

EFFICIENT BIT ENCODING IN BACKSCATTER WIRELESS SYSTEMS

A Thesis
Presented to the
Academic Faculty

By

Patrick A. Graf

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science in Electrical and Computer Engineering

Georgia Institute of Technology

May, 2010

EFFICIENT BIT ENCODING IN BACKSCATTER WIRELESS SYSTEMS

Approved by:

Dr. Gregory D. Durgin, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Andrew F. Peterson
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Mary Ann Ingram
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: April 5, 2010

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my parents, Harvey and Kathleen Graf, for being there and giving me advice throughout my life and academic career. Without their support, I would never have been able to accomplish what I have to this point. I would like to thank my adviser, Dr. Gregory Durgin, for the advice he has given me throughout my time at the Georgia Institute of Technology. His probing questions, refreshing outlook on research and support throughout my time in the Propagation Group were extremely important in shaping me as an engineer. I would like to thank Dr. Andrew Peterson and Dr. Mary Ann Ingram for giving me advice throughout the writing of the thesis and for putting up with my last minute requests. Also, I would thank Ryan Pirkl and Gregory Koo for their assistance in generating the base periodic versions of the specific pseudorandom sequences under test. Finally, I would like to thank my undergraduate professors Dr. Steven Jacobs, Dr. Kevin Chen and Dr. Joel Falk for spurring me to think outside the box and for encouraging me to pursue graduate education.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	iii
LIST OF TABLES	vi
LIST OF FIGURES	viii
SUMMARY	ix
CHAPTER 1: OVERVIEW OF CURRENT TECHNOLOGY	1
1.1: Brief History and Motivation for Passive Sensing.....	1
1.2: Wireless Backscatter Radio and its Application in RFID.....	3
1.2.1: Semi-Passive UHF RFID Tags.....	5
1.2.2: Passive UHF RFID Tags.....	7
1.2.3: Chipless RFID Technology.....	9
1.3: Wireless Sensors and Backscatter Radio.....	10
1.3.1: Traditional Wireless Sensing.....	10
1.3.2: Application of Wireless Backscatter Techniques for Sensing.....	11
1.3.3: Overview of Current Backscatter Sensing Protocol.....	12
CHAPTER 2: SPREAD SPECTRUM: PERIODIC VS. APERIODIC	14
2.1: Basic Theory behind Spread Spectrum Operation	14
2.2: Pseudorandom Sequence Generation	16
2.2.1: Maximal Length Sequences	16
2.2.2: Other Sequences Used in Spread Spectrum	17
2.2.2.1: Gold Sequences	18
2.2.2.2: Small Set of Kasami Sequences	18
2.3: Analysis of the Cross-Correlation Function	19
2.3.1: The Periodic Cross-Correlation Function	19
2.3.2: The Aperiodic Cross-Correlation Function	21
2.3.3: A Practical Modification to the Aperiodic Cross-Correlation Function..	24
2.4: Issues with Using Spread Spectrum in Backscatter Communication	27
2.4.1: The Near-Far Problem.....	28
2.4.2: The Power Requirements of Spread Spectrum.....	30
CHAPTER 3: DEVELOPMENT OF APERIODIC CODE SETS	33
3.1: Characteristics Guiding Code Selection	33
3.1.1: Zero-Length	34
3.1.2: Codes Included in Group	34
3.1.3: Code Rotation Found in System	36
3.2: Test Plan	40
3.2.1: Code Configurations to Be Studied	40

3.2.2: Specific Codes under Test	41
3.2.3: Code Rotations Tested in the System	43
3.2.3.1: Minimizing Partial Correlation	43
3.2.3.2: Standardizing Computation	44
3.2.3.2.1: Same Number of Rotations per Code (Primary Test)	44
3.2.3.2.2: Varying Number of Rotations per Code (Secondary Test) ..	45
3.2.3.3 Analyses	46
CHAPTER 4: ANALYSIS OF SIMULATION RESULTS	47
4.1 Overall Analyses to Be Performed	48
4.2 Single Node System	50
4.2.1: One Node, One Code	50
4.2.2: One Node, Two Codes	52
4.2.3: One Node, Four Codes	53
4.3 Two Node Sequence	54
4.3.1: Two Nodes, One Code	55
4.3.2: Two Nodes, Two Codes	56
4.3.3: Two Nodes, Four Codes	57
4.4 Three Node System	58
4.4.1: Three Nodes, One Code	59
4.4.2: Three Nodes, Two Codes	61
4.4.3: Three Nodes, Four Codes	62
4.5 Comparison between the Best and Worst Case Code Configurations	64
CHAPTER 5: CONCLUSION AND FUTURE WORK	66
5.1 Analysis of Periodic Code Generation.....	66
5.1.1 Methods for Generating Code Configurations	66
5.1.2 Comparison of Aperiodic Results with Periodic Spread Spectrum	67
5.2 Future Work	67
APPENDIX A: BASE CODES USED IN THE SIMULATIONS	69
REFERENCES	73

LIST OF TABLES

Table 1:	Relevant cross-correlations for 3x2 sequence configuration	39
Table 2:	Various Sequence Configurations	40
Table 3:	Codes under Test	41
Table 4:	Number of Code Rotations in Analysis (Primary)	45
Table 5:	Number of Code Rotations in Analysis (Secondary)	46
Table 6:	Code Configurations for 63-Chip Kasami, 127-Chip Gold and 255-Chip Kasami Codes	48
Table 7:	Number of Correlations Considered for Each Code Configuration.....	49
Table 8:	Code Selection and Rotation for One Node with One Code Used per Node	51
Table 9:	Code Selection and Rotation for One Node with Two Codes Used per Node	52
Table 10:	Code Selection and Rotation for One Node with Four Codes Used per Node	53
Table 11:	Code Selection and Rotation for Two Nodes with One Code Used per Node	55
Table 12:	Code Selection and Rotation for Two Nodes with Two Codes Used per Node	56
Table 13:	Code Selection and Rotation for Two Nodes with Four Codes Used per Node	57
Table 14:	Code Selection and Rotation for Three Nodes with One Code Used per Node	59
Table 15:	Code Selection and Rotation for Three Nodes with Two Codes Used per Node	61
Table 16:	Code Selection and Rotation for Three Nodes with Three Codes Used per Node	63

Table 17: Comparison between the Code Configurations that Provide the Lowest and Highest Overall Average Correlation	64
Table 18: 63-Chip Sequences from the Small Set of Kasami Sequences	69
Table 19: 127-Chip Sequences from the Set of Gold Sequences	70
Table 20: 255-Chip Sequences from the Small Set of Kasami Sequences	71

LIST OF FIGURES

Figure 1: Basic Transmission Line Circuit	4
Figure 2: Semi-Passive RFID Circuit Diagram (based on example from [4]).....	5
Figure 3: BPSK Signal Space Diagram (Normalized by $E_g^{1/2}$)	6
Figure 4: Passive RFID Circuit Diagram (based on example from [4]).....	7
Figure 5: OOK Signal Space Diagram (Normalized by $E_g^{1/2}$)	8
Figure 6: Basic Block Diagram for the Traditional Wireless Sensor	10
Figure 7: Block Diagram of WISP Functionality (based on block diagram from [12])	11
Figure 8: Multiplication of the Data Stream with a Fast-Rate Spread Spectrum Sequence	15
Figure 9: Simple Graphical Explanation of Spread Spectrum	15
Figure 10: Shift Register Method for m-Sequence Generation	17
Figure 11: Aperiodic Sequences from the Small Set of 63-Chip Kasami Sequences (Codes 2 and 3)	22
Figure 12: Aperiodic Autocorrelations of the 2 nd and 3 rd Codes from the 63-Chip Small Set of Kasami Sequences	24
Figure 13: Sequence Composed of the 2 nd and 3 rd Kasami Codes, Separated by Zero-Spacing	25
Figure 14: Aperiodic Cross-Correlation of the Joint Sequence with Each of the Constituent Kasami Sequences (Codes 2 and 3)	26
Figure 15: Plot of Two Code Rotations of the 2 nd Kasami Sequence (Rotations 0 and 32)	37

SUMMARY

As the size and power consumption of microelectronic circuits continues to decrease, passively-powered sensors promise to come to the forefront of commercial electronics. One of the most promising technologies that could realize this goal is backscatter sensing. Backscatter sensors could harvest power from and modulate data onto an impinging carrier waveform. Currently radio frequency identification (RFID) technology passively powers itself and transmits statically stored data. However, this technology has two major weaknesses: lack of resiliency against narrowband interference and slow data rates. Both of these issues could be detrimental in sensing applications. This thesis will lay out a method for addressing both of these weaknesses through a unique application of spread spectrum encoding. Instead of spread spectrum being viewed as the multiplication of an already encoded data sequence with a periodic pseudorandom sequence, each sequence could be viewed in an aperiodic manner, where a single period of a pseudorandom sequence represents a data symbol. In this manner, backscatter sensors not only benefit from the increased resiliency that spread spectrum provides, but also can have higher data rates, since multiple bits can be encoded on a single symbol and multiple nodes can be read simultaneously, using spread spectrum multiple access techniques. In this thesis, 63-chip and 255-chip Kasami sequences, as well as 127-chip Gold sequences, will be analyzed for their use in various aperiodic direct sequence spread spectrum/multiple access system configurations (systems that have up to three nodes and use up to four different aperiodic sequences per node to represent different symbols). For each different configuration, near-“ideal” code configurations/rotations will be determined for use in the system.

CHAPTER 1: OVERVIEW OF CURRENT TECHNOLOGY

1.1 Brief History and Motivation for Passive Sensing

Throughout history, advances in communication and technological innovation have gone hand in hand. In ancient times, long distance communication first became feasible as infrastructure, such as an extensive road network, was constructed by the Romans, arguably the world's first "engineers." Later, with the onset of the industrial revolution, the speed at which people could communicate increased rapidly. Due to innovations, such as the steam ship, letters that previously took months to deliver only took weeks. The development of the telegraph allowed messages to be carried across an ocean in a matter of seconds. Finally, in the 20th century, the telephone allowed people in other states and even other countries to communicate verbally with one another as if they were in the next room. However, with the publication of Claude Shannon's "A Mathematical Theory of Communication", communication was changed forever.

In a single paper, Shannon founded modern information theory and the field of digital telecommunication [1]. This paper showed that vast amounts of digital data could be transmitted without errors, which was previously thought to be impossible. The primary guideline governing error-free communication (and its theoretical limitations) can be seen in Shannon's Channel Capacity formula:

$$C = B \log_2(1 + SNR) \quad (1)$$

where

$$SNR = P / (N_o B) \quad (2)$$

and

C: Channel Capacity

P: Received Power

B: Channel Bandwidth

N_o: Noise Distributed over
Frequency

As can be seen in formula (1), the capacity, i.e. maximum symbol rate, of a communication channel is related to both the bandwidth (*B*) and the signal-to-noise ratio (SNR) of the link, presupposing noise can be modeled as white, Additive White Gaussian Noise (AWGN). *B* is a variable that is dictated by the physical limitations of the channel and, in the case of wireless communication, the regulations imposed by the local authority, which is the Federal Communications Commission (FCC) in the case of the United States. On the other hand, SNR is related to both the received power of the signal and the noise added by the channel. In the case of wireless communication, the transmit power is finite and, in mobile applications, is even more so. Take the cell phone or smart phone for example. A smart phone is an extremely useful handheld device that allows the user not only the ability to communicate with others, but also to surf the web (via both the cell network and wireless internet), get directions, etc. As each of these different applications is accessed, different transceivers must be activated and power is drained as a result. Thus, the handset must be recharged on almost a nightly basis. This example shows how costly communication can be in terms of power, especially when the wireless device must be powered from a battery.

Although power limitations can be seen most clearly in everyday devices, such as smart phones, they exist for almost every mobile, wireless device. In particular, this problem

has plagued the field of wireless sensing. Wireless sensors can be extremely useful for acquiring data and transmitting the information back to a receiver. However, like cell phones, they have a limited lifetime, which is dictated primarily by battery life. Either the battery technology must be improved or the sensor must be powered remotely, in order to extend the sensor lifetime. Remotely powering sensors is becoming more of a reality as advances in backscatter wireless communication are made.

In the world today, one of the most promising technologies is backscatter wireless communication. The most prominent application of this technology is radio frequency identification (RFID). As can be seen in [2], RFID tags are used as “keys” to open electronic locks, as a means of storing personal information in both credit cards and passports and as tracking mechanisms in the shipping industry. Although RFID is extremely useful, backscatter radio could be implemented in a number of other areas, in particular wireless sensor networks. With backscatter communication, the power limitations of the traditional wireless sensor can be circumvented. Due to the limited hardware capabilities of passively-powered, backscatter sensors, new modulation schemes must be developed that can allow simultaneous communication from multiple nodes while maintaining a “reasonable” data rate (this is currently not feasible in traditional RFID implementations).

1.2 Wireless Backscatter Radio and its Application in RFID

In wireless backscatter radio, communication is not carried out through the traditional use of a transceiver unit. Rather, this form of wireless communication is built upon transmission line theory and reflection off a load. An overview of transmission line theory and radio frequency (RF) circuits can be found in [3] and a diagram of a basic transmission line structure (connected to a general source) can be found in Figure 1.

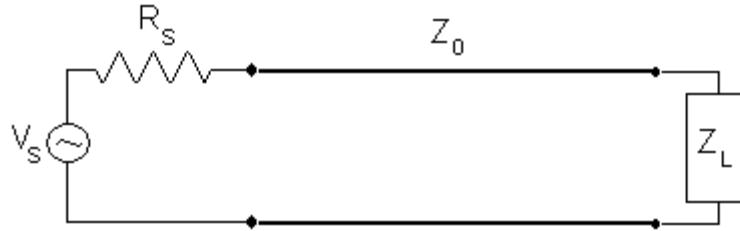


Figure 1: Basic Transmission Line Circuit

In simplest terms, an electromagnetic wave with frequency f_o impinges on an antenna and excites a current with the same frequency of the original wave. The current waveform propagates along the transmission line and reflects off the load with impedance Z_L . The reflected current waveform propagates back toward the antenna and is reradiated. In the static case, where the load impedance remains constant, the reradiated wave has the same frequency as the original impinging wave, but with a phase offset caused by the load. However, when the load is allowed to change, this setup is capable of modulating data upon the original waveform, which can be received and decoded. How the impinging wave is affected by the load Z_L is dictated by the formula for the reflection coefficient (Γ), found in [3]:

$$\Gamma = \frac{Z_L - Z_o}{Z_L + Z_o} \quad (3)$$

where

Γ : Reflection Coefficient

Z_L : Load Impedance

Z_o : Line Impedance

For backscatter radio in particular, three states are especially useful: $\Gamma = -1$ (when the load resistance is a short circuit), $\Gamma = 1$ (when the load resistance is an open circuit) and

$\Gamma = 0$ (load resistance is matched to the line resistance). The significance of these states will be explained in the following sections.

1.2.1 Semi-Passive UHF RFID Tags

Semi-passive UHF RFID tags operate in the far-field of the RFID reader (where field strength falls off at a rate of $1/r$) and use battery assistance to operate the tag circuitry. An example circuit semi-passive RFID circuit diagram can be found in Figure 2:

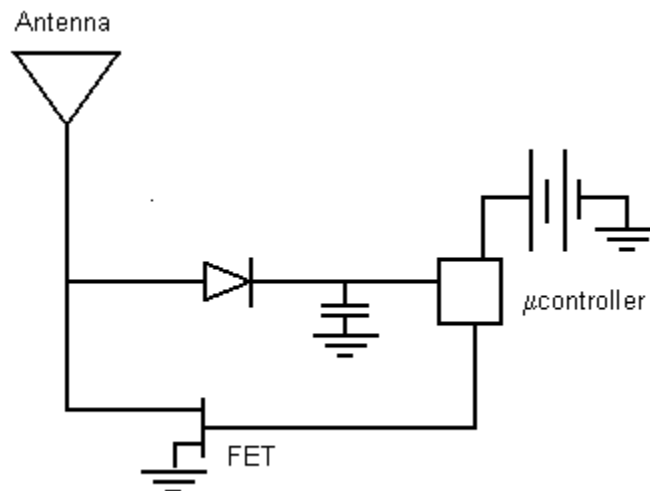


Figure 2: Semi-Passive RFID Circuit Diagram (based on example from [4])

As can be seen above, communication is performed through the backscatter of an impinging wave by a modulating load. The states of the load impedance are controlled by a microcircuit, which is powered by a local power source (battery). Since there is a local power source, the circuit does not rely on incoming radiation to power its circuitry. Thus, in order to reradiate the highest signal power, the load should switch between a short and

an open circuit, which in turn generates reflection coefficients of -1 and 1. (In practice some semi-passive circuits do not take advantage of this benefit and use the less efficient states of 0 and 1, which will be explained in Section 1.2.2) Theoretically, the impinging wave is reflected completely and maximum power is returned. However, as was shown by Griffin in [5], this is not always the case, since load impedance can be affected by environment.

Since this is a communication link, it is useful to analyze communication scheme using more traditional telecommunication techniques. Since the reflection coefficient switches between -1 and 1, this is technically a two-state, antipodal modulation scheme, otherwise known as binary phase shift keying (BPSK), as is defined in [6]. The signal-space diagram for such a modulation scheme can be seen in Figure 3.

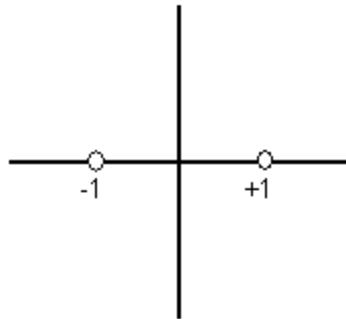


Figure 3: BPSK Signal Space Diagram (Normalized by $E_g^{1/2}$)

In the above picture, the constellation points are spaced apart by $2\sqrt{E_g}$, where E_g is the energy of the communication symbol, or bit, in the case of BPSK. Since this is a semi-passive backscatter link, the energy E_g is entirely dependent upon the path loss accrued in the forward link. Semi-passive tags are actually quite popular in practice, due to their

reliability and read range, and can be found in multiple applications, most notably in highway toll passes, such as EZPASS, which is discussed in [7] and [8].

1.2.2 Passive UHF RFID Tags

As with the semi-passive tags in the previous section, the passive UHF tags also operate in the far-field of the RFID reader. However, unlike the semi-passive tags, these tags have no local power source and must harvest energy from the impinging wave, in order to power the on-board circuitry. An example of this tag design can be found in Figure 4.

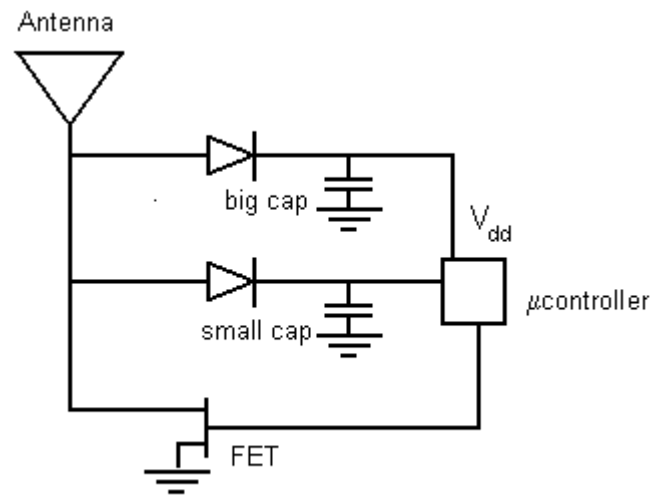


Figure 4: Passive RFID Circuit Diagram (based on example from [4])

This tag uses a charge pump, which is a type of diode rectifier, to turn the incoming wave into dc power. However, since this circuit harvests power from that wave, it cannot reflect all of the power back for communication. Thus, instead of switching between short and open circuits (with $\Gamma = -1$ and 1) this tag switches between other loads, such as

a short (or open) circuit and a matched circuit (with $\Gamma = -1$ or 1 and 0), so that power may be harvested and not all is reflected.

Because this modulation scheme uses a matched circuit as one of its states, it resembles on-off keying (OOK) instead of BPSK. The differences are laid out by Proakis in [6]. The signal-space diagram for OOK can be seen in Figure 5.

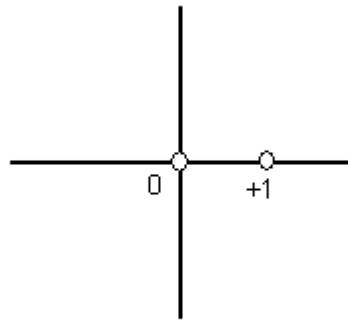


Figure 5: OOK Signal Space Diagram (normalized by $E_g^{1/2}$)

Since the constellation points now reside at 0 and 1 , the distance between both is $\sqrt{E_g}$ which in turn provides a lower SNR than the BPSK modulation used in the semi-passive tag. This makes sense since a purely passive tag must harvest energy from the impinging wave, while the semi-passive tag has a local power source. Thus, less of the incoming radiation can be used for the reradiated signal. This technology is currently used in supply chain management, as can be seen in [9], and shows the most promise for sensor applications.

1.2.3 Chipless RFID Technology

A third type of RFID that is passive but does not use a charge pump and load modulation to provide identification is chipless RFID. The most prominent of these technologies is the Surface Acoustic Wave (SAW) RFID tag. Like traditional RFID, the data is statically stored on the remote tag. However, the modulation technique is markedly different than traditional passive RFID and a brief description of operation is included in this section.

Unlike traditional passive RFID, which can be manufactured using traditional materials and processes, SAW RFID is extremely material dependent. As is described in [10], SAW tags use an interdigital transducer (IDT) to convert the impinging RF wave into a surface wave. This surface wave travels along the chip and is reflected off a series of wave reflectors back toward the antenna. Due to the piezoelectric effects of the SAW tag material, this surface wave can be reradiated, encoded with data by the wave reflectors.

This type of technology has both advantages and disadvantages. Since the tag has no electronics to power, no power must be harvested from the impinging wave and all of that energy can be used to return modulated data to the receiver. This is a distinct advantage of SAW vs. traditional RFID, since this increased reradiated power allows for longer read distances, increased reliability or both. However, since there is no microcircuit on the tag side, the reader has no control over whether the tag radiates and the reader cannot reconfigure tag data. Thus, if multiple tags are in the reader's field of view, coherence is lost and no data can be demodulated. Also, each tag must be individually manufactured, leading to high production costs. Thus, chipless RFID is an interesting technology, but its actual applications are rather limited.

1.3 Wireless Sensors and Backscatter Radio

1.3.1 Traditional Wireless Sensing

Wireless sensors have become an integral part of environmental and structural monitoring, as well as many other applications. Although the theory behind sensors (and subsequently sensor networks) can be extremely complicated, the wireless sensor is composed of four elements: the environmental sensor, a processor, a transceiver and a power supply, as is stated in [11]. A block diagram of a simple sensor can be found in Figure 6 below.

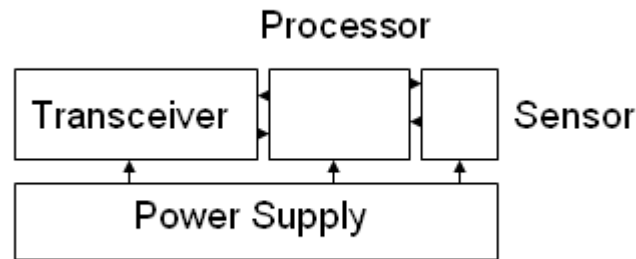


Figure 6: Basic Block Diagram for the Traditional Wireless Sensor

In its simplest form, a sensor contains hardware that is responsible for collecting and encoding data, as well as providing the modulation scheme. On the market, there are currently a number of commercially available sensors, including Zigbee, Bluetooth and Dash-7 sensors. More importantly though, a sensor has both full receiver and transmitter chains, which are comprised of a series of filters, amplifiers and at least one high speed oscillator. These analog components consume too much energy to be powered from an incident RF wave and, therefore, a local power source (usually a battery) must be located onboard to supply the power. Because a battery has a finite lifetime, the sensor must be

subsequently serviced or discarded after several years. Since monitoring applications might be longer than a traditional sensor's lifetime, such as structural sensing, it would be beneficial to remove the necessity of an onboard power supply and backscatter radio is one of the clearest paths forward.

1.3.2 Application of Wireless Backscatter Techniques for Sensing

Although most sensing today is based upon the traditional sensor model found in Figure 6, advances are being made in the area of sensing based upon passive, wireless backscatter techniques. The clearest example, which is described in great detail in [12], is the WISP wireless sensing platform generated by a joint venture between the Intel Research Laboratory and the University of Washington. A basic block diagram of the WISP sensor can be found in Figure 7.

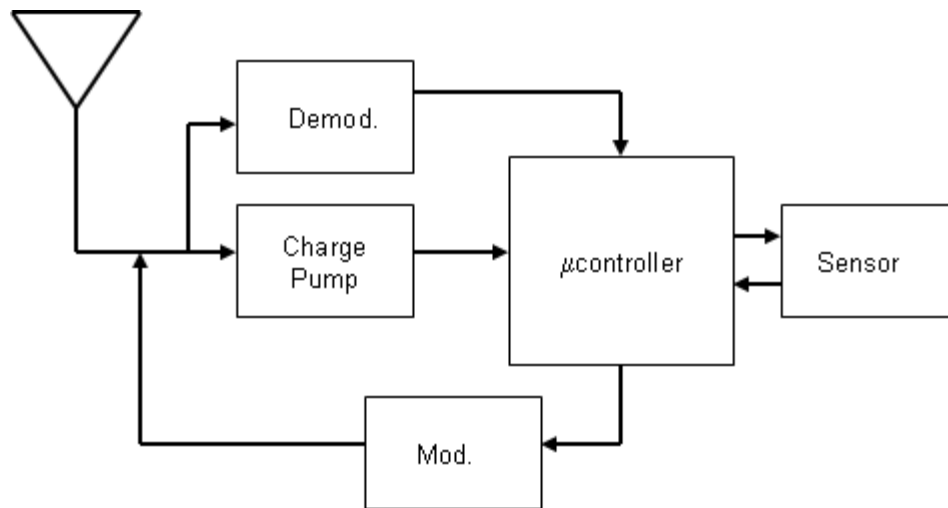


Figure 7: Block Diagram of WISP Functionality (based on block diagram in [12])

This sensor is based upon 915MHz Ultra-High Frequency (UHF) RFID technology. It is capable of passively powering itself and, upon power-up, establishing the two-way communication link that the EPC Global Standard demands of RFID tag. However, as [12] shows, unlike standard RFID tags that have the data stored statically, the WISP can dynamically sense data, build data packets (which include calculating a 16-bit CRC) and transmit the data back to the reader. Thus, this passively-powered sensor removes the necessity of having a local power source and, therefore, is theoretically capable of performing the long-life sensing tasks, such as infrastructure monitoring.

1.3.3 Overview of Current Backscatter Sensing Protocol

As was mentioned in Section 1.3.2, the WISP platform uses traditional RFID readers to carry out communication, meaning the EPC Global protocol governs the radio link. Although this protocol governs all aspects of the communication, it is most interesting to analyze how a reader handles multiple tags present in its field of view. When a reader polls to see if any tags are present, it sends out a general signal that activates any tags that can see the reader, as is described in the EPC Global standard in [13]. These tags respond, sending back their unique identification numbers, and the reader attempts to sort out the signals coming back. If there is only one tag, the reader can easily see which tag is present, get the data and shut it down. However, if there are multiple tag responses, the signals can interfere and make the sorting of data much more difficult.

In order to overcome the difficulties presented by having multiple tags or sensors (WISP) in the field of view, the EPC Global standards employ slotted Aloha reading schemes. In this scheme, the reader polls all tags in the field of view and each tag responds at a predefined time with its tag identification number. However, if two tags respond simultaneously, the reader cannot distinguish between the two and has to poll again. Once the reader can identify a tag, it then has the capability of shutting it down and

polling again to see what unidentified tags still remain in the field of view. Researchers, such as those in [14], are currently looking for ways to modify Aloha schemes so that large numbers of tags can be identified and sorted in the shortest amount of time.

Although the Aloha scheme is capable of separating and identifying many nodes, it is still capable of communicating with only one node at any given time. For RFID tags, which have only a small amount of data that must be communicated only once, and for slow sensing operations, this scheme is adequate for the data bandwidth demands placed upon the system. However, if larger amounts of data must be communicated by multiple nodes, this scheme could prove to be too slow and, in a manner of speaking, bandwidth could be easily saturated. Thus, other techniques, such as spread spectrum multiple access, must be employed in order to solve this looming issue.

CHAPTER 2: SPREAD SPECTRUM: PERIODIC VS. APERIODIC

As can be seen in the current wireless backscatter protocol, which teams an EPC Global standard with Aloha polling scheme at the reader, a “multiple-access” scheme is achieved. However, in this setup, there is a fair amount of overhead time to detect what nodes are in the field of view. More importantly, only one node is capable of transmitting at a time. In some sensor applications this may be acceptable, but when a lot of data must be received from multiple nodes and processed in real-time, this scheme is lacking. Thus, to allow multiple nodes to transmit simultaneously, spread spectrum multiple-access must be employed.

2.1 Basic Theory behind Spread Spectrum Operation

As can be seen in [15] and [16], spread spectrum is a scheme that allows multiple users to transmit at the same time in the same frequency band. Each node is assigned a unique pseudorandom sequence, which is generated according to methods found in Section 2.2. At the last stage before transmitting data, the node multiplies (XORs) each data symbol by the pseudorandom sequence, as can be seen in Figure 8.

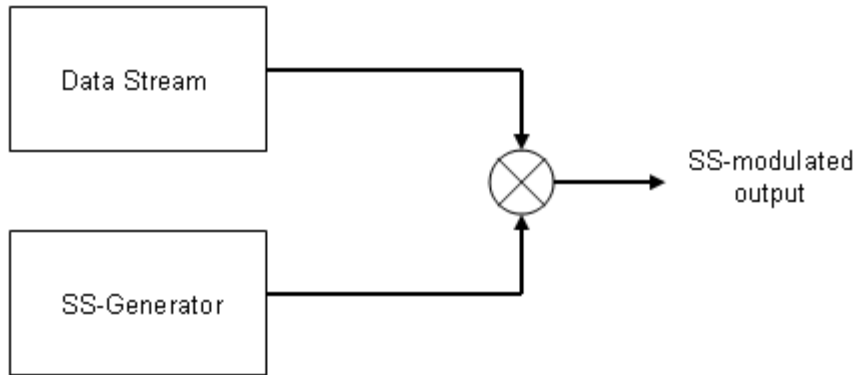


Figure 8: Multiplication of the Data Stream with a Fast-Rate Spread Spectrum Sequence

The multiplication by the pseudorandom sequence takes the narrowband data and “spreads” its spectrum over a larger frequency band. This spread data spectrum, thus, becomes more resilient against narrowband interference, since less data content is present at the frequency being distorted. A simple graphical example of how the frequency spectrum of a narrowband data signal is changed through this process can be found in Figure 9.

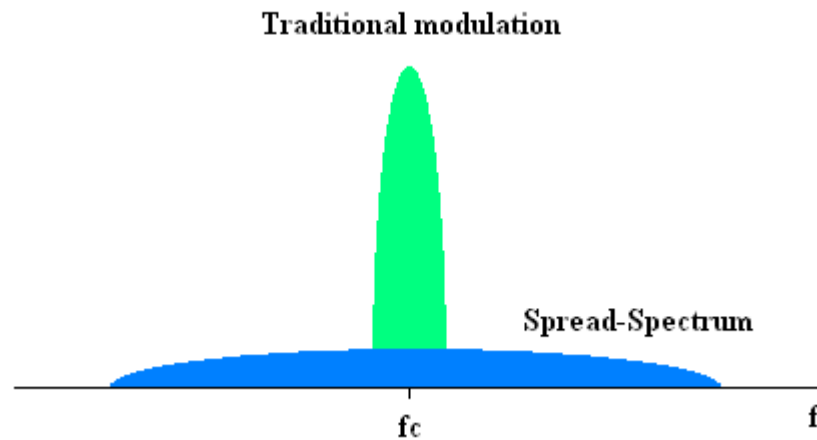


Figure 9: Simple Graphical Explanation of Spread Spectrum

In the single user case, spread spectrum modulation gives the signal greater resiliency. In the multiple user case, however, additional benefits of spread spectrum can be utilized. Since each node is assigned a unique sequence, the frequency content of the various simultaneously transmitting data streams can be viewed as additive and existing in the same time/frequency space. The receiver uses the pseudorandom sequence of each node to separate the various data streams and reconstruct the data. In order to separate the data sequences effectively, it is important that the pseudorandom sequences being employed have very good cross-correlation properties. The generation of pseudorandom sequences with good cross-correlation properties, as well as the cross-correlation function itself, will be studied in the following sections.

2.2 Pseudorandom Sequence Generation

In order to perform Direct Sequence Spread Spectrum/Multiple Access (DSSS/MA), it is first important to determine what sequences are useful in multiple access communication. Some authors, such as Komo and Liu in [17], have attempted to modify existing sequences to find new sets of pseudorandom codes for spread spectrum without much success. Finding new code sets is a daunting task, since the mathematics behind code generation is extremely difficult and there are very few ways to go about code generation using theory. Over the years, one group of sequences has shown up in communications time and again. This group is known as maximal length sequences (m-sequences).

2.2.1 Maximal Length Sequences

Maximal length sequences have very good autocorrelation properties and, more importantly, are easy to generate. An m-sequence can be generated using a simple shift register with feedback taps. An illustration of the generation scheme can be found in Figure 10.

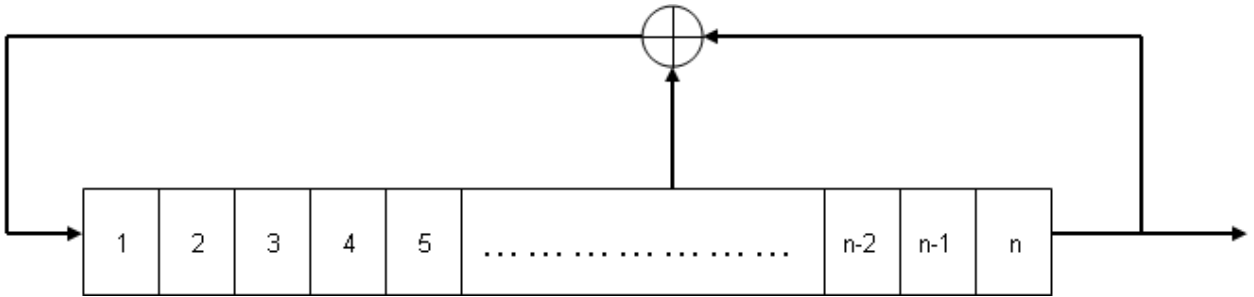


Figure 10: Shift Register Method for m-Sequence Generation

As the Figure 10 shows, an n -stage shift register, with feedback, is used to produce various m -sequences of period length $(2^n - 1)$ chips. Placement of the feedback taps on the shift register changes the m -sequences that are generated. Several authors, such as Skaug and Hjelmstad in [18] list the tap placements required for different register lengths to produce different m -sequences.

As field theory also demonstrates, preferred pairs of m -sequences, which have not only excellent autocorrelation properties, but also good mutual cross-correlation properties (a three-valued bounded cross-correlation) may exist. However, as Sarwate and Pursley showed in [19], the number of preferred pairs varies with code length and some code lengths have no preferred pairs. Thus, in order to perform DSSS/MA, larger sets of sequences with both good auto- and cross-correlation properties must be used.

2.2.2 Other Sequences Used in Spread Spectrum

As was stated above, m -sequences are the class of sequences that have excellent autocorrelation properties. However, there are only a small number of preferred pairs (or in some cases no preferred pairs) for m -sequences of different lengths. For that reason, other sequence sets that have more uniform cross-correlation properties were sought and

the two most often used in asynchronous DSSS/MA are Gold sequences and the small set of Kasami sequences, both of which are based upon m-sequences.

2.2.2.1 Gold Sequences

Because the number of preferred m-sequences is either inconsistent or lacking, mathematicians searched for larger code sets that had good cross-correlation properties. Codes with good correlation properties can actually be generated from m-sequences and one major example is the set of Gold sequences/codes. Gold originally laid out the process of sequence generation in [20]. This code set begins with a preferred pair of m-sequences, which are not only used to generate other codes in the set but are also members of the set themselves. One of the two m-sequences is cyclically shifted (one value at a time) and is subsequently XORed to the other sequence. The result of the XOR is a “Gold sequence.” This operation is continued until all cyclic shifts of the m-sequence have been XORed with the other m-sequence. In the end, the set of Gold sequences contains (N+1) sequences, where N is the original length of the generator m-sequences.

2.2.2.2 Small Set of Kasami Sequences

Gold codes are an example of a set of a larger set of sequences with good cross-correlation properties. However, another code set, called the small set of Kasami sequences, which has even better mutual cross-correlation characteristics than Gold sequences, can be generated. Sarwate and Pursley describe the sequence generation in [19]. Like with Gold codes, this sequence set is generated using m-sequences. First, an m-sequence is generated according to the method stated in Section 2.2.1 and this is the first code of the Kasami set. Then, this same m-sequence is decimated by $2^{n/2}+1$, where n is the length of the shift register used to produce the m-sequence, and a new sequence of the same length as the original m-sequence is produced. However, this sequence is not

part of the set but, rather, is used to generate the remaining codes in the set. The original m-sequence is XORed with the first $2^{n/2}-2$ cyclic shifts of the “generator” sequence. In the end, the small set of Kasami sequences generated should contain $2^{n/2}-1$ unique codes.

2.3 Analysis of the Cross-Correlation Function

In DSSS/MA, two properties of pseudorandom signals stand out above all others: the ability to distinguish a sequence from a cyclic rotation of itself and the ability to distinguish between different sequences within the same set. Both of these can be viewed most appropriately using the cross-correlation function. Generally, the cross-correlation between any two sequences can be found using the following formula from [15]:

$$R_{uv}(\tau) = \int_{-\infty}^{\infty} u(t)v(t-\tau)dt \quad (4)$$

Ideally, if $u(t)$ and $v(t)$ were two pseudorandom signals from the same set, the cross-correlation $R_{uv}(\tau)$ would be 0 for all time. At the same time, the autocorrelation of one of the functions, let us choose $u(t)$, would be:

$$R_{uu}(\tau) = \int_{-\infty}^{\infty} u(t)u(t-\tau)dt \quad (5)$$

In an ideal scenario, this function would have a value only at $t = 0$, when the sequence is perfectly in phase with itself. However, the sequences are rarely ideal and the actual analysis is provided in this section.

2.3.1 The Periodic Cross-Correlation Function

In the traditional analysis of spread spectrum signals, such as Walsh, Gold or Kasami sequences, the periodic cross-correlation function (PCCF) is most analyzed, due to the

more readily available mathematical characterizations. From the equations above, let $u(t)$ and $v(t)$ equal the following (as is portrayed by Sarwate and Pursley in [19]):

$$u(t) = \sum_{n=-\infty}^{\infty} u_n \varphi(t - nT_c) \quad (6a)$$

and

$$v(t) = \sum_{n=-\infty}^{\infty} v_n \varphi(t - nT_c) \quad (6b)$$

where

u : $\{\dots, u_0, u_1, \dots, u_{N-1}, \dots\}$

v : $\{\dots, v_0, v_1, \dots, v_{N-1}, \dots\}$

$\varphi(t)$: pulse of time period T_c

Thus, the functions $u(t)$ and $v(t)$ can instead be handled as infinite sequences, u and v . Also, for this analysis, assume that u and v are periodic sequences. Finally, since this paper handles cross-correlations between sequences from the same code sets, it is assumed that the period lengths of both u and v are equal to N . Thus, a period for each of these sequences would be $\{u_0, u_1, \dots, u_{N-1}\}$ and $\{v_0, v_1, \dots, v_{N-1}\}$. Since these are periodic functions, the cross-correlation function is drastically simplified from a convolution integral to the convolution sum found in equation (7).

$$C(l)(u, v) = \sum_{k=-\infty}^{\infty} u_k v_{l+k} \quad (7)$$

where

u, v : infinite sequences

$C(l)$: cross-correlation function

Although it was stated that u and v were periodic, their periodicity had no bearing on the development of the formula and, therefore, this convolution sum actually holds for the cross-correlation of any two infinite sequences. However, if periodicity is taken into

account, the convolution sum can be further reduced to a basic inner product. As was shown by Sarwate in [19], the PCCF can be expressed as:

$$\theta(l)(u, v) = \langle u, T^l v \rangle \quad (8)$$

where

- θ : periodic cross-correlation
- u, v : one period of each pseudorandom sequence
- T : circular shift operator
- l : amount of code shift

For asynchronous DSSS/MA systems, this function is extremely powerful. For m-sequences, Gold sequences and the small set of Kasami sequences, strict bounds are placed upon the cross-correlation between sequences in each respective set. For example, take the small set of Kasami sequences, where the period length is 63 chips. The autocorrelation of any sequence in the set has a peak of 63. However, the cross-correlation of any sequence with any other in the set is bound to have a peak periodic cross-correlation with a magnitude of at most 9. Thus, there is a deterministic relationship and thresholds for sequence detection can easily be set in a multiple access system.

2.3.2 The Aperiodic Cross-Correlation Function

Although the periodic cross-correlation function has well defined parameters, this is still a limited analysis. In order to gain the most general insight into the behavior of the pseudorandom sequence, the aperiodic cross-correlation characteristics should be studied as well. As seen in equations (6) and (7), the pseudorandom code can be reduced to a sequence and the cross-correlation integral can be reduced to a convolution sum.

However, in the aperiodic case, the code is no longer viewed as just one period in an infinitely long periodic sequence. Rather, the code itself is a time limited sequence and the sequences u and v have the form:

$$u: \{\dots, 0, u_0, u_1, \dots, u_{N-1}, 0, \dots\}$$

$$v: \{\dots, 0, v_0, v_1, \dots, v_{N-1}, 0, \dots\}$$

Unlike the periodic case, the sequence looks like one period of the previous pseudorandom sequence surrounded by zeros on either side. The aperiodic representation of the second and third sequence from the small set of Kasami codes, generated from the m-sequence using feedback taps at [6 1], can be seen in Figure 11.

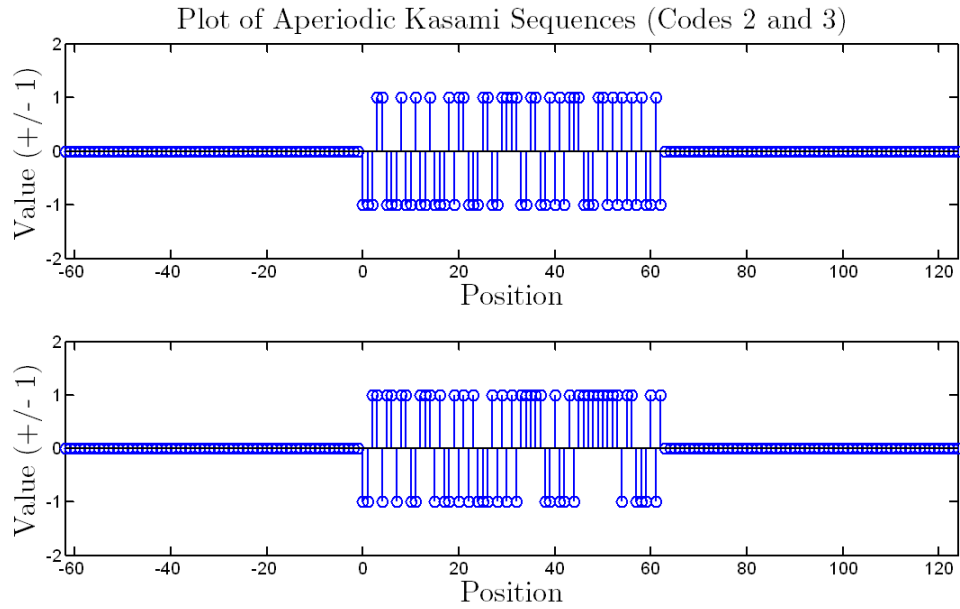


Figure 11: Aperiodic Sequences from the Small Set of 63-Chip Kasami Sequences (Codes 2 and 3)

Because u and v are still infinite sequences (values for a finite number of places and zeros at the rest), the cross-correlation function can be written as an infinite sum, following the relationship set forth in equation (7). However, since values only exist for a limited

number of positions (in our case one period of the pseudorandom sequence), the cross-correlation function can be further reduced and represented as:

$$C(l)(u, v) = \begin{cases} \sum_{k=0}^{N-1} u_k v_{l+k} & -N+1 \leq l \leq N-1 \\ 0 & \text{else} \end{cases} \quad (9)$$

where

u, v : aperiodic sequences

N : code length

C : aperiodic cross-correlation function

It may seem that the aperiodic cross-correlation function here is simpler than found in other sources. However, for this project, it is assumed that both u and v are aperiodic, real sequences of length N . Thus, any complex conjugation necessarily drops out, due to the fact that the sequences are real, and $C(l)$ only has values between $\pm(N-1)$, since both sequences have the same length. The aperiodic autocorrelations of the two sequences plotted in Figure 11 can be seen in Figure 12.

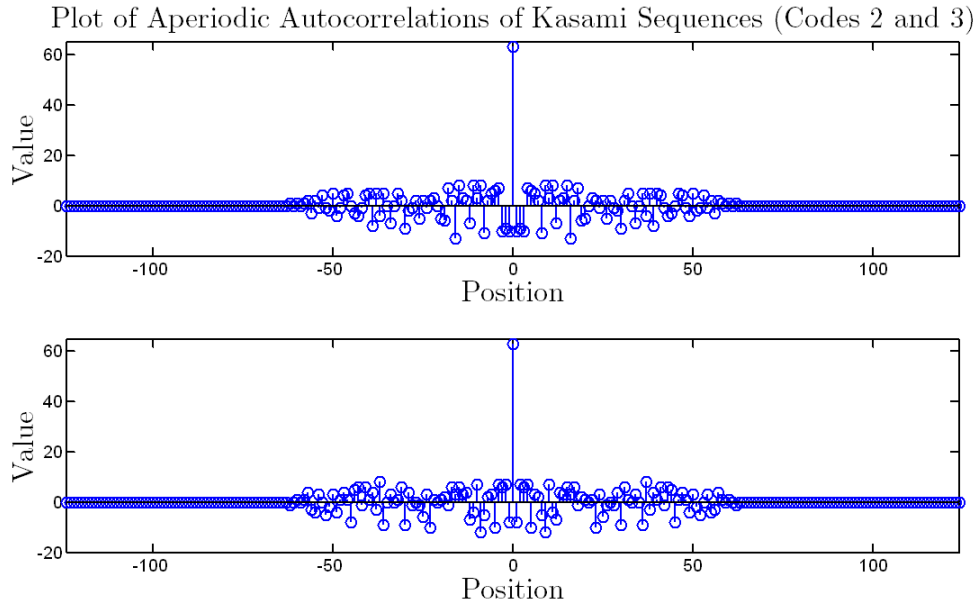


Figure 12: Aperiodic Autocorrelations of the 2nd and 3rd Codes from the 63-Chip Small Set of Kasami Sequences

As can be seen, values for the function exist from -62 to $+62$. The aperiodic cross-correlation is not nearly as simple as the periodic, which can be represented as an inner product. It is simple enough, though, to carry out numerical analyses, with the aid of software tools.

2.3.3 A Practical Modification to the Aperiodic Cross-Correlation Function

As can be seen in the previous section, the ACCF is significantly more cumbersome than the PCCF in Section 2.1.1. The main difference is that a convolution sum, instead of an inner product, must be carried out. However, the ACCF is an integral part of the scheme that is being developed in this chapter. In Section 2.3.2, it is assumed that for a signal $u(t)$ only one period of the pseudorandom sequence exists for all time. When it is cross-correlated against a signal $v(t)$ holding to the same constraint, the resulting function has values for only $2N-1$ contiguous discrete times, while the output is zero everywhere else.

Thus, it can be shown that the values found using the ACCF hold for more general sequences, composed of periods of various pseudorandom sequences, as long a proper number of zeros are inserted between each different sequence. Let's define a new signal z , which has the following form:

$$z: \{u_0, \dots, u_{N-1}, 0, \dots, 0, v_0, \dots, v_{N-1}, 0, \dots\}$$

where

u & v : Pseudorandom Sequences of Length N

0 's: $N-1$ between each sequence

Thus, if an ACCF is performed between z and x , the result will be the aperiodic autocorrelation of x directly followed by the ACCF of y with x . Given the proper number of zeros, at least $N-1$ between sequences of length N , the cross-correlation can be decomposed into the cross-correlations of its constituent sequences. A graphical example of this code implementation can be seen in Figure 13:

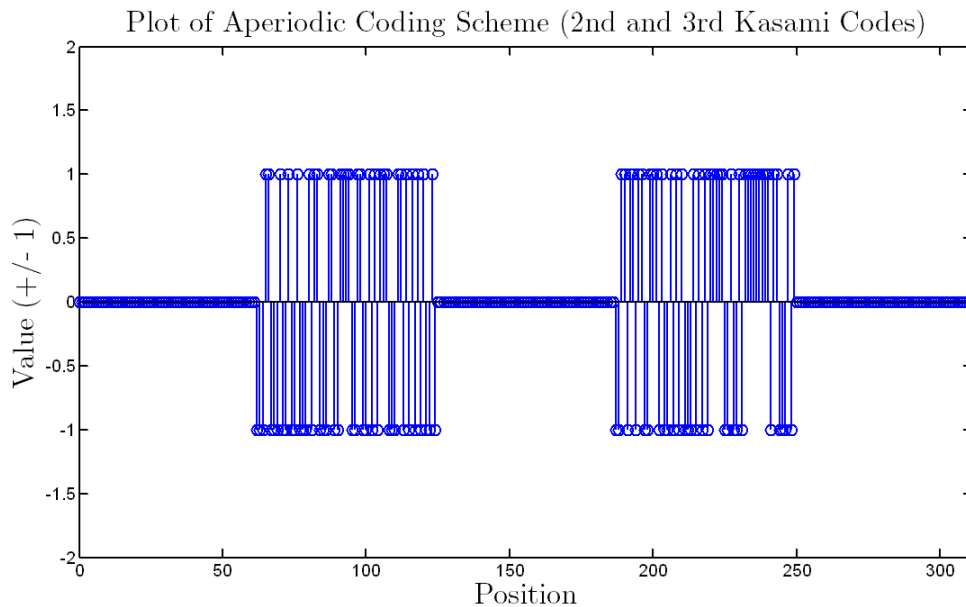


Figure 13: Sequence Composed of the 2nd and 3rd Kasami Codes, Separated by Zero-Spacing

In this figure, the sequences in the figure are the second and third 63-chip length Kasami codes, generated using the m-sequence with feedback taps at 6 and 1 (generation of pseudorandom sequences can be found in Section 2.2). Between the two codes there are exactly 62 chips of zero-spacing. Figure 14 shows the correlation of the “hybrid” sequence against the two base sequences.

Plot of Aperiodic Cross-Correlations of Joint Sequence with Kasami Codes 2 and 3

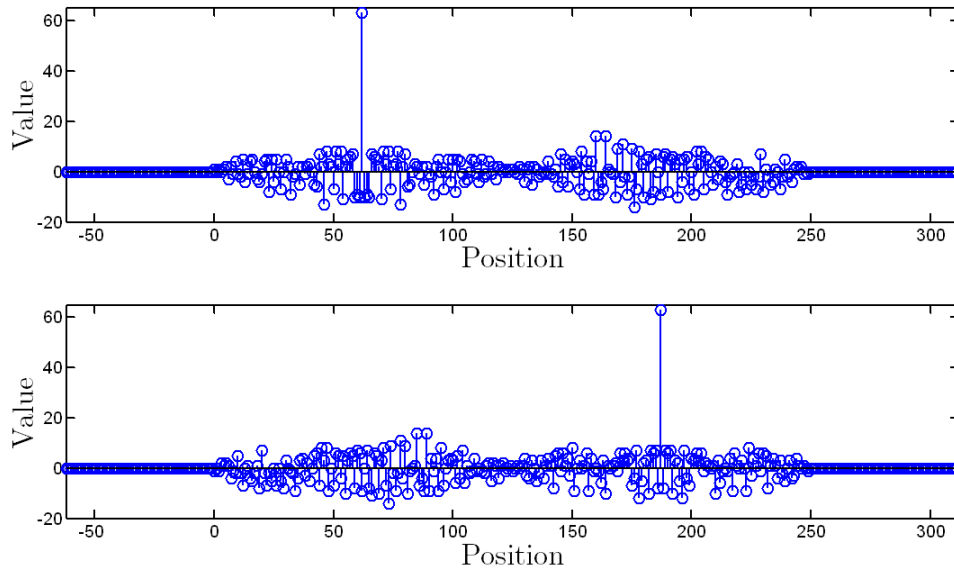


Figure 14: Aperiodic Cross-Correlation of the Joint Sequence with Each of the Constituent Kasami Sequences (Codes 2 and 3)

As can be seen in Figure 14, the autocorrelation of each sequence present in the “hybrid” sequence is identical to the aperiodic autocorrelation of each sequence taken individually (appended to the aperiodic mutual cross-correlation of the two functions). Therefore, the sequences do not interfere with each other in a mathematical sense and various sequences could be used to represent different symbols in a single data transmission. Also, although the sequence no longer adheres strictly to the bounds provided by the periodic cross-

correlation, it is suspected that values close to the periodic bounds may still be achieved and DSSS/MA can still be implemented.

The type of spread spectrum could find its most useful application in backscatter communication. In wireless backscatter communication, there is typically a single reader, which has transmit and receive capabilities, and several remote tags (or nodes) that send data back to the reader, using methods described in Chapter 1. In current backscatter communication, the nodes are only capable of transmitting data in BPSK or OOK. More importantly, multiple nodes are cannot transmit simultaneously. The modified spread spectrum described in this section suits backscatter communication extremely well. Multiple codes could be used in each remote node to represent multiple data symbols. Thus, higher order modulation schemes could be implemented in wireless backscatter through use of “multi-code” nodes. Additionally, since each “symbol” is a spread spectrum sequence, the multiple-access characteristics of direct sequence spread spectrum can be maintained. In this configuration, the receiver could have a priori knowledge of which pseudorandom sequences correspond to specific nodes and, when data comes in, the receiver could distinguish both between individual nodes and the various symbols corresponding to each node. Thus, it would be possible for multiple nodes to transmit data back to the reader simultaneously using higher order modulation schemes.

2.4 Issues with Using Spread Spectrum in Backscatter Communication

Although spread spectrum is widely employed in traditional radio links, such as mobile telephone networks and the Global Positioning System (GPS), its implementation in passive backscatter sensing faces unique problems. First, since the remote node does not have a transmitter and relies solely on the backscattered waveform, there is currently no

way to perform power control on the node side. Thus, the system becomes extremely susceptible to the near-far problem, which can plague spread spectrum multiple-access. Next, communication systems generally send continuous data streams and use cyclic spread spectrum to modulate the data. However, since spread spectrum can increase the power demands on communication, it becomes more difficult to send long streams of spread spectrum-modulated data.

2.4.1 The Near-Far Problem

The Near-Far problem in DSSS/MA is one of the major weaknesses and a problem that must be overcome, as is discussed in [21]. In practice, wireless nodes are never located at exactly the same distance from a receiver, resulting in a range of different powers seen at the receiver. In traditional two-way line-of-sight communication, power seen at the receiver is governed by the standard free-space link budget equation found in [22]:

$$P_R = \frac{P_T G_T G_R}{(4\pi r / \lambda)^2} \quad (10)$$

where

P_R : Received Power

P_T : Transmitted Power

G_T : Gain of Transmit
Antenna

G_R : Gain of Receive
Antenna

R : Path Distance

λ : Carrier Wavelength

It should be noted that this formulation does not account for fading phenomena that might exist in a typical environment. The theory behind multiple user detection presupposes that all of the users have equal power. In a traditional two-way link, the path loss falls off only by a factor of r^2 and the base receiver can coordinate the powers of the remote users

to ensure that the received power from all users is approximately equal. Thus, the various locations of the users are not a large factor and the near-far problem does not degrade the link greatly. However, this problem is not so easily solved in the case of a backscatter link. As was shown by Griffin in [23], a backscatter communication link is governed by a modified link budget equation:

$$P_R = \frac{P_T G_{TR}^2 G_t \lambda^4 X^2 M}{(4\pi r)^4 \Theta^2 B^2 F_2} \quad (11)$$

where

P_R, P_T, λ, r : Same as in equation ()

G_{TR} : Gain of Transmit/ Receive Antenna

G_t : Gain of Tag Antenna

X : Polarization Mismatch

M : Modulation Factor

Θ : On Object Antenna Gain Penalty

B : Path Blockage

F_2 : Fade Margin

This equation, unlike that for free space path-loss, accounts for all of the factors found in an everyday environment and X, M, Θ, B and F_2 are the “gain penalties.” What is most important to recognize from the backscatter link budget equation is that received power falls off by a factor of r^4 for monostatic (single antenna for send and receive) backscatter communication. Also, at this point in time, there is no method for controlling the power reflected by the remote node. Thus, the received power is extremely dependent upon the various backscatter nodes and the near-far problem will affect the ability to perform DSSS/MA.

Although the near-far problem strongly affects communication, several methods can be employed to minimize the degradation and equalize received power. First, if the sensor

setup is stationary, such as it is in structural monitoring, the antenna orientations of the various nodes can be adjusted to create polarization mismatch and equalize power. Thus, the signal seen at the receiver would be composed of signals transmitted by various nodes and weighted by polarization-mismatch, as seen in equation (12):

$$r(t) = \sum_{i=0}^n \{X_i [x_i(t) * h_{fi}(t) * h_{bi}(t)] + n_i(t)\} \quad (12)$$

where

- $r(t)$: Received Passband Signal
- X_i : Polarization mismatch
- $x_i(t)$: Passband Data Signal
- $h_{fi}(t)$: Channel response of forward link
- $h_{bi}(t)$: Channel response of backscatter link
- $n_i(t)$: System Noise

Next, as was noted by Moshavi in [24] for two-way communication, iterative multiple-access interference (MAI) cancellation is another method that can be employed to mitigate the deleterious effects of overly strong users. In iterative MAI cancellation, the system detects and subtracts out the strongest nodes, so that the weaker signals can be received. This can be an effective tool for balancing signals received from multiple nodes, but it must be noted that this algorithm can be computationally intensive and can handle only a limited number of nodes simultaneously. Employing DSSS/MA in a backscatter link is more difficult than in a traditional two-way wireless link, but the near-far effect can still be mitigated in a backscatter environment.

2.4.2 Power Requirements of Spread Spectrum

The Near-Far Problem in spread spectrum is the most obvious issue making the application of DSSS/MA in backscatter sensor networks difficult. Another more subtle

issue must be addressed as well, in order to implement spread spectrum successfully in a passively powered backscatter link: the increased power demands of spread spectrum.

In a traditional backscatter link, the tag uses either BPSK or OOK to encode the data signal. With this scheme, the microchip on the tag only must clock at around the speed of the actual data rate. However, in a spread spectrum link, the microcontroller must clock at least at the chip rate, which is the data rate multiplied by the pseudorandom sequence length. With this increased clock speed, the microcontroller may not be able to operate at its most efficient state and it is questionable whether the tag could harvest enough power. In [25], Rohatgi showed that spread spectrum communication in a backscatter link was possible. However, in his experiments the tags were powered locally, so no measure of increased power consumption was considered.

Although increased power consumption may seem like a problem at first glance, it can be improved with clever engineering. A team of scientists from the Institute of Microelectronics at Tsinghua University in China have shown that spread spectrum could be implemented in RFID in [26]. This finding is great first step and shows that improvement in technology is allowing more complex coding in passive RFID. However, it has not been applied to a backscatter sensor platform, which is inherently more complex and requires more power.

Power demands are important to consider in passive backscatter links, but the benefits of the aperiodic spread spectrum scheme stated in Section 2.3.3 outweigh the power costs in many ways. First, this scheme still employs spread spectrum, albeit a modified version, and, therefore, increases the robustness of the radio link against interference and fading in addition to allowing multiple nodes to transmit simultaneously. Also, unlike traditional spread spectrum in a backscatter link, this scheme allows higher order modulation, which means data can be sent more efficiently and the power consumption is less than that of traditional spread spectrum. Additionally, if there are periods of zero-time (i.e. time

between symbols where no transmission takes place) between each symbol instead of each packet, the charge pump could use smaller capacitors to store energy, since the capacitors are charged more frequently and would not be required to sustain a charge through the entire packet. Therefore, space could be saved on the RFIC, which leads directly to cost savings in production. Thus, although power demands go up with spread spectrum implementations, the benefits of using spread spectrum, especially the aperiodic scheme stated in Section 2.3.3, outweigh the associated costs.

CHAPTER 3: DEVELOPMENT OF APERIODIC CODE SETS

As was shown in the previous sections, periodic pseudorandom codes used in DSSS/MA adhere to strict cross-correlation bounds. Thus, in these systems phase rotations between the codes make no appreciable difference in the ability for a receiver to decode multiple users. However, when multiple codes are used by a single node to represent different symbols, while maintaining multiple access characteristics, the periodic cross-correlation function is no longer useful. Therefore, the aperiodic cross-correlation function must receive greater attention and the aperiodic cross-correlation characteristics of the sequences need to be determined. In this section, the theory behind finding aperiodic code sets will be explained and the actual test plan for finding these codes will be set forth.

3.1 Characteristics Guiding Code Selection

In periodic spread spectrum, code sets such as Gold sequences and the small set of Kasami sequences have mathematically determined boundaries on correlations. However, when the periodicity is thrown away, the boundaries become malleable and dependent upon several factors. First, the amount of zero-length (L_z) between various codes, i.e. the time a node is transmitting nothing, can affect the cross-correlations in the system. Second, the codes used in the system do not necessarily adhere to strict cross-correlation bounds and, therefore, the specific codes (c_i) in the system can make a difference. Finally, the rotation (r_i) of the codes present in the system makes a difference. Thus, for effective code generation, these three additional parameters must be considered in order to produce a set of codes with low cross-correlation properties.

3.1.1 Zero Length

As was stated above, there are the three issues that must be addressed when finding a suitable set of codes for this scheme. In order to find more the most general solution, it will be assumed that:

$$l_z \geq L - 1$$

where

l_z : zero-length

L : length of pseudorandom code

Choosing this value for zero-length ensures that the correlation values correspond to the values found using the aperiodic cross-correlation function. Thus, the only other factors that are important in determining code combinations are the codes used in the system and the relative rotations of the codes.

3.1.2 Codes Included in Group

Choosing the codes that work well together and minimize mutual cross-correlation is an extremely important part of the simulation. However, it is important to determine both the total number of possible combinations and what these combinations actually are. In most cases, there will be more codes in the base set of pseudorandom codes than can actually be used in the modulation scheme.

Take for example the set of length 63 (63-chip) Kasami sequences, which has eight total sequences and the case where there are three transmitting nodes with two codes being used per node. In this situation, there are eight codes in the base set and only six codes used in the system and it may not be clear which six of the eight codes should be used. The problem can be most easily solved by looking at the configuration as a 3x2 matrix:

$$Set = \begin{pmatrix} code_{11} & code_{12} \\ code_{21} & code_{22} \\ code_{31} & code_{32} \end{pmatrix}$$

Here, each row represents a different node and multiple codes representing multiple symbols per node. In this case, there are three rows and two columns, meaning there are three nodes with two codes being used per node. However, as stated before there are eight codes total and a series of combinatorial relationships show how many total code combinations are possible. From basic statistics, a combination can be found using the following formula:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \quad (13)$$

where

n: total number of objects
in a set

k: number of objects
chosen from the set

For each row, two codes must be chosen from the set of available codes. For rows 1 to 3, the available combinations are:

$$\binom{8}{2} = 28, \binom{6}{2} = 15 \text{ and } \binom{4}{2} = 6, \text{ respectively}$$

However, row ordering is unimportant in this scheme and must be normalized out (with a factor equal to $r!$, where r is the total number of rows). Thus, the total number of possible combinations in this case is:

$$\frac{28 \times 15 \times 6}{3!} = 420$$

In general, this process can be generalized to any number of nodes and any number of codes and the formula is:

$$combinations = \frac{\prod_{i=0}^{N-1} \binom{C - i \times k}{k}}{N!} \quad (14)$$

where

C: Total number of codes in base set

k: Number of codes per node

N: Total number of nodes

For small base code sets, the number of combinations stays reasonable, on the order of $\sim 10^3$. However, when choosing only a small number of codes from a large base, this number can explode. For example, if there is a base of 16 codes and there are three nodes with four codes used per node, the total number of combinations is over 10 million! In order for the simulation to be manageable, the number codes in the base set has been limited to twelve, since a large number of computations must be completed for every combination.

3.1.3 Code Rotations Found in System

As was shown in the previous section, the number of possible code combinations can be extremely large, depending on the size of the base code set and the number of codes used in the system. However, an even more difficult and arguably more important determination is the exact rotation of the code set. A code rotation is a cyclic shift of an aperiodic pseudorandom sequence and an example of two different code rotations of a sequence (the 2nd code from the 63-chip Kasami set) can be found in Figure 15.

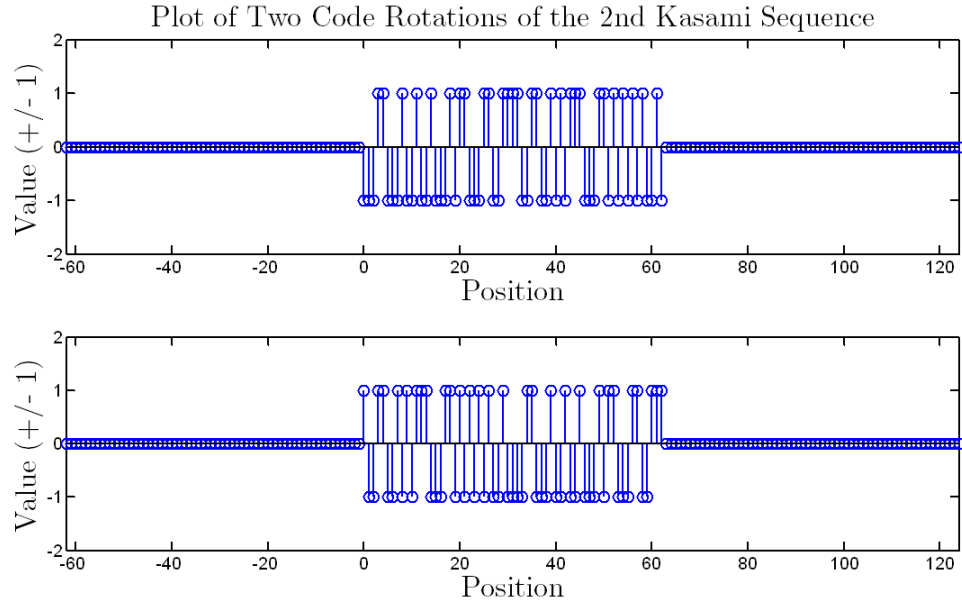


Figure 15: Plot of Two Code Rotations of the 2nd Kasami Sequence (Rotations 0 and 32)

In [27], the authors try to simplify spread spectrum by modeling equal power DSSS transmitters operating in the same frequency band as interferers and using equation (15) to model the scenario:

$$\frac{E_b}{N_j} = \frac{PG}{J/S} \quad (15)$$

where

E_b/N_j : Signal-to-Jammer
Ratio

PG : Code Length

J : Total Interference

Power

S : Signal Power of Interest

However, as was shown in previous sections, pseudorandom codes, especially in the asynchronous DSSS/MA, are never truly orthogonal. Thus, partial correlation and cross-correlation peaks vary depending on the set of pseudorandom codes chosen and, in the case of aperiodic spread spectrum, on the phase rotation of the individual codes present in the system. In DSSS/MA, the partial correlation from the code of interest can add upon the cross-correlation peaks of the other transmitting nodes, making autocorrelation peaks indistinguishable from large interference peaks. Therefore, the partial correlations, as well as the cross-correlation peaks, should be minimized in order to ensure detection can take place.

In the previous section, the example of a 3 node, 2 code-per-node (3x2) system was used to demonstrate how different combinations of codes can be used. Again, the (3x2) system is a good way to show how certain phase rotations should be chosen, given predefined codes in the set. As before, the system can be portrayed as a 3x2 matrix:

$$test = \begin{pmatrix} code_{11} & code_{12} \\ code_{21} & code_{22} \\ code_{31} & code_{32} \end{pmatrix}$$

First, the autocorrelation of each code should be calculated and the strongest partial correlation peak should be determined. In an aperiodic spread spectrum system, the value of this partial correlation peak can vary drastically based upon code rotation, since correlation bounds no longer hold.

After determining the strongest partial correlations, the cross-correlation peaks should be analyzed. In this set, however, only cross-correlations between codes in different nodes are important, since codes in the same node will never interfere with one another. In this case, the cross-correlation functions of interest are located in Table 1.

Table 1: Relevant cross-correlations for 3x2 sequence configuration

$C(\text{code}_{11}, \text{code}_{21})$	$C(\text{code}_{12}, \text{code}_{21})$	$C(\text{code}_{21}, \text{code}_{31})$
$C(\text{code}_{11}, \text{code}_{22})$	$C(\text{code}_{12}, \text{code}_{22})$	$C(\text{code}_{21}, \text{code}_{32})$
$C(\text{code}_{11}, \text{code}_{31})$	$C(\text{code}_{12}, \text{code}_{31})$	$C(\text{code}_{22}, \text{code}_{31})$
$C(\text{code}_{11}, \text{code}_{32})$	$C(\text{code}_{12}, \text{code}_{32})$	$C(\text{code}_{22}, \text{code}_{32})$

In the table above, the reciprocal cross-correlations, such as $C(\text{code}_{21}, \text{code}_{11})$, which is the reciprocal of to $C(\text{code}_{11}, \text{code}_{21})$, are not included because the pseudorandom signals present in the system are real and have the following relationship:

$$\max \|C(\text{code}_{ij}, \text{code}_{kl})\| = \max \|C(\text{code}_{kl}, \text{code}_{ij})\| \quad (16)$$

Thus, in order to ensure that the signal can be detected, the rotations of the codes that minimize all of the partial correlations, as well as, the cross-correlations in Table 1 must be found. If this problem were extended to the $M \times N$ case, the minimization formula would have the following form:

$$\min_{\text{rotations}} \left\{ \sum_{a=1}^M \sum_{b=1}^N \max_2 \|C(l)(\text{code}_{ab}, \text{code}_{ab})\|^2 + \sum_{i=1}^{M-1} \sum_{j=1}^N \sum_{k=i+1}^M \sum_{l=1}^N \max \|C(l)(\text{code}_{ij}, \text{code}_{kl})\|^2 \right\} \quad (17)$$

where

M : Number of Nodes

N : Number of Codes per Node

\max_2 : Second Highest Value in Function

$C(l)$: Aperiodic cross-correlation function

In equation (17), the intent is to minimize a massive sum of correlations. This sum can be broken into two distinct parts: the sum of the largest partial correlations and the sum of the maximum magnitude cross-correlations between interfering sequences.

As can be seen in equation (17), there could potentially be an extremely large computational time. For example, if there were 12 different codes being used by nodes and each code has a length of 255 chips, then there are potentially $255^{12} = \sim 7.6 \times 10^{28}$

different rotation combinations. This number of rotations is nearly impossible to process and there are diminishing returns as computational complexity increases.

3.2 Test Plan

As can be seen in the general theory development, there are no general mathematic equations governing the aperiodic correlation characteristics of pseudorandom sequences, such as Kasami or Gold codes. Therefore, the correlation properties can be found through computational methods alone, not theoretical mathematics. Thus, well-defined test parameters must be declared at the outset to ensure valuable information is obtained in the end.

3.2.1 Code Configurations to be Studied

In traditional communications systems, this type of coding scheme really is not necessary, since higher order modulation schemes exist and computation ability is not as limited as in the backscatter case. Thus, this scheme could find its primary applicability in backscatter sensors and RFID, due to limitations on both modulation schemes and computational capability. In these systems, however, there is currently no way to control power at the node. Although antenna polarization tuning and iterative processing can be used to alleviate the issue, this type of DSSS/MA can only really be used in up to a few nodes at one time. Therefore, only a limited number of cases will be studied and the cases are found in Table 2.

Table 2: Various Sequence Configurations

# of nodes	# of codes (symbols) per node
1	1, 2 or 4
2	1, 2 or 4
3	1 or 2 (if 8 codes in set); 1, 2 or 4 (if 12 codes in set)

Restricting the configuration to have a maximum of three nodes with four codes per node provides for realistic implementations, given the environment. Also, given how quickly the combinatorial relationship in formula (14) can explode, it is important to limit the number of codes present in the system, making computation times reasonable.

3.2.2 Specific Codes under Test

As was discussed in the development of sequence sets, an extremely large number of codes could be generated depending upon length of the sequences. Also, for Kasami and Gold sequences, different sets can be generated depending on which m-sequences are used to produce the set. Thus, the codes under test will be limited to sets listed in Table 3.

Table 3: Codes under Test

Code Set	Code Length (chips)	m-sequences (taps)	# of codes
Kasami (small)	63	[6 1]	8
Gold	127	[7 1], [7 3]	12
Kasami (small)	255	[8 4 3 2]	12

As can be seen in Table 3, three different code lengths are being tested, the maximum of which is 255 chips. As was stated in Section 2.4, the near-far Problem greatly affects the ability to use DSSS/MA over a large number of nodes, due to power control difficulties. Additionally, longer code lengths generally mean higher chip rates, since $chip\ rate = data\ rate * code\ length$. It was also stated in Section 2.4 that high chip rates increase power consumption on the node end, thus hurting a passive node's ability to communicate. Therefore, code lengths no greater than 255 chips are being tested since they can allow DSSS/MA among a limited number of nodes, while minimizing the additional power draw on the passive node.

For 63-chip sequences, it was decided that the small set of Kasami sequences should be used, since they provide the lowest mutual cross-correlation characteristics. This code set is produced according to the method provided in Section 2.2.2.2, using the m-sequence produced by a shift register with feedback taps stated in Table 3. Since only eight codes reside in the set, all of them will be tested.

For the 127-chip sequences, Gold sequences were chosen since there were no Kasami sequences at this period length. The set of codes is produced according to the methods stated in Section 2.2.2.1, using the preferred pair of m-sequences produced by shift registers with feedback taps as stated in Table 3. In order to minimize the number of codes under test, only the first 12 codes produced using this method are included in the simulations. Other codes could clearly be used but, regardless, a subset for the simulation must be chosen. If all 129 codes were included and the configurations discussed in Section 3.2.1 were employed, there could be upwards of 1×10^{20} code combinations.

Finally, for the 255-chip sequences, the small set of Kasami sequences was once again chosen, for the same reasons given for the 63-chip Kasami sequences. Also, only the first 12 sequences of the 16 sequence set were considered. In this case, there are over 10 million combinations. Although this number is far smaller than 10^{20} , it still is almost not computable, given that different mutual rotations between codes in the set must also be analyzed.

Overall, code choice is important but, in order to keep the simulation reasonable, considering the limited computational power available, these specific codes were chosen. A list of all the codes under test (in their base rotations) can be found in Appendix A.

3.2.3 Code Rotations Tested in the Systems

After determining what code configurations and what base code sets should be tested, it is important to figure out what code rotations should be tested when trying to find an optimal set of codes that work together for a certain code configuration. This can be a daunting task because, as was shown in Section 3.1.3, there are upwards of 10^{28} mutual code rotations that must be considered when finding an optimal code set.

3.2.3.1 *Minimizing Partial Correlation*

When rationally deciding a way to limit the number of code rotations that will be tested, it is useful to analyze the base equation (Equation (17)) that is employed to find codes and rotations that minimize cross-correlation. In the ideal scenario, this equation minimizes the summation of the strongest partial correlation of each sequence present in the configuration and the maximum magnitude cross-correlation between interfering signals. The strongest partial correlation is the second highest magnitude value produced by the cross-correlation of a sequence with itself, i.e. an autocorrelation. Interfering signals are signals that operate in the same time/frequency space. Thus, codes that are used by a single node to represent multiple symbols never interfere with one another and their cross-correlations are not considered by the minimization equation.

Therefore, based on this problem description, the strongest partial correlations of codes in a configuration are always considered, independent of actual code configuration and use among different nodes. Meanwhile, only cross-correlations between codes used by different nodes are considered by the minimization formula and are dependent upon code configuration. Since the partial correlations have a more constant characteristic with changing configuration, the code rotations are ranked based upon their strongest partial correlation peaks, the lowest being ranked first and the highest last.

3.2.3.2 *Standardizing Computation*

After rank ordering the rotations of each code based upon partial correlation strength, the number of rotation combinations must be specifically determined. As was stated in Section 3.2.1, there are a number of different code combinations, ranging from one code used by a single node (1 total code) to four codes used by each of three nodes (12 total codes). In order to compromise between computational complexity and uniformity among all cases, a total of ~4000 combinations were analyzed per case. This means that the number of different rotations of each code that could be analyzed was limited based upon total number of codes present in the system. For example, in a one node, two code-per-node system, only two codes are present and at least 64 rotations of each code can be analyzed with respect to the other, leading to around 4000 different rotation combinations. However, in the three node, four-code setup, there are 12 codes present and analyzing two rotations of each code with respect to all of the others leads to around 4000 different combinations. Thus, the number of rotations that can be analyzed is heavily dependent upon the number of codes present in the system since *total rotation combinations* = $(code\ length)^N$, where N is the number of codes present in the system.

Because the number of rotations that can be analyzed may be limited, two paths forward were possible: the same number of rotations for each code could be analyzed or some codes could be held to a single rotation while others could be varied others more widely. In order to address this point, two test plans were laid out.

3.2.3.2.1 *Same Number of Rotations Per Code (Primary Test)*

As was stated, a predefined total number of rotation combinations was defined for simulation. Thus, rotations per code were approximately equal to $(4000)^{1/N}$, where N is

the total number of codes present in the system. Table 4 shows how many rotations were analyzed per code using this test plan.

Table 4: Number of Code Rotations in Analysis (Primary)

# of Codes	# of Rotations*
1	4096
2	64
3	16
4	8
6	4
8	3
12	2

**Perform the lesser of (# of Rotations) and (# of Codes)^(code length)*

As can be seen in the table, the number of code rotations that can be analyzed per code decreases drastically, based upon the number of codes present in the system. The number of rotations dictates how many code rotations can be analyzed per code. The rotations with lowest partial correlation, as stated in Section 3.2.3.1, are chosen to be analyzed.

3.2.3.2.2 Varying Number of Rotations per Code (Secondary Test)

In the previous test plan, no codes are given preference and the same number of rotations is analyzed for every code. However, in the secondary scheme, half of the codes are held to the rotations that provide each code's lowest partial correlation peak, while a much larger number of rotations can be tested for the other codes.

In the specific scheme under consideration, all of the codes in a specific configuration under test are ordered according to ascending code number:

$$code\ group = \{c_1, c_2, \dots, c_n\}$$

They are then broken into 2 groups, where the 1st group consists of all codes at odd positions in the list and the 2nd group consists of codes at even positions in the list. Then, two parts of the simulation are carried out. In part 1, the codes in the 1st group are held to the rotation that minimizes partial correlation peaks, while the second group is allowed to vary. In part 2, the codes in 2nd group are held to their lowest partial correlation rotations, while the codes in the first group are allowed to vary. Thus, the number of code rotations for the codes that are allowed to vary can be found in Table 5.

Table 5: Number of Code Rotations in Analysis (Secondary)

# of Codes	# of Rotations*
1	N/A
2	N/A
3	64
4	64
6	16
8	9
12	4

**Perform the lesser of (# of Rotations) and (# of Codes)^(code length)*

As can be seen by comparing Tables 4 and 5, when the number of codes that are allowed to vary decreases, the number of rotations that can be analyzed for the other nodes increases substantially.

3.2.3.3 Analyses

After setting up the simulation according to the parameters defined in the previous sections, the code configuration that minimizes cross-correlation and partial correlation will be stated for the various cases and code lengths. A comparison will then be made between the different test methods. The more effective method for choosing code sets will be stated and founded upon the data at hand.

CHAPTER 4: ANALYSIS OF SIMULATION RESULTS

As was discussed in the previous chapter, aperiodic spread spectrum may be a way to introduce higher order modulation schemes and multiple access communication simultaneously in a backscatter communication link. In order to do this properly, the optimal combination of spread spectrum sequences, as well as optimal rotations of the sequences, must be determined. The optimal rotations are the rotations that return the “lowest composite correlation” value, as determined by equation (17) in Section 3.1.3. However, depending upon the number of sequences actually utilized in the system, their length and the size of the set from which codes may be chosen, a full search may be impossible. Therefore, the tests proposed in 3.2.3.2 were carried out and the optimal result of the various tests will be the recommended code selection/code rotation for use in the specific configuration (the list of configurations can be found in Table 2 of Section 3.2.1). Also, the results for all code lengths will be examined and the various tests will be compared, in order to see if there is a clear decision concerning which method provided the lowest-correlation results.

As was stated in the previous chapter, the codes that will be tested are the 63-chip Kasami, 127-chip Gold and 255-chip Kasami sequences. Once again, these codes can be found in Appendix A in initial phases (phase rotation = 0). The aperiodic configurations that are of interest can be found in Table 6.

Table 6: Code configurations for 63-Chip Kasami, 127-Chip Gold and 255-Chip Kasami Codes

Case	Configuration
1	1 node, 1 code per node
2	1 node, 2 codes per node
3	1 node, 4 codes per node
4	2 nodes, 1 code per node
5	2 nodes, 2 codes per node
6	2 nodes, 4 codes per node
7	3 nodes, 1 code per node
8	3 nodes, 2 codes per node
9	3 nodes, 4 codes per node

In Table 6, all of the configurations will be possible for 127-chip Gold codes and 255-chip Kasami codes, since there is a starting pool of 12 codes and these configurations require a maximum of 12 codes. However, there are only 8 codes in the 63-chip Kasami sequence set and, therefore, case 9 is impossible to test, since 12 codes would be required.

4.1 Overall Analyses to be Performed

In the following sections, the data will be presented, organized and analyzed based upon the number of nodes in the configuration. Discussions of results for the three different code lengths will be provided (63-chip, 127-chip and 255-chip). The specific code numbers and the rotations, which are clockwise cyclic rotations of the base codes, are provided. The code numbers and base codes for the different code lengths may be found in Tables 18, 19 and 20 of Appendix A. In these tables, the codes are written in their binary representations for ease of portrayal. In the simulations, these codes used their antipodal representations (1 and -1 instead of 1 and 0) to produce more accurate, real-world results. It is important to note that there may not be one single, optimal code group, but several that come out with equal weights. If this is the case, only one suggested code

configuration will be provided, but it will be noted that other configurations may provide similar results.

In order to find a baseline value against which to judge the aperiodic scheme, the results will be compared with periodic spread spectrum. For each configuration, equation (17) in Section 3.1.3 adds together the squared magnitude of the strongest partial correlation for each sequence in the system and the largest mutual cross-correlation peaks. The number of partial correlations and cross-correlations considered by the equation varies based on configuration and these values can be found in Table 7.

Table 7: Number of Correlations Considered for Each Code Configuration

Configuration	# Partial Correlations	# Cross-Correlations	# Total Correlations
1 Node, 1 Code	1	0	1
1 Node, 2 Codes	2	0	2
1 Node, 4 Codes	4	0	4
2 Nodes, 1 Code	2	1	3
2 Nodes, 2 Codes	4	4	8
2 Nodes, 4 Codes	8	16	24
3 Nodes, 1 Code	3	3	6
3 Nodes, 2 Codes	6	12	18
3 Nodes, 4 Codes	12	48	60

Since equation (17) produces an overall sum of the squares of correlations (which varies based upon configuration), an average correlation magnitude for the specific configuration will be found using the following formula:

$$m = \sqrt{\frac{Sum}{N}} \quad (18)$$

where

m: Average Correlation

Magnitude

N: Total # of Correlations

Sum: From Eq. (17)

Value m produced by equation (18) will be compared with the autocorrelation peaks, which are 63, 127 and 255 for the 63-chip Kasami, 127-chip Gold and 255-chip Kasami, respectively. They will also be compared with the periodic correlation bounds, which are 9, 17 and 17 for the 63-chip Kasami, 127-chip Gold and 255-chip Kasami, respectively. These bound values are listed by Sarwate and Pursley in [19]. It will be determined whether the correlation values still allow multiple access communication to take place.

4.2 Single Node System

Unlike the other scenarios, which have 2 or 3 nodes present, this is a single node system and no multiple-access interferers are present. Thus, the optimization attempts to minimize the strongest partial correlations of the sequence in the system. Also, because there are no cross-correlation dependencies, the optimal code rotations occur when all of the strongest partial correlations are minimized for the system. As a side note, since cross-correlations are not considered for this situation, the secondary test will provide the same result as the primary test and, therefore, is unnecessary.

4.2.1 One Node, One Code

In this scenario, there is only a single code present in the system. Since there is only one node present, there are no multiple access interferers and the code choice that provides the lowest correlation value for the system is the code that minimizes partial correlation peaks. After running through the simulation for each code length, the optimal codes and code rotations were found and listed in Table 8.

Table 8: Code Selection and Rotation for One Node with One Code used per Node

Code Type/Length	Node	Codes	Rotations
		1 st	1 st
Kasami – 63	1 st	1	0
Gold – 127	1 st	7	97
Kasami – 255	1 st	1	1

For each code length, the algorithm must only seek out one code and, since there are no interferers, it bases the decision solely on the minimization of correlation peaks. Therefore, an exhaustive search could be carried out and the absolute codes that minimized partial correlation and rotations could be determined. For the 127-chip Gold sequence, there was a three-way tie for different combinations and only one was listed.

For the 63-chip Kasami sequence, an average correlation value of 6 was found and this is actually lower than the bound value of 9. However, this is not contradictory because this code is an m-sequence, as well as, a Kasami sequence and m-sequences have partial correlation values that are slightly better than the bounds guaranteed by Kasami codes. Since this value is so low and there are no multiple access interferers, detection can take place.

For the 127-chip Gold sequence, the average correlation value of 16 was found, which is lower than the expected periodic bound of 17. This result is not contradictory because it is still less than the bound value, which only provides an upper level that cannot be crossed. This value is significantly lower than the autocorrelation peak, so detection can take place.

For the 255-chip Kasami sequence, an average correlation value of 13 was found and this is actually lower than the bound value of 17. Again, this code is an m-sequence, as well as, a Kasami sequence and m-sequences have good partial correlation values, so the result was expected. This value is extremely low in comparison to the autocorrelation spike, so detection can take place.

4.2.2 One Node, Two Codes

The simulation, like the simulation in Section 4.1.1, is searching for the codes and rotations that minimize the partial correlations. However, in this case, it searches for the two codes that minimize the strongest partial correlation for each. The simulations were carried out and the results were placed in Table 9.

Table 9: Code Selection and Rotation for One Node with Two Codes used per Node

Code Type/Length	Node	Codes		Rotations	
		1 st	2 nd	1 st	2 nd
Kasami – 63	1 st	1	4	0	33
Gold – 127	1 st	7	8	97	81
Kasami – 255	1 st	1	2	1	209

Again, an exhaustive search could be carried out but, in this case, there were multiple code rotations for the 127-chip and 255-chip cases that provided “optimal” results. For the 127-chip length Gold codes, 3 code/rotation combinations provided the same partial correlation values while 2 code/rotation combinations provided the same minimization for the 255-chip length Kasami codes. Therefore, only one rotation is listed for each.

For the 63-chip Kasami sequence, the average correlation of the code combination was found to be 7.65, which is still lower than the periodic correlation bound, but not as much as in the single code case. The partial correlation from the other Kasami sequence is offsetting the ideality of the m-sequence. However, the average value is still far less than the autocorrelation value of 63, so detection can still take place.

For the 127-chip Gold sequence, the average correlation of the lowest-correlation group was found to be 16, which is still lower than the periodic correlation bound, but not as much as in the single code case. Since the average value is still far less than the autocorrelation value of 127, detection can take place without any issues.

For the 255-chip Kasami sequence, the average correlation of the group was found to be 18.07, which is actually greater the periodic correlation bound. However, the difference between this value and the periodic correlation bound is less than 0.5% of the autocorrelation peak. Therefore, since this difference is negligible and the average value is still far less than the autocorrelation value of 255, detection can still take place.

4.2.3 One Node, Four Codes

Finally, the simulation was carried out for the three codes lengths for four codes in the system. Due to the lack of multiple-access interference, the algorithm tries to minimize the strongest partial correlations of four functions. The codes/rotation that minimized equation (17) can be found in Table 10.

Table 10: Code Selection and Rotation for One Node with Four Codes used per Node

Code Type/Length	Node	Codes				Rotations			
		1 st	2 nd	3 rd	4 th	1 st	2 nd	3 rd	4 th
K – 63	1 st	1	2	3	4	0	15	3	33
G – 127	1 st	1	7	8	11	55	97	81	45
K – 255	1 st	1	2	3	7	1	209	33	25

Finally, since this case was based solely upon partial correlation minimization, an exhaustive search could be carried out. Here, there were many code combinations that could provide low-correlation results. For the 63-chip Kasami sequences, there were at least 10 code/rotation combinations that provided the same minimization. For both the 127-chip Gold and 255-chip Kasami codes, there were at least 6 combinations that provided low-correlation results. For this case, as well as the previous two cases, the minimization algorithm only used partial correlations to find the optimal combinations, leaving the problem fairly unconstrained and allowing many possible solutions. However, in the two and three node case, interferers are introduced and the algorithm must account

for the mutual cross-correlations as well. Therefore, the problem becomes much more constrained and fewer solutions exist.

In this configuration, the average correlation value of the 63-chip Kasami configuration was found to be 8.9, which is right at the periodic correlation value for this sequence type. Since this value is right at the bound and there are no multiple access interferers, detection should take place without incident. For the 127-chip Gold configuration, the average correlation value is 16.26, which is right below the periodic correlation bound for 127-chip Gold sequences. As with the 63-chip Kasami configuration, detection of the Gold sequences should not be an issue. Finally, the average correlation value for the 255-chip Kasami configuration is 20.4, which is 3.4 higher than the expected periodic bound. However, this difference is less than 1.5% of the auto correlation peak and, since there are no multiple access interferers, detection should occur without problems.

4.3 Two Node System

In Section 4.1 the single node configurations were analyzed and the analysis was extremely straight forward. There were no multiple access interferers so only partial correlations, not mutual cross-correlations, had to be considered by the algorithm. Also, since the cross-correlations did not need to be considered, far fewer rotations needed to be analyzed and exhaustive searches could be executed. However, equation (17) becomes much more complicated with the addition of multiple access interferers. Thus, several simulations must be carried out (as was stated in Sections 3.2.3.2.1 and 3.2.3.2.2). After the results are obtained from the tests, they will be combined and overall rotations that minimize partial and cross-correlations will be chosen from the outputs of all the simulations.

4.3.1 Two Nodes, One Code

In this case, two codes were present in the system but, unlike in Section 4.1.2, they now see each other as multiple access interferers. Thus, according to equation (17), the mutual cross-correlation, as well as the partial correlations in the autocorrelation function, must be considered. Here, there are only 2 codes present and, according to Table 4 in Section 3.2.3.2.1, 64 rotations for each code can be considered. Since the number of rotations is so great, only the test in 3.2.3.2.1, not 3.2.3.2.2, was performed. The 64 rotations that were considered for each code were the 64 rank ordered based upon partial correlation minimization, as is explained in Section 3.2.3.1. The code combinations/rotations that provided the minimization of equation (17) can be found in Table 11.

Table 11: Code Selection and Rotation for Two Nodes with One Code used per Node

Code Type/Length	Node	Codes	Rotations
		1 st	1 st
Kasami – 63	1 st	1	2
	2 nd	3	3
Gold – 127	1 st	1	55
	2 nd	8	34
Kasami – 255	1 st	1	185
	2 nd	12	104

For the 63-chip and 255-chip Kasami sequences, the selection algorithm returned one lowest-correlation code combination per code length. However, for the 127-chip Gold sequence, there were two “optimal” code combinations, which could minimize equation (17).

For the 63-chip Kasami, 127-chip Gold and 255-chip Kasami sequences, the average correlation values were found to be 9.26, 17.34 and 21.2, respectively. These values are greater than their respective periodic correlation bounds by 0.26, 0.34 and 4.2. These

differences are about 0.5%, 0.27% and 1.65% of their respective autocorrelation peaks and are still extremely low values. Therefore, since there are still only two interferers in the system and the deviation from the periodic bounds is relatively low, multiple access communication can still take place.

4.3.2 Two Nodes, Two Codes

For this scenario, this is the first time both tests (Sections 3.2.3.2.1 and 3.2.3.2.2) were performed. Because there are two nodes with two codes being used per node, the four maximum partial correlation (from the autocorrelation of each code in the set) as well as four cross-correlations must be considered. The results can be found in Table 12.

Table 12: Code Selection and Rotation for Two Nodes with Two Codes used per Node

Code Type/Length	Node	Codes		Rotations	
		1 st	2 nd	1 st	2 nd
Kasami – 63	1 st	1	4	33	32
	2 nd	7	8	20	0
Gold – 127	1 st	1	6	53	39
	2 nd	5	9	29	115
Kasami – 255	1 st	1	10	185	159
	2 nd	3	12	33	104

In this case, there was one code combination returned by the algorithm for each code length. For the 63-chip Kasami codes, the secondary test (Section 3.2.3.2.2) actually produced the optimal code combination. For the secondary test an overall weight from equation (17), which was the sum of the squares of the maximum partial and cross-correlation values, was found to be 881 for the secondary test, while it was 889 for the primary test. Although the secondary test provided better results, they were only marginally so (less than 1%). For the other two code lengths, the primary test (Section 3.2.3.2.1) provided the lowest-correlation code combinations. Thus, for the Two Node,

Two Sequence configuration, the primary test provides good results for all cases, although not the lowest correlation result for all cases.

For the 63-chip Kasami, 127-chip Gold and 255-chip Kasami sequences, the average correlation values of the code configuration chosen were found to be 10.49, 18.15 and 23.83, respectively. These values are greater than their respective periodic correlation bounds by 1.49, 1.15 and 6.83 and these differences are 2.37%, 0.9% and 2.68% of their respective autocorrelation peaks. As can be seen in the comparison between all of the cases up to this point, the average correlation value diverges from the periodic correlation bound as more cross-correlations must be considered by equation (17). However, at this point the divergence is still a relative small percentage of the autocorrelation peak and multiple access communication has not yet been disturbed.

4.3.3 Two Nodes, Four Codes

As in Section 4.2.2, all of the simulations stated in Chapter 3 were carried out and, in this case a total of 8 partial correlations and 16 cross-correlations had to be considered by equation (17). The code rotations that minimized the partial and cross-correlations were found and compiled in Table 13.

Table 13: Code Selection and Rotation for Two Nodes with Four Codes used per Node

Code Type/Length	Node	Codes				Rotations			
		1 st	2 nd	3 rd	4 th	1 st	2 nd	3 rd	4 th
K – 63	1 st	1	3	4	5	0	3	30	3
	2 nd	2	6	7	8	4	30	8	1
G – 127	1 st	1	4	6	8	61	46	118	37
	2 nd	9	10	11	12	111	6	77	15
K – 255	1 st	1	2	3	10	1	218	33	147
	2 nd	7	8	9	12	8	167	81	101

Because there are now essentially 24 variables considered in the minimization (8 partial correlations and 16 cross-correlations) the problem is fairly constrained and there were certain code combinations that produced the lowest composite correlation value. For the 63-chip and 127-chip codes, the primary test provided the lowest correlation, while the secondary test (part 1) provided the lowest correlation for the 255-chip code. Again, the primary test provided the best results overall for the same amount of computation but, in each case, the results provided by the various tests differed by only about 1% for each code length.

For the 63-chip Kasami, 127-chip Gold and 255-chip Kasami sequences, the average correlation values of the configuration were found to be 10.49, 18.15 and 23.83, respectively. These values are greater than their respective periodic correlation bounds by 2.95, 2.71 and 8.81 and these differences are 4.68%, 2.13% and 3.45% of their respective autocorrelation peaks. As in the previous case, the divergence between average correlation value and the periodic correlation bound is continuing to increase. Also, this divergence is increasing more rapidly for the Kasami sequences than for the Gold sequences. This result makes sense, since periodic Kasami sequences have extremely low correlation bounds, which are very dependent upon the code structure. Thus, they have more room to degrade than a similar length Gold sequence. However, at this point the divergence is still a relative small percentage of the autocorrelation peak and multiple access communication has not yet been disturbed.

4.4 Three Node System

Finally, after the simpler one and two node system were analyzed for various code configurations and code lengths, the simulation for the three node system was carried out. In these analyses, there are either one, two or four codes being used by each node. For the one and two node cases, all code lengths can be analyzed since these require fewer than

eight codes to be implemented. There are only eight codes total in the entire small set of Kasami sequences. However, for the three node, four code per node case, twelve codes are required and, therefore, only the 127-chip Gold sequences and 255-chip Kasami sequences can be analyzed.

4.4.1 Three Nodes, One Code

In this case, three autocorrelations and three cross-correlations are considered by the algorithm. Also, both the primary and secondary tests were performed and the code combinations that provided the lowest correlation values can be found in Table 14.

Table 14: Code Selection and Rotation for Three Nodes with One Code used per Node

Code Type/Length	Node	Codes	Rotations
		1 st	1 st
Kasami – 63	1 st	1	2
	2 nd	3	3
	3 rd	7	7
Gold – 127	1 st	1	64
	2 nd	8	37
	3 rd	10	6
Kasami – 255	1 st	1	185
	2 nd	3	33
	3 rd	12	104

In this simulation, the primary test provided the lowest correlation for all three code lengths under test. Thus, the algorithm behind the primary test proved to be the most useful in this case and a good starting point for future algorithms.

For the 63-chip Kasami, 127-chip Gold and 255-chip Kasami sequences, the average correlation values of the lowest-correlation configuration were found to be 10.37, 17.84 and 23.4, respectively. These values are greater than their respective periodic correlation bounds by 1.37, 0.84 and 6.4 and these differences are 2.17%, 0.66% and 3.45% of their

respective autocorrelation peaks. As before, these differences are not great but, unlike before, there are now three nodes trying to operate in the same time/frequency space. Therefore, it is important to check whether the processing gain is still great enough for detection to take place.

In the worst case, codes will line up in such a way that two cross-correlation peaks and one partial correlation peak will constructively add to produce an interference peak. Also, it is possible that an autocorrelation peak could be simultaneously degraded by the negative overlap of two cross-correlation peaks. Formulaic representations of this phenomenon are found in equation (19) and (20):

$$I = PC_{\max} + \sum_{n=1}^N CC_{n_{\max}} \quad (19)$$

$$S = AC - \sum_{n=1}^N CC_{n_{\max}} \quad (20)$$

where

I: Interference Spike

S: Signal Spike

AC: Autocorrelation Peak

PC: Partial Correlation
Peak

CC: Cross-Correlation
Peak

N: # of Multiple-Access
Interferers

In order to make a signal decision, *S* should minimally be greater than *I* to avoid the necessity of more complex processing. However, depending on the variability in the environment, it should probably be even greater. For the purpose of this experiment, it is assumed that the environment is static and that autocorrelation and cross-correlation values will not fluctuate much. Taking the average correlation peak to represent both *PC* and *CC*, *S* and *I* are 42.26 and 31.11 for the 63-chip Kasami configuration, 91.38 and 53.52 for the 127-chip Gold configuration and 208.2 and 70.2 for the 255-chip Kasami

configuration. In all three cases, S is greater than I , so communication could take place. However, the 255-sequence configuration has the greatest difference and would provide the most reliable communication.

4.4.2 Three Nodes, Two Codes

For these simulations, a total of six partial correlations and twelve cross-correlations must be minimized according to equation (17). The simulations from Section 3.2.3.2.1 and 3.2.3.2.2 were performed and the code configuration and rotations chosen by the algorithm can be seen in Table 15.

Table 15: Code Selection and Rotation for Three Nodes with Two Codes used per Node

Code Type/Length	Node	Codes		Rotations	
		1 st	2 nd	1 st	2 nd
Kasami – 63	1 st	1	8	2	1
	2 nd	2	7	4	7
	3 rd	3	4	3	32
Gold – 127	1 st	1	11	61	77
	2 nd	4	8	46	37
	3 rd	10	12	6	15
Kasami – 255	1 st	1	8	185	166
	2 nd	3	4	33	38
	3 rd	5	12	33	104

In this case, the primary test provided lowest correlation for the 63-chip Kasami sequence and the 127-chip Gold sequences. For the 255-chip Kasami sequence, the configuration/rotation generated by the secondary test (part two) performed better than the configuration from the primary test. However, this configuration had an overall weight of 11,856 (as determined by equation (17)) as compared with a weight of 11,876, which was found using the primary test. Thus, although the secondary test provided a lower overall composite correlation for the 255-chip Kasami sequence, it was only

marginally better (less than 1 percent). Therefore, the primary test should be used for used to choose the configuration.

For the 63-chip Kasami, 127-chip Gold and 255-chip Kasami sequences, the average correlation values of the configuration returned by the selection algorithm were found to be 11.65, 19.36 and 25.66, respectively. These values are greater than their respective periodic correlation bounds by 2.65, 2.36 and 6.4 and these differences are 4.21%, 1.86% and 3.40% of their respective autocorrelation peaks. As before, the differences are not great but the comparison between the worst case signal and interference spikes should be examined.

Taking the average correlation peak to represent both PC and CC , S and I are 39.7 and 34.95 for the 63-chip Kasami configuration, 88.28 and 58.08 for the 127-chip Gold configuration and 203.68 and 76.98 for the 255-chip Kasami configuration. In all three cases, S is greater than I , so communication could take place. However, in the case of the 63-chip Kasami configuration the ratio S/I is only 1.14, meaning that the worst-case signal and interference spikes are very close in magnitude. For the 127-chip Gold and 255-chip Kasami, S/I is 1.52 and 2.65, respectively. Therefore, although communication should be theoretically possible in all cases, it would be safer to use the 127-chip Gold and 255-chip Kasami configurations over the 63-chip Kasami configuration.

4.4.3 Three Nodes, Four Codes

In this configuration, twelve codes are required and, since there are only eight codes in the entire small set of 63-chip Kasami sequences, only the 127-chip Gold and 255-chip Kasami sequences were analyzed. Here equation (17) must account for 12 partial correlations and 48 mutual cross-correlations. The simulations from Chapter 3 were carried out the code configurations/rotations are included in Table 16.

Table 16: Code Selection and Rotation for Three Nodes with Four Codes used per Node

Code Type/Length	Node	Codes				Rotations			
		1 st	2 nd	3 rd	4 th	1 st	2 nd	3 rd	4 th
G – 127	1 st	1	3	4	11	55	53	28	76
	2 nd	2	6	10	12	30	37	3	15
	3 rd	5	7	8	9	30	99	34	111
K – 255	1 st	1	2	3	10	0	209	33	147
	2 nd	4	8	9	12	244	167	78	104
	3 rd	5	6	7	11	75	30	8	61

In this case, the primary test provided the lowest correlation for the 127-chip Gold codes, while the secondary test (part two) provided the lowest correlation for the 255-chip Kasami sequences. Although the secondary test provided better results, they were only less than one percent better than those provided by the primary test. Also, as is stated in Section 3.2.3.2.2, the secondary test holds half of the codes present in a configuration to a single rotation, while varying the other half more than the primary test can afford. Since it is somewhat arbitrary which codes are varied and the results produced using this method are only marginally better, the primary test is the better method for selecting the correct code configuration.

For the 127-chip Gold and 255-chip Kasami sequences, the average correlation values of the lowest-correlation configuration were found to be 21.21 and 27.75, respectively. These values are greater than their respective periodic correlation bounds by 4.21 and 10.75 and these differences are 3.31% and 4.22% of their respective autocorrelation peaks. Taking the average correlation peak to represent both *PC* and *CC*, *S* and *I* are 84.58 and 63.63 for the 127-chip Gold configuration and 199.5 and 83.25 for the 255-chip Kasami configuration. In both cases, *S* is greater than *I*, so communication could take place, but the 255-chip Kasami configuration would obviously provide greater reliability.

4.5 Comparison Between the Best and Worst Case Code Configurations

After selecting code configurations for each node/code configuration, according to the methods laid out in Chapter 3, it can be useful to view the worst possible code choice for select cases to see how great the improvement may be. For this portion of the experiment, the worst-case code choices for the 255-chip Kasami sequences were examined for each case listed in Table 6. The configurations that provided the highest composite correlation value were found using a method similar to that in Chapter Three. However, in this case, the data was sorted so that only the rotations that provided the highest partial correlations were considered by the algorithm. The average correlation value, as determined by equation (18), for both the lowest and highest correlation cases in each configuration can be found in Table 16.

Table 17: Comparison between the Code Configurations that Provide the Lowest and Highest Overall Average Correlation

Rotations	One Code		Two Codes		Four Codes	
	Lowest	Highest	Lowest	Highest	Lowest	Highest
One Node	13	35	18	35	20.41	34.75
Two Nodes	21.21	34.35	23.83	34.13	25.80	32.74
Three Nodes	23.40	34.33	25.66	32.94	27.75	31.33

In the one-node, one-code case, the highest average correlation is almost three times greater than the lowest average correlation. However, in the three-node, four-code case, the highest average correlation value is only 1.13 times the lowest. If the table is examined in greater detail, one should notice that from left to right or from top to bottom, the lowest case average correlation increases while the highest case average correlation decreases. This trend shows that, as the complexity in the system increases and there are more cross-dependencies between codes, as manifested by the total number of mutual cross-correlations that must be considered by equation (17), the extremes tend to

approach one another. However, the tendency of the extremes to move toward one another as complexity increases may be a result of the fact that only a more limited set of codes and rotations could be tested for these complex configurations. If non-linear optimization techniques could be applied to the selection algorithm to reduce selection complexity, a greater disparity might be seen between the code sets that provide the lowest and highest correlation values. With the current construction of the selection algorithm, the benefits of painstaking code selection are reduced as the number of codes in the configuration increases. Therefore, specific selection of codes and rotations shows the greatest returns for configurations that contain fewer codes.

CHAPTER 5: CONCLUSION AND FUTURE WORK

5.1 Analysis of Aperiodic Code Generation

5.1.1 Methods for Generating Code Configurations

In Sections 4.1 to 4.3, code combinations and rotations were determined and their method of generation (i.e. primary test or secondary test from Section 3.2.3.2.1 and 3.2.3.2.2) were analyzed. At the outset of the experiment, the expectation was that the primary test would be the better algorithm for code determination, since it did not require that half of the codes in a configuration be arbitrarily held to a single rotation, while the other half is varied more widely than allowed in the primary test. Overall, this expectation was met because, in the 14 cases that used both the primary and secondary methods to generate code sets, the results from the primary method were just as good as or better than the results from the secondary method ten times. In the other four cases, the results produced by the secondary method were only marginally better than those produced by the primary method (less than one percent each time). In addition, the secondary method requires that half of the codes be held to a single rotation while the other half is varied. Therefore, if computation time and resources are limited the optimal code configuration should be found using the primary, not the secondary, method in order to produce a code configuration that had low overall correlation properties.

However, as was shown in Section 4.5, the current selection algorithm has varying levels of effectiveness, depending on the overall number of codes and cross-dependencies in a specific configuration. For system configurations that use a low number of codes, such as the three-node, one-code system, there are clear differences between the lowest

correlation and highest correlation code configurations. However, as the system complexity increases, as in the three-node, four-code configuration, the lowest and highest case average correlations are almost identical. Therefore, if the relative system complexity is small, the algorithm is effective in choosing low-correlation code configurations. As the system becomes more complex, though, the algorithm proves to be less useful.

5.1.2 Comparison of the Aperiodic and Periodic Spread Spectrum

In this project, near-“ideal” aperiodic pseudorandom code configurations were generated, but it is important to see how these would perform, using some established benchmark for comparison. The obvious comparison to make is the comparison between the average correlation value of the code set, taking both partial correlations and cross-correlations into account, and the theoretical bounds placed upon periodic spread spectrum. In all cases, it was shown that the average correlation value never differed from the periodic correlation bound by more than 5% of the autocorrelation peak. Also, for the three-node system, where most multiple access interferers were present, it was shown that communication could take place in all cases, although it was cutting it close for the 63-chip Kasami sequence set in the three node, two code configuration. Overall, the code set generated here, following the constraints laid out in Chapter Three, would be capable of encoding multiple bits per symbol while maintaining multiple-access communication capabilities.

5.2 Future Work

As can be seen in equation (17), the optimization becomes increasing more difficult as the number of codes per node but, more importantly, the number of nodes in the system increases. For this project, strict limitations were placed on how many rotations and

exactly which rotations the algorithm should consider when performing the optimization. In order to find a truly optimal configuration, the primary method for code generation should be modified to incorporate adaptive nonlinear optimization techniques. This was not incorporated into the master's thesis because the time required to perform the additional background research and application to the specific problem was prohibitive and the work required would rather suit a PhD dissertation. Instead of placing strict limitations on the analysis at the outset of the simulation, the algorithm should be able to dynamically determine which rotations should be thrown out and find points of convergence. Thus, even when the number of nodes, and therefore, complexity increases, the algorithm could minimize the impact of the additional complexity on computation time.

In addition to improving the selection algorithm, it would be interesting to test the coding scheme in a real-time, wireless backscatter sensing environment. The tags (nodes) would be preconfigured with correct antenna polarization mismatching, so that the power could be properly balanced at the receiver. In addition, reception and decoding would be performed on an FPGA platform, implementing an iterative, DSSS/MA processing algorithm. It would be interesting to see how this implementation of aperiodic spread spectrum would behave in a realistic, fading environment.

APPENDIX A: BASE CODES USED IN THE SIMULATIONS

Table 18: 63-Chip Sequences from the Small Set of Kasami Sequences

Code #	Code Value
1	111111010101100110111011010010011100010111100101000110000100000
2	000110001001001000101100011001111001100101011100011010101010010
3	001101101100111010010101000101010111110010010111111111011000101
4	011010100111011111100111111100001011011100000000110100111101011
5	110100110000010100000010001110110010000000101110100011110110111
6	10100001111000001100100110101100000011001110010001101100001110
7	010001000010101101011110100000100101001011001011010001001111100
8	100011111011110001110000110111101110101110111001101000010011001

Table 19: 127-Chip Sequences from the Set of Gold Sequences

Code #	Code Value
1	0000000010010010100010010110110001101110101011010111100000011110100 00001101001100100011010100000000111110000111011111100110000
2	0000001011011111101101001010111110010110010000010010101011100100001 00000110010010111111111110011001110001100101010111010100001
3	0000011001000101110011110010100001100111100110011000111100010001011 000100001011100001101010010101001101010110010000110110000011
4	000011101110001001110000010011110000100001010001100010011111011111 001111010101111101000001011001001011100011100100101111000111
5	0001110100011000110101100011100001000011010010100101001100101110111 011001101001000100010111000001000110001000001100011101001111
6	0011100111001011000010100000011111001101100011110111110010000100111 110100010000110110111011110001011101011111011101111001011111
7	0111000001101100101100100111100011010000000001010010001111010000110 101111100011010011100010010001101011110001111110110001111111
8	1110001100100011110000101000011011101011000100011001110101111000100 011000000100011001010001010000000110101100111000100000111111
9	1100010110111101001000110111101010011101001110001110000000101000001 110111001010001100110111010011011100010110110100000010111110
10	1000100010000000111000001000001001110001011010100001101010001001010 101001010110100111111011010101101001100010101101000110111100
11	0001001011111011011001110111001110101001110011111110111111001011100 0101011011111100011000110111000000010001010011111001110111000
12	0010011000001100011010001001000000011000100001000000010101001110001 101100011101011101010011000011010101011011111011011110110001

Table 20: 255-Chip Sequences From the Small Set of Kasami Sequences

Code #	Code Value
1	111111100100001010011111010101011100000110001010110011001011111101 1110011011101110010101001010001001011010001100111001111000110110000 1000101110101111011011111000011010011010110110101000001001110110010 010011000000111010010001110001000000010110001111010000
2	00010100000000101001100111101101010011000100101000111111101000001000 0111011100000101011101110111010000011101100111110001000101101111000 0111111000111110100001001010010101001100100111010010111011111001000 10000100111110110000000001011110010011001011001000001
3	0010100101100110111000110010010110111001110110111101010001100010110 110001010011100000010011000011101101010101101010100000010000100001 0110000010001100101110011100000100110110010101011101101101101000111 110111011110010110110010000100100100001000100011110011
4	010100111010111000010110101101000101001011111000000001000100101011 101000010100001011011011111101101000100100000011010001101010010011 0101110111101000110000110000100111000011110001000011000001001011001 01101111101100011010110011010001000101011010110010111
5	101001100011111111111011001011110000100101111111010111010101010001 011010011011011101001010000100000110011101010111111001001111110111 0010011100100000001101101001100000101000111001111110011000001100100 000010111010001000011110100111010001101100111101011111
6	0100110100011100001010111101000000101000001100001111011110110100100 1111100001011100011010011100011000100000111110110110101110100111111 110100101011000111011101101110111111110101000000100101010000011110 110000110101011110001111011101100011100011101011001110
7	10011011010110111000011101011111011100010010111001000101100010011111 1101101110001010001011100110101010101111101000100111010100010101110 0011100110010010000010111111110001010010001011110001001110011101011 01010010101111001010110010100000011111101000111101101
8	0011011111010100110111100100000111000011000100110010000111110011001 1001110000100110101000010011001110110001000100000100100001110001101 111011111101010110100111011100110000101100110001101000011010000000 011100010110101011101011000011001111000000011110101010
9	0110111011001010011011000111110010100111011010011110100100000110101 000100110111111110111111000000110001100011101000011001010111001010 010000110101101011111100110110110111001000011001100010111011010110 001101101100011001100100010101011110111010101100100101

Table 20 (continued)

10	110111001111011100001000000011001101111100111000111100011101101100 0000110111001101100000101110010111110110101111001100011100101000101 0001101001000100010011000101000011011101011101100000110100101111010 101110011001111101111010111001111101001111001000111011
11	101110001000110111000000111100111111110011101110101101100111011110 0011000010101001111110000010110100000011001011010010110000001011011 1010100001111001001010000010101000010101100000111001110011000100011 10100111001011010100011110000011101010010000000000110
12	0111000001111000010100010001100011011101101000010001110010010111010 0100101001100001000011011011110011101000000011101111101001001100110 1100110000000011111000001101111110000100011010001011111100010010001 100110100100100100111101010010110101110010010001111100

REFERENCES

- [1] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal*, Vol. 27, July, October 1948, pp. 379-423, 623-656.
- [2] D. L. Liu, W. W. Y. Ng, D. S. Yeung and H. L. Ding, "A Brief Survey on Current RFID Applications," In *2009 International Conference on Machine Learning and Cybernetics*, Vol. 4, 12-15 July 2009, pp. 2330-2335.
- [3] D. M. Pozar, *Microwave Engineering, 2nd Edition*. Hoboken, New Jersey: John Wiley & Sons, 1998.
- [4] D. M. Dobkin, *The RF in RFID, Passive UHF RFID in Practice*. New York: Elsevier Inc, 2008.
- [5] J. D. Griffin and G. D. Durgin, "RF Tag Antenna Performance on Various Materials Using Radio Link Budgets," *IEEE Antennas and Wireless Propagation Letters*, Vol. 5, Issue 1, Dec. 2006, pp. 247-250.
- [6] J. G. Proakis and M. Salehi, *Digital Communications*, New York: McGraw Hill, 2008.
- [7] Mark IV IVHS, "ROADCHECK Flatpack Transponder," Flatpack Transponder (FPT) datasheet, 2001 [RevA 05-01].
- [8] J. Landt, "The History of RFID," *IEEE Potentials*, Vol. 24, Issue 4, 2005, pp. 8-11.
- [9] S. Sarma, D. Brock and D. Engels, "Radio Frequency Identification and the Electronic Product Code," *IEEE Micro*, Vol: 21, Issue 6, 2001, pp 50-54.
- [10] C. S. Hartmann and L. T. Claiborne, "Fundamental Limitations on Reading Range of Passive IC-Based RFID and SAW-based RFID," In *Proc. 2007 IEEE Conference on RFID*, March 26-28, 2007, pp. 41-48.
- [11] I. F. Akyildiz, W. Su, Y. Sankarabramaniam and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, Volume: 40, Issue: 8, 2002, pp. 102-114.
- [12] J. R. Smith, A. P. Sample, P. S. Powledge, S. Roy and A. Mamishev, "A Wirelessly-Powered Platform for Sensing and Computation," *Ubicomp 2006*, LNCS 4206, 2006, pp. 495-506.
- [13] EPC Global, Inc. "EPC Radio-Frequency Identity Protocols, Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz – 960 MHz" Version 1.2.0, 23 October 2008.

- [14] A. R. Rivera, D. K. Klair and K. W. Chin, "A Simulation Study on the Energy Efficiency of Pure and Slotted Aloha based RFID Tag Reading Protocols," In *Proc. 6th IEEE Consumer Communications and Networking Conference, 2009*. pp. 1-5.
- [15] V. P. Ipatov, *Spread Spectrum and CDMA, Principles and Applications*. Hoboken, New Jersey: John Wiley & Sons, Ltd, 2005.
- [16] R. C. Dixon, *Spread Spectrum Systems with Commercial Applications*. New York: John Wiley & Sons, 1994.
- [17] J. J. Komo and S. C. Liu, "Modified Kasami Sequences for CDMA," In *Twenty-Second Southeastern Symposium on System Theory*, 1990, pp. 219-222.
- [18] R. Skaug and J. F. Hjelmstad. *Spread Spectrum in Communication*. London: Peter Peregrinus Ltd., 1985.
- [19] D. V. Sarwate and M. B. Pursley, "Crosscorrelation Properties of Pseudorandom and Related Sequences," *Proceedings of the IEEE*, Vol. 68, Issue 5, 1980, pp 593-619.
- [20] R. Gold, "Optimal Binary Sequences for Spread Spectrum Multiplexing," *IEEE Transactions on Information Theory*, Vol. 13, Issue 4, 1967, pp. 619-621.
- [21] A. F. Mohammed, "Near-Far Problem in Direct-Sequence Code Division Multiple Access Systems" In 7th IEE European Conference on Mobile and Personal Communications, 13-15 Dec 1993, pp. 151-154.
- [22] T. Pratt, C. W. Bostian and J. E. Allnut, *Satellite Communications, 2nd Edition*, Hoboken, New Jersey: John Wiley & Sons, Inc., 2003.
- [23] J. D. Griffin and G. D. Durgin, "Complete Link Budgets for Backscatter-Radio and RFID Systems," *IEEE Antennas and Propagation Magazine*, Vol. 52, Issue 2, April 2009, pp. 11-25.
- [24] S. Moshavi, "Multi-User Detection for DS-CDMA Communications," *IEEE Communications Magazine*, Vol. 34, Issue 10, October 1996, pp. 124-136.
- [25] A. Rohatgi and G. D. Durgin, "Implementation of an Anti-Collision Differential-Offset Spread Spectrum RFID System," In *IEEE Antennas and Propagation Society International Symposium 2006*, pp 3501-3504.
- [26] Zhu Quiling, et al., "A Robust Radio Frequency Identification System Enhanced with Spread Spectrum Technique," In *IEEE International Symposium on Circuits and Systems 2009*. pp. 37-40
- [27] M. K. Simon, J. K. Omura, R. A. Scholtz and B. K. Levitt, *Spread Spectrum Communications Handbook*. New York: McGraw-Hill, Inc., 1994.