TECHNICAL UNIVERSITY OF CRETE

SCHOOL OF ELECTRICAL & COMPUTER ENGINEERING

DIPLOMA THESIS

Non-Orthogonal Multiple Access in Modern Wireless Networks with Clustering Techniques

Author:
Konstantinos Skyvalakis

Thesis Committee:
Assoc. Prof. Aggelos Bletsas
(Advisor)
Assoc. Prof. George Karystinos
Prof. Michael Paterakis

A thesis submitted in partial fulfilment of the requirements for the degree of Diploma in Electrical and Computer Engineering

December 17, 2018

Non Orthogonal Multiple Access in Modern Wireless Systems with the Use of Clustering Techniques

by Konstantinos Skyvalakis

Thesis Advisor: Aggelos Bletsas

Abstract

Non-orthogonal multiple access is an old problem, which has recently attracted renewed interest. This work revisits the problem in the context of industrial RFIDs and scatter radio. The latter has emerged as a key enabling wireless technology for low-cost and large-scale ubiquitous sensing. Radio frequency identification (RFID) tags/sensors utilize scatter radio technology to transfer sensed information to readers, typically employing Gen2, the industrial RFID protocol. In this thesis, a new system model is developed for the simultaneous transmission of two Gen2 RFID tags, as well as a channel estimation algorithm, based on clustering techniques.

In the system model part of the thesis, two collided RN16 packets from two RFID tags are considered, which may not be perfectly synchronized. This work builds upon prior art and further considers the time offset that can occur in the transmission among the two tags. It is shown that there are three possible scenarios, based on the time offset of the most delayed tag and their connection is further highlighted. Work includes both theoretical analysis of these scenarios, as well as experimental evaluation from a testbed.

The channel estimation part of the thesis employs the clustering algorithm called Affinity Propagation, based on probabilistic graphical models and inference algorithms. Channel estimates are obtained, based on the clustered, received data patterns, in conjunction with a line fitting method. Having obtained the channel estimates, the offset of the most delayed tag is estimated. Finally, the collision scenario is inferred and tag data detection is performed. Future work will focus on different clustering algorithms, as well as memory-based, long-sequence detection.

Acknowledgements

After five years of truly hard yet exciting work, I cannot but feel grateful for the things I lived and also blessed for the people who were by my side all these years, making all I achieved possible.

First and foremost, I would like to thank my advisor, *Prof. Aggelos Bletsas*, for giving me an opportunity to work in his lab, exposing me to research and constantly pointing me towards exciting topics. He was always more than eager to discuss my questions and ideas and support me in any way he could. I also thank him for teaching me my first two MIT courses, which proved to be a great experience.

However, nothing would have been possible without *my family and my close friends.* I want to thank *my parents*, for their constant and unconditional love and support, also *my brother Manolis*, whose achievements inspired me as well, and motivated me to work even harder, and of course, no words are enough to thank *Evgenia*; she has been by my side both in good and bad times, for many years now.

I would also like to thank my lab colleagues *Michael, Giorgos, Vaggelis G., Vaggelis K.* and *Thanasis*, whose help has been more than valuable throughout this thesis work. We had a lot of fun times and it was really a pleasure working with you guys. Finally, a big thank you to my friend *Giorgos K.* for the research related talks we had at times and the ideas we exchanged for the projects we worked on, that really kept me motivated and willing to strive for success.

Contents

Chapter 1

Introduction

> "When wireless is perfectly applied the whole earth will be converted into a huge brain, which in fact it is, all things being particles of a real and rhythmic whole. We shall be able to communicate with one another instantly, irrespective of distance."
>
> Nikola Tesla

Wireless communications have well made their way in our lives throughout these last two decades. Everyday technologies like WiFi, BlueTooth, RFIDs, smartphones, smartwatches, etc. have definitely made a great impact in our daily lives. Great progress has also been noted in the sectors of health care, education, entertainment, news reporting and environmental protection with the aid of wireless communications. However, along with all this technological growth, a massive load of data has also made its way to the surface. Data that need to be downloaded, uploaded or shared as soon as possible, since our daily routine or the circumstances require so from now on.

Throughout these years one could focus on a couple of major cellular network technologies like 3G or 4G that brought along with them higher bandwidth of communication, thus higher transfer rates, better voice quality, faster and safer access to the internet, streaming media etc. However most of these technologies also brought along with them a few new operating frequencies to add in to the already crowded frequency spectrum.

To overcome spectral efficiency issues, various multiple access techniques were invented, like FDMA, TDMA, CDMA, SDMA, PDMA, to list a few. Nonetheless, solving the problem of frequency allocation and also meeting the demands of a great amount of users sometimes cannot be met at the same time, which means that someone is not going to be pleased and that costs money both to the user and the provider of the communication link. Of course, there are scenarios where these techniques have proved what they are capable of and that is the exact reason they are still being used up until today.

That being said, a new generation of wireless communications that will sustain such heavy loads and equally serve every user to the best of its ability, is needed. A new multiple access scheme is required and that new scheme could probably be a Non Orthogonal Multiple Access or NOMA for short. NOMA's application in future 5G networks is already being included in many papers (Ding et al. [30],[25] and Xu et al. [27]) and application experiments are already starting to take place. It is envisioned to be an essential component of the 5G network.

Massive MIMO sytem's ability to serve multiple users - and multiple devices - simultaneously within a condensed area while maintaining fast data rates and consistent performance makes it the perfect technology to address the needs of the forthcoming 5G era (Ding et al. [28],[31],[29]). Massive MIMO is already deployed and being used commercially in China and Japan within the 4G LTE context. While this thesis work mainly focuses on RFID networks, which are a part of the IoT that 5G will revolutionise, the ideas and facts mentioned above can be easily adopted and adjusted to fit the application's framework without deviating from the original course or making any compromises.

1.1 The Promise of NOMA

Orthogonal multiple access techniques have been used during the past and although they still perform great under certain scenarios and circumstances, a better trade-off between system throughput and user fairness needs to be realized. Thus, the promising solution is to break orthogonality.

The basic concept of NOMA is to serve multiple users simultaneously at the same frequency and code. In addition to that, users with better channel conditions would get less power, while successive interference cancellation (SIC) is used at the receivers of all users when needed.

As Liu et al. [23] argue, NOMA promises very high channel capacity gain and better spectral efficiency than the present state of the art technologies thus satisfying the needs of Internet of Things and of 5G, in which a massive amount of users require to be served rapidly for small packet transmissions; with low latency. The concept of 5G is to virtually connect everything together, from our driverless cars to our fridge, thus implementing a ubiquitous connectivity scheme. So far, the only promising architecture to achieve the great throughput needed for such a venture is NOMA.

1.2 Backscatter Radio

Backscatter communication is comprised of a very simple, low-power and low-cost implementation, which basically allows a backscatter node (RFID tag) to communicate with the reader by modulating and reflecting the incident continuous wave from the reader. Lately, it is being considered as a promising solution to power the future Internet of Things in 5G networks. Thus, an application of NOMA on RFID networks seems very interesting.

An RFID tag can be seen as an automatic identification device which uses radio-frequency electromagnetic fields to identify objects carrying tags when they come close to a reader's detection range. A simple RFID network usually consists of a reader and the RFID tags/chips. The reader in our lab's experiments is a USRP (Universal Software Radio Peripheral), which basically is a software defined radio (SDR); meaning you can use your PC to program it they way you want it to operate, while the tags are commercial OTS tags that cost around 0.15$ each.

1.3 Thesis Organization & Contributions

In this section we jointly outline the organization of this thesis and its key contributions.

In **Chapter 2**, basic RFID knowledge is offered along with some details about the FM0 encoding scheme and the EPCglobal Class 1 Gen2 Protocol, that was used in the simulations. The EPCglobal Gen2 Specification protocol can be found here [2].

In **Chapter 3**, basic definitions and theorems are given along with a clustering algorithm that will be used in the channel estimation process. A system model for the simultaneous asynchronous transmission of two Gen2 RFID tags along with some basic knowledge on how RFID systems operate and their applications, is provided for the first time.

In **Chapter 4**, a derived equivalent system model for two tags along with its mathematical derivation is provided. We differentiate between the different scenarios that might occur during the asynchronous RN16 transmission of the tags and the mandatory signal to noise (SNR) relations are provided.

In **Chapter 5**, a technique for channel estimation using clustering is offered along and an offset estimation algorithm. In more detail, the key contributions of the chapter are the following:

- The DC offset of the received signal is estimated.
- We take advantage of our powerful clustering algorithm, to provide a channel estimation scheme based on the clusters generated by the transmitted symbols on the I/Q plane, and their repeating pattern.
- An offset estimation technique based on our channel coefficients estimation is provided. At this step, we are able to distinguish which tag is which on a physical level.

In **Chapter 6**, a detection scheme for the proposed system model is provided, our results are also verified through the Bit Error Rate diagrams.

In **Chapter 7**, we conclude this thesis, and we provide directions for future work and possible extensions to our solutions.

Chapter 2

RFID Systems Basics

In this chapter, the operating process of an RFID system is formalized. Furthermore, the FM0 encoding algorithm is presented, while also our system model for the simultaneous transmission of two Gen2 RFID tags is provided.

2.1 Basics of an RFID system

Radio-frequency identification (RFID), uses electromagnetic fields to identify and track tags attached or embedded to objects. It is a technology with a purpose similar to the one of the barcodes, but unlike a barcode, the tag need not be within the line of sight of the reader. An RFID system usually consists of tags, that are mostly attached to objects and transfer information to a processing device called reader also known as interrogator. Typical RFID applications include logistics and supply chain visibility, item level inventory tracking, materials management, access control, biomedical sensor applications and sensor networks, to name a few.

RFID systems operate in various frequency bands; low frequency (LF), high frequency (HF) and ultra-high frequency (UHF). The LF band covers frequencies from 30 kHz to 300 kHz and is not considered a truly global application because of the slight differences in frequency and power levels throughout the world. The HF band ranges from 3 to 30 MHz and there are several standards in place, such as the ISO 15693 standard for tracking items and others. Last, UHF band covers range from 300 MHz to 3 GHz and across the European Union is regulated by the European Telecommunications Standards Institute (ETSI), and ranges from 865 MHz to 868 MHz, based on the ECPglobal Gen2, which we will go through to the extent that is required for this thesis, in Section 2.2.

An RFID system can have one of the following possible configurations, monostatic or bistatic. A monostatic RFID system utilizes the same antenna circuitry for both transmission and reception, i.e. both antennas in the system first transmit, then switch modes, typically with the help of a circulator, and start receiving RF energy. This system is usually much cheaper than a bistatic system because it requires less antennas and RF cables, but it is also slower than standard bistatic systems because the reader has to switch between transmitting and receiving modes. In a bistatic system each antenna performs one function only, i.e. the antenna will either transmit or receive RF energy. This configuration requires a bistatic reader, that has two separate RF channels for transmitting and receiving RF energy for every port. The transmitter and receiver can also be dislocated offering advantages as increased coverage and lower cost, as argued by Kimionis et al. [18].

RFID tags can be divided into three categories : active, passive and battery-assisted passive. Passive tags, in contradiction to active or semi-passive ones, are not powered by a battery, instead they harvest all the energy the get from the reader's RF signal.

Communication between the reader and passive or semi-passive tags is achieved by means of reflection. A passive tag typically terminates its antenna between two loads; in that way, the incident carrier wave (CW) is reflected with altered amplitude and/or phase and tag information is modulated on the tag's antenna reflection coefficient changes. The work of, Bletsas et al. [7], demonstrates an optimized way of selecting these loads.

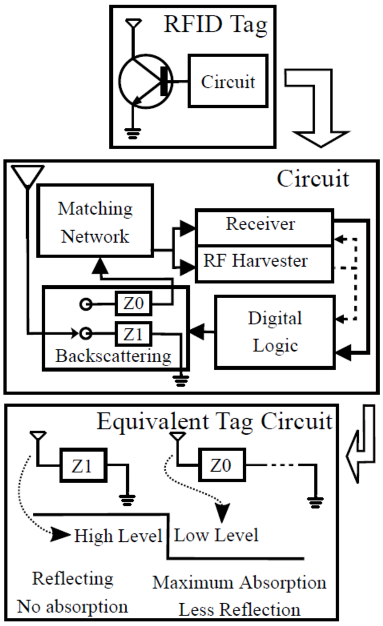


Figure 2.1: Top figure: the RFID tag. Middle figure: RFID tag's circuitry. Bottom figure: RFID tag switching between 2 loads and corresponding (on-off keying) backscattered signal, as depicted in the work of Ouroutzoglou et al. [4].

2.2 EPCglobal Class 1 Gen2 Protocol

This protocol, Gen 2 or EPCglobal Class 1 Generation 2 as its full name is, defines the physical and logical requirements for a passive backscatter, Interrogator (Reader) Talks First (ITF), RFID system operating in the 860-960 MHz frequency range. However, there are usually two basic frequencies of operation at, 860-868 MHz and 902-928 MHz. As mentioned earlier, the tag harvests energy and replies with its data only when instructed to do so by the reader. In addition, readers and tags are not required to 'talk' simultaneously; rather, communications are half-duplex, meaning that readers 'talk' and tags 'listen', or vice versa.

2.2.1 Description of Operating Procedures

In this section, the key elements needed from the Gen 2 protocol will be briefly discussed, in order to formalize the communication procedure between the tags and the reader and vice-versa. No further analysis of the protocol will be conducted, since some of the assumptions in this thesis work, do not require further knowledge of the protocol's experimental features.

2.2.2 Physical Layer

The physical layer of the protocol describes how the tags and the reader exchange information. At all times, the reader must be transmitting a continuous carrier-wave

(CW), in order for the tag to collect energy from it and be able to respond. Furthermore, the modulation needs to be simple in order for a low power tag to decode it. This layer can be seen as the signalling interface, that defines frequencies, modulation, data coding, data rates, and other parameters required for RF communications.

2.2.3 Downlink Communication

Downlink communication refers to reader-to-tag communication. The reader uses a type of amplitude shift keying (ASK) modulation with pulse-interval encoding (PIE), as depicted in Fig. 2.2, to communicate with the tags. The tags receive their operating energy from this same modulated RF carrier. The reader must use a fixed modulation format and data rate for the duration of an inventory round. An analysis over the PIE encoding scheme and its reference times, or the ASK modulation used by the reader, will not be provided since it will not be needed throughout this thesis.

The reader always starts the transmission with either a Preamble or Frame-sync packet. A Preamble precedes a Query command, while all other reader commands begin with a Frame-Sync packet. In this thesis only the Preamble packet was used, since the main interest lies on the Query commands, where the tags reply with their RN16 sequence and a collision might occur.

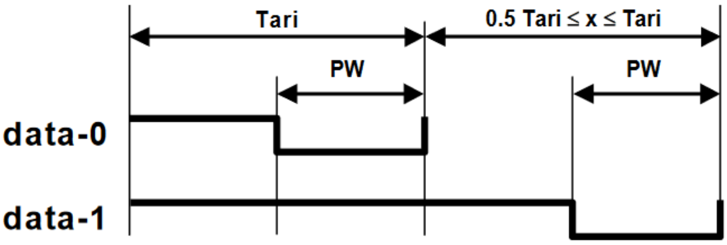


Figure 2.2: PIE encoding for downlink communication, as shown in Gen2 [2].

2.2.4 Uplink Communication - FM0 Encoding

Tags shall encode the backscattered data as either FM0 baseband or Miller modulation of a subcarrier at the given data rate. The reader specifies the encoding type that will be used in the tags response. In this thesis, we only worked with the FM0 encoding scheme, hence, FM0 will be the only technique we will focus on. In FM0 encoding, amplitude level must change at every symbol boundary, while a data-0 transmission has also an amplitude change in the middle of the symbol. The data-0 symbol is transmitted either with a $s_2(t)$ pulse, or with a $s_3(t)$ pulse, depending on the data bit that was previously transmitted. The same applies for data-1, which is either transmitted with a $s_1(t)$ pulse or with a $s_4(t)$ pulse. One can easily grasp that this is a memory-based modulation. The admissible symbol transitions, can be seen in the state diagram offered in Fig. 2.4b, while the four FM0 symbols mentioned above, can be seen in Fig. 2.4a. It is also mandatory, for an FM0 transmission to always end with a dummy data-1 symbol, as shown in Fig. 2.6. All possible sequences that can be created with FM0 encoding, are presented in Fig. 2.5. Last but not least, it should be noted that, each tag response begins with the Preamble sequence shown in Fig. 2.3. The **v** indicates an FM0 violation; an amplitude change should have normally occurred, but since this is the preamble that violation exists on purpose.

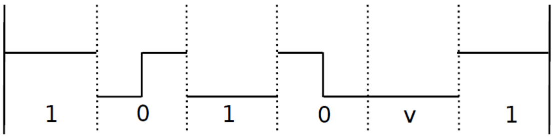
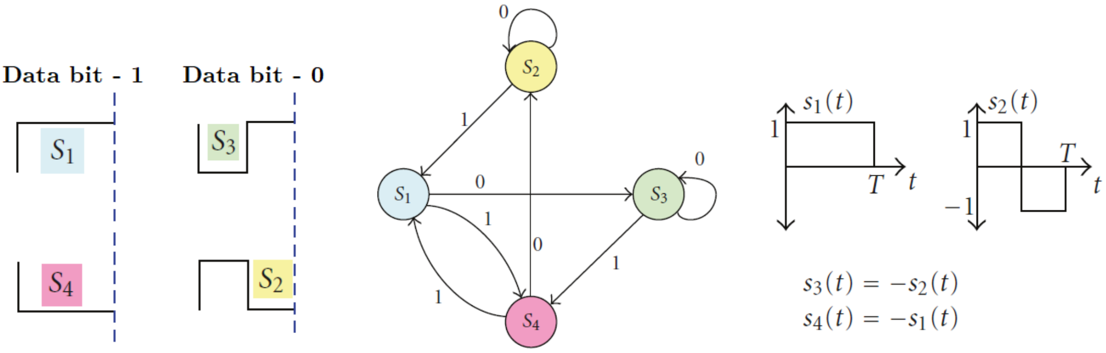


Figure 2.3: Tag Preamble for uplink communication.



(a) FM0 Symbols. (b) FM0 State diagram. (c) FM0 Basis functions.

Figure 2.4: FM0 encoding in RFID uplink communications.

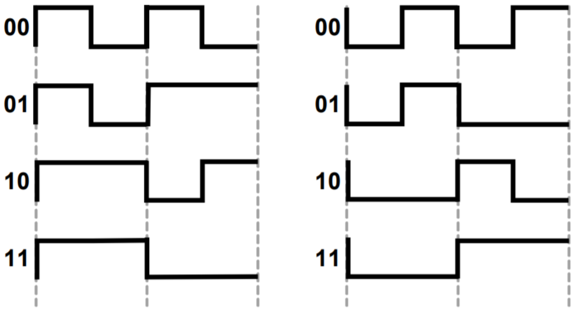


Figure 2.5: FM0 Sequences.

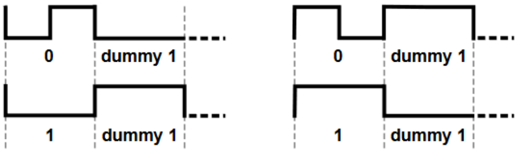


Figure 2.6: FM0 Transmission termination, as shown in Gen2 [2].

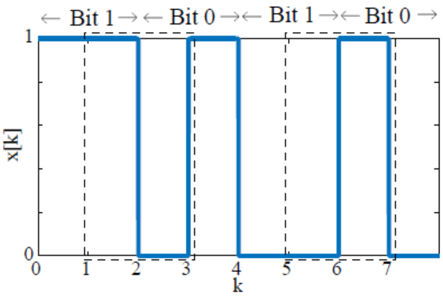


Figure 2.7: FM0 line coding signal. The dashed areas correspond to two orthogonal s-shaped waveforms that can simplify detection.

2.2.5 Decoding the FM0 packet

As mentioned earlier, FM0 is a memory based modulation, where work on Bletsas et al. [5] and Simon et al. [1], has shown that after shifted examination of the transmitted waveform by $T/2$ before the beginning of the bit, where T is the bit (symbol) period, only two possible pulse shapes can be generated, shown in Fig. 2.7 marked with dashed lines.

In order to detect a transmitted bit, the reader has to differentially decode two received symbols (using $2T$ signal observation instead of just T), realizing a gain of 3dB compared to maximum-likelihood symbol-by-symbol detection. More specifically, memory of FM0 signalling is exploited in the detection of two collided FM0 signals by observing the duration of exactly two bits: the bit under observation, half-bit before it, and half-bit after it. In Chapter 6, a more in depth analysis of the detection scheme will be provided.

Chapter 3

Clustering Algorithms & Basic Definitions

In this chapter, basic definitions and theorems will be introduced along with some discussion about the clustering algorithm that was used in the thesis. Furthermore, our proof regarding the equivalence of k-means to EM on GMM is provided. Finally, our system model for the asynchronous transmission of 2 RFID tags is provided.

3.1 Definitions & Theorems

Initially, the notion of Complex Normal random variables and random vectors along with their important subclass of circular symmetry is introduced.

3.1.1 Complex Gaussian Distribution

Before proceeding into explaining the concepts of circular symmetry in complex Gaussian random variables we first have to introduce the complex normal distribution and the proper complex Gaussian distribution.

Definition 3.1.1. If $\mathcal{X}$ and $\mathcal{Y}$ are jointly Gaussian random variables, then $\mathcal{Z} = \mathcal{X} + j\mathcal{Y}$ is also a complex Gaussian random variable.

Definition 3.1.2. A proper complex Gaussian vector, is a Gaussian vector which is complex and has **non-zero** mean ($E[\boldsymbol{z}] \neq 0$) and specific probability density function:

$$p_{\boldsymbol{z}}(\boldsymbol{z}) = \frac{1}{\pi|\boldsymbol{C}|} e^{-(\boldsymbol{z}-E[\boldsymbol{z}])^H \boldsymbol{C}^{-1}(\boldsymbol{z}-E[\boldsymbol{z}])}$$

3.1.2 Circular Symmetry for Gaussian Random Variables

An important subclass of the Complex Normal family is called the Circularly Symmetric Complex Normal or for short CSCN. Circularly Symmetric Complex Normal random variables are used extensively in signal processing, and are sometimes referred to as just Complex Normal in signal processing literature. A Circularly Symmetric Complex Gaussian random variable or vector is a proper complex Gaussian random variable or vector that has **zero** mean. Consider a CSCN random variable $\mathcal{Z} = \mathcal{X} + j\mathcal{Y}$, then the following properties hold.

- $\mathbb{E}[Z] = 0$
- $\mathbb{E}[ZZ^T] = \mathbb{E}[Z^2] = 0 \rightarrow \text{var}(X) = \text{var}(Y),\ \text{cov}(X, Y) = 0$

- If $\mathcal{Z}$ is a circularly symmetric complex Gaussian random variable, its real and imaginary parts are independent and have equal variance.

The probability density function of a complex random variable and its value will be now computed for the case of a circularly symmetric Gaussian.

$$\begin{aligned} f_Z(z) = f_{X,Y}(x,y) &\overset{(!)}{=} f_X(x)f_Y(y) \\ &= \frac{1}{\sqrt{2\pi\frac{\sigma^2}{2}}}e^{-\frac{x^2}{\sigma^2}}\frac{1}{\sqrt{2\pi\frac{\sigma^2}{2}}}e^{-\frac{y^2}{\sigma^2}} \\ &= \frac{1}{\pi\sigma^2}e^{-\frac{x^2+y^2}{\sigma^2}} \\ &= \frac{1}{\pi\sigma^2}e^{-\frac{|z|^2}{\sigma^2}} \end{aligned}$$

Where at (!) we took advantage of the independence between the two random variables that make up the real and the imaginary components of our random variable $\mathcal{Z}$, as mentioned above, to further explain the expression.

3.1.3 Circular Symmetry for Gaussian Random Vectors

Definition 3.1.3. If $\boldsymbol{\mathcal{X}} \sim \mathcal{N}(\boldsymbol{\mu}_x, \frac{\sigma^2}{2}\boldsymbol{I})$ and $\boldsymbol{\mathcal{Y}} \sim \mathcal{N}(\boldsymbol{\mu}_y, \frac{\sigma^2}{2}\boldsymbol{I})$ are jointly Gaussian random vectors , then $\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{X}} + j\boldsymbol{\mathcal{Y}}$ is a complex Gaussian random vector with the characteristics of $\mathcal{CN}(\boldsymbol{\mu}_z, \sigma^2\boldsymbol{I})$, where $\boldsymbol{\mu}_z = \boldsymbol{\mu}_x + \boldsymbol{\mu}_y$.

Definition 3.1.4. A complex Gaussian random vector $\boldsymbol{\mathcal{Z}} \sim \mathcal{CN}(\boldsymbol{\mu}_z, \sigma^2\boldsymbol{I}_n)$ is circularly symmetric iff. $e^{j\phi}\boldsymbol{\mathcal{Z}}$ has the same distribution as $\boldsymbol{\mathcal{Z}}$ for all real values of ϕ, which can only happen when $\mathbb{E}[\boldsymbol{Z}] = 0$.

If $\boldsymbol{\mathcal{Z}}$ is circularly symmetric, then:

- $\mathbb{E}[\boldsymbol{Z}] = \mathbb{E}[e^{j\phi}\boldsymbol{Z}] = e^{j\phi}\mathbb{E}[\boldsymbol{Z}] = 0$, since $\mathbb{E}[\boldsymbol{Z}] = 0$
- $\text{cov}(e^{j\phi}\mathbf{Z}) = \mathbb{E}[e^{j\phi}\boldsymbol{Z}e^{-j\phi}\boldsymbol{Z}^T]$
- Define pseudocovariance matrix of $\boldsymbol{\mathcal{Z}}$ as $\mathbb{E}[\boldsymbol{Z}\boldsymbol{Z}^T]$, then $\mathbb{E}[\mathbf{Z}\mathbf{Z}^T] = \mathbb{E}[e^{j\phi}\boldsymbol{Z}e^{j\phi}\boldsymbol{Z}^T] = e^{2j\phi}\mathbb{E}[\boldsymbol{Z}\boldsymbol{Z}^T] = 0$, since $\mathbb{E}[\boldsymbol{Z}\boldsymbol{Z}^T] = 0$

The formula of the probability density function of the random vector case will not be provided, since it will not be needed throughout this thesis. For further reading please refer to the work of R. Gallager [22] on Circularly-Symmetric Gaussian random vectors.

3.1.4 Law of Iterated Expectation

The following proposition in probability theory is more commonly known as the law of iterated expectation or otherwise as the law of total expectation.

Definition 3.1.5. If X is a random variable whose expected value $\mathbb{E}[X]$ is defined, and Y is any random variable on the same probability space, then, $\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X|Y]]$, i.e., the expected value of the conditional expected value of X given Y is the same as the expected value of X.

One special case states that if $\{A_i\}$ is a finite or countable partition of the sample space, then:

$$\mathbb{E}[X] = \sum_i \mathbb{E}[X|A_i]\mathbb{P}(A_i) \tag{3.1}$$

Proof.

$$\begin{aligned}\mathbb{E}[\mathbb{E}[X|Y]] &= \mathbb{E}_{y}\left[\sum_{x} x\mathbb{P}(X=x|Y)\right]\\ &= \sum_{y}\left[\sum_{x} x\mathbb{P}(X=x|Y)\right]\mathbb{P}(Y=y)\\ &= \sum_{y}\sum_{x} x\mathbb{P}(X=x,Y=y)\end{aligned}$$

If the series is finite, then the summations can be switched around, and the previous expression will become:

$$\begin{aligned}\sum_{y}\sum_{x} x\mathbb{P}(X=x,Y=y) &= \sum_{x} x\sum_{y}\mathbb{P}(X=x,Y=y)\\ &= \sum_{x} x\mathbb{P}(X=x)\\ &= \mathbb{E}[X]\end{aligned}$$

This completes the proof. □

The law of iterated expectation will be needed in Chapter 4 where the SNR relations of our model are going to be calculated.

3.2 Affinity Propagation Clustering

In this part of the thesis it is going to be briefly explain how the Affinity Propagation algorithm operates and some very basic math formulas will be provided in order to help demonstrate the true power and simplicity of this clustering algorithm.

To begin with, Affinity propagation simultaneously considers all data points as possible exemplars, exchanging real-valued messages between them until a high-quality set of exemplars (and corresponding clusters) emerges. Messages are updated on the basis of simple formulae that reflect sum-product or max-product update rules and, at any point in time, the magnitude in each message reflects the current affinity that one point has for choosing another data point as its exemplar, hence the name 'Affinity Propagation'.

Affinity propagation takes as input a collection of real-valued similarities between data points, $\{s(i,k)\}$, where each similarity $s(i,k)$ indicates how well the data point with index k is suited to be the exemplar for data point i. Each data point is paired with a variable node, c_i in a factor graph as shown in Figure 3.1 below. A value of $c_i = k$ for $i \neq k$ indicates that data point i is assigned to a cluster with point k as its exemplar; $c_k = k$ indicates that data point k serves as a cluster exemplar.

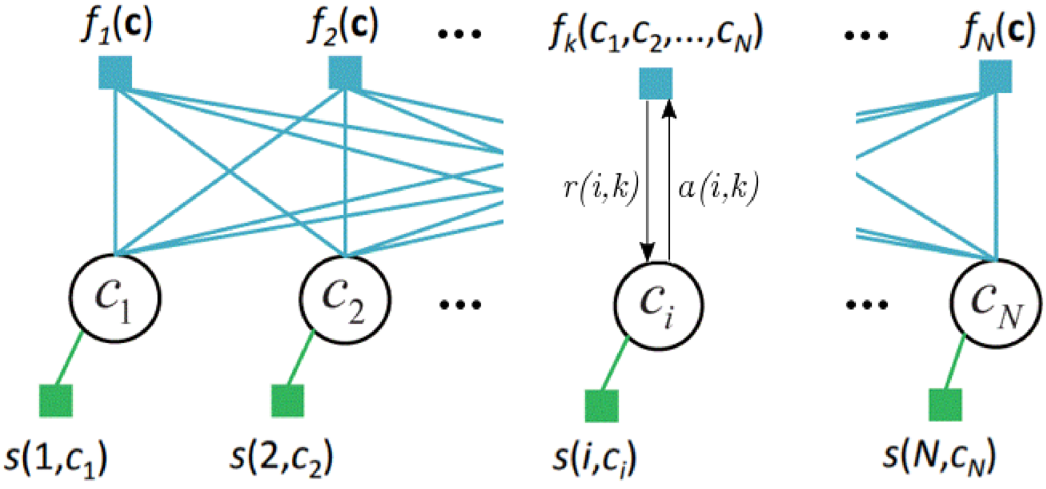


Figure 3.1: Factor Graph Network of Affinity Propagation

The graph's function is a constrained net similarity (exponentiated, so the function is non-negative), defined as follows:

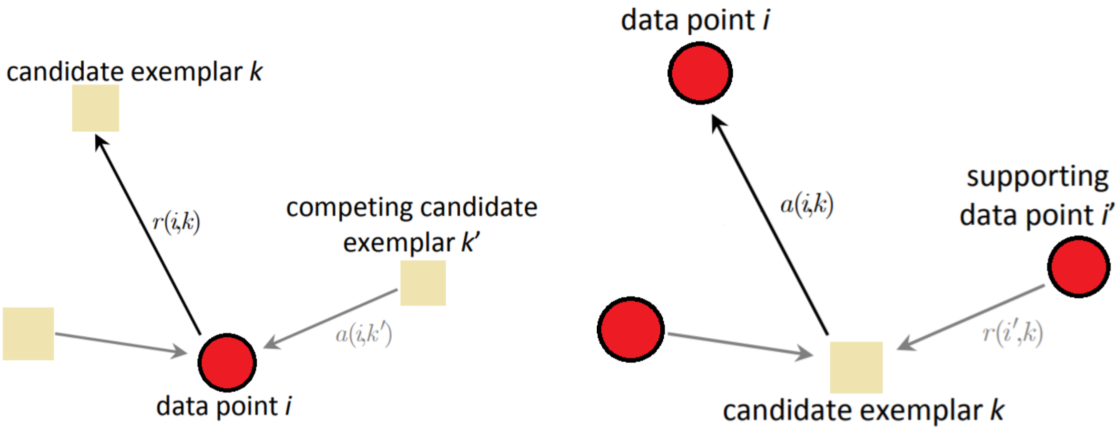
$$F(\boldsymbol{c};\boldsymbol{s}) = e^{\sum_{i=1}^{N} s(i,c_i)+\sum_{k=1}^{N} \log f_k(\boldsymbol{c})} = \prod_{i=1}^{N} e^{s(i,c_i)} \cdot \prod_{k=1}^{N} f_k(c_1, c_2 \ldots, c_N),$$

where, the second term contains a coherence constraint defined as follows:

$$f_k = \begin{cases} 0, & \text{if} \quad c_k \neq k \text{ but } \exists\, i : c_i = k \text{ (disallow clusters without an exemplar)} \\ 1, & \text{otherwise} \end{cases},$$

which causes the function to evaluate to zero for the incoherent configuration of a cluster without an exemplar, i.e., a data point i has chosen k as its exemplar ($c_i = k$) with k having been incorrectly labelled as a non-exemplar ($c_k \neq k$). The goal of affinity propagation is to search over configurations of variables c in the factor graph to maximize $F(\boldsymbol{c};\boldsymbol{s})$. The affinity propagation algorithm went through many stages, until it obtained its last form that is based on the Max-Product algorithm. The analysis of these versions is outside the context of this thesis.

As mentioned earlier, affinity propagation exchanges messages between data-points. There are two type of messages that are being exchanged, the responsibility messages $r(i,k)$ that are passed from variable nodes to function nodes (i.e., data points to candidate exemplars) and availability messages $a(i,k)$ that are passed from function nodes to variable nodes (C), interpreted as candidate exemplars to data points. Figure 3.2 better depicts the message exchange process.



(a) Responsibility messages.

(b) Availability messages.

Figure 3.2: Message exchange in affinity propagation.

The final version of these messages also make up the update rules of the algorithm. The steps of the algorithm are presented below.

```
Algorithm 1: Affinity Propagation
  Input: $\{s(i,j)\}_{i,j\in\{1,\ldots,N\}}$ data similarities and preferences
  Output: Cluster assignments $\hat{\boldsymbol{c}} = (\hat{c}_1,\ldots,\hat{c}_N)$, $\hat{c}_i = \text{argmax}_k\{a(i,k)+r(i,k)\}$
  $\forall i,k:\ a(i,k)=0,\ r(i,k)=0$
  while unconverged do
      $\forall i,k:\ r(i,k) = s(i,k) - \text{argmax}_{k':k'\neq k}\{s(i,k')+a(i,k')\}$
      if $k==i$ then
          $a(i,k) = \sum_{i':i'\neq i}\max\{0, r(i',k)\}$
      else
          $a(i,k) = \min\{0, r(k,k) + \sum_{i':i'\notin\{i,k\}}\max\{0, r(i',k)\}\}$
```

The simplicity and effectiveness of these update equations have made it the standard incarnation of affinity propagation since its initial 2007 publication in Science [10]. All of the above is extensively presented on the PhD work of D. Dueck [11], as well as on the book of Koller and Friedman [33] on probabilistic graphical models and inference algorithms. There also exist, multiple versions of the algorithm available for download at http://www.psi.toronto.edu, like the one where you force affinity to find a specific number of clusters, K, which is going to be used during the detection process in Chapter 6.

3.3 k-Means as a limit case of Expectation Maximization on GMM

k-Means is a clustering algorithm which makes hard assignments of points to clusters, which means that a point either totally belongs to a cluster or not at all. Often, k-Means doesn't work when clusters are not round shaped, and/or may overlap, and/or are unequal. The k-Means algorithm, performs the following steps iteratively until convergence, where a possible convergence criteria could be that the cluster means do not change anymore. Before we proceed to the proof we are going to lay some fundamental pieces of the process that will help us along the way. Below the k-means algorithm is presented in its simplest form.

- **Input:** N data samples $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}; \boldsymbol{x}_i \in \mathbb{R}^D$
- **Initialize:** K cluster means $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\mu}_k \in \mathbb{R}^D$. Usually initialized randomly, but good initialization is crucial and differs from one problem to another; many smarter initialization heuristics exist (e.g.,K-means++, Arthur & Vassilvitskii, 2007)
- **Iterate:**
    - (Re-)Assign each data sample $\boldsymbol{x}_i$ to each closest cluster center.

    $$\boxed{\mathcal{C} = \{i : k = \underset{k}{\operatorname{argmin}} \, ||\boldsymbol{x}_i - \boldsymbol{\mu}_k||^2\}}$$

    ($\mathcal{C}_k$ is the set of data samples assigned to cluster k with center $\boldsymbol{\mu}_k$)
    - Update the cluster means.

    $$\boxed{\boldsymbol{\mu}_k = mean(\mathcal{C}_k) = \frac{1}{|\mathcal{C}_k|} \sum_{i \in \mathcal{C}_k} \boldsymbol{x}_i}$$

    - Repeat until convergence.

A Gaussian Mixture Model (GMM) is a probabilistic model for representing the presence of subpopulations of Normal distributions within an overall population of Normal distributions, i.e. the component distributions are Gaussians. A GMM can be modelled like this:

$$\mathbb{P}(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathbb{P}(\boldsymbol{x}|\boldsymbol{\theta}_k), \;\; \boldsymbol{x} \in \mathbb{R}^D$$

Where π_k's are the mixing weights and it always holds that : $\sum_{k=1}^{K} \pi_k = 1, \;\; \pi_k \geq 0$. Intuitively, π_k is the proportion of data generated by the k-th distribution. Each component distribution $\mathbb{P}(\boldsymbol{x}|\boldsymbol{\theta}_k))$ represents a 'cluster' in the data.

The challenge we face at this point is to learn the parameters of the GMM. Given N observations $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ drawn from the Normal distribution $\mathbb{P}(\boldsymbol{x})$ learning the parameters of the GMM involves.

- Learn the cluster assignments $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_N\}$.
- Estimating the mixing weights $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_K\}$ and the parameters $\theta = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$ of each of the Gaussians.

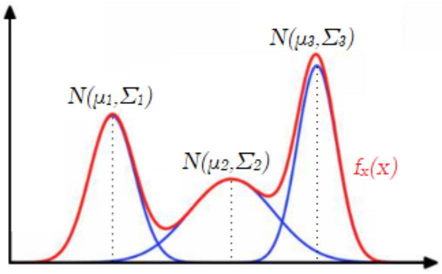


Figure 3.3: GMM of 3 Gaussians.

One of the algorithms that can be used to learn these parameters of our multivariate GMM is Expectation Maximization (EM). The EM algorithm steps and update rules are presented below.

First, initialize parameters θ and the mixing weights π_k and then alternate between the following steps until convergence:

- Given the current estimates of θ and π, estimate the posterior probability that $\boldsymbol{x}_i$ belongs to cluster k.

$$\gamma_{i,k} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum\limits_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j}, \ \forall\ i,k$$

- Given the current estimates of the above posterior probabilities $\gamma_{i,k}$ estimate the mean, the covariance matrix and the mixing weight of each of the Gaussians involved.

$$\begin{aligned}
\boldsymbol{\mu}_k &= \frac{1}{N_k}\sum_{i=1}^{N}\gamma_{i,k}\boldsymbol{x}_i, \quad \forall k,\ where\ N_k = \sum_{i=1}^{N}\gamma_{i,k} \\
\boldsymbol{\Sigma}_k &= \frac{1}{N_k}\sum_{i=1}^{N}\gamma_{i,k}(\boldsymbol{x}_i-\boldsymbol{\mu}_k)(\boldsymbol{x}_i-\boldsymbol{\mu}_k)^T,\ \forall k \\
\pi_k &= \frac{N_k}{N},\ \forall k
\end{aligned}$$

Lemma 1. *k-Means is equal to a hard-assignment EM for GMM, in the case of* $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma} = \sigma^2 \boldsymbol{I}_{N\times N}$*, and with equiprobable cluster centers,* $w_k = \frac{1}{K}$*.*

Proof. Consider the posterior probabilities of cluster assignments

$$\gamma_{i,k} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum\limits_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)} = \frac{\pi_k exp\{-\frac{1}{2\sigma^2}||\boldsymbol{x}_i-\boldsymbol{\mu}_k||^2\}}{\sum_{j=1}^{K}\pi_j exp\{-\frac{1}{2\sigma^2}||\boldsymbol{x}_i-\boldsymbol{\mu}_j||^2\}}, \ \forall\ i,k$$

As $\sigma^2 \to 0$ we observe that in the denominator the term for which $||\boldsymbol{x}_i - \boldsymbol{\mu}_j||^2$ is smallest, will go to zero most slowly, and hence the responsibilities $\gamma_{i,k}$ for the data point $\boldsymbol{x}_i$ all go to zero except for term j, for which the responsibility $\gamma_{i,k}$ will go to unity. Thus, in this limit, we obtain a hard assignment of data points to clusters, just as in the k-means algorithm.

$$\gamma_{i,k} \approx \frac{\pi_j exp\{-\frac{1}{2\sigma^2}||\boldsymbol{x}_i - \boldsymbol{\mu}_j||^2\}}{\pi_j exp\{-\frac{1}{2\sigma^2}||\boldsymbol{x}_i - \boldsymbol{\mu}_j||^2\}} = 1$$

Let us write :

$$||\boldsymbol{x}_i - \boldsymbol{\mu}_k||^2 = \delta_k$$

Then

$$\frac{\pi_k exp\{-||\boldsymbol{x}_i - \boldsymbol{\mu}_k||^2\}/2\sigma^2}{\sum_{j=1}^{K} \pi_j exp\{-||\boldsymbol{x}_i - \boldsymbol{\mu}_j||^2/2\sigma^2\}} = \frac{\pi_k exp\{-\delta_k/2\sigma^2\}}{\sum_{j=1}^{K} \pi_j exp\{-\delta_j/2\sigma^2\}}$$

If we take

$$\delta^* = \operatorname*{argmin}_i \delta_i$$

we have

$$\frac{\pi_k exp\{-\delta_k/2\sigma^2\}}{\sum_{j=1}^{K} \pi_j exp\{-\delta_j/2\sigma^2\}} = \frac{\pi_k exp\{(\delta^* - \delta_k)/2\sigma^2\}}{\sum_{j=1}^{K} \pi_j exp\{(\delta^* - \delta_j)/2\sigma^2\}}$$

where $\delta^* - \delta_k < 0$ except for $k = k^*$ where $\delta^* - \delta_{k^*} = 0$. So, for all $k \neq k^*$,

$$\lim_{\sigma^2 \to 0} \frac{\pi_k exp\{(\delta^* - \delta_k)/2\sigma^2\}}{\sum_{j=1}^{K} \pi_j exp\{(\delta^* - \delta_j)/2\sigma^2\}} = \lim_{\sigma^2 \to 0} \frac{\pi_k exp\{(\delta^* - \delta_k)/2\sigma^2\}}{\pi_{k^*} \sum_{j \neq k^*} \pi_j exp\{(\delta^* - \delta_j)/2\sigma^2\}} = 0$$

since, for $a > 0$,

$$\lim_{\epsilon \to 0} exp\{-a/\epsilon\} = 0$$

while

$$\lim_{\sigma^2 \to 0} \frac{\pi_{k^*} exp\{(\delta^* - \delta_{k^*})/2\sigma^2\}}{\sum_{j=1}^{K} \pi_j exp\{(\delta^* - \delta_j)/2\sigma^2\}} = \lim_{\sigma^2 \to 0} \frac{\pi_{k^*} \times 1}{\pi_{k^*} \sum_{j \neq k^*} \pi_j exp\{(\delta^* - \delta_j)/2\sigma^2\}} = 1$$

For $l \neq j, \gamma_{i,l} \approx 0 \Rightarrow$ hard assignment with $\gamma_{i,j} = 1$ for a single cluster j.Thus for $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma} = \sigma^2 \boldsymbol{I}_{N \times N}$ (spherical covariance matrix) and $\sigma^2 \to 0$, EM for GMM reduces to k-Means. The update rules of EM we mentioned above are :

$$\begin{aligned} z_i &= \operatorname*{argmax}_k \frac{1}{K} \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}_k) \\ &= \operatorname*{argmax}_k exp\Big\{-\frac{1}{2\sigma^2}||\boldsymbol{x}_i - \boldsymbol{\mu}_k||^2\Big\} \end{aligned}$$

The M-step becomes :

$$|\mathcal{C}_k| = \sum_{i=1}^{N} \mathbb{1}_{z_i = k} = \sum_{i : z_i = k} 1 = |\{i : z_i = k\}|$$

$$\boldsymbol{\mu}_k = mean(\mathcal{C}_k) = \frac{1}{|\mathcal{C}_k|} \sum_{i \in \mathcal{C}_k} \boldsymbol{x}_i$$

As we have foresaid the above update equations coincide with the k-Means update rules and this completes the proof. □

3.4 Channel Model for 2 Asynchronous Tags

In this section, our system model is presented for the first time, while in the next chapter an extensive derivation of an equivalent system model is provided that is going to be used in the simulations. Considering the simultaneous transmission of 2 tags, in the equations below we denote as k the number of samples, N the number of symbols to be transmitted and L the oversampling factor in our reader. According to Kimionis et al. [18], the received digitized signal is expressed as:

$$y[k] = h_A x_A[k] + h_B x_B[k] + n[k], \tag{3.2}$$

where, $x_A[k] = \sum\limits_{n=0}^{N-1} S^A_{d(n)}[k - nL - \tau_A], \quad x_B[k] = \sum\limits_{n=0}^{N-1} S^B_{d(n)}[k - nL - \tau_B].$

In the above equations, $n[k] = n(kT_s) \sim \mathcal{CN}(0, 2\sigma_n^2)$ is considered to be CSCN (Circularly Symmetric Complex Normal) additive white noise, i.e. $n[k] \perp n[m]$ for $k \neq m$, $d(n) \in \{0, 1\}$ denotes the transmitted bit, T denotes the nominal bit duration and S_0 and S_1 are set according to Kargas et al. [3] and can be selected between the following two waveforms, since these are the only waveforms that can be observed when we examine the transmitted waveform, shifted by $T/2$ before the beginning of the bit:

$$S_0[k] = \begin{cases} 1, & \text{if} \quad 0 \leq k < \frac{L}{2} \\ 0, & \text{if} \quad \frac{L}{2} \leq k < L \end{cases}, \quad S_1[k] = \begin{cases} 0, & \text{if} \quad 0 \leq k < \frac{L}{2} \\ 1, & \text{if} \quad \frac{L}{2} \leq k < L \end{cases}.$$

Last, $\tau_A, \tau_B \in [0, L-1]$ denote the sampled time offset before each tag starts transmitting. In this thesis work we have made the assumption that, $\tau_A = 0$, i.e. we are fully synchronized with tag A and we receive tag's A signal first, while time offset τ_B is considered to be the delay of tag's B response as demonstrated in Figure 4.1. We refer to them as sampled time offsets since we have oversampled the signal received on our reader, hence we can count by how many samples we observe this misalignment and then translate that into time units.

The received digitized signal $y[k]$ is then filtered with a square pulse impulse response $\Pi[k]$, given by:

$$\Pi[k] = \begin{cases} 1, & \text{if} \quad 0 \leq k < \frac{L}{2} \\ 0, & \text{otherwise,} \end{cases} \tag{3.3}$$

which has a length of $L/2$ taps.

The filtered signal is then given by:

$$y_f[n] = \sum_{k=-\infty}^{\infty} \frac{y[k]}{L/2} \Pi[n-k] \tag{3.4}$$

where we divide by $L/2$ in order to normalize our results.

3.5 Discussion

To summarize, in this chapter we introduced some basic theorems and definitions, mostly in the context of Complex Normal random variables, which are going to use throughout this whole thesis. We demonstrated briefly how the Affinity Propagation clustering algorithm works and how we are going to embed it into our implementation. We gave the complete proof of the k-Means as a limit case of EM algorithm on GMM. Finally, we presented for the first time a signal model for the simultaneous transmission, of two RFID tags, with the only assumption that we are able to identify where the collided RN16 packet starts and after locking on to that packet start we would be able to perform that mathematical analysis that precedes.

Chapter 4

Analysis & Derived Equivalent System Model

In this chapter we are going to prove that when 2 tags are transmitting simultaneously, with a time offset between their corresponding packet start, then the emerging scenarios can be classified into 3 different categories, depending on the time offset of the most delayed tag. It will also be demonstrated how the 2 tags problem becomes a 3 tags problem and under what circumstances it can occur.

4.1 System Model

Assuming that, $\tau_A = 0$ and that $\tau_B \in [0, L-1]$, in terms of samples as mentioned earlier, along with nominal bit duration L samples i.e., asynchronous 2-tags transmission, each FM0 symbol observed with a $L/2$ shift, can be written after matched filtering and sampling at the half of each symbol period, as a 2×1 complex vector of the following form:

$$\boldsymbol{y} \triangleq \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \underbrace{h_A \boldsymbol{x}_A + \boldsymbol{B} h_B \boldsymbol{x}_B}_{\boldsymbol{u}} + \boldsymbol{n} = \boldsymbol{u} + \boldsymbol{n}, \tag{4.1}$$

where, y_0 and y_1 correspond to the first and second half-bits respectively, $\boldsymbol{x}_A, \boldsymbol{x}_B \in \{\boldsymbol{e}_0 \triangleq [1 \quad 0]^T, \boldsymbol{e}_1 \triangleq [0 \quad 1]^T\}$, $\boldsymbol{n} \sim \mathcal{CN}(0, \frac{4}{L}\sigma_n^2 \boldsymbol{I}_2) \equiv \mathcal{CN}(0, \sigma^2 \boldsymbol{I}_2)$, $\boldsymbol{B}$ is a 2×2 shaping matrix and $\boldsymbol{u}$ is a 2×1 vector that its content varies depending on the scenario. Below it is declared the $\boldsymbol{u}$ vectors for each scenario where, the first row of vector $\boldsymbol{u}$ corresponds to the first half-bit and the second row to the second half-bit.

To begin with, there exist 3 scenarios as mentioned earlier, depending on the delay offset of tag B. These 3 scenarios are distinguished based on the number of clusters they form on the I/Q plane.

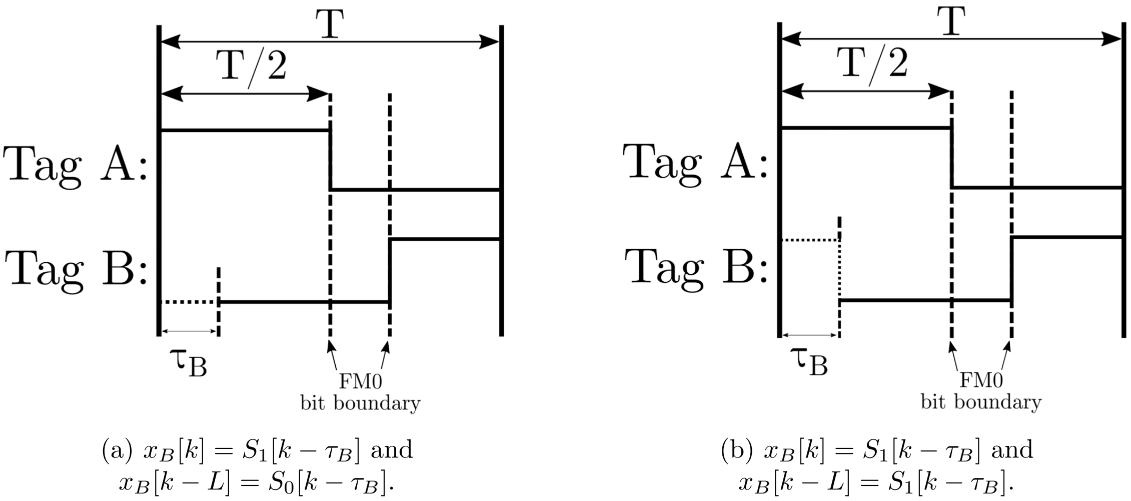
- Scenario 1: $\tau_B \in \{1, \dots, \frac{L}{4}-1, \frac{L}{4}+1, \dots, \frac{L}{2}-1, \frac{L}{2}+1, \dots, \frac{3L}{4}-1, \frac{3L}{4}+1, \dots, L-1\}$
- Scenario 2: $\tau_B \in \{\frac{L}{4}, \frac{3L}{4}\}$
- Scenario 3: $\tau_B \in \{0, \frac{L}{2}\}$

In order to extract the elements of the $\boldsymbol{u}$ vectors for each scenario, the output of the next two equations is needed, Equations (4.2) and (4.3), under different values of $\boldsymbol{x}_A, \boldsymbol{x}_B$, offset τ_B and previously transmitted bit of tag B, with regard to Table 4.1. We should not forget to mention that, the division by $L/2$ is done in order to normalize the results and focus on the offset of tag B.

$$y_0 = \sum_{k=0}^{\frac{L}{2}-1} \frac{y[k]}{L/2} = \sum_{k=0}^{\frac{L}{2}-1} \frac{h_A x_A[k]}{L/2} + \sum_{k=0}^{\frac{L}{2}-1} \frac{h_B x_B[k]}{L/2} + \sum_{k=0}^{\frac{L}{2}-1} \frac{n[k]}{L/2}, \tag{4.2}$$

$$y_1 = \sum_{k=\frac{L}{2}}^{L-1} \frac{y[k]}{L/2} = \sum_{k=\frac{L}{2}}^{L-1} \frac{h_A x_A[k]}{L/2} + \sum_{k=\frac{L}{2}}^{L-1} \frac{h_B x_B[k]}{L/2} + \sum_{k=\frac{L}{2}}^{L-1} \frac{n[k]}{L/2}. \tag{4.3}$$

Each bit a tag transmits, has an ideal duration of T seconds, where at the receiver that bit gets oversampled by a factor of L. This means that each half-bit, is comprised of $\frac{L}{2}$ samples. Focusing the detection on the L samples long window as seen in Figure 4.1, one can better observe the time offset τ_B and the multiple scenarios that can occur with different waveforms and time offsets.



(a) $x_B[k] = S_1[k - \tau_B]$ and $x_B[k-L] = S_0[k-\tau_B]$.

(b) $x_B[k] = S_1[k - \tau_B]$ and $x_B[k-L] = S_1[k-\tau_B]$.

Figure 4.1: Delayed response of Tag B by τ_B samples.

Studying the above figures, the FM0 bit boundary can be clearly noticed, as well as the dotted part of the waveform denoting the last bits of tag's B previous transmission 'intruding' by τ_B samples into our detection window, due to the offset. To make things easier, one could envision that intrusion as a 'pseudotag' C that interferes with the transmission of tag B. Essentially it is the same tag; that is why it was named 'pseudotag' in the first place, and it will obviously have the exact same channel coefficient with tag B, i.e. $h_C \triangleq h_B$; we will also only refer to it as tag C from now on. In Figure 4.1a, it can be noticed that tag C (dotted part of Tag B waveform) was definitely transmitting a S_0 waveform since it is intruding our detection window with τ_B samples of 0's. Equivalently, if the 0's were instead 1's, like in Figure 4.1b, then again it can be claimed with certainty, that the waveform tag C was transmitting was S_1. What is more, from the above figures it can also be observed that tag B is currently broadcasting a S_1 waveform. Bearing in mind the above observations, after the matched filtering process is performed, a $\boldsymbol{u}$ vector for that case can be extracted. To be more specific all that needs to be done is counting, which we is demonstrated below.

Based on Figure 4.1a, on the first half bit for both tags, i.e. y_0, there are $\frac{L}{2}$ 0's for tag B and $\frac{L}{2}$ 1's for tag A. While, on the second half-bit, y_1, there are $\frac{L}{2} - \tau_B$ 1's for tag B and $\frac{L}{2}$ 0's for tag A. Dividing by $\frac{L}{2}$ to normalize the results, and ignoring the noise, we obtain:

$$y_0 = \sum_{k=0}^{\frac{L}{2}-1} \frac{h_A x_A[k]}{L/2} + \sum_{k=0}^{\frac{L}{2}-1} \frac{h_B x_B[k]}{L/2} = \frac{1\frac{L}{2}h_A + 0\frac{L}{2}h_B}{L/2} = h_A,$$

$$y_1 = \sum_{k=\frac{L}{2}}^{L-1} \frac{h_A x_A[k]}{L/2} + \sum_{k=\frac{L}{2}}^{L-1} \frac{h_B x_B[k]}{L/2} = \frac{0\frac{L}{2}h_A + 0\tau_B h_B + 1\left(\frac{L}{2} - \tau_B\right)h_B}{L/2} = \left(1 - \frac{2\tau_B}{L}\right)h_B,$$

which can be written more neatly in the form of,

$$\boldsymbol{u} = \begin{bmatrix} h_A \\ \left(1 - \frac{2\tau_B}{L}\right)h_B \end{bmatrix}.$$

In the same manner as before, the $\boldsymbol{u}$ vector that will obtained if we use the Equations (4.2) and (4.3), on Figure 4.1b, whilst ignoring the additive noise, is:

$$\boldsymbol{u} = \begin{bmatrix} h_A + \frac{2\tau_B}{L}h_B \\ \left(1 - \frac{2\tau_B}{L}\right)h_B \end{bmatrix}.$$

Similarly, the rest of the cases can be studied, by alternating the values of $\boldsymbol{x}_A$, $\boldsymbol{x}_B$, τ_B and the waveform that tag B broadcast last (i.e. tag C transmission), as defined in Table 4.1. In that way, after extensive calculations, the following scenarios will be observed.

4.2 Scenario Analysis

Before proceeding into demonstrating the categories that occurred in our analysis below, it needs to be made clear that scenarios 2 and 3 are special cases of scenario 1. One might have already observed by now that the time offset τ_B values that are 'missing' from scenario 1, are indeed the time offset values that are examined in scenarios 2 and 3. They are considered to be special cases since they produce 6 and 4 clusters on the I/Q plane respectively, and can both be derived if the $\boldsymbol{u}$ vectors of scenario 1 are used, as it will be demonstrated in this section. Furthermore, we should also mention that in an indoor environment, where the distance between the tags and the reader is small and the RFID configuration is most likely monostatic, it is more likely to notice a scenario 3 situation, since the propagation delays do not have a great distance to develop and cannot be observed. On the other hand, in an outdoor environment with an ambient bistatic RFID configuration and large distances that scale up to many km, it is more likely to notice that propagation delay, hence it would be more likely notice a scenario 1 or 2 situation. In the next few pages, the scenarios that occur are listed along with their respective clusters, regardless of the RFID configuration.

Scenario 1

In scenario 1 two cases are distinguished that both produce $N_c = 8$ cluster centres in the I/Q plane:

- Case 0 $(\tau_B < \frac{L}{2})$: $\tau_B \in \{1, \ldots, \frac{L}{4} - 1, \frac{L}{4} + 1, \ldots \frac{L}{2} - 1\}$

- Case 1 ($\tau_B > \frac{L}{2}$): $\tau_B \in \{\frac{L}{2}+1, \ldots, \frac{3L}{4}-1, \frac{3L}{4}+1, \ldots, L-1\}$

Using Eq. (4.2),(4.3) and based on the notion that Figure 4.1 of the previous page introduced, the $\boldsymbol{u}$ vectors for the first case of scenario 1 can be extracted:

$$\begin{aligned}\boldsymbol{u} \in \Big\{ \boldsymbol{u}_0 &= \Big[h_A + (1-\frac{2\tau_B}{L})h_B \quad \frac{2\tau_B}{L}h_B\Big]^T, \boldsymbol{u}_1 = \Big[0 \quad h_A + (1-\frac{2\tau_B}{L})h_B\Big]^T,\\ \boldsymbol{u}_2 &= \Big[h_A \quad (1-\frac{2\tau_B}{L})h_B\Big]^T, \boldsymbol{u}_3 = \Big[(1-\frac{2\tau_B}{L})h_B \quad h_A + \frac{2\tau_B}{L}h_B\Big]^T,\\ \boldsymbol{u}_4 &= \Big[h_A + h_B \quad \frac{2\tau_B}{L}h_B\Big]^T, \boldsymbol{u}_5 = \Big[\frac{2\tau_B}{L}h_B \quad h_A + (1-\frac{2\tau_B}{L})h_B\Big]^T,\\ \boldsymbol{u}_6 &= \Big[h_A + \frac{2\tau_B}{L}h_B \quad (1-\frac{2\tau_B}{L})h_B\Big]^T, \boldsymbol{u}_7 = \Big[h_B \quad h_A + \frac{2\tau_B}{L}h_B\Big]^T\Big\}.\end{aligned}$$

In the same way as in the first case above, the $\boldsymbol{u}$ vectors extracted for the second case are:

$$\begin{aligned}\boldsymbol{u} \in \Big\{ \boldsymbol{u}_0 &= \Big[h_A + (\frac{2\tau_B}{L}-1)h_B \quad (2-\frac{2\tau_B}{L})h_B\Big]^T, \boldsymbol{u}_1 = \Big[(\frac{2\tau_B}{L}-1)h_B \quad h_A\Big]^T,\\ \boldsymbol{u}_2 &= \Big[h_A + (\frac{2\tau_B}{L}-1)h_B \quad 0\Big]^T, \boldsymbol{u}_3 = \Big[(\frac{2\tau_B}{L}-1)h_B \quad h_A + (2-\frac{2\tau_B}{L})h_B\Big]^T,\\ \boldsymbol{u}_4 &= \Big[h_A + (2-\frac{2\tau_B}{L})h_B \quad h_B\Big]^T, \boldsymbol{u}_5 = \Big[(2-\frac{2\tau_B}{L})h_B \quad h_A + (\frac{2\tau_B}{L}-1)h_B\Big]^T,\\ \boldsymbol{u}_6 &= \Big[h_A + (2-\frac{2\tau_B}{L})h_B \quad (\frac{2\tau_B}{L}-1)h_B\Big]^T, \boldsymbol{u}_7 = \Big[(2-\frac{2\tau_B}{L})h_B \quad h_A + h_B\Big]^T\Big\}.\end{aligned}$$

It should be noted that, no matter which case we are in, the cluster centres produced by these equations will 'land' on the same spots on the I/Q plane.

$$\boxed{C_{1,0} = \Big\{0, \Big(1-\frac{2\tau_B}{L}\Big)h_B, \frac{2\tau_B}{L}h_B, h_B, h_A, h_A + \Big(1-\frac{2\tau_B}{L}\Big)h_B, h_A + \frac{2\tau_B}{L}h_B, h_A + h_B\Big\}}$$

$$\boxed{C_{1,1} = \Big\{0, \Big(\frac{2\tau_B}{L}-1\Big)h_B, \Big(2-\frac{2\tau_B}{L}\Big)h_B, h_B, h_A, h_A + \Big(\frac{2\tau_B}{L}-1\Big)h_B, h_A + \Big(2-\frac{2\tau_B}{L}\Big)h_B, h_A + h_B\Big\}}$$

Scenario 2

Similarly to scenario 1, but this time for scenario 2, two cases that both produce $N_c = 6$ cluster centres in the I/Q plane are distinguished. These two cases of τ_B are:

- Case 0: $\tau_B = \frac{L}{4}$

- Case 1: $\tau_B = \frac{3L}{4}$

In this scenario, we can either work by using Equations (4.2) and (4.3), or by taking advantage of the fact that scenario 2 is a special case of scenario 1 and use scenario 1 $\boldsymbol{u}$ vectors that correspond to case 0, i.e. $\tau_B < L/2$, plug in $\tau_B = \frac{L}{4}$ and derive the $\boldsymbol{u}$ vectors. Either way, the $\boldsymbol{u}$ vectors for scenario 2 in case of $\tau_B = \frac{L}{4}$ (case 0) are the following:

$$\begin{aligned}\boldsymbol{u} \in \Big\{ &\boldsymbol{u}_0 = \Big[h_A + \frac{h_B}{2} \quad \frac{h_B}{2}\Big]^T, \boldsymbol{u}_1 = \Big[0 \quad h_A + \frac{h_B}{2}\Big]^T, \boldsymbol{u}_2 = \Big[h_A \quad \frac{h_B}{2}\Big]^T,\\ &\boldsymbol{u}_3 = \Big[\frac{h_B}{2} \quad h_A + \frac{h_B}{2}\Big]^T, \boldsymbol{u}_4 = \Big[h_A + h_B \quad \frac{h_B}{2}\Big]^T, \boldsymbol{u}_5 = \Big[\frac{h_B}{2} \quad h_A + \frac{h_B}{2}\Big]^T,\\ &\boldsymbol{u}_6 = \Big[h_A + \frac{h_B}{2} \quad \frac{h_B}{2}\Big]^T, \boldsymbol{u}_7 = \Big[h_B \quad h_A + \frac{h_B}{2}\Big]^T\Big\}.\end{aligned}$$

Similarly, in this case we could take advantage, once again, of the fact that scenario 2 is a special case of scenario 1 and use the $\boldsymbol{u}$ vectors of scenario 1 that correspond to case 1, i.e. $\tau_B > L/2$, plug in $\tau_B = \frac{3L}{4}$ and extract the $\boldsymbol{u}$ vectors. No matter the way chosen, the $\boldsymbol{u}$ vectors for scenario 2 in case of $\tau_B = \frac{3L}{4}$ (case 1) are the following:

$$\begin{aligned}\boldsymbol{u} \in \Big\{ &\boldsymbol{u}_0 = \Big[h_A + \frac{h_B}{2} \quad \frac{h_B}{2}\Big]^T, \boldsymbol{u}_1 = \Big[\frac{h_B}{2} \quad h_A\Big]^T, \boldsymbol{u}_2 = \Big[h_A + \frac{h_B}{2} \quad 0\Big]^T,\\ &\boldsymbol{u}_3 = \Big[\frac{h_B}{2} \quad h_A + \frac{h_B}{2}\Big]^T, \boldsymbol{u}_4 = \Big[h_A + \frac{h_B}{2} \quad h_B\Big]^T, \boldsymbol{u}_5 = \Big[\frac{h_B}{2} \quad h_A + \frac{h_B}{2}\Big]^T,\\ &\boldsymbol{u}_6 = \Big[h_A + \frac{h_B}{2} \quad \frac{h_B}{2}\Big]^T, \boldsymbol{u}_7 = \Big[\frac{h_B}{2} \quad h_A + h_B\Big]^T\Big\}.\end{aligned}$$

Observing all of the above vectors the same set of $N_c = 6$ distinct clusters is distinguished, for both cases of scenario 2:

$$\boxed{C_{2,0} = C_{2,1} = \Big\{0, \frac{h_B}{2}, h_B, h_A, h_A + \frac{h_B}{2}, h_A + h_B\Big\}}$$

Scenario 3

Once again, in scenario 3, two cases can be distinguished, in the same fashion as before:

- Case 0: $\tau_B = 0$
- Case 1: $\tau_B = \frac{L}{2}$

Similar to scenario 2, the $\boldsymbol{u}$ vectors for scenario 3 in case of $\tau_B = 0$ are the following and can be very easily extracted if plugged $\tau_B = 0$ into scenario 1, case 0 $\boldsymbol{u}$ vectors:

$$\begin{aligned}\boldsymbol{u} \in \Big\{ &\boldsymbol{u}_0 = \Big[h_A + h_B \quad 0\Big]^T, \boldsymbol{u}_1 = \Big[0 \quad h_A + h_B\Big]^T, \boldsymbol{u}_2 = \Big[h_A \quad h_B\Big]^T, \boldsymbol{u}_3 = \Big[h_B \quad h_A\Big]^T,\\ &\boldsymbol{u}_4 = \Big[h_A + h_B \quad 0\Big]^T, \boldsymbol{u}_5 = \Big[0 \quad h_A + h_B\Big]^T, \boldsymbol{u}_6 = \Big[h_A \quad h_B\Big]^T, \boldsymbol{u}_7 = \Big[h_B \quad h_A\Big]^T\Big\}.\end{aligned}$$

Similarly, the $\boldsymbol{u}$ vectors for scenario 3 in case $\tau_B = \frac{L}{2}$ are the following and can be extracted similar to the above, if plugged $\tau_B = \frac{L}{2}$ into scenario 1, case 0 or case 1 $\boldsymbol{u}$ vectors:

$$\begin{aligned}\boldsymbol{u} \in \Big\{ &\boldsymbol{u}_0 = \Big[h_A \quad h_B\Big]^T, \boldsymbol{u}_1 = \Big[0 \quad h_A\Big]^T, \boldsymbol{u}_2 = \Big[h_A \quad 0\Big]^T, \boldsymbol{u}_3 = \Big[0 \quad h_A + h_B\Big]^T,\\ &\boldsymbol{u}_4 = \Big[h_A + h_B \quad h_B\Big]^T, \boldsymbol{u}_5 = \Big[h_B \quad h_A\Big]^T, \boldsymbol{u}_6 = \Big[h_A + h_B \quad 0\Big]^T, \boldsymbol{u}_7 = \Big[h_B \quad h_A + h_B\Big]^T\Big\}.\end{aligned}$$

Once again, by observation, the same set of $N_c = 4$ distinct clusters can be distinguished, for both cases of scenario 3:

$$\boxed{C_{3,0} = C_{3,1} = \{0, h_B, h_A, h_A + h_B\}}$$

4.3 Shaping matrix B

To further explain Eq. (4.1) we only need to form 8 $\boldsymbol{B}$ matrices for $\tau_B \leq \frac{L}{2}$ and 8 $\boldsymbol{B}$ matrices for the case of $\tau_B > \frac{L}{2}$, where on both cases they will be based on scenario 1 $\boldsymbol{u}$ vectors, since the other two scenarios are the special cases of scenario 1, as mentioned earlier. The notation used for $\boldsymbol{B}$ matrices corresponds to the equivalent $\boldsymbol{u}$ vector, i.e. $\boldsymbol{B}_{1,3}$ corresponds to vector $\boldsymbol{u}_3$ for $\tau_B > L/2$, while $\boldsymbol{B}_{0,3}$ corresponds to vector $\boldsymbol{u}_3$ for $\tau_B \leq L/2$. The first column of the matrix corresponds to tag B emitting $x_B = [1\ 0]^T$, while the second column corresponds to tag B emitting $x_B = [0\ 1]^T$. Due to the shifted (by T/2) detection, a signal like $x_B = [1\ 1]^T$ will never be observed and that is the reason why there cannot be values on both columns of the matrix; one column will always be filled with zeros. The rows of the matrix correspond to y_0 and y_1 respectively, i.e. the part of the detection window on which the signal from tag B and tag C is present. The elements of these matrices are essentially the coefficients of h_B on each one of the $\boldsymbol{u}$ vectors above. To further explain this last sentence, we trace back to the $\boldsymbol{u}$ vectors of our scenarios and notice that all vectors' elements are given in the form of $\alpha h_A + \beta h_B$, where $\alpha \in \{0, 1\}$ and $\beta \in \{0, 1, (1 - \frac{2\tau_B}{L}), \frac{2\tau_B}{L}, (\frac{2\tau_B}{L} - 1), (2 - \frac{2\tau_B}{L})\}$, depending on the scenario. The matrices for $\tau_B \leq \frac{L}{2}$ are given below:

$$\boldsymbol{B}_{0,0} = \begin{bmatrix} 1 - \frac{2\tau_B}{L} & 0 \\ \frac{2\tau_B}{L} & 0 \end{bmatrix}, \boldsymbol{B}_{0,1} = \begin{bmatrix} 0 & 0 \\ 0 & 1 - \frac{2\tau_B}{L} \end{bmatrix}, \boldsymbol{B}_{0,2} = \begin{bmatrix} 0 & 0 \\ 0 & 1 - \frac{2\tau_B}{L} \end{bmatrix}, \boldsymbol{B}_{0,3} = \begin{bmatrix} 1 - \frac{2\tau_B}{L} & 0 \\ \frac{2\tau_B}{L} & 0 \end{bmatrix},$$

$$\boldsymbol{B}_{0,4} = \begin{bmatrix} 1 & 0 \\ \frac{2\tau_B}{L} & 0 \end{bmatrix}, \boldsymbol{B}_{0,5} = \begin{bmatrix} 0 & \frac{2\tau_B}{L} \\ 0 & 1 - \frac{2\tau_B}{L} \end{bmatrix}, \boldsymbol{B}_{0,6} = \begin{bmatrix} 0 & \frac{2\tau_B}{L} \\ 0 & 1 - \frac{2\tau_B}{L} \end{bmatrix}, \boldsymbol{B}_{0,7} = \begin{bmatrix} 1 & 0 \\ \frac{2\tau_B}{L} & 0 \end{bmatrix}.$$

In the same fashion as before, the matrices for $\tau_B > \frac{L}{2}$ are given below:

$$\boldsymbol{B}_{1,0} = \begin{bmatrix} \frac{2\tau_B}{L} - 1 & 0 \\ 2 - \frac{2\tau_B}{L} & 0 \end{bmatrix}, \boldsymbol{B}_{1,1} = \begin{bmatrix} 0 & \frac{2\tau_B}{L} - 1 \\ 0 & 0 \end{bmatrix}, \boldsymbol{B}_{1,2} = \begin{bmatrix} 0 & \frac{2\tau_B}{L} - 1 \\ 0 & 0 \end{bmatrix}, \boldsymbol{B}_{1,3} = \begin{bmatrix} \frac{2\tau_B}{L} - 1 & 0 \\ 2 - \frac{2\tau_B}{L} & 0 \end{bmatrix}$$

$$\boldsymbol{B}_{1,4} = \begin{bmatrix} 2 - \frac{2\tau_B}{L} & 0 \\ 1 & 0 \end{bmatrix}, \boldsymbol{B}_{1,5} = \begin{bmatrix} 0 & 2 - \frac{2\tau_B}{L} \\ 0 & \frac{2\tau_B}{L} - 1 \end{bmatrix}, \boldsymbol{B}_{1,6} = \begin{bmatrix} 0 & 2 - \frac{2\tau_B}{L} \\ 0 & \frac{2\tau_B}{L} - 1 \end{bmatrix}, \boldsymbol{B}_{1,7} = \begin{bmatrix} 2 - \frac{2\tau_B}{L} & 0 \\ 1 & 0 \end{bmatrix}.$$

From the above matrices it can be observed that all of them exist two times each. For instance, matrix $\boldsymbol{B}_{0,0} = \boldsymbol{B}_{0,3}$, $\boldsymbol{B}_{0,1} = \boldsymbol{B}_{0,2}$ etc., which means that each matrix has a probability of appearance equal to $\frac{1}{4}$. That holds if it is assumed, without loss of generality, that all matrices have an equal chance of occurring. The same holds for the matrices in the case of $\tau_B > \frac{L}{2}$. It is also worth mentioning that these two sets of matrices hold for all scenarios, with respect to their time offset, only scenario 1 equations are needed to extract them, since it is the master scenario.

As an example, let us assume that the time offset $\tau_B = \frac{3L}{4}$. Since, $\tau_B = \frac{3L}{4} > \frac{L}{2}$, then the second set of matrices will be chosen that holds for values of $\tau_B > \frac{L}{2}$. Plugging $\tau_B = \frac{3L}{4}$ into that second set of matrices and using Equation (4.1) we can return back to the equivalent system model that uses the $\boldsymbol{u}$ vectors, thus proving how that equality is obtained.

In all the above equations the following cases were considered in order to extract the $\boldsymbol{u}$ vectors. Going through the following cases one by one, while using Equations (4.2) and (4.3), the way it was done in Figure 4.1, but with different time offset τ_B values each time, all the above $\boldsymbol{u}$ vectors will eventually be yielded.

	Tag A	Tag B	Tag C
$\boldsymbol{u}_0$	S_0	S_0	S_0
$\boldsymbol{u}_1$	S_1	S_1	S_0
$\boldsymbol{u}_2$	S_0	S_1	S_0
$\boldsymbol{u}_3$	S_1	S_0	S_0
$\boldsymbol{u}_4$	S_0	S_0	S_1
$\boldsymbol{u}_5$	S_1	S_1	S_1
$\boldsymbol{u}_6$	S_0	S_1	S_1
$\boldsymbol{u}_7$	S_1	S_0	S_1

Table 4.1: $\boldsymbol{u}$ vectors' cases for all scenarios.

Concluding, it is noticed that all $\boldsymbol{u}$ vectors of any scenario can be produced from scenario 1 clusters, thus for convenience cluster centres of scenario 1 will be denoted as:

$$\mathcal{U} = \begin{cases} C_{1,0}, & \text{if} \quad 0 \leq \tau_B \leq \frac{L}{2} \\ C_{1,1}, & \text{if} \quad \frac{L}{2} < \tau_B \leq L-1, \end{cases} \tag{4.4}$$

4.4 Signal to Noise Ratio (SNR)

To begin with, it is mandatory to explain how the additive noise to our system was modelled. It is already declared that $y[k] = h_A x_A[k] + h_B x_B[k] + n[k]$, where $n[k] \sim \mathcal{CN}(0, 2\sigma_n^2)$. An analysis is going to be performed on how the different expressions for the noise variances are connected to each other, and how they were derived. Focusing on the noise samples, after the process of match filtering we get:

$$\sum_{k=0}^{\frac{L}{2}-1} \frac{n[k]}{L/2} = \frac{2}{L} \sum_{k=0}^{\frac{L}{2}-1} n[k],$$

where,

$$\sum_{k=0}^{\frac{L}{2}-1} n[k] \sim \mathcal{CN}(0, \frac{L}{2} 2\sigma_n^2) \equiv \mathcal{CN}(0, L\sigma_n^2),$$

hence,

$$\frac{2}{L} \sum_{k=0}^{\frac{L}{2}-1} n[k] \sim \mathcal{CN}(0, \frac{4}{L^2} L\sigma_n^2) \equiv \mathcal{CN}(0, \frac{4}{L}\sigma_n^2) \equiv \mathcal{CN}(0, \sigma^2).$$

It was stated earlier that $n_0 \perp n_1$ with,

$$\boldsymbol{n} = \begin{bmatrix} n_0 \\ n_1 \end{bmatrix} = \begin{bmatrix} \sum\limits_{k=0}^{\frac{L}{2}-1} \frac{n[k]}{L/2} \\ \sum\limits_{k=\frac{L}{2}}^{L-1} \frac{n[k]}{L/2} \end{bmatrix}.$$

It can finally be deduced that, $\boldsymbol{n} \sim \mathcal{CN}(0, \frac{4}{L}\sigma_n^2 \boldsymbol{I}_2) \equiv \mathcal{CN}(0, \sigma^2 \boldsymbol{I}_2)$.

As seen earlier in Section 4.3, all possible $\boldsymbol{B}$ matrices are a function of the offset τ_B and the oversampling factor L. At this point, it is needed to state that τ_B is a random variable with discrete uniform distribution, for the sake of our computations, in the range of $[0, L-1]$, i.e. $\tau_B \sim \mathcal{U}(0, L-1)$, thus $\mathbb{E}[\tau_B] = \frac{L-1}{2}$. As first case, only the signal transmitted from tag A is considered as a useful signal source, hence, tag B signal is treated as interference in the calculations below:

$$\begin{aligned} SNR_A = \frac{\mathbb{E}[||h_A \boldsymbol{x}_A||_2^2]}{\mathbb{E}[||\boldsymbol{B} h_B \boldsymbol{x}_B + \boldsymbol{n}||_2^2]} &= \frac{|h_A|^2}{\mathbb{E}[(\boldsymbol{B} h_B \boldsymbol{x}_B + \boldsymbol{n})^H (\boldsymbol{B} h_B \boldsymbol{x}_B + \boldsymbol{n})]} \\ &= \frac{|h_A|^2}{\mathbb{E}[(h_B^* \boldsymbol{x}_B^T \boldsymbol{B}^T + \boldsymbol{n}^H)(\boldsymbol{B} h_B \boldsymbol{x}_B + \boldsymbol{n})]} \\ &= \frac{|h_A|^2}{\mathbb{E}[(h_B^* \boldsymbol{x}_B^T \boldsymbol{B}^T \boldsymbol{B} \boldsymbol{x}_B h_B + h_B^* \boldsymbol{x}_B^T \boldsymbol{B}^T \boldsymbol{n} + \boldsymbol{n}^H \boldsymbol{B} h_B \boldsymbol{x}_B + \boldsymbol{n}^H \boldsymbol{n})]} \\ &= \frac{|h_A|^2}{\mathbb{E}[h_B^* h_B \boldsymbol{x}_B^T \boldsymbol{B}^T \boldsymbol{B} \boldsymbol{x}_B] + \mathbb{E}[\boldsymbol{n}^H \boldsymbol{n}]} \\ &= \frac{|h_A|^2}{\mathbb{E}[h_B^* h_B]\, \mathbb{E}[\boldsymbol{x}_B^T \boldsymbol{B}^T \boldsymbol{B} \boldsymbol{x}_B] + 2\sigma^2} = \frac{|h_A|^2}{|h_B|^2\, \mathbb{E}[\boldsymbol{x}_B^T \boldsymbol{B}^T \boldsymbol{B} \boldsymbol{x}_B] + 2\sigma^2}. \end{aligned} \tag{4.5}$$

Observing the above equation it becomes clear that the term of $\mathbb{E}[\boldsymbol{x}_B^T \boldsymbol{B}^T \boldsymbol{B} \boldsymbol{x}_B]$ needs to be calculated, which can be done by using the law of iterated expectation like below. As a reminder, the law of iterated expectation was described and proved in Chapter 3.

$$\begin{aligned} \mathbb{E}[\boldsymbol{x}_B^T \boldsymbol{B}^T \boldsymbol{B} \boldsymbol{x}_B] &= \mathop{\mathbb{E}}_{\boldsymbol{x}_B} \{ \underbrace{\mathop{\mathbb{E}}_{\boldsymbol{B}|\boldsymbol{x}_B} [\boldsymbol{x}_B^T \boldsymbol{B}^T \boldsymbol{B} \boldsymbol{x}_B]}_{g(\boldsymbol{x}_B)} \} = \frac{1}{2} g\Big(\boldsymbol{x}_B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}\Big) + \frac{1}{2} g\Big(\boldsymbol{x}_B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}\Big) \\ &= \frac{1}{2}[0 \quad 1]\, \mathbb{E}\Big\{\boldsymbol{B}^T \boldsymbol{B} | \boldsymbol{x}_B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}\Big\} \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \frac{1}{2}[1 \quad 0]\, \mathbb{E}\Big\{\boldsymbol{B}^T \boldsymbol{B} | \boldsymbol{x}_B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}\Big\} \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \end{aligned} \tag{4.6}$$

Since the previous expression was simplified a bit more, there is now left to compute the expected value of $\mathbb{E}[\boldsymbol{B}^T \boldsymbol{B} | \boldsymbol{x}_B]$. Once again, the law of iterated expectation will be used to reach the final result:

$$\mathbb{E}[\boldsymbol{B}^T \boldsymbol{B} | \boldsymbol{x}_B] = \mathop{\mathbb{E}}_{\tau_B} \{ \mathop{\mathbb{E}}_{\boldsymbol{B}^T \boldsymbol{B} | \boldsymbol{x}_B, \tau_B} [\boldsymbol{B}^T \boldsymbol{B} | \boldsymbol{x}_B, \tau_B] \} \overset{!}{=} \mathop{\mathbb{E}}_{\tau_B} \Big\{ \sum_{\boldsymbol{B}^T \boldsymbol{B}} \frac{1}{4} \boldsymbol{B}^T \boldsymbol{B} \Big\}, \tag{4.7}$$

where at (!), the fraction $\frac{1}{4}$ corresponds to the conditional probability:

$$\mathbb{P}(\boldsymbol{B}|\boldsymbol{x}_B) = \frac{\mathbb{P}(\boldsymbol{x}_B, \boldsymbol{B})}{\mathbb{P}(\boldsymbol{x}_B)} = \frac{1/8}{1/2} = \frac{1}{4}.$$

All possible $\boldsymbol{B}$ matrices can be found at any time based on the value of $\boldsymbol{x}_B$, as described in Section 4.3. If $\boldsymbol{x}_B = [0 \quad 1]^T$, then the possible $\boldsymbol{B}$ matrices obtained are, $\boldsymbol{B}_{0,1}/\boldsymbol{B}_{0,2}$, $\boldsymbol{B}_{0,5}/\boldsymbol{B}_{0,6}$, for $\tau_B \leq L/2$, and $\boldsymbol{B}_{1,1}/\boldsymbol{B}_{1,2}$, $\boldsymbol{B}_{1,5}/\boldsymbol{B}_{1,6}$, for $\tau_B > L/2$ as well, totalling 8 (duplicate) matrices, each with a probability of 1/4, as stated in Section 4.3. If $\boldsymbol{x}_B = [1 \quad 0]^T$, then the possible $\boldsymbol{B}$ matrices obtained are, $\boldsymbol{B}_{0,0}/\boldsymbol{B}_{0,3}$, $\boldsymbol{B}_{0,4}/\boldsymbol{B}_{0,7}$, for $\tau_B \leq L/2$, and $\boldsymbol{B}_{1,0}/\boldsymbol{B}_{1,3}$, $\boldsymbol{B}_{1,4}/\boldsymbol{B}_{1,7}$, for $\tau_B > L/2$ as well, totalling 8 (duplicate) matrices, once again each of them occurring with a probability of 1/4. By calculating the products $\boldsymbol{B}_i^T \boldsymbol{B}_i$, for each case of $\boldsymbol{x}_B$, summing them and then dividing by 4, a matrix that is only a function of the r.v. τ_B occurs, on which we have to calculate the expectation, obviously with respect to τ_B. After extensive calculations, it is concluded that for $\boldsymbol{x}_B = [0 \quad 1]^T$:

$$\begin{aligned}\mathop{\mathbb{E}}_{\tau_B}\Big\{\sum_{\boldsymbol{B}^T\boldsymbol{B}} \frac{1}{4}\boldsymbol{B}^T\boldsymbol{B}\Big\} &= \frac{1}{4}\begin{bmatrix}0 & 0 \\ 0 & \mathbb{E}[\frac{8L^2+24\tau_B^2-24\tau_B L}{L^2}]\end{bmatrix} = \frac{1}{4}\begin{bmatrix}0 & 0 \\ 0 & 8+\frac{24}{L^2}\mathbb{E}[\tau_B^2]-\frac{24}{L}\mathbb{E}[\tau_B]\end{bmatrix} \\ &= \frac{1}{4}\begin{bmatrix}0 & 0 \\ 0 & 8+\frac{24}{L^2}\frac{2L^2-3L+1}{6}-\frac{24}{L}\frac{L-1}{2}\end{bmatrix} = \frac{1}{4}\begin{bmatrix}0 & 0 \\ 0 & \frac{4L^2+4}{L^2}\end{bmatrix},\end{aligned} \tag{4.8}$$

while for $\boldsymbol{x}_B = [1 \quad 0]^T$:

$$\begin{aligned}\mathop{\mathbb{E}}_{\tau_B}\Big\{\sum_{\boldsymbol{B}^T\boldsymbol{B}} \frac{1}{4}\boldsymbol{B}^T\boldsymbol{B}\Big\} &= \frac{1}{4}\begin{bmatrix}\mathbb{E}[\frac{12L^2-24\tau_B L+24\tau_B^2}{L^2}] & 0 \\ 0 & 0\end{bmatrix} = \frac{1}{4}\begin{bmatrix}12+\frac{24}{L^2}\mathbb{E}[\tau_B^2]-\frac{24}{L}\mathbb{E}[\tau_B] & 0 \\ 0 & 0\end{bmatrix} \\ &= \frac{1}{4}\begin{bmatrix}12+\frac{24}{L^2}\frac{2L^2-3L+1}{6}-\frac{24}{L}\frac{L-1}{2} & 0 \\ 0 & 0\end{bmatrix} = \frac{1}{4}\begin{bmatrix}\frac{8L^2+4}{L^2} & 0 \\ 0 & 0\end{bmatrix}.\end{aligned} \tag{4.9}$$

Substituting Equations (4.8) and (4.9) into Eq. (4.6), it is yielded:

$$\mathbb{E}[\boldsymbol{x}_B^T\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{x}_B] = \frac{1}{2}[0 \quad 1]\frac{1}{4}\begin{bmatrix}0 & 0 \\ 0 & \frac{4L^2+4}{L^2}\end{bmatrix}\begin{bmatrix}0 \\ 1\end{bmatrix} + \frac{1}{2}[1 \quad 0]\frac{1}{4}\begin{bmatrix}\frac{8L^2+4}{L^2} & 0 \\ 0 & 0\end{bmatrix}\begin{bmatrix}1 \\ 0\end{bmatrix} = \frac{3L^2+2}{2L^2}. \tag{4.10}$$

Further substituting Eq. (4.10) into Eq. (4.5) it is yielded:

$$SNR_A = \frac{|h_A|^2}{\frac{3L^2+2}{2L^2}|h_B|^2 + 2\sigma^2}. \tag{4.11}$$

Working in the same manner as before, to compute the signal to noise ratio of tag B, SNR_B, considering tag A signal as interference to the useful signal, i.e. tag B signal, in the calculations below.

$$SNR_B = \frac{\mathbb{E}[||\boldsymbol{B}h_B\boldsymbol{x}_B||_2^2]}{\mathbb{E}[||h_A\boldsymbol{x}_A + \boldsymbol{n}||_2^2]} = \ldots = \frac{\mathbb{E}[h_B^* h_B]\,\mathbb{E}[\boldsymbol{x}_B^T\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{x}_B]}{\mathbb{E}[h_A^* h_A] + \mathbb{E}[\boldsymbol{n}^H\boldsymbol{n}]} = \frac{\frac{3L^2+2}{2L^2}|h_B|^2}{|h_A|^2 + 2\sigma^2}. \tag{4.12}$$

In case joint detection needs to be performed, both the signal transmitted by tag A and tag B have to be considered as useful signal sources. Below the calculation of the $SNR_{A,B}$ is presented.

$$
\begin{aligned}
SNR_{A,B} &= \frac{\mathbb{E}[||h_A\boldsymbol{x}_A + \boldsymbol{B}h_B\boldsymbol{x}_B||_2^2]}{\mathbb{E}[||\boldsymbol{n}||_2^2]} = \frac{\mathbb{E}[(h_A\boldsymbol{x}_A + \boldsymbol{B}h_B\boldsymbol{x}_B)^H(h_A\boldsymbol{x}_A + \boldsymbol{B}h_B\boldsymbol{x}_B)]}{2\sigma^2} \\
&= \frac{\mathbb{E}[h_A^* h_A \boldsymbol{x}_A^T\boldsymbol{x}_A] + \mathbb{E}[h_A^*\boldsymbol{x}_A^T\boldsymbol{B}h_B\boldsymbol{x}_B] + \mathbb{E}[h_B^*\boldsymbol{x}_B^T\boldsymbol{B}^T h_A\boldsymbol{x}_A] + \mathbb{E}[h_B^*\boldsymbol{x}_B^T\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{x}_B h_B]}{2\sigma^2} \\
&= \frac{|h_A|^2 + \frac{3L^2+2}{2L^2}|h_B|^2 + \boldsymbol{x}_A^T\,\mathbb{E}[h_A^*h_B]\,\mathbb{E}[\boldsymbol{B}\boldsymbol{x}_B] + \mathbb{E}[h_B^*h_A]\,\mathbb{E}[\boldsymbol{x}_B^T\boldsymbol{B}^T]\boldsymbol{x}_A^T}{2\sigma^2}.
\end{aligned}
$$

To simplify the calculations it is considered that $h_A \perp h_B$. This means that the tags are not placed close to each other and their distance from the reader is not equal. Furthermore, their line of sight (l.o.s.) to the reader could also be different. All the above assumptions aid our conclusion, that the channel coefficients of the tags, are two independent random variables. Hence, $SNR_{A,B}$ is defined as:

$$
SNR_{A,B} = \frac{|h_A|^2 + \frac{3L^2+2}{2L^2}|h_B|^2}{2\sigma^2}. \tag{4.13}
$$

Since, $h_A \perp h_B \Rightarrow \mathbb{E}[h_B^*h_A] = \mathbb{E}[h_B^*]\,\mathbb{E}[h_A] = 0$. Similarly, $\mathbb{E}[h_A^*h_B] = \mathbb{E}[h_A^*]\,\mathbb{E}[h_B] = 0$. Swapping $\sigma^2 = \frac{4}{L}\sigma_n^2$ in the above equation it is obtained:

$$
\sigma_n^2 = \frac{L(|h_A|^2 + \frac{3L^2+2}{2L^2}|h_B|^2)}{8SNR_{A,B}}. \tag{4.14}
$$

It is also needed to compute the relation of tag's A signal with respect to noise $\boldsymbol{n}$ and with respect to tag's B signal. With respect to tag's B signal first it is obtained:

$$
\frac{\mathbb{E}[||h_A\boldsymbol{x}_A||_2^2]}{\mathbb{E}[||\boldsymbol{B}h_B\boldsymbol{x}_B||_2^2]} = \frac{|h_A|^2}{\frac{3L^2+2}{2L^2}|h_B|^2},
$$

while, with respect to noise $\boldsymbol{n}$ it is obtained:

$$
\frac{\mathbb{E}[||h_A\boldsymbol{x}_A||_2^2]}{\mathbb{E}[||\boldsymbol{n}||_2^2]} = \frac{|h_A|^2}{2\sigma^2}.
$$

Last, it is needed to compute the relation of tag's B signal with respect to noise $\boldsymbol{n}$ and with respect to tag's A signal. With respect to tag's A signal first it is obtained:

$$
\frac{\mathbb{E}[||\boldsymbol{B}h_B\boldsymbol{x}_B||_2^2]}{\mathbb{E}[||h_A\boldsymbol{x}_A||_2^2]} = \frac{\frac{3L^2+2}{2L^2}|h_B|^2}{|h_A|^2},
$$

while, with respect to noise $\boldsymbol{n}$ it is obtained:

$$
\frac{\mathbb{E}[||\boldsymbol{B}h_B\boldsymbol{x}_B||_2^2]}{\mathbb{E}[||\boldsymbol{n}||_2^2]} = \frac{\frac{3L^2+2}{2L^2}|h_B|^2}{2\sigma^2}.
$$

In the figures below, real world data is offered of the clusters that are generated on each scenario. Later in this thesis, it will be shown that the patterns observed in Figures 4.3 and 4.4 will prove to be useful, since they are going to be used in order to solve the labelling problem and estimate the channel coefficients as well as the time offset τ_B of tag B.

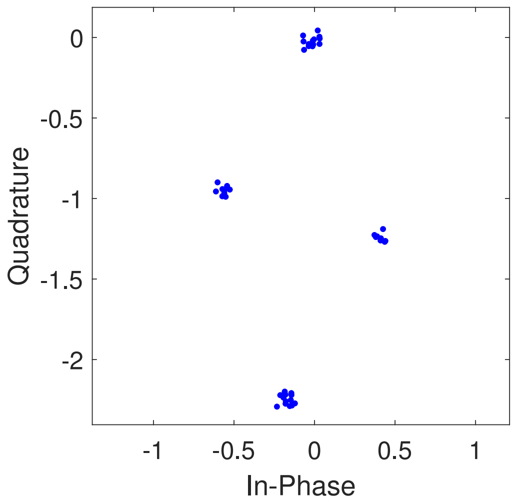


Figure 4.2: I/Q constellation clusters for scenario 3 at $SNR_{A,B} = 30$ dB.

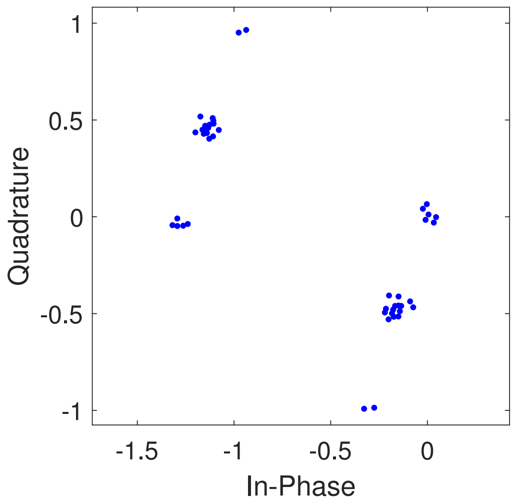


Figure 4.3: I/Q constellation clusters for scenario 2 at $SNR_{A,B} = 30$ dB.

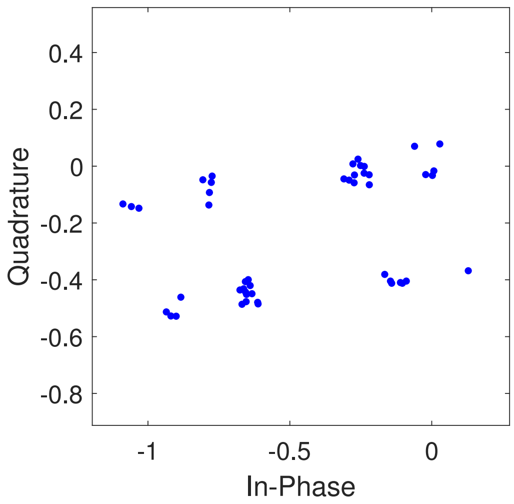


Figure 4.4: I/Q constellation clusters for scenario 1 at $SNR_{A,B} = 30$ dB.

4.5 Discussion

To summarize, in this chapter, an equivalent system model to the one we presented at the end of Chapter 3 was provided, where this one is presented in a more compact form using vectors. Using the system model of the previous chapter and by alternating the values of the time offset of tag B τ_B, 3 scenarios were discovered that defined how the clusters were formed. Based on those clusters that were expressed in vector form we were able to derive the vector equivalent system model using a shaping matrix $\boldsymbol{B}$ that only depends on τ_B to be fully calculated at any given time.

Next, the signal to noise ratios for different cases of useful signal and noise were calculated. As an example, the SNR relation when we are interested on both tags' signal and only the additive noise is considered as interference. Another relation is when we are interested only in tag A and we consider the signal from tag B as interference, and so on and so forth. The reason all these relations were calculated is due to the fact that, the detection scheme relies closely on these relations.

Chapter 5

CSI Estimation with Clustering

In the following pages we will provide an already widely used method for DC offset estimation, a proposed algorithm for channel estimation that takes advantage of the powerful affinity propagation algorithm and the clusters patterns that form especially in high SNR environments and finally an offset estimation method.

5.1 DC Offset Estimation

In a real time RFID application, the DC offset component can be estimated during a Gen2-defined interval before the tag starts switching. This interval is known to the reader and is defined by Gen2 as T_1; its duration depends on the tag's data rate for FM0 encoding. Tag is absorbing energy with corresponding reflection coefficient close to zero; thus, the reflected signal corresponding to one of the two tag load states, can be estimated by averaging the received samples acquired in interval T_1. The received signal during T_1 is given by:

$$y_{T_1}[k] = A_{dc} + n[k], \; k = 0, \ldots, L_{T_1} - 1$$

where $L_{T_1} = \frac{T_1}{T_s}$ is the length of the interval in samples.

The ML estimate of $\widehat{A_{dc}}$ is found by:

$$\begin{aligned}
\widehat{A_{dc}} &= \underset{A_{dc} \in \mathbb{C}}{\operatorname{argmax}} \, f(\boldsymbol{y} | A_{dc}) \\
&= \underset{A_{dc} \in \mathbb{C}}{\operatorname{argmax}} \ln \left(\prod_{k=0}^{L_{T_1}-1} \frac{1}{2\pi\sigma_n^2} e^{\frac{|y[k]-A_{dc}|^2}{2\sigma_n^2}} \right) \\
&= \underset{A_{dc} \in \mathbb{C}}{\operatorname{argmax}} \sum_{k=0}^{L_{T_1}-1} |y[k] - A_{dc}|^2 \\
&= \underset{A_{dc} \in \mathbb{C}}{\operatorname{argmax}} \sum_{k=0}^{L_{T_1}-1} (y[k] - A_{dc})(y[k] - A_{dc})^* \\
&= \underset{A_{dc} \in \mathbb{C}}{\operatorname{argmax}} \sum_{k=0}^{L_{T_1}-1} (y[k]y[k]^* - y[k]A_{dc}^* - A_{dc}y[k]^* + A_{dc}A_{dc}^*)
\end{aligned}$$

differentiating with respect to A_{dc} we get:

$$\sum_{k=0}^{L_{T_1}-1} A_{dc}^* - y[k]^* = 0$$
$$\sum_{k=0}^{L_{T_1}-1} y[k] = L_{T_1} A_{dc}$$
$$\widehat{A_{dc}} = \frac{1}{L_{T_1}} \sum_{k=0}^{L_{T_1}-1} y[k].$$

The estimated DC offset component, which is equal to the arithmetic mean of the received samples, is then subtracted from each sample offering Equation (3.2).

5.2 Cluster Labelling

On this section we are going to solve the labelling problem in scenarios 1 and 2, since it is a little bit more complicated to solve it under scenario 3, where we only have 4 clusters and the line fitting method will not work. The analysis for the methods applied in scenario 3 will be presented later on during this thesis. A successful identification of the cluster labels enables us to perform the channel estimation procedure with great precision, especially in high SNR environments.

- *Cluster of* h_B:

On the first part of this section we will try to locate where the h_B cluster will lie on the I/Q plane at any given scenario (1 or 2). According to Section 4.2, in scenario 2 we will notice the following clusters:

$$C_2 = \left\{0, \frac{h_B}{2}, h_B, h_A, h_A + \frac{h_B}{2}, h_A + h_B\right\}.$$

Assuming a completely noiseless environment, we observe that the clusters, $\{0, \frac{h_B}{2}, h_B\}$, all lie upon a straight line passing from 0, $\frac{h_B}{2}$ and h_B, at this exact order, since:

$$0 < \frac{1}{2} < 1 \Rightarrow 0 < \left|\frac{h_B}{2}\right| < |h_B|.$$

It is obvious that, the remaining clusters $\{h_A, h_A + \frac{h_B}{2}, h_A + h_B\}$, also lie upon a line parallel to the first one but their order depends on the value of h_B and we will further elaborate on this later on. The same parallel lines pattern also holds for scenario 1 in the exact same manner, where instead we observe the following sets of clusters, $C_{1,0}$ and $C_{1,1}$, as given in Section 4.2:

$$C_{1,0} = \left\{0, \left(1 - \frac{2\tau_B}{L}\right)h_B, \frac{2\tau_B}{L}h_B, h_B, h_A, h_A + \left(1 - \frac{2\tau_B}{L}\right)h_B, h_A + \frac{2\tau_B}{L}h_B, h_A + h_B\right\},$$
$$C_{1,1} = \left\{0, \left(\frac{2\tau_B}{L} - 1\right)h_B, \left(2 - \frac{2\tau_B}{L}\right)h_B, h_B, h_A, h_A + \left(\frac{2\tau_B}{L} - 1\right)h_B, h_A + \left(2 - \frac{2\tau_B}{L}\right)h_B, h_A + h_B\right\}.$$

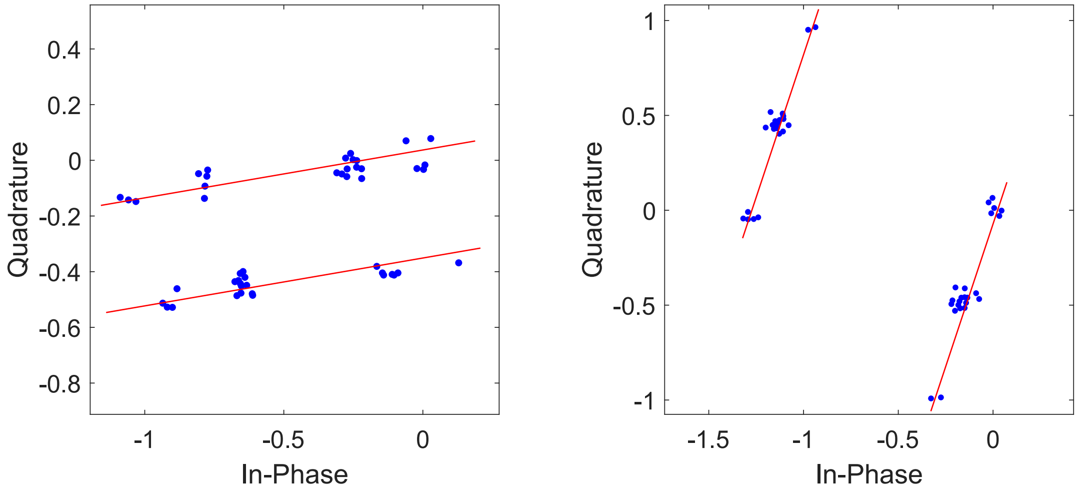
On both sets, the clusters of the form βh_B, where $\beta \in \{0, (1 - \frac{2\tau_B}{L}), \frac{2\tau_B}{L}, 1\}$ for case 0, where $\tau_B < \frac{L}{2}$, and $\beta \in \{0, (\frac{2\tau_B}{L} - 1), (2 - \frac{2\tau_B}{L}), 1\}$ for case 1, where $\tau_B > \frac{L}{2}$, once again lie upon the same straight line crossing 0 towards h_B, in the order that the β values are given for each case. That holds, since for case 0, where $\tau_B < \frac{L}{2}$:

$$0 < \left(1 - \frac{2\tau_B}{L}\right) < \frac{2\tau_B}{L} < 1 \Rightarrow 0 < \left|\left(1 - \frac{2\tau_B}{L}\right)h_B\right| < \left|\frac{2\tau_B}{L}h_B\right| < |h_B|,$$

while for case 1, where $\tau_B > \frac{L}{2}$:

$$0 < \left(\frac{2\tau_B}{L} - 1\right) < \left(2 - \frac{2\tau_B}{L}\right) < 1 \Rightarrow 0 < \left|\left(\frac{2\tau_B}{L} - 1\right)h_B\right| < \left|\left(2 - \frac{2\tau_B}{L}\right)h_B\right| < |h_B|.$$

Once again, the remaining clusters of each case of scenario 1, all lie on a line parallel to the first one in a similar fashion as in scenario 2, but their order depends on the value of h_B. This way, we can estimate the values of h_B at any of the scenarios 1 and 2, as we will demonstrate later on. In Figure 5.1 below, a graphical representation of what we are trying to describe is offered.



(a) Scenario 1

(b) Scenario 2

Figure 5.1: Parallel lines pattern as observed in scenario 1 in (a) and in scenario 2 in (b), under CSCN additive white noise.

- *Cluster of* h_A:

On the second part of this section we will try to locate where the h_A cluster will lie on the I/Q plane at any given scenario (1 or 2). In order to be able to find the labelling of the h_A clusters, we must distinguish two cases for h_B. If $\Re\{h_B\} > 0$, then the first cluster we will meet while scanning the I/Q plane from negative in-phase components to the most positive ones will be cluster h_A, which makes complete sense, since for scenario 2:

$$\Re\{h_A\} < \Re\left\{h_A + \frac{h_B}{2}\right\} < \Re\{h_A + h_B\}.$$

Identically to scenario 2, the exact same holds for scenario 1 for a constant time offset value τ_B, under case 0, where $\tau_B < L/2$:

$$\Re\{h_A\} < \Re\left\{h_A + \left(1 - \frac{2\tau_B}{L}\right)h_B\right\} < \Re\left\{h_A + \frac{2\tau_B}{L}h_B\right\} < \Re\{h_A + h_B\},$$

while, under case 1, where $\tau_B > L/2$:

$$\Re\{h_A\} < \Re\left\{h_A + \left(\frac{2\tau_B}{L} - 1\right)h_B\right\} < \Re\left\{h_A + \left(2 - \frac{2\tau_B}{L}\right)h_B\right\} < \Re\{h_A + h_B\},$$

On the other hand, if $\Re\{h_B\} < 0$, then the first cluster we will meet will be $h_A + h_B$, since:

$$\Re\{h_A + h_B\} < \Re\left\{h_A + \frac{h_B}{2}\right\} < \Re\{h_A\}.$$

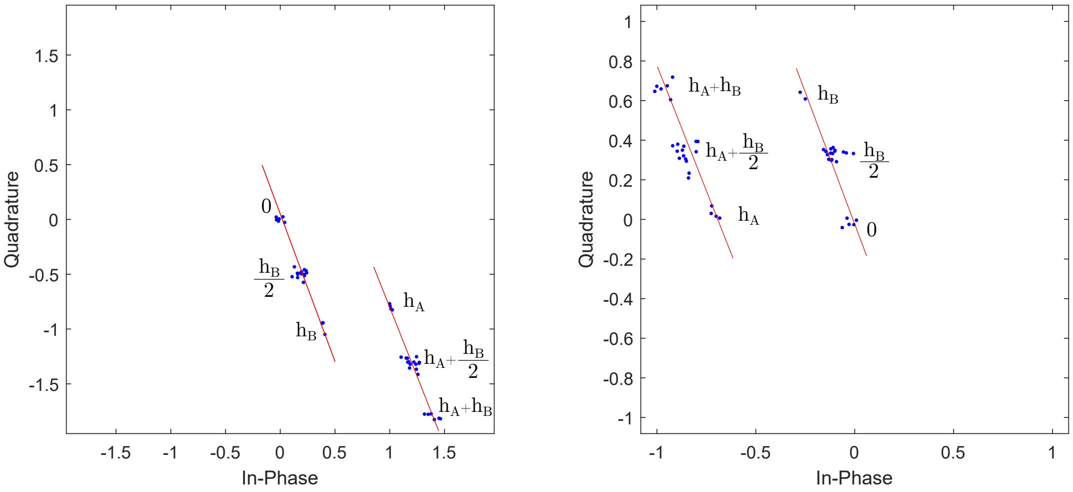
Equivalently in scenario 1, for a constant time offset value τ_B, under case 0, where $\tau_B < L/2$:

$$\Re\{h_A\} > \Re\left\{h_A + \left(1 - \frac{2\tau_B}{L}\right)h_B\right\} > \Re\left\{h_A + \frac{2\tau_B}{L}h_B\right\} > \Re\{h_A + h_B\}.$$

while, under case 1, where $\tau_B > L/2$:

$$\Re\{h_A\} > \Re\left\{h_A + \left(\frac{2\tau_B}{L} - 1\right)h_B\right\} > \Re\left\{h_A + \left(2 - \frac{2\tau_B}{L}\right)h_B\right\} > \Re\{h_A + h_B\}.$$

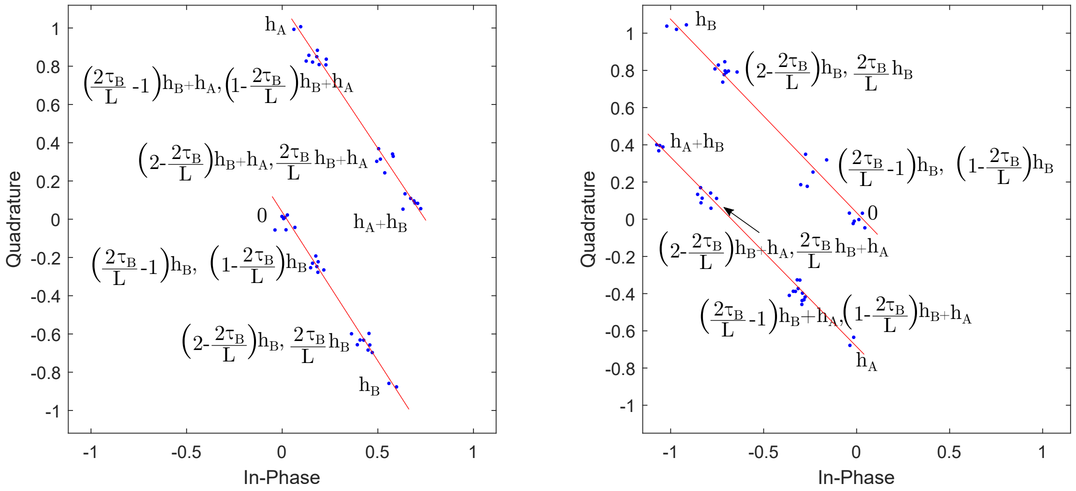
On the first case, $\Re\{h_B\} > 0$, on both scenarios, we just have the estimation of h_A, while on the second case, $\Re\{h_B\} < 0$, we have to subtract by the estimate of h_B to obtain an estimate for h_A. To further aid our estimation of h_A, after having estimated h_B, we notice that under any case and under any scenario all we need is the cluster centre with the minimum real part and the cluster centre with the maximum real part, from the cluster centres that are parallel to the h_B ones. These two cluster centres are h_A and $h_A + h_B$, where which one is max and which one is min is defined with respect to $\Re\{h_B\}$, as it can also be seen on the figures below. We can then simply add these two cluster centres, subtract from them the estimate of h_B and divide by 2 to take the average, which basically leaves us with a stronger estimate of h_A, since we used most of our data. In addition to that, if we are under scenario 2, we can also use the cluster centre of $h_A + \frac{h_B}{2}$ to further enhance the estimation. The latter cannot be applied in scenario 1, since we do not know a priori the value of the offset τ_B. Figures 5.2 and 5.3 below, help us better understand the concept of cluster labels we are trying to explain.



(a) Scenario 2, for $\Re\{h_B\} > 0$.

(b) Scenario 2, for $\Re\{h_B\} < 0$.

Figure 5.2: Scenario 2 cluster labels for two different cases of $\Re\{h_B\}$.



(a) Scenario 1, for $\Re\{h_B\} > 0$.

(b) Scenario 1, for $\Re\{h_B\} < 0$.

Figure 5.3: Scenario 1 cluster labels for two different cases of $\Re\{h_B\}$.

5.3 Channel Estimation

Having solved the cluster labelling problem using line fitting as demonstrated, we can now use the cluster labels and obtain our channel coefficient estimates. The algorithm below, demonstrates the big picture of the most important steps we make in order to obtain the channel coefficient estimates.

```
Algorithm 2: Channel Estimation
  Input: Set $\mathcal{C}$ of cluster centres.
  Output: Estimated channel coefficients $\widehat{h_A}, \widehat{h_B}$.
$\epsilon \leftarrow 10^{-3}$                        // Choose an arbitrarily small value
for $k$ in $\mathcal{C}/\{\mathcal{C}_0\}$ do
    $y = \frac{\Im\{\mathcal{C}_k\}}{\Re\{\mathcal{C}_k\}}(x - \Re\{\mathcal{C}_k\}) + \Im\{\mathcal{C}_k\}$  // Form the line passing from $\mathcal{C}_0$ and $\mathcal{C}_k$
    $\mathcal{A} = \text{find}(d(y, \mathcal{C}_k) < \epsilon)$        // Set of clusters $\mathcal{C}_i$ close to line y
    if $|\mathcal{A}| == \frac{N_c}{2} - 1$ then
        return $\mathcal{A}$
$\alpha^* = argmax_{\alpha \in \mathcal{A}}(||\mathcal{C}_\alpha||_2)$
$\widehat{h_B} = \mathcal{C}_{\alpha^*}$
$\widehat{h_A} = (argmin(||\mathcal{C} - \mathcal{A}||_2) + argmax(||\mathcal{C} - \mathcal{A}||_2) - \widehat{h_B})/2;$
```

5.3.1 Line Fitting

In this section we are going to further explain how the line fitting process works in our application under scenarios 1 and 2. In the previous section, we defined the general label expressions under scenarios 1 and 2. In order to fit a line on these cluster centres we first have to identify cluster zero, which is a pretty simple task. We then create a line passing from (0,0) and from each one of the five or seven remaining cluster centres, depending on the scenario, one at a time. More specifically, what we are trying to find is a slope for which we have a straight line crossing three or four cluster centres, including the axis start (0,0). As a counterexample to make it clear, suppose the green line crossing (0,0) and the cluster centre near $-2+0j$ of the Figure 5.4 below, then there would be no other cluster centres for which their distance from that line would be almost zero, so that cluster is definitely not of the form βh_B. On the other hand, if we create the line passing from (0,0) and the second cluster along the red line on Figure 5.4 we can see that the third cluster automatically fits on the line thus making these two clusters the products of h_B we were looking for. Using this method we can easily distinguish the clusters that are products of h_B in scenarios 1 and 2. Knowing which cluster centres are products of h_B, as well as their corresponding labels, allows us to compute an estimate of the channel coefficient h_B.

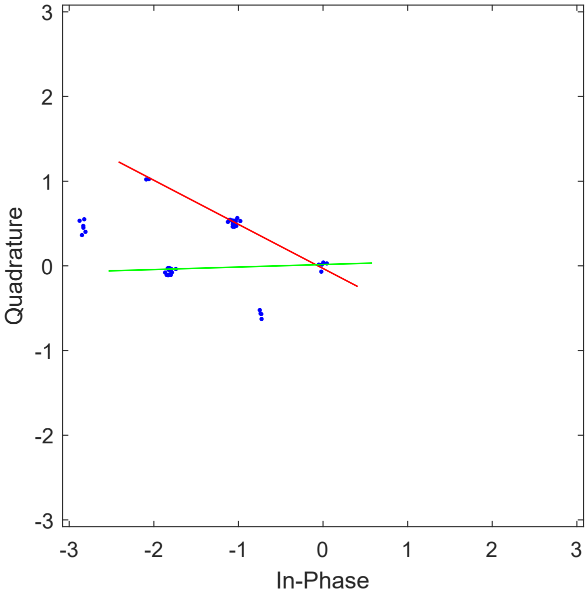


Figure 5.4: Line fitting on h_B cluster products.

In order to extract an estimate of h_A, we must remember that the channel coefficients are complex numbers and can be represented as vectors that have a certain direction. There are almost only two ways these lines can appear on the I/Q plane, since it is practically impossible to have undefined or completely equal to zero slope in our application and here is where we take advantage of the categorization we performed earlier based on $\Re\{h_B\}$. We distinguish these two ways according to the real part values of our h_B estimate, i.e. $\Re\{h_B\} \lesseqgtr 0$, which means that the position of h_B on the I/Q plane will define how the lines will be placed and how we are going to estimate the channel coefficient h_A. It has now become much simpler to obtain the h_A estimate since all we have to do now is use what the method we developed at the end of the previous section.

5.3.2 Channel Estimation - Scenario 3

Scenario 3 does not impose great difficulty, since in that case only four clusters would ideally exist. Being one of the four clusters is the all zeros case, we are left with the remaining three that obviously represent h_A, h_B and $h_A + h_B$. The triangle inequality is a tool we could use in this case, in order to identify the $h_A + h_B$ cluster. All we have to do is find the only two clusters that satisfy the triangle inequality:

$$|h_A + h_B| \leq |h_A| + |h_B|$$

Having found the $h_A + h_B$ cluster, the remaining two would be h_A and h_B respectively. Since, $\boldsymbol{x}_p = [1\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 1]$ are the preamble bits of the transmitted packet of any RFID tag, if the first samples of the received signal are not approximately equal to $h_A + h_B$, since the preamble packet starts with two 1's, and we have observed 4 clusters then the offset is definitely not equal to 0 and is equal to $\frac{L}{2}$. However, if it is not even approximately equal to $h_A + h_B$ then in order to obtain an estimate for the channel coefficients we must trace back to the MSc work of N. Kargas [12]. More specifically what Kargas does is a slightly modified version of the method proposed in [15] for channel estimation. Let $\boldsymbol{y}_{1:N}$ be the received sequence of FM0 symbols that follow the preamble. The algorithm is based on the observation that two states of the total four states are realized during the transmission of the preamble defining an one-dimensional subspace s_p (line). Projecting the received half bits onto the subspace orthogonal to s_p will give a non-zero value if the corresponding half bit is equal to h_A or h_B. The authors in [15] search for points that have the maximum signal strength in this orthogonal subspace. More specifically they process the received waveform after matched filtering y_f and set:

$$k_A = \underset{k \in \{0,1,\ldots,NL\}}{\operatorname{argmax}} \Im\Big\{y_f[k]e^{-j\angle h_A + h_B}\Big\}$$

$$k_B = \underset{k \in \{0,1,\ldots,NL\}}{\operatorname{argmin}} \Im\Big\{y_f[k]e^{-j\angle h_A + h_B}\Big\}$$

and use $y_f[k_A]$ and $y_f[k_B]$ as the channel estimates. It is explicitly underlined that the received signal cannot be jointly sampled using the nominal symbol duration L and thus process FM0 symbols.

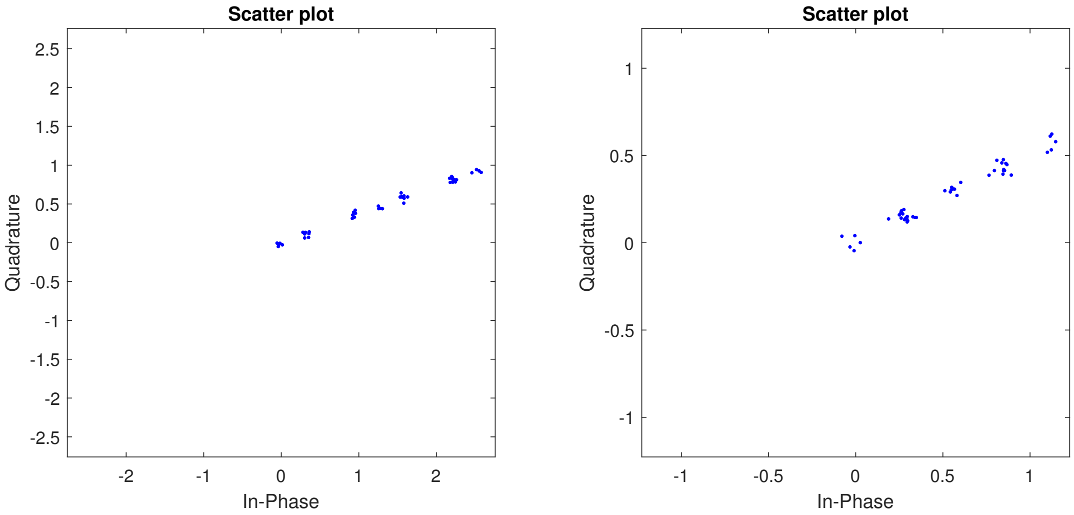
However timing errors are not critical when tags operate using the minimum backscatter link frequency (BLF). We apply the proposed method in case of synchronous tag transmission, i.e. scenario 3. In addition taking into account the FM0 encoding, we can search for a symbol $\boldsymbol{y}$ that maximizes the following metric:

$$k^* = \underset{k \in \{1,2,\ldots,16\}}{\operatorname{argmax}} \Big|\Im\Big\{\boldsymbol{y}_k[0]e^{-j\angle h_A + h_B}\Big\} - \Im\Big\{\boldsymbol{y}_k[1]e^{-j\angle h_A + h_B}\Big\}\Big|$$

where $\Im\{\boldsymbol{y}_k[0]e^{-j\angle h_A + h_B}\}, \Im\{\boldsymbol{y}_k[1]e^{-j\angle h_A + h_B}\}$ are the projections of $\boldsymbol{y}_k[0]$ and $\boldsymbol{y}_k[1]$ onto $s_{p\perp}$. We then set $\hat{h}_A = \boldsymbol{y}_{k^*}[0]$, $\hat{h}_B = \boldsymbol{y}_{k^*}[1]$. It is insignificant if we exchange the estimates of the channel coefficients.

5.3.3 Degenerate Cases

However, good the line fitting method for channel estimation may seem, there is one obstacle that is really hard to overcome and that is the case of collinear clusters. Collinear clusters exist only when $h_A \approx kh_B$, where $k \in \mathbb{R}$. Having, collinear clusters does not allow us to perform the line fitting method with high precision and that way the estimated channel coefficients are not very representative of the real ones. The figures below better illustrate the results of collinear channel coefficients.



(a) Collinear clusters of Scenario 1 when, $\tau_B = 5$ and $h_A = h_B$.

(b) Collinear clusters of Scenario 2 when, $\tau_B = 30$ and $h_A = h_B$.

Figure 5.5: Collinear clusters.

As we can see from the figures above in the first case (Figure 5.5a), we were expecting to see 8 clusters, since we are in scenario 1, but instead we see 7, since the clusters of h_A and h_B have merged into one cluster. On the second case (Figure 5.5b) we observe only 5 clusters, where we would expect to see 6 of them. This is because, the clusters of h_A and h_B are once again merged into one cluster, since they are approximately equal. This phenomenon can have many different outcomes, like when we are in scenario 2 with $\tau_B = 30$ and $h_A = -2h_B$ and we still observe 6 collinear clusters, but do not know how to detect them using line fitting. There could be a possibility that we are able to estimate the channel coefficients if we also take advantage of the cluster transitions, but we are not going to look further into it in this thesis.

5.4 Offset Estimation

In this offset estimation method, we have to find out which τ_B value holds best for our signal. The idea behind this method is to take advantage of the preamble symbols that exist in the start of the packet of the FM0 encoding scheme, and try to align with the received signal. The preamble packet is denoted as, $\boldsymbol{x}_p = [1\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 1]$. We oversample every half-bit of the preamble packet by L/2 and denote it as $\boldsymbol{x}_{p,up}$. We then create a discrete channel equivalent as shown below and pass the preamble symbols through that channel.

$$c[k] = \hat{h}_A * \boldsymbol{x}_{p,up} + [zeros(1, \tau_B)\ \hat{h}_B] * \boldsymbol{x}_{p,up}$$

Then, we have to pass $c[k]$ from the matched filter $\Pi[k]$ to obtain $c_f[k]$.

$$c_f[n] = \sum_{k=-\infty}^{\infty} \frac{c[k]}{L/2} \Pi[n-k]$$

To rule about the value of τ_B, we have to test each value and make a decision like below:

$$\tau_B^* = \underset{\tau_B \in \{0,1,\ldots,L-1\}}{\text{argmin}} \sum |c_f(\tau_B) - y_f[0 : NpL/2]|$$

where, from y_f we only need the first $N_p \frac{L}{2}$ samples, i.e. the preamble bits samples.

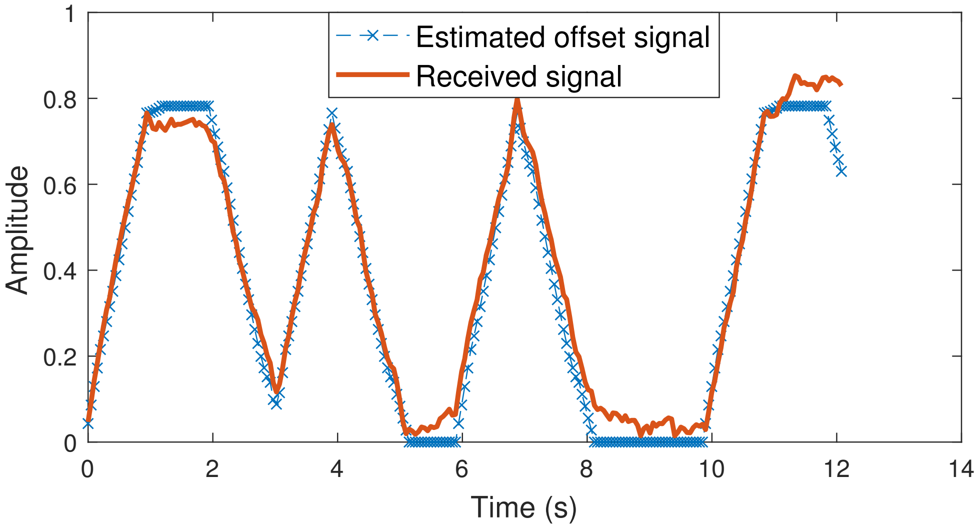


Figure 5.6: Offset estimation in scenario 1, where $\tau_B^* = 5$, in high $SNR_{A,B}$.

5.5 Discussion

To summarize, in this chapter we provided a DC offset estimation for packets received from real life simulations. We solved the cluster labelling problem by noticing the repeating patterns of the clusters and that helped us in proceeding to the channel estimation stage where line fitting was used and the channel coefficients were extracted. Higher precision of the channel estimation algorithm is feasible when we are in a high SNR environment and the clusters do not fall under the degenerate case of collinearity we mentioned earlier above. Having obtained an estimate for the channel coefficients we can easily estimate the time offset of tag B τ_B. However, if the channel estimation is wrong, then most likely the offset estimation will be wrong as well. In the last chapter we propose as future work, alternative solutions for joint channel and offset estimation.

Chapter 6

Detection & Performance Evaluation

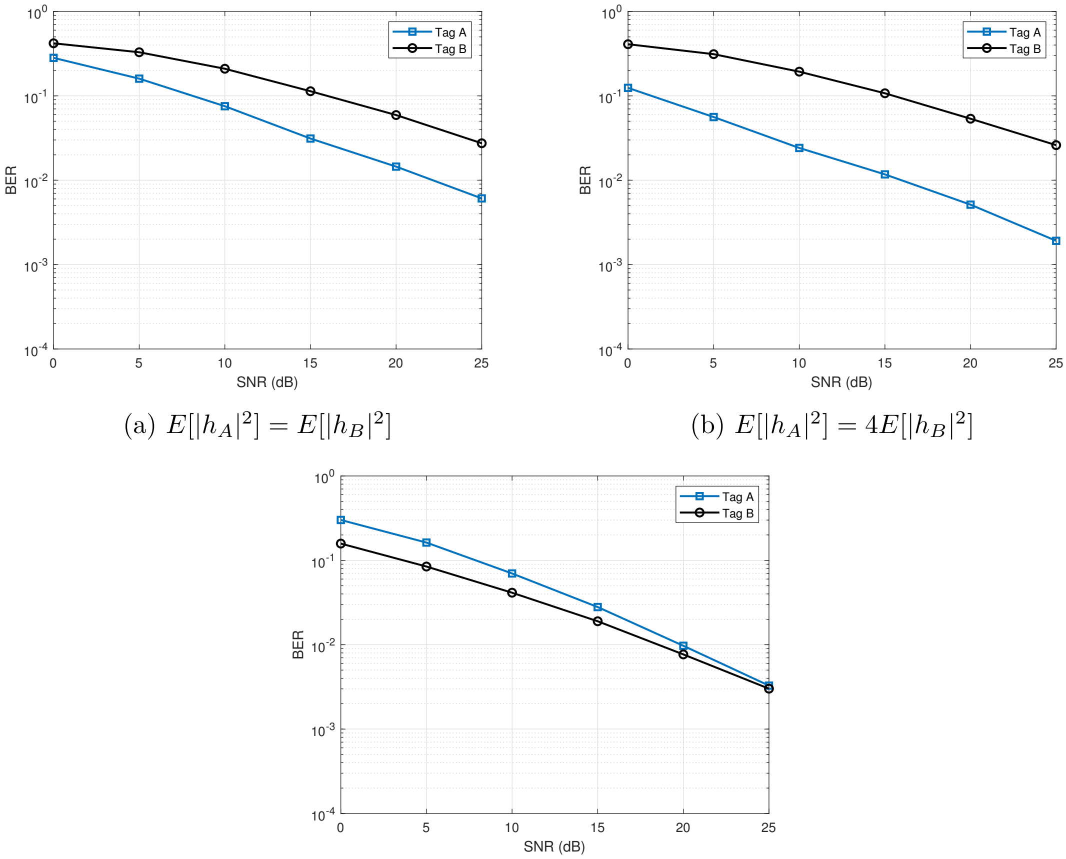
6.1 Detection with Perfectly Known Channel State Information

Assuming an ideal case, where a perfect estimate of our channel coefficients $\hat{h}_A$, $\hat{h}_B$ and the time offset of tag B $\hat{\tau}_B$ was somehow obtained, we can now attempt to detect what each tag sent and try to resolve the RN16 packets collision. However, a reader can only acknowledge a single tag even if it is able to resolve the collision and detect the transmitted sequences of both tags. Thus, even if this method is applied and eventually resolve the collision then the reader will only reply to one of the two tags.

6.1.1 Joint Detection

For the case of joint detection, i.e. when there is interest in decoding both tag A and tag B, then for the conditional pdf of $\boldsymbol{y}$ holds that: $\boldsymbol{y}|\hat{h}_A, \hat{h}_B, \hat{\tau}_B \sim \mathcal{CN}(\hat{h}_A \boldsymbol{x}_A + \hat{\boldsymbol{B}}\hat{h}_B \boldsymbol{x}_B, \sigma^2 \boldsymbol{I}_2)$.

$$\begin{aligned}
\hat{\boldsymbol{x}}_A, \hat{\boldsymbol{x}}_B &= \underset{\boldsymbol{x}_A, \boldsymbol{x}_B \in \{[0\ 1]^T, [1\ 0]^T\}}{\operatorname{argmax}} f(\boldsymbol{y}|\hat{h}_A, \hat{h}_B, \hat{\tau}_B) \\
&= \underset{\boldsymbol{x}_A, \boldsymbol{x}_B}{\operatorname{argmax}} \frac{1}{\pi^2 \det(\sigma^2 \boldsymbol{I}_2)} \exp\Big\{ -(\boldsymbol{y} - \hat{h}_A \boldsymbol{x}_A - \hat{\boldsymbol{B}}\hat{h}_B \boldsymbol{x}_B)^H (\sigma^2 \boldsymbol{I}_2)^{-1} (\boldsymbol{y} - \hat{h}_A \boldsymbol{x}_A - \hat{\boldsymbol{B}}\hat{h}_B \boldsymbol{x}_B) \Big\} \\
&= \underset{\boldsymbol{x}_A, \boldsymbol{x}_B}{\operatorname{argmin}} (\boldsymbol{y} - \hat{h}_A \boldsymbol{x}_A - \hat{\boldsymbol{B}}\hat{h}_B \boldsymbol{x}_B)^H (\frac{1}{\sigma^2} \boldsymbol{I}_2)(\boldsymbol{y} - \hat{h}_A \boldsymbol{x}_A - \hat{\boldsymbol{B}}\hat{h}_B \boldsymbol{x}_B) \\
&= \underset{\boldsymbol{x}_A, \boldsymbol{x}_B}{\operatorname{argmin}} \frac{(\boldsymbol{y} - \hat{h}_A \boldsymbol{x}_A - \hat{\boldsymbol{B}}\hat{h}_B \boldsymbol{x}_B)^H (\boldsymbol{y} - \hat{h}_A \boldsymbol{x}_A - \hat{\boldsymbol{B}}\hat{h}_B \boldsymbol{x}_B)}{\sigma^2} \\
&= \underset{\boldsymbol{x}_A, \boldsymbol{x}_B}{\operatorname{argmin}} ||\boldsymbol{y} - \hat{h}_A \boldsymbol{x}_A - \hat{\boldsymbol{B}}\hat{h}_B \boldsymbol{x}_B||_2^2 \\
&= \underset{u \in \mathcal{U}(\tau_B)}{\operatorname{argmin}} ||\boldsymbol{y} - \boldsymbol{u}||_2^2 \qquad (6.1)
\end{aligned}$$



(a) $E[|h_A|^2] = E[|h_B|^2]$

(b) $E[|h_A|^2] = 4E[|h_B|^2]$

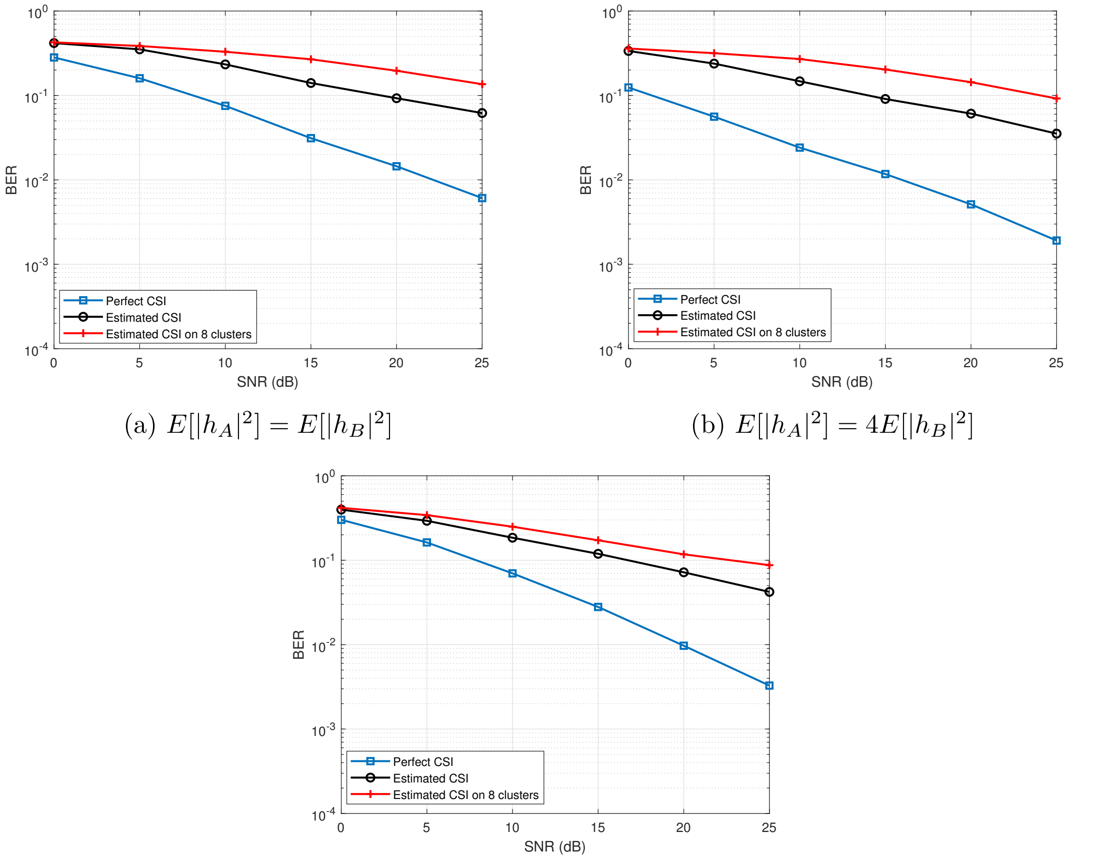
(c) $16E[|h_A|^2] = E[|h_B|^2]$

Figure 6.1: BER at tag A and tag B with perfect channel state information.

The above figures correspond to perfect channel state information, hence they represent the lowest bound that can be achieved in our application. In Figures 6.1a and 6.1b, tag B is observed having a difference of around 10 dB from tag A and that is due to the fact that we are synchronized with respect to tag A and performed our detection that way. In Figure 6.1c we observe the only case where tag B and tag A are asymptotically equal in terms of BER, but this is an extreme case and it is mostly demonstrated it for that purpose.

6.2 Detection with Partially Known Channel State Information

The above described estimation methods can be directly applied in a real-world scenario, since an estimate of the channel and the offset can be obtained from the data and then perform joint ML detection, or single tag detection. The method used to obtain the channel estimates, is the one described in Section 5.3. Another method of clustering is also evaluated, where affinity propagation was forced to find 8 clusters every time and perform the CSI estimation upon these clusters. The concept behind forcing affinity propagation to search for 8 clusters lies behind the fact that all scenarios are 'children' of scenario 1 since for the appropriate offsets the clusters of the other two scenarios can be obtained. A universal form of clusters that only depends on the offset τ_B was formed as shown in Eq. 4.4.



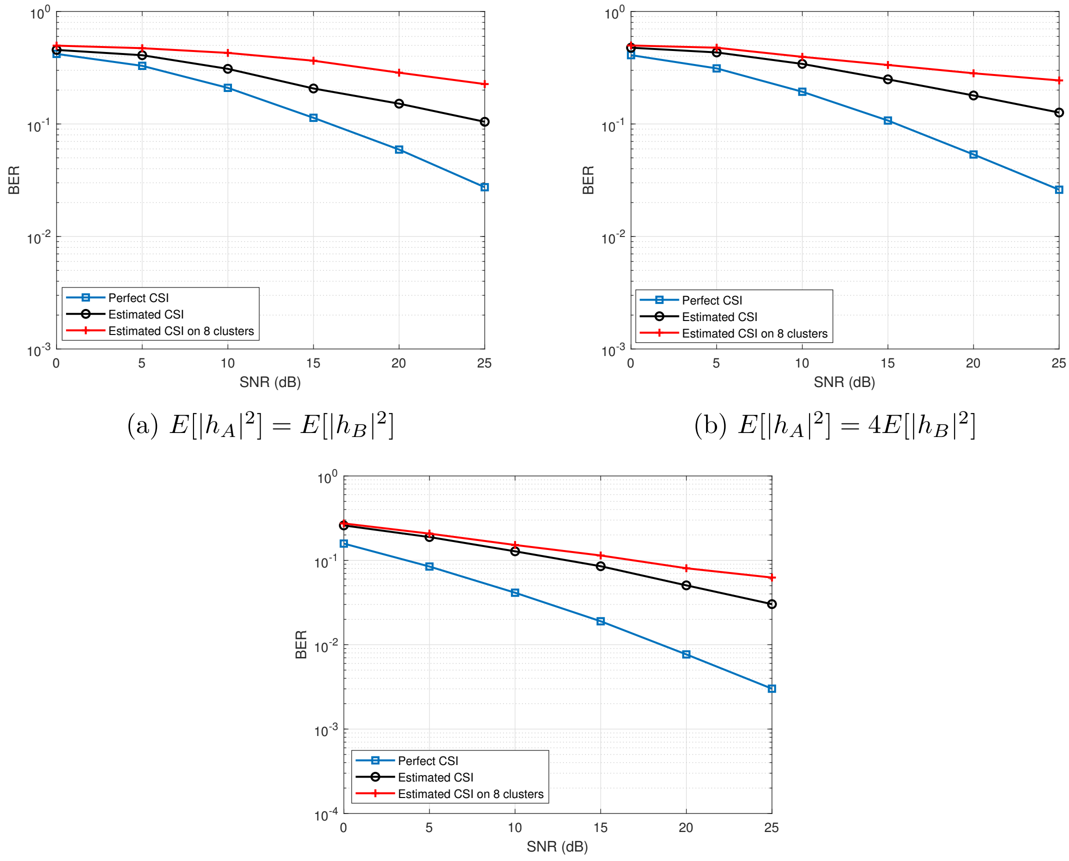
(a) $E[|h_A|^2] = E[|h_B|^2]$

(b) $E[|h_A|^2] = 4E[|h_B|^2]$

(c) $16E[|h_A|^2] = E[|h_B|^2]$

Figure 6.2: BER at tag A in different cases .

The above figures correspond to a comparison between perfect CSI, estimated CSI and estimated CSI using the 8 clusters method, when there is interest in decoding both tags. Once again it can be observed that there is a gap of around 10 dB between the estimated and the perfect CSI curve of tag A, whereas the method of forcing an 8 cluster formation does not have an effect at all in any of our tested cases.



(a) $E[|h_A|^2] = E[|h_B|^2]$

(b) $E[|h_A|^2] = 4E[|h_B|^2]$

(c) $16E[|h_A|^2] = E[|h_B|^2]$

Figure 6.3: BER at tag B in different cases.

The above figures correspond to a comparison between perfect CSI, estimated CSI and estimated CSI using the 8 clusters method, when there is interest in decoding both tags. It can once again be noticed that there is a gap of around 10 dB between the estimated and the perfect CSI curve of tag B, whereas the method of forcing an 8 cluster formation does not have an effect at all in any of our tested cases.

6.3 Discussion

All in all, what can be understood from the figures above is that, since we are synchronized with respect to tag A it is expected to have a better BER curve in tag A than in tag B at all cases and that indeed happens. It is also observed that the method of forcing affinity propagation to search for 8 clusters and use them for the estimation and detection process, did not work out very well, mostly for the reason being that the are many degenerate cases, like collinear clusters, and due to the fact that affinity propagation does not search for patterns of clusters (i.e. 4+4 parallel clusters) in the way it was thought it could. Probably, a pattern searching algorithm could perform better in that case.

Chapter 7

Conclusions & Future Work

In this thesis, a system model for the simultaneous asynchronous transmission of two RFID tags was developed, during the RN16 stage, along with a channel estimation algorithm based on clustering techniques and line fitting.

7.1 Summary of thesis contributions

Although this system model is pretty close to a real life scenario, it does not take into account the deviations that occur in the packets' transmission, like that the bit duration L can change during the transmission, or that offset can also change slightly during the transmission. All these do not necessarily mean that this model would not work, when applied in a real world scenario, but instead that it might need some small adjustments. The most important part of this thesis was the study on the collision of two RFID tags, during the transmission of their RN16 sequences. In our modelling, the tags need not be perfectly synchronized, which is something that does occur a lot in a real world scenario. The tags do not always have the same distance from the reader, thus their individual channel adds to that delay causing a slight offset to the received signal from one of the two tags. Even though the signals of the tags are collided in the air, using this algorithm the collision can be resolved and the packet will not be lost.

It was already mentioned that, the channel estimation algorithm proposed in this thesis has a small defect and that is, the collinear clusters. It can be really hard and sometimes even impossible, to be able to estimate the channel coefficients using line fitting. However, from real world simulations it was observed that this is something that does not happen very often. Thus, for now, it can be safely assumed that this can be a defect, but since it is not an often occurrence we could say, without harm, that it is negligible.

7.2 Direction for future work

As future work, we could try to implement or use an algorithm that performs joint channel and offset estimation, thus probably leading us to more robust performance and better results. Furthermore, hierarchical clustering algorithms like HDBSCAN seem to perform really good in terms of pattern matching, which could prove to be ideal for this type of work. Other clustering algorithms could also be explored that have lower time complexity than affinity propagation, or even modify affinity propagation to search for our expected cluster patterns.

References

[1] M. Simon and D. Divsalar, “Some interesting observations for certain line codes with application to RFID”, *IEEE Transactions on Communications*, vol. 54, no. 4, pp. 583–586, Apr. 2006.

[2] *EPC UHF Gen2 Air Interface Protocol*, EPC Gen2 v. 2.1, GS1, Nov. 2013.

[3] N. Kargas, F. Mavromatis, and A. Bletsas, “Fully-Coherent Reader With Commodity SDR for Gen2 FM0 and Computational RFID”, *IEEE Wireless Communications Letters*, vol. 4, no. 6, pp. 617–620, 2015.

[4] M. Ouroutzoglou, A. Bletsas, and G. N. Karystinos, “Intelligent noncoherent sequence equals coherent detection: Experimental proof in industrial RFID”, in *IEEE 7th International Conference on Modern Circuits and Systems Technologies for Electronics and Communications (MOCAST)*, May 2018.

[5] A. Bletsas, J. Kimionis, A. G. Dimitriou, and G. N. Karystinos, “Single-Antenna Coherent Detection of Collided FM0 RFID Signals”, *IEEE Transactions on Communications*, vol. 60, no. 3, pp. 756–766, 2012.

[6] P. Hu, P. Zhang, and D. Ganesan, “Laissez-Faire: Fully Asymmetric Backscatter Communication”, *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 255–267, 2015.

[7] A. Bletsas, A. G. Dimitriou, and J. N. Sahalos, “Improving Backscatter Radio Tag Efficiency”, *IEEE Transactions on Microwave Theory and Techniques*, vol. 58, no. 6, pp. 1502–1509, 2010.

[8] M. Jin, Y. He, X. Meng, Y. Zheng, D. Fang, and X. Chen, “FlipTracer: Practical Parallel Decoding for Backscatter Communication”, in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, ser. MobiCom ’17, ACM, 2017, pp. 275–287.

[9] X. Zhang and K. G. Shin, “Cooperation without Synchronization: Practical Cooperative Relaying for Wireless Networks”, *IEEE Transactions on Mobile Computing*, vol. 14, no. 5, pp. 937–950, 2015.

[10] B. J. Frey and D. Dueck, “Clustering by Passing Messages Between Data Points”, *Science (New York, N.Y.)*, vol. 315, pp. 972–6, Mar. 2007.

[11] D. Dueck, “Affinity Propagation: Clustering by passing messages”, PhD thesis, University of Toronto, 2009.

[12] N. Kargas, “SDR Readers for Gen2 RFID and Backscatter Sensor Networks”, Master’s thesis, Technical University of Crete, 2015.

[13] Y.-Y. Wang and J.-T. Chen, “A Baseband Signal Processing Scheme for Joint Data Frame Synchronization and Symbol Decoding for RFID Systems”, *EURASIP J. Adv. Sig. Proc.*, vol. 2010, 2010.

[14] P. N. Alevizos, X. Fu, N. D. Sidiropoulos, Y. Yang, and A. Bletsas, “Limited Feedback Channel Estimation in Massive MIMO With Non-Uniform Directional Dictionaries”, *IEEE Transactions on Signal Processing (TSP)*, vol. 66, no. 19, pp. 5127–5141, Oct. 2018.

[15] C. Angerer, “Design and Exploration of Radio Frequency Identification Systems by Rapid Prototyping”, PhD thesis, Vienna University of Technology, 2010.

[16] C. Angerer, B. Knerr, M. Holzer, and A. Adalan, “Flexible simulation and prototyping for RFID designs”, in *In Proceedings of the EURASIP Workshop on RFID Technology*, 2007.

[17] G. Vannucci, A. Bletsas, and D. Leigh, “Implementing Backscatter Radio for Wireless Sensor Networks”, in *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, Sep. 2007, pp. 1–5.

[18] J. Kimionis, A. Bletsas, and J. N. Sahalos, “Increased Range Bistatic Scatter Radio”, *IEEE Transactions on Communications*, vol. 62, no. 3, pp. 1091–1104, Mar. 2014.

[19] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”, in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD’96, Portland, Oregon: AAAI Press, 1996, pp. 226–231.

[20] R. J. G. B. Campello, D. Moulavi, and J. Sander, “Density-Based Clustering Based on Hierarchical Density Estimates”, in *Advances in Knowledge Discovery and Data Mining*, J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 160–172.

[21] R. J. G. B. Campello, D. Moulavi, A. Zimek, and J. Sander, “Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection”, *ACM Trans. Knowl. Discov. Data*, vol. 10, no. 1, 5:1–5:51, 2015.

[22] R. G. Gallager, *Circularly-Symmetric Gaussian random vectors*, http://www.rle.mit.edu/rgallager/documents/CircSymGauss.pdf, Online, 2008.

[23] Y. Liu, Z. Qin, M. Elkashlan, Z. Ding, A. Nallanathan, and L. Hanzo, “Nonorthogonal multiple access for 5g and beyond”, *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2347–2381, Dec. 2017.

[24] Y. Saito, A. Benjebbour, Y. Kishiyama, and T. Nakamura, “System-level performance evaluation of downlink non-orthogonal multiple access (NOMA)”, pp. 611–615, Sep. 2013.

[25] Z. Ding, Z. Yang, P. Fan, and H. V. Poor, “On the performance of non-orthogonal multiple access in 5g systems with randomly deployed users”, *IEEE Signal Processing Letters*, vol. 21, no. 12, pp. 1501–1505, Dec. 2014.

[26] M. F. Hanif, Z. Ding, T. Ratnarajah, and G. Karagiannidis, “A Minorization-Maximization Method for Optimizing Sum Rate in Non-Orthogonal Multiple Access Systems”, *IEEE Transactions on Signal Processing*, vol. 64, May 2015.

[27] P. Xu, Z. Ding, X. Dai, and H. V. Poor, “A new evaluation criterion for non-orthogonal multiple access in 5g software defined networks”, *IEEE Access*, vol. 3, pp. 1633–1639, 2015.

[28] Z. Ding, F. Adachi, and H. V. Poor, “The Application of MIMO to Non-Orthogonal Multiple Access”, *IEEE Transactions on Wireless Communications*, vol. 15, pp. 537–552, 2016.

[29] Z. Ding, R. Schober, and H. V. Poor, “A General MIMO Framework for NOMA Downlink and Uplink Transmission Based on Signal Alignment”, *IEEE Transactions on Wireless Communications*, vol. 15, pp. 4438–4454, 2016.

[30] Z. Ding, M. Peng, and H. V. Poor, “Cooperative Non-Orthogonal Multiple Access in 5G Systems”, *IEEE Communications Letters*, vol. 19, no. 8, pp. 1462–1465, Aug. 2015.

[31] Z. Ding and H. V. Poor, “Design of massive-mimo-noma with limited feedback”, *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 629–633, May 2016.

[32] Y. Liu, Z. Ding, M. Elkashlan, and H. V. Poor, “Cooperative Non-orthogonal Multiple Access With Simultaneous Wireless Information and Power Transfer”, *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 4, pp. 938–953, Apr. 2016.

[33] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.