3-24-2016

Jamming Cognitive Radios

Travis J. Freeman

JAMMING COGNITIVE RADIOS

THESIS

Travis J. Freeman, Captain, USAF

AFIT-ENG-MS-16-M-015

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government.

AFIT-ENG-MS-16-M-015

JAMMING COGNITIVE RADIOS

THESIS

Presented to the Faculty
Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Electrical Engineering

Travis J. Freeman, B.S.E.E.
Captain, USAF

March 2016

AFIT-ENG-MS-16-M-015

JAMMING COGNITIVE RADIOS

THESIS

Travis J. Freeman, B.S.E.E.
Captain, USAF

Committee Membership:

Richard K. Martin, PhD
Thesis Advisor

Julie A. Jackson, PhD
Committee Member

Lt Col Phillip M. Corbell, PhD
Committee Member

AFIT-ENG-MS-16-M-015

Abstract

The goal of this thesis is to identify and evaluate weaknesses in the rendezvous process for Cognitive Radio Networks (CRNs) in the presence of a Cognitive Jammer (CJ). Jamming strategies are suggested and tested for effectiveness. Methods for safeguarding the Cognitive Radios (CRs) against a CJ are also explored. A simulation is constructed to set up a scenario of two CRs interacting with a CJ. Analysis of the simulation is conducted primarily at the waveform level. A hardware setup is constructed to analyze the system in the physical layer, verify the interactions from the simulation, and test in a low signal-to-interference and noise ratio (SINR) environment. The hardware used in this thesis is the Wireless Open-Access Research Platform.

Performance metrics from open literature and independent testing are compared against those captured from the jamming tests. The goal of testing is to evaluate and quantify the ability to delay the rendezvous process of a CRN. There was some success in delaying rendezvous, even in a high SINR environment. Jamming strategies include a jammer that repeats an observed channel-hopping pattern, a jammer with random inputs using the same algorithm of the CRs, a jammer that estimates channel-hopping parameters based on observations, and a random channel-hopping jammer. Results were compared against control scenarios, consisting of no jamming and a jammer that is always jamming on the same channel as one of the CRs. The repeater, random inputs to the CR algorithm, observation-based estimation jammer, and the random channel hopping jammer were mildly successful in delaying rendezvous at about 0%, 9%, 0%, and 1%, respectively. The jammer that is always on the same channel as a CR had an overall rendezvous delay about 13% of the time.

Table of Contents

Page

Abstract iv

List of Figures vii

List of Tables x

I. Introduction 1

1.1 Motivation 1
1.2 Background 2
1.3 Problem Statement 2
1.4 Assumptions and Resources 4
1.5 Thesis Organization 5

II. Background 6

2.1 Cognitive Radio 6
2.2 Spectrum Sensing 7
Matched Filtering 7
Energy Detector-Based Sensing 9
Other Spectrum Sensing Techniques 10
2.3 Binary Phase Shift Keying 11
2.4 Rendezvous 12
Jump-Stay Channel Hopping 17
2.5 Jamming 21
2.6 Prior Research 22
2.7 Conclusion 23

III. Methodology 24

3.1 Simulation 24
Initialize CRN Parameters 25
Rendezvous 28
Transmit and Receive Signal Generation 35
CRN Communication 36
Jamming the Rendezvous Process 39
3.2 Hardware 45
Constructing a communication and jammer network using WARP 45
Hardware Functions and Limitations 47
Initializing WARP Parameters 48
Transmit and Receive Signals Using WARP 49
Other Differences Between Simulation and WARP Code 56

Page

IV. Results and Analysis . . . 60

4.1 Simulation Results . . . 60
Designing the Jamming Waveform - Simulation . . . 60
Correlator Effects due to Jamming Waveform . . . 61
Observing Cognitive Radios and Jamming Strategies . . . 61
Effects of Jamming Strategies on Rendezvous . . . 66
Observations and Commentary . . . 71
4.2 Hardware Verification . . . 74
Designing the Jamming Waveform - WARP . . . 74
Correlator Effects due to Optimized Jammer using WARP . . . 75
Effects of Optimized Jamming Strategies on Rendezvous using WARP . . . 76
Observations and Commentary . . . 80

V. Conclusions . . . 82

5.1 Conclusions . . . 82
5.2 Future Work . . . 84
Bibliography . . . 86

List of Figures

Figure Page

1. Main sensing methods in terms of accuracies and complexities 9

2. Example BPSK and jamming waveforms in time domain 12

3. Comparison of maximum time to rendezvous for varied number of channels 15

4. Comparison of Expected Time To Rendezvous for Varied Number of Channels 17

5. Channel-Hopping Sequence: inner and outer rounds of Jump-Stay 20

6. Jump-Stay Example 21

7. Generic CRN and jammer logic flow diagram 25

8. Spectrum of the band-limited noise and transmitted signal 44

9. System configuration diagram for WARP node 1 47

10. Ethernet switch used to parse Matlab commands 48

11. Daughtercard radio board 49

12. Daughtercard architecture 50

13. WARP nodes 2 and 3 51

14. Antenna configuration 52

15. Time delay between transmit and receive signals on same channel using simulation 53

16. Time delay between transmit and receive signals on same channel using WARP 54

17. Time delay between transmit and receive signals on different channels using simulation 54

Figure Page

18. Time delay between transmit and receive signals on different channels using WARP . . . 55

19. Spectrogram of transmitted and received signals on the same channel using WARP: Transmitted Signal . . . 55

20. Spectrogram of transmitted and received signals on the same channel using WARP: Received Signal . . . 56

21. Spectrogram of transmitted and received signals on different channels using WARP: Received Signal . . . 57

22. Spectrogram of transmitted and received signals on different channels using WARP: Transmitted Signal . . . 57

23. Degraded preamble-based correlator output due to optimized jammer . . . 62

24. Degraded transmit/receive-based correlator output due to optimized jammer . . . 63

25. Unlimited spectrum observation parameter estimation effectiveness . . . 64

26. Half spectrum observation parameter estimation effectiveness . . . 66

27. Randomized Jump-Stay input effectiveness . . . 67

28. Repeater effectiveness . . . 68

29. Time steps required for rendezvous with preamble-based correlator . . . 70

30. Time steps required for rendezvous with transmit/receive-based correlator . . . 72

31. Channels states at rendezvous for full observation estimated Jump-Stay parameters for preamble-based correlator . . . 73

32. Degraded preamble-based correlator output due to optimized jammer using WARP . . . 76

33. Degraded transmit/receive-based correlator output due to optimized jammer using WARP . . . 77

Figure Page

34. Time steps required for rendezvous with preamble-based correlator using WARP . . . 78

35. Mean Time steps required for rendezvous with transmit/receive-based correlator using WARP . . . 79

36. Jammer always on same channel as Radio A for preamble-based correlator using WARP . . . 80

List of Tables

Table Page

1. Comparison of upper bound of maximum time to rendezvous for 2 users . . . 14

2. Comparison of upper bound of expected time to rendezvous for 2 users . . . 14

3. Overall summary of representative channel-hopping algorithms . . . 16

4. Jamming Waveform Parameters for Simulation . . . 61

5. Step size and index combinations to avoid . . . 65

6. Time steps exceeded by jamming strategy . . . 71

7. Jamming Waveform Parameters for WARP . . . 75

JAMMING COGNITIVE RADIOS

I. Introduction

This chapter outlines the basis of this research. The chapter includes the motivation for research, background information, goals for research, assumptions of the system being tested, and an outline of the thesis.

1.1 Motivation

Control of the radio frequency (RF) spectrum is a key objective for military operations to maintain air superiority. United States Air Force (USAF) doctrine on Electronic Warfare (EW) states “Military forces rely heavily on the electromagnetic spectrum (EMS) to sense, communicate, strike, and dominate offensively and defensively across all warfighting domains. EW is essential for protecting friendly operations and denying adversary operations within the EMS” [1]. Even in the US involvement in the Ukrainian conflict, “the Ukrainian military’s institutional knowledge as a former Soviet republic will help it understand how Russia fights, and its troops will have trained to operate while being jammed,” while the “(United States) lack(s) not only tactics, techniques and procedures but the training to fight in a communications-degraded environment” [2].

An emerging communication technology is the cognitive radio. CRs operate opportunistically wherever there is underutilized RF bandwidth and are able to drastically increase communication efficiency. CRs also have great potential in electronic warfare roles, since they are required to sense the RF spectrum and interact with other users based on their ability to sense the RF spectrum. Leveraging the potential of cognitive radios allows for more efficient use of the RF spectrum and furthers effectiveness of electronic warfare techniques.

1.2 Background

CRNs are a technology that enable more efficient use of spectrum among a multitude of users. Low-usage rates of allocated spectrum in wireless communications has prompted the methodology of CRNs to scan their environment and utilize unused channels so as to more effectively use spectrum as the opportunity arises. Such use of the spectrum on the part of the CRN must be accomplished without interfering with licensed users, also known as Primary Users (PUs) [3], [4].

Challenges with large-scale implementation of CRNs still exist and are the target of active research. Such challenges include spectrum sensing and management, signal processing, spectrum access control, autonomous network discovery and organization, routing, learning mechanisms, and adaptive control methods. With a space-time spectrum diverse environment, distributed structure of the spectrum for access is appropriate for coordination among users. Generally, interacting without impeding the other users in a given spectrum is a challenge in itself [3], [4].

1.3 Problem Statement

Research efforts have been made to develop neighbor-discovery algorithms for CRNs that are resilient to jammers [4]. A noise jammer's interaction with their environment has only been analyzed in simple scenario. The scenarios presented are that of a jammer that is tuned to a single center frequency and a jammer that changes its center frequency in a random manner after a given time interval [5]. While CRN countermeasure efforts have been made in other research, it is worth noting that the assumption in prior research is that the jammer itself is not typically cognitive or capable of spectrum sensing [5].

CRNs have to interact with various players in the spectrum of operation. A CR has to primarily avoid PUs. Since the PUs are licensed users, and the CRs are not, the CRN is expected to operate around the PUs. Less amiable users in the spectrum could be greedy users, that is CRs that are uncooperative with other users when it comes to sharing the spectrum. Uncooperative CRs can be treated much the same as an interference source, since their actions are not malicious, just inconsiderate. However, it is always possible that there are also jammers present in the spectrum [6].

A CJ is a specific type of CR. The CJ differentiates itself from standard jammers by being capable of spectrum sensing, thereby enabling it to seek out individual systems to disrupt rather than disrupting an entire spectrum. A CJ could be made out of any generic CR but would have embedded software designed to disrupt communications rather than enable them [7].

It is likely that any jammers present would be targeting PUs, since they are known to operate at a particular frequency. However, it is still possible that malicious users could target CRNs. Assuming a CRN is cooperative with spectrum sharing, a cognitive jammer can disrupt communications further by forcing the CRs to leave their current operating frequency. Without premonition from the CRN to implement some countermeasures, or develop a strategy to deal with jammers, the jammer could continue to degrade the communications of the affected CRs indefinitely. In applications where a device that fits a user's needs can be modeled with cognitive radios, such as a mobile radio communication device seeking to improve efficiency and throughput or to operate in electronically contested environments, it is imperative to understand the weaknesses of such a system so it can be exploited or defended effectively.

How cooperatively CRs will alter their spectral position to avoid PUs will largely determine how easily they are forced to move. If a CR primarily uses power measurements to decide to change channels, then the CR will be unable to tell the difference between a PU, a greedy user, or a jammer. While it may still be in the CR's best interest to move channels in the presence of any of these aforementioned interference sources, it is helpful to the CR to "learn" what category of interference is in particular channels such that it can adapt to its environment.

The goals of the research accomplished in this thesis are as follows:

- Investigate the consequences CRNs face due to CJs
- Determine effective CRN strategies to mitigate problems caused by CJs
- Explore design considerations that better protect CRs from CJs
- Determine effective jamming strategies to disrupt CR activity
- Refine spectrum-sensing strategies to make the CRN more resilient to jamming and interference

Simulations will be conducted to provide a baseline of deliverables and analysis for the CRN scenario. The same scenario will be implemented in the physical layer using WARP boards to provide proof of concept and validate the analysis from the simulations.

1.4 Assumptions and Resources

The network under test and analysis is a simplified one. These networks can be rather large, but this one is assumed to have just two communicating CRs and one CJ. The network is assumed to be symmetric — meaning the same channels are available to all users in the network. In this network, the CJ is assumed to be present, band limited, but not necessarily transmitting at all times in the spectrum. The CRs will behave as if the network is much more crowded than it is. Prior knowledge of the

CRs within the network will be explicitly stated as it becomes relevant. The network is assumed to be cooperative; the cognitive radios will only tune to channels that allow licensed users to communicate unimpeded. The CRs will also tune to different channels to alter their own explicitly-defined performance metrics — specifically when the radio is jammed sufficiently and performance is unacceptable. Such performance metrics will be defined and suggested when relevant. When radios have converged on a new channel or are in the midst of a rendezvous sequence, the jammer must be able to sense the spectrum to find them and continue to jam them. To an extent, the jammer must be a specific breed of cognitive radio.

Resources available include lab space, an AFIT network computer, an AFIT laptop, three WARP boards, MATLAB licenses, various LaTeX compilers, and a Wi-Spy USB spectrum analyzer.

1.5 Thesis Organization

The thesis is organized into five chapters. Chapter II provides the basic concept of CRNs and the background concepts that allow CRs to work, like the rendezvous processes and spectrum sensing. Chapter II also includes background information about jammers. Chapter III presents the methodology for constructing a cognitive radio network in simulation, which characterizes the wireless channel and stand-off jammer. Chapter III also presents the methodology for constructing the same cognitive radio and jammer network utilizing WARP communication boards to characterize the network in the hardware layer. Chapter IV provides analysis and results for the network models from subsequent chapters. Chapter V presents conclusions and summaries from the thesis, and suggests future areas for research.

II. Background

This chapter provides the background for topics discussed in this thesis. First, it describes cognitive radios and their properties of operation, including spectrum sensing, rendezvous processes, and channel-hopping methodology. Next, there is a brief section describing jamming concepts. Lastly, there is a section discussing prior research efforts.

2.1 Cognitive Radio

A CR is a technology that enables more efficient use of spectrum among a multitude of users. Low-usage rates of allocated spectrum in wireless communications has prompted the methodology of CRNs to scan their environment and utilize unused channels. Opportunistic spectrum utilization is more efficient than strict frequency assignments and allows for more users to be present on the same band without interfering with licensed users, also known as PUs [3], [4].

Challenges with large-scale implementation of CRNs still exist and are the target of active research. Such challenges include spectrum sensing and management, signal processing, spectrum access control, autonomous network discovery and organization, routing, learning mechanisms, and adaptive control methods. With a space-time spectrum-diverse environment, distributed structure of the spectrum for access is appropriate for coordination among users [3], [4].

Channels are defined as being available for use by CRNs if the channel can be used without interfering with PUs. Since there is no centralized control node to control the CRNs, the CRNs are therefore required to organize autonomously through collaboration among themselves. In CRNs, communication nodes are said to be neighbors if they are within transmission range of one another and share at least one common channel. Due to PU activities and fellow CRNs, CRNs will have varying availability to channels of the spectrum. Therefore, effective and efficient spectrum sharing is highly dependent on sharing spectrum information and sharing strategies during collaboration. Since new CRs can join at any time, further CRN time synchronization may not be available to all the nodes within the network [3], [4].

2.2 Spectrum Sensing

Spectrum sensing is a generalized problem in detecting and understanding what signals are present in a given frequency band. The problem of spectrum sensing is further complicated within the framework of CRNs, since there are potentially many CR users in a given frequency band. Because CRNs act opportunistically and cooperatively, spectrum sensing therefore needs to be cooperative as well.

Spectrum sensing is the most important component of cognitive radio. Spectrum sensing is the task of obtaining awareness about the spectrum usage and existence of primary users in a geographical area. The ability to measure and learn of the parameters related to the radio channel characteristics is the cornerstone of the cognitive radio concept, thereby enabling CRNs to operate opportunistically and yet not impede PUs. CRNs can obtain awareness of spectrum by using energy detector-based sensing, cyclostationary sensing, radio-identification based sensing, and matched filtering [8].

Matched Filtering.

Known patterns are generally utilized in wireless systems to assist with synchronization or for identification. Such patterns include preambles, midambles, regularly transmitted pilot patterns, spreading sequences, etc. A preamble is a known sequence transmitted before each burst, and a midamble is transmitted in the middle of a burst or slot. If a pattern is known, sensing can be done by cross-correlating the received signal with the known pattern. Waveform-based sensing, also called coherent sensing, outperforms energy detector-based sensing in reliability and convergence time. Additionally, the longer the known pattern is, the higher the performance of the waveform sensing [8].

Cognitive radios are able to detect the presence of a PU or another cognitive radio using waveform-based sensing. A basic algorithm for matched filtering is presented in Algorithm 1 [8].

Algorithm 1 Basic Matched Filtering with known preamble

```
Require: $n$ = number of lengths of preamble within Received Signal
  for index = 1 : $n$ do
    response(index) = correlate preamble with Received Signal(index)
  end for
  coherent response =abs(sum(response))
  if coherent response $\geq$ threshold then
    Intended User is Present
  else
    Intended User is Not Present
  end if
```

Matched filtering can also be done by correlating an entire signal instead of just the preamble. Generally, as the preamble increases in length, the more likely the matched filtering will output an appropriate coherent response in the presence of an intended user, also known as a successful detection. As seen in Fig. 1, waveform-based sensing is also much more accurate than energy based detection and about as accurate as matched filtering. Therefore, so long as a pattern like a preamble is known, waveform-based sensing offers a highly favorable accuracy-to-complexity ratio. In the event that a pattern is not known, estimation algorithms could be implemented but would significantly complicate the sensing operation and degrade accuracy. Alternatively, given enough observation time for an individual system, a waveform pattern could be determined and utilized. If a pattern is not known, energy-

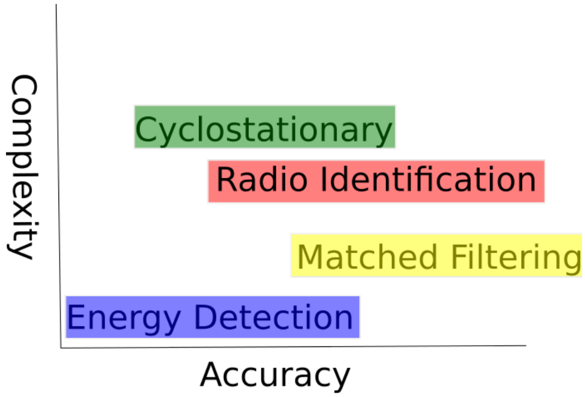

Figure 1. Main sensing methods in terms of accuracies and complexities [8]

based sensing is likely a better method to use; even though sensing accuracy would suffer, implementation complexity would also be considerably reduced [8].

Energy Detector-Based Sensing.

Energy detector-based sensing, also known as radiometry or periodogram, is the most common spectrum sensing method in use. It is computationally less complex than other algorithms and is also simpler to implement. An energy-based detector does not require any information about a system in order to detect it; however, the user is unable to differentiate interference and noise from a user. Energy-based detectors also have poor performance in low SINR scenarios, have poor performance detecting spread spectrum signals, and it is difficult to pick a threshold that will detect a PU reliably [8].

For CRs, energy-based detection is not the preferred method of spectrum sensing, due to the detection ambiguities involved. Since it is reasonable to assume CRs will have prior information about one another to facilitate communication, matched filtering is highly preferable for its increased performance and similar implementation complexity. When signal information necessary for matched filtering is unavailable, energy-based detection is a reasonable alternative for spectrum sensing. Therefore, energy-based detection, while not preferable for CRs, is a good option for CJs because of the limited user information available to the CJ.

Other Spectrum Sensing Techniques.

Cyclostationary feature detection works by exploiting the cyclostationary features in a received signal. Cyclostationary features are caused by periodicity in a signal, a signal's statistics like mean and autocorrelation, or can be intentionally induced into the received signals. A cyclic correlation function is used to detect signals present in a given spectrum. Cyclostationarity detection algorithms can differentiate noise from primary users signals because noise is wide-sense stationary (WSS) with no correlation to the signals of interest. However, since the modulated signals are cyclostationary with spectral correlation due to the redundancy of signal periodicities, the modulated signals can be detected in the presence of noise. Furthermore, the correlation properties of cyclostationarity can be used to distinguish different types of transmissions and primary users [8].

Cyclostationary algorithms are significantly more complex than matched filtering algorithms. Therefore, cyclostationary algorithms are not preferable for use in CR applications. Also, it is unlikely that CRs would have statistical information required for cyclostationarity but would not have identifiable information required for matched filtering. Cyclostationarity could be useful in CJ applications where statistical information regarding the signal of interest is known or where a CJ has enough observation time to determine said statistical information. Although cyclostationarity is a promising spectrum-sensing method for CJs, it is out of the scope of this thesis and therefore will not be investigated.

Radio identification-based sensing requires a complete knowledge regarding the transmission technologies used by PUs. CRs need to use feature extraction algorithms to successfully identify other CRs before communication. The two main tasks involved in feature extraction for communications systems are initial mode identification (IMI) and alternative mode monitoring (AMM). In IMI, the CR immediately searches for a transmission mode upon the powering on of the CR. AMM is the task of monitoring other modes while the CR is communicating in a certain mode [8].

Radio identification is more complex than matched filtering and requires more extensive knowledge from the other users. A CR would need to have a library of features about all the users in the spectrum and use embedded logic to determine which user is identified by which combination of features. If several users have the same combination of features when fewer features are available, misidentification of users could be common. While feature-based identification may be necessary in some CR applications, the complex bookkeeping is cumbersome and not ideal when unique identifiers, such as preambles, are available. Some CJ applications could make use of radio identification. In particular, when a CJ has limited identifying information on its signal of interest with no unique identifiers, the CJ is more likely to identify the signal of interest than by using energy-based detection.

2.3 Binary Phase Shift Keying

This research only looks at Binary Phase Shift Keying (BPSK) as a type of signal modulation and demodulation. To keep tests consistent and simple, BPSK is the modulation used throughout, since it is indicative of radio data transmission. BPSK is a data sequence with entries only being 1 or -1 that is modulated on a carrier frequency. The carrier frequency is mapped from a Wi-Fi channel on the 2.4 GHz band, which contains 14 channels. In BPSK modulation, the modulating data signal shifts the phase of the waveform to one of two states, 0 or 180 degrees. Fig. 2 shows an example of a BPSK and jamming waveforms that are used commonly in this thesis.

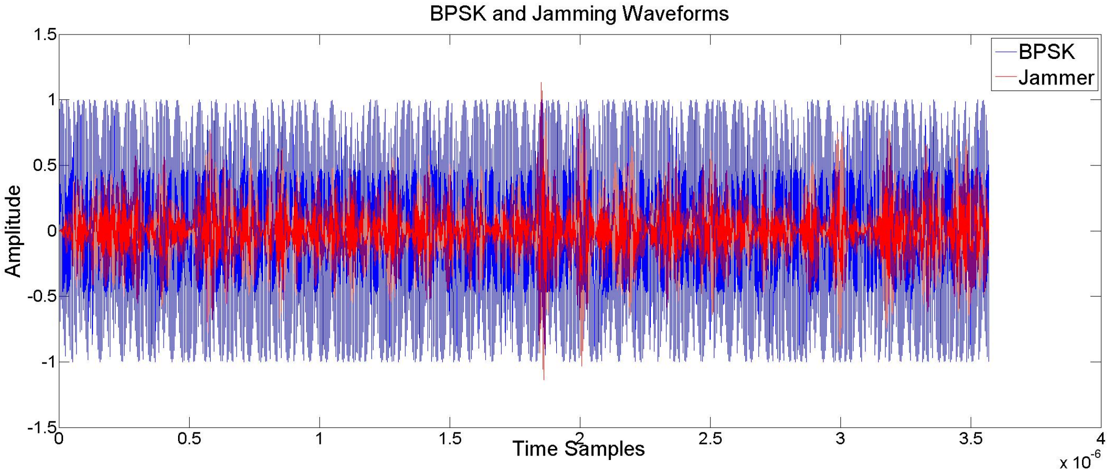

Figure 2. Example BPSK and jamming waveforms in time domain $f_s = 241.2MHz$**,** $T = 1.428ns$**, square pulse shape**

When there is a change in phase, the waveform changes abruptly. The example presented in Fig. 2 makes it difficult to see the abrupt transitions for several reasons.

There are 10 samples per message bit, so phase changes can only happen every 10 samples. More importantly, the signal is oversampled by 100 MHz more than required by Nyquist criteria to prevent aliasing in the signal processing. Therefore, a lot of data "in between" bits are included, rather than just the 1 and -1 values that are expected. However, it is more important to present a signal used in this thesis as an example rather than a generic BPSK plot, even if the generic plot is simpler [9].

2.4 Rendezvous

The rendezvous problem is when two or more radios are attempting to find one another and establish connection in a dynamic-spectrum access (DSA) environment. CRNs attempt to work within this environment and connect opportunistically, while avoiding interfering with PUs. Therefore, the rendezvous problem is highly applicable to CRNs, since they will have to rendezvous as often as necessary to avoid conflicting with PUs [10]. The main performance metrics associated with rendezvous methods are guaranteed convergence of the rendezvous process and mean time to rendezvous [10], [11].

The rendezvous problem is broken into a basic taxonomy of aided or unaided techniques. Using an aided system consists of a centralized controller directing CRs to available channels and may even have the capability to set up communication links and schedule transmissions. Using an unaided system leaves the radios to their own capabilities to find one another in a common spectrum. Unaided techniques that do not require any centralized control channels are referred to as blind rendezvous techniques [10], [11].

In a CRN, channel-hopping is a common technique to accomplish connection in a blind rendezvous scenario. In channel hopping (CH), each radio in the CRN changes the channel it is operating on within its available channels in order to connect with neighbors in the CRN. If all radios in the network have the same available channels, it is called a symmetric model. Otherwise, it is called an asymmetric model, when radios might have different channels available [11].

Table 1 and Table 2 provide a concise summary of the maximum time to rendezvous and the expected time to rendezvous, respectively. Here, the variables M, P, and G represent the number of channels available in the entire spectrum, the next greater prime number from M, and the number of channels available to all users, respectively. Note that it is possible for more than two users to rendezvous, but rendezvous performance metrics for more than two users will not be discussed [11].

As seen in Table 1 the asymmetric model typically requires longer rendezvous times than the symmetric models. Also, not all rendezvous algorithms included are expected to converge. The Jump Stay and Generated Orthogonal Sequence methods are generally better in terms of reliability in the rendezvous process. Fig. 3 is an analysis of the tabled data in Table 1, but only for the Jump Stay, Generated Orthogonal

Table 1. Comparison of upper bound of maximum time to rendezvous for 2 users [11]

Algorithm	**Symmetric Model**	**Asymmetric Model**
Jump Stay	$3P$	$3MP(P-G)+3P$
Generated Orthogonal Sequence	$M(M+1)$	NA
Modular Clock	$2P$ (not guaranteed)	unknown
Modified Modular Clock	unknown	unknown
Deterministic Rendezvous Sequence	$2M+1$	NA
Channel Rendezvous Sequence	$\geq (P-1)(3P-1)$	$P(3P-1)$

Table 2. Comparison of upper bound of expected time to rendezvous for 2 users [11]

Algorithm	**Symmetric Model**	**Asymmetric Model**
Jump Stay	$\frac{5P}{3}+3$	$\frac{2MP(P-G)+(M+5-P-(2G-1)}{(M)P}$
Generated Orthogonal Sequence	$\frac{(M^4+2M^2+6M-3)}{(3M^2+3M)}$	NA
Modular Clock	$2P^2/(P-1)$	unknown
Modified Modular Clock	unknown	unknown
Deterministic Rendezvous Sequence	unknown	NA
Channel Rendezvous Sequence	unknown	unknown

Sequence, and Modular Clock rendezvous methods.

There are several values of M worth noting. When M is equal to 3, the upper bound of maximum time to rendezvous values for Jump Stay, Generated Orthogonal Sequence, and Modular Clock rendezvous methods are very close. However, for M values above 3, the upper bound of maximum time to rendezvous values begin to diverge. The upper bound of maximum time to rendezvous values when M equals 14 is worth noting, since the number of channels used in this thesis is 14. When M equals 14, the expected time to rendezvous (ETR) is 51 for Jump Stay, 182 for Generated Orthogonal Sequence, and 34 for Modular Clock. While the ETR for Modular Clock is less than Jump Stay at M equal to 14, it should be noted that the convergence of Modular Clock is not guaranteed. The values for G in the asymmetric model are assumed to be half of M at all times. At M equal to 14, Jump Stay has an ETR of 6981 under the asymmetric model, while the Generated Orthogonal

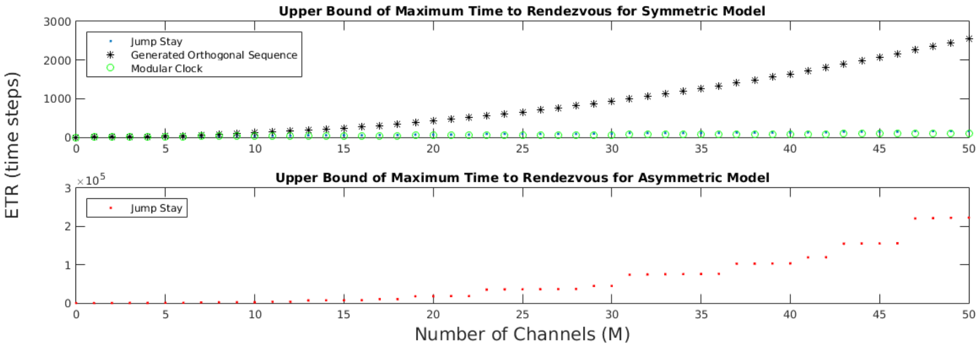

Figure 3. Comparison of maximum time to rendezvous for varied number of channels

Sequence and the Modular Clock methods do not converge under the asymmetric model. When M equals 50, the highest number of channels analyzed, the ETR is 159 for Jump Stay, 2550 for Generated Orthogonal Sequence, and 106 for Modular Clock. Again, the Modular Clock shows better potential performance but does not have guaranteed rendezvous. At M equal to 50, Jump Stay has an ETR of 222675 under the asymmetric model.

As seen in Table 2, like in Table 1, the asymmetric model typically requires longer rendezvous times than the symmetric models and not all rendezvous algorithms included are expected to converge. Fig. 4 is an analysis of the tabled data in Table 2 but only for the Jump Stay, Generated Orthogonal Sequence, and Modular Clock rendezvous methods.

There are several values of M worth noting. When M is equal to 9, the upper bound of maximum time to rendezvous values for Jump Stay, Generated Orthogonal Sequence, and Modular Clock rendezvous methods are very close. However, for M values above 9, the upper bound of maximum time to rendezvous values begin to diverge. The upper bound of maximum time to rendezvous values when M equals 14 is worth noting, since the number of channels used in this thesis is 14. When M equals 14, the ETR is 31.333 for Jump Stay, 53.0661 for Generated Orthogonal Sequence, and 36.125 for Modular Clock. For expected ETR values, the Modular Clock no

Table 3. Overall summary of representative channel-hopping algorithms [11]

Algorithm	**Need time sync?**	**Need extra info?**	**Works w/symmetric model?**	**Works w/asymmetric model?**	**Uses identical sequence?**
Jump Stay	No	No	Yes	Yes	No
Generated Orthogonal Sequence	No	No	Yes	No	Yes
Modular Clock	No	No	No	Yes	No
Modified Modular Clock	No	No	No	Yes	No
Deterministic Rendezvous Sequence	No	No	Yes	No	Yes
Channel Rendezvous Sequence	No	No	Yes	Yes	Yes

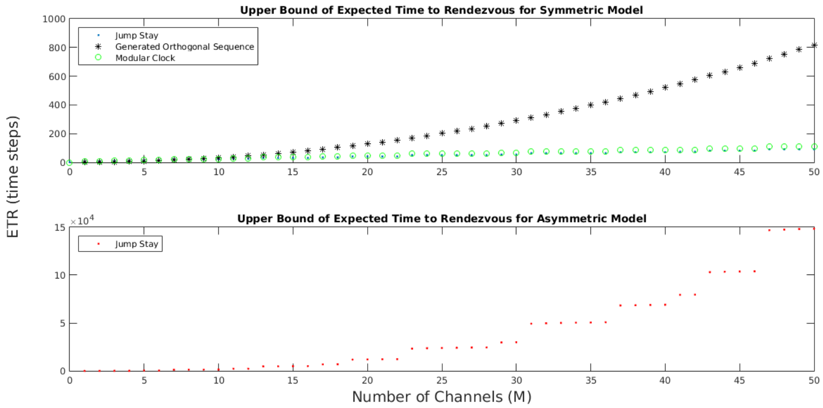

Figure 4. Comparison of Expected Time To Rendezvous for Varied Number of Channels

longer shows better potential performance over Jump Stay. The values for G in the asymmetric model are assumed to be half of M at all times. At M equal to 14, Jump Stay has an ETR of 4642.308 under the asymmetric model, while the Generated Orthogonal Sequence and the Modular Clock methods do not converge under the asymmetric model. When M equals 50, the highest number of channels analyzed, the ETR is 91.333 for Jump Stay, 817.686 for Generated Orthogonal Sequence, and 108.038 for Modular Clock. At M equal to 50, Jump Stay has an ETR of 148454 under the asymmetric model.

An overall summary of representative channel-hopping algorithms is presented in Table 3. Given the superior performance characteristics described in Tables 1 and 2 and the small informational footprint and reliability described in Table 3, the Jump-Stay algorithm was chosen as the choice of rendezvous method in this thesis.

Jump-Stay Channel Hopping.

The Jump-Stay channel-hopping algorithm for blind rendezvous is a promising method for CRs to connect with one another. Jump-Stay does not require time synchronization, does not require additional information from the other radios, does work under the asymmetric and symmetric models, does not require users to use an identical channel-hopping sequence, and has guaranteed rendezvous convergence for both the symmetric and asymmetric models [11].

As is standard for most CH, the network is assumed to have discrete time slots of fixed length. At each time slot, each radio hops on a channel in an attempt to rendezvous with other radios. Time synchronization is not available in the network, however the overlap of two time slots is sufficient to complete all necessary steps for rendezvous. In this sense, the CH sequences are equivalent to be slot-aligned, even without the time synchronization. Two CH sequences are considered to be slot-aligned, so long as the sequences overlap at some point [11].

The Jump-Stay algorithm generates CH sequences in rounds, with each round consisting of two jump patterns and one stay pattern. The jump pattern and stay pattern are specific segments of CH sequences. A radio will change its channel of operation during the jump sequences and remain on its current channel during the stay sequence. In each round of sequences, the jump sequence precedes the stay sequence. The number of channels available to a radio is represented as M. The radio user must first choose three parameters beforehand [11]:

- P– the smallest prime number greater than M
- r – the step length that exists on the set $[1, M]$
- i – an index on the set $[1, P]$
- Note: The user does not need to determine the time slot t, as it will be indexed as the algorithm iterates. However, the initial value of t is zero.

During each round, the jump pattern consists of $2P$ time slots and the following stay pattern consists of P time slots, such that each round has $3P$ total time slots.

In the jump pattern, the radio starts with index i and keeps incrementing by step length r by using the modulo operations on P. In the following stay pattern, the radio remains on channel r for the remainder of the round. Therefore, the algorithm for Jump-Stay channel-hopping logic is presented in Algorithm 2 [11].

Algorithm 2 Jump-Stay channel-hopping [11]

```
Input: $M$, $P$, $r$, $i$, $t$
Output: channel($j$)
$t$ = modulo($t$,3$P$)
**if** ($t < 2P$) **then**
    $j$ = modulo($i + tr - 1$,$P$) +1;
**else**
    $j = r$;
**end if**
**if** ($j > M$) **then**
    $j$ = modulo(channel($j$)$-$1,$M$) +1;
**end if**
```

In Alg. 2, the modulo operation is written such that "modulo(t,3P)" mathematically represents the remainder of $\frac{t}{3P}$. Additionally, the first "IF" statement represents the jump pattern and the "ELSE" portion represents the stay pattern. The last "IF" statement represents a remapping operation, which is required in the event that the channel index j exceeds M. Since P is greater than M, it is possible for j to exceed M. Fig. 5 shows a visualized representation of how the inner and outer rounds of the Jump-Stay algorithm work and increment parameters. As shown, the radio will "jump" channels for $2P$ time slots and "stay" for P time slots for each inner round. The radio will jump channels according to the formula $(i+tr-1)\%P+1$, where "$\%''$ represents the modulo math operation, as seen in Fig. 5. Each outer round consists of M inner rounds, which totals $3MP$ time slots. The radio will continue to channel hop until there is a successful rendezvous.

If a rendezvous is perceived as requiring too many time slots, the radio user can define a number of time slots to be a cutoff, or timeout, and re-start the rendezvous

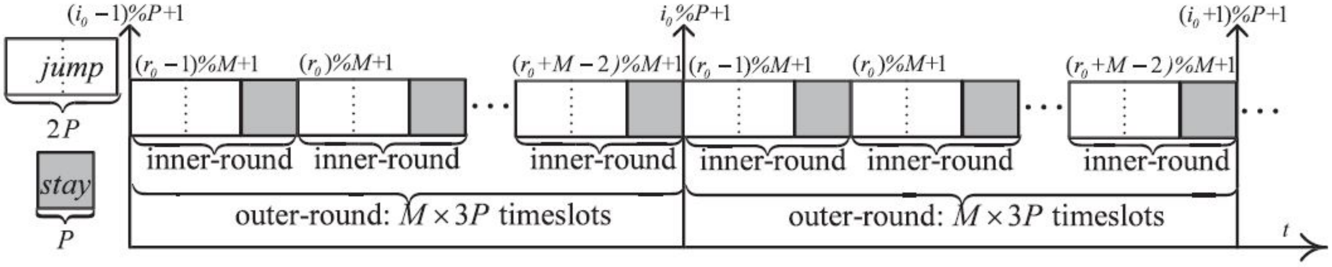

Figure 5. Channel Hopping Sequence: inner and outer rounds of Jump-Stay [11]

process. Since the Jump-Stay algorithm is guaranteed to rendezvous within a reasonable amount of time, such a timeout is not strictly necessary, but is recommended to prevent the unlikely event of a non-convergent rendezvous.

Each radio will call the Jump-Stay algorithm in each time slot instance until rendezvous is achieved. The parameter P will be constant, but r and i should be adjusted properly, according to the following rules, to guarantee the rendezvous of users [11]:

1. r must be an integer within $[1, M]$.
2. In each round of $3P$ time slots, r must be constant.
3. r is indexed within $[1, M]$ every $3P$ time slots.
4. i must be an integer in within $[1, P]$.
5. i is indexed within $[1, P]$ every $3MP$ time slots.

A numerical example for the Jump-Stay algorithm is shown in Fig. 6. In this example, $M = 4$, $P = 5$, $r1 = 2$, $i1 = 3$, $r2 = 1$, and $i2 = 2$. The nomenclature $r1$ represents the r value for Radio 1. Radio 1 and Radio 2 have channel hopping sequences of [3 1 2 4 1 3 1 2 4 1 2 2 2 2 2] and [2 3 4 1 1 2 3 4 1 1 1 1 1 1 1], respectively [11].

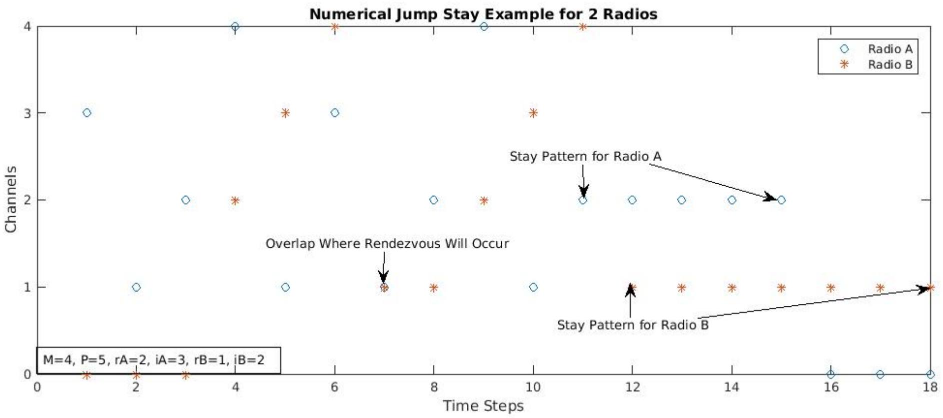

Figure 6. Jump-Stay Example [11]

Radio 1 begins its channel-hopping sequence at time step 1 and ends at time step 15. Radio 2 begins its channel-hopping sequence at time step 4 and ends at time step 18. The radios are started at different time step indexes to show that the radios do not have to be synchronized to successfully rendezvous using the Jump-Stay algorithm. Where the radios do not have overlapping sequence indexes, the channel values for the radios are set to 0. As seen in Fig. 6, the radios will rendezvous at time step 7 on Channel 1. The stay patterns are also notated in Fig. 6 for the radios. Radio 1 and Radio 2 have stay patterns on Channels 2 and 1, respectively [11].

2.5 Jamming

Whether it is a radar, communication, or global-positioning system (GPS), all wireless transmission systems are susceptible to jamming and interference. In general, jamming and interference are described as external sources that degrade or deny the use of a wireless transmission system. While interference is not considered to be intentional, jamming is considered to be intentional, and therefore malicious [12].

Jamming is typical for application against data link communications systems and is subdivided according to what modulation scheme the communication itself is using. The type of noise jamming involved is typically more effective if it inserts noise in an envelope mimicking the modulation of the communications system that is being jammed. Typical modes include amplitude modulation (AM) noise jamming, frequency modulation (FM) noise jamming, random binary code modulation jamming, and pseudo-noise code minimum shift keying (MSK) modulation jamming [12].

2.6 Prior Research

Research efforts have been made to develop neighbor-discovery algorithms for CRNs that are resilient to jammers. A noise jammer's interaction with its environment has only been analyzed in relatively simple scenarios. The scenarios presented are that of a jammer tuned to a single center frequency and a jammer that changes its center frequency periodically in a random manner. While CRN countermeasure efforts have been made, it is worth noting that the assumption is that the jammer itself is not cognitive or capable of spectrum sensing in generalized cognitive radio research [5].

Additionally, research has been conducted to develop anti-jamming techniques for space communication systems. A "cognitive jammer reactively senses channels using energy detection and jams the channel using 'detect and jam' strategy while the legitimate transmitter-receiver pair uses a joint frequency hopping and power/rate adaptation approach to avoid the impact of the jamming" [7]. While this research is more similar to that of this thesis, the results are intended for space systems and has the radios and jammers interacting in a "zero-sum" game, rather than developing more generalized strategies for the radios and jammers. The jamming and anti-jamming strategies investigated in this thesis are to be considered more general and more aggressive in nature than the prior research in space systems [7].

This thesis aims to understand the vulnerabilities of the Jump-Stay algorithm at the waveform level. The vulnerabilities are specific to the rendezvous process and are due to jamming. Jamming of the rendezvous process is accomplished by using a CJ with techniques suggested and developed in this thesis.

2.7 Conclusion

A CR is a technology that enables more efficient use of spectrum among a multitude of users. Opportunistic use of the EMS requires the CR to be capable of conducting accurate spectrum sensing. In this thesis, all spectrum sensing for CRs is accomplished using a matched filtering method and for CJs is accomplished using energy-based methods. All communication signals produced by radios in this thesis are BPSK modulated signals. In order to establish a communications link, the radios must rendezvous with one another. Rendezvous requires channel-hopping and the algorithm studied in this thesis is the Jump-Stay channel-hopping method. Jamming was conducted to interfere with the CR communications and rendezvous process to characterize the effects on CRNs. The effects of a CJ on CRNs have not been studied, but are studied in this thesis at the waveform level.

III. Methodology

This chapter provides the methodology for topics discussed in this thesis. First, it describes how the simulated cognitive radio scenario works and the logical progression of the scenario, including all the functionality required to run the simulation. Next, the hardware involved to realize the scenario is discussed. The hardware section is divided into how the required physical hardware is set up, then what functionality and code are required to create the same scenario described in the simulation section. The differences in algorithms between hardware and simulation are highlighted in the hardware section, since many of the algorithms in the simulation section are reused in the hardware.

3.1 Simulation

The simulation written is a functional baseline for proof of concept for the scenario of CRNs and jammer network. Said simulation provides a direct comparison for the hardware component that was performed with the WARP boards, such that performance parameters may be extracted and analyzed. The same parameters are extruded and analyzed from the WARP implementation.

The simulation works on a set of modular functions that are called to fit a scenario of interest. A flow chart detailing the logical progression of the CRNs and the jammer is outlined in Fig. 7. Each block in the diagram will be discussed, to include what sub-functionality is required.

The simulation is considered to be a passband system. All signals are generated for transmit and receive at frequencies 10 times lower than the Wi-Fi frequencies, but since this frequency set is consistent throughout the simulation, the simulation is still considered to be a passband system. Compared to the hardware version, which

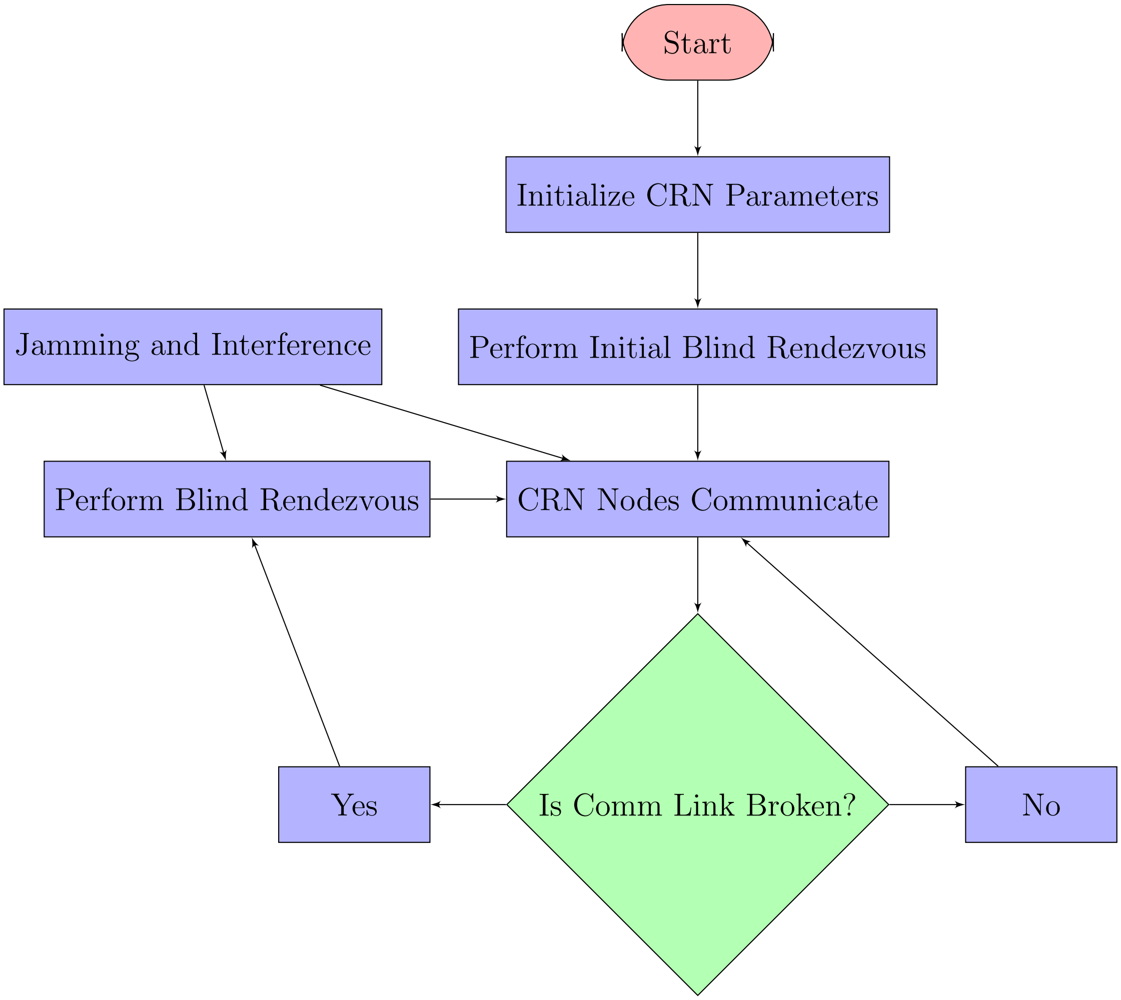

Figure 7. Generic CRN and jammer logic flow diagram

generates all signals at the actual Wi-Fi frequencies, the simulation can be considered an Intermediate Frequency (IF) system, but individually the simulation is a passband system.

Initialize CRN Parameters.

Several parameters must be defined before any functionality is implemented, since they will be inputs to functions required. The preamble, message, and samples per bit must be defined externally to the function calls that make up a scenario.

The preamble should be a fixed, pseudo-random sequence, containing only the values 1 and -1, and should only be generated once per scenario run. The preamble can change across iterations of the same scenario or different scenarios but must not change within any of the function calls themselves. The samples per bit must also be generated externally and remain constant in individual iterations of a scenario. The message may change but must have a fixed length. When the message does not change, the system operates more ideally, but said operational performance is not indicative of a realizable scenario. When the preamble is generated, it may be nearly any length, but fewer than 4 bits in length has generally been too short of a sequence for the correlator, discussed in the next sub-section, to correctly identify. With a preamble generated, the preamble must be expanded to represent N samples per bit. Therefore, the preamble must be expanded into a matrix, where each row represents each bit and the columns are the samples in each bit. The message vector is generated the same way as the preamble, except that its initial bit length is a ratio of the preamble. The preamble being 20% of the total bit sequence has been found to work well.

The variables M, P, Fs, $NoiseFigure$, and $jammerOn$ need to be initialized prior to running any scenarios, since they are function inputs. M represents the number of commonly-available channels to the radios attempting rendezvous, and P is the smallest prime number greater than M. Fs is the sampling frequency used. $NoiseFigure$ describes receiver noise by relating the system temperature to standard temperature. The noise figure is a unitless value [13]. The variable $jammerOn$ is a switch that determines whether or not jamming energy is being injected into the system under test.

The initial channel settings and r and i values must also be defined, since they are inputs. Recall that r is the channel-hopping step length and i is the channel-hopping index value. It is necessary to define the channels the radios are tuned to prior to rendezvous so that when the radios begin channel-hopping, they do not end up back on the same channel that became unusable. It does not matter whether the radios were previously on the same channel before rendezvous for the process to be successful. If the radios were tuned to the same channel prior to rendezvous, the radios presumably broke their communication link in order to move to a more advantageous channel. However, if the radios were tuned to different channels prior to rendezvous, the radios were presumably rendezvousing to establish an initial communications link.

If the radios in the network have a predefined initial r and i values, the radios will always rendezvous in 1 time step, provided proper spectrum sensing. The problem with assigning static r and i values is twofold. First, the radios become very predictable to an outside observer. For example, if the radios both always attempt to rendezvous on channel 1, it can be inferred that the radios are using an initial r value of one, making interception of the radios extremely easy. Second, if the radios decided to break their communication link and perform their rendezvous process to get away from the same channel that their initialized r value is set to, they will instead continue to go back to the very channel they sought to escape.

Additionally, if the radios do not re-initialize their channel settings, r, and i values before rendezvousing away from a channel, the radios will attempt to rendezvous again on the same channel they were trying to get away from. Therefore, a comprehensive and efficient means to ensure the radios are unlikely to repeat rendezvous on a channel they need to get away from is to reset the channel values to a pseudo-random number on the set of available channels prior to rendezvous. Setting r and i to be equal to the same as the reset channel value helps to ensure less predictable behavior and a more stable way to start any channel hopping that needs to take place. There is a small chance that the radios will all reset to the same channel they were trying to get away from, but the probability is very small. So long as the channel to be rendezvoused away from is not allowed to be used as the index i, the radios will not rendezvous on the undesired channel. Also, the only consequence of this event is that the radios would require another rendezvous if it happens. While it is a small vulnerability, it still exists.

Rendezvous.

The rendezvous processes, described in Fig. 7, utilize the Matlab function "rendezvous.m," which performs a blind rendezvous for two radios such that they end up on the same channel. The rendezvous themselves simply represent the first time the radios establish a common channel, or that the radios have had to move to different channels. Note that after a connection has been broken, the radios must re-initialize their respective channels before attempting a blind rendezvous. Otherwise, the radios will attempt to rendezvous on the same channel they were just on, which makes the move pointless.

The inputs for the rendezvous function are:

- preamble – the preamble used to generate signals. It works best as a fixed, pseudo-random sequence, containing only the values 1 and -1, and should not be generated or overwritten within the rendezvous function.

- msg – the message that is appended to the preamble before being BPSK modulated on a carrier frequency that corresponds to a Wi-Fi standard channel. It is does not matter if the message is fixed, pseudo-random, or even how it is generated. The message data contains only the values 1 and -1.

- jammerOn – binary operation to inject jamming energy into the rendezvous process or not. Valid inputs are 1 or 0. This parameter may or may not matter, based on how the jamming waveform is generated.

- Fs – sampling frequency used to generate signals. Unit in Hertz.
- NoiseFigure – describes receiver noise by relating the system temperature to standard temperature. The noise figure is a unitless value and is described by Eq. (1). Generally, the higher NoiseFigure, the lower SINR is [13].
- initial channel values for radios A and B, and the jammer.
- r values for radios A and B, and the jammer. Recall r values are the step sizes used in the Jump-Stay channel-hopping.
- i values for radios A and B, and the jammer. Recall i values are the indices used in the Jump-Stay channel-hopping.

All generated receive signals add complex noise to the transmitted signals. The complex noise added in the receive signal function follows Eq. (1) [13].

$$P_n = kT_0FB \tag{1}$$

where

- P_n represents the receiver thermal noise power
- k represents Boltzmann's constant ($1.381 \times 10^{-23} watt - sec/K$)
- T_0 is the standard temperature (also known as room temperature, $290K$)
- B is the receiver bandwidth in Hz ($20MHz$ in this case)
- F is the noise figure for the system (unitless, 1 in this case)

The channel hopping logic used in the rendezvous process is a function that executes the Jump-Stay logic described in Alg. 4. Recall that the output is the next channel in the channel hopping sequence and the inputs are:

- M – the number of commonly available channels
- P– the smallest prime number greater than M
- r – the step length that exists on the set $[1, M]$
- i – an index on the set $[1, P]$
- Note: the user does not need to determine the time slot t, as it will be indexed as the algorithm iterates. However, the initial value of t is zero.

The logical progression of the rendezvous is described in Alg. 3 and Alg. 4. Note that Alg. 3 requires knowing the preamble of the radios to work correctly, whereas Alg. 4 does not require knowledge of the preamble except that the signals utilized by both radios are the same or very similar in structure. The preambles must be the same or similar for Alg. 4 to work, otherwise any signal containing the same carrier frequency as one of the radios will cause a false-positive identification, which will result in the radio incorrectly attempting to establish a communication link. Signals that have the same, or similar, preamble modulated on the same carrier frequency will have much higher correlation output than signals with a different preamble modulated on a different carrier frequency.

In both versions of the rendezvous code, a troubleshooting metric was added to confirm whether an appropriate connection was attempted or if the rendezvous process had improperly aligned the radio channels. The rendezvous code will only continue to operate if the radios are, in fact, on the same channel. The information to affirm that the radios successfully rendezvoused is not known in a network without a centralized controller but was utilized in the laboratory setting to evaluate effectiveness of the rendezvous process itself. This misalignment code can be disabled at any time and was disabled for all testing and analysis accomplished.

Algorithm 3 Blind rendezvous using known preamble

```
  Input: preamble, msg, jammerOn,M,P,Fs,NoiseFigure,channelA,channelB,rA,iA,
  rB,iB
  Output: timeVal,t,channelA,channelB,channelJammer
  while channelA $\neq$ channelB do
Ensure: all $r$ and $i$ values are within appropriate bounds
    generate transmit and receive signals for radios
    correlate received signals with known preamble
    if Max Val of Preamble Correlation Output $\geq$ threshold then
      radios are on the same channel. Establish Connection (break while loop).
    else
      radios are not on the same channel. Implement channel hopping logic.
    end if
  end while
```

Algorithm 4 Blind rendezvous using transmitted waveforms

```
  Input: preamble, msg,jammerOn,M,P,Fs,NoiseFigure,channelA,channelB,rA,iA,
  rB,iB
  Output: timeVal,t,channelA,channelB,channelJammer
  while channelA $\neq$ channelB do
Ensure: all $r$ and $i$ values are within appropriate bounds
    generate transmit and receive signals
    correlate received signals with transmitted signals
    if Max Val of RxTx Correlation Output $\geq$ threshold then
      radios are on the same channel. Establish Connection (break while loop).
    else
      radios are not on the same channel. Implement channel hopping logic.
    end if
  end while
```

If the radios and the jammer all rendezvous to the same channel, it is ambiguous whether or not the radios are being spoofed or not by the jammer. If the jammer is broadcasting noise or not at all, the radios are not being spoofed based on the correlation properties discussed in Chapter 2. If the jammer is broadcasting a signal similar enough to the expected radio transmission pattern, the possibility of the radios being spoofed cannot be ignored in this scenario. If the radios rendezvous to the same channel, and it is different than the channel the jammer is on at the time of rendezvous, then the radios have successfully rendezvoused without the possibility of being spoofed by the jammer. If either radio or the jammer rendezvous to the same channel without the other radio also being on that channel, the radio that rendezvoused with the jammer has been successfully spoofed by the jammer. Any other rendezvous conditions are considered to be errors and indicative of a flaw in the system. Note that as presented in Alg. 4, jamming energy will only be added to receive signals that are tuned to the same channel as the jammer.

The constraints on M and P are minimal. Even in an asymmetric model, the channel set just needs to map to a physical channel with a center frequency. All work performed has been for a symmetric model but could be easily modified to represent an asymmetric model by utilizing a simple set-mapping algorithm tailored to a specific set. For example, M could equal 5 to indicate 5 common channels between the radios, which is a discrete set of channels not necessarily corresponding to the Wi-Fi channels 1 to 5. Note that P does not need to map to a set of channels but is just used in algorithmic processing during the Jump-Stay channel hopping logic.

The input r is also the channel that the algorithm will remain on during the stay pattern. A lot of the strategy for the radios to avoid interference is based on the input parameter r, since it is an observable parameter of the system. Once observed and determined, it is easy for the observer to predict and intercept the radio during rendezvous. To perform this interception, the intercepting party must be using the same rendezvous logic as the radio. Since that is the case, the only advantage to intercepting radios during their rendezvous process would be to further delay the rendezvous, which may not always be possible depending on how quickly the radios can rendezvous. As discussed later, it is much simpler and effective for the jammer to perform spectrum sensing and intercept the radios after rendezvous. Alternatively, the jammer can follow the radios in their rendezvous process as a third radio attempting rendezvous in their network, then commence jamming once the other radios have resumed communication. Note that the i and r values used for the jammer are decided internally and produces the channel the jammer hops to.

The output from the channel-hopping logic is used in subsequent iterations of the rendezvous process until the radios end up on the same channel, and a connection is established. One of the distinct advantages to using the Jump-Stay channel-hopping algorithm is guaranteed convergence of the radio channels. However, regardless of the robustness of the algorithm, it is recommended to implement a “time-out” failsafe, such that if the rendezvous does not occur within a number of time steps, the process starts over. This is a particularly useful countermeasure for a radio to implement when the radio is in a high-interference environment.

The simulation and the WARP boards have a channel bandwidth of 20 MHz, which matches the Wi-Fi standard. The CR The equation governing the calculated complex noise vector used to represent the system receiver noise is outlined in Eq. (1). Therefore, a noise figure of 1 yields a reasonable approximation of the WARP board receiver noise at room temperature [13].

The outputs of the rendezvous function are:

- timeVal – the processing time required by Matlab to complete the rendezvous process. Units are in seconds.

- t – the number of time steps required to complete the rendezvous process. This output is unitless.

- channelA – the channel on which node A has rendezvoused. It should be the same as channel B.

- channelB – the channel on which node B has rendezvoused. It should be the same as channel A.

- channelJammer – the channel on which the jammer has rendezvoused. Depending on the channel hopping logic used by the jammer, it may or may not be the same as the other radios.

The output, t, is a suitable measure of performance. Comparing the number of time steps across platforms allows proper evaluation of the systems without regard to the processing capabilities of individual systems. Observing required time steps, regardless of how long each time step takes to process, is also a fair method to evaluate the effectiveness of spectrum-sensing methods used in the different rendezvous algorithms previously presented. The processing time each system requires is an interesting metric, worthy of observation and comment, but is not suitable for proper analysis with regard to cross-system effectiveness. If multiple methodologies of channel hopping and spectrum sensing are being tested on the same platform, then the processing time is a metric that can be fairly analyzed.

The channel outputs are very useful for re-establishing communication links between the radios. The channel output for the jammer is only useful if the jammer is known to be using the same channel hopping logic as the radios being targeted. If scanning the spectrum for the new channel that the radios are on proves to be a cumbersome task, it would be beneficial to attempt to mimic the channel-hopping behavior of the radios. In any case, it is useful to attempt to predict the channel the radios will rendezvous on to give the jammer a non-arbitrary channel to use to begin spectrum-sensing.

Transmit and Receive Signal Generation.

Functionality for generating the transmit and receive signals is done throughout the simulation. In the rendezvous process, transmit and receive signals are generated as a way for radios to identify one another during their spectrum sensing processes.

All generated transmit signals are BPSK modulated preambles and messages on carrier frequencies that are mapped from Wi-Fi standard channels. The preamble and message that are used in transmit signal generation must be generated externally from the transmit signal function. The transmit signal generation function itself concatenates the preamble and message vectors, then modulates the concatenated vector on a carrier frequency. The carrier frequency is mapped from the specified channel input, which is consistent with the Wi-Fi standard. To make the required sampling frequency less cumbersome, the operation frequencies have been assumed to have been stepped down by a factor of 10 using analog mixing circuitry. The transmit signal generation is represented by Alg. 5.

Algorithm 5 Transmit signal generation

```
Input: channel, message, preamble, Fs
Output: TxSignal
if channel = 14 then
    $f_0 = 2.484 \times 10^8$ %determining center frequency $f_0$
else
    $f_0 = (0.005 \times 10^8) \times (channel - 1) + 2.412 \times 10^8;$
end if
TxSignal = [preamble message] %concatenate preamble and message vectors
$Ts = \frac{1}{Fs}$ % define sampling time interval based off of sampling frequency
$t = 0 : Ts : Ts \times (length(TxSignal) - 1)$ %define a time vector
$TxSignal = TxSignal. * sin(2 * \pi * f_0 * t)$ %modulate onto carrier frequency $f_0$
```

Complex noise is vectorized to match the size of the transmit signal, then added to the transmit signal to generate the receive signal. The receive signal logic can be represented by Alg. 6.

Algorithm 6 Receive signal generation

```
Input: TxSignal, NoiseFigure
Output: RxSignal
$k_B = 1.381 \times 10^{-23}$
$T_0 = 290$
$B = 20 \times 10^6$
$noise = \sqrt{k_B * T_0 * NoiseFigure * B/2}. * ones(1, length(TxSignal)) + 1i. \times randn(1, length(TxSignal))$ %complex noise row vector
$RxSignal = TxSignal + noise$
```

CRN Communication.

The CRN communication is accomplished using the same functions to generate the transmit and receive signals. An iteration of CRN communication constitutes generating transmit and receive signals, generating any jamming and interference sources, and checking the suitability of the communication link. Any channel hopping is performed only during the rendezvous process, which happens to establish an initial communications link, or when a communications link is found to be unsuitable.

Checking the Suitability of the Communication Link.

In a system that operates in real-time, the suitability of the communication link will require occasional checking to ensure that the channel is still usable. For this thesis, such functionality was developed and tested, but was not implemented since it was not needed to test for delaying the rendezvous process for CRs. All scenarios involved the assumption that rendezvous was necessary and it was irrelevant as to whether it was the initial rendezvous or a rendezvous required due to an unusable communications link.

The radios will regularly check on the performance of their current communication link. The two ways the radios can check on the quality of their communication link is by computing bit error ratio (BER) or SINR. The checking function works by inputing the received signal,the preamble, the number of samples per bit, and the option of evaluating the BER or SINR and outputs a binary logical value that indicates whether or not the connection is suitable to keep open.

The radios check BER by comparing the received signal with the known preamble. First, the received signal is aligned such that only the portion containing the preamble is considered. The alignment is accomplished by correlating the received signal with the known preamble, then indexing the received signal according to the output of the correlation such that the received signal only has values where it contains the preamble. Next, the error is computed as the difference between the received preamble and the known preamble. The error is translated to a ratio of number of errors to number of bits. The BER will be a number between 0 and 1, so for 1 error in 10 bits, the BER would be 0.1. If the BER exceeds the threshold, the communication link will break and cause the simulation to enter the rendezvous process.

The radios check SINR by taking the same error values computed for the BER, but in this case the error values are best thought of as noise and interference. The SINR is then computed as the ratio of the preamble signal power to the noise and interference power. The mean value of the SINR is then compared to a threshold. Mean values of the SINR equal to 1 indicate equal signal and noise power, whereas values less than 1 indicate mean signal power being lower than the mean noise power, and values greater than 1 indicate mean signal power being greater than the mean noise power. The SINR is expected to be less than 1; however if the SINR is less than the threshold, this will break the communication link and cause the simulation to enter the rendezvous process.

For proof of concept purposes, it is appropriate to use the communication suitability output from each radio, because that information is readily available. However, in a real network, that information may not be available. Therefore, a more appropriate approach in such a situation would be to have the radios independently decide to break the communications link. If one radio breaks connection and enters the rendezvous process before the other, the latter radio will also break connection and enter the rendezvous process since their BER and SINR will immediately become insufficient to maintain the communications link.

Jamming and Interference.

In order to generate the jammer signals, the jammer must first find the center frequency of the transmitted signals. The jammer does this using a one-sided periodogram approach [14]. Once the power spectral density (PSD) is estimated, some logical indexing is used to isolate the frequency band of interest, which are frequencies greater than 240 MHz. Again, the simulation assumes there has been analog circuitry that has stepped down the frequencies by a factor of 10. The frequency location of the maximum value in the PSD estimate is considered to be the center frequency of the transmitted signal. For signals that are modulated on a carrier frequency, this method of determining the center frequency is appropriate. Once the center frequency is determined, it is mapped back into a channel on the Wi-Fi set. Note that if the jammer is targeting specific radios, identifying information unique to the target will be required to precisely jam a specific radio. Therefore, simply using the center frequency to identify the radio would not be an appropriate measure in a crowded spectrum where the jammer is seeking an individual target.

Once the jammer has found the center frequency of the radio in question, the jammer produces noise that is band-limited to the vicinity of that center frequency.

The bandwidth of the generated noise signal is 20 MHz, which is the same bandwidth of the transmitted signals in both the simulation and the WARP boards. In order to be more covert, the jammer could alternatively produce a pseudo-random sequence the same length as the transmitted signal modulated on the same carrier frequency. Generating such a signal would look a lot more like another communication system and would be less obviously a jammer. However, for sabotaging the SINR or BER of the communication system, it is not necessary for proof of concept, but should be considered in a fielded device.

Jamming the Rendezvous Process.

The rendezvous process required for cognitive radio interaction was identified as a potential weakness to exploit through strategic jamming. Traditional jamming is typically done with high power levels and wide bandwidths. Jamming the CRs during the rendezvous process is a more elegant approach with the benefit of less electronic collateral damage. Additionally, analysis in jamming the rendezvous process provided insight into how to best guard against weaknesses.

To understand how to most effectively jam the rendezvous process, several steps were taken. First, strategies to jam the rendezvous process were explored and tested for maximum potential effectiveness. Next, the jamming waveform was optimized for degradation against the correlator that determines whether or not rendezvous has been successful or not. Then behavioral performance metrics were analyzed for effectiveness of the jamming strategies.

Jamming Strategies.

Strategies were explored for maximum effectiveness in jamming the rendezvous process. In order to determine potential, the strategies were first described and tested at the network level. The potential of each strategy was measured in how efficiently the jammer could recreate the channel-hopping pattern of a radio. For a single value of M, a radio sweeps all possible combinations of i and r, and for each i and r combination the radio produces a channel-hopping sequence of arbitrary length. For every channel-hopping sequence the radio produces, each jamming strategy creates its own channel-hopping sequence in an attempt to match the channel-hopping sequence produced by the radio and is evaluated for efficiency. The efficiency is measured as the number of times the jammer was on the same channel as the radio divided by the total length of the channel-hopping sequence, as given by:

$$\eta_{jammer} = \frac{Correct\ Jammer\ Channel - Hops}{Total\ Channel - Hops} \tag{2}$$

The first strategy explored considers the possibility of estimating Jump-Stay parameters from observed channel-hopping data. To first understand the potential in this method, the jammer is assumed to be capable of observing all of the channels simultaneously and correctly identify every channel hop. Once the jammer has estimated the parameters for Jump-Stay to recreate the radio's channel-hopping sequence, the jammer delays its sequence appropriately and "catches up" with the radio's channel-hopping sequence.

To estimate Jump-Stay parameters from observed data, the jammer must first create a history of observed data. After three consecutive observations that observe the radio "channel wrap-around" effect, e.g. the radio channel-hopping from channel 13 to channel 3, the jammer is able to begin making calculations for estimation. The jammer takes the three observations and performs a difference operator on them, giving:

$$O_n = length\, n\, vector\, of\, wrap{-}around\, observations \delta_n = [O_1 - O_2, O_2 - O_3, ..., O_{n-1} - O_n] \tag{3}$$

Observing the "wrap-around effect" is defined mathematically by analyzing δ_n from Eq. (3). When the sign of consecutive samples in δ_n changes, this is defined as observing of the "wrap-around effect." Recall that the cornerstone of Jump-Stay is the modulo operator. The importance of observing the "wrap-around" effect is so appropriate step values can be determined using the "modulo-compliment." For example, if samples in δ_n include -5 and 6, $6\%11 = 6$ and $-5\%11 = 6$ when $P = 11$; both situations need to be observed to accurately determine the correct step value, r for Jump-Stay. If there is just -5 or 6 in δ_n, r cannot be determined without a known value of P.

The observation differences need to be modulated against values of P. When the modulation outputs are the same on the same index, that output is the step value, r, and the index is the value P. The initial index in Jump-Stay, i, can be defined as the first channel observed in O_n from Eq. (3).

Estimating M from P is tricky. Since P is the smallest prime number greater than M, multiple values of M can result in the same value of P. For example, if $P = 11$, M could be any integer from 7 to 10. A good place to start is to calculate M as the highest prime number less than P, which would be 7 in the example where $P = 11$. If there was an observation greater than 7, then M is redefined as that observation. So, if $P = 11$ and M was initially estimated to be 7, but a channel value of 10 was observed, M is redefined as 10.

Checks are also done to ensure that all the M,P,r,i values are appropriately related to one another. For example, P cannot be less than M, r cannot be greater than M, etc. Violations of the checks result in recalculating the erroneous values.

So, taking Eq. (3) and a set of P values to be considered results in Alg. 7.

Algorithm 7 Estimating Jump-Stay parameters

```
  Input: $O_n$
  Output: $M$,$P$,$r$,$i$
Require: $O_n$ contains samples with “wrap-around” effect
  $\delta$ = difference($O_n$)
  for $P = 1 : max(P)$ do
    $w(P) = \delta_1 \% P$
    $z(P) = \delta_2 \% P$
  end for
  if $max(w) < max(z)$ then
    $r = max(w(w == z))$
  else
    $r = max(z(w == z))$
  end if
  if $length(find(w == r)) > length(find(z == r))$ then
    $P = find(z == r)$
  else
    $P = find(w == r)$
  end if
  $i = O_1$
  $M$ = nextprime($P$,'below')
  if maximum value of $O_n \geq M$ then
    $M$ = maximum value of $O_n$
  end if
```

When the jammer is presumed to have a limited number of channels it can observe, cut-off channel observations are treated as zeros. Therefore, the jammer requires three consecutive, non-zero observations. Observing the “wrap-around effect” becomes considerably more difficult when channels are limited, but otherwise follows the same routines as when channel observation is unlimited.

Another strategy for jamming the rendezvous process is by using a repeater. The repeater works on the idea that Jump-Stay is periodic every $3P$ channel-hops during each set of $3MP$ channel-hops. Therefore, if the jammer can observe a radio and rebroadcast based on observed channel history, the jammer can be efficient in being on the same channel as the radio by the number of channel-hops rebroadcast proportionally to the total number of channel-hops in the radio's sequence. When a jammer observes $3P$ channel-hops and rebroadcasts them, the jammer would be 100% efficient for a maximum of $3MP$ channel-hops, assuming proper delays to synchronize with the radio's channel-hopping sequence. However, it is not realistic to observe the radio's channel-hopping sequence for so long, since most radios will rendezvous much sooner. Therefore, the radio is observed for P channel-hops. Since the jump portions of the Jump-Stay are the same and P channel-hops in length, the repeater is at maximum $\frac{2}{3}$ efficient by rebroadcasting P observed channel-hops.

The final strategy considered is having a jammer channel-hop by using random Jump-Stay inputs. The jammer will rendezvous with other radios as if it is another radio in the network. Since Jump-Stay works in a symmetric or asymmetric model, it doesn't matter whether M is known or not. If M is known a-priori, the jammer will rendezvous with the radio of interest sooner than if M is not known. If M is not known, it can be reasonably estimated from channel-hop history data observed from the radios of interest. While the efficiency of this strategy is presumably low, the strategy leverages the strength of Jump-Stay against the other radios by jamming during guaranteed convergence. Jamming while other radios would otherwise rendezvous matters a lot more than jamming at any other time during the rendezvous process.

Optimizing Jamming Resources.

The jamming waveform used is band limited White Gaussian Noise (WGN). Therefore, being varied to optimize the jamming waveform are power and bandwidth. Fig. 8 shows an example of the power density estimates of the jamming waveform and the transmitted signal. In Fig. 8, the waveform has a power parameter of 1 dBW and a bandwidth of 20 MHz.

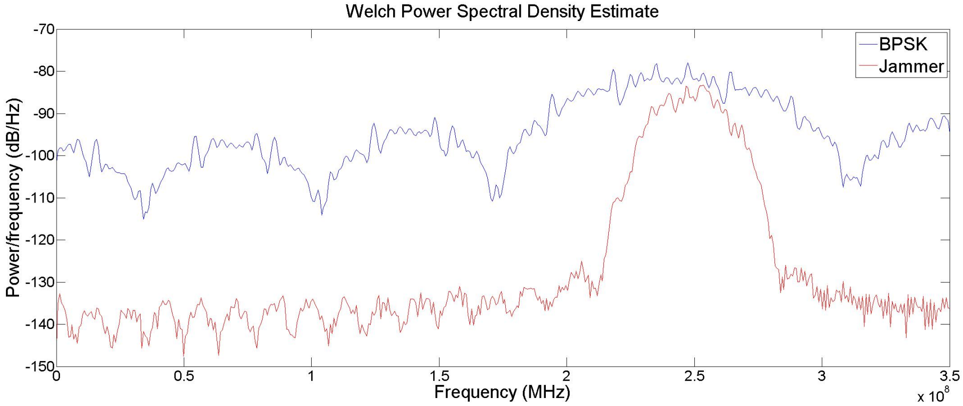

Figure 8. Spectrum of the band-limited noise and transmitted signal

Power and bandwidth combinations will be swept for maximum degradation of correlator output. The more degraded the correlator output is, the more likely rendezvous will be interrupted. Metrics for optimized degradation of the correlator output involves lowering the ratio of the correlator's peak output to the correlator's mean output, which is defined as the Peak-to-Mean-Ratio (PMR). The combination of bandwidth and power parameters that minimizes the correlator output's PMR is considered to be the optimal parameter set to jam the CRs with band-limited noise, and thus most likely to interfere with the rendezvous process.

PMR is a proxy to understand the noise present in the environment. Signal power cannot be taken away to degrade the performance of spectrum sensing operations. However, more noise can be added to the system to raise the probability of false alarm, P_{fa}, and lower the probability of detection, P_d. An alternative to analyzing PMR levels would be to test the performance of the spectrum sensing as a function of P_d versus P_{fa}. Analyzing the P_d and P_{fa} of the spectrum sensing operations as functions of various system parameters does not add value to the analysis of the rendezvous problem presented in this thesis, and is therefore considered to be out of scope.

Behavioral Performance Metrics.

Performance metrics are analyzed to determine the effectiveness of the jamming strategies on the rendezvous process. The two metrics analyzed will be time to rendezvous and rendezvous errors. Each performance metric will be measured over many realizations. Time to rendezvous is measured as the number of time steps, also called channel-hops, required to rendezvous for a given realization. Rendezvous errors are defined as the occurrence of radios rendezvousing on incorrect channels. For example, if Radio A rendezvous on channel 1 and Radio B rendezvous on channel 2, this is considered an error.

Best case and worst case scenarios were also considered for evaluation criteria. These scenarios are when no jamming is present and when the jammer is always on the same channel as the radio. Including these scenarios allows evaluation of what results are independent of the jammer and what are the optimum rendezvous-degradation results possible for this optimized jammer.

3.2 Hardware

This section describes the hardware setup and coding commands necessary to make the WARP boards complete the same tasks as the simulation in the physical layer.

Constructing a communication and jammer network using WARP.

This subsection details the components of a communications link using the WARP boards. This includes how the boards work and how MATLAB code is executed and parsed from a controlling laptop to the boards [15].

The setup consists of a laptop connected to an Ethernet switch, which runs Ethernet lines to each individual WARP board. Fig. 9 shows the overall setup. The laptop runs Matlab scripts, which parse commands via the Ethernet switch to the board desired. Fig. 10 shows the Ethernet switch used, which requires a 1 Gigabit performance specification [15]. WARP boards are able to be commanded by specific board and daughtercard [15], [16], [17].

Each WARP board is a field-programmable gate array (FPGA) with four daughtercard radios, which can be configured for up to a 4×4 multi-input multi-output (MIMO) communication system. Figure 9 shows a version 2.2 WARP FPGA board with daughtercard radios installed. Each WARP radio board contains a digital-to-analog converter (DAC) for the transmitted digital in-phase and quadrature (IQ) signals, two analog-to-digital converters (ADCs) for received digital IQ and digital received signal strength indication (RSSI) respectively, a RF transceiver for transmitted and received analog IQ and received analog RSSI signals, a dual-band power amplifier for transmitted RF, an antenna switch for transmitted and received RF, and a port for an antenna. Fig. 11 shows a daughtercard radio board, and Fig. 12 shows the system block diagram. Installing four radio boards into the WARP FPGA board enables up to 2×2 MIMO wireless communications per WARP board [15], [16], [17].

Each WARP board has one antenna per daughtercard radio. The antennas connected to the daughtercards are a mono-static arrangement and can be used for transmit and receive based on any desired duty cycle. The three WARP boards are arranged such that the outside boards, as shown in Fig. 9, have their antennas mounted above the boards, while the inside board has two of the antennas in the front of it and the other two behind it. Fig. 9 shows the left-most WARP board, Fig. 13 shows the middle and right-most boards, which are consistent with Fig. 9. Fig. 14 shows the antenna setup for the outside boards.

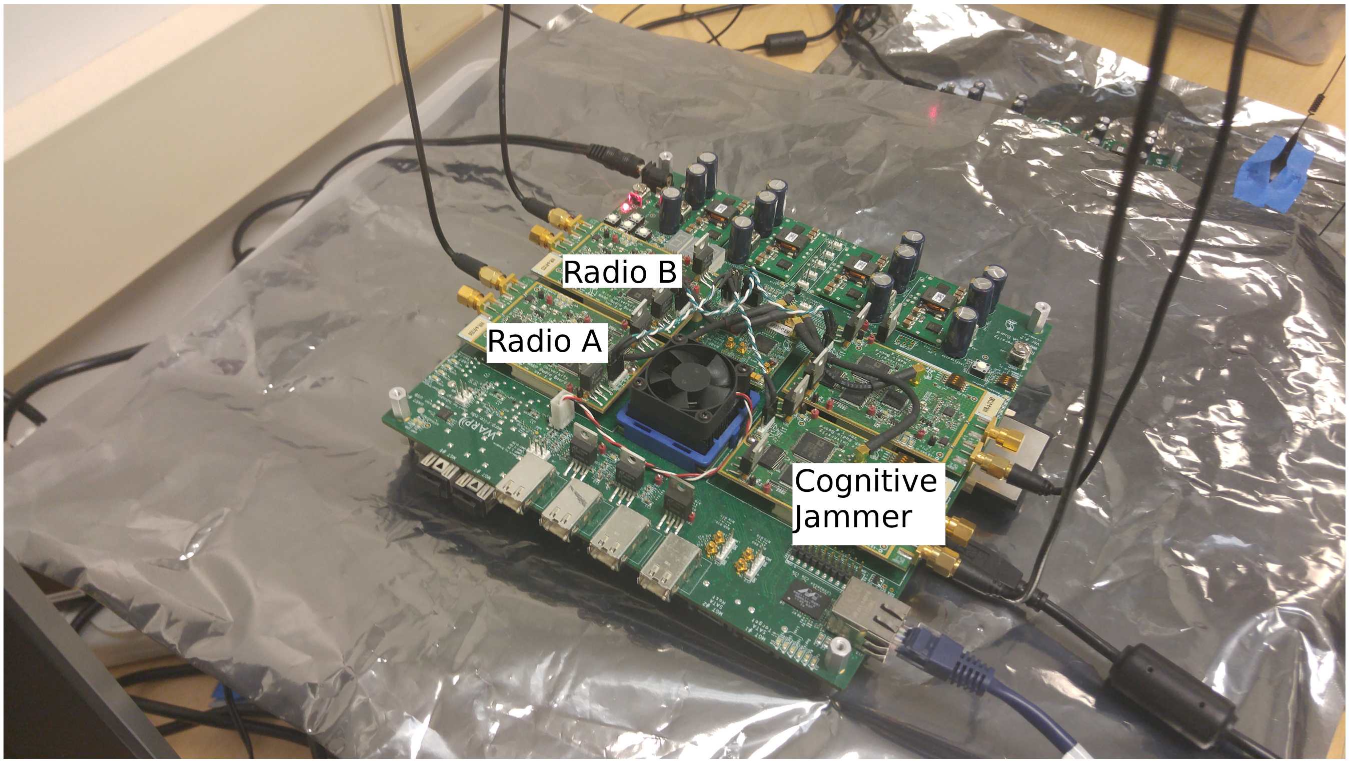

Figure 9. System configuration diagram for WARP node 1
Note: not pictured are the antennas coming from each daughtercard, the ethernet switch and the laptop, however the on-board connections are in-frame [15]

Hardware Functions and Limitations.

The code that enables the WARP boards to conduct the same functions as the simulation is a culmination of modified example code provided by the WARP project and modified code from the simulation. A primary limitation for the WARP boards is their inability to synchronize time with the current setup. Since the Matlab code is parsed to the boards, and commands are executed sequentially from the Matlab files, the boards can only complete a single action at a time. Each action the boards complete corresponds to a single line of Matlab code. The WARP code is written such that the boards will generate and physically transmit and receive signals. The commands that generate signals take longer than their simulated counterparts; however, the transmit and receive waveforms are timed appropriately such that the process is

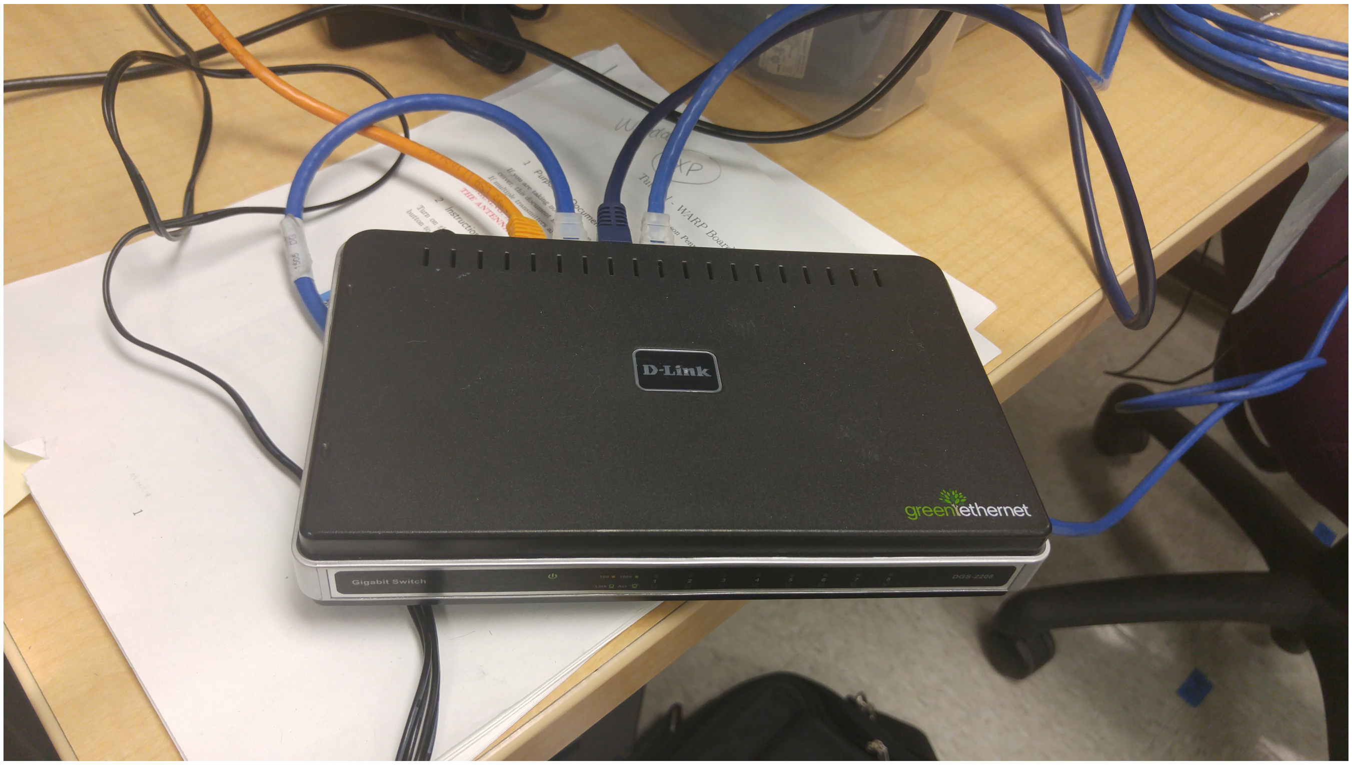

Figure 10. Ethernet switch used to parse Matlab commands

indicative of a real-time communications system.

Initializing WARP Parameters.

The parameters that need to be initialized for the WARP boards to enter the rendezvous process with one another are:

- M - commonly available channels for radios and jammer.
- P - smallest prime number greater than M.
- the initial channels for radio A, radio B, and the jammer.
- r values for Radio A, Radio B, and the jammer.
- i values for Radio A, Radio B, and the jammer.

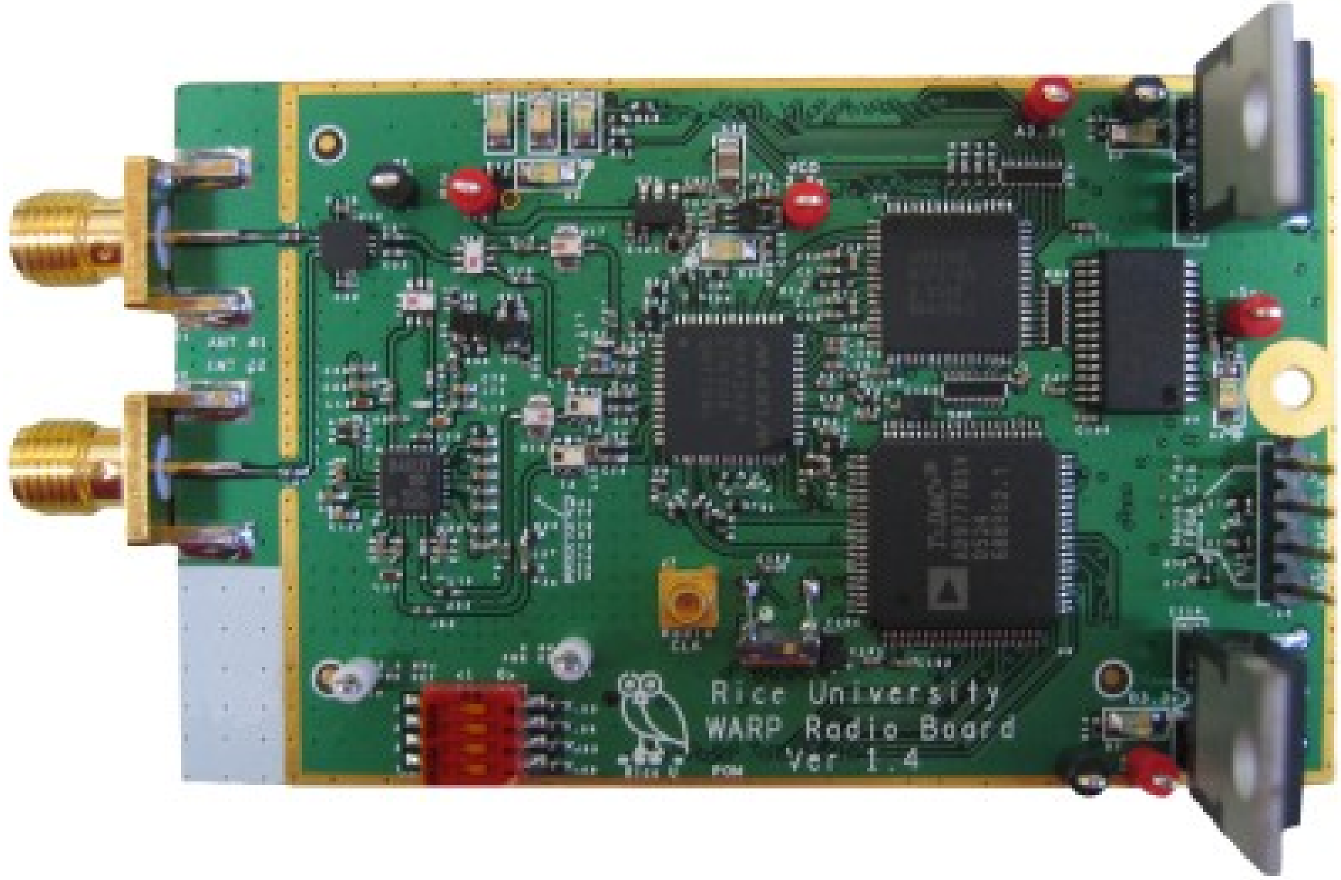

Figure 11. Daughtercard radio board [16]

The parameters for the WARP rendezvous are the same as in the simulation, and the input values are interchangeable.

Transmit and Receive Signals Using WARP.

Writing data to a board requires the creation of a vector of node objects for each board. To characterize the interactions between radios, it is sufficient to have the daughtercards on a single board act as independent radios. Because there are four daughtercards per board, a single board can represent a scenario with a two or three radio CRN and a CJ. For the appropriateness of the network to hold, the radios cannot simultaneously transmit and receive. To characterize radios that are capable of simultaneously transmitting and receiving, individual daughtercards would need

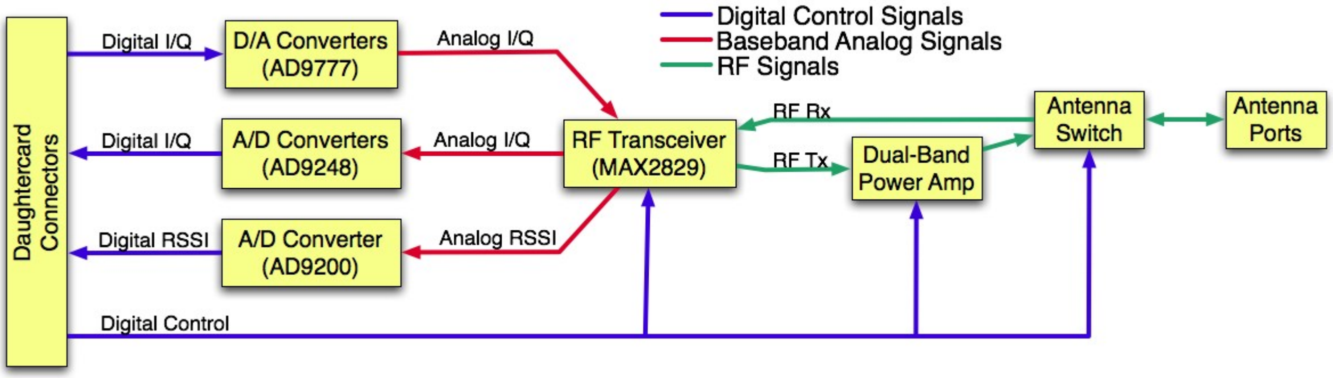

Figure 12. Daughtercard architecture [16]

to be used for each function; since there are only four daughtercards per board, the use of multiple boards would be necessary.

Once a vector of nodes is created for a given board, writing data to the daughtercards requires the retrieval of daughtercard interface identifiers from the board. Once the daughtercards are each assigned an identifier, generally denoted as "RFA, RFB, RFC, RFD" to represent the four daughtercards, writing data and signal-generating functions is as simple as altering predefined example code to reflect radio interactions between intended radios. For single input single output (SISO) radio applications, one daughtercard is designated as the transmitter while another daughtercard acts as a receiver. The SISO function written for the WARP enables the daughtercards to select one daughtercard to transmit and another to receive independently on inputted channels.

To have the WARP boards communicate with one another, the following parameters must be defined:

- NUMNODES - the WARP boards are assigned node numbers, and this input declares which board will be used. Each board has a physical display, which shows that board's node number

- ITER - the number of iterations the transmitter and receiver will generate a

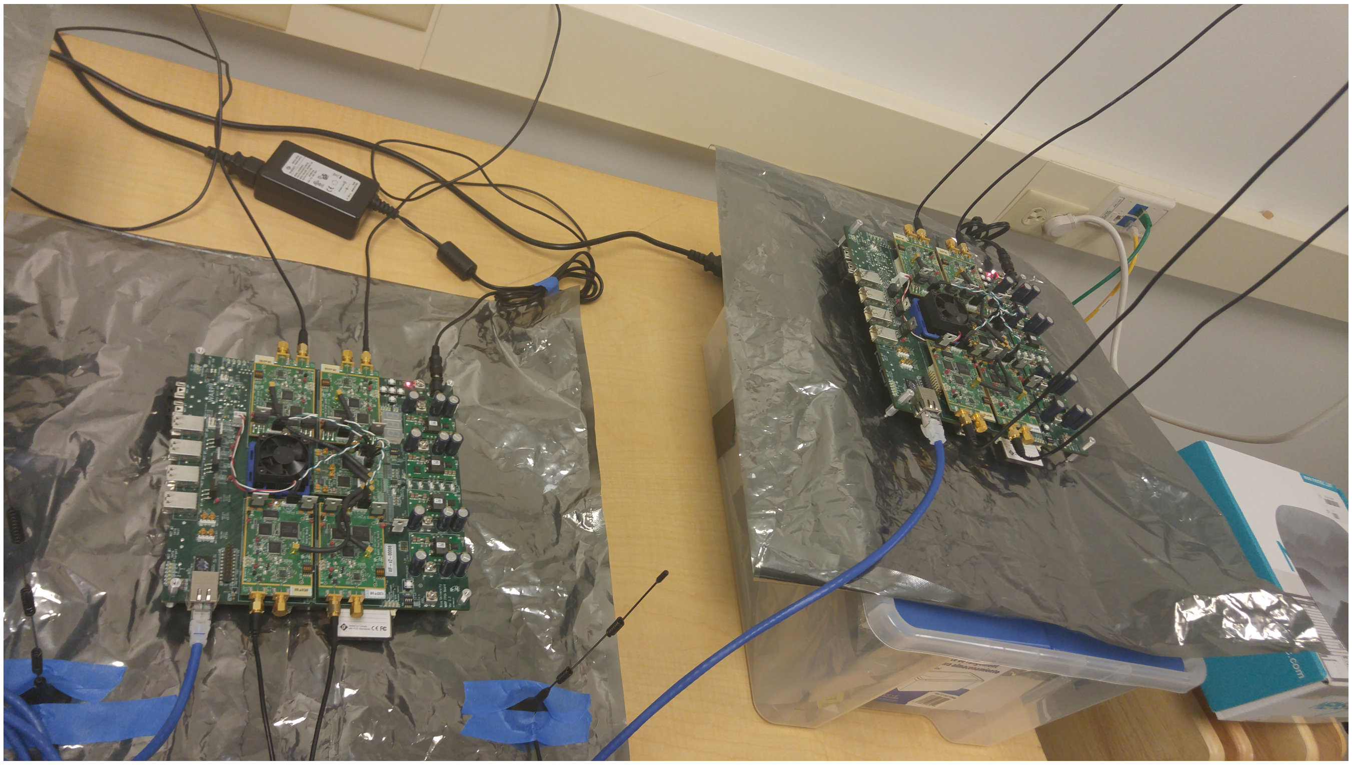

Figure 13. WARP nodes 2 **and** 3

given payload

- preamble - vector that defines the preamble. This can be the same as in the simulation
- msg - message vector. This can be the same as in the simulation
- Fs - sampling frequency
- channel A - initial channel value for radio A
- channel B - initial channel value for radio B

The payload is simply an inputted message with preamble modulated on an inputted carrier frequency, just like in the simulation. There are other modulation schemes written for the WARP boards but, for the sake of consistency with the simulation, only the BPSK option is being considered in this thesis. For the WARP

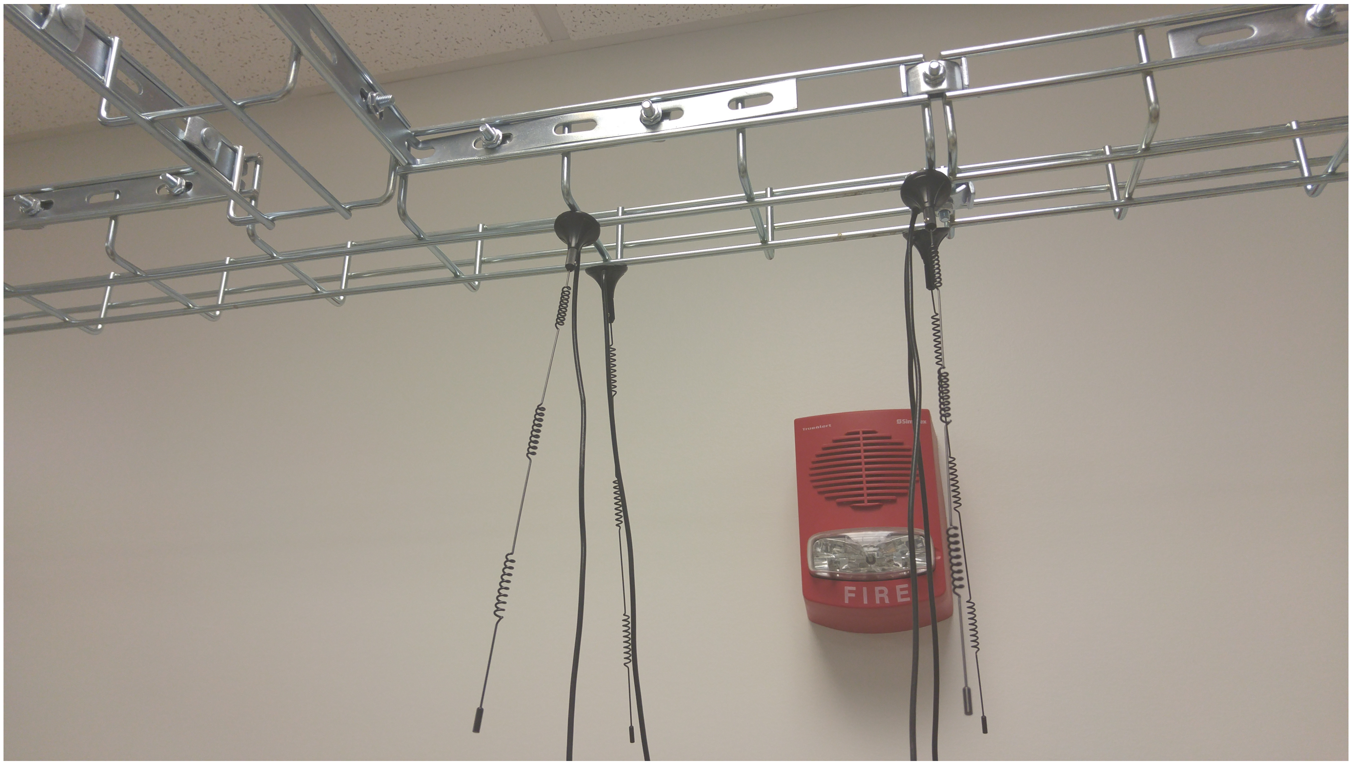

Figure 14. Antenna configuration

boards, daughtercards are selected to act as independent radios A and B, with selected channels independent and consistent to each radio.

There exists a time delay between transmit and receive functionality for the daughtercards. The same delay is present in the simulation as well. This delay is mostly caused by the computer controlling the scenario processing and generating the transmit functions before the receive functions. Fig. 15 shows the time delay between transmit and receive for the simulation, given the radios are tuned to the same channel, which is about 3.7×10^{-3} seconds. Fig. 16 shows the time delay between transmit and receive for the WARP, given the radios are tuned to the same channel, which is about 7.54×10^{-7} seconds. The time delays associated with the transmitters and receivers being tuned to different channels for the simulation and WARP are approximately 1.5×10^{-3} and 2.97×10^{-6} seconds, respectively, and are shown in Figures 17 and 18, respectively. The time delay associated with the WARP is consistently smaller

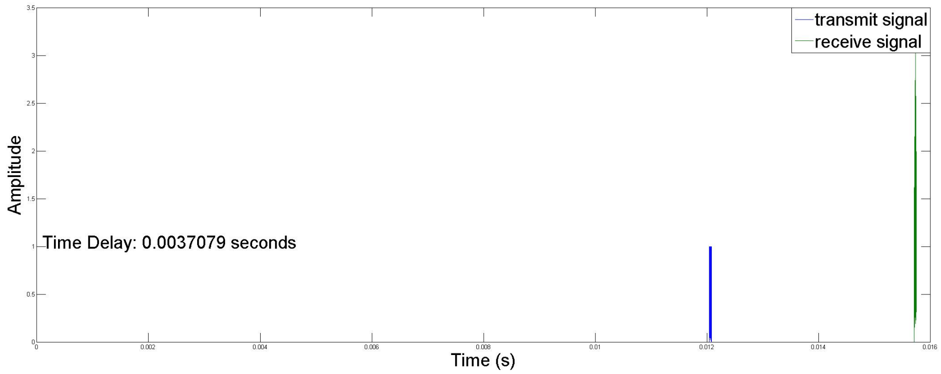

Figure 15. Time delay between transmit and receive signals on same channel using simulation

than the simulation. Once the commands for the WARP are parsed via the ethernet switch, the transmit and receive behavior is able to execute very quickly, because the WARP is not burdened by retrieving data from memory like the simulation. This memory situation occurs only when the WARP is broadcasting signals. The limiting factor for the simulation in this regard is the processor on the computer on which the simulation is being run. It should be noted that both time delays are small enough to adequately represent a physical communications system. It should also be noted that, while there is a short time delay between the transmit and receive signals for the WARP, this is only after the signals have been created and parsed to the boards, and the call to generate a single coupled transmit and receive waveform set takes considerably longer than the simulation. Preliminary testing has shown that a single time step in the rendezvous process takes approximately four times longer for the WARP to execute over the simulation.

To ensure the validity of the transmitted and received signals, the data written via Matlab were plotted as a spectrogram. The scenarios that are encapsulated by the spectrograms are transmitted and received signals when the radios are on the same channel, then again when the transmitter and receiver are not tuned to the same channel. To generate this scenario, one daughtercard acts as a transmitter,

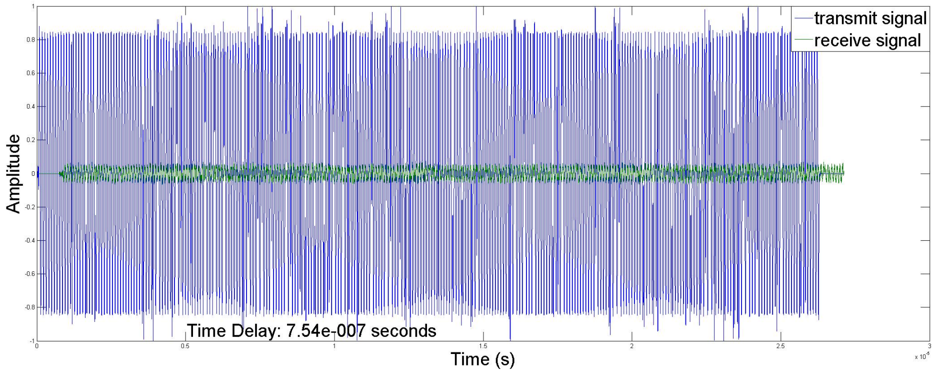

Figure 16. Time delay between transmit and receive signals on same channel using WARP

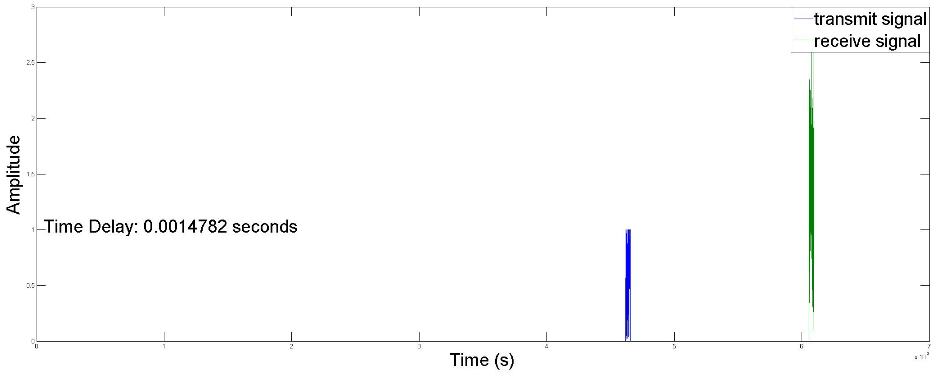

Figure 17. Time delay between transmit and receive signals on different channels using simulation

and another separate daughtercard acts as a receiver, which requires only two calls of the WARP SISO function. Unfortunately, physical measurements could not be taken. The spectrum proved to be too noisy to get measurements that are identifiable as WARP broadcasts. Multiple attempts to use a spectrum analyzer did not yield identifiable broadcast signals, with the exception of broadcasting band limited WGN utilizing four daughtercards to cover the Wi-Fi spectrum.

When the transmitter and receiver are tuned to the same channel, the spectrogram for the transmitted signal is shown in Fig. 19, and the spectrogram for the received signal is shown in Fig. 20. Note that the transmit spectrogram and the receive

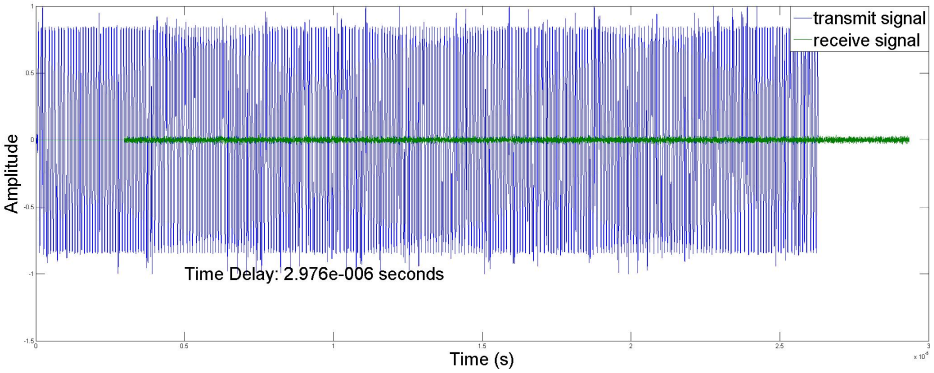

Figure 18. Time delay between transmit and receive signals on different channels using WARP

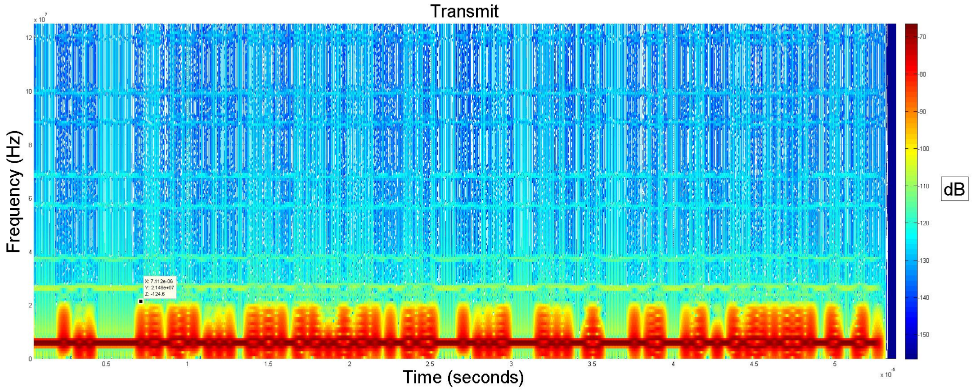

Figure 19. Spectrogram of transmitted and received signals on the same channel using WARP: Transmitted Signal

spectrogram are very similar. In Fig. 19, the base signal frequency is consistent with 2.412 GHz, which translates to the radio being set to Channel 1. And in Fig. 20, the base signal frequency matches that of the transmitted signal, since the receiver has sampled the signal, the signal in the next Nyquist zone is also shown at 5 GHz.

When the transmitter and receiver are tuned to different channels, the spectrogram for the transmitted signal is shown in Fig. 22 and the spectrogram for the received signal is shown in Fig. 21. Note that the transmit spectrogram and the receive spectrogram are very different. In Fig. 22, the base signal frequency is consistent with 2.412 GHz, which translates to the radio being set to Channel 1. And in Fig. 21,

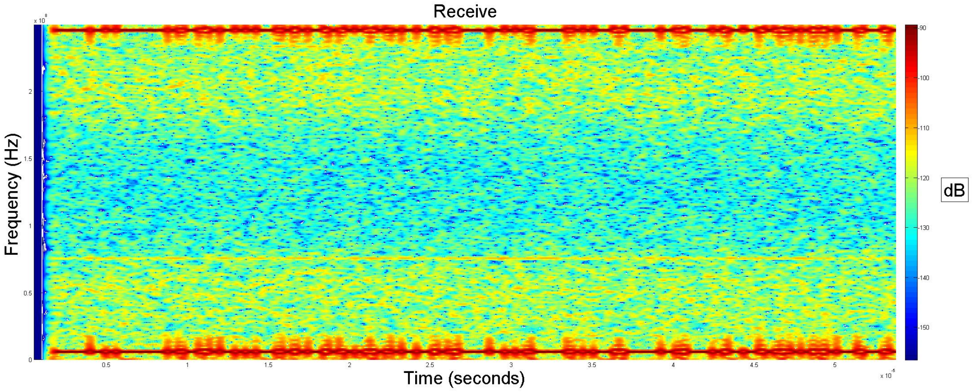

Figure 20. Spectrogram of transmitted and received signals on the same channel using WARP: Received Signal

the power level at the base frequency is about 20 dB lower than the transmitted signal. This is consistent that the receiver should have no power return, when on a different channel than the transmitter. However, since there is so much bandwidth overlap between adjacent channels, if the transmitter and receiver are on adjacent channels, the receiver will receive significant energy returns from the transmitter. Rendezvous thresholds have been set sufficiently high enough to avoid false positive rendezvous that can be caused when the radios are tuned to adjacent channels.

NOTE: These histogram plots have frequencies an order of 100 lower than the typical 2.4 GHz band to lower the overall sampling rate required. Everything else is consistent and operating frequencies mentioned above are interpolated from the known factor of 100.

Other Differences Between Simulation and WARP Code.

The rendezvous process for the WARP is the same as in the simulation, except that the transmit and receive waveforms are generated with WARP-specific code. In the rendezvous process for the WARP boards, one daughtercard will transmit while another receives — each independently tuned to a given channel. Then, the daughtercards will switch transmit and receive roles, while tuned to their same respective

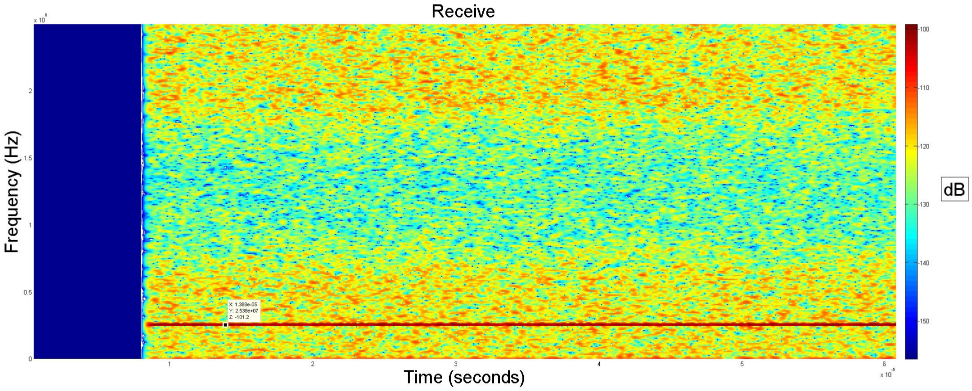

Figure 21. Spectrogram of transmitted and received signals on different channels using WARP: Received Signal

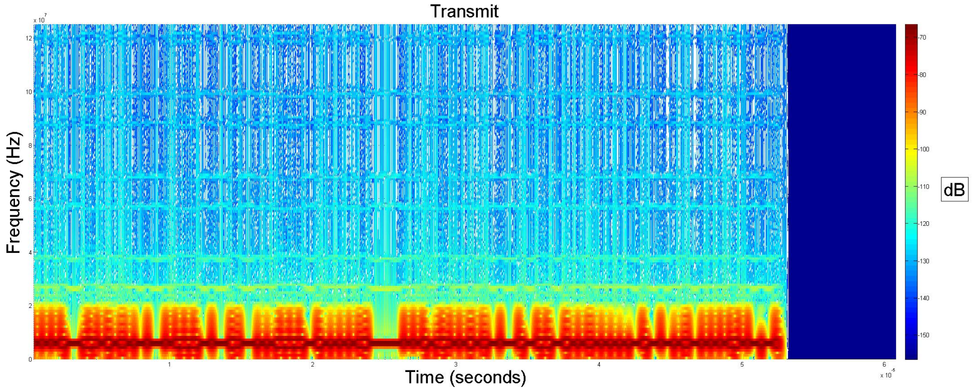

Figure 22. Spectrogram of transmitted and received signals on different channels using WARP: Transmitted Signal

channels. Otherwise, the rest of the rendezvous code is the exact same as in the simulation. This includes that the WARP boards can use both versions of the rendezvous code that the simulation does, where one version correlates preambles and the other correlates entire signals.

Checking the suitability of the communications link using WARP is using the same functionality as in the simulation, just with different threshold values to check BER and SINR that are suitable to the WARP. It is the same function using different inputs.

Jamming and Interference Using WARP.

The jammer was very easy to develop, mostly because a non-frequency-hopping version was already written by FLT LT Kate Yaxley. FLT LT Yaxley developed a script for the WARP that would noise jam a remote controlled "I-Spy Tank." Since the tank always broadcasts control commands on the same channel, jamming it was rather easy. The "Tank Jammer" file is a modified SISO WARP example file, much like what is used to transmit and receive waveforms described in Subsection 3.2. Next, a function wrapper was written to allow for input of which channel the jammer would tune to jam. Making the jamming broadcasts a function, like the transmitter, allows for all the channel hopping controls of the jammer to be controlled externally the same way as the transmitters and receivers.

Additionally, the "Tank Jammer" file was expanded to allow all four daughtercards to transmit on simultaneous frequencies to create a separate wideband jammer, which covers all the frequencies that are allowed for commercial use. Alternatively, the wideband jammer can tune to the non-overlapping channels 1, 6, 11, and 14, in the Wi-Fi spectrum to cover most of the 2.4 GHz spectrum. This wideband jamming implementation allows the jammer to have another mode when desired.

The CJ's spectrum-sensing capabilities are based on the same example WARP SISO file as all the other functionality; however, instead of only using the transmitting portion, the CJ only uses the receive function. There is logic in the code that makes the CJ continue jamming on the same frequency, if the radios have not moved after jamming them. Once the CJ cannot sense the radios on the previous channel, the CJ will begin to search the spectrum for the radios. Once the radios are found, the jammer will resume jamming the radios. In summary, the jammer must find the radios, jam them, check if the radios have tuned away from the channel being jammed, then restart its search for the radios.

Jamming Strategies for WARP.

The jamming strategies and methods for evaluation are the same for the WARP with the exception of the estimation algorithms. Using WARP hardware, there is not sufficient bandwidth for the daugthercards to properly observe the spectrum as required to attempt estimate calculations as described in section 3.1.

IV. Results and Analysis

This chapter describes results from simulations and hardware verification.

4.1 Simulation Results

A summary of tests and analysis conducted using simulated data is presented in this section.

Baseline testing of the correlator used to accomplish rendezvous was accomplished first. The correlator output was measured without jamming, in an ideal-high SINR environment, to ascertain what degradation is possible for the jammer to accomplish. Next,the jammer was optimized for maximum correlator output degradation. Then, the effects on the correlator due to optimized jammer were analyzed.

A CR was then observed at a network level interacting with a jammer. The radio and jammer were simply generating channel states, and no waveforms were generated. Finally, the effects of optimized jammer on the rendezvous process, utilizing the various jamming strategies, was analyzed at the waveform level using two radios and a jammer.

Designing the Jamming Waveform - Simulation.

The jammer used a set of operating points from a set of test points to maximum PMR degradation. The two parameters considered for optimization of the jammer were the bandwidth and power for a noise-jammer. The jammer is assumed to be capable of frequency agility, like the other CRs, but only broadcasts noise. The bandwidth parameter assumes band-limited noise centered on the frequency associated with a given channel. The power parameter is defined as decibels relative to one Watt. The values swept for power and bandwidth were -10 to 3 dBW in 1 dBW

Table 4. Jamming Waveform Parameters for Simulation

	Simulation - Preamble	Simulation - TxRx
Jammer Power	3 dBW	3 dBW
Jammer Bandwidth	18.5 MHz	18.5 MHz
Resulting PMR	14.67	36.57
Baseline PMR	16.68	39.72

increments and 1 to 20 MHz in 100 kHz increments, respectively.

Table 4 provides a summary of the jamming parameters used and the resulting PMRs and compares them to baseline PMRs, which includes no jamming energy in the system. It is intuitive that a noise jammer would be more effective as a function of higher bandwidth and power.

Correlator Effects due to Jamming Waveform.

The correlator outputs were measured again with jamming signals generated utilizing the optimized parameter sets. The effects on the correlator due to the optimized jammer were analyzed.

Fig. 23 shows the degraded output for the preamble-based correlator due to the optimized jammer. Fig. 24 shows the degraded output for the transmit/receive-based correlator due to the optimized jammer. Note that since there is more signal energy present in the correlator, the mean and peak output values for both correlators are higher. However, the PMR for the preamble-based correlator is significantly reduced, as evident in Fig. 23. The correlator which relies on a shorter signal is considerably more affected by jamming and interference.

Observing Cognitive Radios and Jamming Strategies.

A CR was then observed at a network level interacting with a jammer. The radio and jammer were simply generating channel states, and no waveforms were generated.

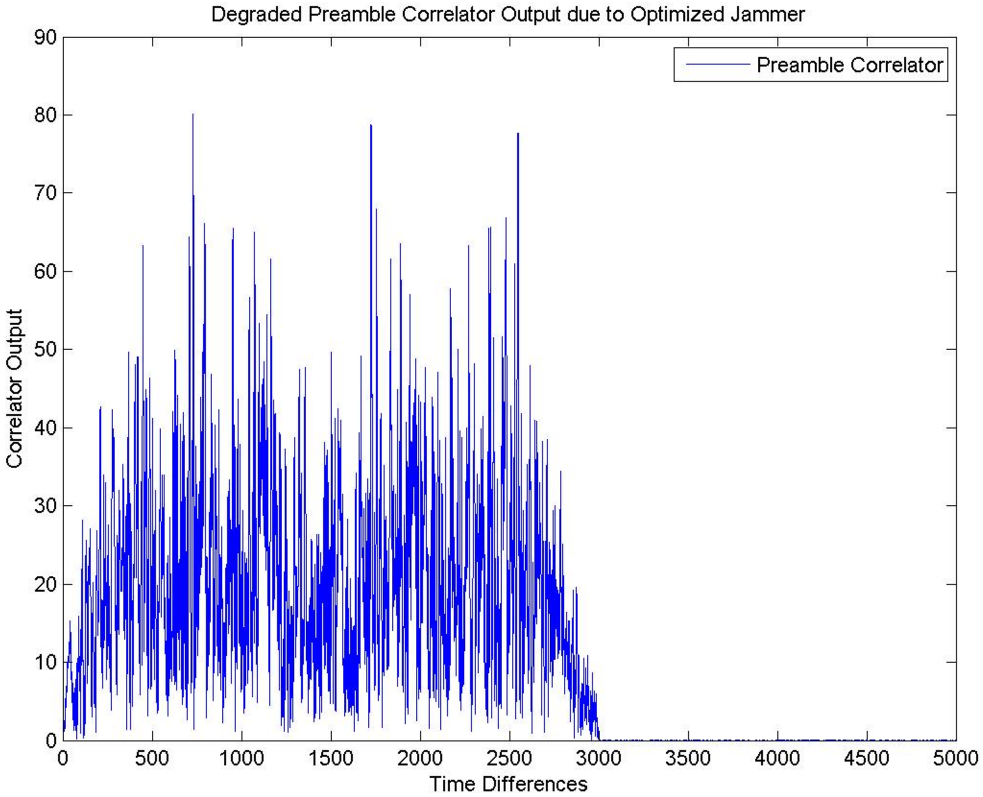

Figure 23. Degraded preamble-based correlator output due to optimized jammer

The jammer interacted with the radio according to four jamming strategies. The first three jamming strategies assume that the Jump-Stay channel-hopping logic is known a-priori.

Fig. 25 shows the effectiveness of a jammer hopping channels based on observed channel state data from the radio. It was assumed that the jammer could observe all of the channels available to the radio. With a channel state history for the radio, the jammer attempted to estimate the Jump-Stay parameters the radio was using, then intercept the radio using the Jump-Stay logic with estimated parameters. As seen in Fig. 25, some combinations of step size (r) and index (i) values are easier to observe than others. This is due to the modulo math that is the basis of the Jump-Stay algorithm, so these combinations make it easy to observe the "wrap-around" effect

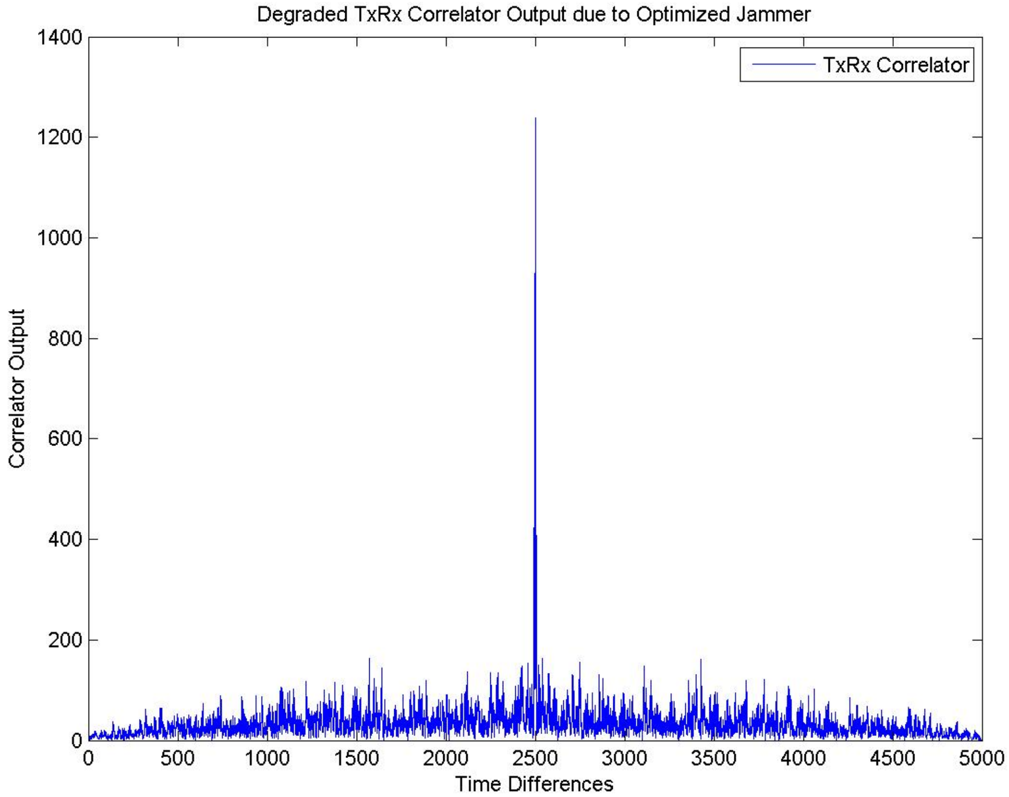

Figure 24. Degraded transmit/receive-based correlator output due to optimized jammer

that occurs when a channel-hop results in a channel states less than prior channel states. In order to accurately estimate Jump-Stay parameters, the jammer must observe this "wrap-around" effect. Therefore, the r and i combinations in red should be avoided by radios to avoid a CJ attempting to estimate the radio's Jump-Stay parameters. If the radio is using r and i values corresponding to the red areas of Fig. 25, the jammer can be expected to follow the radio upwards of 95% of the time, given favorable SINR conditions.

Fig. 26 shows the effectiveness of a jammer hopping channels based on observed channel state data from the radio. It was assumed that the jammer could observe half of the channels available to the radio. With a channel state history for the radio, the jammer attempted to estimate the Jump-Stay parameters the radio was using,

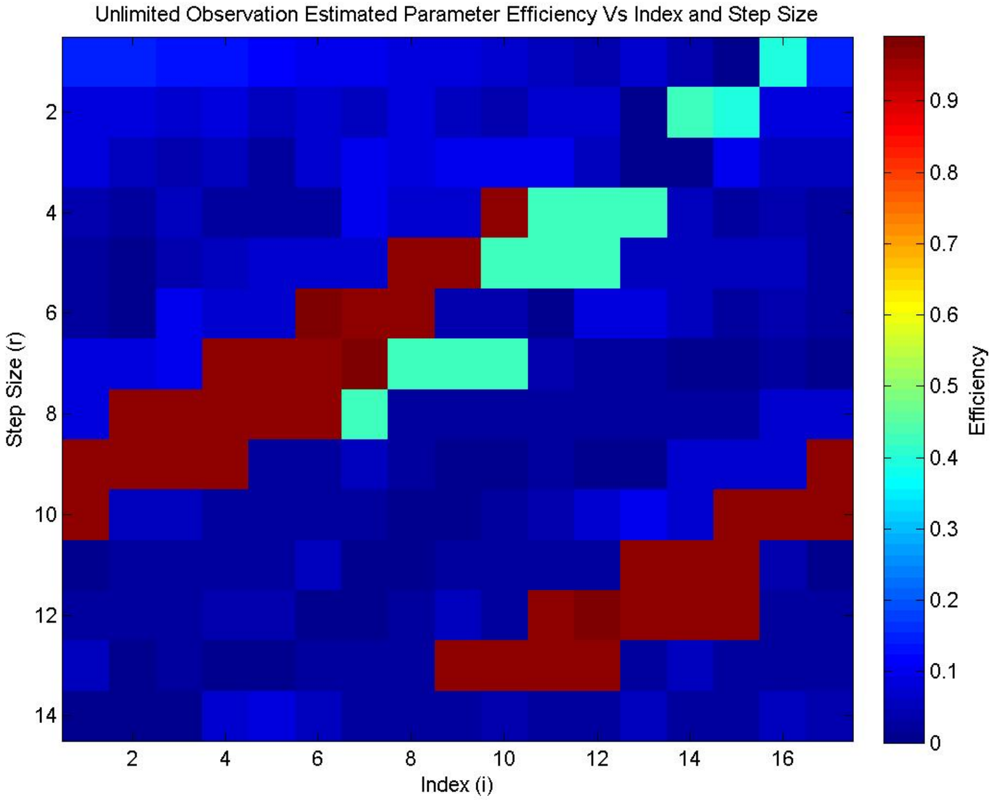

Figure 25. Unlimited spectrum observation parameter estimation effectiveness

then intercept the radio using the Jump-Stay logic with estimated parameters. As seen in Fig. 26, the jammer had limited success estimating the radio's Jump-Stay parameters. This is because the jammer is largely unable to observe the "wrap-around" effect necessary to accurately estimate the Jump-Stay parameters. Even when the whole spectrum is considered observable, it is difficult for the jammer to accurately estimate the Jump-Stay parameters. Observing half of the channel-space, the jammer cannot be expected to follow a radio more than 25% of the time, even under favorable SINR conditions.

Table 5 summarizes step size and index value combinations to avoid that are prone to higher risk of observation-based estimation.

Fig. 27 shows the effectiveness of a jammer hopping channels based on randomized

Table 5. Step size and index combinations to avoid

Step Size, r	**Index,** i
1	16
2	$14 - 15$
3	NA
4	$10, 11 - 13$
5	$8 - 9, 10 - 11$
6	$5 - 7, 8 - 11$
7	$4 - 7, 8 - 10$
8	$2 - 6, 7$
9	$1 - 5, 17$
10	$1, 15 - 17$
11	$13 - 15$
12	$11 - 15$
13	$9 - 12$
14	NA

step size and index Jump-Stay inputs. It was assumed that the number of channels available to the radio was known. The jammer attempts to intercept the radio leveraging the strength of the Jump-Stay convergence. As seen in Fig. 27, the jammer had limited success converging with the radio. However, under favorable SINR conditions, the jammer can be expected to converge with the radio within P time steps due to the properties of the Jump-Stay channel-hopping algorithm.

Fig. 28 shows the effectiveness of a jammer repeating observed channel state data from the radio. It was assumed that the jammer could observe all of the channels available to the radio and the jammer observed P channel-hops. Recall that there are P channel-hops in every jump pattern, which each are presumed to have a duration of one time-step. With a channel state history for the radio, the jammer attempted to rebroadcast observed channel state data with appropriate timing delays to synchronize with the radio's channel-hopping pattern. Given that the "jump" portion of the pattern repeats, the jammer can at best be $\frac{2}{3}$ effective. As seen in Fig. 28, depending on how accurate the implemented timing delays are, the jammer can be 60 to 67%

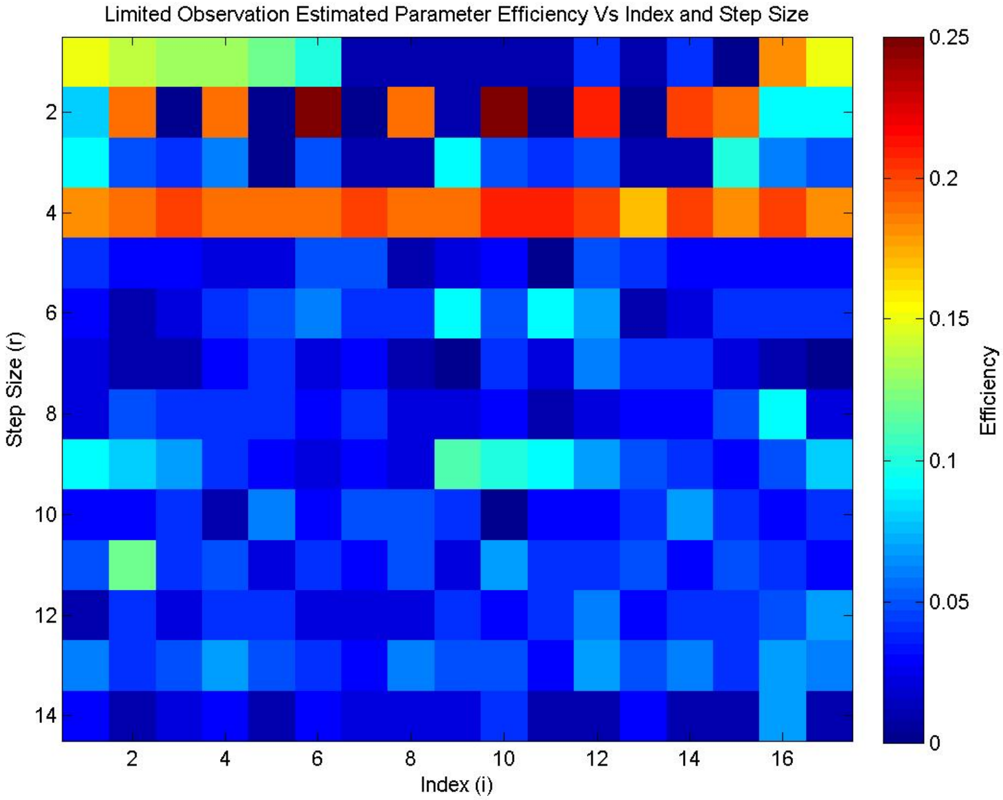

Figure 26. Half spectrum observation parameter estimation effectiveness

accurate.

Effects of Jamming Strategies on Rendezvous.

The effects of the optimized jammer on the rendezvous process, utilizing the various jamming strategies, was analyzed at the waveform level using two radios and a jammer.

Fig. 29 shows the time steps required for rendezvous for the preamble-based correlator under duress of various jamming strategies. The jamming strategies presented in Fig. 29 are the repeater-based jammer, the randomized Jump-Stay inputs, unlimited observation Jump-Stay parameter estimates, limited observation Jump-Stay parameter estimates, and the jammer randomly hopping channels at every time step.

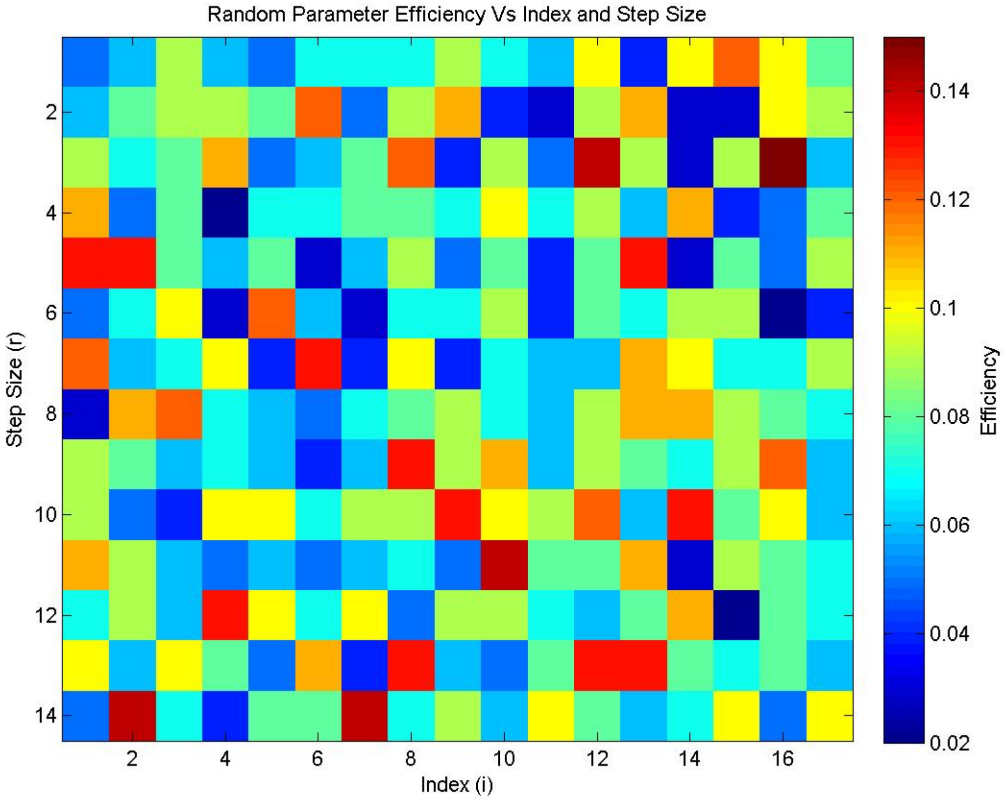

Figure 27. Randomized Jump-Stay input effectiveness

For control scenarios, the best case of no jamming, and the worst case of the jammer always being on the same channel as Radio A, were also considered. In the estimated data sets, the number of channels available to Radios A and B is assumed to be unknown, but the jammer defaults to initial guesses in the time steps before estimates can be computed. However, the Jump-Stay logic is assumed to be known.

As seen in Fig. 29, attempting to estimate the Jump-Stay parameters based on observed channel state data was ineffective. There are several reasons that estimating the parameters isn't very effective. First, the jammer has to positively identify the radio and accurately estimate the radio's current channel based on its center frequency just to build channel state history data. Second, assuming the channel state history data is accurate, the jammer has to observe the "wrap around" effect men-

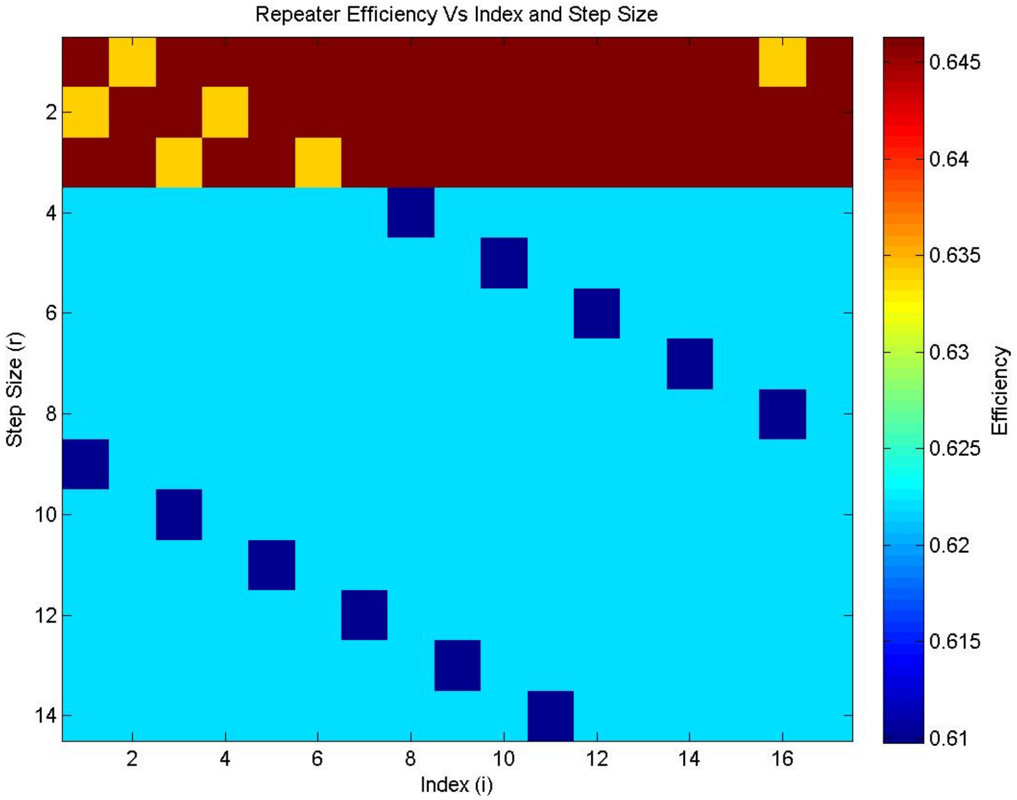

Figure 28. Repeater effectiveness

tioned earlier to begin to accurately estimate Jump-Stay step lengths. Third, even if the Jump-Stay step length is estimated correctly, the jammer may estimate another parameter incorrectly, which would flaw the rest of the estimates. The jammer could even estimate a parameter so incorrectly that it would not be in an allowable set of integers for Jump-Stay, which would cause the jammer to attempt to correct the estimates such that Jump-Stay can be performed. The "corrections" could wrong, but without them, Jump-Stay couldn't be performed. It was discovered during development and trouble-shooting of the parameter estimation algorithm that there are many such possible situations that the jammer can "over-correct" based on non-allowable estimated parameters. Fourth, even if all the Jump-Stay parameters were estimated correctly, the time delay needed to accurately intercept the channel-hopping pattern the radio of interest is using in real time is difficult to compute accurately. Even if everything lines up perfectly, the jamming energy may not delay rendezvous anyhow.

Fig. 29 also shows that random Jump-Stay inputs are almost as effective as the ideal situation where the jammer is constantly on the same channel as one of the radios. The jammer randomly hopping channels delays rendezvous sometimes, and may be a good option if the channel-hopping logic utilized is unknown to the jammer. Additionally, the repeater is not effective at all because by the time it is ready to broadcast energy, the radios have typically already rendezvoused.

There are a lot of weaknesses in the process of parameter estimation as it currently stands. Additionally, the performance of the jammer always being on the same channel as one of the radios isn't much better than using random Jump-Stay inputs. Therefore the efforts involved in refining the parameter estimation algorithm until it has ideal performance would be largely wasted. However, in other implementations where other channel-hopping algorithms are utilized, further improvement on parameter estimation may be of great benefit.

Fig. 30 shows the time steps required for rendezvous for the transmit/receive-based correlator under duress of the same jamming strategies presented in the preamble-based correlator case. Since the correlated sequence utilized in the transmit/receive-based correlator is five times longer than that of the preamble-based correlator, the rendezvous times remain undisturbed by any jamming efforts. It is expected that the longer the correlated sequence is, the more resilient the system will be to jamming. In the case of the transmit/receive-based correlator, utilizing the entire signal length for spectrum sensing in the rendezvous process enabled the radio rendezvous times to not increase and there to not be any rendezvous declared on incorrect channels. Table 6 summarizes the effects of jamming on rendezvous times for the simulation. Note that since the transmit/receive-based correlation method had no rendezvous delays,

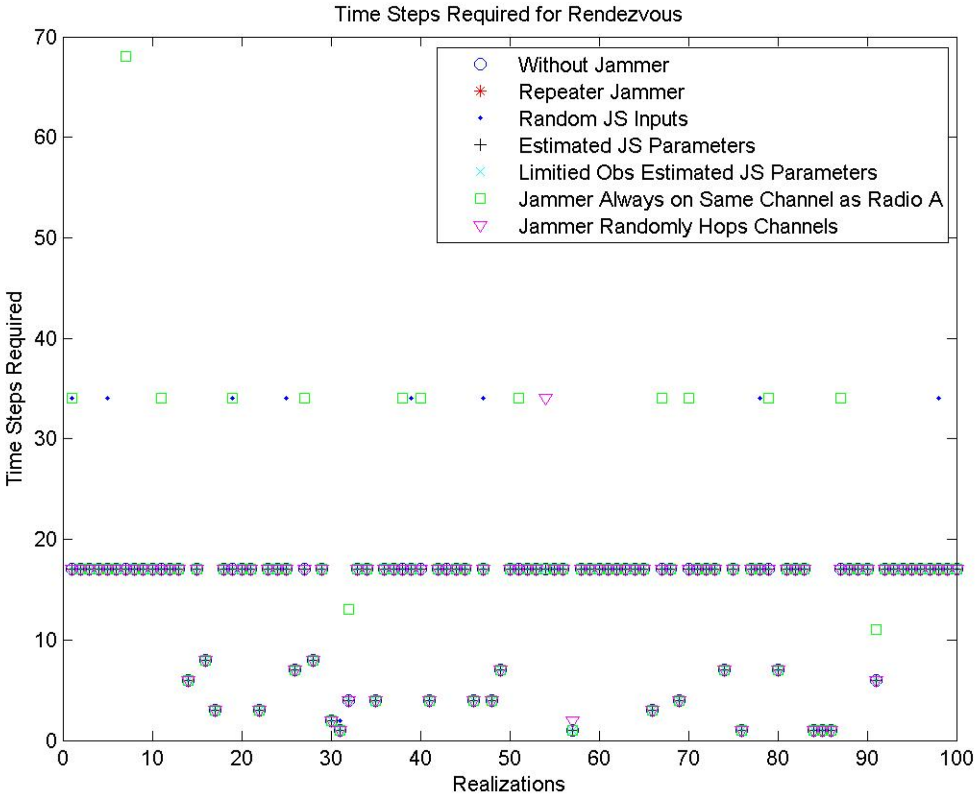

Figure 29. Time steps required for rendezvous with preamble-based correlator

information in Table 6 only covers the preamble-based correlation method. When jamming strategies did work, the maximum time to rendezvous (MTTR) doubled with respect to the MTTR when no jamming was present.

Channel states at rendezvous were recorded for each combination of jamming strategy, preamble-base correlator, and transmit/receive-based correlator. Each plot is similar to Fig. 31, but the more effective the jammer was at following Radio A, the more often the jammer, Radio A, and Radio B were on the same channel at rendezvous. Fig. 31 shows the channel states at rendezvous for full channel-space observation for estimating Jump-Stay parameters for the preamble-based correlator.

Table 6. Time steps exceeded by jamming strategy

Jamming Strategy	% delays caused	MTTR
No Jamming	0%	17
Repeater	0%	17
Random JS Inputs	9%	68
JS full estimation	0%	17
JS partial estimation	0%	17
Always on Radio A	13%	68
Random CH	1%	34

Observations and Commentary.

Overall, jamming the preamble-based correlator and the transmit/receive-based correlator did not cause either system to rendezvous on the incorrect channel. Delays in time to rendezvous were caused in the preamble-based correlator but not the transmit/receive-based correlator.

Analysis of the jamming strategy potential can determine what step size and index values to *not* use on part of the radios avoiding a jammer.

The repeater strategy is a simple, easily implemented, and potentially effective approach, but due to the robustness of the Jump-Stay algorithm, the radios will typically converge in P time steps. So, by the time the repeater has observed enough channel state data to be effective, the radios have probably already converged.

Generally, the problem of the jammer intercepting the radios is difficult. There is a small window of time that the jammer has to observe the radios before they converge. The solution that the jammer can estimate in that short time is not always a very robust one. If the jammer can observe all the channels simultaneously, it can make a much better estimate of the Jump-Stay parameters than if its observation space is limited. However, even when the jammer can make accurate estimates in short amounts of time, the robustness of the radio's correlator can nullify jamming attempts. Estimating parameters for the Jump-Stay algorithm required that obser-

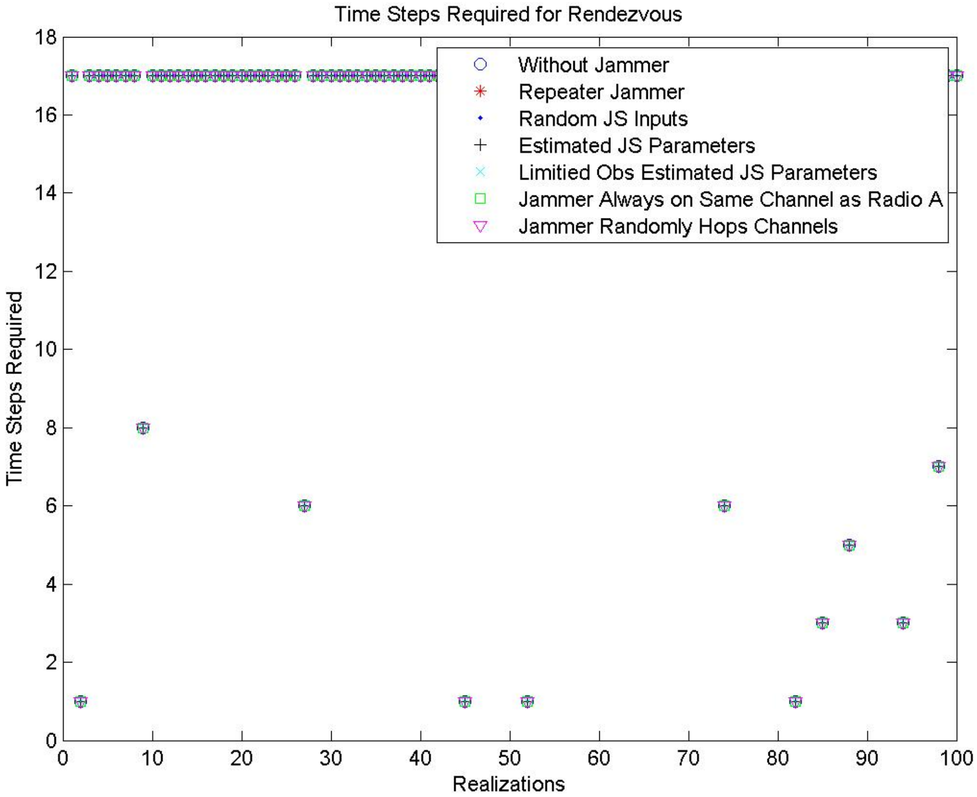

Figure 30. Time steps required for rendezvous with transmit/receive-based correlator

vations be made that could catch the "wrap-around" effect.

Correlators need to be able to make internal, dynamic calculations about what the threshold for detection should be based on information about the signals of interest. Time delays must similarly be internally calculated to enable real-time synchronization of a jammer targeting a radio. Without proper time delays, a jammer cannot intercept a radio when attempting to predict its channel-hopping pattern, even when the channel-hopping parameters are calculated correctly.

In a high SINR setting with appropriate correlator thresholding, rendezvous times via Jump-Stay matches published data at the waveform level as well as the network level. The lower the SINR, the worse the time to rendezvous performance will presumably be. The effectiveness of the correlator is the cornerstone to the effectiveness

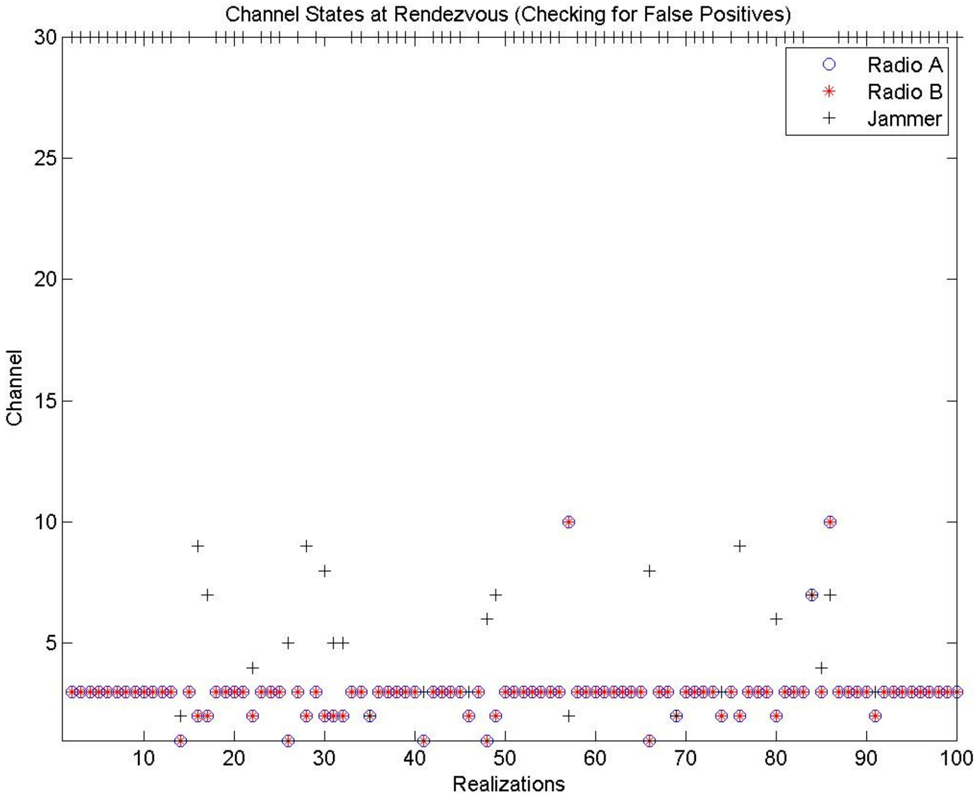

Figure 31. Channels states at rendezvous for full observation estimated Jump-Stay parameters for preamble-based correlator

of the rendezvous process. Longer correlation signals used in a system yielded better rendezvous performance under jamming conditions.

4.2 Hardware Verification

Baseline testing of the correlator used to accomplish rendezvous was accomplished first. The correlator output was measured without jamming, in the most favorable SINR environment available, to ascertain what degradation is possible for the jammer to accomplish. Next,the jammer was optimized for maximum correlator output degradation. Then, the effects on the correlator due to optimized jammer were analyzed.

Jamming strategies were not analyzed at a network level using WARP. However, the effects of optimized jammer on the rendezvous process, utilizing the various jamming strategies, was analyzed at the waveform level using two radios and a jammer. With the WARP, three daughtercards were used to act as the radios and jammer, allowing all testing to be done on a single WARP board.

Designing the Jamming Waveform - WARP.

The jammer used a set of operating points from a set of test points to maximum PMR degradation. The two parameters considered for optimization of the jammer were the bandwidth and power for a noise-jammer. The jammer is assumed to be capable of frequency agility, like the other CRs, but only broadcasts noise. The bandwidth parameter assumes band-limited noise centered on the frequency associated with a given channel. The power parameter is defined as decibels relative to one Watt. The values swept for power and bandwidth were -10 to 3 dBW in 1 dBW increments and 1 to 20 MHz in 100 kHz increments, respectively.

Table 7 provides a summary of the jamming parameters used and the resulting PMRs and compares them to baseline PMRs, which includes no jamming energy in the system. It is intuitive that a noise jammer would be more effective as a function of higher bandwidth and power. The lower the SINR, the less power is required,

Table 7. Jamming Waveform Parameters for WARP

	WARP - Preamble	**WARP - TxRx**
Jammer Power	-10 dBW	-2 dBW
Jammer Bandwidth	5 MHz	20 MHz
Resulting PMR	11.11	8.60
Baseline PMR	7.88	54.67

but the more advantageous a spread spectrum (wider bandwidth) approach becomes. Note that the correlator outputs were so low in the Preamble WARP case that radios were never able to rendezvous, so the decreasing PMR trend seen in all the other situations is not replicated.

Correlator Effects due to Optimized Jammer using WARP.

The correlator outputs were measured again with jamming signals generated utilizing the optimized parameter sets. The effects on the correlator due to the optimized jammer were analyzed.

Fig. 32 shows the degraded output for the preamble-based correlator due to the optimized jammer using WARP. Note that since there is more signal energy present in the correlator, the mean and peak output values for the preamble-based correlator are higher. However, though the PMR for the preamble-based correlator has increased, the insufficiently short preamble makes spectrum sensing via matched filtering incredibly difficult.

Fig. 33 shows the degraded output for the transmit/receive-based correlator due to the optimized jammer using WARP. The peak value for the transmit/receive-based correlator is about the same, but the mean value is much higher. The increase in mean value is what has degraded the PMR so effectively. The transmit/receive-based correlator remains susceptible to jamming and interference.

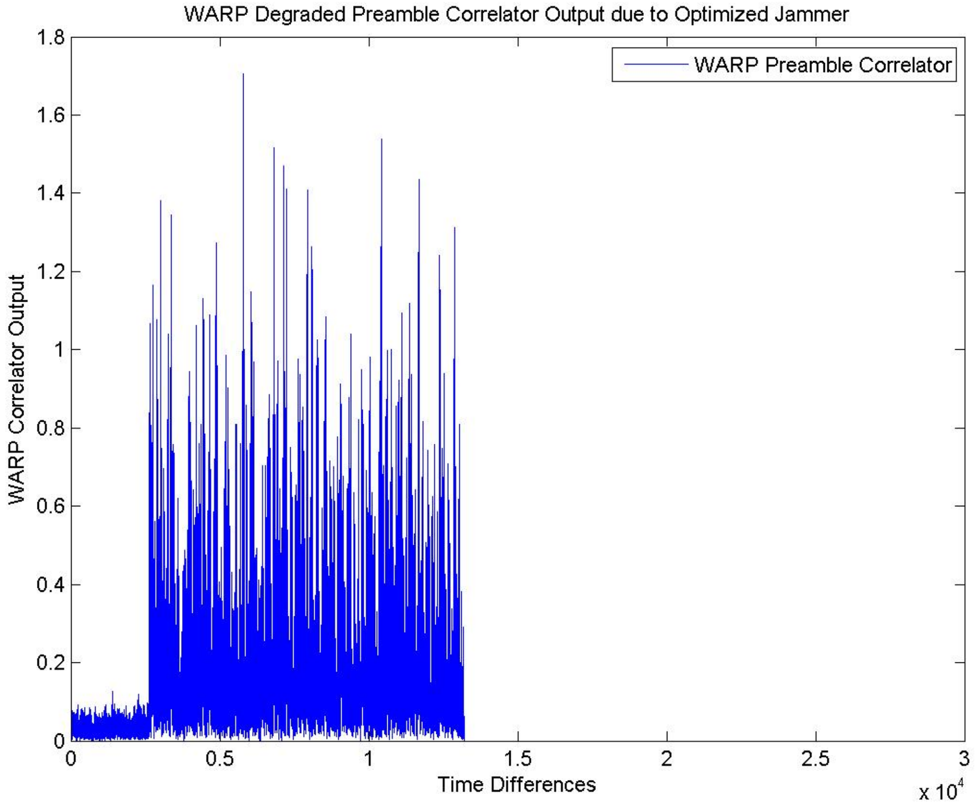

Figure 32. Degraded preamble-based correlator output due to optimized jammer using WARP

Effects of Optimized Jamming Strategies on Rendezvous using WARP.

The effects of the optimized jammer on the rendezvous process, utilizing the various jamming strategies, was analyzed at the waveform level using two radios and a jammer. Three daughtercards on a single WARP board were used to act as the two radios and one jammer.

Fig. 34 shows the time steps required for rendezvous for the preamble-based correlator under duress of various jamming strategies using WARP. The jamming strategies presented in Fig. 34 are the randomized Jump-Stay inputs and the jammer randomly hopping channels at every time step. For control scenarios, the best case of no jamming, and the worst case of the jammer always being on the same channel as Radio

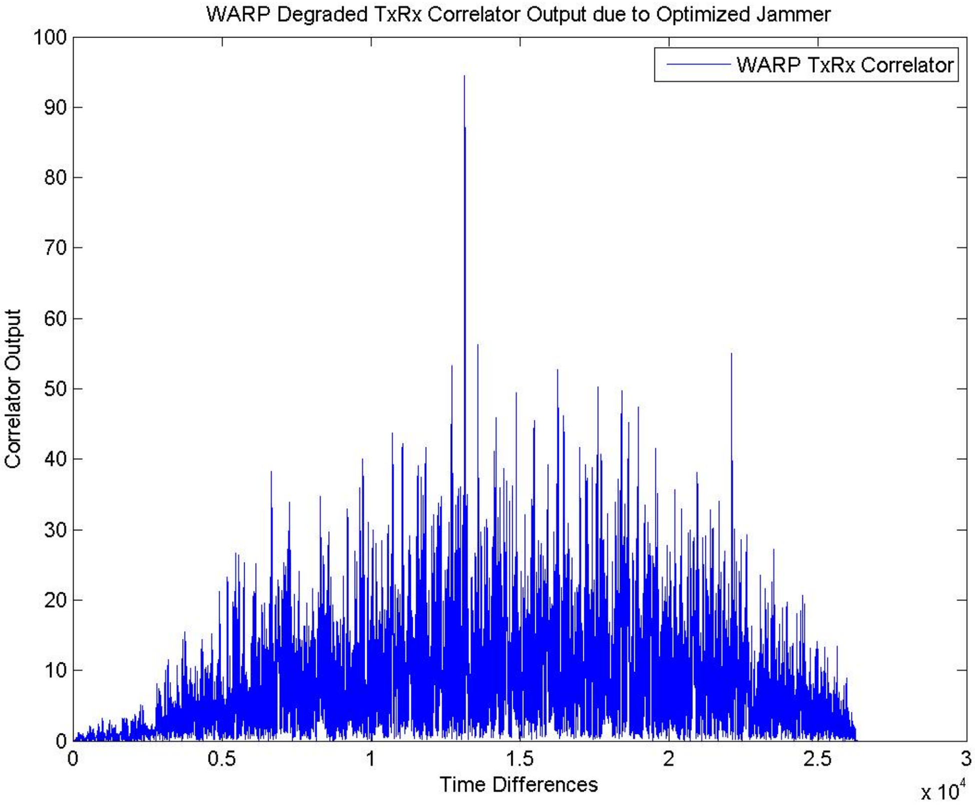

Figure 33. Degraded transmit/receive-based correlator output due to optimized jammer using WARP

A, were also considered. The Jump-Stay logic is assumed to be known.

Fig. 34 shows that the 4 bit preamble was too short to perform successful matched filtering. This is evident because all scenairos the radios were attempting to rendezvous in reached the 200 time step "timeout". In a noise-sterile environment, the radios are expected to rendezvous in 17 time steps given the rest of the conditions remain the same. It is possible that the radios could rendezvous in some astronomically higher number of time steps, but given time and resource constraints and the marginal gain of knowledge such results could provide, the timeout condition value was not raised. Also, previous attempts utilized higher timeout condition values and greater numbers of realizations, and after about 48 hours of continuously running, the WARP board crashed and had to be restarted. Therefore, it is still possible that

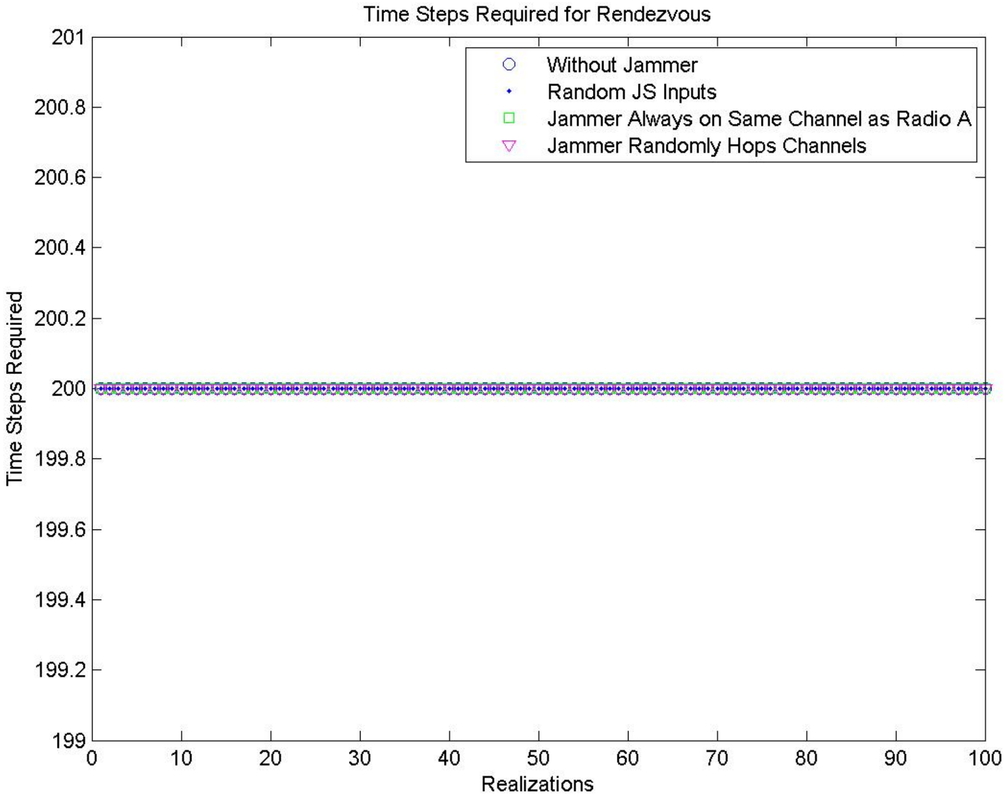

Figure 34. Time steps required for rendezvous with preamble-based correlator using WARP

the WARP can't run long enough to generate more ideal data.

Fig. 35 shows the time steps required for rendezvous for the transmit/receive-based correlator under duress of the same jamming strategies presented in the preamble-based correlator case. Since the correlated sequence utilized in the transmit/receive-based correlator is five times longer than that of the preamble-based correlator, the rendezvous times do not always reach the timeout condition. The timeout condition is still met with relative frequency in all cases due to the greater noise energy in the environment and relatively short correlation signal of 20 bits. It is expected that the longer the correlated sequence is, the more resilient the system will be to jamming, but on WARP hardware, it is not possible to insert a longer sequence. If the timeout condition and number of realizations were raised, there would be a greater

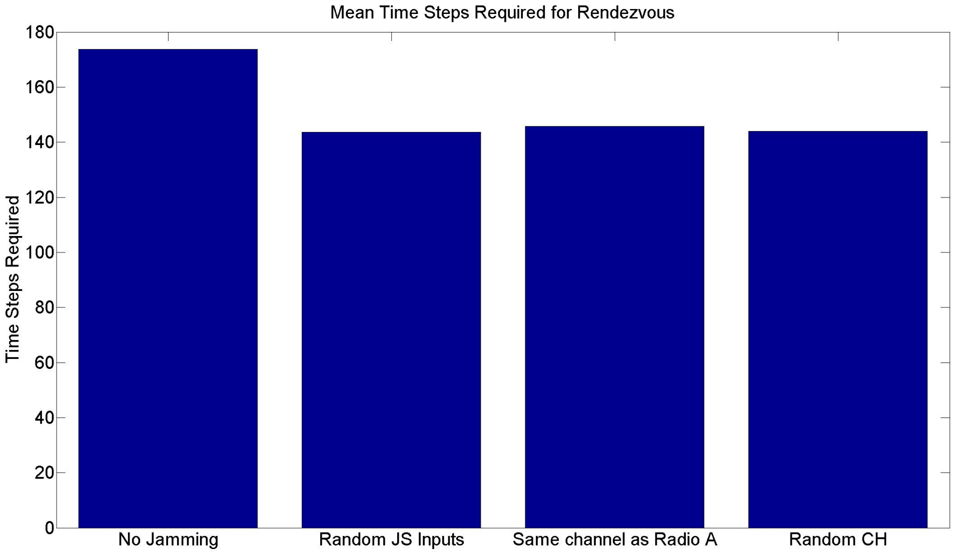

Figure 35. Mean Time steps required for rendezvous with transmit/receive-based correlator using WARP

spread of time steps required to rendezvous across the various jamming strategies. However, the aforementioned runtime limitations were respected and usable data was still produced. What was unexpected is that the CRs rendezvoused more quickly with jamming energy present than under no jamming energy. This is perhaps due there being more energy in the system when jamming is present. With the runtime limitations, it is difficult to ascertain whether or not this behavior is a indicative of a steady-state scenario or not.

Overall, the randomly hopping jammer and the jammer that always followed Radio A seemed to perform the best. Given the constraints on timeout and realizations, it is difficult to accurately evaluate performance. Perhaps future work could run similar tests on more robust hardware to make more practical and novel observations.

Channel states at rendezvous were recorded for each combination of jamming strategy, preamble-base correlator, and transmit/receive-based correlator. When the radios could not rendezvous, like with the preamble-based correlator, the data are taken at the timeout condition. Each plot is similar to Fig. 36, but the more effective

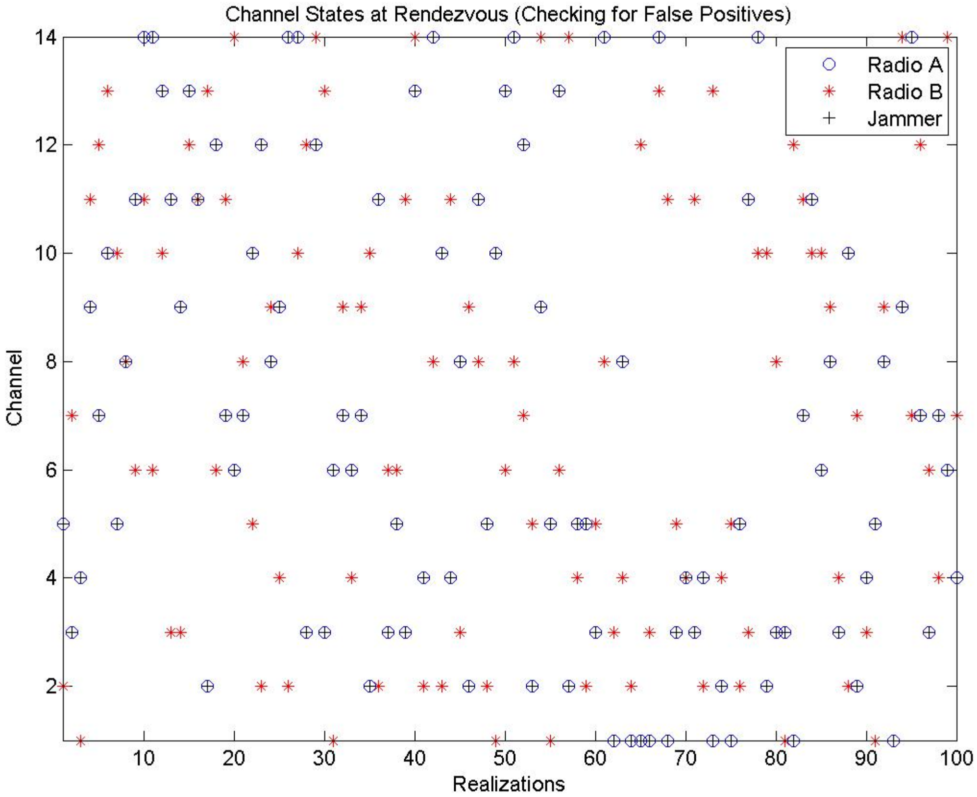

Figure 36. Jammer always on same channel as Radio A for preamble-based correlator using WARP

the jammer was at following Radio A, the more often the jammer, Radio A, and Radio B were on the same channel at rendezvous. Fig. 31 shows the channel states at rendezvous for the jammer always being on the same channel as Radio A for the preamble-based correlator using WARP.

Observations and Commentary.

The hardware limitations of the WARP boards were a significant obstacle in verification. The buffer size limited signal input to be 20 bits with 10 samples per bit, otherwise the buffer would overflow and cause the boards to crash. The insufficiently sized buffer caused the rendezvous process on the WARP boards to be considerably less effective than the potential shown in the simulated version. The short, 4 bit preamble required caused the preamble-based correlator to be completely ineffective.

The WARP provided the ability to simultaneously verify the rendezvous process and jamming thereof as well as test them in a low SINR environment. If a longer correlation sequence could have been used, the WARP's rendezvous process should have proven to be resilient to a low SINR environment.

There was a considerable jump in time to rendezvous. The WARP reached the timeout condition of 200 time steps with high frequency; recall that in a high SINR environment, the time to rendezvous is expected to be 17 time steps. As a consequence, the time to rendezvous was not affected as much by jamming in the WARP case as it was in the simulation case.

The location of the antennas did not likely affect the performance of the tests because all the signals are handled in MATLAB. Although the signals were physically broadcast, the way they interact was done entirely in MATLAB acting as a centralized controller. The exception was the generated receive signal, which was appropriately affected by the noise and interference from the environment and the jammer. A more appropriate way to test the CR interaction is using boards that are programmed separately and do not have a centralized controller. It should also be noted that the lab used is a high-interference environment for the Wi-Fi spectrum. The WARP boards were attempting to rendezvous while other experiments were being done on the same band and wireless internet service was being provided. Such an environment is an appropriate one for CRs to prove real-world algorithm effectiveness and is therefore recommended to be used in any future testing in the physical layer.

V. Conclusions

This chapter summarizes conclusions and suggests future work.

5.1 Conclusions

Longer rendezvous correlation signals are much harder to jam. Shorter rendezvous correlation signals are exceptionally susceptible to low SINR environments. In a high SINR setting with appropriate correlator thresholding, rendezvous times via Jump-Stay matches published data at the waveform level as well as the network level. The lower the SINR, the worse the time to rendezvous performance will presumably be. The effectiveness of the correlator is the cornerstone to the effectiveness of the rendezvous process. Longer correlation signals used in a system yielded better rendezvous performance under jamming conditions. Correlators need to be able to make internal, dynamic calculations about what the threshold for detection should be based on information about the signals of interest. Time delays must similarly be internally calculated to enable real-time synchronization of a jammer targeting a radio. Without proper time delays, a jammer cannot intercept a radio when attempting to predict its channel-hopping pattern, even when the channel-hopping parameters are calculated correctly.

For the simulated version of rendezvous, jamming the preamble-based correlator and the transmit/receive-based correlator did not cause either system to rendezvous on the incorrect channel. Delays in time to rendezvous were caused in the preamble-based correlator but not the transmit/receive-based correlator.

Analysis of the jamming strategy potential can determine what step size and index values to *not* use on part of the radios avoiding a jammer. Particularly, analysis of observation-based parameter estimation yielded vulnerable index and step size, i and

r, values which should be avoided by a CR. The repeater jamming strategy is a simple, easily implemented, and potentially effective approach, but due to the robustness of the Jump-Stay algorithm, the radios will typically converge in P time steps. So, by the time the repeater has observed enough channel state data to be effective, the radios have probably already converged.

Generally, the problem of the jammer intercepting the radios is difficult. There is a small window of time that the jammer has to observe the radios before the converge. The solution that the jammer can estimate in that short time is not always a very robust one. If the jammer can observe all the channels simultaneously, it can make a much better estimate of the Jump-Stay parameters than if its observation space is limited. However, even when the jammer can make accurate estimates in short amounts of time, the robustness of the radio's correlator can nullify jamming attempts. Estimating parameters for the Jump-Stay algorithm required that observations be made that could catch the "wrap-around" effect.

The hardware limitations of the WARP boards were a significant obstacle in verification. The buffer size limited signal input to be 20 bits with 10 samples per bit, otherwise the buffer would overflow and cause the boards to crash. The insufficiently sized buffer caused the rendezvous process on the WARP boards to be considerably less effective than the potential shown in the simulated version. The short, 4 bit preamble required caused the preamble-based correlator to be completely ineffective.

The WARP provided the ability to simultaneously verify the rendezvous process and jamming thereof as well as test them in a low SINR environment. If a longer correlation sequence could have been used, the WARP's rendezvous process should have proven to be resilient to a low SINR environment. There was a considerable jump in time to rendezvous. The WARP reached the timeout condition of 200 time steps with high frequency; recall that in a high SINR environment, the time to rendezvous is expected to be 17 time steps. As a consequence, the time to rendezvous was not effected as much by jamming in the WARP case as it was in the simulation case.

The location of the antennas did not likely effect the performance of the tests because all the signals are handled in MATLAB. Although the signals were physically broadcast, the way they interact was done entirely in MATLAB acting as a centralized controller. The exception was the generated receive signal, which was appropriately effected by the noise and interference from the environment and the jammer. A more appropriate way to test the CR interaction is using boards that are programmed separately and do not have a centralized controller.

The contribution of this thesis has been to show that the rendezvous process for CRNs are vulnerable to jamming and interference. Time to rendezvous can be increased and CRs can be made to rendezvous on incorrect channels by jamming during the rendezvous process. Channel-hopping estimation techniques were developed, but require improvement. Design considerations to protect a CRN from CJs were also significant contributions.

5.2 Future Work

Utilization of different hardware which does not require a centralized controller would provide further useful analysis of CRNs. Further, if a different set of hardware is used, it should have a larger buffer size for transmitting signals to boost correlator performance, assuming the spectrum-sensing is accomplished in a similar fashion. Hardware with a wide receiver bandwidth would also be useful for being able to observe CRN communication history. If alternative hardware is unavailable, optimizing WARP functionality is advised. More specifically, optimization of base functions that generate waveforms would be most helpful in the event the WARP boards are used again.

Optimizing observation-based parameter estimation is also a possibility for future work, since parameter estimation showed great promise in initial testing. Altering implementation to get closer to the performance initially observed could prove to be a worthwhile endeavor.

Investigation of different jamming waveforms, rather than just WGN could be useful. It's likely that if the jammer broadcasts the same waveform-modulation type at the CRs, it would be more effective than band-limited WGN, however there may be less obvious conclusions to reach as well. Researching what jamming waveforms are more effective could also make estimating parameter sets a more useful approach to denying CR communications.

Researching what rendezvous schemes are common in commercial systems would provide tangible and actionable data. Investigating and exploiting weaknesses in real-world rendezvous applications is useful, but would also likely require being protected as classified information. Additionally, research into making real world rendezvous application schemes more resilient would also be useful. An analysis of how robust the other rendezvous methods are at the waveform level would be required.

Bibliography

1. USAF Curtis E. Lemay Center for Doctrine Development and Education, “Annex 3-51 electronic warfare,” 2015.

2. Defense News, “Electronic warfare: What u.s. army can learn from ukraine,” 2015.

3. Zhang Jian-zhao, Zhao Hang-sheng, and Yao Fu-qiang, “A fast neighbor discovery algorithm for cognitive radio ad hoc networks,” in *2010 12th IEEE International Conference on Communication Technology (ICCT)*, 2010, pp. 446–449, ID: 1.

4. C. J. L. Arachchige, S. Venkatesan, and N. Mittal, “An asynchronous neighbor discovery algorithm for cognitive radio networks,” in *3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2008. DySPAN 2008.*, 2008, pp. 1–5, ID: 1.

5. A. Asterjadhi and M. Zorzi, “Jenna: A jamming evasive network-coding neighbor-discovery algorithm for cognitive radio networks,” in *2010 IEEE International Conference on Communications Workshops (ICC)*, 2010, pp. 1–6, ID: 1.

6. Jongmin Shin, Dongmin Yang, and Cheeha Kim, “A channel rendezvous scheme for cognitive radio networks,” *IEEE Communications Letters*, vol. 14, no. 10, pp. 954–956, 2010, ID: 1.

7. D.B. Rawat and Min Song, “Securing space communication systems against reactive cognitive jammer,” in *2015 IEEE Wireless Communications and Networking Conference (WCNC)*, March 2015, pp. 1428–1433.

8. T. Yucek and H. Arslan, “A survey of spectrum sensing algorithms for cognitive radio applications,” *Communications Surveys & Tutorials, IEEE*, vol. 11, no. 1, pp. 116–130, 2009, ID: 1.

9. Bernard Sklar, *Digital Communications*, Prentice Hall, 2 edition, 2001.

10. N. C. Theis, R. W. Thomas, and L. A. DaSilva, “Rendezvous for cognitive radios,” *Mobile Computing, IEEE Transactions on*, vol. 10, no. 2, pp. 216–227, 2011, ID: 1.

11. Zhiyong Lin, Hai Liu, Xiaowen Chu, and Y. W Leung, “Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio networks,” in *2011 Proceedings IEEE INFOCOM*, 2011, pp. 2444–2452, ID: 1.

12. Dagang Xie, Jing Li, Chun Wang, Qifeng Liu, and Sheng Ye, “Analysis and simulation of typical mode of jamming on data link communication system,” in *2013 IEEE 5th International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications (MAPE)*, 2013, pp. 530–533, ID: 1.

13. M. A. Richard, J. A. Scheer, and W. A. Holm, *Principles of Modern Radar Basic Principles*, Sci-Tech Publishing, Edison, NJ, 2010.

14. Mathworks, “Periodogram power spectral density estimate,” 2015.

15. Kate Yaxley, “Communication and jamming bda of ofdma communication systems using the software defined radio platform warp,” M.S. thesis, Air Force Institute of Technology, 2015.

16. WARP Project, “Warp hardware users guide,” 2015.

17. K. Amiri, Yang Sun, P. Murphy, C. Hunter, J.R. Cavallaro, and A. Sabharwal, "Warp, a unified wireless network testbed for education and research," in *IEEE International Conference on Microelectronic Systems Education, 2007. MSE '07*, June 2007, pp. 53–54.

JAMMING COGNITIVE RADIOS

Travis J. Freeman

March 2016

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*
24-03-2016

2. REPORT TYPE
Master's Thesis

3. DATES COVERED *(From — To)*
Sep 2014 — Mar 2016

4. TITLE AND SUBTITLE
Jamming Cognitive Radios

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Freeman, Travis, J., Capt, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT-ENG-MS-16-M-015

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Lab, Sensors Directorate, Electronic Warfare Branch
2241 Avionics Circle
WPAFB OH 45433-7765
COMM (937) 785-5579 ext 4245
Email: vasu.chakravarthy@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RYWE

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES
This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

The goal of this thesis is to identify and evaluate weaknesses in the rendezvous process for Cognitive Radio Networks (CRNs) in the presence of a Cognitive Jammer (CJ). Jamming strategies are suggested and tested for effectiveness. Methods for safeguarding the Cognitive Radios (CRs) against a CJ are also explored. A simulation is constructed to set up a scenario of two CRs interacting with a CJ. Analysis of the simulation is conducted primarily at the waveform level. A hardware setup is constructed to analyze the system in the physical layer, verify the interactions from the simulation, and test in a low signal-to-interference and noise ratio (SINR) environment. The hardware used in this thesis is the Wireless Open-Access Research Platform.

15. SUBJECT TERMS
LaTeX,Thesis

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr. Richard Martin,(ENG)
					19b. TELEPHONE NUMBER *(include area code)*
U	U	U	U	101	(937) 785-3636 x4625;richard.martin@afit.edu