol rate, and the symbol rate determines the amount of bandwidth that is used as a result of modulation. The bounds for this bandwidth value are listed in Table 2.1 After this modulation stage, the signal is transmitted and after passing through a modeled propagation channel, is received by a listener. In this simulation, a noise floor is designated and the power of the transmit source is determined by the separation in

Table 4.1: Signal Generation Properties

Class	PSK
Order	4
Filter	RRC
Roll Off	0.25

dBm from the noise floor, also known as the SNR. At this point in simulation, a number of sources are modulated using this process with constant parameters listed in Table 4.1. The resultant output is a stream of time series complex IQ data, containing mixed signals that can be extracted with a detection algorithm. The method to perform this extraction and the error associated with it will be discussed in the next section.

4.3 Time-Frequency Waterfall Derivation

Time-frequency signal processing involves using both the time domain and frequency domain concurrently. This approach is attractive in detecting FHSS signals due to the wide band of frequency available for use, as well as the typical hopping behavior that occurs over time. In contrast with traditional methods of using either time or frequency exclusively to detect stationary signals, TF analysis provides a method to distinguish hopping signals that are likely overlapping in time or frequency at any given point. A comparison of these three signal processing techniques can be seen in Fig. 4.4.

There are several methods commonly used in deriving this TF distribution. These methods include, the Wigner-Ville distribution, localized forms of the Fourier transform, filter banks, spectrum gradient, related energy densities and others [71]. An in-depth comparison of TF derivation methods can be found in [72, 73]. The method to obtain the time-frequency distribution in this work was the localized windowed FT. There are several advantage to using this method over the others listed. The windowed FT is simple to implement, robust, and

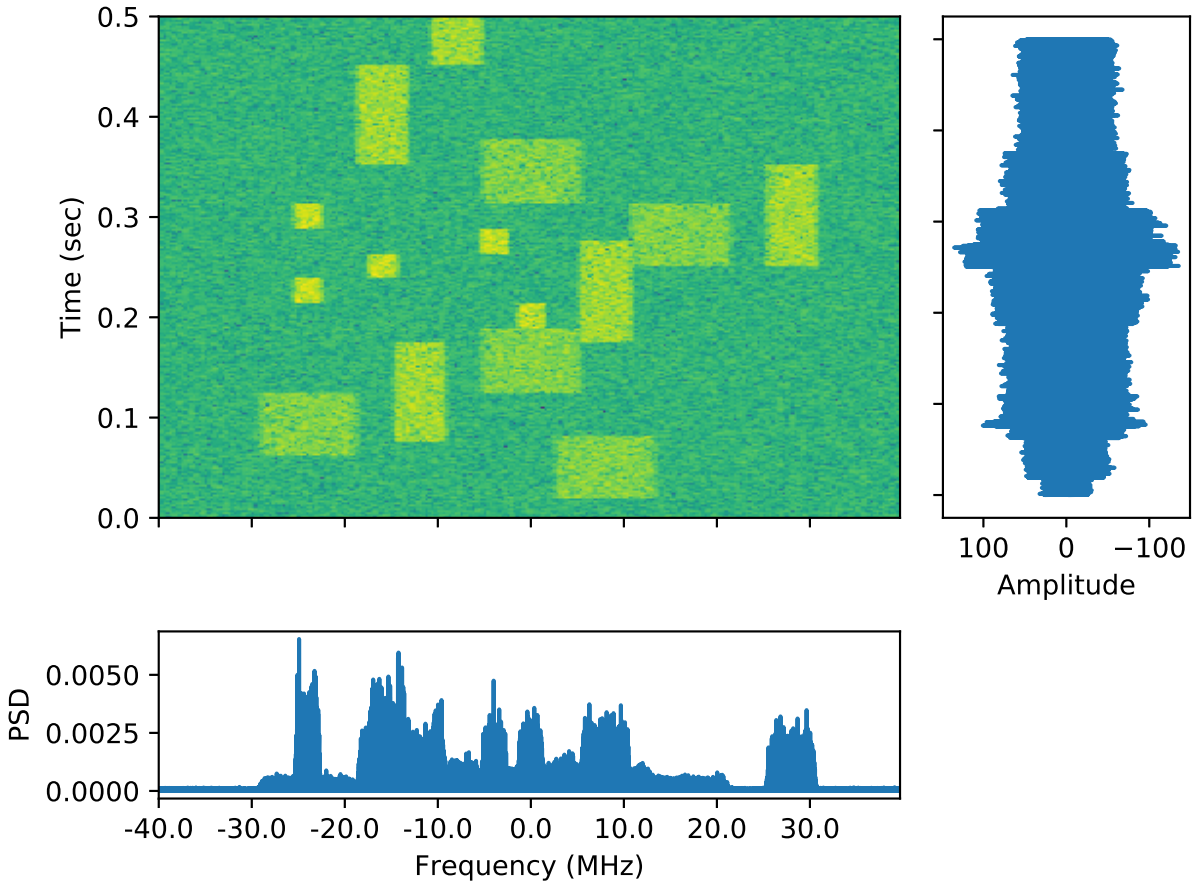

Figure 4.4: A comparison of three signal processing technique using time domain (right plot), frequency domain represented by a Power Spectral Density (PSD) calculation (lower plot), and time-frequency analysis (center plot) to observe spectrum over time. The use of either time or frequency domain independently provides the least amount of separability when observing FHSS signals.

does not create artifacts that can be present in the Wigner-Ville distribution [71]. However, the disadvantage of using this method is the sensitivity it has to the window size. The window size is the number of time samples considered when calculating a discrete FT. The discrete FT is defined in (4.1) with x as the discretely sampled signal, N as the number frequency bins used for calculation, and n is the current sample.

$$y[k] = \sum_{n=0}^{N-1} e^{-2\pi j \frac{kn}{N}} x[n] \tag{4.1}$$

A TF waterfall is generated by continuously calculating this DFT across consecutive time windows, until the end of the observation window is reached. As can be seen in (4.1), the number of time samples N used in calculation directly determines the same number of resultant number of frequency bins N. This means that a higher resolution in the frequency domain requires more time samples, resulting in reduced resolution in the time domain. A FT alone can be calculated across the entire length of observation time and is shown in the bottom image in Fig. 4.4. This calculation however provides no time resolution, which is vitally important when not only observing consecutive hopping behavior across the spectrum, but also calculating temporal properties of signals such as start time and end time. This time-frequency duality, commonly known as the uncertainty principle or Gabor limit [74], can play a major role in the detection of any or all received frequency hopping signals. Without proper time resolution a hopping signal with a short dwell time could be completely missed. Conversely, without proper frequency resolution a signal with a small bandwidth could be entirely missed. The lack of resolution in either domain can reduce estimation accuracy which leads to less distinguishable hopping signals. To balance this trade off, the minimum detectable hopping signal as shown in Table 2.1, by the minimum dwell time and bandwidth, results in a minimum hop size of 3x3 TF pixels. This means that the smallest detectable hop will have both 3 pixels in time domain, and 3 pixels in frequency

Table 4.2: Properties of the Discrete Fourier Transform and TF resolution

Property	**Time (ms)**	**Frequency (MHz)**
Observation Length	500	80
FFT samples	4166	2400
Resolution	0.12	0.0333

domain. The resultant DFT parameters that achieve this are shown in Table 4.2. With the properties given, an example time-frequency waterfall of real, FHSS signals is shown in Fig. 4.5.

4.4 Signal Detection

A TF waterfall, or spectrogram, provides a means of visualizing a two-dimensional spectral image representing spectral energy in some bandwidth over time. Within this spectrogram, given ample received power, a signal's bandwidth, duration, amplitude, and spectral shape provides a visual fingerprint that can be used to identify a known or potential signal of interest. Autonomously extracting signals within this window however requires a unique solution. Several works operating on a TF spectogram attempt to provide that solution. In [75] a TF spectrogram is calculated with a STFT and then treated as a digital image. Image enhancement techniques such as contrast stretching and binarization are used along with morphological transformations in an attempt to reveal hopping signals dispersed in background noise. The work in [76] similarly transforms received data to a TF spectrogram and performs computer vision techniques such as edge detection to define boundaries of received signals which are then compared to a bank of stored reference signals. The work in [77] presents an adaptive detection method after generating a TF spectrogram with a STFT. This method presents work in detection verification with an adaptive thresholding based on received signal size. In [78] a spectrogram is calculated and used along with a signal masking

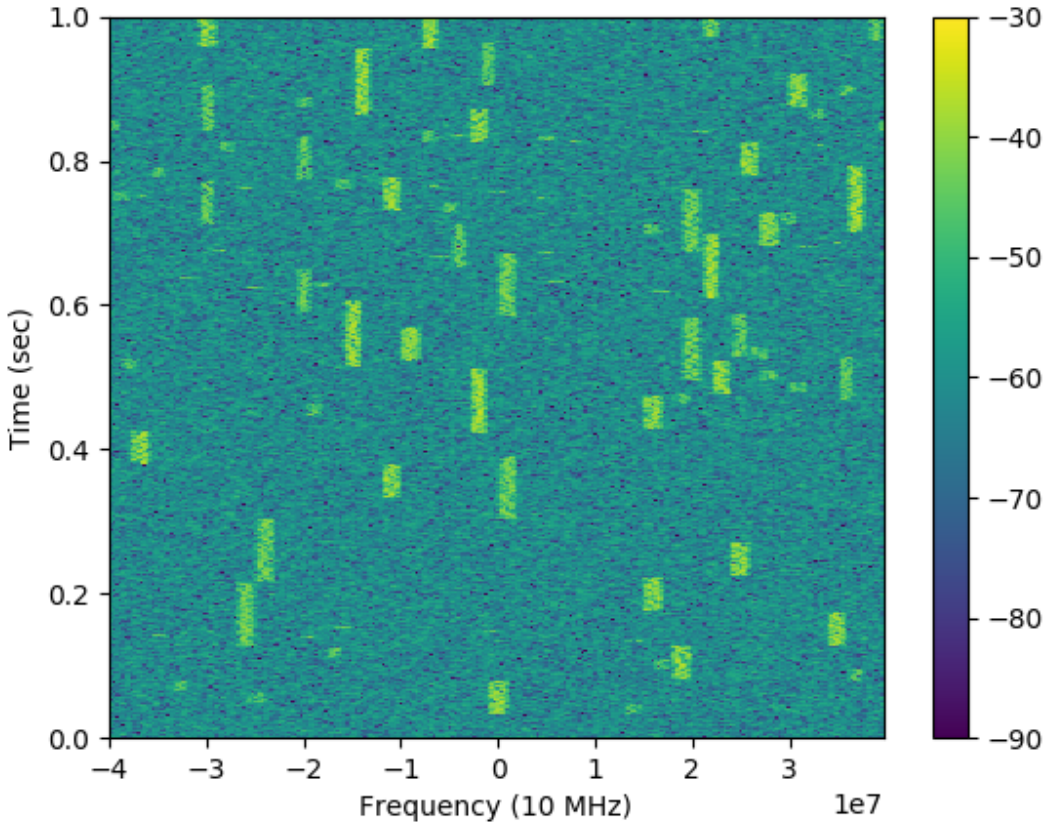

Figure 4.5: Example of a realistic hopping scenario where a band is shared by 6 hopping sources with variable powers, hop rate, and bandwidths.

method based on local statistical and morphological features of the signals of interest. The Expectation Maximization algorithm is used in [79] after signal hop timing estimates are extracted from a spectrogram.

While each of these works performs some type of image operations on a TF waterfall to enhance signal detection, they do not discuss implications in using these operations on signal detection and estimation. The prior work also do not consider the use of various image segmentation algorithms. Alternatively, this work will demonstrate how each of the image operations effect estimation accuracy at a range of SNRs. Several image segmentation methods will also be compared and discussed in terms of detection properties like true positives and false positives. In addition, unlike the previous chapter, the detection error for each received hop will not be based on a Gaussian distribution with a variance scaled by the received SNR. The error for each received hop in this chapter will instead be from actual signal detection from the raw data generated as described in the previous Subsection 4.2. The analysis in the previous chapter also demonstrated the sensitivity of constrained clustering to to timing estimation errors. Now, with actual detected signals, the performance of the constrained clustering algorithm will be shown with the timing error associated with the best detection method determined in the following sections.

The following section is structured as follows. Subsection 4.4.1 will introduce and discuss common image instance segmentation tools and the effects they may have on signal detection and estimation. Subsection 4.4.2 will compare the segmentation methods and demonstrate which method is best fit for this application. Finally, Subsection 4.4.3 will provide a deeper analysis of the best detector in terms of parameter estimation accuracy.

4.4.1 Segmentation Tools

Instance segmentation is a form of image segmentation where every single object in an image gets an independent label. In a signal separation domain, when operating on a TF waterfall, the goal of instance segmentation would be to separate and individually label every possible signal burst. This section will discuss common tools used to perform this, and the impact they may have on a signal detection. To visually demonstrate the function of each segmentation tool, a TF waterfall consisting of hopping signals with variable power, bandwidth, and dwell time will be used. The TF waterfall of these signals can be seen in Fig. 4.6, with the depiction of the changing parameters in the upper image. Common tools used in image segmentation algorithms fit into three categories, filtering, morphology, and measurements. The remainder of this section will discuss the tools belonging to each of these categories, and the impact that tool may have on detection and estimation of FHSS signals. The tools for each category will be discussed in no particular order.

Filtering

Filtering involves the two dimensional convolution of a specified filter with a TF waterfall. This convolution is a sliding operation in which every pixel is manipulated with the filtering function that operates on the neighbors of the original pixel. Every filtering operation in this work is performed simultaneously. In other words, each neighborhood considered for the filtering at a specific pixel only consists of pixels from the original, unmodified TF waterfall. While the mathematical operation of each filter is the same, what differentiates the effect of the filter is the filter design. The two types of filter designs used in this work are smoothing and edge enhancement filters.

Smoothing operations are typically applied initially to reduce background noise present in an

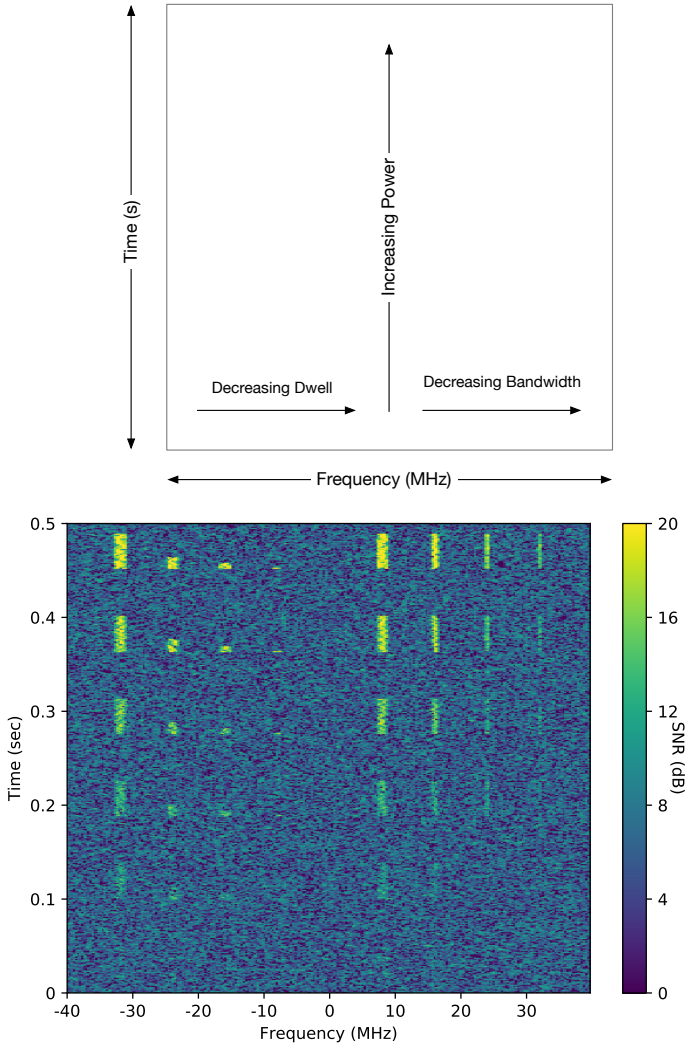

Figure 4.6: The hopping configuration used to test detection algorithms. The bandwidth, power, and dwell time are all swept according to the upper image. The resultant spectrogram of this sweep in shown in the lower image.

image. Common functions used in this smoothing operation are Gaussian, Mean, Median, and Bilateral. The type of smoothing function effects how the smoothing is applied and the size of the filter determines the local area of which the function is applied to. An example of this smoothing operation with a Gaussian filter can be seen in Fig. 4.7. The smoothing operation acts as a low pass filter, which reduces sharp peaks in the white noise in the background. This can be seen by the lighter shade of background in the image. However, the background is not the only portion of the image that is affected. As can be seen, the sharpness of the foreground signal edges is visibly reduced, and in the case of lower power signals, the smearing of spectral energy reduces the visibility of the signal altogether. This initial smoothing function may reduce large spikes in the random white background noise leading to a smaller amount of FP in the detection stage, however, the reduction in sharpness may reduce the accuracy in detecting the edges and therefore the dwell time and bandwidth of the signals. Also, with lower powered signals, the spectral energy could be reduced to the point of where it is indistinguishable from background noise, leading to a reduced number of TP.

The next type of operation used in this work is edge detection, or edge enhancement. Edge detection is used to identify sudden changes or discontinuities in a image. These discontinuities are best represented by maxima or minima of the gradient magnitude in an image and can be approximated with Sobel filter [80]. An example of edge detection with the Sobel filter can be seen in Fig. 4.8. As can be seen, borders are drawn around the majority of the spectral energy in the image. The Sobel filter can also be designed for specific orientations of edges to be detected. Since the signals in this problem have a constant center-frequency over the duration of the hop, they are rectangular in the TF waterfall and so only horizontal and vertical edges are desired. The filter used in Fig. 4.8 was designed to detect both of these directions simultaneously, although the filter can be designed to detect each

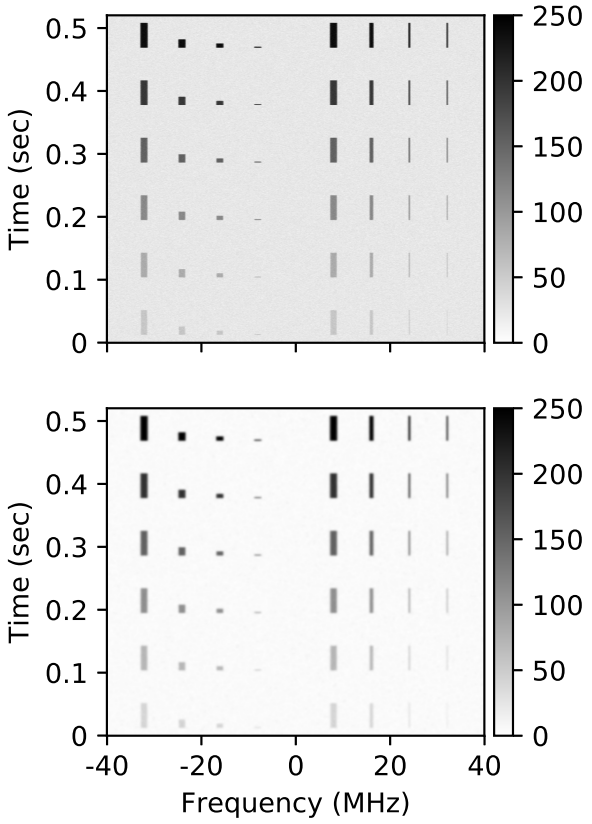

Figure 4.7: Demonstration of the effects of a Gaussian smoothing filter with the original gray-scaled spectrogram on top and the same spectrogram after smoothing on the bottom.

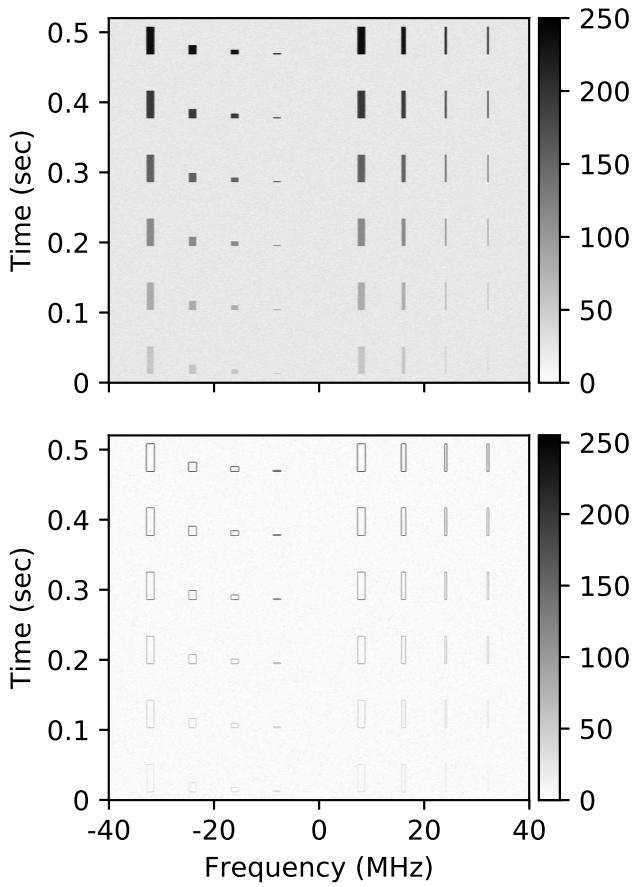

Figure 4.8: Demonstration of the effects of edge detecting Sobel filter with the original gray-scaled spectrogram on top and the same spectrogram after edge detection on the bottom.

separately. Edge detection operations may perform well with high powered signals, but as the SNR decreases, the magnitude of change of the image gradient becomes weaker and less distinguishable, leading to partially detected, or completely missed edges. In the case of a partially detected edge, a single signal may be split into several separate signals, which could lead to an increased number of FP and a reduction in accuracy in detecting the full signal.

Morphology

The next group of segmentation tools used in this work are morphological operations. These non-linear operations are related to the shape or structural features of an image and include erosion, dilation, opening, and closing. The concept and analysis for each of these operations are further explored in [81]. In the current work, these operations will be used in an attempt to remove noisy imperfections while at the same time enhancing the geometry of signals of interest.

In an erosion operation, every pixel in an image is reduced to the minimum value within that pixel's neighborhood matrix. Similar to the smoothing function, every value in the neighborhood matrix is drawn from the original TF waterfall. This effectively erodes collections of random noise that are smaller than the filter size. For a signal received with ample power, the main spectral shape of the signal of interest will be slightly eroded around the edges, and small high intensity regions of noise will be reduced. As can be seen in Fig. 4.9, each signal is reduced in size, and the signal with the smallest dwell time is completely removed from the image. If all considered hops were large, this operation would be the most beneficial, where large hops would remain mostly unaffected, and small regions of random noise would be removed from the image. The trade-off in using this operation though depends on the smallest signal under consideration. If the defined eroding filter size is greater than the smallest signal in terms of pixels, the small hop could be completely removed, which is shown by the absence of the signal with the smallest dwell time.

Dilation performs the opposite function, where again the specified number of neighbors are considered for every pixel, and each pixel is assigned the maximum value of all the neighboring values. This operation is often used to expand the foreground objects of an image. In the case of a TF waterfall, the size of small hops can be increased by expanding

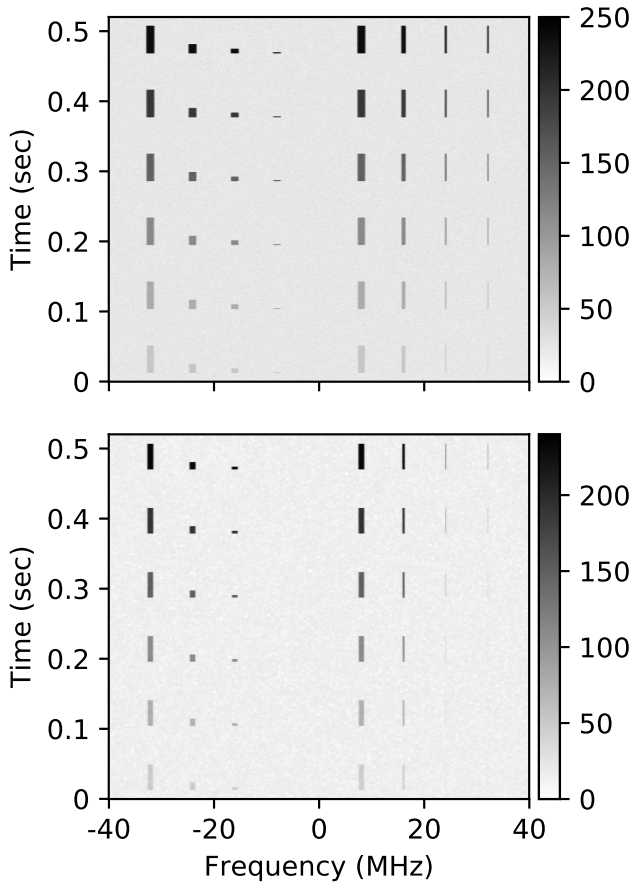

Figure 4.9: Demonstration of the effects of a morphological erosion with the original gray-scaled spectrogram on top and the same spectrogram after erosion on the bottom.

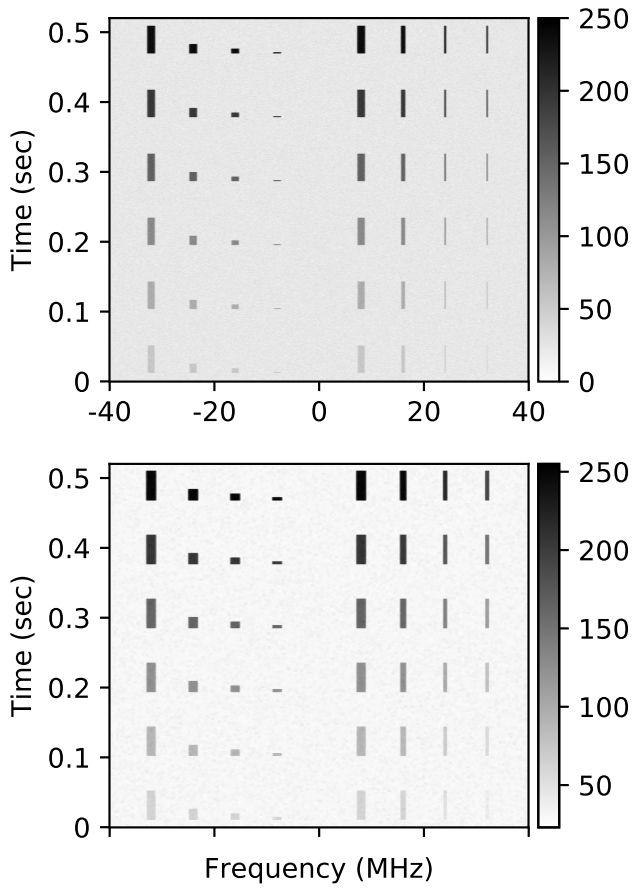

Figure 4.10: Demonstration of the effects of a morphological dilation with the original gray-scaled spectrogram on top and the same spectrogram after dilation on the bottom.

their main mass through a dilation operation, at the cost of increasing the size of previously small noise bursts. As can be seen in Fig. 4.10, the area of each hops grows, making smaller hops more identifiable, but deforming them slightly in the process, which could reduce the separability of sources with small dwell times and bandwidths. In Fig. 4.10, the increase in noise is not easily observable, but if thresholding is used, and noise surpasses the threshold, dilation can visibly increase the area of the noise, potentially creating a signal-like object in the image.

Typically, a square neighborhood matrix is used, however, a horizontal or vertical neighbor-

hood matrix can be specified for both erosion and dilation. When this shape is specified for dilation, as an example, the expansion of the image will only occur in that specified horizontal (frequency) or vertical (time) direction. In a TF spectrogram, the use of this operation and specification may be beneficial in connecting weak signal masses that may have been corrupted by background noise.

Two other commonly used operations are opening and closing. Opening is a two step operation that consists of erosion, followed by dilation. This is commonly used to reduce collections of random noise outside of a foreground object in an image, and then dilating the objects in an attempt to restore the edges that were lost with erosion. Closing is the exact opposite, it is a dilation followed by an erosion operation. This operation is usually used to remove "holes" in object in an image by dilating the objects, and then eroding away the excess edge created by the dilation. For this application, closing can be used to connect nearby regions of detected signal energy in a low receive power scenario. Similar to the erosion operation, opening can be used to remove small regions of noise without manipulating the foreground objects in an image.

Measurement

The measurement techniques used in this work are thresholding, and labeling. First, there is a binary thresholding, which labels every point in an image greater than the threshold 1, and less than the threshold 0. An example of this threshold operation can be seen in Fig. 4.11 In this application, a low threshold could increase the number of signals detected, but would result in more false detections of noise that is above this threshold level. Alternatively, a high threshold could result in missed low power signals of interest, but could remove a majority of the unwanted low powered noise.

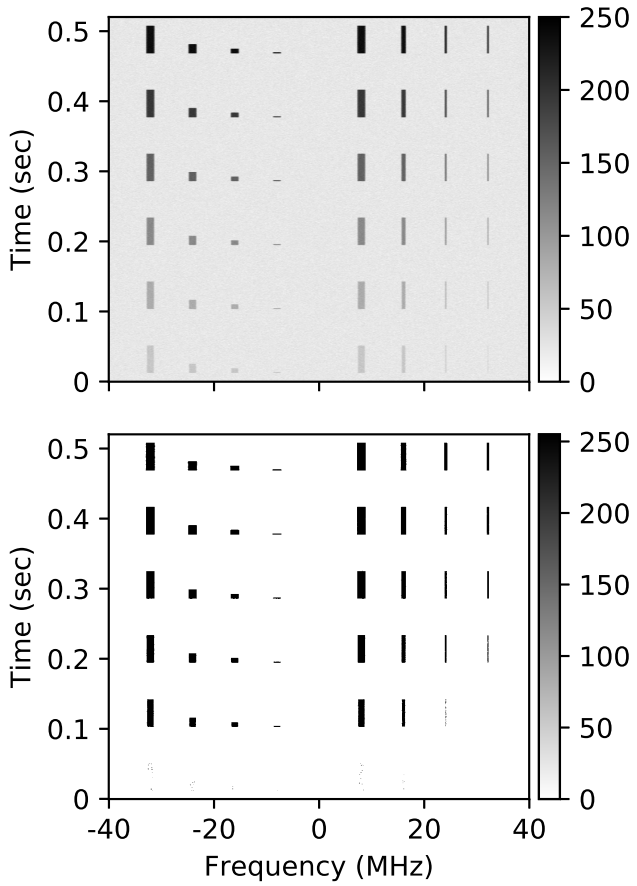

Figure 4.11: Demonstration of the effects of thresholding on a gray-scaled spectrogram with the original as the upper image and the thresholded result on the bottom.

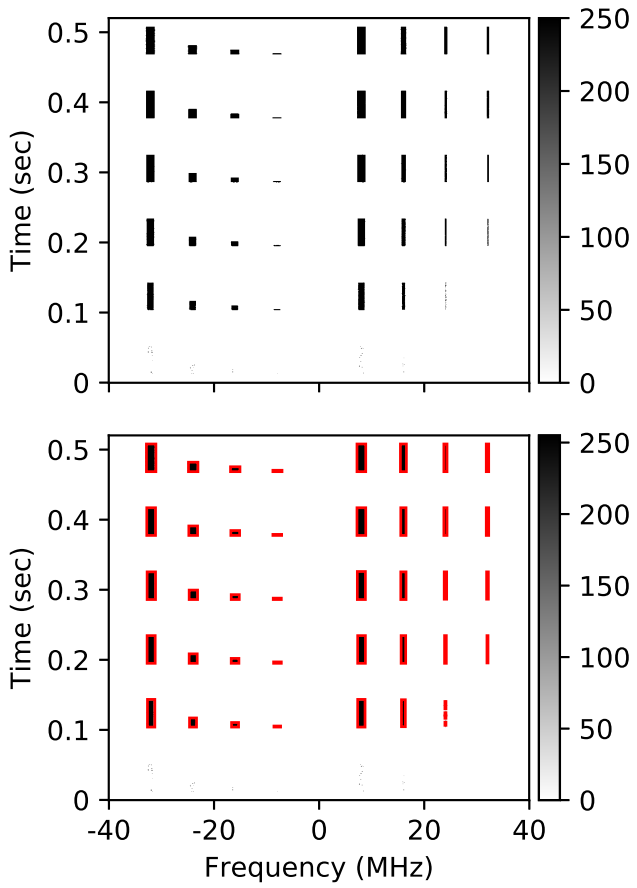

Figure 4.12: A visual representation of the boxing function used to set the boundaries on the estimated hopping signals. The previous threshold image is used as the base image, and the hops boxed with red boxes are shown in the bottom image.

The other measurement technique involves surrounding an object with an enclosing box. This enclosing box is derived the the minimum and maximum coordinates on all sides of a detected regions. If the region is larger than a specified threshold size, the object in the image will be surrounded by a box, as can be seen in Fig. 4.12. With the TF waterfall resolution discussed in the previous section as 3x3 pixels, a minimum possible signal size of 9 can be specified. This limit will reduce the number of noise samples being mistaken for signals. The resultant edges of the bounding boxes provide location estimations for each detected signal

parameters. From this box, the start time, end time, lower and higher bandwidth can be used to estimate dwell time, and bandwidth. Also, by observing the spectral energy within the bounding box, the power of the signal can be estimated. Considering the impact that accurate and consistent measurements of these parameters has on signal separation quality, it is important that each of these operations are used intelligently and cohesively to maximize detected hops, minimize false noise detections, and provide accurate and reliable parameter estimation that can be used with the clustering algorithms shown in the previous chapter. The next section will discuss the collective use of these operation in three different image segmentation techniques.

4.4.2 Algorithm Comparison

Although prior works have used image operations applied on a spectrogram to aid in signal detection, the impacts that each operation had on detection, as well as the result of using a combination of operations was not discussed. This work alternatively compares several segmentation methods and demonstrates the benefits and downfalls of each. Three algorithms proposed including an edge detection approach, a watershed approach, and a thresholding approach is used. The comparison of segmentation techniques provides insight as to which scenarios benefit each algorithm. For example, one algorithm may provide the best solution to detect small hops in low to moderate noise, but at the cost of an increased estimation error and a decreased overall hop resolution. Alternatively, another method may provide accurate signal bounding boxes but miss hopping signals with small dwell times or bandwidths. The use and comparison of the detection algorithms aims to resolve some of these uncertainties while providing a realistic detection solution that will be used to demonstrate the signal separation capability of the clustering algorithm developed in the previous chapter.

The first task in this signal detection problem is detecting the largest number of real signals, while minimizing the number of false detections of noise. To find the best parameters for each detector, a range of values for every step of each detection algorithm will be analyzed. A comparison of the three detection algorithms used, as well as the tools used in each algorithm will be discussed in the following section.

The parameter combinations in Tables 4.3, 4.4, and 4.5 were analyzed for each algorithm and the best combinations of parameters in terms of TP, FN, and FP were selected. These sets of parameters, shown in bold for each algorithm, provided the best detection performance on a Monte Carlo simulation of randomly generated hops bounded by the hopping properties in Table2.1. For the sake of clarity, the impact of each individual parameter in this sweep will not be discussed in this work. The purpose of this section is not to provide an optimal solution to this detection problem, but to discuss the differences in detection algorithms and provide a realistic estimation error that may be encountered in a real system. With that being said, the best parameters in bold are used for all further analysis for each algorithm. To provide a visual demonstration of algorithm stages and performance with these parameters, a FHSS signal sweep of SNR, bandwidth, and dwell time is used, and can be seen in Fig. 4.6. This section will provide a description of each algorithm along with a visual demonstration of the algorithm on the test hops in Fig. 4.6. The performance comparison for each algorithm in terms of signal estimation accuracy, will be shown in Section 4.4.3.

Edge Segmentation

The first algorithm considered, edge detection, functions by locating boundaries or discontinuities in an image. This boundary is typically calculated with the image gradient, where a large slope represents a sharp change in an image. This change typically undergoes some threshold value where the gradient must be sufficiently large to be considered an edge. In

Table 4.3: Edge Segmentation Parameters Examined

Parameter	**Value**	**Description**
fil_fn	Gaussian, **Mean**, Bilateral	Smoothing filter function
fil_size	(3,3), **(5,5)**	Smoothing filter size
fil_sigma	0.5, 1.5	Gaussian smoothing std. deviation
edge_th_min	1, **3**, 5	Minimum threshold value
edge_dil_iter	0, **2**, 4	Edge dilation iterations
edge_ero_iter	0, **2**, 4	Edge erosions iterations
min_size	**9**	Minimum object size

Algorithm 2 Edge Segmentation

procedure Edge(im) ▷ Edge segmentation of image, im
1: grayscale(im) ▷ Convert image to 0-255 grayscale
2: smooth($im, fil\_fn, fil\_size, fil\_\sigma$) ▷ Convolve im and smoothing filter
3: edge(im) ▷ Approximate image gradient with Sobel filter
4: threshold($im, edge\_th\_min$) ▷ Binary threshold image gradient
5: open(im) ▷ Morphological opening
6: dilate($im, edge\_dil\_iter$) ▷ Horizontal morphological dilation
7: dilate($im, edge\_dil\_iter$) ▷ Vertical morphological dilation
8: erode($im, edge\_ero\_iter$) ▷ Horizontal morphological erosion
9: erode($im, edge\_ero\_iter$) ▷ Vertical morphological erosion
10: open(im) ▷ Morphological opening
11: fill(im) ▷ Fill enclosed areas
12: label(im, min_size) ▷ Label objects larger than threshold
13: box(im) ▷ Draw bounding box around remaining objects
return im ▷ Segmented image

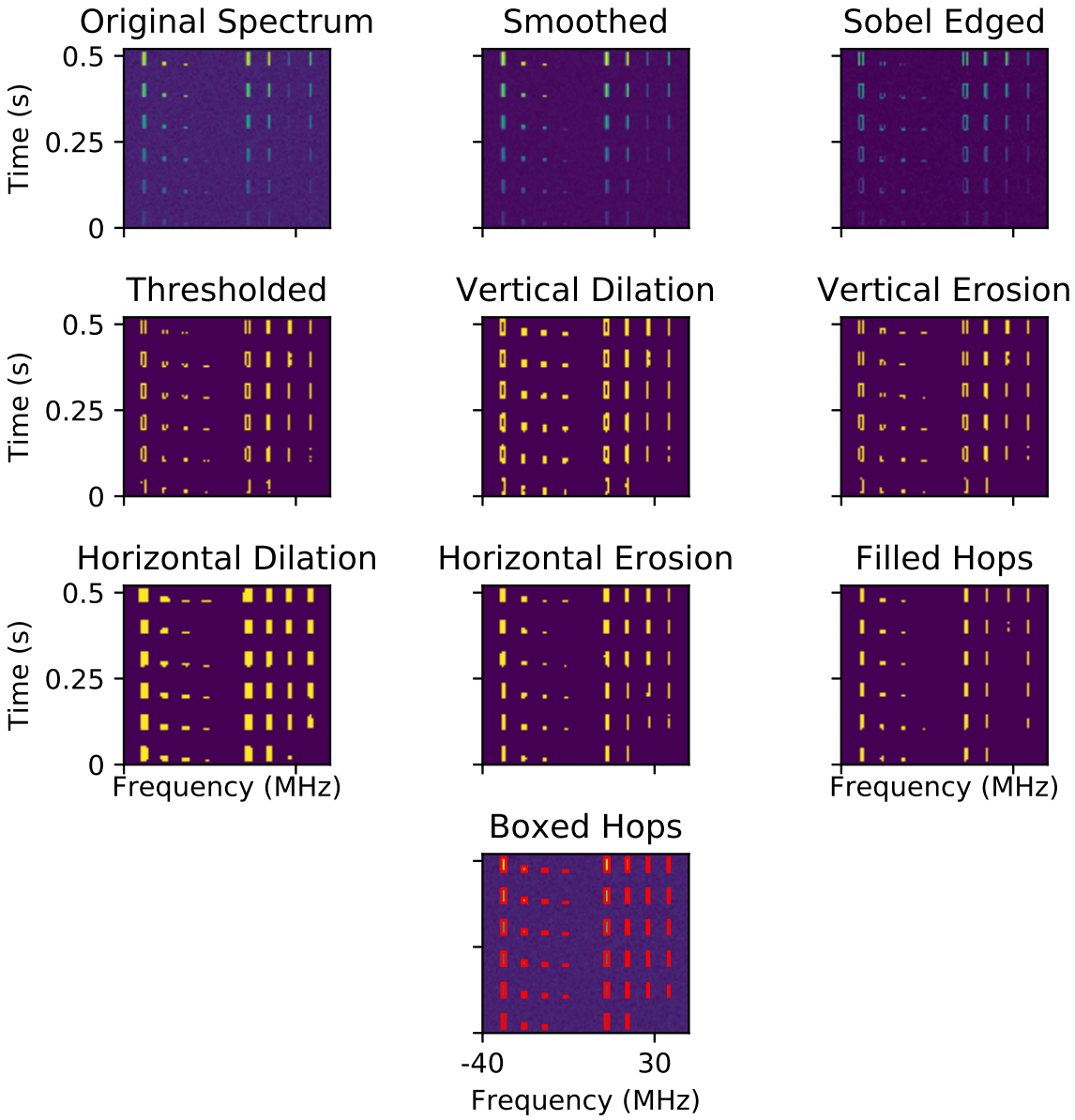

Figure 4.13: A Visual demonstration of the steps of the edge detection algorithm used in this work.

Table 4.4: Watershed Segmentation Parameters

Parameter	**Value**	**Description**
th_min	0, 0.5, **1**, 1.5, 2	Minimum threshold values
th_max	2.5, 3.125, **3.75**, 4375, 5	Maximum threshold values
min_size	**9**	Minimum object size

the case of FHSS signals, each signal, given ample power, will have defined boundaries that this edge detection algorithm will detect. With a reduced SNR however, the edges may be less defined, which can cause either false edges, or a reduced number of edges detected in general. Each step of this edge detection algorithm used is shown in Algorithm 2. A visual demonstration of every stage of this algorithm excluding the morphological opening, can be seen in Fig. 4.13. The opening stage is omitted due to its relatively low impact on the performance. Opening is best for reducing small peaks of noise, in this test case, the impact is difficult to visually observe. The order of image operations used in this edge detection algorithm follow a typical pattern for edge detection algorithms [82]. However, the vertical and horizontal dilation and erosion are specifically selected for this current work. After the edge detection operation listed in Algorithm 2, a threshold operation removes any edge magnitude less than the minimum threshold. Vertical and horizontal dilation can be used to connect any broken edges of a signal that may result from this thresholding. The size of the dilation filter will determine the distance between broken segments that will be attached after the operation. After dilation, an erosion is performed to reduce the expanded signal content closer to its initially detected size, which can reduce the estimation error that could be caused by using dilation alone.

Watershed Segmentation

Next, the watershed algorithm [83], treats the input image as a topographical map, and then marks the highest peaks of the image as well as the lowest valleys. If a signal is received with

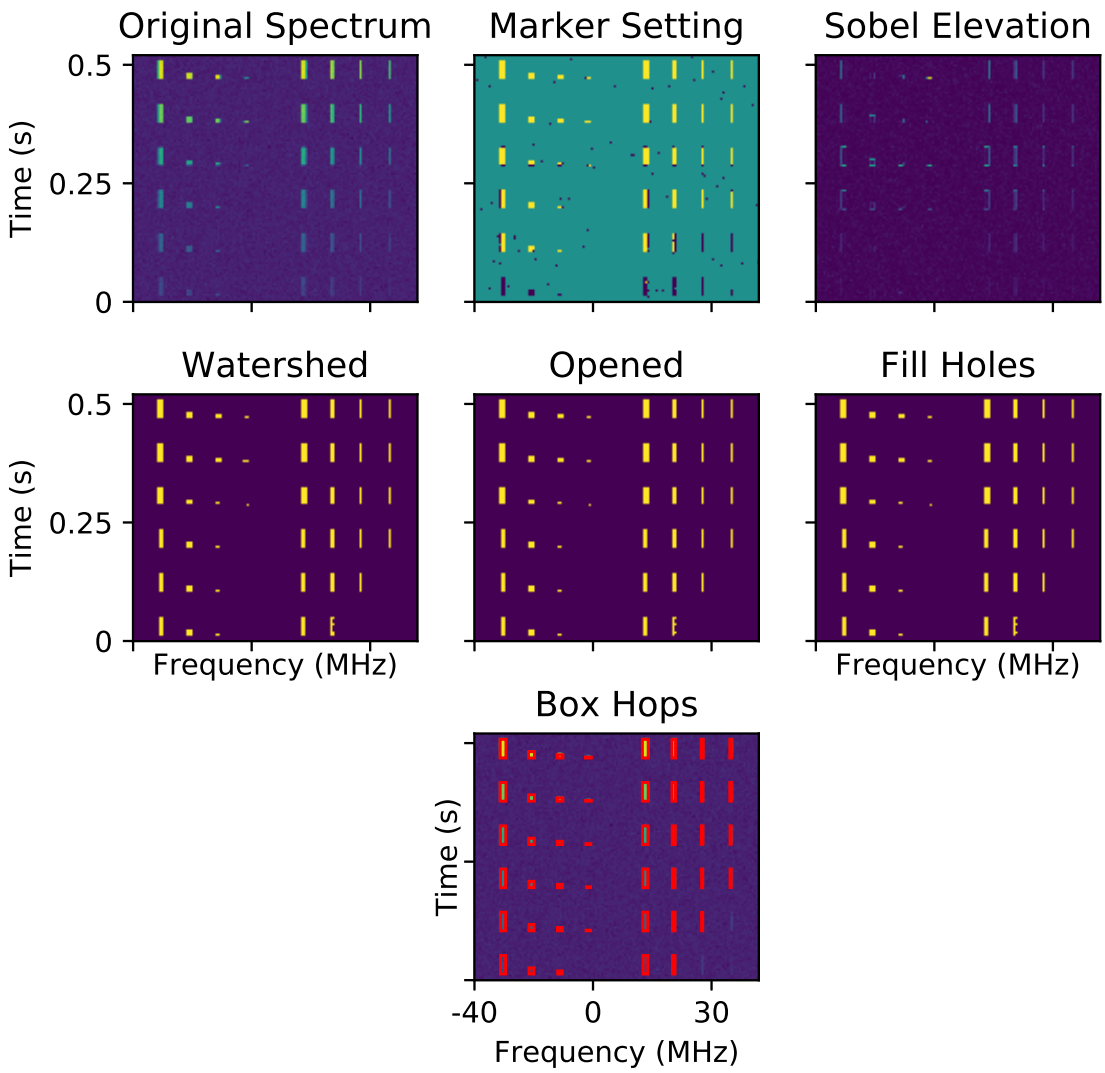

Figure 4.14: A visual demonstration of the steps of the watershed algorithm used in this work.

Algorithm 3 Watershed Segmentation

procedure Watershed(im) ▷ Watershed segmentation of image, im
1: grayscale(im) ▷ Convert image to 0-255 grayscale
2: mark(im, th_min, th_max) ▷ Mark img peaks and valleys
3: edge(im) ▷ Approximate gradient with Sobel filter
4: watershed(im) ▷ Watershed algorithm by Vincent and Soille
5: open(im) ▷ Morphological opening
6: fill(im) ▷ Fill enclosed areas
7: label(im, min_size) ▷ Label objects larger than threshold
8: box(im) ▷ Draw bounding box around remaining objects
return im ▷ Segmented image

ample power, the spectral energy will be marked as peak, and the surrounding noise will be marked as a valley. Next, boundaries are drawn around every peak, and are increased while at the same time, valleys are flooded to "wash out" lower values that are at or near the noise floor. This algorithm in theory should work well when signals in a TF waterfall are clearly visible, which occurs when signals are received at high SNR. However, if the valleys of the image, or the noise floor in this case, has an improper valley threshold, or is similar in power to the received signal, many noise samples could be labeled as peaks, which could cause an increased number of false detections. An example of Algorithm 3 is shown in Fig. 4.14 using the parameters in bold in Table 4.4. The order of image operations in this algorithms were performed based on the algorithms described in [83].

Threshold Segmentation

Finally, threshold segmentation algorithms are simple yet powerful algorithms used in image segmentation. The steps of the threshold algorithm used in this work can be seen in Algorithm 4. A visual representation of each step of this algorithm can be seen in Fig. 4.15. Threshold segmentation performs best when there is a large discrepancy between background and foreground objects. In this current work, a binary threshold level will be used in an at-

Algorithm 4 Threshold Segmentation

procedure Threshold(im) ▷ Threshold segmentation of image, im
1: grayscale(im) ▷ Convert image to 0-255 grayscale
2: smooth($im, fil\_fn, fil\_size, fil\_\sigma$) ▷ Convolve im and smoothing filter
3: threshold(im, th_min) ▷ Binary threshold image
4: dilate(im, dil_iter) ▷ Morphological dilation
5: erode(im, ero_iter) ▷ Morphological erosion
6: open(im) ▷ Morphological opening
7: label(im, min_size) ▷ Label objects larger than threshold
8: box(im) ▷ Draw bounding box around remaining objects
return im ▷ Segmented image

Table 4.5: Threshold Segmentation Parameters

Parameter	**Value**	**Description**
fil_fn	Gaussian, **Mean**, Bilateral	Smoothing filter function
fil_size	(3,3), **(6,6)**, (9,9)	Smoothing filter size
fil_sigma	0.5, 1.5, 3	Gaussian smoothing std. deviation
th_min	0, **1.25**, 2.5, 3.75, 5	Minimum threshold value
dil_iter	0, **1**, 2, 3	Dilation iterations
ero_iter	0, **1**, 2, 3	Erosion iterations
min_size	**9**	Minimum object size

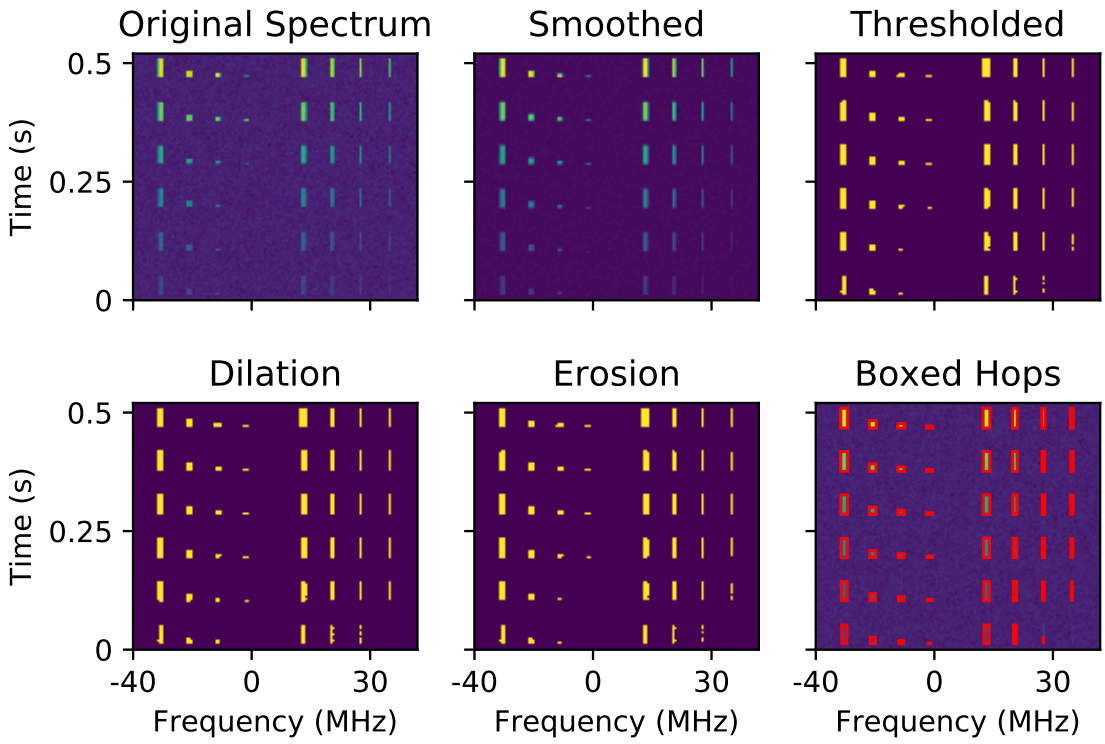

Figure 4.15: A visual demonstration of the steps of the threshold algorithm used in this work.

tempt to separate the signals in the foreground, from the noise in the background. With assumed knowledge of the noise floor, the minimum threshold value, th_{min} can be set in terms dBm above the noise floor.

Summary

Overall the algorithms presented in this section are functionally very different and cover a range of possible solutions that may be tried in a real detection system. A small parameter sweep for each algorithm was performed to provide similar detection capabilities in terms of TP, FP and FN. With the parameters for each detection algorithm chosen to provide similar detection performance on received signals, the next section will compare the estimation accuracy of each algorithm as a function of received signal power.

4.4.3 Detection Results

The previous section outlined three different image segmentation algorithms, and provided a small parameter sweep to enable fair comparison of detection performance of those algorithms. This section analyzes the estimation error of these algorithms by testing their performance on the hopping scenario listed in Table 4.6. The performance metrics used for this analysis are Error, Bias, IOU, and the number of TP, FN, and FP. Error in this section is simply the difference between the ground truth, and estimated hopping parameters. Bias can be defined as the average difference between the estimator and the ground truth values. IOU is a common method to measure the amount of overlap that an estimated area has with the actual area and can be defined as IOU = (Estimated Area $\cap$ Actual Area)/(Estimated Area $\cup$ Actual Area).

First, the number of TP, FP, FN along with the IOU of TP will be used to demonstrate the

Table 4.6: A table with the parameters swept to compare energy detection methods.

Property	**(Min, Max, Step)**
Power (dB)	(0, 20, 10)
Bandwidth (kHz)	(100, 1000, 10)
Dwell Time (ms)	(.3125, 100, 10)

general detection capabilities of each segmentation algorithm. After that, Error and Bias will be used to evaluate the estimation quality on the parameters using in signal separation, which are dwell time, bandwidth, and SNR. Since start and end time are also used to define constraints, the impact that estimation has on these constraints will be discussed. The distribution of the estimation error will also be compared with the previously modeled Gaussian error.

The first metrics that should be used to analyze the detection algorithms is the detection statistics. These include TP, FP, and IOU. In this work, a TP is regarded as the detection with the largest intersection of a ground truth signal and is considered a successful detection. Any other detection with a smaller intersection with the ground truth hop region will be considered a FP. Hops that have no intersection at all with a true hop will also be considered a FP. A FN is the result when the detector entirely misses a ground truth signal and is the complement of the number of TP. In this scenario, a FN may occur when the signal received has insufficient signal strength or SNR to be separated from the background noise.

As can be seen in Fig. 4.16, each detection algorithm begins to miss hops with the reduction in receive SNR. Without first properly detecting the presence of each hopping signal, the separation of signal sources cannot be reliably executed. With this in mind, the threshold segmentation technique misses the least amount of hops. Similar performance can be seen in the watershed algorithm. However, for edge detection, a significant number of FP occurs with a reduced SNR, most likely caused by the reduction of the strength in signal edges.

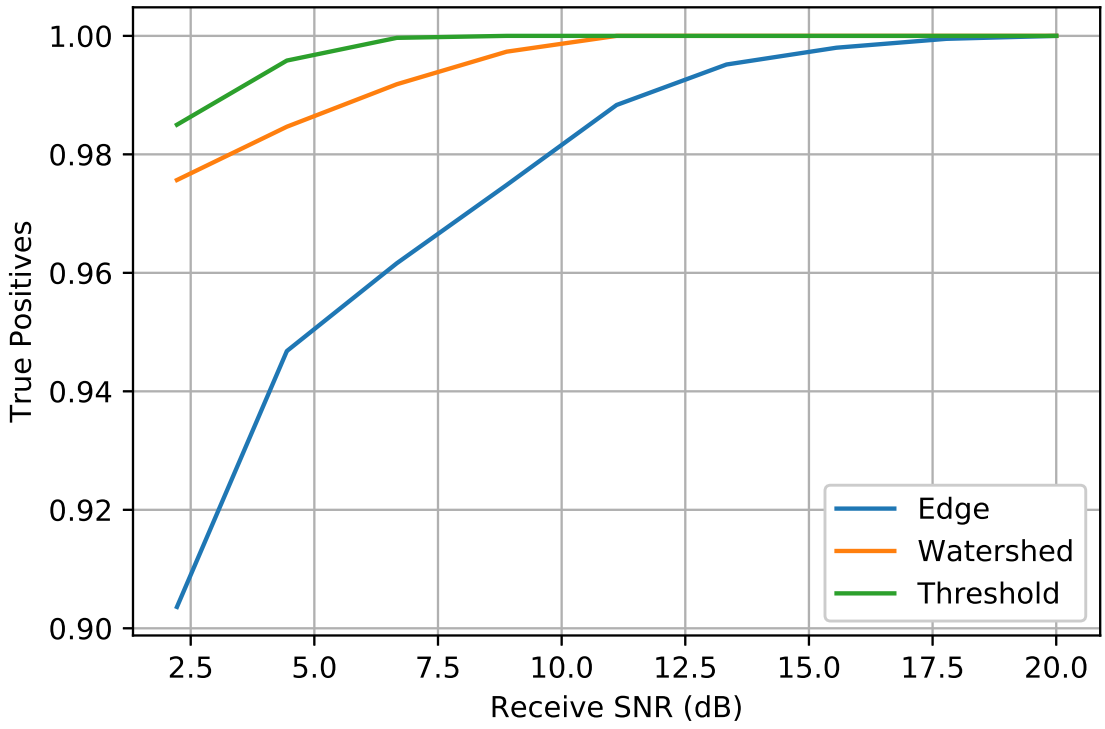

Figure 4.16: A comparison of the percentage of True Positives detected with a range of receive SNR for each detection algorithm.

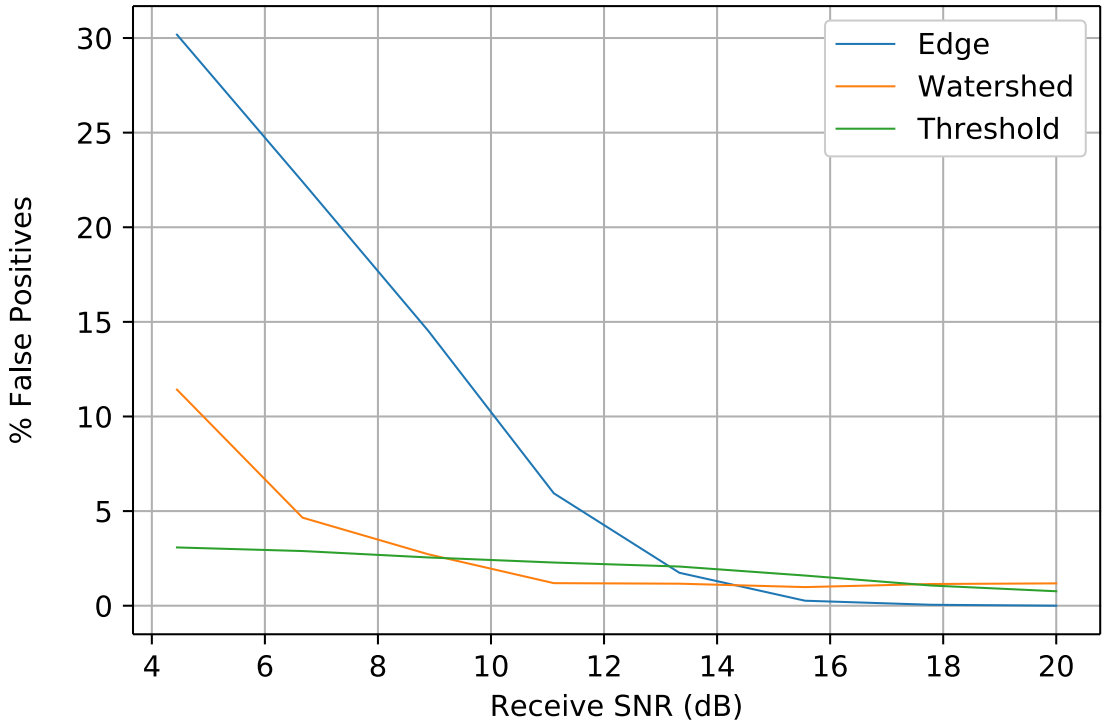

Figure 4.17: Number of false positives as a function of received SNR compared with the three detection algorithms used in this work.

Typically, a higher number of detected signals comes at the cost of an increased number of false detections, or FP. The percentage of FP for each algorithm can be seen with respect to receive signal SNR, in Fig. 4.17. As can be seen, at lower receive SNR, distinguishing between signals and noise become increasingly difficult. As for the threshold algorithm, the minimum threshold value is based strictly on the noise floor, where the threshold values for edge detection and watershed segmentation is determined with the gradient calculation of the spectrogram. As can be seen, the thresholding algorithm provides the smallest number of false positives, which for signal separation can not only confuse the clustering algorithms with additional outliers, but also can create a false sense of the number of users in the observed spectrum. The number of false positives observed both in the edge detection and

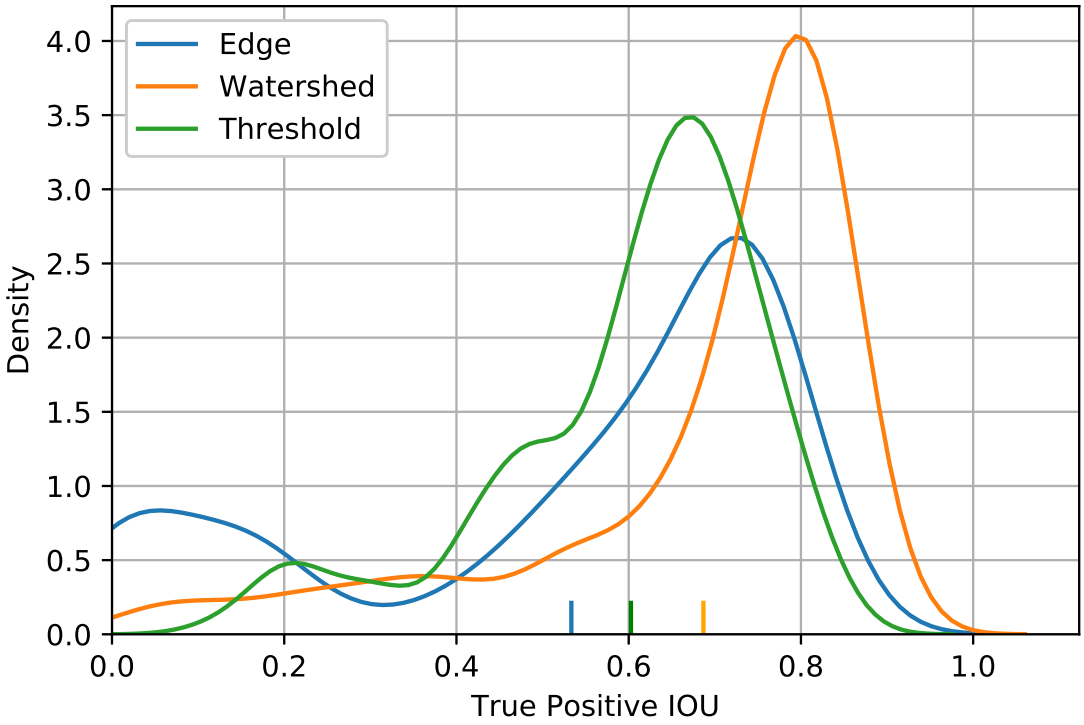

Figure 4.18: Density of IOU measurement between ground truth hops and detected true positives. A higher value represents move overlap in the estimated location area of the signal burst and the actual location of the signal burst. The mean of each algorithm's density is shown with the small lines on the X axis. A higher value for this IOU means the TP detections were better aligned with the location of the true signal location.

watershed segmentation algorithm may cause issues if the signals are not received at a high SNR.

Now that the raw detection metrics have been discussed, the quality of those detections will be covered with IOU, Error, and Bias. Considering only the detected signals labeled TP, the IOU is calculated based on the the area of overlap of the ground truth hop and the box constructed by the detection algorithm. IOU measurements demonstrate how well the estimated signal bounds represent the ground truth signal. A comparison of average IOU based on the signal sweep can be seen in Fig. 4.18.

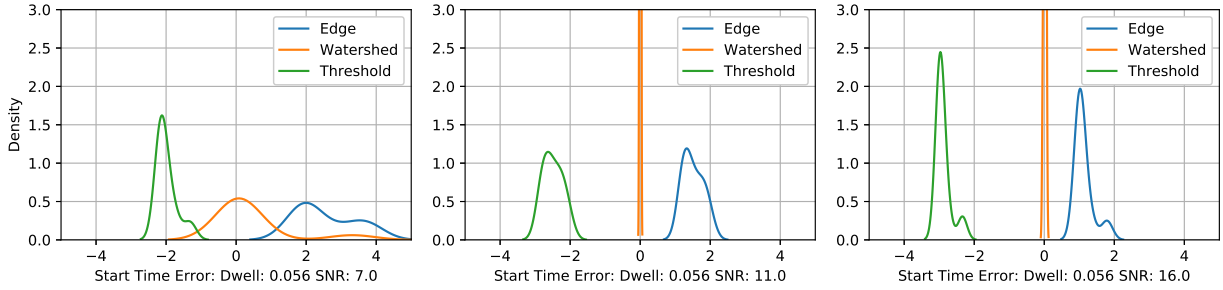

Figure 4.19: A comparison of start time estimation error for each detection algorithm with increasing SNR from left to right.

As can be seen, the watershed algorithm provides the most accurate detection box when compared to the ground truth signal, followed by the threshold segmentation and then edge detection. The densities for watershed and threshold segmentation are fairly similar, however edge detection shows a small increase in IOU values around zero. This demonstrates an inconsistency in the boxed region, which is most likely caused by broken edges and improperly sized bounding boxes, making edge detection undesirable.

Since signal separation in this work is dependent on three parameters used in clustering namely, dwell time, power, and bandwidth, and two that are used to define constraints, start and end time, accurate and consistent estimation of these parameters is necessary to reveal similarities present in sequential hops from the same source. In the previous chapter, it was shown that large errors in timing estimation significantly reduces the effectiveness of applying constraints, shown in Fig. 3.7. This timing estimation performance is shown for each detector in Fig. 4.19 and Fig. 4.20, with a varied amount of receive SNR.

As can be seen, the distribution of timing estimation error is largely affected by receive SNR. For consecutive hops, large variation in timing estimates may cause two consecutive signals from the same source to being cannot-link constrained. With enough of these erroneous constraints, the constraint based clustering effectiveness is reduced. Clustering performance will also be affected by the general dwell time estimation accuracy. Dwell time estimation

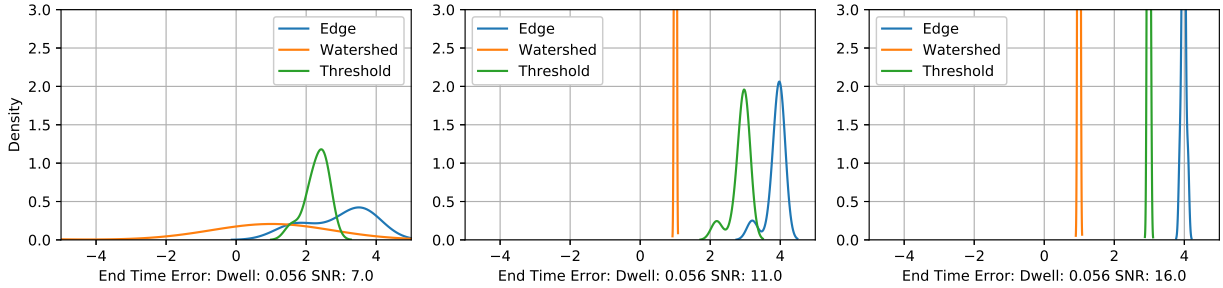

Figure 4.20: A comparison of end time estimation error for each detection algorithm with increasing SNR from left to right.

is derived directly from the start and end time estimates and will be shown in detail in the next section.

In terms of detection algorithms, each algorithm shows reduced error variance with increased SNR. However, over all SNRs shown in Fig. 4.19 and Fig. 4.20, the threshold algorithm is the most consistent in terms of error magnitude and variance of timing estimations, with the Edge and watershed method suffering largely at lower SNR. These consistently accurate timing estimations are essential in the functionality of constrained clustering, which shows again that the threshold technique is desirable. The mean of the error estimation demonstrates a common issue with estimators, which is bias. The bias describes the consistent offset that the error converges to given ample receive SNR. Across every SNR shown, each algorithm demonstrates a consistent timing error bias. Typically a detector bias is unwanted, however if the bias is consistent for every estimate, it can be stated as known and the mean bias value can be subtracted from every signal detection. An inconsistent bias is much less predictable, and can lead to a large detection error overall, reducing signal separation quality. Although the watershed algorithm provides practically non-biased estimations, the increased variance at lower received SNRs make this segmentation algorithm impractical.

The next set of parameters that are estimated are start band edge, and end band edge. The start band edge of a signal is the lowest frequency where spectral energy begins, and the end

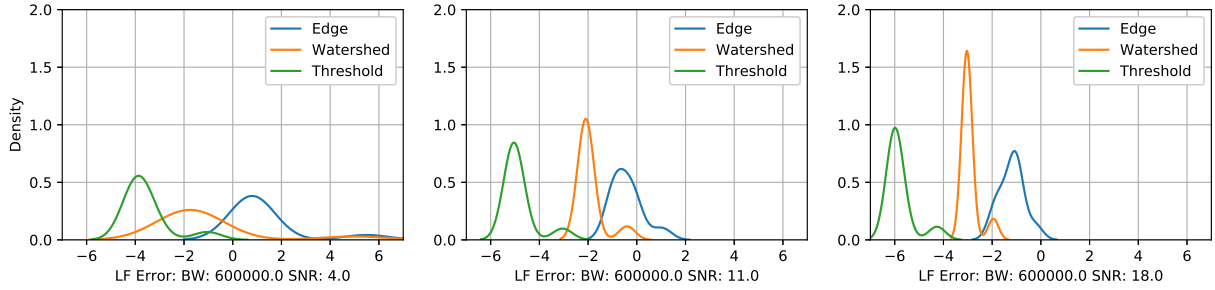

Figure 4.21: Error of start band estimates for every segmentation technique with increasing SNR from left to right.

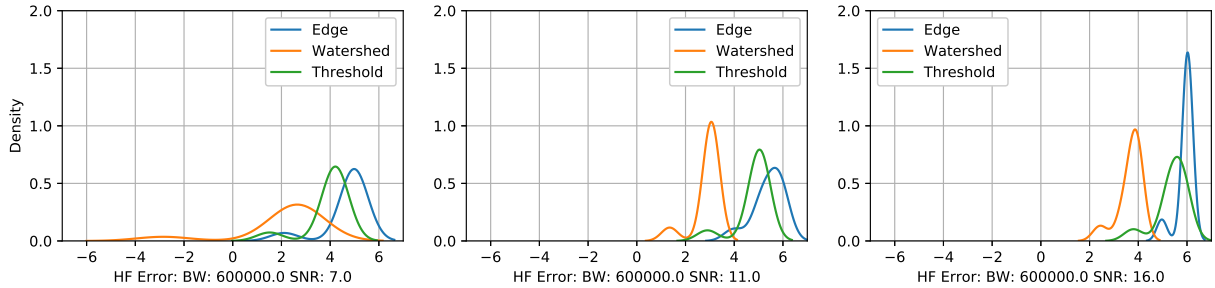

Figure 4.22: Error of end band estimates for every segmentation technique with increasing SNR from left to right.

band is where the spectral content ends at the higher frequency. Start band and end band are typically symmetric about a center frequency, and span a total width equal to the signal bandwidth. The bandwidth is directly estimated using both of these parameters and is used as another general parameter in the clustering algorithm.

The estimation of start and end bands can be seen in Fig 4.21 and Fig. 4.22. Similar to start and end time estimation, the variance in the estimation decreases as signal SNR increases. A noticeable difference from the timing estimation is the total variance of bandwidth estimates at higher SNRs. This variance is most likely due to the sharper transition of the start and end transitions of a signal, discussed previously as the transient, in comparison with the more gradual roll off of the bandwidth. Observing the bias of the start and stop band shows that this mean error is not constant, like it was in timing estimations. The cause of this SNR-

dependent frequency bias can be seen in Fig. 4.23. The power spectral density estimation shown in these figures demonstrates that with a constant noise floor, more spectral energy is visibly above the noise floor, which directly impacts the start and end band estimation for each of the segmentation algorithms. Overall, the important aspect of start and end band estimation is the variance of the estimations. Although the variance is larger than the previous timing estimations, it is still consistent enough to provide reliable estimates for each received hop.

Finally, the power estimation in this segmentation problem is based upon the labeled samples that are within the bounding boxes shown previously. Every value gets averaged within that bounding box that has been segmented and labeled from prior steps of the algorithms. The accuracy of this measurement is therefore dependent on IOU amount in which the estimate boxes overlap the true signal. The distribution of error from every hopping scenario is shown in Fig. 4.24. Similar to the IOU result shown previously, each algorithm performs fairly similarly. Since SNR is dependent on multiple factors such as the bounding box and spectral content, a specific probability density function is not likely, which is shown in Fig. 4.24. One characteristic of this estimate is the negative bias. The reason this negative bias occurs is because often times, as seen in previous error plots, the segmentation algorithm over estimates the size of the true signal. Since these signals are surrounded with white noise, this noise gets summed with the actual power measurements that are within the bounding boxes, lowering the estimated SNR. With this error, the power estimate can still show drastic distances between received hops, given that the SNR is different enough between sources. Out of all the previously estimated parameters, the power estimate showed the least consistency, but remained within a approximately 5dB window which ultimately aided in the signal separation process given sufficient difference in SNR.

In conclusion, based on detection quality alone using TP, FP, FN and IOU as the performance

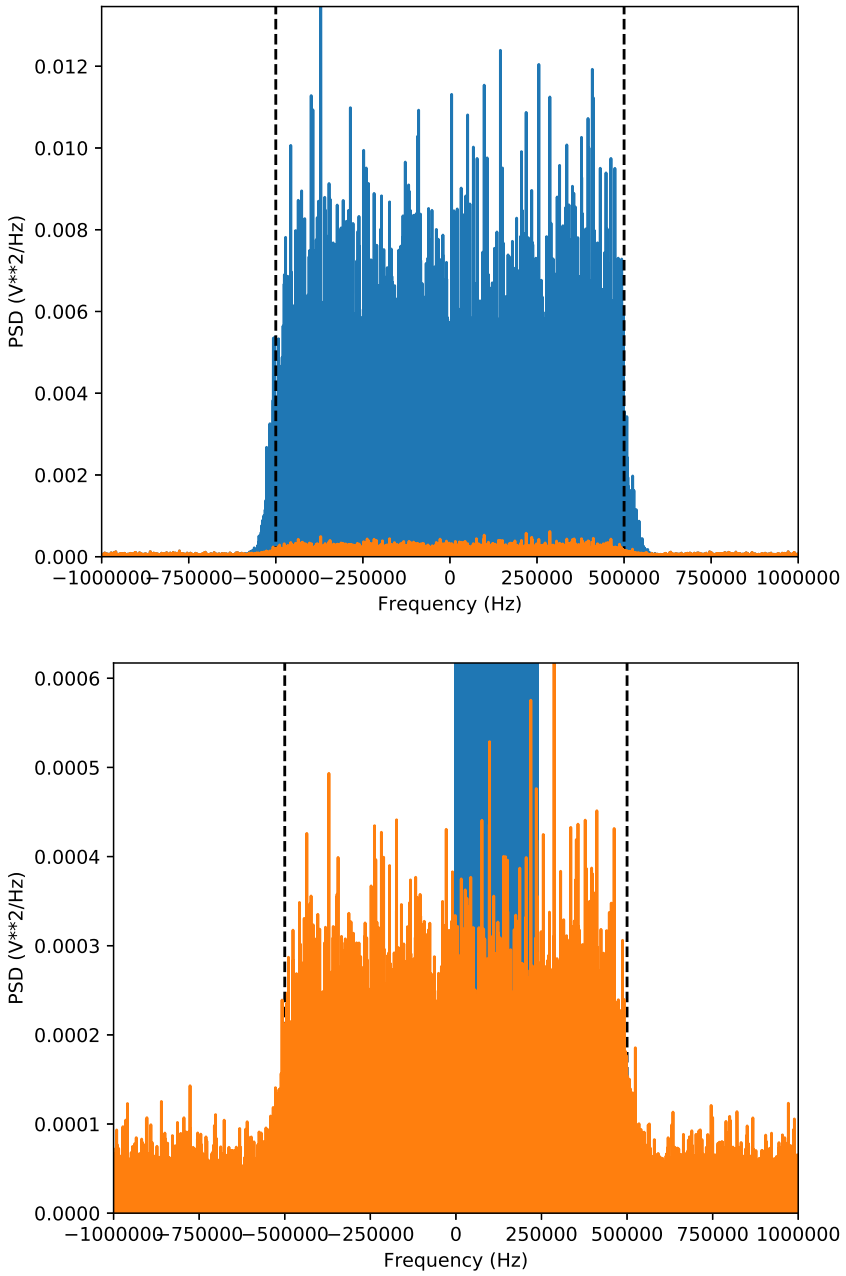

Figure 4.23: A comparison of the PSD calculation of hops that are 20dB (Blue) and 5dB(Orange) above the noise floor to show the increased bandwidth based on SNR. The true bandwidth in this figure is shown by the dashed black line. The reduced peak amplitude reduces the amount of spectral energy appearing outside of the 1MHz bandwidth.

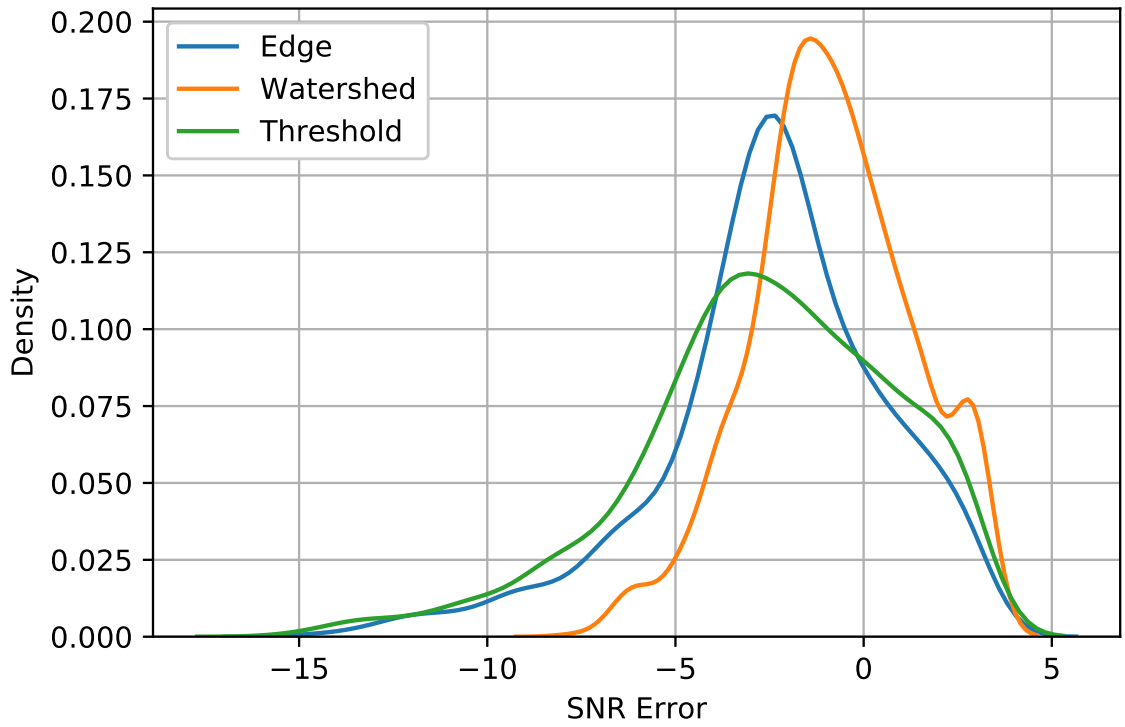

Figure 4.24: The estimation error density of received SNR. This SNR is calculated based on the enclosed spectral content within the bounding boxes found by each detection algorithm. The SNR for this Figure was varied from 0 to 20. A negative error value represents an estimated power that is less than the ground truth signal SNR. A Positive value respresnets an estimated SNR higher than the ground truth value.

metrics, threshold segmentation provided the most consistent detection characteristics. In terms of estimation quality of hopping parameters such as hop timing, frequency band, and bandwidth, each algorithm fluctuated similarly in terms of estimation consistency and accuracy. However, the threshold segmentation algorithm easily outperformed the other in terms of start and stop time estimation accuracy and consistency. Considering the impact that these metrics have on the clustering algorithm used for signal separation in this current work, threshold segmentation was found to be the best method for signal detection and estimation. For all future hopping analysis, threshold segmentation will be used with the best parameters found previously.

4.5 Clustering Performance on Synthetic Signals

Now with synthetic signals generated, detected, and estimated with the threshold segmentation algorithm, the performance of the signal separation approach used in this work can be evaluated on this synthetic data. Previously, the error associated with detection was modeled to be Gaussian with an increased variance that was based on lower receive SNR. Now, after providing an actual detection and estimation stage, it was revealed that SNR has the largest impact on the variance of error when estimating start and end times, start and end bands, and power. The impact that this realistic estimation error has on the performance of signal separation using constraint-based clustering is presented in this section. This analysis will be performed on hopping signals that are generated by the same bounds listed in Table 2.1, that are then detected by the threshold segmentation algorithm discussed in the previous section, and then clustered with the same constraint-based signal separation algorithm used in Chapter 3.

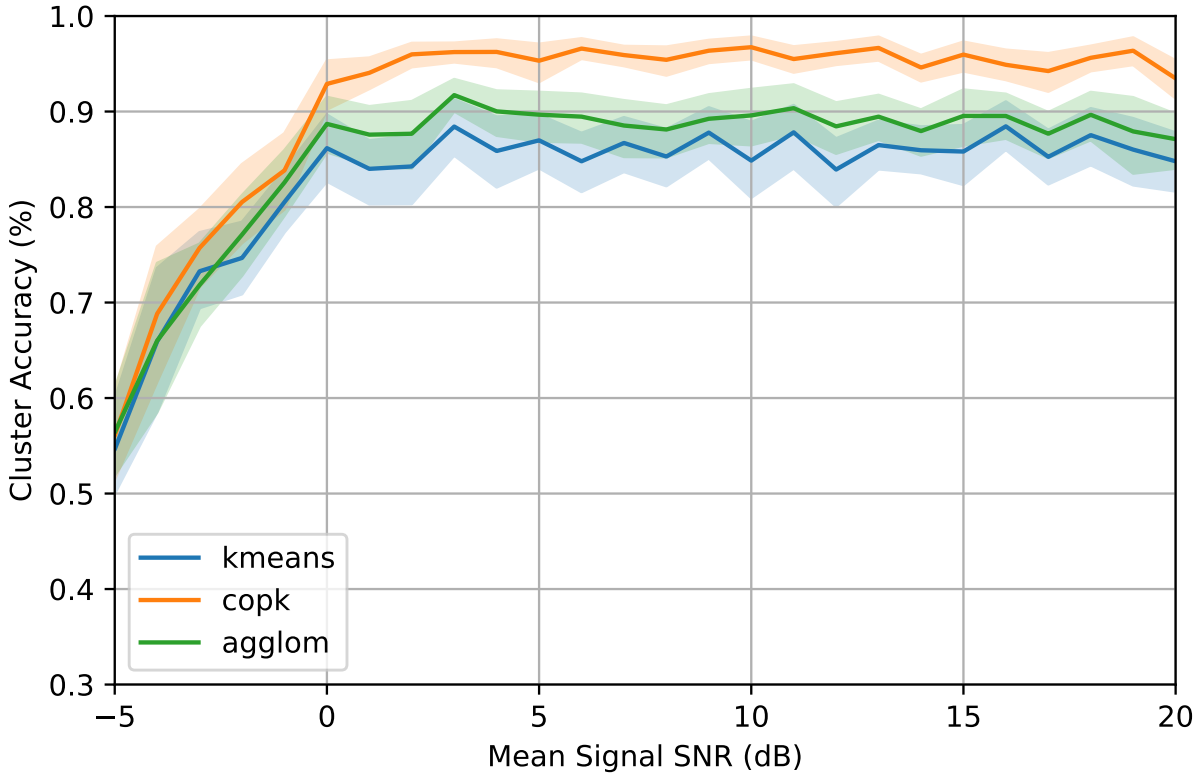

Figure 4.25: Clustering accuracy comparison of two classic clustering algorithms, K-means and Agglomerative, and the modified COP-Kmeans shown as a function of average received SNR of hopping signals.

4.5.1 Clustering Results

To demonstrate the performance difference in using constraints while clustering, K-means, constrained (COP-K) K-means, and agglomerative clustering will be shown. Accuracy and NMI are previously defined in Section 3.4.1, and are used as the performance metrics for the analysis in this section. The minimum SNR for any random hopping scenario is swept from -5 to 20dB SNR which demonstrates how the SNR-dependent estimation error can affect clustering performance. The resultant clustering Accuracy and NMI can be seen in Fig. 4.25 and Fig. 4.26 respectively.

As can be seen, each clustering algorithm suffers when the received signal SNR is below 0 dB.

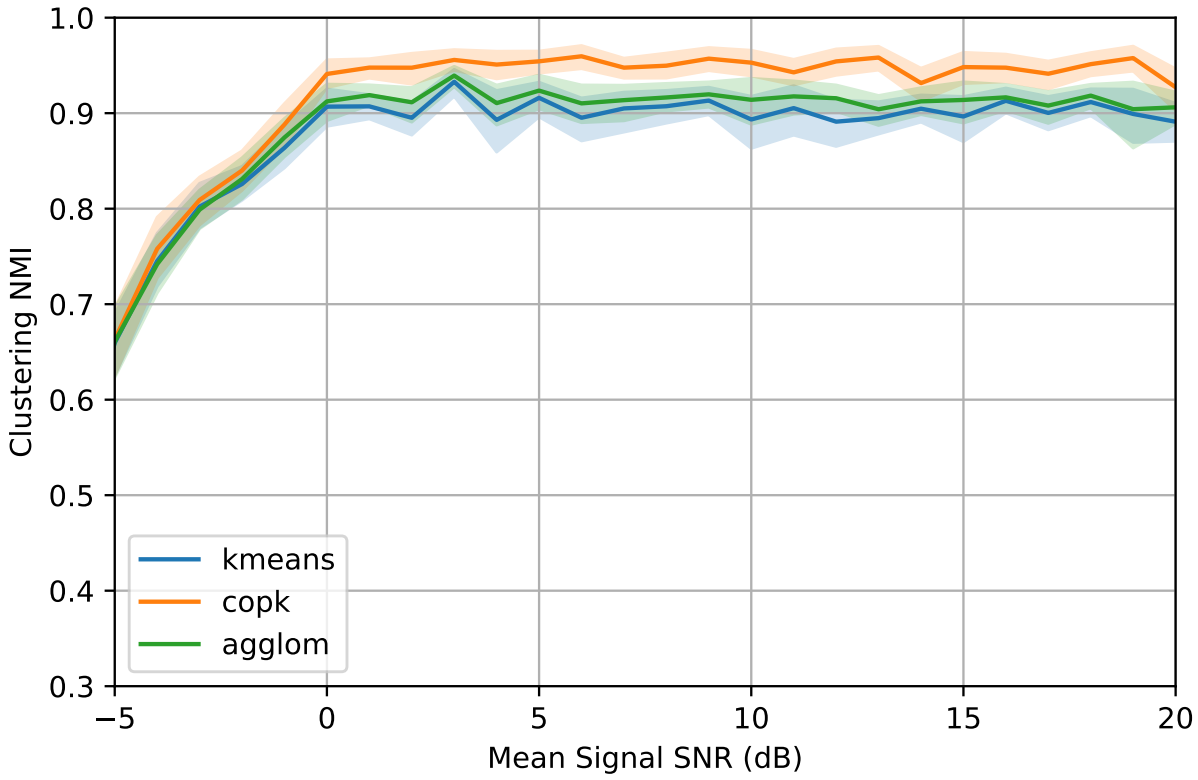

Figure 4.26: Clustering NMI comparison of two classic clustering algorithms, K-means and Agglomerative, and the modified COP-Kmeans shown as a function of average received SNR of hopping signals. NMI represents the similarity of the ground truth labels to the labels designated by the clustering algorithm.

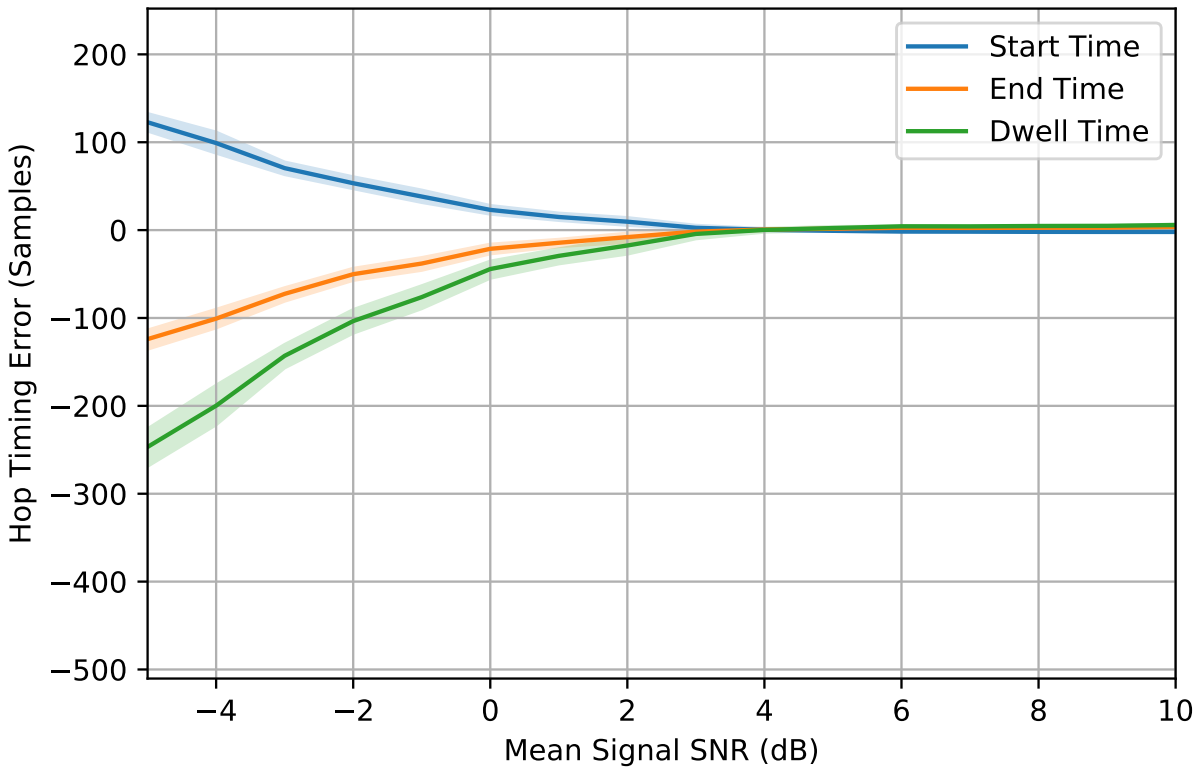

Figure 4.27: The estimation of start time, end time and dwell time as a function of mean SNR of detected hopping signals. A positive value indicates the estimated time occurs earlier than the actual time. A negative value indicates the estimation is later in time than the true value.

In this negative region, constraints are still beneficial over clustering strictly with distance, like K-means, however, as the SNR approaches -5 dB, the increased estimation error of the threshold segmentation algorithm reduces the accuracy in parameter estimation, which results in signals that are less distinguishable. At SNRs greater than 0dB, the separation accuracy does not increase, which can be attributed to the fairly constant estimation error present in threshold segmentation.

In comparison to the modeled Gaussian error Chapter 3, the estimation accuracy in terms of clustering performance can be compared

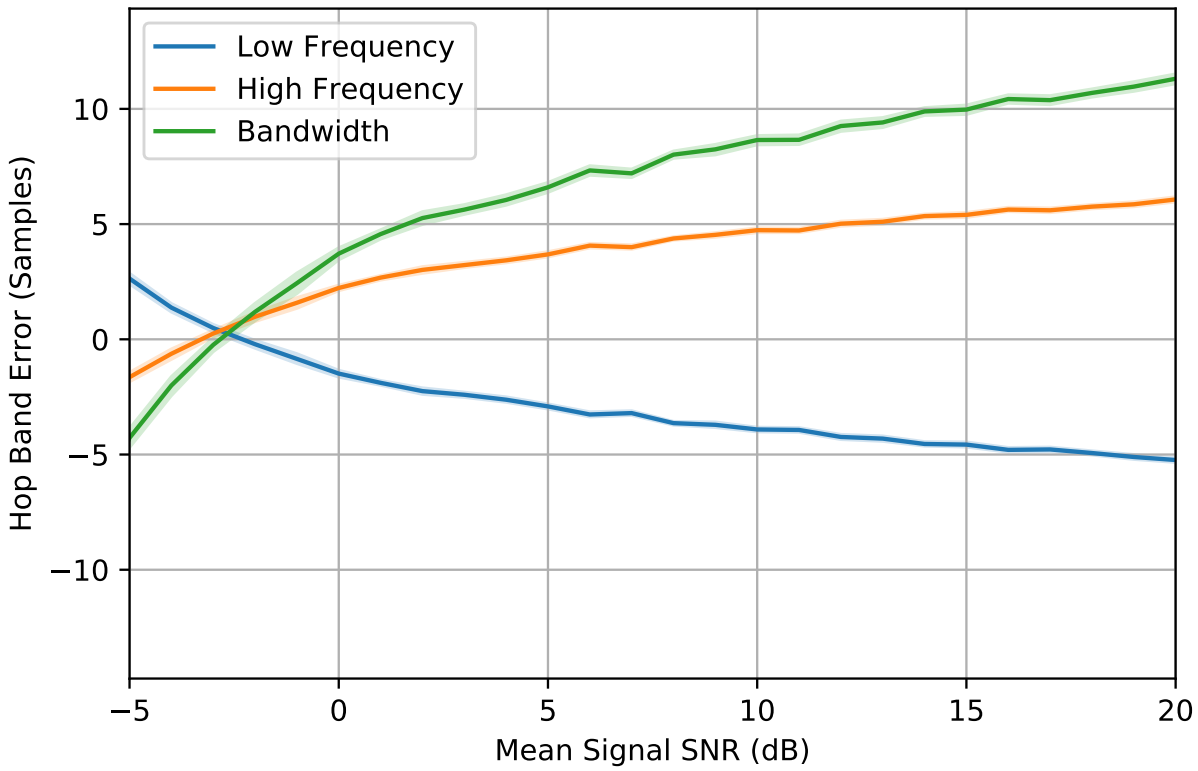

Figure 4.28: The error in estimating the lower and higher bandwidth edges of hopping signals. The total resultant error of the bandwidth estimation is also shown. A positive values indicates the band estimate is a higher frequency than the true value and a negative value indicates a lower frequency estimate against the true value.

The estimation for dwell time is shown in Fig. 4.27. Here negative error is associated with an over estimate in time, and a positive error represents a under estimate in time. The total error for dwell time is the sum of these two errors, represented by the line labeled "dwell time." As can be seen, the estimation is least consistent at lower SNRs, shown by the shaded region representing the standard deviation of the estimation error. In terms of clustering performance, this timing estimation variance at lower SNRs not only makes signal from the same source less distinguishable due to fluctuations in dwell time, but can also cause the assignment of erroneous constraints in the constrained clustering algorithm.

Bandwidth estimation error can be seen in Fig. 4.28. Unlike timing estimates, at lower

SNR, the estimated bandwidth is an underestimate when compared to the ground truth value, shown by the region under 0 for the line labeled "Bandwidth." At around -3dB SNR, the start and end band estimates become over estimates, representing a bandwidth larger than the ground truth value. As discussed in the previous section and shown in Fig. 4.23, this is most likely due to the increase visible spectral energy above the noise floor, which can be detected by the threshold segmentation algorithm. This constant increase is likely to follow the shape of the PSD of the signal in as can be seen in Fig. 4.23. A trend of this estimation, similar to timing estimation, is the overall reduction in estimation variance with an increase of SNR, shown by the reduction in the size of the shaded region for each estimation error. This variance plays the largest role in the clustering accuracy, since inconsistencies in the estimation of bandwidth causes consecutive hops from the same source to differ. Finally, the power estimate is shown in Fig. 4.29. With the dependency on the bandwidth and dwell time estimation, the power estimation does not behave as consistently across a range of receive SNR. The distribution of power estimates is mostly centered around 0 dB, but fluctuates to +-10dB. While this estimate has a larger variance than dwell time and bandwidth, if received signal sources are spread far enough in SNR, this power estimation can still provide information useful to separate hopping sources.

Overall, the clustering separation accuracy has proven to be heavily dependent on receive SNR, as can be seen in Fig. 4.25. Specifically at low SNRs, the increase in detection error causes significant changes in hopping parameters which reduces clustering performance. This detection error can be seen for each algorithm in Fig. 4.27 and Fig. 4.28. When the receive signal SNR was greater than 0dB, the estimation error was reduced to the point of providing little benefit in signal separation quality, shown by the constant accuracy for each algorithm in Fig.4.25. Across all SNRs, the use of constraints still proved to be beneficial in signal separation, with an increase of nearly 10 % Accuracy across higher received signal power

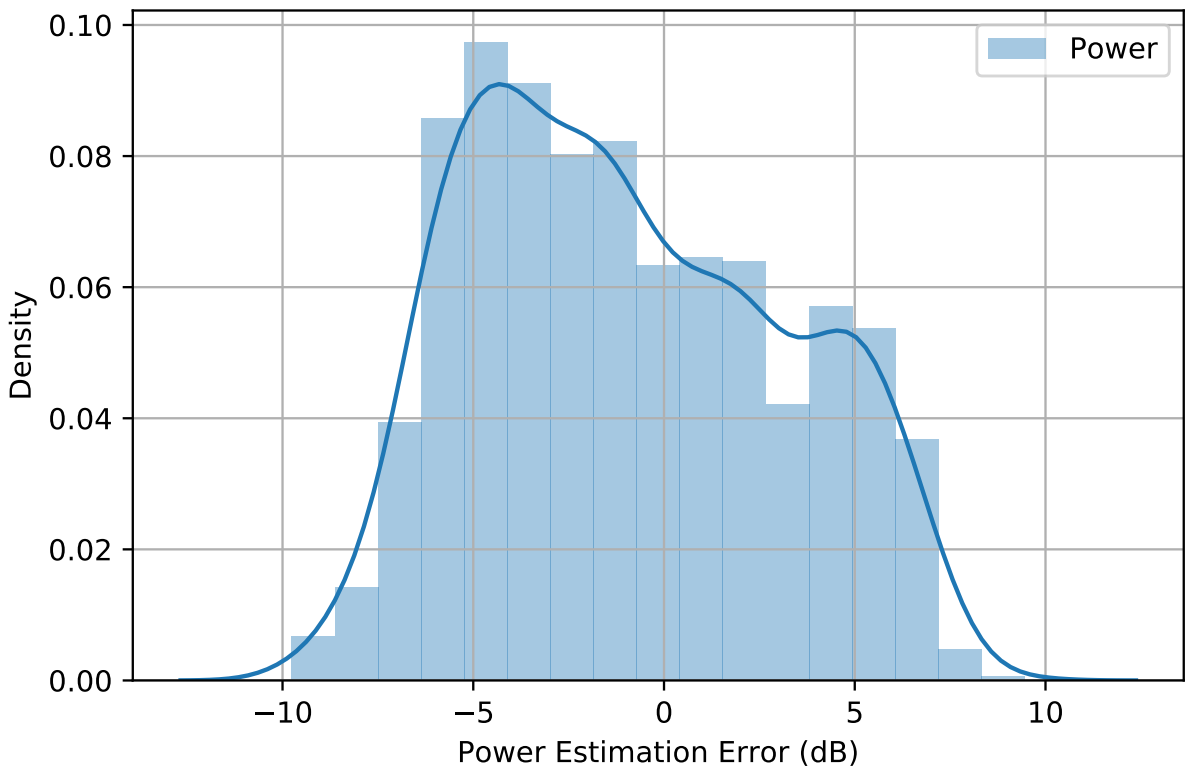

Figure 4.29: The estimation of the signals SNR based on the bounding boxes found by estimating the previously shown bandwidths, and dwell times. The estimation accuracy shown is the average error across a range of signal SNRs ranging from -5 to 20dB.

as can be seen in Fig. 4.25. At lower SNRs, the clustering performance for each algorithm began to converge, showing the limitation of the segmentation approach used in the current work.

4.6 Summary

This chapter has shown an end-to-end solution beginning with the generation of raw data to the conversion to the TF domain, followed by detection and estimation of signals and finally the grouping of these signals to their respective transmitters. In doing so, first, the potential impact of TF signal analysis was discussed in terms of impact it may have on signal detection. Then, the capabilities of three image segmentation techniques including edge, threshold, and a watershed approach were shown, providing a comparison that has yet to be covered in the prior literature. The implications of the segmentation techniques were also discussed and related to the signals estimation and separation solution presented in this chapter. Next, the modified constrained clustering algorithm presented in the previous chapter was validated on detected signal, and showed promising results when compared to classical clustering methods. More specifically, the constraints that are heavily dependent on start and stop time accuracy posed no threat to the performance of the clustering algorithm when a TF image segmentation algorithm is used for parameter estimation. This result demonstrates that the method used in this current work is at least robust to the estimation error present in detecting signals in an AWGN channel, which opens the possibility of applying this approach in a real system. Future work for this problem may involve attempting to further separate the performance of constraint based clustering in the lower SNR region shown in previous sections. Since the performance of each algorithm was heavily impacted by the estimation error variance below 0 dB SNR, an improvement in this region is obviously necessary. As

long as detection remains within reach of the segmentation algorithm, the use of additional constraints, or constraints that consider possible errors, such as soft constraints, may provide an advantage over other separation techniques at this difficult SNR region.

4.7 Acknowledgement

I would like to thank Joeseph D. Gaeddert for his time and assistance in generating the raw I/Q data and TF waterfall used in this chapter. Joseph provided a signal generation tool based on of his software defined radio [84]. This not only provided a real stream of I/Q data, but also allowed the same generation of signals in Chapter 4, with the hopping signal generator used in Chapter 3, Section 3.2. Joeseph also provided insight and assistance in generating the TF waterfall. The properties of the waterfall could easily be specified using the tool he provided. This waterfall creation allowed for the immediate application of FHSS detection and estimation algorithms, which were required to analyse the signal separation techniques presented in Chapter 3, on real estimated signals.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

The growth in the number of wireless communicating devices in recent years is pushing the need for intelligent and cooperative use of the finite available spectrum. External monitoring of user activity can help maintain this cooperative and collective use of the spectrum. However, this monitoring will usually have to be done without knowledge of the current and future activity of those users. FHSS is often employed by users in shared spectrum environment due to its narrow band interference resistance. This modulation technique provides a unique challenge in identifying and separating active users in a shared frequency band since traditional separation methods such as RF fingerprinting, and geo-location are not well adapted for this type of communication, as discussed in Subsections 2.1.1 and 2.1.2. Prior works that have been used for FHSS signal separation either relied on *a priori* knowledge of hopping characteristics, or do have not considered the difficulty in detecting and estimating signals received at decreasing SNR. The work in this thesis has addressed these issues in the subsequently discussed chapters.

Chapter 3 presented a novel separation solution using the detection-based characteristics including bandwidth, dwell time, and receive SNR, that removed the requirement of transmitter specific information that was required by many previous works. This solution demonstrated that using these parameters, within a reasonable amount of estimation error, provided

successful separation of FHSS sources mixed in time or frequency. The separation quality in this chapter was shown with classical clustering methods, but was shown to be improved with the novel application of constraint-based clustering. Since a single FHSS source only transmits one signal at any point in time, constraints could be used to restrict the grouping of temporally overlapping hops. The improvement in clustering accuracy provided a higher error tolerance, which in a real application, could reduce the cost of the receiver required to perform signal separation.

Chapter 4 expanded upon the work in chapter 3 by simulating raw data and demonstrating what realistic detection error may look like in a real world application. To demonstrate this error, first, raw signal data was generate for a mixed number of sources. This data was then converted into a TF waterfall, which enabled the detection using image segmentation techniques. Several image segmentation techniques were compared and the best result was used to demonstrate the performance advantage of using constrained clustering in signal separation.

In summary, the two chapters have contributed to the growing need of signal separation in the following ways:

- In Chapter 3, the FHSS signal parameters in Table 2.1 that can be estimated with only the detection of FHSS signals was discussed, and shown to be effective in separating signals from different sources by using several different clustering algorithms.
- The novel application of constrained clustering which used collisions in time domain to restrict the grouping of overlapping signals showed promise in improving signal separation accuracy.
- The detection of FHSS signals was shown in Chapter 4 by comparing several image segmentation techniques used to estimate hopping parameters. The implications of

both the TF generation and detection tools were discussed in terms of their impact on estimation accuracy and on the signal separation approach presented in this work.

- Finally, the use of the constrained clustering algorithm was proven to be beneficial with the simulated and detected FHSS signals presented in Chapter 4.

5.2 Future Work

There are several areas for future work following the analysis presented in this current work. First, the constrained clustering approach first presented in Chapter 3 could be expanded by reducing the impact that errors in the hard constraints have on the clustering performance by incorporating soft constraints. When accurate timing estimates are unavailable, the use of soft constraints could reduce the negative impact that erroneous hard constraints can have, without removing them completely. Another area to expand this chapter would be the inclusion and weighting of additional domain knowledge such as RF fingerprinting and localization information. These properties would not only provide additional constraints, but could also be weighted based on their estimation accuracy known by the receiver. This additional information could be used in combination with the temporal domain knowledge presented in Chapter 3 to improve clustering performance when estimates of detection-based hopping characteristics are not solely sufficient. Also, the detection technique presented in Chapter 4 could further be expanded by evaluating the impact of each parameter for every tool in the image segmentation detection algorithms. The parameter search for each detection algorithm in this work was fairly limited, in a case where optimal detection performance is needed, a larger parameter sweep and variations of the image operations presented in Chapter 4 could be explored. Finally, other methods of constrained clustering could be explored. Specifically, a constrained clustering method that removes the requirement of pre-assigning

the number of clusters would provide a more autonomous solution in comparison with the modified K-means approach presented in this thesis.

Bibliography

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys Tutorials*, vol. 17, pp. 2347–2376, Fourthquarter 2015.

[2] M. R. Palattella, M. Dohler, A. Grieco, G. Rizzo, J. Torsner, T. Engel, and L. Ladid, "Internet of things in the 5g era: Enablers, architecture, and business models," *IEEE Journal on Selected Areas in Communications*, vol. 34, pp. 510–527, March 2016.

[3] "Federal communications commission (2019). electronic code of federal regulations. pp.title 47 chapter 1 subchapter a part 2 subpart c 2.202.," 2019.

[4] Bluetooth Special Interest Group, *Bluetooth Core Specification 5.1.* 2019.

[5] L. J. Wong, W. C. Headley, S. Andrews, R. M. Gerdes, and A. J. Michaels, "Clustering learned cnn features from raw i/q data for emitter identification," in *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, pp. 26–33, Oct 2018.

[6] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, MobiCom '08, (New York, NY, USA), pp. 116–127, ACM, 2008.

[7] K. Talbot, P. Duley, and M. Hyatt, "Specific emitter identification and verification," 01 2003.

[8] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal,

"Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, pp. 54–69, July 2005.

[9] R. M. Vaghefi, M. R. Gholami, and E. G. Ström, "Rss-based sensor localization with unknown transmit power," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2480–2483, May 2011.

[10] S. Tomic, M. Beko, and R. Dinis, "Rss-based localization in wireless sensor networks using convex relaxation: Noncooperative and cooperative schemes," *IEEE Transactions on Vehicular Technology*, vol. 64, pp. 2037–2050, May 2015.

[11] A. Gök, S. Joshi, J. Villasenor, and D. Cabric, "Estimating the number of frequency hopping interferers using spectral sensing with time and frequency offset measurements," in *MILCOM 2009 - 2009 IEEE Military Communications Conference*, pp. 1–7, Oct 2009.

[12] M. Pospíšil, R. Marsalek, and J. Pomenkova, "Wireless device authentication through transmitter imperfections — measurement and classification," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 497–501, Sep. 2013.

[13] I. O. Kennedy, P. Scanlon, F. J. Mullany, M. M. Buddhikot, K. E. Nolan, and T. W. Rondeau, "Radio transmitter fingerprinting: A steady state frequency domain approach," in *2008 IEEE 68th Vehicular Technology Conference*, pp. 1–5, Sep. 2008.

[14] A. M. Y. H. H. S. Howard C. Choe, Clark E. Poole, "Novel identification of intercepted signals from unknown radio transmitters," vol. 2491, 1995.

[15] A. Kawalec and R. Owczarek, "Specific emitter identification using intrapulse data," in *First European Radar Conference, 2004. EURAD.*, pp. 249–252, Oct 2004.

[16] J. Hall, M. Barbeau, and E. Kranakis, "Radio frequency fingerprinting for intrusion detection in wireless networks," *IEEE Transactions on Defendable and Secure Computing*, vol. 12, pp. 1–35, 2005.

[17] K. Bonne Rasmussen and S. Capkun, "Implications of radio fingerprinting on the security of sensor networks," in *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops - SecureComm 2007*, pp. 331–340, Sep. 2007.

[18] M. Barbeau, "Detection of rogue devices in bluetooth networks using radio frequency fingerprinting," Citeseer.

[19] M. Barbeau, J. Hall, and E. Kranakis, "Detecting impersonation attacks in future wireless and mobile networks," in *International Workshop on Secure Mobile Ad-hoc Networks and Sensors*, pp. 80–95, Springer, 2005.

[20] O. Ureten and N. Serinken, "Wireless security through rf fingerprinting," *Canadian Journal of Electrical and Computer Engineering*, vol. 32, no. 1, pp. 27–33, 2007.

[21] K. Ellis and N. Serinken, "Characteristics of radio transmitter fingerprints," *Radio Science*, vol. 36, no. 4, pp. 585–597, 2001.

[22] R. M. Gerdes, T. E. Daniels, M. Mina, and S. Russell, "Device identification via analog signal fingerprinting: A matched filter approach.,"

[23] O. Ureten and N. Serinken, "Bayesian detection of wi-fi transmitter rf fingerprints," *Electronics Letters*, vol. 41, no. 6, pp. 373–374, 2005.

[24] J. Hall, M. Barbeau, and E. Kranakis, "Detection of transient in radio frequency fingerprinting using signal phase,"

[25] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, MobiCom '01, (New York, NY, USA), pp. 166–179, ACM, 2001.

[26] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, pp. 1067–1080, Nov 2007.

[27] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, pp. 54–69, July 2005.

[28] F. Izquierdo, M. Ciurana, F. Barcelo, J. Paradells, and E. Zola, "Performance evaluation of a toa-based trilateration method to locate terminals in wlan," in *2006 1st International Symposium on Wireless Pervasive Computing*, pp. 1–6, Jan 2006.

[29] C. Hoene and J. Willmann, "Four-way toa and software-based trilateration of ieee 802.11 devices," in *2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1–6, Sep. 2008.

[30] R. M. Vaghefi, M. R. Gholami, and E. G. Ström, "Rss-based sensor localization with unknown transmit power," in *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 2480–2483, IEEE, 2011.

[31] S. Tomic, M. Beko, and R. Dinis, "Rss-based localization in wireless sensor networks using convex relaxation: Noncooperative and cooperative schemes," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 5, pp. 2037–2050, 2014.

[32] G. Wang and K. Yang, "A new approach to sensor node localization using rss measurements in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 10, pp. 1389–1395, May 2011.

[33] G. Wang, H. Chen, Y. Li, and M. Jin, "On received-signal-strength based localization with unknown transmit power and path loss exponent," *IEEE Wireless Communications Letters*, vol. 1, no. 5, pp. 536–539, 2012.

[34] Z. Yang and Y. Liu, "Quality of trilateration: Confidence-based iterative localization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, pp. 631–640, May 2010.

[35] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," *IEEE Network*, vol. 18, pp. 45–50, July 2004.

[36] Y. S. Meng, Y. H. Lee, and B. C. Ng, "The effects of tropical weather on radio-wave propagation over foliage channel," *IEEE Transactions on Vehicular Technology*, vol. 58, pp. 4023–4030, Oct 2009.

[37] T. Kos, I. Markezic, and J. Pokrajcic, "Effects of multipath reception on gps positioning performance," in *Proceedings ELMAR-2010*, pp. 399–402, Sep. 2010.

[38] R. P and M. L. Sichitiu, "Angle of arrival localization for wireless sensor networks," in *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, vol. 1, pp. 374–382, Sep. 2006.

[39] P. Biswas, H. Aghajan, and Y. Ye, "Integration of angle of arrival information for multimodal sensor network localization using semidefinite programming," in *Proceedings of 39th Asilomar conference on signals, systems and computers*, 2005.

[40] C. Yang and H. Shao, "Wifi-based indoor positioning," *IEEE Communications Magazine*, vol. 53, pp. 150–157, March 2015.

[41] D. Niculescu and Badri Nath, "Ad hoc positioning system (aps) using aoa," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, vol. 3, pp. 1734–1743 vol.3, March 2003.

[42] S. Barbarossa and A. Scaglione, "Parameter estimation of spread spectrum frequency-hopping signals using time-frequency distributions," in *First IEEE Signal Processing Workshop on Signal Processing Advances in Wireless Communications*, pp. 213–216, April 1997.

[43] M. K. Simon, U. Cheng, L. Aydin, A. Polydoros, and B. K. Levitt, "Hop timing estimation for noncoherent frequency-hopped m-fsk intercept receivers," *IEEE Transactions on Communications*, vol. 43, pp. 1144–1154, Feb 1995.

[44] L. Aydin and A. Polydoros, "Hop-timing estimation for fh signals using a coarsely channelized receiver," *IEEE Transactions on Communications*, vol. 44, pp. 516–526, April 1996.

[45] D. Angelosante, G. B. Giannakis, and N. D. Sidiropoulos, "Estimating multiple frequency-hopping signal parameters via sparse linear regression," *IEEE Transactions on Signal Processing*, vol. 58, pp. 5044–5056, Oct 2010.

[46] L. Xiangqian, N. D. Sidiropoulos, and A. Swami, "Joint hop timing and frequency estimation for collision resolution in fh networks," *IEEE Transactions on Wireless Communications*, vol. 4, pp. 3063–3074, Nov 2005.

[47] X. Liu, J. Li, and X. Ma, "An em algorithm for blind hop timing estimation of multiple fh signals using an array system with bandwidth mismatch," *IEEE Transactions on Vehicular Technology*, vol. 56, pp. 2545–2554, Sep. 2007.

[48] D. Cohen, L. Pollak, and Y. C. Eldar, "Carrier frequency and bandwidth estimation of cyclostationary multiband signals," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3716–3720, March 2016.

[49] Y. T. Chan, J. W. Plews, and K. C. Ho, "Symbol rate estimation by the wavelet transform," in *Proceedings of 1997 IEEE International Symposium on Circuits and Systems. Circuits and Systems in the Information Age ISCAS '97*, vol. 1, pp. 177–180 vol.1, June 1997.

[50] X. Jun, W. Fu-ping, and W. Zan-ji, "The improvement of symbol rate estimation by the wavelet transform," in *Proceedings. 2005 International Conference on Communications, Circuits and Systems, 2005.*, vol. 1, pp. 100–103 Vol. 1, May 2005.

[51] T. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2nd ed., 2001.

[52] K. M. Neil Diener, "System and method for spectrum management of a shared frequency band," September 2007.

[53] S. Rayanchu, *Wireless and Systems for Understanding Wireless Interference*. PhD thesis, University of Wisconsin-Madison, 2012.

[54] X. L. N. Sidiropoulos and A. Swami, "Blind high-resolution localization and tracking of multiple frequency hopped signals," *IEEE Transactions on Signal Processing*, vol. 50, pp. 889–901, April 2002.

[55] V. S. Frost and B. Melamed, "Traffic modeling for telecommunications networks," *IEEE Communications Magazine*, vol. 32, pp. 70–81, March 1994.

[56] H. V. Poor and S. Verdú, "Probability of error in mmse multiuser detection," *IEEE transactions on Information theory*, vol. 43, no. 3, pp. 858–871, 1997.

[57] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, pp. 645–678, May 2005.

[58] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *Kdd*, vol. 96, pp. 226–231, 1996.

[59] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, pp. 395–416, Dec 2007.

[60] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[61] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 881–892, July 2002.

[62] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained k-means clustering with background knowledge," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, (San Francisco, CA, USA), pp. 577–584, Morgan Kaufmann Publishers Inc., 2001.

[63] D. Dua and C. Graff, "UCI machine learning repository," 2017.

[64] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, (Philadelphia, PA, USA), pp. 1027–1035, Society for Industrial and Applied Mathematics, 2007.

[65] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. New York, NY, USA: Wiley-Interscience, 2006.

[66] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, pp. 2837–2854, Dec. 2010.

[67] C. Manning, P. Raghavan, and H. Schütze, "Introduction to information retrieval," *Natural Language Engineering*, vol. 16, no. 1, pp. 100–103, 2010.

[68] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53 – 65, 1987.

[69] K. Wagstaff, *Intelligent Clustering with Instance-Level Constraints*. PhD thesis, Cornell University, 2002.

[70] J. G. Proakis and M. Salehi, *Digital communications*, vol. 4. McGraw-hill New York, 2001.

[71] B. Boashash, *Time-frequency signal analysis and processing: a comprehensive reference*. Academic Press, 2015.

[72] B. Boashash, *Time-frequency signal analysis and processing: a comprehensive reference*. Academic Press, 2015.

[73] L. Cohen, *Time-frequency analysis*, vol. 778. Prentice hall, 1995.

[74] D. Gabor, "Theory of communication. part 1: The analysis of information," *Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering*, vol. 93, pp. 429–441, November 1946.

[75] X. Mankun, P. Xijian, L. Tianyun, and X. Mantian, "A new time-frequency spectrogram analysis of fh signals by image enhancement and mathematical morphology," in *Fourth International Conference on Image and Graphics (ICIG 2007)*, pp. 610–615, Aug 2007.

[76] J. F. Turpin and K. A. Engholm, "Signal recognition and triggering using computer vision techniques," Oct. 20 2015. US Patent 9,164,131.

[77] S. Luo and L. Luo, "Adaptive detection of an unknown fh signal based on image features," in *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–4, Sep. 2009.

[78] J. Sildam, "Masking of time-frequency patterns in applications of passive underwater target detection," *EURASIP J. Adv. Signal Process*, vol. 2010, pp. 6:1–6:7, Jan. 2010.

[79] X. Liu, J. Li, and X. Ma, "An em algorithm for blind hop timing estimation of multiple fh signals using an array system with bandwidth mismatch," *IEEE Transactions on Vehicular Technology*, vol. 56, pp. 2545–2554, Sep. 2007.

[80] I. Sobel, "An isotropic 3x3 image gradient operator," *Presentation at Stanford A.I. Project 1968*, 02 2014.

[81] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, pp. 532–550, July 1987.

[82] K.-S. Fu and J. Mui, "A survey on image segmentation," *Pattern recognition*, vol. 13, no. 1, pp. 3–16, 1981.

[83] P. J. Soille and M. M. Ansoult, "Automated basin delineation from digital elevation models using mathematical morphology," *Signal Processing*, vol. 20, no. 2, pp. 171–182, 1990.

[84] J. D. Gaeddert, "liquid-dsp," 2019.