Multiple Radar Environment Emission Deinterleaving and PRI Prediction

Prepared by:

Kaveer Manickchand

MNCKAV002

Prepared for:

Dr. A. K. Mishra

Radar Remote Sensing Group

Department of Electrical Engineering

Engineering & Built Environment

University of Cape Town

External study leader:

J. J. Strydom

Council for Scientific and Industrial Research

March 2017

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfillment of the academic requirements for a

Master of Science in Engineering in Radar and Electronic Defence

Declaration

I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own. This thesis/dissertation has been submitted to the Turnitin module (or equivalent similarity and originality checking software) and I confirm that my supervisor has seen my report and any concerns revealed by such have been resolved with my supervisor.

Signed by candidate

13/03/2017

Kaveer Manickchand

Date

EBE Faculty: Assessment of Ethics in Research Projects (Rev2)

Any person planning to undertake research in the Faculty of Engineering and the Built Environment at the University of Cape Town is required to complete this form before collecting or analysing data. When completed it should be submitted to the supervisor (where applicable) and from there to the Head of Department. If any of the questions below have been answered YES, and the applicant is NOT a fourth year student, the Head should forward this form for approval by the Faculty EIR committee: submit to Ms Zulpha Geyer (Zulpha.Geyer@uct.ac.za; Chem Eng Building, Ph 021 650 4791).
NB: A copy of this signed form must be included with the thesis/dissertation/report when it is submitted for examination

This form must only be completed once the most recent revision EBE EIR Handbook has been read.

Name of Principal Researcher/Student: Kaveer Manickchand Department: Faculty of Engineering & Built Environment

Preferred email address of the applicant: kmanickchand@gmail.com

If a Student: Degree: MSc (Eng) Electrical Specialising in Radar Supervisor: Dr. Amit Mishra

If a Research Contract indicate source of funding/sponsorship:

Research Project Title: Multiple Radar Environment Emission Deinterleaving and PRI Prediction

Overview of ethics issues in your research project:

Question 1: Is there a possibility that your research could cause harm to a third party (i.e. a person not involved in your project)?	YES	NO ✓
Question 2: Is your research making use of human subjects as sources of data? If your answer is YES, please complete Addendum 2.	YES	NO ✓
Question 3: Does your research involve the participation of or provision of services to communities? If your answer is YES please complete Addendum 3.	YES	NO ✓
Question 4: If your research is sponsored, is there any potential for conflicts of interest? If your answer is YES, please complete Addendum 4.	YES	NO ✓

If you have answered YES to any of the above questions, please append a copy of your research proposal, as well as any interview schedules or questionnaires (Addendum 1) and please complete further addenda as appropriate. Ensure that you refer to the EIR Handbook to assist you in completing the documentation requirements for this form.

I hereby undertake to carry out my research in such a way that

- there is no apparent legal objection to the nature or the method of research; and
- the research will not compromise staff or students or the other responsibilities of the University;
- the stated objective will be achieved, and the findings will have a high degree of validity;
- limitations and alternative interpretations will be considered;
- the findings could be subject to peer review and publicly available; and
- I will comply with the conventions of copyright and avoid any practice that would constitute plagiarism

Signed by:

	Full name and signature	Date
Principal Researcher/Student:	Kaveer Manickchand	01/03/2017
This application is approved by: Supervisor (if applicable):		01/3/17
HOD (or delegated nominee): *Final authority for all assessments with NO to all questions and for all undergraduate research.*		1/3/17
Chair : Faculty EIR Committee For applicants other than undergraduate students who have answered YES to any of the above questions.		

Abstract

The aim of this study was to research TOA based tracking and deinterleaving algorithms suited to radar emitters in an EW environment for application on the CSIR 5^{th} generation DRFM platform. The research problem statement stipulated that the only defining characteristic of the different emitters be the time of arrival (TOA) of their pulses. The pulse repetition interval (PRI) schemes considered in the study was constant, jittered, staggered and dwell and switch.

The different TOA based deinterleaving algorithms investigated were sequence search (SS), TOA difference histogram, CDIF, SDIF, CDIF with SS (CDIF SS), SDIF with SS (SDIF SS) and interleaved pulse train spectrum estimation. The interleaved pulse train spectrum estimation algorithm results could not be replicated and were not included in simulations. The TOA based tracking algorithms that were also investigated were Delta-τ histogram, Kalman filter, alpha-beta filter and alpha-beta-gamma filter. The alpha-beta-gamma filter became unstable during simulations and hence their results have also been excluded.

The algorithms were simulated in MATLAB against EW environments with varied TOA measurement noise, number of emitters, PRI schemes and interference pulses (missing and spurious). General conclusions drawn from the deinterleaving simulations were the success of the algorithms decrease with the increase of emitters in the EW environment, interference pulses increased the success of some algorithms and the success of algorithms increased with TMNR (time measurement to noise ratio). General conclusions drawn from the tracking simulations were track loss of the algorithms decrease with increase in TMNR, tracking error decreases with increase in TMNR and interference pulses affected the initial estimates used to initialise the filters.

The performance of the deinterleaving (CDIF & CDIF SS) and tracking (Delta-τ histogram & alpha-beta filter) algorithms were compared on the DRFM platform. On the DRFM platform, the CDIF algorithm deinterleaved in fewer pulses but had more false detections as compared to the CDIF SS algorithm. The alpha-beta filter performed better with lower TMNR than the Delta-τ histogram, on the DRFM platform.

The CDIF SS algorithm and alpha-beta filter were chosen, based on their performance on the DRFM, to be implemented on a DRFM based system that would deinterleave and then track emitters in an EW environment. The system was successfully implemented and met all requirements that were placed on it. Possible improvements to the system and the future improvements to the research are also discussed.

Acknowledgments

I would like to thank my supervisor, Dr Amit K. Mishra for his support and guidance throughout my master's degree. His advice on even the little things such as which word processing tool to use was extremely helpful and time-saving.

I would like to thank my mentor Jurgen J. Strydom, who helped me formulate the focus of my research. His "door was always open" for discussions, and I have learned much from him. I would also like to thank him for his inputs with this document.

I would like to thank Rene Broich for passing on his knowledge on the CSIR 5^{th}generation DRFM.

I would like to thank my family and friends for their understanding and support.

Finally, I would like to thank the CSIR for affording me this opportunity and allowing me to use CSIR resources.

Table of Contents

List of Figures

List of Tables

Nomenclature

µs	Microseconds.
A-DIF	All Difference Histogram.
ADC	Analogue to Digital Converter
AOA	Angle of Arrival.
CDIF	Cumulative Difference Histogram.
CDIF SS	Cumulative Difference Histogram with Sequence Search.
CSIR	Council of Scientific and Industrial Research.
CW	Continuous Wave.
DAC	Digita to Analogue Converter
DOA	Direction of Arrival.
DRFM	Digital Radio Frequency Memory.
EA	Electronic Attack.
ECCM	Electronic Counter Counter Measures.
ECM	Electronic Counter Measures.
ELINT	Electronic Intelligence.
EM	Electromagnetic.
EP	Electronic Protection.
ES	Electronic Support.
ESM	Electronic Support Measures.
EW	Electronic Warfare.
FMCW	Frequency Modulated Continuous Wave.

FPGA	Field Programmable Gate Array
LFM	Linear Frequency Modulation.
MTI	Moving Target Indicator.
NLFM	Non-Linear Frequency Modulation.
ns	Nanoseconds.
PA	Pulse Amplitude.
PD	Pulse Duration.
PDW	Pulse Descriptor Word.
PRF	Pulse Repetition Frequency.
PRI	Pulse Repetition Interval.
PW	Pulse Width.
RCS	Radar Cross Section.
RF	Radio Frequency.
SDIF	Sequential Difference Histogram.
SDIF SS	Sequential Difference Histogram with Sequence Search.
SS	Sequence Search.
TMNR	Time Measurement to Noise Ratio.
TOA	Time of Arrival.
TWS	Track-while-scan.
uint32	Unsigned 32-bit Integers.

Chapter 1

Introduction

Since the introduction of radar systems in World War II, the technology has advanced to the point that modern military forces depend heavily on them [1]. Modern military forces use these electromagnetic (EM) systems for a broad range of applications [1]. Electronic warfare (EW) is the act of preserving the utilisation of the electromagnetic spectrum for friendly EM systems while impairing or denying its usage to enemy EM systems [57]. EW can be broken down into three subdivisions, namely electronic attack (EA), electronic protection (EP) and electronic support (ES) [56].

Modern EW environments, due to a large number of emissions from different EM systems, are congested. The pulses from various emitters are mixed or "interleaved" into one pulse train observed at an EW receiver [2]. The time at which an EW receiver observes a pulse is called the time of arrival (TOA). The process of analysing and separating the interleaved pulse train into pulses belonging to their emitters is called "deinterleaving" [28]. These emitters can then be monitored or "tracked" for an assortment of applications.

Pulsed radars transmit pulses of radar waves after a predefined interval of time. This interval of time, from the time a pulse is transmitted to the time the next pulse is transmitted, is referred to as the pulse repetition interval (PRI). Modern radars use different PRI schemes based on its specific function [1]. For example, a dwell and switch PRI sequence, where the PRI of the radar is held constant for a certain amount of time before switching to another PRI, is used to resolve velocity or range ambiguities[1] in a pulsed Doppler radar [1]. The PRI or PRI sequence (in schemes where PRI is varied) characteristic of a radar plays a very significant role in deinterleaving and tracking algorithms [25].

1.1 Problem Statement

The research was conducted at the behest of the Council of Scientific and Industrial Research (CSIR) and shall subsequently be referred to as the user or client. The problem statement below was proposed by the CSIR as the framework in which the research will be constrained.

[1] Ambiguous velocity and range are the velocity and range at which a radar cannot correctly observe targets. These concepts will be fully explained in section 2.2.2.

"Study TOA based tracking and deinterleaving algorithms suited to radar emitters in the EW environment for application on the CSIR 5^{th} generation DRFM platform."

The focus of this research is to identify and compare TOA based algorithms for the purpose of deinterleaving multiple radar emitters and tracking their respective PRI schemes. The user requires the best-suited algorithms be implemented on a system that will deinterleave multiple radar emitters and track their pulse time of arrival. This system is a subsystem of a larger electronic support (ES) system. The subsystem being implemented will subsequently be referred to as the system or solution system unless otherwise stated. PRI schemes considered for this research shall be restricted to constant[2], staggered[3], jittered[4] and dwell and switch[5] PRIs. Algorithms will utilise extracted pulse descriptor words[6] (PDWs), as the generation of these PDWs are not in the scope of this research. The performance of the identified algorithms should also be evaluated against the noise in TOA measurement and the occurrence of missing and spurious pulses. Other evaluation criteria are derived from the requirements placed on the system by the end user.

1.2 Objectives and Analysis

Using the problem statement and its associated description the major objectives required for the research can be concisely outlined.

1. Research algorithms to deinterleave radar emitters using only time of arrival information.
2. Research algorithms to track and predict only pulse time of arrival.
3. Algorithms shall operate against emitters having PRI scheme types: constant, jittered, staggered or dwell and switch.
4. Simulate the algorithms and evaluate their suitability to be implemented on the CSIR 5^{th} generation digital radio frequency memory (DRFM) platform [83, 84].
5. The two most suitable deinterleaving algorithms will be implemented on the CSIR 5^{th} generation DRFM platform to observe any change in performance between simulations and hardware implementations.
6. The two most suitable tracking algorithms will be implemented on the CSIR 5^{th} generation DRFM platform to observe any change in performance between simulations and hardware implementations.
7. A system will be implemented on the CSIR 5th generation DRFM platform that can deinterleave and then track radar emitters in an EW environment.

[2] A constant PRI sequence is a PRI sequence where the PRI has no variations. If variations do exist, they are purely accidental.

[3] A staggered PRI sequence is a sequence of several different PRIs in a repeating pattern.

[4] A PRI sequence where random variations are purposely added to reduce susceptibility to jamming and tracking.

[5] Different PRIs are automatically selected from a predetermined set. The PRI stays constant for a fixed number of pulses before it switches to another PRI from the set.

[6] Output from a radar pulse feature extraction algorithm, that contains parameters describing the descriptive properties of a radar pulse.

Part of performing engineering is dealing with complexity, while ensuring a project is on time, on budget and on brief [21, 22]. Systems Engineering is the industry standard for managing complexity while adhering to all the project constraints [21, 22]. The process of Systems Engineering can be broken down into three broad conceptual steps namely, in sequential order, requirements analysis, functional analysis and synthesis [60]. Synthesis includes the design and implementation of a system. The Systems Engineering process is a continual process and repeatedly iterates to ensure a sufficiently optimal design.

This dissertation follows an aggressively tailored version of the Systems Engineering approach outlined above. Tailoring is important as to not overload a simple research project with things that adds little value. Put differently, tailoring ensures that more time is spent on the things that add the most value while time is not wasted on things that no longer adds significant value. Firstly an industry comparison study is completed to determine requirements.

1.2.1 Industry Comparison Study

The deinterleaving and tracking functionality is found within an electronic support (ES) system. Therefore, the solution system will fall under the ES system category. Current ES systems were compared using brochures to find general characteristics that they possess and the type of EW (electronic warfare) environments in which they operate. Completing this survey assisted in understanding and guiding the system boundaries and performance parameters the system should at least operate within. It also assisted in making system design choices. This knowledge will also help in creating practical requirements (in section 1.2.2) against which the subsystem shall be benchmarked.

1.2.1.1 General ES System Characteristics

The brochures surveyed are extremely vague, for example several brochures state that “volume and weight are minimum” without any dimensions or weight values given. Other values are also omitted such as how many emitters constitute a “very dense environment” or how fast is “fast detection and identification”. Claims such as “fully autonomous” are quite bold, as they are most likely fully autonomous only in certain situations.

Some of the commonly mentioned properties of the commercially available ES systems seen in the brochures are listed below:

- Detect, deinterleave and track radar emitter pulse trains present in the EW environment [15].
- Intra-pulse and inter-pulse measurements [11].
- Fast detection and identification [11, 14, 16, 18].
- High probability of intercept [11, 12, 19].
- Provide data on radars present in environment [12].
- Fully autonomous [12, 13, 14].

- System induces low RCS on platform [11, 14, 16].
- Modular [11, 12, 13].
- Low maintenance [11, 17].
- High reliability [11, 17].
- Low integration cost [13, 14, 18].
- Compact pod solution to fit many platforms [12].
- Volume and weight are minimum [13, 14, 17, 18].
- Analyse and predict on pulse to pulse basis [15].
- All events are recorded [17, 18, 19].
- Provides situation awareness [17].
- Immune operation in very dense environment [11, 16, 18, 20].

1.2.1.2 EW Environment Characteristics

During test and evaluation of the system, algorithms are tested against simulated EW environments (see section 3.1). These EW environments were generated incorporating knowledge extracted from the types of EW environments the commercially available ES systems are designed for.

Here are some commonly mentioned properties of an EW environment as seen in the brochures:

- Spectrum 2-18GHz [11, 13, 16, 18]
- Bandwidths: 25 MHz to 1 GHZ [18, 19]
- Pulse widths: 50nsec – CW [11, 20]
- Frequency Measurement Resolution: 1 MHz [19]
- PRI range: 2 μs to 15 ms [15]
- Sensitivity: greater than -65 dBm [11, 17, 19, 20]
- Direction Accuracy: 2 degrees [11, 19, 20]
- Azimuth coverage: 360 degrees [13, 14, 16, 17], 210 degrees [19]
- Elevation coverage: +90 degrees to -40 degrees [14], 70 degrees [19]
- Minimum sensitivity to noise and spurious pulses [15]
- Keep track through a series of missing pulses [15]

- Able to detect changes in the EW environment and respond [13, 14]
- Very dense environment, meaning many emitters are present [11, 16, 18, 20]

The characteristics of ES systems and the EW environment found in these brochures cannot be conclusive because brochures contain many ambiguities. Brochures are written this way on purpose due to the marketing nature of the information, or the fact that systems perform differently in different scenarios and only list the best possible performance of a certain capability. However, some valuable information can still apply to this research, such as the ranges of PRIs and pulse widths.

1.2.2 Requirements Analysis

Requirements analysis is the process of taking the client's stated problem and converting it into measurable and understandable statements [22]. This process includes direct consultation with the client to ensure the requirements capture the client's needs. Requirements are first converted to system specifications, against which the system is verified. Validation is performed against the original problem statement, in other words, does the solution meet the client's needs. Verification is to make sure the system is performing the correct task. Validation is to ensure the system designers, are correctly performing the required task. Verification and validation is an ongoing process and is done throughout the design process [22].

A system is designed, implemented and integrated against its system specifications. The system specifications for this system were deduced from the problem statement and are as follows:

1. The system shall utilize PDWs of pulses in the EW environment as inputs.
2. The system shall process pulses that are indistinguishable from one another except in time of arrival.
3. The system shall utilise PDWs consisting of only time of arrival information.
4. The system shall be able to deinterleave pulsed radar emitters by using their associated PDWs as they are input into the system.
5. The system shall deinterleave a minimum of four pulsed radar emitters.
6. The system shall be able to track deinterleaved pulsed radar emitters by using their associated PDWs as they are input into the system.
7. The system shall predict the TOA of the next radar pulse for each emitter.
8. The system shall track a minimum of four pulsed radar emitters.
9. The system shall operate in an EM environment where a maximum of 10% of transmitted radar pulses are not sensed at the receiver of the system.
10. The system shall operate in an EM environment where no more than 10% of radar pulses are spurious[7] .

[7]Pulses that are sensed by the receiver of the system but do not belong to any emitters being tracked. These could originate from out of band interference, intentional deception attempts, or other systems operating in the same band.

11. The system shall operate in an EM environment where the minimum time measurement to noise ratio is 8 dB.
12. The system shall operate in an EM environment where the maximum time measurement to noise ratio is 26 dB.
13. The system shall deinterleave the PRI schemes types of constant, staggered, jittered and dwell and switch.
14. The system shall track the PRI scheme types of constant, staggered, jittered and dwell and switch.
15. The system shall begin tracking pulsed radar emitters by 4 received pulses if they utilise a constant PRI scheme.
16. The system shall begin tracking pulsed radar emitters by 4 received stagger PRI sequence repetitions if they utilise a staggered PRI scheme.
17. The system shall track jittered pulsed radars with 10% jitter relative to their PRI.
18. The system shall begin tracking jittered pulsed radars by 4 received pulses if they utilise a jittered PRI scheme.
19. The system shall begin tracking dwell and switch PRI sequences by 2 received dwell periods if they utilise a dwell and switch PRI scheme.
20. The system shall output the predicted time of arrival of the next pulse for each radar that is being tracked.
21. The system shall store the radars it is currently tracking in the EM environment.
22. The system shall store the PRI sequence, time of last received pulse and predicted time of next pulse of each radar it is currently tracking in the EM environment..
23. The system shall operate on a CSIR 5^{th} generation DRFM platform.

1.2.3 System Context Diagram

Figure 1.1 shows the solution system within the context of the systems that it interfaces with. Here the solution system boundary[8] is defined. It illustrates the inputs of the solution system and the systems from which it obtains them. The outputs, as well as to which systems these outputs will be sent to are also shown. The input of the proposed solution system will be the PDWs created by the parameter extractor. The system shall output the TOA of the next predicted pulse to the jammer[9] and output the deinterleaved pulse trains to a classifier. The classifier and jammer, which are not within the scope of this dissertation will then try to identify the type of radar found in the EW environment and use the predicted TOA information to jam enemy radar systems, respectively.

[8] System Boundary is the interface between the system and others systems or the environment.

[9] A jammer is an EA device that deceives an enemy EM system.

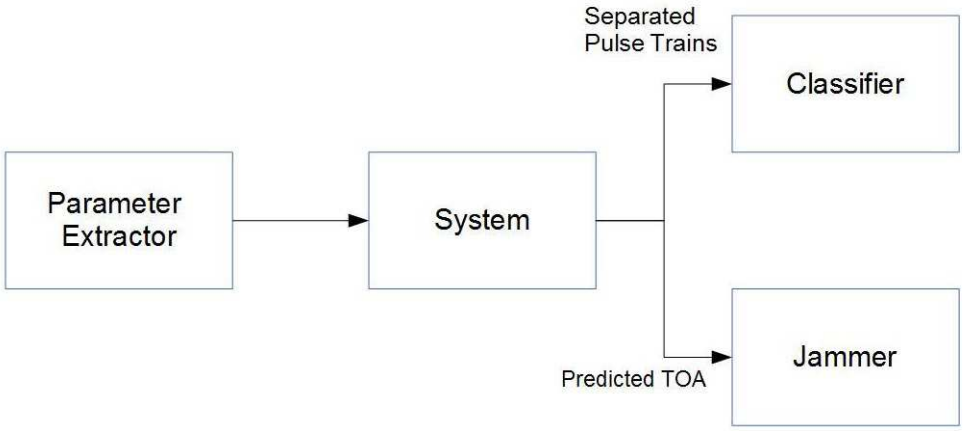


Figure 1.1: System Contextual Diagram

1.2.4 Functional Analysis

Functional Analysis is the next conceptual step in the Systems Engineering process after the requirements analysis. In this step, the system is divided into smaller components called functional elements [22]. Each functional element is defined based on the function to be performed. This definition does not specify how the function is performed, merely that it is performed. The logical flow of the design is also defined in this step.

Figure 1.2 shows the proposed solution system boundary as well as the inner functional elements of the system. Black arrows on the sides of functional elements illustrate the logical flow of the system, while blue arrows depict the item flow between the functional elements. Arrows flowing into the top and out the bottom of the functional element describes the item flow inputs and outputs, respectively.

There are five functional elements into which the system can be divided. When a new PDW is received by the system, the “match PDW with a tracker” functional element will determine if the PDW belongs to a tracked emitter or not. If it belongs to an emitter, the “track emitter” functional element will predict the TOA of the next pulse from that emitter. If the PDW did not belong to a tracker, the PDW will be stored in a buffer with the “store PDW” functional element. The “Deinterleave” functional element will then try to deinterleave any emitter pulse trains that are stored in the buffer. If a new emitter is found, the “start a new tracker” functional element will start a new tracker for that emitter.

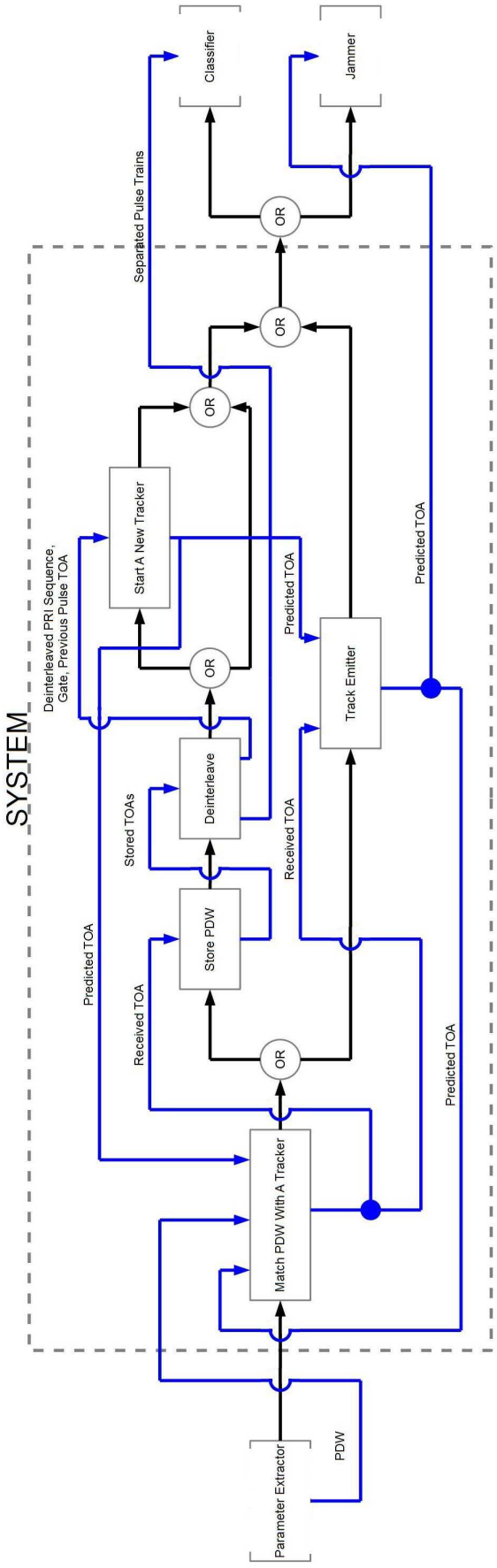


Figure 1.2: System Functional Design

1.3 Conclusion

This section has introduced the reader to the problem statement considered in this study. The problem statement was analysed to establish the objectives of the research. An industrial comparison survey was conducted to assess reasonable requirements to be placed on the system designed during this research. Using the Systems Engineering approach the requirements for the designed system were concisely outlined as per the problem statement and a survey of commercially available ES systems. The system boundary and the context in which the solution system interacts with other subsystems were defined. Finally, the functional analysis of the system was done.

In keeping with the Systems Engineering approach, the next steps of design and implementation are followed in Chapters 2 to 5. The final steps of verification and validation are performed in Chapters 5 and 6.

Chapter 2 provides a literature review that summarizes radars and its various types. It goes on to explain that the system is interested in pulsed radars and the importance of PRI in pulsed radars. A brief introduction to electronic warfare is also provided. The concepts of tracking and deinterleaving are explained in detail before TOA based deinterleaving and tracking algorithms are surveyed.

The surveyed deinterleaving algorithms are simulated in MATLAB in Chapter 3. The results of the simulations are evaluated against the requirements of the system discussed in section 1.2.2. The two algorithms that performed the best against the requirements are then implemented on the DRFM platform. The difference in results between deinterleaving simulations and the implementation on the DRFM are analysed. The best performing deinterleaving algorithm implemented on the DRFM is then selected to be used in the system that will deinterleave and track radar emitters in an EW environment. Chapter 3 also covers the simulated EW environment, in which the algorithms are evaluated.

In Chapter 4, the surveyed tracking algorithms are simulated in MATLAB. As in Chapter 3, the simulation results are evaluated against the requirements to choose the two best-performing algorithms to be implemented on the DRFM platform. The difference in results between the tracking simulations and the implementation on the DRFM are analysed. The best performing tracking algorithm implemented on the DRFM is then selected to be used in the system that will deinterleave and track radar emitters in an EW environment.

Chapter 5 covers the integration and implementation of the deinterleaving and tracking algorithms into one system on the CSIR 5^{th} generation DRFM platform. The performance of the system is evaluated against the requirements. This Chapter is where the verification and validation of the system begins.

Final conclusions are drawn, possible improvements to the system and future improvements to the research are discussed in Chapter 6.

Chapter 2

Literature Review

Radar is an acronym for Radio Detection and Ranging [23, 37, 38, 39, 41, 42]. It is an electrical system that transmits radio frequency (RF) electromagnetic (EM) waves toward a region of interest in such a manner that measurable amounts of these EM waves are reflected back to the radar from objects in that region. The EM wave that was reflected from an object and received by the radar receiver is referred to as a "radar echo" [23, 34, 37].

The first radar can be traced back to 1901, when German engineer, Christian Huelsmeyer built a simple ship detection device. Its purpose was to help avoid collisions during heavy fog [1, 24, 34, 37, 44]. Huelsmeyer built upon theories and findings of Michael Faraday[1], James Maxwell[2] and Heinrich Hertz[3]. However, the first widely used radar technology was developed for military purposes during the Second World War [1]. Radar technology, in the modern era, has grown considerably. It has found its way into many applications and contributes greatly in the fields of navigation, weather forecasts, airport traffic control, level measurements, mining, altimeters and automotive distance control [1, 38, 40, 43, 44].

The EM spectrum is the term used to describe the continuous range of frequencies or wavelengths of EM radiation [71]. A wide range of devices make use of the EM spectrum. The devices of most interest in an electronic warfare (EW) environment are either communication devices or radars. Radars in these environments can either be continuous wave or pulsed radars. The pulsed radars, which are of interest to this study, periodically emit pulses of an EM wave in what is known as a pulse train. The time interval between one pulse is emitted to the next is called the pulse repetition interval (PRI) [2]. Due to a large number of these EM systems operating simultaneously in the EW environment, an EW receiver will receive the different pulse trains associated with each emitter at the same time [33]. The resultant superimposed pulse train is referred to as an interleaved pulse train [33]. The process of determining the number of pulse trains present and separating the pulses according to their source is called deinterleaving [33]. After the deinterleaving process, each source can then be tracked. Tracking involves predicting the next pulse time of arrival (TOA) pulse from the emitter being tracked and

[1]Faraday proved that an electric current produces a magnetic field and when the current is stopped, the energy in this field returns to the circuit.

[2]Maxwell postulated that both radio and light waves are really electromagnetic waves governed by the same fundamental laws but having different frequencies. This means that radio waves will travel at the same speed of light.

[3]Hertz confirmed Maxwell's theory and also proved that radio waves can be reflected by metallic and dielectric bodies.

associating the prediction with the next measured pulse TOA from that emitter [54]. It will also try to make the predictions more accurate as time progresses [54]. It should be noted that by predicting the TOA of the next pulse, the PRI is also essentially being predicted.

2.1 Electromagnetic Spectrum

All radars and consequently EW systems radiate electromagnetic energy in the radio frequency (RF) band [23, 37, 42]. Their frequency can range from as little as 3 MHz to 300 GHz [45]. However, more commonly used frequencies range from 3 MHz to 110 GHz depending on its application [1, 23, 36] (Appendix A shows radar frequencies and their typical uses.). Due to the fact that the propagation coefficient of the atmosphere is close to that of a vacuum, EM waves travel approximately at the speed of light[4] (c_0), which is $2.9979 \times 10^8 m/s$ [2].

Since the speed at which the EM waves travel is known, the time taken, $\triangle T$, for the EM wave to return to the radar after it was transmitted can be used to determine the distance to the target from the radar, R. Provided the assumption is made that it did not bounce off of anything else (such as the ground in the case of multipath). The wave makes a round trip in this time. Therefore, only half of the travel time determines the distance to the target. Equation (2.1) shows the relationship between the range of the target and time taken to receive the echo after it was transmitted [37].

$$R = \frac{c_o \triangle T}{2} \tag{2.1}$$

Microseconds (μs) and in some cases nanoseconds (ns) are used to measure time in radar applications because of the high speed of the pulses [23]. Putting this into perspective, if a radar echo returns to receiver 1μs after it was transmitted, the object it reflected off is located approximately 150m away from the receiver.

2.2 Different Radar System Types

Radar systems can be classified into different types according to the waveform that it emits [36, 64]. The diagram in the figure below shows one method of classifying radars.

[4] Proven by Maxwell.

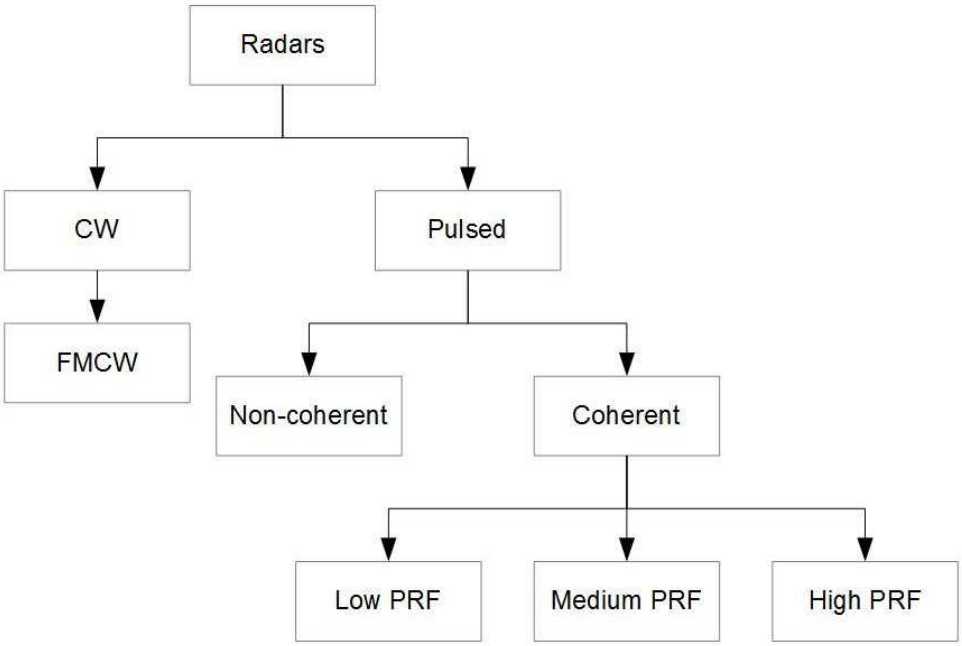


Figure 2.1: Radar Classification by Waveform, adapted from [36, 64]

The two most broad categories any radar can fall under are continuous wave and pulsed.

2.2.1 Continuous Wave (CW) Radars

A continuous wave radar system continually transmits an EM wave through its antenna with a fixed frequency. There is no frequency modulation in this type of radar. The receiver of the radar also continuously receives the reflected signal [37]. Meaning this category of radar system needs multiple antennas, at least one for transmission and one for receiving [57]. The antennas need to be properly isolated to prevent energy from the transmitting antenna leaking into the receiving antenna, especially when they are located in close proximity [57]. Since this isolation can never be perfect, the transmitted signal competes with the received signal [37]. One of the reasons the transmit power of CW radars are relatively lower compared to pulsed radars is to overcome this [37]. Due to their lower transmitting power, CW radars are mainly used for short-range applications [37].

CW radars can measure the radial velocity of a target with high resolution and accuracy as the target is continually illuminated with a continuous radar signal [55]. The radial velocity of a target is measured using the Doppler shift[5] present in the received signal [1]. When the radar signal is reflected off a stationary target, there is no Doppler frequency shift.

Unfortunately, CW radars do not provide any range information of the target. It is also blind to slow moving ground clutter, making it well suited to detect low-flying aircraft that attempts to follow a flight profile close to

[5] Apparent change in frequency of a wave by an observer moving relative to its source.

the ground (also known as "hugging the ground") to avoid detection by air defence systems [1]. CW radars are typically used as target illuminators in fire control systems, monitoring traffic and as motion sensors [1, 37, 55].

2.2.1.1 Frequency Modulated Continuous Wave Radar

The main disadvantage associated with CW radar is that it cannot measure the radial range of a target as there is no time referencing in the signal [1]. It is, however, possible to measure the radial range of stationary objects by using a frequency modulated continuous wave (FMCW) radar [1]. This type of radar adjusts the frequency of the transmitted signal periodically over time, to produce a time reference in the signal. Similar to a pulsed radar, where the echo will be received with a delay offset, the FMCW radar uses the offset in frequency of the received signal to determine the delay offset [24].

It is possible to transmit signals with complicated frequency patterns. However, basic ramp and triangular modulation are most commonly used [55].

2.2.2 Pulsed Radar

A pulsed radar transmits EM waves in pulses, separated by periods of time, in which it listens for echoes of the transmitted pulses [57]. Simple pulsed radars provide range (and directional) information of a detected target. The directional information of the target is determined using the radiation direction of the rotating antenna when the echo is received [1]. The transmit time of the pulse can be referred to as the pulse width (τ) or pulse duration (PD) [57]. The time between the leading edge of one pulse to the leading edge of the next is the pulse repetition interval (PRI). The pulse repetition frequency (PRF) is the inverse of the PRI and is defined as the number of times in one second that a radar completes the transmit/receive cycle [37]. The PRI of a radar plays a significant role in its performance [2].

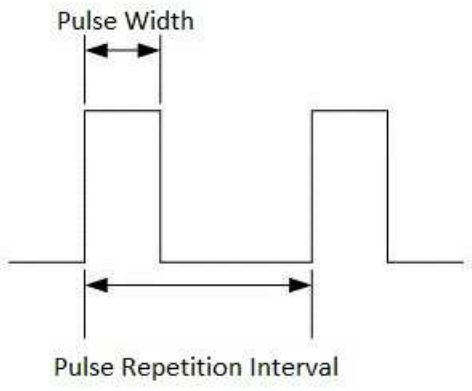


Figure 2.2: Pulsed Radar Waveform

The range of the target is found based on the time difference between the transmitted pulse and the received reflected pulse as stated by equation 2.1. The maximum range for which an echo pulse can return to antenna

before it begins transmitting the next pulse is called the maximum unambiguous range. After the maximum unambiguous range, range ambiguities start to occur [35]. It is possible for the echo of a target further away than the unambiguous range to return to the receiver after another pulse has already been transmitted [35]. In this situation the wrong range of the target will be determined by the radar as the delay between the second pulse (instead of the first pulse) and the echo will be used in the range calculation, known as a range folding [35]. All ranges above the maximum unambiguous range are essentially "folded" back into the ambiguous range interval [66]. The maximum unambiguous range (R_u) can be calculated with PRI of the radar using the following equation.

$$R_u = \frac{c_o PRI}{2} = \frac{c_o}{2PRF} \tag{2.2}$$

Pulsed radar systems usually use one antenna to transmit and receive the radar signal. During transmission, the receiver is disconnected to prevent the transmitter's high-power EM waves from damaging its sensitive components [37]. The minimum range that an object needs to be away from the antenna of a pulsed radar system (R_{min}) corresponds to the range in which an echo from that object will not be received because it is transmitting a pulse and the receiver is disconnected [35]. The minimum range of a pulsed radar system, therefore, depends on the pulse width of the pulse being transmitted and the switching time (t_{switch}) of the duplexer used to switch between transmitter and receiver, as per the equation below.

$$R_{min} = \frac{(\tau + t_{switch})\, c_o}{2} \tag{2.3}$$

The blind ranges (R_{blind}) of a pulsed radar system refer to ranges in which objects are located, that produce echoes that return to the antenna while it is transmitting [65]. These objects are therefore not detected at all, similar the case of R_{min}. Pulses are transmitted every PRI, meaning the receiver is disconnected every PRI. Blind ranges are therefore located at multiples of $R_u$and last for R_{min}.

Range resolution ($\triangle R$) of a radar system is its ability to distinguish between two or more closely spaced targets [37]. The range resolution (measured in meters) of a simple pulsed radar is determined by its pulse width [2], see the pulse width dependent part of the equation below.

$$\triangle R = \frac{c_o \tau}{2} = \frac{c_o}{2B} \tag{2.4}$$

To achieve better ranges on radar systems the transmit power needs to increase, making pulse widths longer [55]. However, longer pulse widths mean a higher range resolution which is undesirable. On newer more complex radars, pulse compression is used to improve the range resolution without decreasing pulse width [4]. If a pulse is processed coherently, explained below, the range resolution is inversely proportional to bandwidth (B) of the signal [6, 37]. By increasing the bandwidth of the EM wave in the pulse, the range resolution is improved. This effect can be seen in the bandwidth dependent part of equation 2.4. The commonly used modulation techniques in pulse compression are linear frequency modulation (LFM), non-linear frequency modulation (NLFM) and encoded pulse phase modulation [1].

A pulsed radar can either be coherent or non-coherent [64]. Non-coherent radar systems only detect the amplitude of the received EM wave, whereas a coherent radar system detects both the amplitude and phase of

the received EM wave [37]. Non-coherent radar can be used in cases where the expected target signal power (amplitude) will be greater than any clutter signal [37]. Early radar systems where non-coherent and therefore relied on the operator to distinguish between a target and clutter returns.

Most modern pulsed radar systems are coherent. Coherent radar signals are produced by transmitting intervals of a reference sinusoid [57]. The reference sinusoid is implemented in the form of a local oscillator, which also serves as a reference for the received signal. A quasi-coherent or coherent-on-receive radar system stores the phase of every pulse that it transmits [63]. The reference phase used for measuring the phase of the received signal is usually the phase of the most recently transmitted pulse. By measuring the phase of the received pulse, the radar can determine the phase shift of the signal. The phase shift can be used to provide target motion information and the ability to image a target [37].

Doppler information of a target can only be extracted using a coherent radar system. The phase shift, calculated using the difference in phase between two returns of the same target, can be used to determine the instantaneous frequency of the returned signal. Using equation 2.5, instantaneous frequency (f_i) of a signal is proportional to the time derivative of the signal phase (ϕ) [63].

$$f_i = \frac{1}{2\pi}\frac{d}{dt}\phi\left(t\right) \tag{2.5}$$

If a target moves with a radial velocity that produces a 360º phase shift, or a multiple (n) of it, the phase shift of the signal will be regarded as zero. These radial velocities are what is know as blind velocities, make targets appear to be stationary. The blind velocity of the radar system can be calculated using its PRF and transmitted frequency (f_{tx}) [6], as shown below.

$$v_{blind} = \frac{nc_o}{2PRF\, f_{tx}} \tag{2.6}$$

The pulse Doppler radar system is a coherent radar system that provides radial velocity information of the target in addition to the information (direction and range) provided by a simple pulse radar [1]. As in the case of a simple pulse radar, the pulse Doppler radar periodically transmits pulses of radar waves. However, if there is relative motion between the target and antenna, there is a frequency shift in the echo of the target. The Doppler frequency shift (f_d) is the difference between frequency of the received wave (f_i from equation 2.5) and the transmitted frequency (f_{tx}) [37]. The radial velocity of a target (v_r) can be calculated with the Doppler frequency shift, using equation 2.7. The radial velocity (and f_d) is positive for a target approaching the antenna [37].

$$f_d = \frac{2v_r f_{tx}}{c_o} \tag{2.7}$$

To increase the unambiguous range of a radar system, the PRF has to be decreased [66]. However, with pulsed Doppler radar, decreasing PRF decreases the maximum unambiguous velocity (v_u) that it can measure. This trade-off is known as the "Doppler dilemma" [2, 67]. The maximum unambiguous velocity that a Doppler radar can observe is at a velocity that produces a ± 180º phase shift in the return signal [67]. It is also called the Nyquist velocity [67]. If a target moves half a wavelength away from the antenna between two consecutive pulses, it produces a phase shift of $+180$º, if it was moving toward the antenna it would produce a phase shift

of -180^{o}. The phase shift is essentially the same, but the direction is ambiguous. This is the first velocity ambiguity, therefore equation 2.8, shows the relationship between PRF and maximum unambiguous velocity [66, 78]. Velocities greater than v_u are folded back into the v_u interval, similar to range folding.

$$v_u = \pm \frac{c_o PRF}{4 f_{tx}} \tag{2.8}$$

Pulsed radars operate in one of three PRF regimes. These different regimes are low PRF, medium PRF and high PRF [2]. A radar with low PRF receives the echoes of targets in all ranges of interest before transmitting the next pulse. In this type of radar, range measurements are always unambiguous and are as a result used as search radars. Low PRF radars are usually moving target indicator (MTI[6]) radars [36]. The echoes of targets in high PRF radars return with Doppler shifts less than the PRF. Therefore the velocity of targets in these radars is always unambiguous. Medium PRF radars only receive target echoes after few PRIs elapse. These echoes also have Doppler shifts that are several times greater than the PRF. This results in both the velocity and range measurements of these radars being ambiguous [2]. High and medium PRF radars are usually pulsed Doppler radars [36].

2.3 Pulse Repetition Interval (PRI)

From the previous section, the PRI of a radar system influences the ambiguous ranges and velocities of a pulsed radar system. The blind ranges and velocities are also affected by the PRI. Accidental variations in PRI are unavoidable and are present in all radars. These variations are caused mainly by the components used in the radar system, such as the transmitter, amplifier and oscillator [2, 6]. However, in some cases, there are intentional variations included in the PRI. Sometimes even going as far as changing the PRI. The number of these PRI schemes is seemingly infinite [2]. The PRI determines the maximum unambiguous range and unambiguous radial velocity of a target that the radar can detect. By switching between PRIs, the radar system can operate at different unambiguous range and velocity values for any given time or even unfold the unambiguous range and velocity measurements [65]. The most widely used PRI schemes were given category names. The most commonly used PRI schemes in pulsed radar systems are: constant, jittered, dwell and switch, stagger, sliding, scheduled, periodic variations and pulse groups. One of the major goals of electronic intelligence (ELINT) signal analyst is to analyse a radar signal and then classify that emitter under one of these PRI types [2]. Each PRI type can be associated with a radar function, as seen in Table 2.1.

[6] A type of radar system that only detects if objects with a radial velocity is in the direction of the antenna.

Type	Typical Function
Constant	Common search and track radars.
Jittered	Reduces the effects of some types of electronic attack.
Dwell & Switch	Resolves velocity and range ambiguities.
Stagger	Eliminates blind speeds in MTI systems.
Sliding	Provides constant altitude coverage during elevation scanning.
Scheduled	Used in electronic scan, multiple function computer controlled systems.
Periodic Variations	Missile guidance systems.
Pulse Groups	May improve velocity or range resolution.

Table 2.1: Common PRI types And Their Function, adapted from [2]

The common PRI schemes are explained in greater detail below.

2.3.1 PRI Schemes

2.3.1.1 Constant or Stable PRI

A constant PRI scheme (see Figure 2.3) is a PRI scheme where the PRI has no variations. If variations do exist, they are purely accidental and are no more than about 1% of the mean PRI [2]. If a circuit malfunctions and a large variation does occur, it serves no purpose. The mean PRI value is used to determine the maximum ambiguous range and velocity as explained in equations 2.2 and 2.8. These radars are commonly used as search and track radars [2]. A constant PRI pulse train can be seen in Figure 2.3.

2.3.1.2 Jittered (Step) PRI

Radars with a jittered or step PRI scheme have intentional random variations added to their PRI each pulse. This type of PRI scheme is used in radars as a form of electronic defence against some types of jamming [2], as their PRI is continually changing. The parameters of interest for a jittered PRI radar are similar to a constant PRI one; only more emphasis should be placed on the jitter pulse train and the statistics of the pulse train [2]. The ELINT analyst should be able to determine the mean PRI of the pulse train, the distribution curve of jitter and the range in which the PRI varies. The common distribution curves of the random jitter are Gaussian, uniform and U-Shaped [4]. Jitter can either be cumulative or non-cumulative. Non-cumulative jitter can be seen as jitter added to each pulse in a constant pulse train. This means each pulse will be transmitted around the same time it should have been transmitted if it was a constant PRI radar system with some variation. Cumulative jitter on the other hand, adds the jittered PRI after the previous pulse. This may result in the pulse train shifting slightly when compared to the constant pulse train, depending on the distribution of the jitter. Figure 2.3 shows a cumulative and non-cumulative jitter pulse train as compared to a constant pulse train.

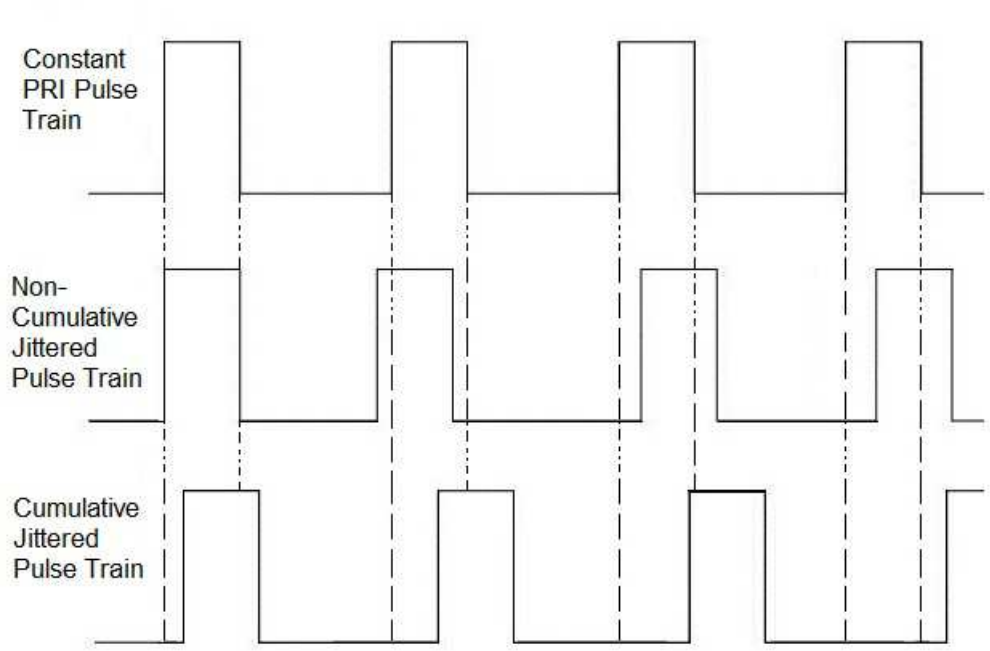


Figure 2.3: Jittered PRI Pulse Train In Comparison To Constant PRI Pulse Train

2.3.1.3 Staggered PRI

Blind speeds in MTI (Moving Target Indication) radar systems are usually eliminated with the use PRI stagger [65]. A staggered PRI scheme is a sequence of two or more PRIs in a repeating fixed pattern (periodic) [2, 3]. A stagger sequence may contain more than one instance of a PRI. Figure 2.4 shows an example of a staggered PRI pulse train, with different intervals in between pulses.

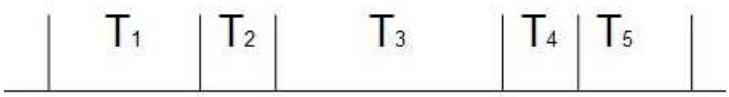


Figure 2.4: Staggered Pulse Train

Staggered PRI schemes are classified by the number of "positions" and number of "levels" [2]. Positions are the number of PRIs in a sequence before it repeats itself and levels are the number of different PRI elements present in the sequence. For example, the staggered PRI sequence in Figure 2.4 has four distinct pulse intervals (because $T_2 = T_4$) and a period of five intervals. It is referred to as a 4-level, 5-position staggered PRI sequence with stagger elements of T_1, T_2, T_3 and T_5. The period (the summation of all PRIs present before the sequence repeats itself) of the staggered PRI scheme is referred to as the frame rate [27].

2.3.1.4 Dwell and Switch PRI

This type of PRI scheme uses different PRIs, and periodically switches between them. The dwell time is the time PRI is used by a radar before it is switched to another PRI [2, 5]. Each PRI will have its own specific dwell time. Dwell and switch radars are used to eliminate velocity or range ambiguities in pulse Doppler radars and to remove blind speeds in MTI radars [65]. The parameters of interest for a dwell and switch PRI radar include the PRIs used, the switching between PRIs pattern and the dwell times, in addition to parameters of interest of a constant PRI radar [2]. The figure below shows a dwell and switch PRI scheme with two PRIs.

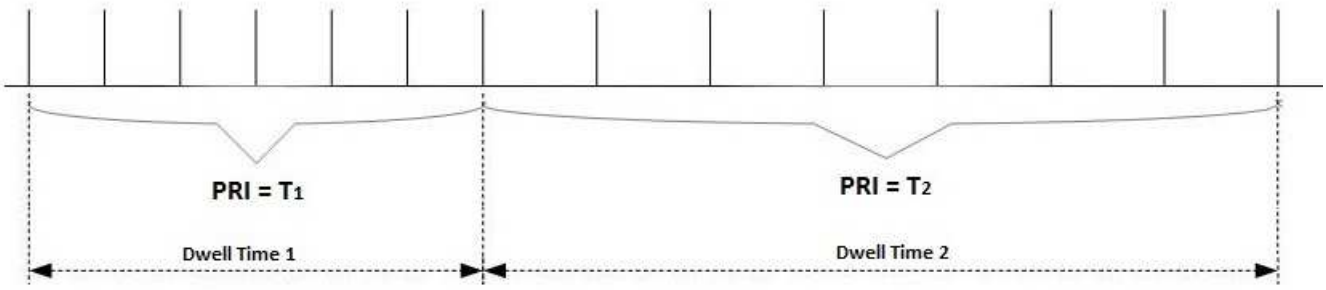


Figure 2.5: Dwell and Switch Pulse Train

A dwell and switch PRI scheme is similar to a traditional stagger PRI scheme. However, the distinction is made with a dwell and switch scheme having between four and eight consecutive pulses of the same PRI before a switch [69].

2.3.1.5 Sliding PRI

The sliding PRI scheme consists of a PRI that is continuously changing, either increasing or decreasing within its maximum and minimum PRI limits [6]. The change in PRI from minimum to maximum is called the PRI sweep. The sweep time is the time required to change from the minimum to maximum PRI [5]. The sliding pattern of the PRI, which defines how the PRI sweep increases or decreases, is usually periodic. An example of this sequence can be seen in Figure 2.6. This type of PRI scheme is used to eliminate blind ranges [2]. An ELINT analyst should be able to determine the PRI sliding pattern, the minimum and maximum PRI values, as well as the time taken to complete one sweep [2].

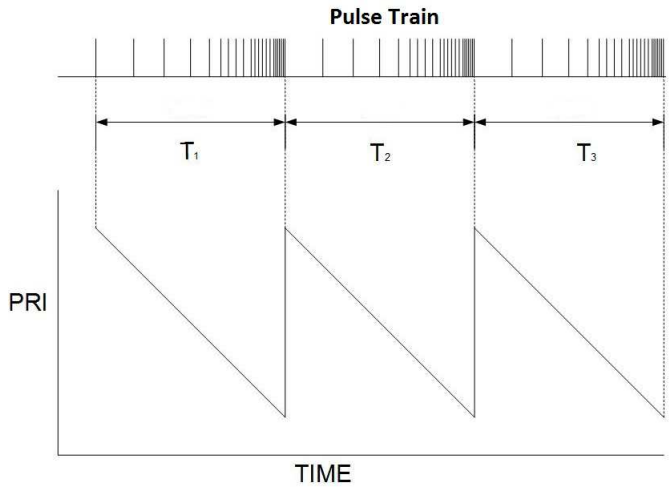


Figure 2.6: Sliding Pulse Train

2.3.1.6 Scheduled PRI

Computer controlled electronic scan radars use scheduled PRI schemes to switch between search and track functions. The variations in the PRIs used and their sequences are determined by the controlling software of the computer [6, 2]. The number of PRIs used in this scheme are dependent on the number of targets being tracked, their location and the environment [2]. The ELINT analyst should determine the all PRIs used and their typical sequences. Correlating this information with characteristics of the target is useful.

2.3.1.7 Periodic PRI Variations

Periodic PRI variations are similar to that of sliding PRI, except the PRI sliding pattern is sinusoidal. It can be used to eliminate eclipsing (blind ranges) for ranging but is more commonly used with conical scan tracking systems as a missile guidance system [2, 6]. An ELINT analyst should determine average PRI, the minimum and maximum PRIs as well as the frequency of the sinusoidal PRI sliding pattern [2].

2.3.1.8 Pulse Groups

Some radar systems may emit a series of pulses at short intervals, called groups, followed by a longer interval before the next group [2]. Pulse repetition group intervals (PRGIs) may be handled exactly as PRIs. Pulse groups can be used to increase range and velocity resolution of pulsed Doppler radar system and eliminate blind speeds in MTI systems. An ELINT analyst should be able to determine the pulse durations, the interval between pulses together with PRGIs [2]. If the number of pulses in a group changes, the ELINT analyst should also be able to determine the maximum and minimum pulses in a group [2].

2.4 Electronic Warfare

From the previous sections, it can be seen that the choice of PRI heavily influences the performance of a pulsed radar system. Therefore, determining the PRI scheme of an unknown radar system is an essential part of electronic warfare.

Electronic warfare (EW) can be defined as the act of preserving the use of the electromagnetic spectrum for friendly EM systems while impairing or denying its usage to enemy EM systems [57]. EW can be broken down into three subdivisions, namely electronic attack (EA), electronic protection (EP) and EW support (ES) [56]. The subdivisions named is the current standard. Previously, electronic warfare could be divided into electronic counter measures (ECM), electronic counter counter measures (ECCM), electronic support measures (ESM) and anti-radiation weapons.

Electronic attack primarily deals with using EM energy or directed energy[7] or anti-radiation weapons[8] to attack enemy facilities or equipment with the intent of degrading or destroying their combat capability [56]. Electronic attack includes any action carried out to reduce or prevent the enemy from effectively using the EM spectrum, with actions such as jamming and EM deception [56]. It also includes weapons that use EM or directed energy as their primary destructive mechanism [56].

Electronic protection was previously known as ECCM. Electronic protection involves both passive and active actions taken to protect facilities or equipment against the effects of enemy and friendly EM systems that could degrade or destroy combat capability [56]. Electronic protection should not be confused with defensive electronic attack. Defensive electronic attack protects against potential electronic attacks by denying enemy weapons use of the EM spectrum, a form of EA. Conversely, EP protects against the effects of EA.

Electronic Support systems perform surveillance of an area to determine the capability and identity of radar emitters in that area [28, 57]. Electronic support was previously called ESM. Passive[9] ES systems monitor the EM spectrum by measuring the parameters of intercepted pulses [26]. The measured parameters for each pulse are combined and stored as a single data packet known as pulse descriptor words (PDWs) [51, 61]. The parameters that are measured and stored vary from system to system and are dependent on the application of the system [61]. PDWs are stored on a DRFM (digital radio frequency memory).

The primary purpose of the DRFM is to store the received transmit pulse of a radar with the intention to re-transmit the pulse, slightly alerted, to produce a fake target on the radar [68]. The DRFM can create a realistic target with a Doppler frequency shift, range and a specific RCS (Radar Cross Section) with a Swerling case[10] [68]. This is where being able to predict the next pulse TOA of the target radar system comes into play. The DRFM usually consists of an analogue to digital converter (ADC), a field programmable gate array (FPGA), digital memory and a digital to analogue converter (DAC) [68]. The DRFM can, therefore, be used as part of an EA or ES system. The CSIR fifth generation DRFM will be used to implement the system outlined in Chapter 1. Specifications, schematics and additional information on the CSIR fifth generation DRFM can be found in [83] and [84].

[7] A concentrated beam of EM energy or atomic or subatomic particles.

[8] Weapons designed to detect and destroy an enemy radio emission source.

[9] Passive systems only receive and do not transmit signals.

[10] Used to describe how the RCS of a target changes from pulse to pulse or scan to scan.

One possible task of an ES system is to sort and classify the received pulses, using their PDWs, in a process known as deinterleaving [28].

2.5 Deinterleaving

More often than not, an EW receiver will receive multiple periodic pulse trains[11] from different sources that are all superimposed [33]. The resultant pulse train that the receiver receives is referred to as an interleaved pulse train, which can be seen clearly in Figure 2.7. Pulse train deinterleaving is the process that determines the number of pulse trains present in the interleaved pulse train, as well as associating each pulse with a source [33], which is also shown in Figure 2.7. This is done by detecting and extracting repeating pulse parameters [28]. One application of pulse train deinterleaving is radar detection [30]. This deinterleaving process relies on the assumption that different pulse train sources have different characteristics or parameters. Therefore, all deinterleaving algorithms are based on the analysis of these parameters.

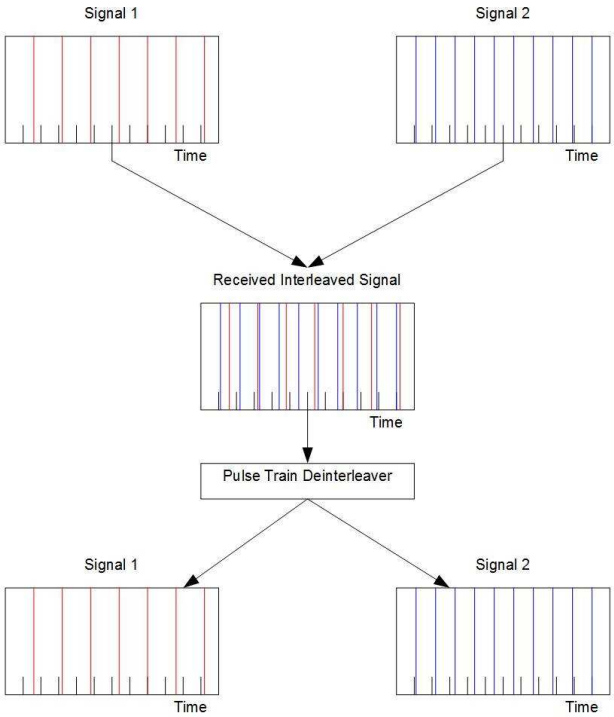


Figure 2.7: Concept of Interleaving and Deinterleaving

The major pulsed-radar emitter parameters that an electronic support system can measure are pulse amplitude (PA), angle or direction of arrival (AOA or DOA), pulse width (PW), time of arrival (TOA), polarisation and carrier radio frequency (RF) [10, 28, 51]. Using the difference in time of arrival between two pulses, the PRI can be estimated. This may not be limited to consecutive pulses as due to the superimposition of pulse trains, two consecutive pulses could originate from different sources. The parameters of each pulse are made available to the deinterleaving algorithms using PDWs.

[11] The pulse train of pulsed radar is referred to as a periodic pulse train.

Typically only the TOA and AOA parameters are employed in deinterleaving algorithms [25], as certain parameters are unreliable when identifying emitters, for example, the channel status greatly influences the PA [25, 51]. Parameters such as AOA are difficult for an emitter to vary and remain relatively constant to the EW receiver during a scanning interval. These parameters are therefore used to cluster pulses together before deinterleaving can occur [58]. This makes deinterleaving easier and faster as there are fewer pulses present to process in each cluster. Figure 2.8 shows the structure of a typical deinterleaver. After PDWs for each pulse are generated, clustering of the pulses may then be carried out. Thereafter, a deinterleaving algorithm will deinterleave each cluster.

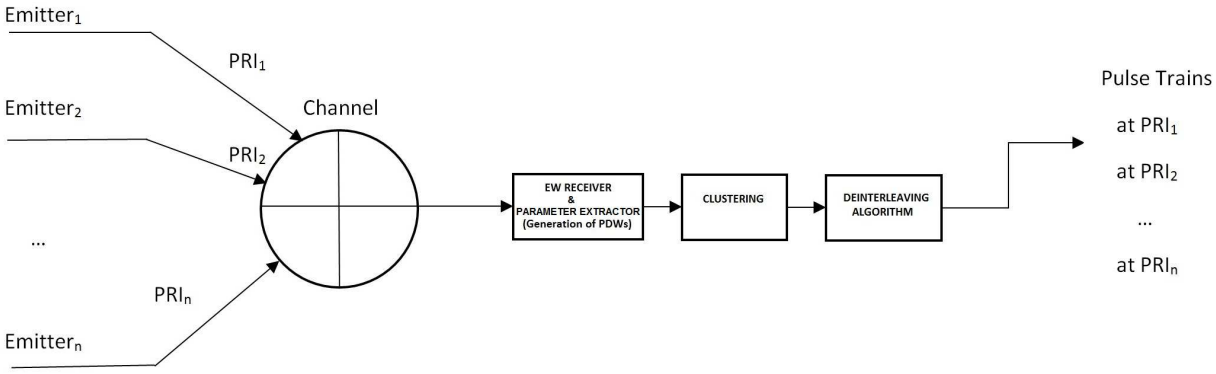


Figure 2.8: Typical Deinterleaver

As part of the research objectives, it is stated that algorithms that deinterleave radar emitters using only TOA information needed to be researched.

2.6 TOA Based Deinterleaving Algorithms

The solution system, defined in Chapter 1, will receive pulses from emitters that will have constant, dwell and switch, jittered and staggered PRI schemes. A jittered PRI scheme can be thought of as a constant PRI with more noise present [3]. We can, for this reason, treat jittered and constant PRI schemes in a similar fashion in subsequent sections. Meaning if an algorithm is suitable for a noisy constant PRI pulse train, it should also be suitable for a jittered PRI pulse train. A dwell and switch pulse train consisting of x number of PRIs can also be considered as a x level[12] staggered PRI pulse train [69]. Algorithms suitable for staggered PRI are therefore also suitable for dwell and switch PRI schemes.

All the deinterleaving algorithms that were researched in this Chapter only extract constant PRI pulse trains from an interleaved pulse train. This is acceptable because a staggered PRI pulse train of level x will be extracted as x constant PRI pulse trains, all having the same PRI [10]. The PRI will be equal to the frame rate of the staggered PRI scheme [27]. Each algorithm is discussed in detail below.

[12]Level is the number of unique PRIs in a PRI sequence, see section 2.3.

2.6.1 Pulse Sorting Algorithm

The Pulse Sorting Algorithm is a simple algorithm that identifies adjacent matching PRIs in the pulse train. Firstly, three adjacent matching PRIs are searched for [6]. When such an identification is made, the PRI sequence for this identified PRI is extended in both directions to discover and remove all instances of it in the original pulse train [6]. This process is repeated until all three adjacent matching PRIs are removed from the pulse train. Finally matching adjacent PRI pairs are identified and removed as before in the case of three adjacent matching PRIs [6]. The algorithm ends when all three adjacent and pairs of adjacent matching PRIs are removed.

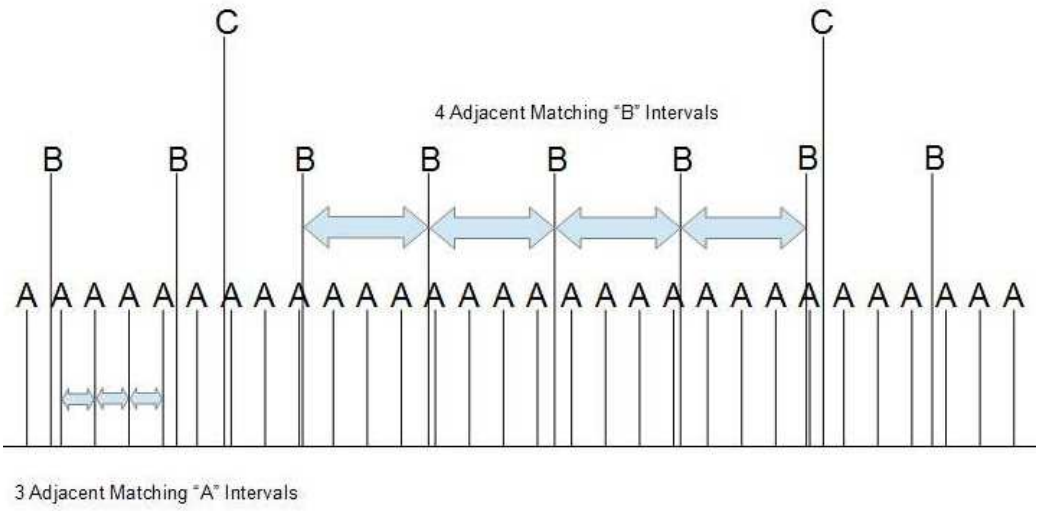


Figure 2.9: Example of Pulse Train

Figure 2.9 is an example of an interleaved pulse train with three different signals (A, B and C), each with their own PRI. Using the pulse sorting algorithm discussed above, signal A will be detected first as there are three adjacent matching PRIs. The signal will be extended in both directions and will be removed from the pulse train. Thereafter, signal B will be removed as three adjacent matching PRIs now exist for that signal because signal A was removed. Finally, signal C will be removed.

This method is similar to the sequence search algorithm, explained in the following section. However, in the sequence search method, pulses do not have to be adjacent for a match to be made. For a pulse train to be extracted using the pulse sorting algorithm, it needs to have PRF that is at least double the PRF of the next highest PRF signal i.e. less than half the PRI for at least two PRIs to be adjacent. This will not be the case in most situations, and as a result, this method will not be simulated as the sequence search algorithm is very similar but applies to more situations.

2.6.2 Sequence Search Algorithm

The sequence search (SS) algorithm is one of the least complex TOA based deinterleaving algorithms. At the expense of processing speed, due to the larger number of computations, the algorithm provides greater

reliability and accuracy than histogramming based deinterleaving methods [27]. This algorithm works by estimating sequences with all the possible PRI values and matching these sequences are with the received interleaved pulse train. A tolerance window or gate is used when matching to account for variations in the PRI [29]. A typical tolerance window assumes variation in a pulse train is no more than 10% of the mean PRI [29]. This equates to ± 3 standard deviations from the mean PRI [29].

The sequence search algorithm can be broken down into two parts. The first part of the algorithm is sometimes referred to as the primer algorithm[13] [47].The first step in the primer algorithm is to make an initial PRI estimate. The algorithm then uses the TOA of the first pulse in the buffer or cluster (at t_1) as the first pulse to be matched with the estimated PRI sequence. It then searches for pulses at $(t_1 + PRI)$ and at $(t_1 + 2PRI)$ [26], keeping in mind tolerances. If these pulses are not found, the estimated sequence is not a possible match and the second pulse TOA in the buffer is now used as the first pulse to be matched with the estimated PRI sequence. This process repeats, using each subsequent TOA in the buffer to match with the estimated PRI sequence. When a possible match is found, the algorithm moves onto the second part of the algorithm. Should the algorithm reach the end of the buffer without finding a pulse train, the primer algorithm begins again with new PRI estimate until there are no more PRI estimates to be made. It should be noted that the initial estimates of the PRI are the smallest to avoid extracting multiples of the correct PRI [27, 29]. Other than the previous point, the algorithm proposes no other guideline in choosing the PRIs to be tested [26].

The second part of the sequence search algorithm determines if the possible match from the primer algorithm is an actual pulse train present in the buffer. It continues searching forward in the buffer for pulses that match the estimated PRI sequence [27], only stopping when two or more consecutive pulses from the sequence are not found. If the pulses successfully matched is greater than a predefined value or threshold, the estimated sequence is considered to be an actual pulse train present in the received pulse train. The pulse is then removed from the interleaved pulse train in the buffer to lessen the complexity of future processing [26, 29]. The threshold is set by the user to achieve the desired sensitivity of the sequence search algorithm [26]. The author Mardia specifies a minimum of five pulses to identify a sequence in [27].

Assuming that all possible PRI estimates of a sequence can be from the TOA of the first received pulse to the last received pulse, the number of computations required to search all possible sequences will need the order of N^2 computations [27]. Where N is the number of possible PRIs. This is much higher than the number of computations required for Histogramming methods.

In dense radar environments, measurement errors and missing pulses will occur. For this reason, a more robust algorithm is needed [27]. The original sequence search algorithm has undergone different modifications in attempts to accomplish this. For example researchers at TUBITAK[14][47], have reduced the three pulse initial search in the primer algorithm to two in an attempt to identify more emitters with fewer pulses. Aslan [26] also suggests tracking the mean and standard deviation of all pulses from found pulse trains to adjust the tolerances allowed when running the sequence search algorithm. In order to use the least amount of computations possible, Bildøy [29] uses the time between any pulse and all subsequent pulses present in the buffer to estimate the PRIs to be tested. This approach was implemented for the simulations in section 3.2 and the flow chart for this method can be seen in Figure 2.10.

[13]The primer algorithm is the part of the sequence search algorithm that is similar to the pulse sorting algorithm.

[14]The Scientific and Technological Research Council of Turkey.

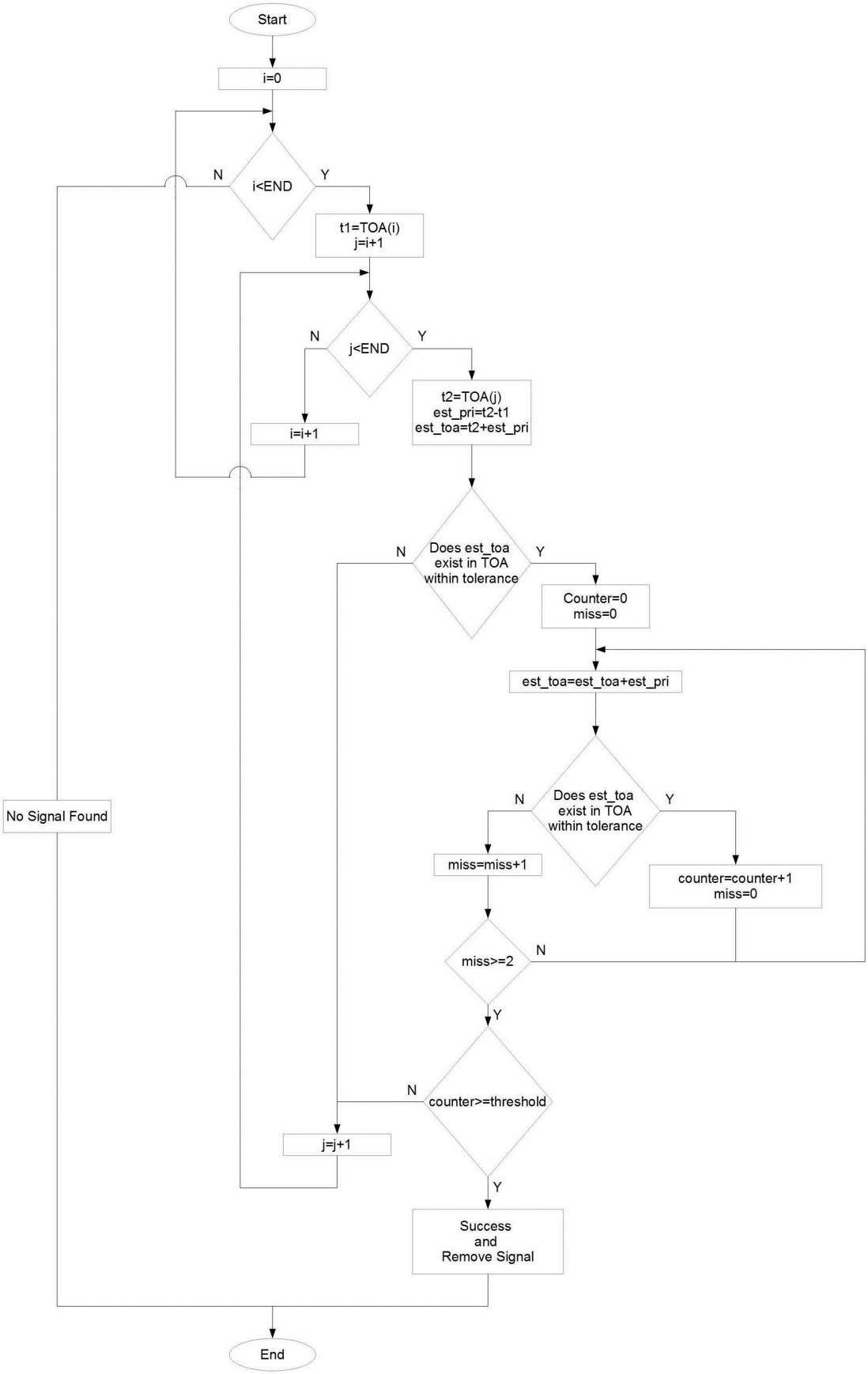


Figure 2.10: Sequence Search Algorithm Flowchart

2.6.3 TOA Difference Histogramming

The TOA difference histogram is another simple algorithm [27]. It is sometimes referred to as the all difference histogram (A-DIF) and the Delta-τ histogram [26]. As the name suggests, the histogram is constructed by binning the differences in TOA between every and all possible combinations of TOAs in the set of TOAs [27]. As this algorithm is based on subtractions, it offers faster processing than the sequence search algorithm [27]. The number of computations required for E elements in the buffer to be processed is in the order of

$$\sum_{i=0}^{E} i \approx \frac{E\,(E-1)}{2} \tag{2.9}$$

where $E \gg 1$ [29].

Applying this technique to a single constant PRI pulse train will result in peaks in the histogram at the PRI and at multiples of the PRI [27]. When several pulse trains are present, peaks also occur in the histogram at multiples, sums and differences of all the different PRIs, leading to ambiguous results [27]. A threshold must be defined to differentiate between peaks of the actual PRIs and these other peaks. If the frequency of one of the bins is above the threshold, a pulse train with PRI equal to the difference associated with that bin is said to be present [29]. Once a pulse train is found, it is removed from the interleaved pulse train, and the histogram has to be redrawn.

The threshold must be defined while taking the effects of missing and spurious pulses[15] into account [27]. The amount of non-detected and false identifications due to these interference pulses are determined by this threshold [27]. The threshold is, for this reason, usually determined experimentally for an acceptable non-detection and false identification rate [9]. Some authors like Mardia [27] recommend that the multiple (harmonic) of the found PRI should also be above the threshold for it to count as a detection. However, if the PRI was not identified accurately (due to incorrect bin sizes), the harmonic check will fail [26]. If there is no harmonic checking, the algorithm can stop as soon as the threshold is exceeded, decreasing processing time [48].

The threshold for the TOA difference histogramming method is set by [26]

$$Threshold = k\left(\frac{T}{PRI} - h\right) \tag{2.10}$$

where T is the total time interval of the data used in the histogram and h is the harmonic number [26]. For example the threshold to test for an emitter: $h = 0$ for the bin at PRI, $h = 1$ for the first harmonic $(2 \times PRI)$ bin and so on. k is a constant between 0 and 1 determined experimentally that sets the non-detection and false identification rate of the algorithm [26].

From the requirements in section 1.2.2, an emitter needs to be identified after four pulses. This equates to three PRIs, therefore the threshold in the simulations of the algorithm (section 3.2) was set to a constant three instead of the threshold mentioned here. There is also no harmonic checking implemented in the simulations.

[15] Collectively called interference pulses.

2.6.4 Cumulative Difference (CDIF) Histogramming

The cumulative difference (CDIF) histogram algorithm is an improvement over the TOA difference histogram algorithm [26]. The CDIF histogram is created using different levels (or depths) of TOA differences [10]. TOA differences between any pulse and the pulse that precedes it is called the first level differences [10]. TOA differences between any pulse and the second pulse that was received before it, is called second level differences. Continuing this trend, the n^{th} level difference refers to the TOA differences between any pulse and the n^{th} pulse that arrived before that pulse.

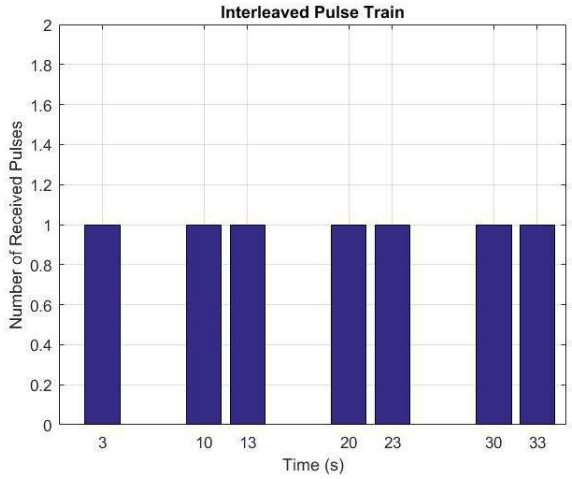
The algorithm starts by creating a histogram with only the first level differences. If the frequency of one of the bins exceeds a predetermined threshold function (explained in section 2.6.4.1), an emitter with a constant PRI corresponding to that bin is assumed to be detected [9]. The pulses corresponding to the detected PRI are removed, and the algorithm begins again without those pulses. If the count in more than one bin exceeds the threshold, the lowest PRI is assumed to be detected [48].

However, if none of the counts from any of the bins exceed the threshold, the algorithm moves on and calculates the next level differences [29]. In this case, the second difference values are calculated. A new histogram is then created accumulating all the difference values from the current difference level as well as the lower levels, giving the algorithm its name [48]. In this case, the new histogram has both the first and second difference values. A new threshold function for the current difference level is also calculated and used to check if a pulse train can be extracted. The process continues increasing difference levels until an emitter is detected and extracted or the algorithm stops at a predetermined difference level [26].

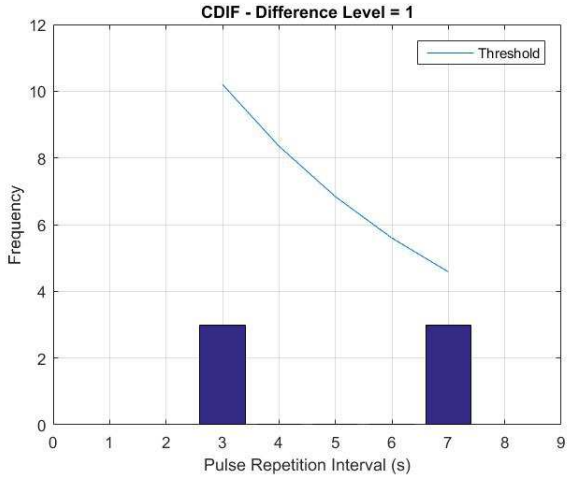
The CDIF histogram algorithm uses fewer differences than the TOA difference histogramming algorithm, resulting in clearer peaks [27]. Meaning the peaks at multiples, sums and differences of all the different PRIs are not as dominate as they were before. It also uses fewer computations as compared to the TOA difference histogramming algorithm [27]. This is because the CDIF histogram stops at a predefined difference level and not all difference levels are computed [70]. The number of computations required for E elements in the buffer to be processed for x difference levels are in the order of

$$\sum_{i=0}^{E} i \approx \frac{E^2 - (E-x)^2}{2} \tag{2.11}$$

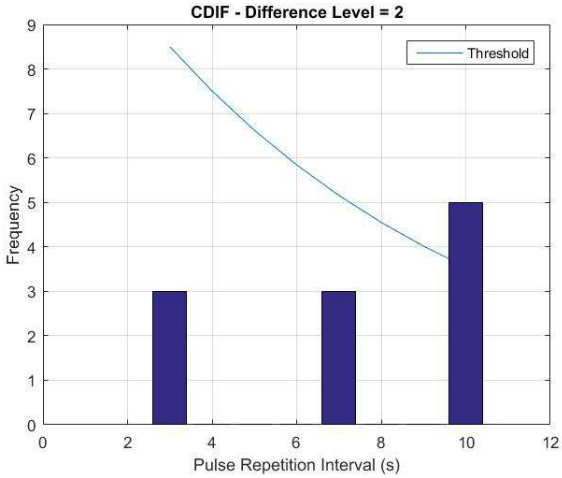
A simple example of the CDIF histogramming algorithm can be seen in Figure 2.11. The interleaved pulse train together with each difference level histogram and threshold is also shown. The interleaved pulse train has two signals with a PRI of 10s. The TOA of the first pulse for first pulse train is 3s and the TOA of the first pulse for the second pulse train is 7s. When the first difference histogram is drawn, peaks are found at 3s and 7s but they do not exceed the threshold. In the second difference histogram, a new peak that exceeds the threshold forms at 10s, meaning a pulse train with a PRI of 10s is detected. As there is an accumulation of difference values, the peaks at 3s and 7s are still present even though no second level difference values were 3s or 7s. The threshold curve seen in the figures is the optimal threshold function and is explained in the following subsection.



(a) The Interleaved Pulse Train



(b) First Difference Level



(c) Second Difference Level

Figure 2.11: CDIF Histograms and Threshold

2.6.4.1 Optimal Threshold Function

The threshold function is a very crucial component to any histogramming technique [29]. The threshold value is inversely proportional to the bin number τ [10]. This is the case as the histogram bins correspond to the time interval between different pairs of pulses. Therefore, a higher bin number equates to a longer time interval between pulses. Since the observed time period in which the histogram is compiled for is bounded, it can stand to reason that the more time between pulses of a signal is observed, the fewer instances of that signal will be present within the observed time period [29]. This means the threshold required for a detection can decrease as the bin number increases.

The widely accepted [9, 10, 25, 28, 29, 48, 58, 70] optimal threshold function is :

$$Threshold\,(\tau) = x\,(E-c)\exp\left(\frac{-\tau}{kN}\right) \tag{2.12}$$

Where:

- τ is the bin number
- E is the number of pulses observed
- c is the difference level
- N is the total number of bins in the histogram
- x and k are constants between 0 and 1, determined experimentally [9].

Bildøy proposes checking if both the interval and double the interval are above the threshold before a detection is assumed [29]. However like in the previous algorithm, an inaccurate PRI can cause the harmonic checking to fail.

The threshold curves used in Figure 2.11 is the optimal threshold function discussed here. The same x and k constants are used in Figures 2.11b and 2.11c but the curves differ because the difference level and total number of bins in the two histograms differ.

Due to the same reason discussed in section 2.6.3, the optimum threshold function was not used as the threshold in the CDIF simulations in section 3.2. The threshold was set to a constant value of three and no harmonic checking was used.

2.6.5 Sequential difference (SDIF) Histogramming

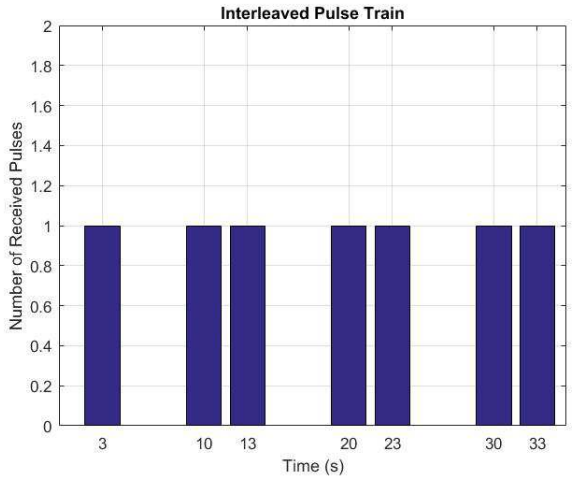
The sequential difference (SDIF) histogram algorithm was introduced as an improvement over the CDIF histogram algorithm [28]. In radar environments where there are many missing pulses, the CDIF histogram algorithm will often falsely detect sequences at harmonics of the PRI instead of at the actual PRI [10]. The SDIF histogramming algorithm is less sensitive to missing and spurious pulses [28]. As a result, SDIF should perform better in radar environments with many missing pulses as well as pulse dense radar environments [29]. The CDIF and SDIF algorithms are widely used in real deinterleaving situations with high efficiency [70].

The SDIF histogram algorithm is the same as the CDIF histogram algorithm except for one difference: in an SDIF histogram, when a new histogram is created for a difference level, only the difference values for that level are used for the new histogram. All previous difference level values are discarded [9]. The optimum threshold function used with CDIF is also used here. As these algorithms are so similar, the number of computations required to perform the SDIF algorithm will be the same for CDIF, as described by equation 2.11.

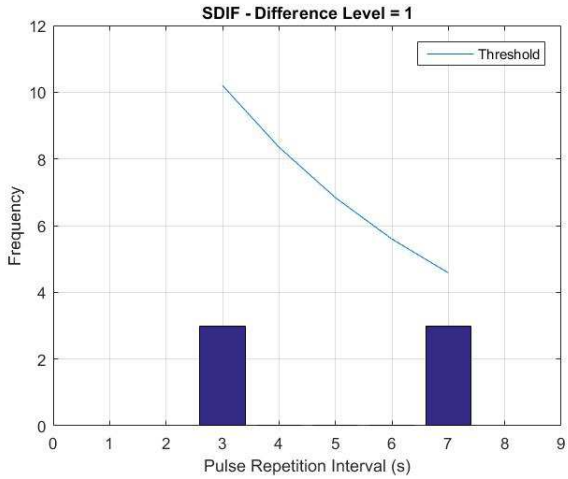
The SDIF algorithm will produce fewer counts that exceed the threshold as there is no accumulation from one difference level to the next [10]. It should also be noted that emitters with smaller PRIs will only be detectable in low difference level SDIF histograms due to their difference values being discarded in higher difference level histograms [25]. Harmonic checking is generally not done with this algorithm as harmonics will only be seen at higher difference level histograms [48].

Figure 2.12 shows the SDIF algorithm processing the same interleaved pulse train like that in Figure 2.11. The constants used in the optimal threshold function curves are also the same. The threshold curve cannot be seen

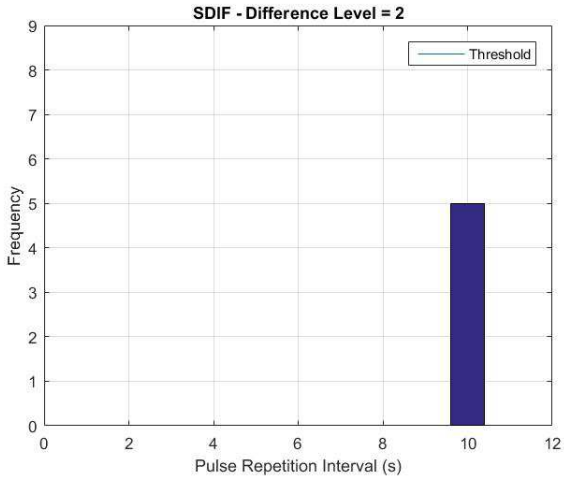
in Figure 2.12c because there is only one bin in the second level difference algorithm and the peak is over the threshold for that bin. Resulting in the correct detection of a pulse train with a PRI of 10s. All difference values from the first level difference histogram are discarded to draw the second level difference histogram. Here we can clearly see fewer peaks and fewer values being added to existing peaks, which proves the claim that the SDIF is less sensitive to spurious pulses true.



(a) The Interleaved Pulse Train



(b) First Difference Level



(c) Second Difference Level

Figure 2.12: SDIF Histograms and Threshold

The optimum threshold function was not used as the threshold in the simulations of the SDIF algorithm in section 3.2. The threshold was set to a constant value of three as explained in section 2.6.3.

2.6.6 The Two-pass Weighted-search Algorithm

The two-pass weighted-search algorithm combines a histogramming algorithm with the sequence search algorithm [10]. The first step of this algorithm is to run the histogramming algorithm [10]. The histogramming

algorithm quickly provides probable PRIs to be tested with the sequence search algorithm in the second step. If a PRI is identified, all pulses corresponding to that pulse train are removed, and the process repeats.

The sequence search algorithm provides more reliable pulse train deinterleaving than all other histogramming based algorithms at the expense of processing speed [27]. Using this two-staged approach, the processing time of the sequence search algorithm is reduced as only the probable PRIs are tested [27]. Figure 2.13 is a flowchart of the two-pass weighted-search algorithm.

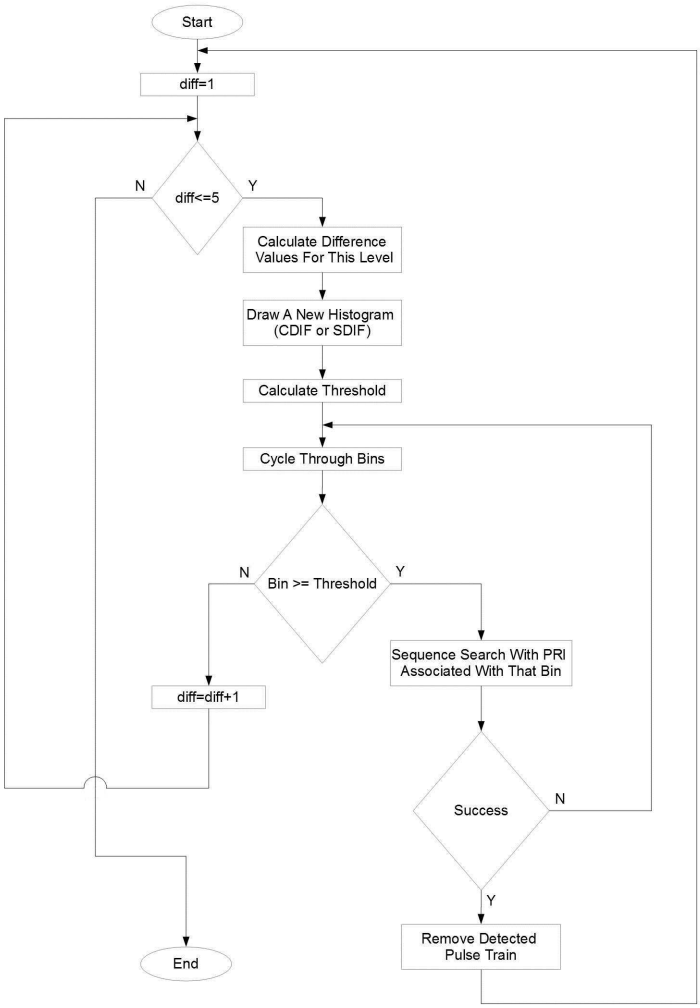


Figure 2.13: Two-pass Weighted-search Algorithm Flowchart

This dissertation investigated this two-pass weighted-search algorithm using both the CDIF and SDIF algorithms with the sequence search. The CDIF with sequence search (CDIF SS) and SDIF with sequence search (SDIF SS) were simulated (see section 3.2) using a constant three as the threshold.

2.6.7 Interleaved Pulse Train Spectrum Estimation

The interleaved pulse train spectrum estimation is a slightly different type of deinterleaving algorithm than the ones previously mentioned [30]. Instead of immediately trying to deinterleave the interleaved pulse train, this

method determines the number of constant PRI pulse trains present and their PRFs. This is done by estimating the spectrum of the interleaved pulse train using only the TOA of the pulses. Once the number of pulse trains and their PRFs present in the interleaved pulse train are known, it is relatively easy to find and remove them [30].

Estimating the spectrum of the interleaved pulse train is less computationally expensive compared to the other deinterleaving methods investigated [33]. For an interleaved pulse train consisting of N pulses, estimating the spectrum will require the order of $N\log_{10}(N)$ computations [30, 33, 59].

The TOAs of an interleaved pulse train, with M number of sources and $N+1$ consecutive pulses can be expressed as [31, 59]

$$t_0, t_1, t_2, t_3, \ldots, t_N \tag{2.13}$$

The first step in estimating the spectrum of the interleaved pulse train (equation 2.13) is to normalise its length to approximately 2π and then wrap this normalised interval around the unit circle [30]. To accomplish this, t_0 is first set to zero and the signal $x(n)$ is calculated. The signal $x(n)$ is the normalised and wrapped interleaved pulse train, defined in equation 2.14 [33].

$$x(n) = \exp\left(j\frac{2\pi}{t_N}t_n\right) \qquad for\ n = 0,1,2,3,\ldots,N-1 \tag{2.14}$$

As j in equation 2.14 represents $\sqrt{-1}$, $x(n)$ is a complex signal. The spectrum of the interleaved pulse train is then estimated by taking the N length discrete Fourier transform (DFT) of $x(n)$. It is important to remember N is related to the number of pulses. The DFT ($X(k)$) is calculated as follows [32]

$$X(k) = \sum_{n=0}^{N-1} x(n)\exp\left(\frac{-2\pi jkn}{N}\right) \qquad , k\ \varepsilon\ Z \tag{2.15}$$

The magnitude plot of the transformed signal carries all the information of interest, i.e., the number of pulse trains present and their PRFs [30]. The magnitude of $X(k)$ at 0 Hz is an artefact of the processing method and must be ignored. The artefact's peak is approximately equal to N [33]. In the case that only one pulse train is present in the interleaved pulse train, only one peak is observed at 0 Hz. This artefact is usually ignored, so in this situation, the PRF estimate corresponding to the pulse train is given by the highest frequency bin [30]. In cases with more than one pulse train present in the interleaved pulse train, the largest magnitude peak (apart from the artefact at 0 Hz) corresponds to a pulse train present, and the frequency corresponding to this peak is taken as the PRF estimate ($P\hat{R}F_1$) for that pulse train [30]. Harmonics of a pulse train's PRF (located at $2P\hat{R}F_1$, $3P\hat{R}F_1$, etc.) also exists in the magnitude plot and is responsible for most of the spurious peaks in the plot [30]. Any harmonics are removed, and their magnitudes are added to the magnitude at $P\hat{R}F_1$ [30]. In practice, the frequency bins around harmonics and the peak are also removed. This practice is to ensure in situations where magnitude from one pulse train is spread over multiple bins, will not result in multiple detections of the same pulse train before searching for the next highest peak. The process repeats for the second highest peak in magnitude, which is associated with the second PRF estimate ($P\hat{R}F_2$), and continues to repeat for each

subsequent highest peak until [30]

$$t_N \left(P\hat{R}F_1 + P\hat{R}F_2 + \ldots + P\hat{R}F_M \right) \approx N \tag{2.16}$$

In the case that no M can be found such that the condition in equation 2.16 is met, the pulse train associated with the largest magnitude peak (artefact ignored) is removed from the interleaved pulse train, and the spectrum is estimated again [30].

When considering this method of deinterleaving for the system to be implemented on the DRFM, outlined in the problem statement, it seems very attractive due to it requiring a low number computations. However, the effect N and therefore the number of pulses needed to estimate the spectrum was investigated in [30] and [59]. The smaller the amount of pulses used, leads to a reduction in resolution of the spectrum. This means that two pulse trains with similar PRFs will incorrectly be detected as one pulse train [30]. Noisy data processing results are improved by increasing N, from increasing number of pulses [30, 59]. The effects of one percent missing pulses were investigated and shown to be inconsequential at $N = 4096$. The typical values of N used in [30], [31], [33] and [59] range from 1024 to 4096 or 1025 to 4097 pulses. [30] even uses $N = 256$ as an example of too low N. According to the requirements in section 1.2.2, a constant PRI emitter needs to be identified after four pulses. This means the interleaved pulse train spectrum estimation method may not be suitable for the system.

2.6.7.1 Remarks

The authors of [30] and [33] are the only authors to have implemented this method of interleaved pulse train spectrum estimation. They have also published the same set of results in both of their publications. While trying to simulate this algorithm in MATLAB to investigate if it could be suitable for the system, the results in papers [30] and [33] could not be replicated. Bildøy in [29] could not replicate the results as well. Attempts made to contact these authors proved unsuccessful.

The first step, where an interleaved pulse train needs to be normalised to 2π and then wrapped around the unit circle, works as expected. After the DFT of the normalised and wrapped interleaved pulse train, the resultant spectrum is nothing like what the authors achieved. The spectrum achieved is random, with no harmonics and peaks that correspond to any signal PRF. It is likely that a step is missing, where the resultant of the DFT is applied to some function to achieve the correct spectrum.

The interleaved pulse train spectrum estimation method could therefore not be simulated in section 3.2. Regardless, the method was likely not suitable for the system, as it needed much more pulses than the pulse requirement specified in section 1.2.2.

2.7 Tracking

After the system, from Chapter 1, identifies an emitter by the process of deinterleaving, it will then have to track that emitter.

2.7.1 Target Tracking

When a radar system initially detects a target, it has to continue detect and associate relevant detections with the target as it moves [54]. The radar also has to use the measured parameters of targets to make predictions of future values of the measured parameters [54]. It should also make more accurate predictions of the parameters over time [54]. These are all the functions of a tracker [54]. Tracking radar systems usually measure and track a target's azimuth direction, range, radial velocity and/or elevation [63]. Tracking has many applications and can be used for fire control and missile guidance systems [24].

Target tracking can be broken down into measurement-to-track data association and track filtering [37]. Track filtering is the process of making a prediction of the next state (azimuth direction, range, radial velocity and elevation parameters) of the target from its measured current state. The estimate always has an uncertainty associated with it [37]. Measurement-to-track data association, or just data association, is the process of assigning a measurement to a currently running tracker or starting a new tracker [37]. If a measurement falls within the validation region (or gate) of a predicted state of a tracker, it is then assigned to the tracker and the current state of the tracker is updated. In the case two or more measurements lie within the validation region of the tracker, the measurement closest to the prediction is used to update the tracker. Should a measurement lie outside the validation region of a running tracker, the measurement can be considered a false measurement or used to initialize a new tentative tracker [37]. The validation region is sometimes called the tracking window.

Track-while-scan (TWS) radar systems measure the state of targets only once per scan. One scan is one revolution as these radars continually rotate their antenna beam 360º [53]. It then uses smoothing or prediction filters to estimate the state of the target between scans [63]. If the state of a target is not measured during a scan, then the track is "coasted", meaning that the prediction for the next state is made without the measured state [54], with a larger tracking window [3]. If the target measurement is missing for a number of consecutive scans, then the track is terminated [54].

As part of the research objectives and proposed system, it is stated that radar emitters need to be tracked. Here the targets are radar emitters in the EW environment, not a moving object with parameters such as range pulse, radial velocity, etc. The TOA of pulses associated with target emitters needs to be tracked. Therefore the time of a pulse that reaches the EW receiver needs to be mathematically modelled based on all factors that will affect it.

2.7.2 Modelling The TOA From An Emitter

After a pulse train is deinterleaved and emitters are found, each emitter will be tracked by the system. Tracking of an emitter involves predicting the TOA of its the next pulse as the system only deals with TOA information. Figure 2.14, shows a pulse train from one emitter.

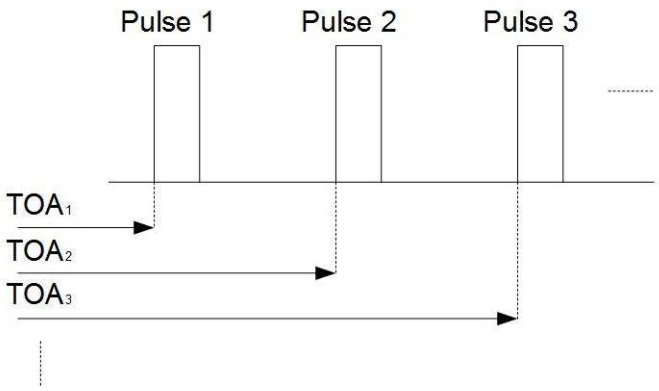


Figure 2.14: TOA Values

Variations in PRI occur in any radar system, even in constant PRI schemes accidental variations do occur [2]. The noise in TOA due to the transmitting radar system and propagation medium (basically all types noise external to the EW receiver) can be modelled as white Gaussian noise (w_j) having a zero mean[16] and variance σ_w^2 [3]. Therefore the distribution of w_j can be denoted as $N(0,\sigma_w^2)$. The time that the pulse arrives at the EW receiver is expressed below.

$$TOA_1 = TOA_0 + PRI_0 + w_1$$

$$TOA_2 = TOA_0 + PRI_0 + PRI_1 + w_1 + w_2 = TOA_1 + PRI_1 + w_2$$

$$TOA_3 = TOA_0 + PRI_0 + PRI_1 + PRI_2 + w_1 + w_2 + w_3 = TOA_2 + PRI_2 + w_3$$

$$\vdots$$

$$TOA_{j+1} = TOA_j + PRI_j + w_{j+1} \tag{2.17}$$

PRI_j in equation 2.17 represents the j^{th} PRI in the PRI sequence [3]. In PRI schemes such as staggered and dwell and switch, there is a finite number of PRIs (N) before the PRI sequence starts repeating itself. If the PRI sequence is $\begin{bmatrix} PRI_1 & PRI_2 & \dots & PRI_N \end{bmatrix}$, then the remainder from $j \bmod N$ will give the i^{th} PRI in the sequence. If $i = 0$ because j is a multiple of N then $i = N$. PRI_i will be used in the place of PRI_j in equation 2.17 for emitters having staggered or dwell and switch PRI schemes.

For a constant or jittered PRI scheme, PRI_j will remain constant for any j as there is only one PRI in the scheme. In the case of a jittered PRI scheme, the jitter can be thought of as noise intentionally added to the TOA of pulses

[16] White Gaussian noise has a mean of zero by definition.

[3]. This noise is regarded white Gaussian noise source (β_j) with distribution $N(0,\sigma_j^2)$ independent to w_j [3]. Equation 2.18 shows how equation 2.17 will be modified for an emitter with a jittered PRI scheme.

$$TOA_{j+1} = TOA_j + PRI_j + w_{j+1} + \beta_{j+1} \tag{2.18}$$

When the TOA is measured in the EW receiver, some noise called measurement noise is also added to the TOA [3]. The measurement noise (v_j) can also be modelled as white Gaussian noise with distribution distribution $N(0,\sigma_j^2)$. The measured TOA, using equation 2.17, can be expressed as

$$measuredTOA_j = TOA_j + v_j = TOA_j + PRI_j + w_{j+1} + v_j \tag{2.19}$$

2.7.3 Pulse Train Period Estimation

For the purposes of this research, the period of a PRI sequence before it starts repeating itself is needed in tracking emitters with staggered or dwell and switch PRI schemes. If a PRI sequence is $\begin{bmatrix} PRI_1 & PRI_2 & \dots & PRI_N \end{bmatrix}$, then by determining the value of N (period) then the tracking algorithm can choose the correct PRI from the PRI sequence to make the next TOA prediction. Below are some algorithms that where investigated to determine the period of PRI sequence.

2.7.3.1 Forward Search Procedure

The forward search procedure is used to find the period (N) of the PRI sequence in a dataset of PRIs [3, 8]. PRI values are easily calculated from TOA values. A histogram is firstly created with the same number of bins as the length of the dataset. The histogram is populated by taking a reference PRI and searching for subsequent similar PRIs in the dataset [3]. A PRI value is similar if it falls within a gate of the reference PRI, see equation 4.2 for the gate calculation. After the search has reached the end of the dataset, the next PRI is used as a reference. The process continues until all PRIs in the dataset were used as a reference [3]. Every time a similar PRI value is found then the bin number corresponding to the distance away from the reference PRI is increased by one. For example, if the reference PRI is at position k and two similar PRIs were found at k+5 and $k+10$, then bins 5 and 10 will be increased by one in the histogram. The bin with the highest frequency is the period of the PRI sequence. In the case of no missing or spurious pulses, the second highest bin will be double the period [3]. By increasing the length of the dataset, the performance of the algorithm increases even with interference pulses (missing and spurious) [8].

2.7.3.2 Autocorrelation

The period of a discrete signal can be found by first subtracting the mean of the signal from every element of the signal and then autocorrelating it [75]. The first step determine the period (N) of the PRI sequence in a dataset of n measured PRIs will be

$$x = measuredPRI_n - mean\left(measuredPRI_n\right) \tag{2.20}$$

The autocorrelation of a periodic sequence is the correlation of the signal with a delayed copy of itself. The autocorrelation (r_{xx}) of the sequence $x(n)$, having n elements is defined as[74]

$$r_{xx}(k) = \sum_{m=1}^{n} x(m)\,x(m+k)^{*} \tag{2.21}$$

The $*$ symbol denotes the complex conjugate of a complex number. r_{xx} is a decaying periodic wave that has the same period as the measured PRI dataset. The value of k that corresponds to the peak of r_{xx} is the period (N) of the PRI sequence as that equates to one wavelength.

2.7.3.3 Spectrum Estimation

The algorithm investigated in section 2.6.7, can be used to determine the period of a PRI sequence as well. The pulse TOAs from the emitter will be normalised to 2π and wrapped will be around the unit circle using equation 2.14 [30]. The DFT of the normalised and wrapped signal will produce the spectrum of the single pulse train. When a single pulse train is used with the algorithm, then the number of signals that it finds can be regarded as the period of the PRI sequence. A constant PRI scheme pulse train will not have any peaks, only the artefact at 0 Hz [30]. However, the algorithm results could not be replicated in simulations.

2.7.3.4 Maximum Likelihood Period Estimation

The log-likelihood function for a vector z of n observations, or in this case measured PRI, is [72]

$$L(T,\theta,s;z) = -\left\|z - Ts - \theta 1_n\right\|^2 \tag{2.22}$$

$\|.\|^2$ represents the Euclidean norm. T is the period and s is the vector of PRI indices, $s = \begin{bmatrix} 1 & 2 & \dots & n \end{bmatrix}$. θ is the offset in time from the origin and 1_n represents a vector that has n elements with the value 1. The log-likelihood function can maximized for θ by differentiating respect to θ and substituting the estimate θ for back into log-likelihood [72]. This results in the log-likelihood function equation below [72]

$$L(T,s;z) = -\left\|Q \times (z - sT)\right\|^2 \tag{2.23}$$

Q is the projection matrix defined [73], with I_n being a $n \times n$ identity matrix.

$$Q = \frac{I_n - 1_n 1_n^T}{n} \tag{2.24}$$

If we define $\zeta = Qz$, $x = Qs$ and $f = \frac{1}{T}$ then equation 2.23 can be rewritten as

$$L(f,x;\zeta) = -\frac{1}{f^2} \times \left\|f\zeta - x\right\|^2 \tag{2.25}$$

If s is held constant (by setting the s^{th} element to s and setting all other elements to 0), then the log-likelihood of T is

$$T(s) = \frac{\zeta^T x}{x^T x} \tag{2.26}$$

The value of s that corresponds to the maximum of $T(s)$ in equation 2.26 is the period (N) of the PRI sequence.

2.7.3.5 Conclusion

After the TOA based deinterleaving algorithms had been investigated earlier in this Chapter, it was found that they only detect constant PRI pulse trains. For this reason, none of the period estimation techniques needed to be implemented in the system. Pulse trains having the same PRI can be merged during track management to form staggered PRI schemes. Therefore when the tracking algorithms were investigated in the next section, staggered PRI schemes and by extension dwell & switch PRI schemes were still considered.

2.8 TOA Based Tracking Algorithms

As part of the research objectives, it is stated that algorithms that track radar emitters using only TOA information needed to be researched. When investigating the TOA based tracking algorithms, if the algorithm was suitable for constant PRI and staggered PRI schemes only considered. This is because jittered and dwell and switch PRI schemes can be simplified to constant and staggered PRI schemes, described in detail in section 2.6.

2.8.1 Delta-τ Histogram

The Delta-τ histogram is the TOA difference histogramming algorithm explained earlier in this Chapter, meaning it can also be used for deinterleaving [2]. Each TOA is subtracted from every subsequent TOA, and the frequency of these differences are recorded in the bins in a histogram [2]. This means the Delta-τ histogram contains all the differences from the first to the last specified difference level[17]. The total amount of computations required to process N pulses using the the Delta-τ Histogram algorithm is in the order of $\frac{N(N-1)}{2}$, see equation 2.9 [2].

This algorithm is not very susceptible to noise and interference pulses [2]. This is the case because the PRI of a signal and its multiples will accumulate in peaks in the histogram [2]. Table 2.2, adapted from Wiley [2], shows the expected peak distributions for different PRI type signals.

PRI Signal Type	Distribution
Constant	A Spike
Jittered	Bell centered about a PRI
Dwell & Switch	Spikes
Staggered	Spikes

Table 2.2: Typical Histogram Distribution Shapes

[17] Difference levels were explained in section 2.6.4.

In histograms, the mean and standard deviations of data are not affected by the order in which the data is collected. It is for this reason the Delta-τ histogram is not very useful in analysing the order of PRI sequences [2]. Histograms can be used to determine the overall statistics of the PRI sequence, like the PRIs present in the dataset (level number of a staggered sequence). In the case of no missing or spurious pulses, if one PRI value is repeated x number of times in the sequence, then that PRI value should be $x-1$ multiple times more frequent than PRI values that appear only once in the sequence. If there are missing or spurious pulses finding the period of the PRI sequence becomes difficult. This is because the rule about a repeated PRI being $x-1$ multiple times more frequent than other PRIs no longer applies. The intervals on the histogram need to be carefully chosen by the analyst so that the histogram can make sense [2].

With every new pulse received, a new histogram is created, which affects the PRI found in the histogram that is used to track the signal. Prediction of the TOA associated with the next pulse is made by adding the found PRI to the TOA of the pulse that was just received. If the PRI scheme of the emitter being tracked is a staggered one, then the PRI that will be added to the TOA of the most recently received pulse will be the next expected PRI. The next expected PRI is the PRI that follows the PRI that was just measured in PRI sequence of the staggered PRI scheme. The PRI sequence is determined before track is initialised.

2.8.2 Kalman Filter

The Kalman filter is an algorithm or digital filter that is used to estimate or predict the state of a system, based on its noisy outputs [3]. Essentially filtering out noise, either in the system itself or in the sensors used to measure or observe the system [7]. Over a longer period of time or after many iterations the Kalman filter becomes more accurate as sensors with the most noise are given less weighting in the final measurement [7]. During each iteration, the error in each prediction is also corrected. The Kalman filter has become one of the greatest discoveries in the history of statistical estimation theory [3]. There are numerous applications for Kalman filters, which includes signal processing, econometrics as well as guidance, navigation and control of vehicles and aircraft. In Radar applications, Kalman filters are used in the tracking component of the radar system [53]. The filter is used to predict the next state of the target after its current state is observed. This prediction is used to estimate whether the next measurement is relatively correct.

The Kalman filter assumes a linear system with multiple inputs and outputs. However, it is also widely used in estimating nonlinear systems [7]. The Kalman filter models the true state of the system (X_k) at time k as :

$$X_k = FX_{k-1} + BU_k + w_k \tag{2.27}$$

The Kalman filter assumes that the current state of the system (X_k) can be determined from the previous state of the system (X_{k-1}) at time $k-1$ and the input of system (U_k) at time k [53]. U_k is the vector of inputs referred to as the control vector. How the previous state of the system and the current input to the system affects the current state of the system, in equation 2.27, are the state transition model vector (F) and the control-input model vector (B). The state transition model, which is applied to the previous system state, describes how the system changes from one state to the next without taking into consideration the input of the system. The control-input model, which is applied to the control vector, predicts how the system changes in the state dependent on

its input. Finally w_k is the system noise vector, it is assumed to be drawn from a zero mean multivariate normal distribution with covariance Q_k [50].

Equation 2.27 is referred to as the system equation. This is because it is the equation associated with the actual state of the system. The observation equation or measurement equation (equation 2.28) is also shown below. It is used to see observe system outputs using its state.

$$Y_k = HX_k + v_k \tag{2.28}$$

Y_k is the measurement vector, this is the actual measurement of the system [50]. The measurement of the system is determined by its current state and its measurement matrix, H. The measurement matrix describes how the system state affects measurements. The noise associated with determining the system measurement from the current state is represented with v_k. v_k is the measurement White Gaussian noise vector, having zero cross-correlation with w_k from equation 2.27.

Equations 2.27 and 2.28 are used to describe "high level" operation of the Kalman filter [79]. However, in practice the Kalman filter estimates the system state using the feedback in the form of the measurements of the system [79]. The filter can be divided into two stages, namely in order of execution, prediction (time update) and correction (measurement update) [7].

The first stage of the filter is prediction stage. In this stage, the state and error covariance of the system after the next measurement of the output/s are made. The first thing to do is to apply equation 2.29 to predict the state of the system after the measurement $\left(\hat{X_k}\right)$ [7].

$$\hat{X_k} = FX_{k-1} + BU_k \tag{2.29}$$

If this is the first iteration of the filter, some values may be assumed, such as the previous state of the system (X_{k-1}), as they will be updated for future iterations [7]. The second step of the prediction stage of the filter is to predict the error covariance after the next measurement of the output is made. This is done by using equation 2.30.

$$\hat{P_k} = FP_{k-1}F^T + Q \tag{2.30}$$

$\hat{P_k}$ is the predicted error covariance of the system [79] and P_{k-1} is the previous actual error covariance of the system at time $k-1$. Q is the covariance of the error noise [79], which describes the distribution of noise.

After measurement of the output/s of the system, the second stage of the filter can begin. The goal of this stage of the filter is to make corrections to the Kalman filter model depending on actual measured values and those predicted in the previous steps. First, the Kalman gain (K) needs to be calculated using equation 2.31. The Kalman gain is generally associated with how reliable the measured result is [7].

$$K = \hat{P_k}H^T \left[H^T \hat{P_k} H^T + R_k\right]^{-1} \tag{2.31}$$

R_k is the covariance of the observation noise [79], it is used to describes the noise present in the measurement. Usually, this is the noise associated with the sensor or transducer. The next step is to update the state of the

system with the measurements (Z_k) and predicted state ($\hat{X}_k$) from the prediction stage. This is achieved using equation 2.32, shown below.

$$X_k = \hat{X}_k + K_k \left[Z_k - H\hat{X}_k \right] \tag{2.32}$$

The final step of the update stage of the filter is to update the error covariance (P_k) for this iteration. Equation 2.33 accomplishes this. I is the identity matrix and $\hat{P}_k$is the predicted error covariance from the predicted stage.

$$P_k = [I - K_k H]\,\hat{P}_k \tag{2.33}$$

Two different implementations of the Kalman filter as a radar pulse TOA tracker were researched and analysed. The two different implementations of Kalman filter set up the system state matrix (X_k) differently. The first implementation will be called the time domain Kalman filter as the system state matrix included the TOA and PRI of the emitter. The second implementation will be called the Fourier domain Kalman filter as the system state matrix includes the average PRI and Fourier coefficients of the PRI sequence.

2.8.2.1 Time Domain Kalman Filter

The system equation (2.27) and observation equation (2.28) of the Kalman filter can be rewritten as 2.34 and 2.35, respectively. These new equations model the Kalman filter as an emitter TOA tracker.

$$X_{k+1} = FX_k + u_k \tag{2.34}$$

$$Y_k = HX_k + v_k \tag{2.35}$$

Equation 2.19 models all the components of measured TOA i.e., the measured TOA is a summation of the previous TOA, the PRI, the measurement noise and the noise due to the transmitting system and the medium. Applying equation 2.19 to the Kalman filter above, the system state vector can be modelled as $X_k = \begin{bmatrix} TOA_k \\ PRI_i \end{bmatrix}$ [77]. The PRI_i used in the system state matrix is the i^{th} PRI in the PRI sequence such that in a sequence of N PRI, $i = j \bmod N$. If $i = 0$ because j is a multiple of N then $i = N$. Cycling through the PRI like this accounts for staggered PRI schemes. Since the next state of the system is defined by the next TOA (equation 2.19) and the PRI, the state transition matrix is modelled as $F = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ [77]. The system noise vector accounts for unintentional noise by the transmitting system and medium ($w_k \sim N(0, \sigma_k^2)$) and the intentional noise added for jitter (β_k~$N(0, \sigma_\beta^2)$.). β_k is set to zero for PRI schemes with no intentional jitter. Therefore the system noise vector is defined as $u_k = \begin{bmatrix} w_k \\ \beta_k \end{bmatrix}$.

Y_k is the measured TOA value. Therefore, to obtain the TOA value from the system state matrix, the measurement vector is $H = \begin{bmatrix} 1 & 0 \end{bmatrix}$ [77]. v_k is the measurement noise of the EW receiver, with distribution $N(0, \sigma_v^2)$.

The three matrices not yet defined that are needed to run the Kalman filter, as they are not present in the system and observation equations, are P_k, Q and R_k. The covariance matrix of u_k is denoted by Q and is given by $Q = \begin{bmatrix} \sigma_w^2 + \sigma_\beta^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix}$. R_k is the covariance of the observation noise and is given by $R_k = \sigma_w^2 + 2\sigma_v^2$ [3]. The initial elements of the error covariance (P_k) are set to zero. This implementation of the Kalman filter can now be run as all matrices are defined. The second implementation of the Kalman filter is explained below.

2.8.2.2 Fourier Domain Kalman Filter

The second implementation of the Kalman filter as an emitter TOA tracker uses the Fourier coefficients and mean of the PRI sequence to determine the system state vector [8]. As a result, this implementation tracks the measured PRI, not the measured TOA. The system equation and observation equation of this Kalman filter implementation are the same as the time domain Kalman filter implementation. This implementation offers much better tracking performance than the time domain Kalman filter.

A PRI sequence having N PRIs can be thought as a periodic discrete time series, in which the PRIs are plotted against equally spaced pulse indices [8]. The Fourier theorem states that any periodic function can be expressed as a constant added to the summation of cosine and sine terms [8]. Therefore the PRI sequence can be expressed as [8]

$$PRI(n) = \overline{PRI} + a_1 \cos\left(2\pi \frac{1}{N} n\right) + b_1 \sin\left(2\pi \frac{1}{N} n\right) + \ldots + a_m \cos\left(2\pi \frac{m}{N} n\right) + b_m \sin\left(2\pi \frac{m}{N} n\right) \tag{2.36}$$

$\overline{PRI}$ is the mean of all the PRIs in the PRI sequence. The Fourier coefficients are a and b. m is a function of the period (N) of the PRI sequence. If the period is even, then $m = \frac{N}{2}$, else if the period is odd, then $m = \frac{N-1}{2}$.

The Fourier coefficients a_m and b_m may be calculated as follows [3]

$$a_m = \frac{2}{N} \sum_{n=0}^{N-1} PRI_n \cos\left(\frac{2\pi mn}{N}\right) \tag{2.37}$$

$$b_m = \frac{2}{N} \sum_{n=0}^{N-1} PRI_n \sin\left(\frac{2\pi mn}{N}\right) \tag{2.38}$$

After the Fourier coefficients and the mean of the PRI sequence are calculated, it can be expressed in the form of equation 2.36. The system state vector (X_k) in this implementation of the Kalman filter is expressed using the mean and Fourier coefficients as seen in equation 2.39 [8].

$$X_k = \begin{bmatrix} \overline{PRI} \\ a_1 \\ b_1 \\ \vdots \\ a_m \\ b_m \end{bmatrix} \tag{2.39}$$

As the state of the system should remain constant from iteration to iteration, the state transition matrix (F_k) is, therefore, $F_k = 1$ [3].

Y_k is the measured PRI value [3], this can be the time between the most received pulse and the pulse received before it. The measurement vector (H_n) is used to obtain the PRI from the current state vector and is defined as $H_k = \begin{bmatrix} 1 & \cos\left(2\pi\frac{1}{N}k\right) & \sin\left(2\pi\frac{1}{N}k\right) & \dots & \cos\left(2\pi\frac{m}{N}k\right) & \sin\left(2\pi\frac{m}{N}k\right) \end{bmatrix}$ [3]. The measurement vector has to be calculated with each iteration of the filter. The covariance matrix of u_k, Q is set to zero. The covariance of the observation noise and is given by $R = \sigma_w^2 + 2\sigma_v^2$ [3].

The initial error covariance (P_0) is defined in equation 2.40 [3].

$$P_0 = \begin{bmatrix} R & -\sigma_v^2 & 0 & 0 & 0 \\ -\sigma_v^2 & R & -\sigma_v^2 & 0 & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & 0 & -\sigma_v^2 & R & -\sigma_v^2 \\ 0 & 0 & 0 & -\sigma_v^2 & R \end{bmatrix}_{N\times N} \tag{2.40}$$

This implementation of the Kalman filter requires much more computations than the time domain Kalman filter as the measured PRI, and the measurement vector needs to be calculated each iteration in addition to the filter itself. The alpha-beta filter was also investigated as it requires fewer computations than both Kalman filter methods.

2.8.3 Alpha-Beta Filter

The alpha-beta filter can be derived as a steady state two-dimensional Kalman filter [50]. The computational complexity of the alpha-beta filter is far less than that of the Kalman filter [49]. Unfortunately, the reduced complexity comes at the cost of performance compared to the Kalman filter [80], but this does not mean the filter performs poorly [49].

The Kalman filter changes its coefficients to optimise its performance each iteration [49]. By using fixed coefficients, the alpha-beta filter can avoid this unnecessary computational overhead every iteration [49]. The alpha and beta coefficients need to be determined before the filter can be started. The alpha (α) and beta (β) coefficients, from where this filter gets its name [50], are determined using equations 2.41 to 2.44 below [52]. Sometimes the coefficients are obtained from a lookup table instead of being calculated to lower computational complexity [80].

$$\lambda = \frac{\sigma_w}{\sigma_v} \tag{2.41}$$

$$r = \frac{4 + \lambda - \sqrt{8\lambda + \lambda^2}}{4} \tag{2.42}$$

$$\alpha = 1 - r^2 \tag{2.43}$$

$$\beta = 2(2-\alpha) - 4\sqrt{1-\alpha} \tag{2.44}$$

As with the Kalman filter, the alpha-beta filter can also be divided into prediction and correction stages. The prediction stage equations can set up as follows for tracking of pulse TOAs from an emitter.

$$\hat{TOA}_k = \hat{TOA}_{k-1} + \hat{PRI}_k \tag{2.45}$$

$$\hat{PRI}_k = PRI_i \tag{2.46}$$

$\hat{TOA}_k$ and $\hat{PRI}_k$ are the current predictions of the next TOA to be measured and PRI_k respectively. In equation 2.46, the PRI_i assigned to the PRI estimate is the next expected PRI in the PRI sequence, such that i cycles from 1 to N. Where N is the number of PRIs in the PRI sequence. This was originally explained when deriving equation 2.17.

After the actual TOA received for this iteration of the filter is measured (TOA_k), the correction stage of the alpha-beta filter consists of the three following equations:

$$r_k = TOA_k - \hat{TOA}_k \tag{2.47}$$

$$\hat{TOA}_k = \hat{TOA}_k + \alpha r_k \tag{2.48}$$

$$PRI_i = \hat{PRI}_i + \beta r_k \tag{2.49}$$

r_k is called the residual. It is the difference between the measured TOA and the predicted TOA.

As the alpha-beta-gamma filter is an extension of the alpha-beta filter [81], its suitability to track pulse TOAs from an emitter was also investigated

2.8.4 Alpha-Beta-Gamma Filter

The alpha-beta-gamma filter is used when the second state variable increases or decreases over time [81]. The velocity of an object is the second state variable when modelling the change in position of an object [81]. Therefore the filter is used in the case of a change in velocity (acceleration) of the object, which either increases or decreases over time [81].The second state variable of modelling the pulse TOAs of an emitter is the PRI. An emitter with varying PRI could possibly be tracked with this filter.

Like with the alpha-beta filter, the coefficients need to be determined before the filter can be started. The difference here is that there are three coefficients to be determined namely, alpha (α), beta (β) and gamma (γ). Equations 2.50 to 2.61 show the calculations needed to determine the all the coefficients [82].

$$\lambda = \frac{\sigma_w}{\sigma_v} \tag{2.50}$$

$$b = \frac{\lambda}{2} - 3 \tag{2.51}$$

$$c = \frac{\lambda}{2} + 3 \tag{2.52}$$

$$d = -1 \tag{2.53}$$

$$p = c - \frac{b^2}{3} \tag{2.54}$$

$$q = \frac{2b^3}{27} - \frac{bc}{3} + d \tag{2.55}$$

$$v = \sqrt{q^2 + \frac{4p^3}{27}} \tag{2.56}$$

$$z = -\sqrt[3]{q + \frac{v}{2}} \tag{2.57}$$

$$s = z - \frac{p}{3z} - \frac{b}{3} \tag{2.58}$$

$$\alpha = 1 - s^2 \tag{2.59}$$

$$\beta = 2\left(1 - s\right)^2 \tag{2.60}$$

$$\gamma = \frac{\beta^2}{2\alpha} \tag{2.61}$$

The equations for the prediction stage of the alpha-beta-gamma filter modelling the emitter TOA tracker are [82]:

$$\hat{TOA}_k = \hat{TOA}_{k-1} + \hat{PRI}_k + \frac{\triangle \hat{PRI}_i}{2} \tag{2.62}$$

$$\hat{PRI}_k = PRI_i + \triangle \hat{PRI}_i \tag{2.63}$$

$\hat{TOA}_k$ and $\hat{PRI}_k$ are the current predictions of the next TOA to be measured and PRI_k respectively. PRI_i refers to the i^{th} PRI in the PRI sequence, where i cycles between 1 and the period of the PRI sequence, originally explained when deriving equation 2.17.

After the actual TOA received for this iteration of the filter is measured (TOA_k), the correction stage of the alpha-beta-gamma filter begins. Like the alpha-beta filter, the residual (r_k) is calculated first. The correction stage equations are [82]:

$$r_k = TOA_k - \hat{TOA}_k \tag{2.64}$$

$$\hat{TOA}_k = \hat{TOA}_k + \alpha r_k \tag{2.65}$$

$$PRI_i = \hat{PRI}_i + \beta r_k \tag{2.66}$$

$$\triangle\hat{PRI}_i = \triangle\hat{PRI}_i + \frac{\gamma}{2} r_k \tag{2.67}$$

2.8.4.1 Remarks

Unfortunately, when the filter was implemented in MATLAB, it was found to be unable to correctly track the TOA of a constant PRI scheme emitter. The results are shown in Figure 2.15.

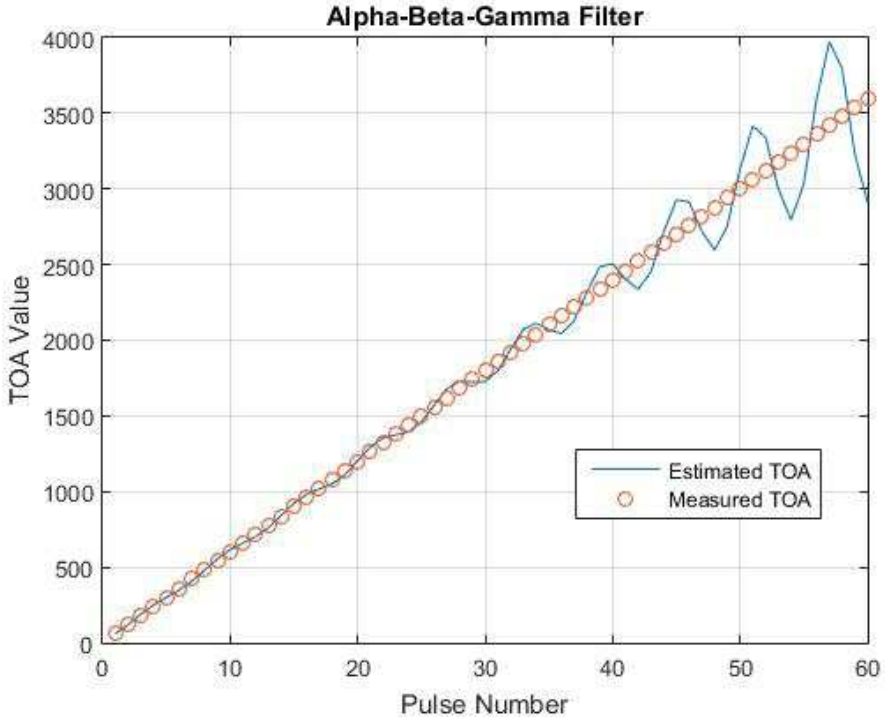


Figure 2.15: α-β-γ Filter Tracking A Constant PRI Scheme Emitter

In the figure above, it can be seen that after tracking about thirty pulses of the constant PRI scheme emitter, the filter becomes unstable. This is because the filter is designed for cases when the second state variable increases or decreases over time. In the case of an emitter tracker, the second state variable will be the PRI. The PRI of a constant PRI scheme can vary due to noise, but noise has a zero mean, meaning the PRI will not increase or decrease over time. Therefore the filter cannot be used to track constant PRI schemes and was not implemented in simulations. The alpha-beta-gamma filter will be well suited for tracking sliding or periodic PRI schemes.

Chapter 3

Emitter Deinterleaving

The aim of this Chapter is to determine the most suitable TOA based deinterleaving algorithm to be used in the system discussed in Chapter 1. To accomplish this, all the relevant deinterleaving algorithms researched in the literature review (section 2.6), were first simulated in MATLAB. After comparing the obtained simulation results, the two best-performing algorithms were chosen to be implemented on the hardware platform (the DRFM). After the hardware results of the selected algorithms had been obtained, comparisons with the simulation results were drawn. A choice as to which algorithm was most suitable for the system was made using the hardware results.

Before the algorithms could be tested in simulations, the EW environment in which the algorithms (for both deinterleaving and tracking) had to be conceptualised.

3.1 Simulated EW Environment

All deinterleaving and tracking algorithms that were implemented in MATLAB, and later the hardware platform, were exposed to a simulated EW environment. The algorithms were sequentially given TOA values of pulses from the simulated EW environments. The characteristics of the EW environments were varied to determine the impact those characteristics have on the performance of the different algorithms.

In dense EW environments, an EW receiver cannot intercept all pulses called missing pulses [47]. The EW receiver can also receive pulses that are not of interest, called spurious pulses. The EW environments that were simulated varied the percentage of random pulses associated with each emitter not detected by the EW receiver (missing pulses) from zero to ten percent. Missing pulses due to rotating of an antenna beam [78] is not considered here. Therefore, missing pulses are not bunched together. The amount of random spurious pulses in the EW environments were also varied from zero to ten percent of the total number of transmitted pulses in the EW environment.

When testing the deinterleaving performance of algorithms, the number of emitters in the environment was varied between two and four. Each emitter transmitted 100 pulses. When evaluating TOA tracking algorithms, only the emitter being tracked was in the EW environment. The emitter transmitted 446 pulses in this case, so plots of up to 400 pulses could show tracking error versus pulse number even if there were 10% missing pulses

$(446-(0.1\times 446)-1)$. The range of the PRI used by the emitters in the EW environments was from 2 μs to 15 ms, as discovered from the industry study in section 1.2.1.2. There is no pulse at $t = 0s$ but the dataset for each emitter includes the zeroth pulse (TOA_0), such that $TOA_0 = 0s$. The dataset for interleaved pulse trains only includes one zeroth pulse.

As discussed in section 2.7.2, each measured TOA has noise associated with it. The effect of noise in time measurement on the performance of the algorithms was also investigated as per the requirements, in section 1.2.2. To generate TOA values with a specific time measurement to noise ratio (TMNR), the variance of noise added by the EW receiver (σ_v^2) and transmitting radar system (σ_w^2) had to be calculated to achieve the required signal to noise ratio (SNR). The SNR of a discrete signal is equal the ratio of the mean of the signal (μ) to the standard deviation of the signal (σ) [62].

$$SNR = 20\log_{10}\left(\frac{\mu}{\sigma}\right) \tag{3.1}$$

$$\frac{SNR}{20} = \log_{10}(\mu) - \log_{10}(\sigma) \tag{3.2}$$

The standard deviation (σ) represents all the noise and other interference in the signal [62]. If the PRI sequence is the discrete signal then, from equation 4.1, the variance of noise (σ^2) associated the PRI measurement is $\sigma_w^2 + 2\sigma_v^2$. Since the mean of the noise is zero, the the mean PRI ($\overline{PRI}$) and the standard deviation of noise can be substituted into equation 3.2. This SNR will now be called TMNR as to avoid confusion, as we are talking about the noise in the measurement of time.

$$\frac{TMNR}{20} = \log_{10}\left(\overline{PRI}\right) - \frac{1}{2}\log_{10}\left(\sigma_w^2 + 2\sigma_v^2\right) \tag{3.3}$$

The noise added to a measurement by the receiver (σ_v) will be constant as the same receiver is used for all measurements. The typical standard deviation of time measurement noise added by the receiver (σ_v) is about 0.1 [3]. This value was used as the standard deviation of noise added to the TOA measurements by the EW receiver. Substituting σ_v back into equation 3.3.

$$\frac{TMNR}{20} \approx \log_{10}\left(\overline{PRI}\right) - \frac{1}{2}\log_{10}\left(\sigma_w^2 + 0.02\right) \tag{3.4}$$

$$\frac{TMNR}{20} - \log_{10}\left(\overline{PRI}\right) \approx -\frac{1}{2}\log_{10}\left(\sigma_w^2 + 0.02\right)$$

$$\frac{1}{2}\log_{10}\left(\overline{PRI}\right) - \frac{TMNR}{10} \approx \log_{10}\left(\sigma_w^2 + 0.02\right)$$

$$\sigma_w^2 + 0.02 \approx 10^{\frac{1}{2}\log_{10}\left(\overline{PRI}\right)} \times 10^{-\frac{TMNR}{10}}$$

$$\sigma_w^2 + 0.02 \approx \frac{\sqrt{\overline{PRI}}}{10^{\frac{TMNR}{10}}}$$

$$\sigma_w \approx \sqrt{\frac{\sqrt{\overline{PRI}}}{10^{\frac{TMNR}{10}}} - 0.02} \tag{3.5}$$

Using equation 3.5, the standard deviation of noise, σ_w, is calculated for each emitter to achieve the required TMNR. The TMNR was varied between 8 and 26 dB in the simulated EW environments.

Every deinterleaving and tracking algorithm is simulated one thousand times in different EW environments with a constant set of characteristics before the characteristics are varied. This means that there will be one thousand results (simulation and hardware) for an algorithm in one thousand different EW environments with *x* missing pulses, *y* spurious pulses and *z* dB TMNR. These values are averaged over the one thousand runs to eliminate any noise in the results. The characteristics are changed one at a time to ensure there are results for all combinations. When testing deinterleaving algorithms, the number of emitters was also varied.

3.2 Deinterleaving Simulation Results

Applicable TOA based deinterleaving algorithms investigated in the literature review, were simulated in MATLAB. Each algorithm was given simulated TOA measurements from different simulated EW environments, as explained in section 3.1. When algorithms are implemented on the FPGA, variable and array sizes can not be dynamic and need to be static as specific memory addresses are allocated to variables when the FPGA is programmed. For this reason, the number of bins used in the TOA difference and SDIF based histogramming methods was set to a constant 30 bins. In the case of the CDIF based histogramming methods, the number of bins started at 10 and could increase to a maximum of 40 bins. A first in, first out (FIFO) type ring buffer with a size of 40 elements was implemented to store TOA values as they were received. After the algorithm finishes executing with the TOA values stored in the buffer, the next TOA value from the EW environment is added to the buffer. The size of the bins in the histograms $(\triangle)$ was determined by the range $(TOA_{Newest} - TOA_{Oldest})$ of the TOA values stored in the buffer, such that $\triangle = \frac{TOA_{Newest} - TOA_{Oldest}}{bins}$. In the case of the CDIF based histogramming methods, were the number of bins could change, the size of the bins remained the same for all difference levels after it was originally calculated for 10 bins in the first difference level.

None of the authors in the literature reviewed on the algorithms simulated published their actual results, only their conclusions. Theses authors did not investigate the effect of interference pulses, time measurement noise and number of emitters have on the algorithms. They also did not make use of a FIFO buffer as explained above. Their conclusions were mentioned in section 2.6 and some conclusions can be used in the analysis of the results.

The algorithms were exposed to three types of interleaved pulse trains. The first type consisted only of constant PRI pulse trains and the second type of interleaved pulse train consisted of only jittered PRI pulse trains. The last type of pulse train consisted of a mixed combination of constant and jittered PRI pulse trains. Staggered PRI schemes, and therefore dwell and switch PRI schemes, were not simulated here as an *x* level[1] staggered pulse train will result in algorithms finding *x* constant pulse trains [10].

[1] Level is the number of unique PRIs in a stagger PRI sequence, see section 2.3.

A snapshot of the simulation results for the deinterleaving algorithms can be seen in Tables 3.1 to 3.12. The complete emitter deinterleaving simulation results can be found in Appendix C.

3.2.1 Constant PRI Signals

The first type of interleaved pulse train tested with the deinterleaving algorithms consisted of emitters having only constant PRI schemes. The PRIs for each emitter were as follows:

Emitter 1: 50 μs, Emitter 2: 75 μs, Emitter 3: 31 μs, Emitter 4: 88 μs

In the case of two emitters being interleaved, the pulses from emitters 1 and 2 are present in the interleaved pulse train. In the case three emitters are interleaved, the pulses from emitters 1 to 3 are present in the interleaved pulse train. Finally, in the case of 4 interleaved emitters, the pulses from emitters 1 to 4 are present in the interleaved pulse train. The results of each algorithm were measured using the percentage of success and mean number of pulses to deinterleave.

3.2.1.1 Percentage of Success

Success is defined as when an algorithm correctly deinterleaves all emitters in an EW environment without any false emitters being detected. Therefore, the percentage of success is the percentage number of times the algorithm correctly deinterleaved all emitters in the environment without any false emitters being detected in the one thousand runs. A higher percentage of success indicates better performance. Table 3.1 shows the percentage of success each algorithm achieves with varying emitters and TMNR (time measurement to noise ratio). There are no missing or spurious pulses. High TMNR refers to the case of a 26 dB TMNR, mid TMNR refers to the 16 dB case and low TMNR refers to the 8 dB case.

	Percentage of Success (%) – No missing or spurious pulses								
Emitters	**2**			**3**			**4**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	19	52.3	86.6	10.3	39.7	35.5	9.5	21	0
CDIF SS	13.7	47.7	72.1	6.5	34.6	27.8	5.4	16.6	0
TOA Difference Histogram	3	48	100	0.3	14.9	82.8	0	0.3	5.5
SDIF	0.4	14.9	38	1.1	14.4	36.3	0.7	2	0
SDIF SS	0.3	16.6	37.6	0.4	11.7	31	0.6	1.6	0
Sequence Search	0.9	29.9	100	0.6	22.6	100	1.4	17.3	100

Table 3.1: Percentage of Success - Constant PRI - Emitters

From a quick glance at the table, the general trend is that as the number of emitters increase, the performance decreases.

The sequence search algorithm is the only algorithm that does not completely fail to correctly deinterleave and is least affected by number of emitters. Also observed with the SS algorithm is that when TMNR decreases, so does the performance. The SS algorithm is the most negatively affected by lower TMNR compared to other algorithms. Matching in the SS algorithm is based on a prediction and a tolerance, if there is more noise in

TOA measurements matching becomes harder. When more emitters are present, the rate at which performance decays with TMNR is greater. This is due to there being more emitters resulting in more pulses to falsely be matched with.

The CDIF algorithm performs better than all other algorithms in all but three scenarios. In the first scenario, with two emitters and TMNR of 26 dB, the sequence search and TOA difference histogramming algorithms are just better because of better matching and more difference levels, respectively. In the second scenario, with three emitters and TMNR of 26 dB, the same can be said for the sequence search and TOA difference histogramming algorithms. The SDIF algorithm also outperforms the CDIF algorithm. This is expected, as the algorithm was designed to be an improvement over the CDIF in emitter dense environments. In the third scenario, with four emitters and TMNR of 26 dB, more emitters means more pulses and could result in higher counts of bins associated with smaller differences between pulses causing a false detection. Another possibility is that with more emitters, higher difference levels had to be used and when the number of bins was increased, it exceeded the maximum of 40.

The CDIF SS algorithm did not improve over the standard CDIF algorithm in any situation. The decrease in performance because when matching, correct detections are rejected as false detections. This explains the reduction of performance at lower TMNR situations as well as situations with more emitters. In the situation with two emitters and TMNR of 26 dB, the PRI estimate from the bin range might differ so much from the PRI needed to correctly make matches in the SS part, such that the pulses lie outside the matching window.

The TOA difference histogram also performs better at higher TMNR. Its performance deteriorates as the number of emitters increase. More pulses from having more emitters lead to higher counts in bins associated with smaller differences.

The SDIF algorithm was introduced as an improvement over the CDIF algorithm. However, in all but one situation the CDIF algorithm outperforms it. This is because the optimum threshold function is not being used. The threshold used in simulations is less than the optimum threshold function.

The SDIF SS algorithm also has a drop in performance when compared to the SDIF algorithm. The reasons for this are the same as those outlined when combining the CDIF and SS algorithms.

Table 3.2 shows the percentage of success each algorithm achieves with varying interference pulses (spurious and missing) and TMNR. The number of emitters is kept constant at two.

	Percentage of Success (%) – 2 Emitters											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	19	52.3	86.3	32.6	76.1	85.5	21.8	69.8	79.4	28.5	74.4	79.6
CDIF SS	13.7	47.7	72.1	27.4	48	22.5	18.2	66.6	63	23.7	55.5	38.5
TOA Difference Histogram	3	48	100	9.4	72.6	88.2	2.2	53.9	84.6	5.6	57.8	74.6
SDIF	0.4	14.9	38	2.9	54.5	77.3	2.3	40.4	78.2	3.8	51.8	79.3
SDIF SS	0.3	16.6	37.6	3.2	41.7	8.6	1.3	40.8	72.2	2.6	47.2	32.7
Sequence Search	0.9	29.9	100	2.6	31	69.7	1.4	34.8	99	2.6	36.3	65.1

Table 3.2: Percentage of Success - Constant PRI - Interference Pulses

The performance of the CDIF algorithm increases with missing pulses and spurious pulses at low to mid TMNR. At high TMNR however, the performance decreases. This is because at higher TMNR the bin sizes are smaller,

leading to counts being spread over multiple bins. Therefore, resulting in bins associated with the wrong PRI reaching the threshold count of three before a bin associated with the right PRI. Both the CDIF SS and TOA difference histogram algorithms also experience the same change in performance as the CDIF algorithm.

The SDIF algorithm experiences an increase in performance as missing and spurious pulses increase. This aligns with the theory of the histogram performing better in such environments. The performance of the SDIF SS and sequence search algorithms also increases as the number of missing and spurious pulses increase. However, when there are missing pulses at high TMNR, performance decreases. In these cases, the matching window becomes smaller, and compounded with missing pulses, the sequence search algorithm fails more often. The sequence search algorithm failing more often could also be contributing reason as to why there is a larger decrease in performance than expected with missing pulses at high TMNR for the CDIF SS algorithm over the CDIF algorithm alone.

3.2.1.2 Mean Number of Pulses To Deinterleave

The mean number of pulses is the average number of pulses processed before an algorithm successfully deinterleaved all emitters in the EW environment, i.e. the average number of pulses processed when there was a success. The lower the mean number of pulses needed to deinterleave the better the performance of the algorithm. This number includes the zeroth pulse, $TOA_0 = 0s$. Table 3.3 shows the mean number of pulses each algorithm takes to deinterleave while varying emitters and TMNR. There are no missing or spurious pulses.

	Mean Number of Pulses To Deinterleave – No missing or spurious pulses								
Emitters	**2**			**3**			**4**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	14.5	13.7	13.4	16.3	17.5	16.6	15	19.6	N/A
CDIF SS	14.6	13.8	13.5	16.4	17.6	16.8	14.7	19.8	N/A
TOA Difference Histogram	30.7	25.7	22.4	31	38.7	42	N/A	57.7	76.3
SDIF	20.8	19.5	18	21.4	25.8	21.71	21.6	28.2	N/A
SDIF SS	18.3	19.4	17.7	20.8	25.8	21.8	20.5	31.8	N/A
Sequence Search	11	6.5	5	9.5	8.7	7	11.4	10.86	9

Table 3.3: Mean Number of Pulses - Constant PRI - Emitters

A not applicable value is when the algorithm failed to deinterleave all the emitters in the EW environment correctly. As the number of emitters increases, the number of pulses in the EW environment will also increase. An increase in the mean number of pulses to deinterleave as emitters increase is expected due to the more pulse dense environment. Big jumps in mean pulses to deinterleave like that of the TOA difference algorithm can be explained as the algorithm cannot handle emitter dense environments well.

The sequence search algorithm required the least amount of pulses to correctly deinterleave all emitters in an EW environment. There are expected increases in the mean number of pulses as the emitters increase at mid and high TMNR. Low TMNR decreases the number of pulses needed to deinterleave because bigger gates are used at lower TMNR, so when correct detections are made, they are made quicker. The TOA difference histogram required the most pulses to deinterleave compared to other algorithms. The CDIF algorithm required

fewer pulses than the SDIF algorithm to deinterleave. When comparing CDIF SS and SDIF SS with CDIF and SDIF, respectively, it can be seen that the number of pulses does not change significantly. This is sensible as after the original CDIF or SDIF algorithm detects a possible PRI, it is then sent to the sequence search without proceeding to process the next pulse until the sequence search has completed.

Table 3.4 shows the mean number of pulses each algorithm takes to deinterleave while varying interference pulses (spurious and missing) and TMNR. The number of emitters is kept constant at two. In the mean number of pulses, the zeroth pulse ($TOA_0 = 0s$) and spurious pulses are counted. Missing pulses are not counted.

	Mean Number of Pulses To Deinterleave – 2 Emitters											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	14.5	13.7	13.4	12.2	10.4	10.5	12.7	11.2	11	11.8	10	9.6
CDIF SS	14.6	13.8	13.5	12.3	11.2	12.5	12.8	11.1	11.1	11.7	10.3	10
TOA Difference Histogram	30.7	25.7	22.4	25.8	16.5	13.8	27.8	19.6	15.4	24.5	16.3	12.52
SDIF	20.8	19.5	18	18.5	13.5	11.5	21.48	15.8	13.13	20.7	13.7	11.2
SDIF SS	18.3	19.4	17.7	20.3	14.1	16.7	20.7	15.5	13.2	21.5	13.5	11.9
Sequence Search	11	6.5	5	9.5	6.7	5.1	11.1	6.7	5.2	9.8	6.8	5.3

Table 3.4: Mean Number of Pulses - Constant PRI - Interference Pulses

It is expected that spurious pulses should increase the mean number of pulses to deinterleave as there are more pulses in the environment. Depending on where missing pulses are situated, they could increase or decrease the pulses required to deinterleave. An increased number of pulses could mean that the algorithm had to run longer than previously because pulses that were missing belonged to a pulse train that it identifies quickly. A decreased number of pulse numbers could mean that the missing pulses belonged one of the pulse trains that the algorithm identified towards the end.

The amount of missing and spurious pulses does not seem to affect the sequence search algorithm significantly. Low TMNR helps reduce the number of pulses needed to deinterleave as explained above, in the discussion for Table 3.3. In the case of missing pulses, the algorithm takes fewer pulses to deinterleave. CDIF also uses fewer pulses to deinterleave in the presence of missing pulses. Interestingly enough, spurious pulses help decrease the number of pulses. The spurious pulses probably add counts to bins associated with actual PRIs, resulting in the count exceeding the threshold quicker. The CDIF SS and TOA difference histogram algorithms also experience the same change in performance. The best case of these algorithms is in the presence of both missing and spurious pulses. The spurious pulses probably behave in the same way as they did in the CDIF algorithm. The SDIF and SDIF SS algorithms perform better or the same in the presence of spurious and missing pulses.

3.2.2 Jittered PRI Signals

The second type of interleaved pulse train tested with the deinterleaving algorithms consisted of emitters having only jittered PRI schemes. The PRIs for each emitter were as follows:

Emitter 1: 50 μs (10% jitter), Emitter 2: 75 μs (10% jitter), Emitter 3: 31 μs (10% jitter), Emitter 4: 88 μs (10% jitter)

In the case of two emitters being interleaved, the pulses from emitters 1 and 2 are present in the interleaved pulse train. In the case three emitters are interleaved, the pulses from emitters 1 to 3 are present in the interleaved

pulse train. Finally, in the case of 4 interleaved emitters, the pulses from emitters 1 to 4 are present in the interleaved pulse train. The results of each algorithm were measured using the percentage of success and mean number of pulses to deinterleave.

3.2.2.1 Percentage of Success

Table 3.5 shows the percentage of success each algorithm achieves with varying emitters and TMNR. There are no missing or spurious pulses.

	Percentage of Success (%) – No missing or spurious pulses								
Emitters	**2**			**3**			**4**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	16.2	22.6	19.1	11.2	14.6	11.3	9.3	13.1	10.1
CDIF SS	12	18.2	13.3	8.1	11.2	7.8	6.2	9	6.4
TOA Difference Histogram	1.7	6.9	3.2	0.5	0.8	1.2	0	0	0
SDIF	0.9	1.3	1.5	1.3	1	0.9	1.4	0.7	1.1
SDIF SS	0.7	0.6	0.7	0.5	0.6	0.4	0.6	0	0
Sequence Search	1.3	3	2.3	0.7	0.6	0.9	1.1	0.5	0.2

Table 3.5: Percentage of Success - Jittered PRI - Emitters

Jittered PRI scheme pulse trains can be thought of constant PRI scheme pulse trains with more noise in their measured time. This extra variations or noise greatly affects all the algorithms. For example, the additional noise took the sequence search algorithm from the best performing algorithm at high TMNR to having the one of the worst performance overall. The TOA difference algorithm now fails completely after 4 emitters are added to the environment, due to both the extra noise and pulses. The SDIF and SDIF SS performance have also been extremely reduced compared their performance with constant PRI pulse trains. This leads to the conclusion that SDIF susceptible to noise as well. The CDIF and CDIF SS are the only two algorithms that reached percentage of success rates in the double digits. The CDIF algorithm slightly outperformed the CDIF SS algorithm, due to the sequence search algorithm being so susceptible to noise. In the situation with 4 emitters and TMNR of 26 dB, the CDIF algorithm failed to deinterleave the constant PRI pulse trains. With jittered pulse trains, however, this is not the case. Due to the larger noise in the TOA measurement, the size of bins in the histogram are likely higher, resulting in a wider range of differences being grouped together in a bin associated with the correct PRI.

Table 3.6 shows the percentage of success each algorithm achieves with varying interference pulses (spurious and missing) and TMNR. The number of emitters is kept constant at two.

	Percentage of Success (%) – 2 Emitters											
Missing (M) & Spurious (S) Pulses	M=0% S=0%			M=10% S=0%			M=0% S=10%			M=10% S=10%		
TMNR	8 dB	16 dB	26 dB	8 dB	16 dB	26 dB	8 dB	16 dB	26 dB	8 dB	16 dB	26 dB
CDIF	16.2	22.6	19.1	31	56	57.2	20.3	48.7	49	27.6	57.5	60.3
CDIF SS	12	18.2	13.3	25.8	46.3	45.2	18.4	44.5	43.8	21	49.7	51.4
TOA Difference Histogram	1.7	6.9	3.2	7.4	28.2	26.7	2	8.9	7.6	4.9	18.9	17.7
SDIF	0.9	1.3	1.5	3	13.8	14.7	2	5.5	6.4	3.5	12.2	14.6
SDIF SS	0.7	0.6	0.7	1.5	13.8	12.9	1	5.3	5	3	10.8	11.6
Sequence Search	1.3	3	2.3	2.2	6.6	6.7	1.3	3.8	4	2.9	7.2	7.4

Table 3.6: Percentage of Success - Jittered PRI - Interference Pulses

The performance of the CDIF algorithm increases with missing and spurious pulses. With constant PRI pulse trains, performance decreased at high TMNR. However, this is no longer the case because with jitter the TMNR of a signal is effectively lower. The CDIF SS experiences the same change in performance as the CDIF algorithm.

The performance of the TOA difference histogram, SDIF, SDIF SS and sequence search algorithms all improve with missing and spurious pulses. Their performance is also much lower than when compared to their performance with constant PRI signals (shown in Table 3.2) due to the extra noise.

3.2.2.2 Mean Number of Pulses To Deinterleave

Table 3.7 shows the mean number of pulses each algorithm takes to deinterleave while varying emitters and TMNR. There are no missing or spurious pulses. Fewer pulses is a better result.

	Mean Number of Pulses To Deinterleave – No missing or spurious pulses								
Emitters	2			3			4		
TMNR	8 dB	16 dB	26 dB	8 dB	16 dB	26 dB	8 dB	16 dB	26 dB
CDIF	16.2	18.5	18.5	16	17.8	16.9	14.4	15.4	14.9
CDIF SS	15	15	14.9	16.5	18.5	18.4	15.6	17.3	16.6
TOA Difference Histogram	30.2	29.9	29	29.4	32.8	33.7	N/A	N/A	N/A
SDIF	22.3	22.2	22	20.9	24	25.2	21	21.4	21.5
SDIF SS	21.9	21	20.9	18	28.5	28.5	21.5	N/A	N/A
Sequence Search	11.2	8.6	8.7	9.7	10.3	11.2	11	12.6	11.5

Table 3.7: Mean Number of Pulses - Jittered PRI - Emitters

The pulses required to deinterleave using the CDIF algorithm remained approximately the same when the number of emitters was increased to three. When a fourth emitter was introduced, the number of pulses decreased. Logically this number should have increased as there are more pulses in the environment but never because the algorithm has a higher success percentage for this scenario. CDIF SS has the same change in performance with more emitters as the CDIF algorithm. As the TOA difference histogram algorithm fails with four emitters, there are no mean pulse data for it. In the case of three emitters, the number of pulses increases as it is expected with the additional emitter. However, at TMNR of 8 dB, the number of pulses decrease. This is because the additional noise leads to bigger bin sizes which allow more differences to be added to the same bin, resulting in a faster detection.

Table 3.8 shows the mean number of pulses each algorithm takes to deinterleave while varying interference pulses (spurious and missing) and TMNR. The number of emitters is kept constant at two.

	Mean Number of Pulses To Deinterleave – 2 Emitters											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	16.2	18.5	18.5	12	11.1	10.7	13	12.3	11.6	12.1	10.7	10.5
CDIF SS	15	15	14.9	12.2	11.3	10.8	13.2	13.2	11.6	11.8	11	10.6
TOA Difference Histogram	30.2	29.9	29	24.9	20.5	20.2	28.2	24.2	22.8	25.4	21.3	20.1
SDIF	22.3	22.2	22	20	16.3	15.7	21.6	19.1	17.8	19.7	16.5	16.6
SDIF SS	21.9	21	20.9	16.9	16.3	16.3	20.7	19.8	17.6	20.6	16.5	16.9
Sequence Search	11.2	8.6	8.7	9.8	8.7	8.2	8.6	8.6	8.3	9.6	8.1	8.4

Table 3.8: Mean Number of Pulses - Jittered PRI - Interference Pulses

The number of pulses to deinterleave remain approximately the same for the sequence search algorithm in the presence of missing and spurious pulses at high and mid TMNR. At low TMNR the number of pulses decreases, as the success of the algorithm increases in this case. The performance changes due to missing and spurious pulses seen here are exactly the same as the performance changes seen in Table 3.4.

3.2.3 Mixed PRI Signals

The third and final type of interleaved pulse train tested with the deinterleaving algorithms consisted of emitters having both constant and jittered PRI schemes. The PRIs for each emitter were as follows:

Emitter 1: 50 μs (10% jitter), Emitter 2: 75 μs, Emitter 3: 31 μs (10% jitter), Emitter 4: 88 μs

In the case of two emitters being interleaved, the pulses from emitters 1 and 2 are present in the interleaved pulse train. In the case three emitters are interleaved, the pulses from emitters 1 to 3 are present in the interleaved pulse train. Finally, in the case of 4 interleaved emitters, the pulses from emitters 1 to 4 are present in the interleaved pulse train. The results of each algorithm were measured using the percentage of success and mean number of pulses to deinterleave.

3.2.3.1 Percentage of Success

Table 3.9 shows the percentage of success each algorithm achieves with varying emitters and TMNR. There are no missing or spurious pulses.

	Percentage of Success (%) – No missing or spurious pulses								
Emitters	**2**			**3**			**4**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	17.3	48.2	76.9	10.6	31.6	51.5	9.9	19	33.4
CDIF SS	14	41.2	71.3	7.2	25.7	48.5	6	16.9	27.5
TOA Difference Histogram	2.5	19.7	37	0.2	2.4	3.3	0.1	0	0
SDIF	0.3	4.8	7.2	1.3	2.9	3.9	1	0.6	2.7
SDIF SS	0.4	4.9	5.6	0.5	2	3.8	0.5	0.3	2.7
Sequence Search	0.4	10.4	16.1	0.5	2.6	4	1	1.5	4.4

Table 3.9: Percentage of Success - Mixed PRI - Emitters

The performance of all the algorithms in the table above are better than the performance seen with interleaved pulse trains consisting of only jittered PRI schemed emitters but below the performance that was seen with interleaved pulse trains consisting of only constant PRI schemed emitters. The only exceptions to this are the SDIF and SDIF SS algorithms deinterleaving at low TMNR, where the algorithms perform worse than the interleaved jittered PRI pulse train. As this is not the case with CDIF and CDIF SS, we can conclude that by discarding difference counts from lower difference levels, thereby changing bin sizes with every new histogram, is not beneficial for deinterleaving a pulse train with pulse trains that do not have the same amount of noise[2]. The change in performance in all the algorithms keep to what was discussed when analysing Tables 3.1 and 3.5. As number of emitters increase, the performance of the algorithms generally decreases.

Table 3.10 shows the percentage of success each algorithm achieves with varying interference pulses (spurious and missing) and TMNR. The number of emitters is kept constant at two.

	Percentage of Success (%) – 2 Emitters											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	17.3	48.2	76.9	31.9	72.1	79.8	23.7	65	79.3	25.8	69	77.3
CDIF SS	14	41.2	71.3	24.1	45.2	32.1	18.6	62.9	75	22	52.3	45.7
TOA Difference Histogram	2.5	19.7	37	9.2	40.2	46.1	2.5	20.3	24.3	5.3	29	30.3
SDIF	0.3	4.8	7.2	4.1	30.5	35.4	1.7	15.8	24.1	5.3	28.2	37
SDIF SS	0.4	4.9	5.6	2.5	21.6	22.7	1.1	14.4	19.1	2.9	22.2	23.2
Sequence Search	0.4	10.4	16.1	1.7	15.4	19.1	1.7	12.3	20.8	2.9	16	20

Table 3.10: Percentage of Success - Mixed PRI - Interference Pulses

The performance of all the algorithms with mixed PRI interleaved pulse trains lie between the constant PRI interleaved pulse trains, and the jittered PRI interleaved pulse trains as discussed above. The performance of all algorithms increases with missing and spurious pulses, with two exceptions. Both exceptions involve missing pulses at high TMNR using the CDIF SS algorithm, the reasons for this reduction in performance was discussed in the analysis of Table 3.2.

3.2.3.2 Mean Number of Pulses To Deinterleave

Table 3.11 shows the mean number of pulses each algorithm takes to deinterleave while varying emitters and TMNR. There are no missing or spurious pulses. Lower pulses is a better result.

	Mean Number of Pulses To Deinterleave – No missing or spurious pulses								
Emitters	**2**			**3**			**4**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	14.9	14.3	14.3	16.4	18	17.8	15.9	18.6	19.5
CDIF SS	14.4	14.7	14.3	16.3	18.1	17.7	14.6	18.9	19.4
TOA Difference Histogram	28.3	27.4	28.4	34.5	35.7	38.6	36	N/A	N/A
SDIF	21.7	20.5	19.1	21.7	26	24.3	20.5	19.7	30
SDIF SS	22.5	21.6	19.3	19.6	28.5	24.9	25.6	24	30.8
Sequence Search	12.3	7.3	6.3	12.6	10.5	8.2	10.7	11.3	10.2

Table 3.11: Mean Number of Pulses - Mixed PRI - Emitters

[2] A jittered PRI scheme is a constant PRI scheme with intentionally added noise.

The mean number of pulses required to deinterleave are expected to increase as the number of emitters increase. This is the case for CDIF, CDIF SS, SDIF and SDIF SS. The TOA difference histogram algorithm also experiences the increase of mean pulses with emitters, until it completely fails to deinterleave in the cases of mid and high TMNR with four emitters in the EW environment. The mean number of pulses required to deinterleave using the sequence search algorithm stays approximately the same with the increase in emitters, meaning that it required fewer pulses with more emitters.

Table 3.12 shows the mean number of pulses each algorithm takes to deinterleave while varying interference pulses (spurious and missing) and TMNR. The number of emitters is kept constant at two.

	Mean Number of Pulses To Deinterleave – 2 Emitters											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	14.9	14.3	14.3	12.3	10.7	10.7	13	11.5	11.6	11.7	10.3	10
CDIF SS	14.4	14.7	14.3	12.4	11.4	11.5	13	11.5	11.6	11.9	10.5	10.3
TOA Difference Histogram	28.3	27.4	28.4	23.9	19.6	18.73	26.3	22.2	22.5	24.7	18.8	18.5
SDIF	21.7	20.5	19.1	18.9	15.1	13.2	19.4	16.8	15.6	19.1	15	13.7
SDIF SS	22.5	21.6	19.3	18.1	14.9	13.7	21.1	17.1	15.2	20.3	15.6	14.2
Sequence Search	12.3	7.3	6.3	9.9	7.5	6.8	9.7	7.5	6.5	9.7	7.5	6.9

Table 3.12: Mean Number of Pulses - Mixed PRI - Interference Pulses

The change in performance for all the algorithms is the same as discussed in the analysis of Tables 3.4 and 3.8. All algorithms take fewer or the approximately the same number of pulses to deinterleave with missing and spurious pulses.

3.2.4 Conclusion

After analysing the results of the algorithms in MATLAB, conclusions on the suitability of the algorithms in the final system were made. Since different thresholds are used in literature and a limited number of pulses were processed when running algorithms, these conclusions could only apply to the algorithms in the context of the system to be implemented on the DRFM.

For all algorithms, generally as the number of emitters increase, their success percentage decreases. Another general observation is that success percentage increases with missing and spurious pulses. With regards to the mean number of pulses each algorithm needs to process before all the pulse trains are correctly deinterleaved, it generally increases with the number of emitters. This is to be expected as more emitters mean more pulses in the EW environment. The presence of missing and spurious pulses actually helps reduce the number of pulses needed to deinterleave in all algorithms. The performance of all deinterleaving algorithms was severely reduced when deinterleaving interleaved pulse trains consisting only of emitters with jittered PRI schemes.

The success percentage of the sequence search algorithm decreases with the increase of noise in TOA. It was noticed that when correct detections are made at lower TMNR, the algorithm requires fewer pulses. This is attributed to wider gates being used at lower TMNR. The sequence search algorithm is the least affected by the number of emitters in the EW environment, in comparison to other algorithms simulated. The algorithm is also negatively affected the most by missing pulses compared to other algorithms simulated.

The TOA difference histogram algorithm is, in relation to the other algorithms simulated, most adversely affected by number of emitters. The algorithm performs much better with higher TMNR. From all the algorithms simulated, it requires the most amount of pulses to properly deinterleave pulse trains.

The CDIF offers the best success percentage compared to all other algorithms simulated in most situations. It requires fewer pulses to deinterleave than the SDIF algorithm, which according to literature was introduced as an improvement over the CDIF algorithm. The reason behind this is the CDIF algorithm does not discard counts from lower difference levels. Therefore the threshold can be exceeded with fewer pulses. In the case of many emitters, having jittered PRI pulse trains increased the success percentage because the bin sizes were larger.

The CDIF SS algorithm is supposed to be an improvement over the standard CDIF algorithm, but it actually offers a slightly lower success percentage. This is because the sequence search part of the algorithm is still affected by noise. It does require the same amount of pulses as the CDIF algorithm to properly deinterleave pulse trains.

The SDIF did not offer a better success percentage than the CDIF algorithm. This may be because by discarding counts, the number of bins and therefore threshold changes often to correctly make detections. The SDIF has the highest performance gain when introduced to missing and spurious pulses as compared to the other algorithms simulated. This was stated in the literature and is confirmed here. The SDIF algorithm is very susceptible to noise in TOA, which was not even mentioned in the literature. This is only made worse when deinterleaving pulse trains that have a significant difference in noise.

The SDIF SS algorithm also has a drop in success percentage when compared to the SDIF algorithm. It also uses the same amount of pulses as SDIF to correctly deinterleave.

The CDIF algorithm has the highest success percentage in most cases compared to the other algorithms simulated. The CDIF SS algorithm is slightly outperformed by the CDIF algorithm in all the situations considered. Behind the sequence search algorithm, the CDIF and CDIF SS algorithms use the least amount of pulses to properly deinterleave pulse trains. The conclusions drawn from these simulation results show that the CDIF and CDIF SS algorithms were best suited to be implemented in hardware (the DRFM).

3.3 Deinterleaving Hardware Results

The DRFM is an FPGA based system, on which two TOA based deinterleaving algorithms were chosen to be implemented on. Their performance in hardware was compared against their performance in MATLAB, as well as to one another on the DRFM in order to select the best suited algorithm to be implemented on the system. Conclusions which were drawn from the simulation results, such as the effects of the number of emitters, TMNR and interference pulses have on the algorithms will not be discussed here.

The CSIR 5^{th} generation DRFM platform consists of two FPGAs [83]. A control FPGA (Xilinx Virtex 5 - XC5VSX35T-2FFG665C) and a processing FPGA (Xilinx Virtex 6 - XC6VSX475T-2FFG1759C). Due to branching logic of the algorithms, they were implemented on a softcore processor (a MicroBlaze) on the processing FPGA. The control FPGA controls all the other devices in the DRFM, such as memory banks, LEDs,

interfaces, etc. The control FPGA is of interest in this study because it manages all communications protocols of the DRFM. It controls the communication between MATLAB on a laptop and the processing FPGA in the DRFM over an Ethernet cable using the User Datagram Protocol (UDP). A picture of this can be seen in Appendix B.

The EW environments were simulated in MATLAB, exactly as before. As only unsigned 32-bit integers can be used with the DRFM, TOA values from these EW environments were first converted from double-precision floating-point numbers to unsigned 32-bit integers (uint32) before being sent to the DRFM. Before the conversion, the TOAs were multiplied by 100 and rounded to the nearest integer to at least preserve two decimal places of the original TOA. To determine the uncertainty or loss of resolution associated with this conversion, MATLAB was used to convert random double-precision floating-point numbers to uint32 as they would be on the DRFM and then back to double-precision floating-point numbers. Using the error between the numbers that underwent the conversions and the original numbers, it was found that the conversion led to $\pm 0.5\%$ uncertainty in TOA measurements.

The algorithms implemented on the DRFM were exposed to three types of interleaved pulse trains exactly as before. A snapshot of the hardware results for the deinterleaving algorithms can be seen in Tables 3.13 to 3.24. The complete emitter deinterleaving hardware results can be found in Appendix D.

3.3.1 Constant PRI Signals

The first type of interleaved pulse train tested with the deinterleaving algorithms consisted of emitters having only constant PRI schemes. The PRIs for each emitter were as follows:

Emitter 1: 50 μs, Emitter 2: 75 μs, Emitter 3: 31 μs, Emitter 4: 88 μs

In the case of two emitters being interleaved, the pulses from emitters 1 and 2 are present in the interleaved pulse train. In the case three emitters are interleaved, the pulses from emitters 1 to 3 are present in the interleaved pulse train. Finally, in the case of 4 interleaved emitters, the pulses from emitters 1 to 4 are present in the interleaved pulse train. The results of each algorithm were measured using the percentage of success and mean number of pulses to deinterleave.

3.3.1.1 Percentage of Success

Success is defined as when an algorithm correctly deinterleaves all emitters in an EW environment without any false emitters being detected. Therefore, the percentage of success is the percentage number of times the algorithm correctly deinterleaved all emitters in the environment without any false emitters being detected in the one thousand runs. A higher percentage of success indicates better performance. Table 3.13 shows the percentage of success each algorithm achieves with varying emitters and TMNR (time measurement to noise ratio). There are no missing or spurious pulses. High TMNR refers to the case of a 26 dB TMNR, mid TMNR refers to the 16 dB case and low TMNR refers to the 8 dB case.

	Percentage of Success (%) – No missing or spurious pulses								
Emitters	**2**			**3**			**4**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	39.3	89.1	49.5	12.4	33.5	30.1	2.2	3.5	0
CDIF SS	37.7	99.6	95.4	21.3	59.1	15.6	8.9	28.2	0

Table 3.13: Hardware Results: Percentage of Success - Constant PRI - Emitters

The performance of the CDIF SS algorithm has increased for all situations compared to its simulation results. This is due to loss of resolution from changing number schemes, which actually helps the sequence search part of the algorithm perform better. Variations in TOA is effectively much more, effectively widening variations allowed while matching in the sequence search algorithm. Using the method described in section 3.3, it was calculated that the number conversion process varied the variations in TOA by $\pm 1\%$. This led to a gate and bin size change between -0.75% and $+1.4\%$.

The performance of the CDIF algorithm has increased in situations with two and three emitters at low and mid TMNR compared to simulations. In all other situations, the performance decreased or stayed the same. The CDIF algorithm also experiences the effectively wider bins, due to variations in TOA being effectively less but there is no sequence search to disprove false detections. The wider bins at high TMNR and situations with 4 emitters results in more false detections at high TMNR, resulting in a decreased percentage of success.

The CDIF SS now beats the performance of the CDIF in six out of nine situations on the DRFM.

Table 3.14 shows the percentage of success each algorithm achieves with varying interference pulses (spurious and missing) and TMNR. The number of emitters is kept constant at 2.

	Percentage of Success (%) – 2 Emitters											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	39.3	89.1	49.5	52	85.3	66	35.3	72.4	41.5	36.2	66.3	55.6
CDIF SS	37.7	99.6	95.4	45.8	92.8	87.7	33.5	86	81	39.5	79.5	76

Table 3.14: Hardware Results: Percentage of Success - Constant PRI - Interference Pulses

The CDIF algorithm performance is higher in simulations for high TMNR. The same reasoning that was previously applied, where by wider bins allow for more false detections, is also applicable in this scenario. The CDIF SS algorithm performance increases for all situations compared to simulation results. The CDIF SS offers better performance in nine out of twelve situations.

3.3.1.2 Mean Number of Pulses To Deinterleave

The mean number of pulses is the average number of pulses processed before an algorithm successfully deinterleaved all emitters in the EW environment, i.e. the average number of pulses processed when there was a success. The lower the mean number of pulses needed to deinterleave the better the performance of the algorithm. This number includes the zeroth pulse, $TOA_0 = 0s$. Table 3.15 shows the mean number of pulses each algorithm takes to deinterleave while varying emitters and TMNR. There are no missing or spurious pulses.

	Mean Number of Pulses To Deinterleave – No missing or spurious pulses								
Emitters	**2**			**3**			**4**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	7.7	6.3	6.3	9	9	8.6	10.2	11	N/A
CDIF SS	11.7	9.9	8.3	14.7	14.9	12.9	17.7	19.1	N/A

Table 3.15: Hardware Results: Mean Number of Pulses - Constant PRI - Emitters

The CDIF algorithm needs fewer pulses to deinterleave correctly in the hardware implementation as compared to the simulations. In section 3.3.1.1, it was reasoned that the CDIF algorithm has more false detections on the hardware implementation, this corresponds to it detecting emitters earlier as effectively more pulses can be associated with one bin.

The CDIF SS algorithm requires more pulses to deinterleave in situations with four emitters correctly. In hardware, there are more false detections by the CDIF algorithm before it is checked by the SS algorithm. By making the SS test for an emitter earlier, the number of pulses required has decreased. In the cases where the pulses have increased compared to simulations, success percentage has increased. This means that the SS disproves a wrong detected emitter, making the CDIF SS algorithm process more pulses before an actual emitter is detected.

CDIF deinterleaves much earlier than the CDIF SS algorithm in all situations.

Table 3.16 shows the mean number of pulses each algorithm takes to deinterleave while varying interference pulses (spurious and missing) and TMNR. The number of emitters is kept constant at two. In the mean number of pulses, the zeroth pulse $(TOA_0 = 0s)$ and spurious pulses are counted, while missing pulses are excluded.

	Mean Number of Pulses To Deinterleave – 2 Emitters											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	7.7	6.3	6.3	7.8	6.1	5.8	7.7	6.5	6.6	7.6	6.3	6.1
CDIF SS	11.7	9.9	8.3	11.2	9.8	8.5	11.6	10.3	9.1	10.9	10	9.2

Table 3.16: Hardware Results: Mean Number of Pulses - Constant PRI - Interference Pulses

In all situations, both the CDIF and CDIF SS algorithms deinterleave in fewer pulses than what was required in simulations. This is attributed to the loss of resolution by changing the number system, making bins effectively wider which led to more pulses being associated with one bin and exceeding the threshold sooner. The CDIF algorithm requires fewer pulses to deinterleave than the CDIF SS algorithm.

3.3.2 Jittered PRI Signals

The second type of interleaved pulse train tested with the deinterleaving algorithms consisted of emitters having only jittered PRI schemes. The PRIs for each emitter were as follows:

Emitter 1: 50 μs (10% jitter), Emitter 2: 75 μs (10% jitter), Emitter 3: 31 μs (10% jitter), Emitter 4: 88 μs (10% jitter)

In the case of two emitters being interleaved, the pulses from emitters 1 and 2 are present in the interleaved pulse train. In the case three emitters are interleaved, the pulses from emitters 1 to 3 are present in the interleaved pulse train. Finally, in the case of 4 interleaved emitters, the pulses from emitters 1 to 4 are present in the interleaved pulse train. The results of each algorithm were measured using the percentage of success and mean number of pulses to deinterleave.

3.3.2.1 Percentage of Success

Table 3.17 shows the percentage of success each algorithm achieves with varying emitters and TMNR. There are no missing or spurious pulses.

	Percentage of Success (%) – No missing or spurious pulses								
Emitters	**2**			**3**			**4**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	39.1	75.6	62	11.3	18.1	11.4	2.4	2.6	1.2
CDIF SS	43.1	68.7	67.8	20.9	36.1	37.7	5.8	13.3	14

Table 3.17: Hardware Results: Percentage of Success - Jittered PRI - Emitters

The CDIF algorithm increases performance for all situations except in situations with four emitters, where its performance decreases compared to simulations. Four emitters mean more pulses for the algorithm to process and since pulses have added jitter, it can be deduced that the CDIF algorithm detects more false emitters in noisy pulse dense environments.

The CDIF SS algorithm experiences an increase in performance for all situations compared to simulations, except for the situation with four emitters at low TMNR. The low TMNR together with the added noise from the jitter has adverse effects on the SS part of the algorithm. It is likely that the SS part of the algorithm rejects most of the emitters it encounters, including the correct ones.

The CDIF SS algorithm performs better than the CDIF algorithm in eight out of nine situations.

Table 3.18 shows the percentage of success each algorithm achieves with varying interference pulses (spurious and missing) and TMNR. The number of emitters is kept constant at two.

	Percentage of Success (%) – 2 Emitters											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	39.1	75.6	62	47.4	76.1	62.8	30	61.5	39.9	35.6	60.9	49.1
CDIF SS	43.1	68.7	67.8	48	74.4	76.1	36.1	67.1	68.2	39.3	66.3	65.4

Table 3.18: Hardware Results: Percentage of Success - Jittered PRI - Interference Pulses

The performance of both the CDIF and CDIF SS algorithms have improved over their performance observed in simulations. There is one exception though, in the presence of ten percent missing and spurious pulses, the performance of the CDIF algorithm at high TMNR decreases. The reason for this is there are more false detections made by the CDIF algorithm on the DRFM due to the loss of resolution or precision from changing number schemes.

The CDIF SS algorithm outperforms the CDIF algorithm in ten out of twelve situations.

3.3.2.2 Mean Number of Pulses To Deinterleave

Table 3.19 shows the mean number of pulses each algorithm takes to deinterleave while varying emitters and TMNR. There are no missing or spurious pulses.

	Mean Number of Pulses To Deinterleave – No missing or spurious pulses								
Emitters	**2**			**3**			**4**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	8	7.4	6.4	8.9	9.3	9.8	9.8	10.2	10.8
CDIF SS	10.9	10.5	10.2	14.9	15.2	15.1	18.1	18.5	18.4

Table 3.19: Hardware Results: Mean Number of Pulses - Jittered PRI - Emitters

The CDIF algorithm deinterleaves quicker on the DRFM compared to simulations. The CDIF SS algorithm also deinterleaves quicker on the DRFM, except for the situations with 4 emitters, as compared to simulations. The reason for this change in pulses required to deinterleave was explained in the discussion related to Table 3.15.

The hardware implementation of the CDIF algorithms takes fewer pulses to deinterleave than the hardware implementation of the CDIF SS algorithm.

Table 3.20 shows the mean number of pulses each algorithm takes to deinterleave while varying interference pulses (spurious and missing) and TMNR. The number of emitters is kept constant at two.

	Mean Number of Pulses To Deinterleave – 2 Emitters											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	8	7.4	6.4	7.6	7.2	6.4	7.9	7.6	6.8	7.8	7.3	6.6
CDIF SS	10.9	10.5	10.2	11.1	10.6	10.6	10.9	11	10.8	10.8	10.8	10.8

Table 3.20: Hardware Results: Mean Number of Pulses - Jittered PRI - Interference Pulses

The CDIF requires fewer pulses to deinterleave on the DRFM as compared to simulations. The CDIF requires fewer pulses as well for cases with 2 and 3 emitters as compared to simulations. The reason for this because the performance of the SS part of the algorithm increases, therefore more false emitters are disproved before the actual emitters are found after more pulses are processed.

The CDIF algorithm again requires fewer pulses to deinterleave compared to the CDIF SS algorithm.

3.3.3 Mixed PRI Signals

The third and final type of interleaved pulse train tested with the deinterleaving algorithms consisted of emitters having both constant and jittered PRI schemes. The PRIs for each emitter were as follows:

Emitter 1: 50 μs (10% jitter), Emitter 2: 75 μs, Emitter 3: 31 μs (10% jitter), Emitter 4: 88 μs

In the case of two emitters being interleaved, the pulses from emitters 1 and 2 are present in the interleaved pulse train. In the case of three emitters being interleaved, the pulses from emitters 1 to 3 are present in the

interleaved pulse train. Finally, in the case of 4 interleaved emitters, the pulses from emitters 1 to 4 are present in the interleaved pulse train. The results of each algorithm were measured using the percentage of success and mean number of pulses to deinterleave.

3.3.3.1 Percentage of Success

Table 3.21 shows the percentage of success each algorithm achieves with varying emitters and TMNR. There are no missing or spurious pulses.

	Percentage of Success (%) – No missing or spurious pulses								
Emitters	**2**			**3**			**4**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	39	82.3	85.3	12.3	17	19.1	3.2	3	2.3
CDIF SS	43.4	88.3	90.4	20.9	40.7	42.5	8.2	14.8	15.5

Table 3.21: Hardware Results: Percentage of Success - Mixed PRI - Emitters

The percentage of success increases for the hardware implementation of CDIF SS algorithm compared simulations for all situations, except for four emitters at high TMNR. This is due to the mixture of jittered and constant PRI schemes in the interleaved pulse train. At high TMNR, with the two different amounts of noise in the signals, the SS algorithm gate became too big and accepted more false detections. The performance of the CDIF SS algorithm is still better than the performance of the CDIF algorithm in this situation.

The performance of the DRFM implementation of the CDIF algorithm decreases as compared to the simulation results, except in the case of two emitters. The reason for this is, bins which are already effectively wider due to the change in number scheme, are made even wider by the extra noise added by jittered PRI pulse trains. These extremely wide bins are used to deinterleave constant PRI pulse trains, which results in more false detections.

The CDIF SS algorithm performs better than the CDIF algorithm in all cases.

Table 3.22 shows the percentage of success each algorithm achieves with varying interference pulses (spurious and missing) and TMNR. The number of emitters is kept constant at two.

	Percentage of Success (%) – 2 Emitters											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	39	82.3	85.3	48.2	80.7	85.1	30.3	63.8	65.2	34.6	67	62.2
CDIF SS	43.4	88.3	90.4	53.3	84.7	88.2	36.8	76.5	78.9	41.5	71.2	77.4

Table 3.22: Hardware Results: Percentage of Success - Mixed PRI - Interference Pulses

The performance of the CDIF algorithm implemented in hardware increases compared to simulations. The reason explained earlier where jittered PRI pulse trains make bin sizes even wider while detecting constant PRI pulse trains does not apply here as the two interleaved pulse trains consists of one jittered and one constant PRI pulse train.

The CDIF SS algorithm performance in hardware is better than its performance in simulations. The CDIF SS algorithm also out performs the CDIF algorithm on the DRFM.

3.3.3.2 Mean Number of Pulses To Deinterleave

Table 3.23 shows the mean number of pulses each algorithm takes to deinterleave while varying emitters and TMNR. There are no missing or spurious pulses.

	Mean Number of Pulses To Deinterleave – No missing or spurious pulses								
Emitters	**2**			**3**			**4**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	7.7	7	7	8.8	9.3	9.2	9.7	10.5	10.7
CDIF SS	10.8	10.4	9.6	14.9	15.1	14.6	18	18.8	17.3

Table 3.23: Hardware Results: Mean Number of Pulses - Mixed PRI - Emitters

Both the CDIF and CDIF SS algorithms require fewer pulses to deinterleave in hardware compared to their simulations. The CDIF algorithm needs fewer pulses to deinterleave than the CDIF SS algorithm with this type of interleaved pulse train as in the case of the other interleaved pulse trains.

Table 3.24 shows the mean number of pulses each algorithm takes to deinterleave while varying interference pulses (spurious and missing) and TMNR. The number of emitters is kept constant at two.

	Mean Number of Pulses To Deinterleave – 2 Emitters											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
CDIF	7.7	7	7	7.6	6.8	6.7	7.6	7.1	7.2	7.8	7	6.9
CDIF SS	10.8	10.4	9.6	10.7	10.2	9.5	10.9	10.4	10	10.7	10.5	9.9

Table 3.24: Hardware Results: Mean Number of Pulses - Mixed PRI - Interference Pulses

Both algorithms require fewer pulses than their simulation counterparts. The CDIF algorithm again requires fewer pulses than the CDIF SS algorithm.

3.3.4 Conclusion

The CDIF and CDIF SS algorithms were implemented on the DRFM platform to investigate any difference between simulation and hardware performance. The algorithm that behaved the best will be implemented in the system outlined in Chapter 1.

EW environments were simulated in MATLAB, and the simulated TOA measurements associated with the pulses in the environment were sent to the DRFM. To accomplish this, the double-precision floating-point TOA values in MATLAB had to be converted to unsigned 32-bit integers. This change in number scheme meant some resolution ($\pm 0.5\%$) in the simulated TOA measurements was lost.

The loss of resolution resulted in increased variations in the measured TOA ($\pm 1\%$), which effectively changed the gate and bin sizes of the histograms (between -0.75% and $+1.4\%$) drawn in the CDIF and CDIF SS algorithms. This in turn caused the CDIF algorithm to detect emitters faster, however it also allowed for the detection of an increased number of false emitters. Fortunately the CDIF SS algorithm tests if a detected

emitter is an actual emitter in the environment using the search sequence algorithm. This allowed the CDIF SS algorithm to have a higher percent of success than the CDIF algorithm in most cases.

It was also found when more than one jittered pulse train is a part of an interleaved pulse train, performance is degraded because bin sizes are made even wider with the extra intentionally added jitter.

From the deinterleaving hardware results, it can be concluded that the CDIF SS algorithm performs better than the CDIF algorithm. The CDIF algorithm is quicker as it requires fewer pulses to deinterleave, but this is because it does not do a secondary check like the CDIF SS algorithm. The secondary check is what disproves false emitter detections and increases the success percentage of CDIF SS algorithm. The CDIF SS algorithm was chosen to be implemented on the system to minimise false detections. More false detections mean more trackers are started which can consume resources in the DRFM.

Chapter 4

Emitter TOA Tracking

The aim of this Chapter is to choose the most suitable TOA based tracking algorithm to be used in the system discussed in Chapter 1. This is accomplished by first simulating all the relevant tracking algorithms researched in the literature review (section 2.8). After comparing the simulation results, the two best-performing algorithms were chosen to be implemented in hardware (the DRFM), and their simulation results were compared with the hardware results. The hardware results were used to choose which algorithm is most suitable for the system was made. In all simulation and hardware testing, the tracking algorithms were exposed to single emitter EW environment as outlined in section 3.1.

Before a tracking algorithm is run, certain estimates in the initial set up of the algorithms need to be made.

4.1 Initial Set Up

It is assumed when investigating the tracking algorithms that an initial estimate for the PRI (or PRI sequence) of the emitter and tracking window (or gate) is obtained from the deinterleaving algorithm that identified the emitter. The tracking window or gate is about 1.5 times the size of the histogramming bin size when the emitter was found. Some of the tracking algorithms also need an initial estimate of the noise variances σ_w^2 and σ_j^2, mentioned in section 2.7.2. The variance associated with a measured PRI can be calculated, using equation 2.19, as follows [8]

$$measuredPRI_n = measuredTOA_n - measuredTOA_{n-1} = PRI_n + w_n + v_n - v_{n-1}$$

$$var\,(measuredPRI_n) = \sigma_w^2 + 2\sigma_v^2 \tag{4.1}$$

The noise variance associated with each measurement is therefore $\sigma_w^2 + 2\sigma_v^2$. Since the width of the gate (tracking window) is usually ± 4 standard deviations of the prediction [8], the noise variances σ_w^2 and σ_j^2 can be estimated.

$$gate = \pm 4\sqrt{\sigma_w^2 + 2\sigma_v^2} \tag{4.2}$$

$$\sigma_w^2 + 2\sigma_v^2 = \frac{gate^2}{16} \tag{4.3}$$

It is safe to assume that the noise added by the EW receiver (v_j) is much smaller than the noise due to external factors and the transmitting radar system (w_j). If the assumption that $3\sigma_v \approx \sigma_w$ is made then from equation 4.3.

$$(3\sigma_v)^2 + 2\sigma_v^2 \approx \frac{gate^2}{16}$$

$$11\sigma_v^2 \approx \frac{gate^2}{16}$$

$$\sigma_v^2 \approx \frac{gate^2}{176}$$

$$\sigma_v \approx \sqrt{\frac{gate^2}{176}} \tag{4.4}$$

From the assumption that $3\sigma_v \approx \sigma_w$,

$$\sigma_w \approx 3 \times \sqrt{\frac{gate^2}{176}} \tag{4.5}$$

Equations 4.4 and 4.5 can be used to make initial estimates for σ_v and σ_w, respectively based on an assumption made. Equations 4.4 and 4.5 were used to make initial estimates for σ_v and σ_w in simulations, hardware testing and as well as the final system.

The tracking algorithms will also need to determine the period (number of PRIs before the PRI sequence repeats itself) in order to properly track an emitter [3]. However, this is not as important because the investigated deinterleaving algorithms only extract constant PRI pulse trains.

4.2 Tracking Simulation Results

After TOA based tracking algorithms had been investigated, they were implemented in MATLAB. Each algorithm was given simulated TOA measurement of pulses in different EW environments. The different EW environments were also simulated in MATLAB and was explained in detail in section 3.1. Since a tracker would start after the deinterleaving algorithm finds an emitter in the system described in Chapter 1, it can be assumed that the tracker will be able to get some information from the deinterleaving algorithm. Some tracking algorithms will need to get an initial PRI estimate, the last received TOA from the pulse train, a tracking gate estimate and time measurement noise estimates before it can begin. A first difference level histogram was created before the each simulation began to get a bin size for the algorithm to estimate a tracking gate.

The initial PRI (or PRI sequence) estimate and the last received TOA will be obtained directly from the deinterleaving algorithm. The deinterleaving algorithms investigated only extracts constant PRI pulse trains. Section 2.6 explained how all types of PRI pulse trains, of interest to this study, can be broken down into a noisy constant PRI pulse train or a series of constant PRI pulse trains. At a later stage a track manager, that merges

trackers with the same PRI into a staggered or dwell and switch PRI tracker could be added to the system. Therefore, the performance of the tracking algorithms with all PRI schemes of interest was investigated.

The gate is estimated as one and a half times the bin size from the deinterleaving algorithm when the emitter was detected. The gate estimate is then used to make initial estimates of the variance of the noise added by the EW receiver (σ_v^2) and the variance of the noise added by the transmitting radar (σ_w^2) using equation 4.4 and 4.5, respectively.

In simulations, and later in the hardware implementation, when an algorithm is tracking an emitter it makes a prediction of the next expected pulse. If the recently received TOA arrives before the predicted pulse TOA and associated tracking window, the pulse is considered to be a spurious pulse. The tracker algorithm will do nothing in this case as the pulse does not belong to the emitter being tracked. In the case of Fourier domain Kalman filter, this TOA will be ignored when tracking the PRI. The spurious pulses are discarded because the tracker algorithms, which are being investigated in this section, will not do anything with the spurious pulse. In the final system where multiple emitters exist, the spurious pulse will be added to the buffer, where a new emitter can be found by deinterleaving.

If the recently received TOA arrives after the predicted pulse TOA and associated tracking window, then the algorithm assumes that there is a missing pulse. The algorithm makes a new prediction for the subsequent expected pulse, and the tracking window is lengthened to account for variances of two pulse TOA measurements. This is called coasting.

None of the authors of the literature reviewed on the tracking algorithms simulated published their actual results, only their conclusions. Theses authors did not investigate the effect varying interference pulses and time measurement noise. A snapshot of the simulation results for the tracking algorithms investigated for all the different PRI schemes can be seen in Tables 4.1 to 4.16. The complete emitter TOA tracking simulation results can be found in Appendix E.

4.2.1 Constant PRI Type Signals

A constant PRI scheme emitter was tested against all the algorithms first. The emitter had a PRI of 50 μs. The results of each algorithm were measured using mean track loss, average standard deviation of normalised tracking error and average normalised tracking error versus pulse number.

4.2.1.1 Mean Track Loss

Track of an emitter was considered lost if there were fifteen (15) pulses in a row missing. Each time track was lost, the track was immediately regained at the point it was lost until all transmitted pulses from the emitter were processed. The mean track loss refers to average number times track was lost by the algorithm when processing all pulses belonging to one emitter. Table 4.1 shows the mean track loss of each algorithm with varying interference pulses (spurious and missing) and TMNR (time measurement to noise ratio). High TMNR refers to the case of a 26 dB TMNR, mid TMNR refers to the 16 dB case and low TMNR refers to the 8 dB case.

	Mean Track Loss											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Delta-τ Histogram	249.8	120.1	0	122.8	0.5	0	222.2	41.6	0.1	107.5	5.1	4.3
Alpha-Beta Filter	244.6	120.2	0	121.7	8.1	0	217.6	48.6	4.3	105.4	17.4	13.9
Kalman Filter - Time Domain	182.1	15.6	0	74.3	0	0	193.1	17.6	0	83.2	4.2	4.1
Kalman Filter - Fourier Domain	100.3	3.9	0	6.3	0	0	105.9	18.2	16.2	11.8	6.2	6.8

Table 4.1: Mean Track Loss - Constant PRI

From the table, we can see that performance of all the algorithms increases as TMNR increases. This is expected as less noise means predictions are more accurate. Missing pulses also increases the performance of all algorithms. The main reason why missing pulses affect the performance of the tracking algorithms because the tracking algorithms are intialised with a first difference histogram of a dataset that has missing pulses. Therefore, the initial estimates of the tracking gate, noise variance estimates and other variables associated with their specific tracking algorithms are affected because of the missing pulses. The performance of the tracking algorithms are affected because the initial estimates are affected. Another reason of this increase in performance is because when the algorithm is coasted because of a missing pulse, the tracking window gets bigger for the next pulse, i.e. the gate becomes twice as large to account for twice the noise in the time measurement of the pulse. This means that pulses that might have fallen out of the tracking window may now fall into it, helping decrease the number of times track is lost. The tracking window continues to widen as the algorithm is coasted until a pulse falls into the tracking window or track is lost. After a pulse falls into the tracking window, the tracking window is returned to its original size.

Performance of the Delta-τ histogram and the alpha-beta filter increases in spurious pulses. While spurious pulses decrease performance of both Kalman filter algorithms, more greatly in the Fourier domain version. This seems odd because spurious pulses are discarded without affecting the algorithm, therefore there should be no change in performance. However, estimates for the gate and noise variances are made from the histogram bin size before the algorithm begins. A first difference level histogram was used to make these estimates and that histogram included the spurious pulses. The extra pulses affect the bin size that dictates the estimates for the gate and noise variances, which affects the performance of the algorithms. This can apply to any situation where spurious pulses cause a change in performance. The Fourier domain Kalman filter seems to perform the best with constant PRI schemes.

4.2.1.2 Average Standard Deviation of Normalised Tracking Error

The normalised tracking error is the difference between the predicted TOA and the measured TOA normalised to the PRI. The standard deviation of normalised tracking error is therefore standard deviation in normalised tracking error in one run of the algorithm. Finally, the average standard deviation of normalised tracking error is the standard deviation in normalised tracking error of the algorithm averaged over one thousand runs. A lower average standard deviation of normalised tracking error means better overall tracking. Table 4.2 shows the average standard deviation of normalised tracking error for all the different algorithms with varying interference pulses and TMNR.

	Average Standard Deviation of Normalised Tracking Error ($\times 10^{-3}$)											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Delta-τ Histogram	3.6	0.9	0.1	6.9	1	0.1	4.3	0.9	0.1	6.8	1	0.3
Alpha-Beta Filter	3.7	1.2	0.1	8	2	0.1	4.5	1.5	0.1	9.2	2.6	1
Kalman Filter - Time Domain	0.8	0.2	0	1.7	0.3	0	0.8	0.2	0	1.7	0.4	0.1
Kalman Filter - Fourier Domain	7.1	1.9	0.1	9.5	1	0.1	7.2	2.1	0.4	9.1	2.5	2.1

Table 4.2: Average Standard Deviation of Normalised Tracking Error - Constant PRI

The performance of all algorithms increases with TMNR. All algorithms are negatively affected by missing pulses. The lower the TMNR, the greater the effect. Spurious pulses also degrade the performance of all algorithms, having a greater effect in cases with lower TMNR. This can also be attributed to the initial estimates made.

4.2.1.3 Average Normalised Tracking Error versus Pulse Number

The average normalised tracking error versus pulse number is the tracking error of the n^{th} pulse the algorithm processed from an emitter for one thousand runs averaged normalised to the PRI. This number is then averaged for the hundred pulses and then the second hundred pulses. As the number of pulses increases, the tracking algorithm should get better at predicting the TOA, making the normalised tracking error smaller. Since the average normalised tracking error can be a negative number, a value closer to zero is better. The normalised tracking error is the difference between the predicted TOA and the measured TOA normalised to the PRI. A negative normalised tracking error means that measured TOA is received after it was predicted to be received. Table 4.3, shows the effect TMNR has on tracking error as the number of pulses increase. There are no missing or spurious pulses.

	Average Normalised Tracking Error ($\times 10^{-3}$)							
TMNR	**8 dB**		**10 dB**		**20 dB**		**26 dB**	
Pulse Number	**0-100**	**101-200**	**0-100**	**101-200**	**0-100**	**101-200**	**0-100**	**101-200**
Delta-τ Histogram	4.2	4	2.1	2.1	0.2	0.1	0	0
Alpha-Beta Filter	4.1	4	2	2	0.2	0.1	0	0
Kalman Filter - Time Domain	1	0.6	0.8	0.3	0	0	0	0
Kalman Filter - Fourier Domain	30.1	29.4	18.1	17.8	-0.2	-0.2	0.1	0.1

Table 4.3: Average Normalised Tracking Error versus Pulse Number - Constant PRI - Effect of TMNR

The normalised tracking error is minimal at higher TMNR. At lower TMNR, the tracking error can be seen getting smaller as the number of pulses increase. At higher TMNR, the tracking error stays the same as the number of pulses increase. This is because it reaches an average steady tracking error for the associated TMNR.

Table 4.4, shows the effect interference pulses have on tracking error as the number of pulses increase. The TMNR is kept constant at 20 dB.

	Average Normalised Tracking Error @ 20 dB ($\times 10^{-3}$)							
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**		**M=10% S=0%**		**M=0% S=10%**		**M=10% S=10%**	
Pulse Number	**0-100**	**101-200**	**0-100**	**101-200**	**0-100**	**101-200**	**0-100**	**101-200**
Delta-τ Histogram	0.2	0.1	0.01	0.1	0.6	0.7	0.2	0.1
Alpha-Beta Filter	0.5	0.5	0.3	0.3	0	0	2.5	2.4
Kalman Filter - Time Domain	0.01	-0.01	0.01	0.01	0	0	0.02	0.01
Kalman Filter - Fourier Domain	-0.2	-0.2	-1.2	-1.2	0.5	0.7	20.2	22.3

Table 4.4: Average Normalised Tracking Error versus Pulse Number - Constant PRI - Effect of Missing and Spurious Pulses

The Fourier domain Kalman filter is negatively affected by missing and spurious pulses. The other three algorithms actually have a lower tracking error for the first hundred pulses before returning to the tracking error if there was no missing pulses. Spurious pulses increase tracking error in all algorithms. Initial estimates are to blame here as well.

4.2.2 Dwell and Switch PRI Type Signals

The second PRI scheme tested against all the algorithms was a dwell and switch PRI scheme. The dwell and switch PRI sequence of the emitter was 50 µs, 50 µs, 50 µs, 30 µs, 30 µs, 30 µs. The results of each algorithm were measured using mean track loss, average standard deviation of normalised tracking error and average normalised tracking error versus pulse number.

4.2.2.1 Mean Track Loss

Table 4.5 shows the mean track loss of each algorithm with varying interference pulses (spurious and missing) and TMNR.

	Mean Track Loss											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Delta-τ Histogram	178	109.6	3.4	91.6	3.6	3.6	171.5	43.4	3.9	90.5	15.3	13.5
Alpha-Beta Filter	172.6	114.3	3	91.4	12.3	3.1	166.7	60.6	3.8	91.1	32.4	19.9
Kalman Filter - Time Domain	157	14.2	0	73.6	0.5	0.6	164.8	18.7	0	82.3	12.7	14.5
Kalman Filter - Fourier Domain	218.8	276.8	293	123.9	149.5	190.8	132.9	160.6	189.5	221.8	277.9	282.8

Table 4.5: Mean Track Loss - Dwell and Switch PRI

The Fourier domain Kalman filter offers the worst performance for this PRI scheme. This is noticeable because it performed the best with a constant PRI scheme. The increase in TMNR decreases performance. It can be concluded that the Fourier domain Kalman filter is not suitable for dwell and switch PRI schemes. This could be because it dwells on a PRI before switching to another like a staggered PRI scheme or it does not track each individual PRI in the PRI sequence. All other algorithms have a lower amount of mean track loss as compared to their performance with a constant PRI scheme.

Apart from the Fourier domain Kalman filter, increase in TMNR increases performance for all algorithms. All the algorithms performance increases with missing pulses. Performance of the Delta-τ histogram, time domain

Kalman filter and the alpha-beta filter increases in spurious pulses. The time domain Kalman filter seems to perform the best here.

4.2.2.2 Average Standard Deviation of Normalised Tracking Error

Table 4.6 shows the average standard deviation of normalised tracking error for all the different algorithms with varying interference pulses and TMNR.

	Average Standard Deviation of Normalised Tracking Error ($\times 10^{-3}$)											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Delta-τ Histogram	3.8	0.9	0.1	7.7	1.1	0.2	4.3	1.2	0.1	8	1.9	0.9
Alpha-Beta Filter	3.6	1.1	0.1	8.2	1.9	0.2	4.4	1.4	0.1	9.1	4	2.6
Kalman Filter - Time Domain	0.9	0.2	0	2.3	0.3	0.1	0.9	0.2	0	2.3	0.4	0.2
Kalman Filter - Fourier Domain	5.4	8.1	0	8.9	9.5	12.5	5.4	8.4	4.9	8.1	9	10.7

Table 4.6: Average Standard Deviation of Normalised Tracking Error - Dwell and Switch PRI

With the exception of the Fourier domain Kalman filter, the performance of all algorithms increase with TMNR. Both missing and spurious pulses decrease the performance of all algorithms. The Fourier domain Kalman filter offers the worst performance for dwell and switch. The average standard deviation of normalised tracking error is slightly higher in this set of simulations as compared to the constant PRI scheme simulations. This is probably because the algorithms have to cycle between PRIs in this scheme, so there aren't as many corrections to each tracked PRI.

4.2.2.3 Average Normalised Tracking Error versus Pulse Number

Table 4.7, shows the affect TMNR has on tracking error as the number of pulses increase. There are no missing or spurious pulses.

	Average Normalised Tracking Error ($\times 10^{-3}$)							
TMNR	**8 dB**		**10 dB**		**20 dB**		**26 dB**	
Pulse Number	**0-100**	**101-200**	**0-100**	**101-200**	**0-100**	**101-200**	**0-100**	**101-200**
Delta-τ Histogram	3.5	4.3	3.7	3.5	0.5	0.4	0.01	0.01
Alpha-Beta Filter	4.6	3.9	3.6	2.7	0.3	0.3	0	0
Kalman Filter - Time Domain	1.5	1.4	1.6	1.1	0	0	0	0
Kalman Filter - Fourier Domain	9.4	9.3	10.3	9.3	7.1	5.8	0	0

Table 4.7: Normalised Tracking Error versus Pulse Number - Dwell and Switch PRI - Effect of TMNR

The Fourier domain Kalman filter offers the worst performance for here as well. The results for TMNR of 26 dB case for the Fourier domain Kalman filter only appears low because it lost track and then restarted track so many times. As TMNR increases, the normalised error decreases for all algorithms, except the Fourier domain Kalman filter. As the number of pulses increase, the normalised error decreases for all algorithms.

Table 4.8, shows the affect interference pulses have on tracking error as the number of pulses increase. The TMNR is kept constant at 20 dB.

	Average Normalised Tracking Error @ 20 dB ($\times 10^{-3}$)							
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**		**M=10% S=0%**		**M=0% S=10%**		**M=10% S=10%**	
Pulse Number	**0-100**	**101-200**	**0-100**	**101-200**	**0-100**	**101-200**	**0-100**	**101-200**
Delta-τ Histogram	0.45	0.42	7.7	0	0.17	0.22	11.6	3.8
Alpha-Beta Filter	0.3	0.3	121.6	0.991	0.03	0.02	14.8	8.21
Kalman Filter - Time Domain	0	0	-1.3	-0.01	0	0	-7.6	0.18
Kalman Filter - Fourier Domain	0.7	5.8	4.49	0.41	0.68	3.26	0.40	0.42

Table 4.8: Normalised Tracking Error versus Pulse Number - Dwell and Switch PRI - Effect of Missing and Spurious Pulses

With no missing or spurious pulses, the normalised error increases with pulse number for the Fourier domain Kalman filter. This confirms that the Fourier domain filter is not suited to dwell and switch PRI schemes.

Missing pulses negatively affects the performance of all other algorithms, which is extremely apparent in the alpha-beta filter. The reasons for this negative change in performance is because missing pulses influence the initial estimates made and there are fewer corrections made to each PRI as the filter is coasted more often. Spurious pulses increase performance because the extra pulses influence the initial estimates.

4.2.3 Jittered PRI Type Signals

The third PRI scheme tested against all the algorithms was a jittered PRI scheme. The jittered PRI of the emitter was 50 μs with 10% jitter. The results of each algorithm were measured using mean track loss, average standard deviation of normalised tracking error and average normalised tracking error versus pulse number.

4.2.3.1 Mean Track Loss

Table 4.9 shows the mean track loss of each algorithm with varying interference pulses (spurious and missing) and TMNR.

	Mean Track Loss											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Delta-τ Histogram	249.3	226	223.8	125.7	46.45	39.07	223	175.2	170.5	112.1	22.6	16.1
Alpha-Beta Filter	241.1	217.8	212.6	124.9	60.5	54.3	219.5	168.5	166.2	107.9	45.1	39.1
Kalman Filter - Time Domain	186.5	121.8	112.9	76	3.4	1	192.7	130.7	124.3	80.5	10.2	7.5
Kalman Filter - Fourier Domain	96.8	94.7	90.6	5.9	2.7	1.3	99.9	103.9	102.7	10.1	7.7	7.1

Table 4.9: Mean Track Loss - Jittered PRI

Delta-τ histogramming and alpha-beta filter algorithms experience the most track loss for this PRI scheme compared to the other schemes investigated. For all the algorithms performance increases with TMNR and missing pulses. In the presence of spurious pulses, performance increases with an increase in TMNR for all algorithms except the Fourier domain Kalman filter. The performance of the Delta-τ histogramming and alpha-beta filter algorithms increase with spurious pulses. Spurious pulses have a negative effect on the performance of the Kalman filter algorithms. As a jittered PRI scheme can be thought of as a constant PRI scheme with more noise, the Fourier domain Kalman filter has the best performance of all the algorithms again.

4.2.3.2 Average Standard Deviation of Normalised Tracking Error

Table 4.10 shows the average standard deviation of normalised tracking error for all the different algorithms with varying interference pulses and TMNR.

	Average Standard Deviation of Normalised Tracking Error ($\times 10^{-3}$)											
Missing (M) and Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Delta-τ Histogram	4	1.7	1.5	6.9	2.7	2.6	4.6	2.2	2	7	2.7	2.6
Alpha-Beta Filter	3.7	1.8	1.7	7.9	4.9	4.6	4.7	2.8	2.4	8.9	5.5	4.9
Kalman Filter - Time Domain	0.8	0.4	0.4	1.9	0.8	0.8	0.8	0.4	3.9	1.9	0.9	0.8
Kalman Filter - Fourier Domain	7.1	4.5	4.1	9.6	4	3.3	6.9	4.7	4.3	9.3	4.9	4.6

Table 4.10: Average Standard Deviation of Normalised Tracking Error - Jittered PRI

The extra noise added by the intentional jitter increases the average standard deviation of normalised tracking error of all algorithms in most cases. The only exceptions are the alpha-beta filter at TMNR of 8 dB with ten percent missing and ten percent spurious pulses and the Fourier domain Kalman filter at TMNR of 8 dB with ten percent missing pulses. The increases in performance are small and can be attributed to the change in initial estimates.

4.2.3.3 Average Normalised Tracking Error versus Pulse Number

Table 4.11, shows the affect TMNR has on tracking error as the number of pulses increase. There are no missing or spurious pulses.

	Average Normalised Tracking Error ($\times 10^{-3}$)							
TMNR	**8 dB**		**10 dB**		**20 dB**		**26 dB**	
Pulse Number	**0-100**	**101-200**	**0-100**	**101-200**	**0-100**	**101-200**	**0-100**	**101-200**
Delta-τ Histogram	4.1	4.2	2.3	2.7	2.7	2.8	2.6	2.5
Alpha-Beta Filter	4.8	4.9	2	2.5	3.1	3.2	3	3.6
Kalman Filter - Time Domain	1	0.7	1	0.4	0.7	0	0.6	0
Kalman Filter - Fourier Domain	31.3	28.8	22.7	22.7	-0.2	0.1	1.1	1.4

Table 4.11: Normalised Tracking Error versus Pulse Number - Jittered PRI - Effect of TMNR

At higher TMNR, the tracking error is still measurable. It was extremely small in the constant PRI scheme simulations. Tracking effect has increased for all algorithms due to the added jitter in the PRI. Tracking error of the Fourier domain Kalman filter increases with more pulses at TMNR of 26 dB. The tracking error of the Delta-τ histogram increases as pulses increase for all TMNR except TMNR of 26 dB. Tracking error of the alpha-beta filter increases with pulses for all TMNR. The time domain Kalman filter is the only algorithm that tracking error does not increase with pulses. We can conclude that added jitter does make it harder to track as the number of pulses increase. The Kalman filter, therefore, rejects the most noise compared to other algorithms.

Table 4.12, shows the affect interference pulses have on tracking error as the number of pulses increase. The TMNR is kept constant at 20 dB.

	Average Normalised Tracking Error @ 20 dB ($\times 10^{-3}$)							
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**		**M=10% S=0%**		**M=0% S=10%**		**M=10% S=10%**	
Pulse Number	**0-100**	**101-200**	**0-100**	**101-200**	**0-100**	**101-200**	**0-100**	**101-200**
Delta-τ Histogram	2.7	2.8	3	2.4	5.8	5.7	3.3	3.2
Alpha-Beta Filter	3.1	3.2	10.9	11.9	7.5	8.2	7.5	7.3
Kalman Filter - Time Domain	0.7	0	0.04	0.01	0.6	0.03	0.09	0.04
Kalman Filter - Fourier Domain	-0.2	0.06	9.9	9.6	3.6	5	31.6	33.5

Table 4.12: Normalised Tracking Error versus Pulse Number - Jittered PRI - Effect of Missing and Spurious Pulses

Tracking error increases with missing pulses for all algorithms, except the time domain Kalman filter. This is because the time domain Kalman needs fewer pulses to make corrections of more noisy time measurements. Spurious pulses increase the Kalman filter based algorithms and decrease alpha-beta and Delta-τ histogram performance. The reason for this is due to the initial estimates being affected by the spurious pulses.

4.2.4 Staggered PRI Type Signals

Finally, a staggered PRI scheme was tested against all the algorithms. The staggered PRI of the emitter was 50 μs, 30 μs, 80 μs. The results of each algorithm were measured using mean track loss, average standard deviation of normalised tracking error and average normalised tracking error versus pulse number.

4.2.4.1 Mean Track Loss

Table 4.13 shows the mean track loss of each algorithm with varying interference pulses (spurious and missing) and TMNR.

	Mean Track Loss											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Delta-τ Histogram	172.2	130	125.9	90	27.6	25.1	172.6	100.2	87.7	94.6	27.6	23.3
Alpha-Beta Filter	171	44.8	0	95.2	4.7	1	171.2	20.9	3	96	18.7	11.7
Kalman Filter - Time Domain	159.8	1.4	0	72	0.1	0	167.3	2.3	0	78.4	5	4.7
Kalman Filter - Fourier Domain	192.3	8.2	0	81.8	2.4	2.7	216.5	70.9	64.5	107.1	41.4	41.5

Table 4.13: Mean Track Loss - Staggered PRI

The Delta-τ histogram loses track the most for this PRI scheme. As TMNR increase, the mean track loss decreases for all algorithms, except the Fourier domain Kalman filter in the presence of missing pulses. The increase is very small and can be attributed to the initial estimates. Missing pulses increase performance for all the algorithms. Spurious pulses decrease the performance of all the algorithms, the Fourier domain Kalman filter is affected the most. The time domain Kalman filter seems to have the best performance for staggered PRI schemes.

4.2.4.2 Average Standard Deviation of Normalised Tracking Error

Table 4.14 shows the average standard deviation of normalised tracking error for all the different algorithms with varying interference pulses and TMNR.

	Average Standard Deviation of Normalised Tracking Error ($\times 10^{-3}$)											
Missing (M) and Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Delta-τ Histogram	6.7	3.3	3.1	10.9	4.6	3.7	6.9	3.9	3.6	10.7	4.4	3.5
Alpha-Beta Filter	6.4	2.2	0.2	11.2	3	0.4	6.8	2.5	0.3	10.7	4	1.7
Kalman Filter - Time Domain	1.9	4.7	0	3	0.6	0.1	1.9	0.5	0	2.5	0.7	0.2
Kalman Filter - Fourier Domain	8	1	0.1	9.3	1.2	0.4	7.4	1	0.2	8.2	1.4	0.9

Table 4.14: Average Standard Deviation of Normalised Tracking Error - Staggered PRI

The performance of all the algorithms increases with TMNR. The same conclusions drawn in the discussion related to Table 4.2 can be drawn here. The performance with this emitter is lower than with the constant PRI scheme emitter. The reason for this can be a combination of every PRI in the PRI sequence are updated fewer times, and the initial estimates for noise are much greater what they actually are because of the big difference in PRI values.

4.2.4.3 Average Normalised Tracking Error versus Pulse Number

Table 4.15, shows the affect TMNR has on tracking error as the number of pulses increase. There are no missing or spurious pulses.

	Average Normalised Tracking Error ($\times 10^{-3}$)							
TMNR	**8 dB**		**10 dB**		**20 dB**		**26 dB**	
Pulse Number	**0-100**	**101-200**	**0-100**	**101-200**	**0-100**	**101-200**	**0-100**	**101-200**
Delta-τ Histogram	16.2	17	13.6	12.9	11.1	10.7	10.5	10
Alpha-Beta Filter	15.1	15.8	12.5	12.7	2.3	1.5	0.1	0.1
Kalman Filter - Time Domain	4	3.4	3	1.3	0	0	0	0
Kalman Filter - Fourier Domain	27.3	26.6	14.2	14.4	0.2	0.3	-0.1	-0.1

Table 4.15: Normalised Tracking Error versus Pulse Number - Staggered PRI - Effect of TMNR

The table shows that at lower TMNR, the tracking error increases as the number of pulses increase for the Delta-τ histogram and alpha-beta filter algorithms. The tracking error for all the algorithms, except the Fourier domain Kalman filter is also much higher with this emitter as compared to an emitter with a constant PRI scheme. The reason behind this each PRI is updated less.

Table 4.16 shows the affect interference pulses have on tracking error as the number of pulses increase. The TMNR is kept constant at 20 dB.

	Average Normalised Tracking Error @ 20 dB ($\times 10^{-3}$)							
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**		**M=10% S=0%**		**M=0% S=10%**		**M=10% S=10%**	
Pulse Number	**0-100**	**101-200**	**0-100**	**101-200**	**0-100**	**101-200**	**0-100**	**101-200**
Delta-τ Histogram	11.1	10.7	10.1	9.3	15.4	14.4	9.6	10
Alpha-Beta Filter	2.3	1.5	1.3	0.6	0.02	0.02	3.8	2.6
Kalman Filter - Time Domain	0	0	0	0	0.03	0.01	0	0.03
Kalman Filter - Fourier Domain	0.2	0.3	-6.9	-7.4	1.7	2	0.2	0.3

Table 4.16: Normalised Tracking Error versus Pulse Number - Staggered PRI - Effect of Missing and Spurious Pulses

From the table, we can see tracking error is decreased with more pulses in all algorithms except the Fourier domain Kalman filter. Tracking error is decreased with missing pulses and increased with spurious pulses for the Delta-τ histogram, alpha-beta filter and time domain Kalman filter algorithms. The tracking error experienced by these algorithms is much greater with this emitter as compared to the constant emitter.

4.2.5 Execution Time

Table 4.17 shows the time taken for each algorithm to process all the EW environments containing different emitters. This time includes, in addition to running the algorithms, simulating all the different EW environments, running a first difference level histogram to make initial estimates and all other calculations. In the solution system, multiple trackers will be running at the same time. Each emitter that is found will have at least one tracker dedicated tracker[1]. Once all the emitters in the environment are found, the only pulses that are sent to the deinterleaving algorithm should be spurious pulses or pulses from a new emitter. Therefore the time needed to run the deinterleaving algorithm is not as important as the time required to execute the tracking algorithms. In the deinterleaving simulations, after all the emitters were found or a false emitter was detected, the algorithm stopped processing the rest of the pulses for that EW environment. This means how many pulses it takes to detect an emitter affected the execution time of the simulation, therefore the number of pulses required to deinterleave was more important than the execution time of the deinterleaving algorithms.

It should be noted that execution time is greatly affected by the computer that the MATLAB simulation is run on. To keep the execution times as similar as possible, the same computer was used with no user applications running in the background.

	Execution Time (s)			
PRI Scheme	**Constant**	**Dwell & Switch**	**Jittered**	**Staggered**
Delta-τ Histogram	4 849.9	2 312.4	6 482.9	5 754.7
Alpha-Beta Filter	1 921.9	2 003.3	2 522.1	1 993.5
Kalman Filter - Time Domain	10 934	11 186	25 303	11 226
Kalman Filter - Fourier Domain	50 178	125 170	53 739	66 756

Table 4.17: PRI Tracking Execution Time

A general trend noticed from the above table is that when an algorithm losses track more often, it takes a longer time to execute. From all the algorithms simulated, the fastest to slowest algorithms are alpha-beta filter, Delta-τ histogram, time domain Kalman filter and Fourier domain Kalman filter. The Kalman filter based algorithms take longer execution time due to the greater complexity of the algorithms. The Kalman filter requires many matrix multiplications and inversions. The order of the matrices increases in the case of the Fourier domain Kalman filter, this is why it is the most complex algorithm and takes the most time to execute.

The Fourier domain Kalman filter takes the longer when working with an emitter that has dwell and switch PRI scheme than other PRI schemes. This is to be expected as it lost track the most when processing those EW environments and the order of matrices are higher. The alpha-beta filter failed the most with a jittered PRI scheme emitter, but it was still the fastest executing algorithm, due to low complexity for that PRI scheme.

[1] Staggered and dwell & switch PRI schemes will have more than one

4.2.6 Conclusion

The main purpose of simulating all the TOA based tracking algorithms in MATLAB was to evaluate which algorithms were most suitable to be implemented in the system.

After testing each algorithm with simulated EW environments that included emitters with the PRI schemes that are of interest rest to this study, some common conclusions can be drawn. Track loss is inversely proportional to the time measurement to noise ratio meaning with less noise in the time measurement, there was less track loss. Tracking error, the error between the predicted TOA and the measured TOA, is inversely proportional to TMNR. Another conclusion is if the number of pulses increase, the tracking error usually becomes less until it reaches a steady state average value.

Spurious pulses have an influence on the algorithms, this is odd as the spurious pulses are discarded and should have no interaction with the running algorithm. However, when the initial estimates of the gate and noise variances are made, the spurious pulses are also included in the estimations. We see a decrease in track loss in the presence of missing pulses. The missing pulses also affects the estimation of these values, such that track loss increases for Kalman filter based algorithms and decreases for the Delta-τ histogram and alpha-beta filter algorithms.

Missing pulses also affects the tracking error. Tracking error was evaluated in two ways i.e. the average standard deviation of normalised tracking error and the average normalised error as the pulses increase for the one thousand runs of the different EW environments with the same characteristics. Whenever there is a missing pulse, the tracker is coasted, meaning that there is a new prediction without updating the old prediction. In a PRI sequence, this means the PRI estimate is not updated until the sequence repeats. This increases both the average standard deviation of normalised tracking error and average normalised error as the pulses increase.

Using the response of algorithms with a constant PRI type signal as a baseline for all comparisons, change in performance can be judged. With the dwell and switch PRI type signals the Fourier domain Kalman filter often fails no matter the TMNR or amount of interference pulses compared to the baseline. The other tracking algorithms all experience less track loss with a dwell & switch PRI scheme as compared to the baseline results. This could be because they track each PRI in the PRI sequence separately or because the Fourier domain Kalman filter is best suited to staggered PRI schemes with high level values[2]. The track loss of all algorithms increases with jittered PRI schemes compared to the baseline results. This is because of the extra noise from the added jitter decreases TMNR and track loss is inversely proportional to TMNR. In the presence of jitter, the Kalman filter has the lowest tracking error meaning it rejects noise better than the other algorithms. With staggered PRI signals, the change in performance is similar to the dwell and switch case but the Fourier domain Kalman filter does not fail as much.

The two algorithms that took the least amount of time to execute, due to their lower complexity, were chosen to be implemented on the DRFM. The execution time of trackers is significant as there will be multiple trackers running at the same time in the system, in conjunction with a deinterleaver. The algorithms chosen were the alpha-beta filter and Delta-τ histogram. In many situations the alpha-beta filter and Delta-τ histogram performed on par with the time domain Kalman filter such as in the presence of interference pulses and mid

[2] A dwell & switch PRI scheme is similar to a staggered PRI scheme with a low level value.

to high TMNR. The performance gain when using a time domain Kalman filter is not justified due to its complexity.

4.3 Tracking Hardware Results

The alpha-beta filter and Delta-τ histogram algorithms were implemented on the DRFM and the EW environments in which they were tested were implemented in MATLAB, similarly as in section 3.3. The number system had to be changed from double-precision floating-point numbers to unsigned 32-bit integers here as well, resulting in the same introduction in uncertainties as discussed in section 3.3. The number system conversion resulted in a $\pm 0.5\%$ uncertainty in TOA measurements, which resulted in the tracking gate size change between -0.75% and $+1.4\%$.

To implement the alpha-beta tracker, a first difference histogram was created in MATLAB so that the estimates of the gate size, noise variance due to the EW receiver (σ_v^2) and the noise variance added by the transmitting radar (σ_w^2) can be made. The estimate for the noise variance due to the EW receiver (σ_v^2) increases between 2.4% and 3% due to changing number systems, while the estimate for the noise variance added by the transmitting radar (σ_w^2) increased between 21.7% and 27.1%. The increases in noise variance estimates in turn affects the calculation of the alpha and beta coefficients. The alpha coefficient is reduced by 114%, while the beta coefficient is reduced by 92%.

Since the tracker is implemented separately from the from the EW environment and there is no deinterleaving method to initialise a new track, regaining track after it was lost could not be implemented like it was in the simulations. Therefore, instead of measuring mean track loss, the amount of pulses processed before track is lost was measured.

After track is lost the algorithm stops and moves onto the next simulated EW environment. As the algorithm will not always process all pulses the EW environment before the track is lost and the amount of pulses it does process can vary. The average normalised tracking error versus the number of pulses will also not be measured as the algorithms may not track all the emitter pulses. If the average normalised tracking error values are obtained, they will be noisy for some pulses because they could not be averaged properly, for example, if only one simulation run makes it to the n^{th} pulse out of the one thousand runs then there will only be one value to be averaged from that one simulation run. This result could be an outlier, such that if the algorithm always made it to the n^{th} pulse the average result could be totally different. This can be called a noisy result. The reason why one thousand different EW environments are simulated for any given characteristics is to average out the results and reduce “noise” in the results.

Execution time is also not measured here as the UDP communication added to the time. It was noticed that randomly the MicroBlaze would reboot. This was a firmware problem on the CSIR 5^{th} generation DRFM platform. Since testing certain emitters on the DRFM could take days, this had to be accounted for on the MATLAB code. Whenever the UDP connection timed out, the tracker had to be set up again and the pulses from the EW environment had to be sent to the DRFM from the beginning again. This affected the time measurements as well.

The algorithms implemented on the DRFM were tested against the different PRI schemes of interest as in the simulations section. The conclusions from the simulations section will not be discussed here. The performance of the algorithms in hardware versus simulations will be discussed, if applicable. Their performance versus each other in hardware and their performance versus their performance processing the constant PRI scheme emitter in hardware will also be discussed. A snapshot of the hardware results for the tracking investigated for all the different PRI schemes can be seen in Tables 4.18 to 4.25. The complete emitter TOA tracking hardware results can be found in Appendix F.

4.3.1 Constant PRI Type Signals

A constant PRI scheme emitter was tested against the algorithms first. The emitter had a PRI of 50 μs. The results of each algorithm were measured using percentage of pulses processed before track is lost and average standard deviation of normalised tracking error.

4.3.1.1 Percentage of Pulses Processed Before Track is Lost

Every time track is lost, the number of pulses it had processed before the track was lost is averaged over the one thousand runs. This number is normalised to the number of pulses the EW receiver receives from the emitter i.e. transmitted pulses - missing pulses and expressed as a percentage. Table 4.18 shows the percent of pulses processed before the track is lost by the algorithms with varying interference pulses and TMNR. The higher the percentage the better. One hundred percent means the track was not lost.

	Percentage of Pulses Processed Before Track Is Lost (%)											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Delta-τ Histogram	35.3	12.6	100	21.4	12.4	11.7	52	80.2	100	39.7	36.2	37.3
Alpha-Beta Filter	57.2	15.6	100	55.7	35.5	28.6	64.5	70.6	86.1	71.3	78.7	76.8

Table 4.18: Hardware Results: Percentage of Pulses Processed Before Track is Lost - Constant PRI

The values in this table cannot be compared to simulation results. The type of change in performance for both algorithms is the same in all situations. Spurious pulses increase the performance of the algorithms, performance was influenced by the initial estimates made with the spurious pulses. Missing pulses degrade performance at low and high TMNR but increases performance at mid TMNR. Missing pulses should degrade performance, and it does at low and high TMNR. In the case of mid TMNR, the loss in resolution by changing number schemes is to blame for the increase in performance. The loss of resolution means an effective wider tracking gate.

In the hardware implementation, the track could not be regained after it was lost like it was in simulations. In hardware results, missing pulses meant that track was lost sooner in some cases than in the cases without missing pulses. In simulations missing pulses meant the track was lost less often each run. In simulations track could have been lost earlier with missing pulses but was not measured, the characteristic used to measure the performance is different in cases of hardware testing and simulations.

The alpha-beta filter performance better or the same as the Delta-τ histogram in ten out of twelve situations.

4.3.1.2 Average Standard Deviation of Normalised Tracking Error

Table 4.19 shows the average standard deviation of normalised tracking error for the algorithms in the hardware implementation with varying interference pulses and TMNR. Lower is better.

	Average Standard Deviation of Normalised Tracking Error ($\times 10^{-3}$)											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Delta-τ Histogram	12.4	6	0.1	28.8	5.3	0.4	13.4	2.1	0.1	22.9	2.8	1
Alpha-Beta Filter	11.5	13.7	117.4	40.2	29.8	14.2	20.6	13.1	3.3	33.6	20.7	15.1

Table 4.19: Hardware Results: Standard Deviation of Normalised Tracking Error - Constant PRI

The average standard deviation of normalised tracking error for both algorithms is higher than on simulations. One reason for this is because for the average standard deviation of normalised tracking error obtained in simulations the algorithm is processing all the pulses in the EW environment from the emitter it is tracking, and in hardware, it stops when the track is lost. The standard deviation is calculated from the first pulse to the last pulse, and it was concluded that tracking error gets smaller as the number of pulses increases. Therefore since these algorithms does not always reach the last pulse transmitted, the lowest tracking error is greater than the lowest tracking error of the simulations. The standard deviation is larger because there are more large tracking error values.

Another reason the average standard deviation of normalised tracking error is bigger could be due to loss of resolution by changing the number system. Predictions and measurements are made with less resolution or more uncertainty. Therefore when calculating tracking error and mean tracking error in order to calculate standard deviation, the loss of resolution compounds.

It should be noted that the largest value in the table of 117.4×10^{-3} equates to about 0.1 standard deviation of normalised tracking error. The tracking error in this scenario is therefore $\pm 5\mu s$. While jamming in this scenario, the false target maybe projected to the target radar with a $\pm 750m$ uncertainty in range (equation 2.1) each pulse. The target radar should discard the false target, as typical tracking windows are between $15m - 40m$. In a scenario where the PRI is less than $2.67\mu s$ (using equation 2.1) the 0.1 standard deviation of normalised tracking error, will allow for successful jamming.

The alpha-beta filter out performs the Delta-τ histogram in one situation.

4.3.2 Dwell and Switch PRI Type Signals

The second PRI scheme tested against the algorithms was a dwell and switch PRI scheme. The dwell and switch PRI sequence of the emitter was 50 µs, 50 µs, 50 µs, 30 µs, 30 µs, 30 µs. The results of each algorithm were measured using percentage of pulses processed before track is lost and average standard deviation of normalised tracking error.

4.3.2.1 Percentage of Pulses Processed Before Track is Lost

Table 4.20 shows the percent of pulses processed before track is lost by the algorithms with varying interference pulses and TMNR.

	Percentage of Pulses Processed Before Track Is Lost (%)											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Delta-τ Histogram	29.2	11.8	100	16.1	8.4	8.7	57.8	96.2	100	35.2	19.7	19.1
Alpha-Beta Filter	62.6	11.3	100	41.1	34.5	30.4	85.7	97.4	99.5	71.7	72.9	71.4

Table 4.20: Hardware Results: Percentage of Pulses Processed Before Track is Lost - Dwell and Switch PRI

As seen constant PRI type signals, missing pulses increases performance at mid TMNR and decreases performance at low and high TMNR. Spurious pulses also increases the performance of the algorithms.

When compared to the results of the algorithms processing a constant PRI scheme in hardware, missing pulses have more of a negative impact on performance. This is because missing pulses in a dwell and switch scheme means the missing PRI in the sequence will only be updated the next time the PRI sequence repeats. In a constant PRI scheme, it is updated every newly received pulse. The performance of the Delta-τ histogram has decreased as compared to the performance with a constant PRI scheme. The alpha-beta filter performance has increased at lower TMNR and decreased at mid TMNR. The greater range of PRI values alters the initial estimates.

The alpha-beta filter performs the same or better than the Delta-τ histogram in ten out of twelve situations.

4.3.2.2 Average Standard Deviation of Normalised Tracking Error

Table 4.21 shows the average standard deviation of normalised tracking error for the algorithms in the hardware implementation with varying interference pulses and TMNR.

	Average Standard Deviation of Normalised Tracking Error ($\times 10^{-3}$)											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Delta-τ Histogram	13.3	5.8	8.2	30	28.2	3.3	11.7	2	0.8	24.1	37.7	10.1
Alpha-Beta Filter	10.8	12.5	15.4	49	35.4	16.2	8.4	11.4	2.3	30.6	23.5	17.7

Table 4.21: Hardware Results: Standard Deviation of Normalised Tracking Error - Dwell and Switch PRI

The average standard deviation of normalised tracking error in hardware is greater than the simulations results for these algorithms due to the reason explained in the discussion related to Table 4.19.

The alpha-beta filter implemented on the DRFM performs better with the dwell and switch PRI scheme than with a constant PRI scheme. The same cannot be said for the Delta-τ histogram. The alpha-beta filter out performs the Delta-τ histogram in two out of twelve situations.

4.3.3 Jittered PRI Type Signals

The third PRI scheme tested against the algorithms was a jittered PRI scheme. The jittered PRI of the emitter was 50 μs with 10% jitter. The results of each algorithm were measured using percentage of pulses processed before track is lost and average standard deviation of normalised tracking error.

4.3.3.1 Percentage of Pulses Processed Before Track is Lost

Table 4.22 shows the percent of pulses processed before track is lost by the algorithms with varying interference pulses and TMNR.

	Percentage of Pulses Processed Before Track Is Lost (%)											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Delta-τ Histogram	37.7	16.5	16.6	22	7.1	6.5	50.7	33.3	33.8	42.2	20.2	21.3
Alpha-Beta Filter	56.4	30.9	32.3	52.3	45	47	75.3	67.3	66.8	72.9	81.3	83.1

Table 4.22: Hardware Results: Percentage of Pulses Processed Before Track is Lost - Jittered PRI

The same conclusions about the effect of missing and spurious pulses from the discussion related to Table 4.18 can be drawn here. There is one exception though; missing pulses also increases the performance of the alpha-beta filter at high TMNR. This can be attributed to the extra intentional noise in the jittered PRI scheme. The additional noise negatively affects the Delta-τ histogram more than the alpha-beta filter when compared to their performance with a constant PRI scheme. The alpha-beta filter outperforms the Delta-τ histogram in all situations. Therefore the alpha-beta filter can process more pulses before losing track in low TMNR than the Delta-τ histogram.

4.3.3.2 Average Standard Deviation of Normalised Tracking Error

Table 4.23 shows the average standard deviation of normalised tracking error for the algorithms in the hardware implementation with varying interference pulses and TMNR.

	Average Standard Deviation of Normalised Tracking Error ($\times 10^{-3}$)											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Delta-τ Histogram	14	14.1	14.3	28	48.7	43.9	13.5	20.2	18.7	21.4	48.9	43.7
Alpha-Beta Filter	12.4	14.1	16	42.4	42.9	43.5	11.1	16.1	14.7	32.1	23.3	23.9

Table 4.23: Hardware Results: Standard Deviation of Normalised Tracking Error - Jittered PRI

The average standard deviation of normalised tracking error in hardware is greater than the simulations results for these algorithms due to the reasons explained in the discussion related to Table 4.19. The standard deviation of normalised tracking error of the algorithms here versus the constant PRI scheme in hardware is slightly worse. However, the standard deviation of normalised tracking error of the alpha-beta filter for high TMNR is much better here. The alpha-beta filter performs the same or better than the Delta-τ histogram in ten out of twelve situations.

4.3.4 Staggered PRI Type Signals

Finally, a staggered PRI scheme was tested against the algorithms. The staggered PRI of the emitter was 50 μs, 30 μs, 80 μs. The results of each algorithm were measured using percentage of pulses processed before track is lost and average standard deviation of normalised tracking error.

4.3.4.1 Percentage of Pulses Processed Before Track is Lost

Table 4.24 shows the percent of pulses processed before track is lost by the algorithms with varying interference pulses and TMNR.

	Percentage of Pulses Processed Before Track Is Lost (%)											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Delta-τ Histogram	45.9	78.2	100	20	7.8	7.7	63.5	98.1	100	38.4	18.7	17.1
Alpha-Beta Filter	78.7	96.9	100	68.1	43	43.8	88.6	98.8	99.3	83.1	75.6	73.7

Table 4.24: Hardware Results: Percentage of Pulses Processed Before Track is Lost - Staggered PRI

TMNR increases the performance of both algorithms. Missing pulses decreases the performance of both algorithms for all TMNR and is more apparent in mid and high TMNR. Spurious pulses increase the performance of both algorithms, but in the case of high TMNR of the alpha-beta, there was a slight drop in performance. It does not seem like a big drop to be concerned.

The performance of the algorithms shown here is better than the performance, on the DRFM, of the algorithms with a constant PRI scheme. The alpha-beta filter performs the same or better in eleven out of twelve situations.

4.3.4.2 Average Standard Deviation of Normalised Tracking Error

Table 4.25 shows the average standard deviation of normalised tracking error for the algorithms in the hardware implementation with varying interference pulses and TMNR.

	Average Standard Deviation of Normalised Tracking Error ($\times 10^{-3}$)											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Delta-τ Histogram	17.8	28	24.4	36.4	30.4	2.5	14.7	6.4	1.1	27.5	38	8.6
Alpha-Beta Filter	13	13.3	98	23.4	42.8	24.6	10.6	11.9	4.8	18.5	26.3	18.4

Table 4.25: Hardware Results: Standard Deviation of Normalised Tracking Error - Staggered PRI

The average standard deviation of normalised tracking error increases with TMNR. Missing pulses increase the standard deviation of tracking error for both algorithms except at high TMNR for the alpha-beta filter, where it is decreased. Spurious pulses increase performance for both algorithms.

The average standard deviation of normalised tracking error for both algorithms are more here compared to simulations. Compared to the constant PRI scheme in hardware, performance of the alpha-beta filter and Delta-τ histogram has decreased. The alpha-beta filter performs the same or better than the Delta-τ histogram in six out of twelve situations.

4.3.5 Conclusion

The Delta-τ histogram and alpha-beta filter algorithms were implemented on the DRFM platform to investigate any difference between simulation and hardware performance. The algorithm that behaves the best will be implemented in the system outlined in Chapter 1.

Unfortunately when the algorithms were implemented on the DRFM platform, when track was lost it could not be regained. Therefore, mean track loss had to be changed to the percentage of pulses processed before track is lost. This also meant that average normalised tracking error versus pulse number was no longer feasible. The execution time of the algorithms was not measured because the MicroBlaze could randomly reboot due to a firmware issue on the DRFM and the UDP communication added more time to the execution time.

Changing of the number system from double-precision floating-point numbers to unsigned 32-bit integers caused a loss in precision. This affected tracking gate estimates and in the case of the alpha-beta filter, estimates of the noise variances. These noise variances are used to calculate the alpha-beta coefficients. The loss in precision also affected the calculation of tracking error.

The standard deviation of normalised tracking error is calculated from the first pulse to the last pulse, and it was concluded that tracking error gets smaller as the number of pulses increases. Therefore since these algorithms did not always reach the last pulse transmitted, the lowest tracking error is greater than the lowest tracking error of the simulations. The calculated standard deviation is larger for the hardware implementation because there are more large tracking error values due to the change in the number system.

While analysing the results of the algorithms with a jittered PRI scheme emitter, it was concluded that the alpha-beta filter could process more pulses before losing track in low TMNR than the Delta-τ histogram. In fact for all the PRI schemes the algorithms were tested in, the alpha-beta filter could process more pulses before losing track the Delta-τ histogram. The standard deviation of normalised tracking error performance of the Delta-τ histogram was better for constant and dwell and switch PRI schemes. However, the worse case scenario of the alpha-beta filter was about 0.1 standard deviation of the PRI.

Mainly due to the alpha-beta being able to process more pulses before track is lost and that in simulations it took much less time to execute, it was chosen to be implemented in that would deinterleave and track radar emitters in an EW environment.

Chapter 5

Implementation and Integration

After the best suited deinterleaving and tracking algorithms have been selected, the solution system defined in Chapter 1 can be implemented on the CSIR 5^{th} generation DRFM. The system shall deinterleave and then track radar emitters in an EW environment. The system boundary and requirements were specified in Chapter 1. Using the system functional design (see Figure 1.2) the flowchart of the system was conceptualised.

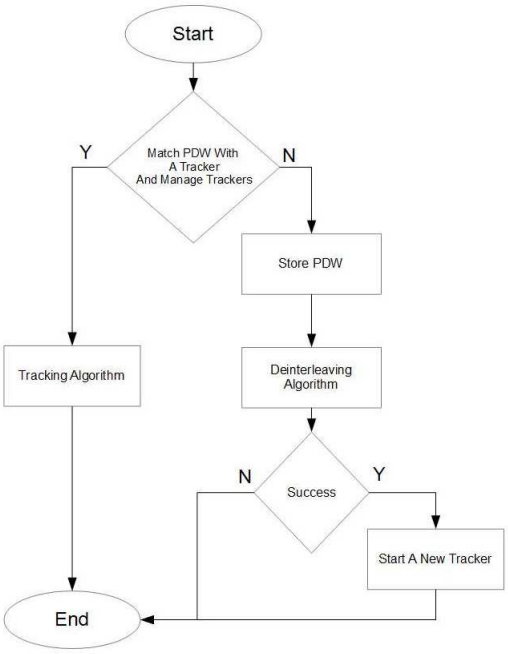


Figure 5.1: System Flowchart

Figure 5.1 is a flowchart of the system. When the TOA of a new pulse is input to the system, the system will first decide if the pulse belongs to any of the emitters that it is currently being tracked by comparing the TOA with all of the predicted TOAs. If the received TOA is received after the predicted TOA and associated tracking window of a tracker, that tracker is coasted and a missing pulse is assumed. If a tracker has fifteen consecutive missing pulses, the tracker is discarded. If the TOA belongs to a tracker, that tracker is updated using the

alpha-beta filter selected as the tracking algorithm in Chapter 4. A new predicted TOA is calculated before the system starts to wait for the next pulse.

If the TOA does not belong to any tracker, it is stored in a FIFO ring buffer. The CDIF SS algorithm, selected as the deinterleaving algorithm in Chapter 3, is then run with the TOA values stored in the buffer. If no emitter was found the system does nothing and starts to wait for the next pulse. If an emitter is found, then the associated pulses are removed from the buffer a new tracker is started. The new tracker is initialised using the found PRI and one and a half times the bin size of the histogram when the emitter was found as the tracking gate estimate. The gate estimate is used to determine the noise variance estimates, which are needed to calculate the filter coefficients. The alpha and beta coefficients of the new tracker and a predicted TOA is calculated before the system starts to wait for the next pulse.

It should be noted that the CDIF SS algorithm extracts constant PRI pulse trains from an interleaved pulse train. Therefore constant PRI pulse trains will be tracked as normal by the system. Jittered PRI pulse trains will be tracked by the system as constant PRI pulse train with a larger noise estimate. In cases of staggered, and by extension dwell and switch PRI schemes the system will initialise constant PRI trackers equal in number to the level[1] of the PRI scheme [10].

5.1 Hardware Results

After the system had been implemented on the DRFM, it was tested in EW environments simulated in MATLAB as previously done in sections 3.3 and 4.3. First, the system was tested in EW environments containing one emitter followed by EW environments containing multiple emitters. The performance of the system will be measured using the mean number of trackers that it initializes, the average percentage of pulses correctly predicted and the mean number of pulses processed before a tracker was initialised.

Verification of the system will be done in this section using the requirements specified in section 1.2.2. The mean number of trackers initialised is the mean number of trackers that were used to process the thousand EW environments of constant characteristics. The number of trackers increases if track is lost and the tracker for an emitter has to be reinitialised. If there is a false emitter detection, then the number of trackers will also increase. Ideally, this number should be equal to the number of emitters in the EW environment. The lower the number of trackers initialised the better the performance.

The mean number of pulses processed before a tracker was initialised is the mean number of pulses that the system processes before it identifies an emitter in the EW environment and initialises a tracker to track that emitter. It is not the number of pulses processed to fully deinterleave the EW environment, which was measured in Chapter 3. It includes spurious pulses that the system has to process and excludes missing pulses that the system does not receive. This number also includes the zeroth pulse $(TOA_0 = 0s)$, explained in section 3.1, meaning the value in the table is one less. The lower the mean number of pulses processed before a tracker is initialised the better performance. In the case of more than one emitter, the fewer number of pulses processed before a tracker is started for more emitters is the better performance case.

[1] Level is the number of unique PRIs in a PRI sequence, see section 2.3.

The average percentage of pulses correctly predicted is the number of pulses that a tracker predicted for an emitter that belonged to the actual emitter. Since the system does not check if more than one pulse falls in a tracking window, the first pulse to fall tracker's tracking window is assumed to belong to the tracker. If a spurious pulse or a pulse from another emitter is associated with the wrong emitter it is counted as an incorrect prediction, decreasing percentage of pulses correctly predicted. Also, the total number of pulses used to calculate the percentage is the number of pulses that the emitter transmits that the EW receiver detects, meaning the total number = pulse number - missing pulses. The total number includes the number of pulses received before a tracker was started, so a higher "mean number of pulses processed before a tracker was initialised" negatively affects the average percentage of pulses correctly predicted.

It is important to remember that the CDIF SS algorithm is now used to make initial estimates for the alpha-beta filter.

5.1.1 Single Emitter

First, the system was tested in EW environments with one radar emitter. The emitter transmitted 75 pulses. The emitter PRI scheme was changed to all the PRI schemes of interest in this study. A snapshot of system results with a single emitter can be seen in Tables 5.1 to 5.4 in this section. However, the complete results can be found in Appendix G.

5.1.1.1 Constant PRI Type Signals

The first PRI scheme that the single emitter EW environment had was a constant PRI of 50 μs. Table 5.1 shows the performance of the system with varying interference pulses and TMNR.

	Single Signal Results - Constant PRI											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Mean Number of Trackers Initialised	3.1	2.6	1	2.8	2	1	3.3	2.8	1.8	3.1	2.6	2
Average Percentage Pulses Correctly Predicted	77	82.9	96	76.9	84.8	94.5	78.9	82.5	86.9	78.4	81.2	85
Mean Number of Pulses Processed Before Tracker is Initialised	11.1	7.6	6	9.9	7	6.5	10.8	7.3	6.5	9.5	7.1	7

Table 5.1: Single Signal Results - Constant PRI

From the table, it can be seen that as TMNR increases, the number of trackers decreases. This means there are less false detections or track losses. The number of pulses before track is initialised also decreases with an increasing TMNR. The lower the number of pulses needed to start tracking increased the total number of correctly predicted pulses. It can be seen that an increase in spurious pulses decreases the percentage of pulses correctly predicted because sometimes spurious pulses are associated with the tracking window before the actual transmitted pulse. Spurious pulses can also cause false emitter detections as seen in the increase in the number of trackers initialised. If a spurious pulse is positioned before an emitter is found, the number of pulses required before a tracker is initialised is increased.

Missing pulses only have an effect on the mean number of pulses processed before a tracker is initialised if it lies before a tracker is initialised. When this happens, due to the missing pulse the difference value added to the CDIF SS algorithm will be bigger than it was originally. For example for four pulses with a constant

PRI, if one of the pulses (pulse 2 or pulse 3) is missing then the resulting differences will be two counts of the PRI and one count of two times the PRI versus three counts of the PRI if no pulses were missing. By taking longer to start tracking an emitter the percentage of correctly predicted pulses decreases as well. The number of trackers initialised decreases at lower TMNR, this could be due to the tracker being coasted. When the tracker is coasted, the tracking window is increased.

From the requirements in section 1.2.2, the number of pulses required to start tracking a constant emitter is 4 transmitted pulses. The mean number of transmitted pulses before tracking is started at high TMNR is 5 (because number includes $TOA_0 = 0$). This is above the requirement but since it is the mean , there may be some cases where the system took 4 pulses. In the case of one emitter with only one constant PRI, an initial ten bin CDIF histogram is created with only pulses from that one emitter with one PRI. This means pulses for the same PRI are split over many bins meaning the probability of pulses being grouped together in the same bin and exceeding the threshold is lower. More pulses are therefore required to detect an emitter in this case as pulses with the smallest variations need to be grouped together in one bin. It was proven that the CDIF algorithm could detect emitters faster than CDIF SS but with a higher false detection rate in section 3.3. However, it would experience the same problem outlined here. Therefore, there may not be any of the required performance gains for a single emitter EW environment with a constant PRI scheme. A possible solution is to use less than 10 bins when the measured PRIs has a small range $(range = TOA_{Newest} - TOA_{Oldest})$.

5.1.1.2 Dwell and Switch PRI Type Signals

The second PRI scheme that the single emitter EW environment had was a dwell and switch PRI of 50 μs, 50 μs, 50 μs, 30 μs, 30 μs, 30 μs. Table 5.2 shows the performance of the system with varying interference pulses and TMNR.

	Single Signal Results - Dwell & Switch PRI											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Mean Number of Trackers Initialised	3.4	3.6	3.4	3.2	3.5	3.6	3.7	3.8	4.1	3.5	3.7	4
Average Percentage Pulses Correctly Predicted	76.3	76.4	79	74.7	74.1	74.6	78	77.5	77.3	76.2	75.4	75.4
Mean Number of Pulses Processed Before Tracker is Initialised	10.6	7.4	5.9	9.8	7.4	6.9	10.2	7.3	5.9	9.6	7.5	6.8

Table 5.2: Single Signal Results - Dwell and Switch PRI

The same conclusions of the effect of TMNR and interference pulses from section 5.1.1.1 can be found here. From the requirements, the number of pulses needed to start tracking a dwell and switch PRI sequence was by two dwell periods. This is equal to 12 pulses as the stagger sequence has a period of six. As only 3.4 trackers are initialised, it is safe to say that the tracker initialised after 5.9 pulses was not a false detection. This is probably a tracker with PRI 50 μs and equates to the 1^{st} dwell period. The system meets this requirement.

5.1.1.3 Jittered PRI Type Signals

The third PRI scheme that the single emitter EW environment had was a jittered PRI of 50 μs with ten percent jitter. Table 5.3 shows the performance of the system with varying interference pulses and TMNR.

	Single Signal Results - Jittered PRI											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Mean Number of Trackers Initialised	3.2	2.7	2.7	2.8	2.5	2.4	3.4	3	3	3.2	2.9	2.9
Average Percentage Pulses Correctly Predicted	76.8	79	79.5	76.7	79.3	79.5	79.6	79.5	79.9	78.1	79.2	79.4
Mean Number of Pulses Processed Before Tracker is Initialised	11.2	12.1	11.7	9.9	8.7	8.7	9.7	10.1	9.7	9.6	8.6	8.4

Table 5.3: Single Signal Results - Jittered PRI

The same conclusions of the effect of TMNR and interference pulses from section 5.1.1.1 can be found here, even with the extra noise from the added jitter. From the requirements, the system needs to be able to track a jittered PRI sequence with 10% jitter. As the system correctly predicts 76.8% of pulses in the worst case scenario (TMNR of 8 dB and no interference pulses) and in that scenario it takes about 13.8% $\left(\frac{(11.2-1)\times 100}{74}\right)$ of the pulses to initialize track in that scenario, therefore the system does not track 9.4% of pulses (7 pulses). The system meets this tracking requirement, as this is the worst case scenario. The next requirement is that the system starts tracking a jittered PRI sequence by 4 received pulses. Unfortunately, due to the extra noise in the signal and the problem relating to binning a single PRI into 10 bins discussed in section 5.1.1.1, the system takes about 7.4 pulses in the best case scenario. The system does not meet this requirement for a single jitter PRI scheme emitter EW environment.

5.1.1.4 Staggered PRI Type Signals

The final PRI scheme that the single emitter EW environment had was a staggered PRI of 50 µs, 30 µs, 80 µs. Table 5.4 shows the performance of the system with varying interference pulses and TMNR.

	Single Signal Results - Staggered PRI											
Missing (M) & Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Mean Number of Trackers Initialised	3.9	3.5	2	3.5	2.9	2.3	3.9	3.4	3	3.7	3.3	3.1
Average Percentage Pulses Correctly Predicted	74.5	76	84.9	73.6	78.4	84.5	78	77.6	79.2	76.6	76.7	77.7
Mean Number of Pulses Processed Before Tracker is Initialised	10	9	9.4	9.7	8.9	9	9.6	9	9.2	9.4	9	9

Table 5.4: Single Signal Results - Staggered PRI

The same conclusions of the effect of TMNR and interference pulses from section 5.1.1.1 can be found here. In the case of no interference pulses and high TMNR, the number of trackers initialised is less than the level of three, which is unexpected. It correctly predict 84.9% of pulses and took 11.4% $\left(\frac{(9.4-1)\times 100}{74}\right)$ of pulses to start tracking. This leaves 3.7% of pulses (2.7 pulses) unaccounted for because they were not tracked. This is acceptable because the system managed to found only two trackers such that it could correctly predicted an average of 62.8 pulses (0.849×74) out of the remaining 65.5 pulses transmitted by the emitter after the track was intialised. This is a more optimum tracking solution than using three trackers, and it is not the only case where the average number of trackers initialised was less than the level of three. If the system were to run longer, more pulses will not be tracked and added to the buffer. Eventually, the system would then initialize a third tracker to track these outliers.

From the requirements, the system must start tracking a staggered PRI sequence by 4 received stagger PRI sequence repetitions. This is equal to 12 pulses as the stagger sequence has a period of 3. The system takes about 7.9 pulses to start tracking the staggered PRI sequence in the best case scenario. The system meets this requirement.

5.1.2 Multiple Emitters

After the system had been tested in EW environments with one radar emitter, EW environments with multiple emitters were introduced to the system. Each emitter transmitted 75 pulses. A snapshot of system results with multiple emitters can be seen in Tables 5.5 to 5.12 in this section. However, the complete results can be found in Appendix H.

5.1.2.1 Constant PRI Signals

First an EW environment with only constant PRI scheme emitters was tested against the system. The PRIs for each emitter were as follows:

Emitter 1: 50 μs, Emitter 2: 75 μs, Emitter 3: 31 μs, Emitter 4: 88 μs

In the case of two emitters being interleaved, the pulses from emitters 1 and 2 are present in the interleaved pulse train. In the case three emitters are interleaved, the pulses from emitters 1 to 3 are present in the interleaved pulse train. Finally, in the case of 4 interleaved emitters, the pulses from emitters 1 to 4 are present in the interleaved pulse train. Table 5.5 shows the performance of the system with varying emitters and TMNR.

	No missing or spurious pulses								
Emitters	**2**			**3**			**4**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Mean Number of Trackers Initialised	4.3	4.1	2.2	4.9	4.9	4.8	5.3	5.2	5.5
Average Percentage Pulses Correctly Predicted	86.2	86.4	93.3	90.1	89.5	90.4	92.2	91.9	91.2
Mean Number of Pulses Processed Before Tracker is Initialised	9	8	7.8	8.6	9.4	9	8.3	9	7

Table 5.5: Interleaved Signal Results - Constant PRI - Emitters

As TMNR increases, the number of trackers initialised and the number of pulses before a tracker is initialised decreases. The number of pulses required to initialize a tracker stays relatively the same for up to 4 emitters, this likely because emitter 3 has three pulses in the first 5 (3 emitters case) or 6 (4 emitters case) transmitted pulses. Therefore the PRI values of the emitters in the EW environment determines the number of pulses required to initialise a tracker to some extent.

As there are more emitters in the environment, there needs to be more trackers initialised to track them.

For the one of worst cases, i.e. more pulses before a tracker is started and fewer emitters, the received interleaved constant pulse train that the system processes in the case of 2 emitters is (without noise and jitter) $0, 50\mu s, 75\mu s, 100\mu s, 150\mu s, 150\mu s, 200\mu s, 225\mu s$. The tracker is initialised with a mean of 7.8 pulses. Therefore pulses 7 and 8 are of interest. Note that the first pulse in this pulse train is $TOA_0 = 0$, which is not transmitted by any emitter. The 7^{th} pulse in the pulse train corresponds to $200\mu s$, which is the fourth pulse of emitter one. The 8^{th} pulse in the pulse train corresponds to $225\mu s$, which is the third pulse of emitter two. From this, we can say that the requirement that the system start tracking a constant emitter is 4 transmitted pulses is met.

A requirement of the system is to deinterleave and track at least 4 radar emitters. As the system tracks at least 91.9% of pulses in the case of 4 emitters, these requirements are met.

Table 5.6 shows the performance of the system with varying interference pulses and TMNR.

	2 Emitters											
Missing (M) and Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Mean Number of Trackers Initialised	4.3	4.1	2.2	4	3.6	2.7	4.6	4.2	3.9	4.4	4.2	4
Average Percentage Pulses Correctly Predicted	86.2	86.4	93.3	85.3	87	91	87.7	87.6	88.6	86.7	86.8	87.6
Mean Number of Pulses Processed Before Tracker is Initialised	9	8	7.8	9	7.8	7.4	8.6	8.1	7.9	8.9	8	7.6

Table 5.6: Interleaved Signal Results - Constant PRI - Interference Pulses

The same conclusions about varying TMNR and interference pulses from section 5.1.1.1 can be reached here as well.

5.1.2.2 Jittered PRI Signals

Next an EW environment with only jittered PRI scheme emitters was tested against the system. The PRIs for each emitter were as follows:

Emitter 1: 50 μs (10% jitter), Emitter 2: 75 μs (10% jitter), Emitter 3: 31 μs (10% jitter), Emitter 4: 88 μs (10% jitter)

In the case of two emitters being interleaved, the pulses from emitters 1 and 2 are present in the interleaved pulse train. In the case three emitters are interleaved, the pulses from emitters 1 to 3 are present in the interleaved pulse train. Finally, in the case of 4 interleaved emitters, the pulses from emitters 1 to 4 are present in the interleaved pulse train. Table 5.7 shows the performance of the system with varying emitters and TMNR.

	No missing or spurious pulses								
Emitters	**2**			**3**			**4**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Mean Number of Trackers Initialised	4.2	4.2	4.2	5	4.9	4.9	5.3	5.3	5.3
Average Percentage Pulses Correctly Predicted	86.4	85.8	85.9	90	89.8	89.8	92.2	92.1	92.1
Mean Number of Pulses Processed Before Tracker is Initialised	8.8	9.3	9.3	8.4	9.3	9.4	8.4	8.9	9

Table 5.7: Interleaved Signal Results - Jittered PRI - Emitters

The same conclusions drawn on the affect emitters have on performance from section 5.1.2.1 apply here as well. With the extra added noise in the jittered signals, the results stay relatively the same as those in section 5.1.2.1.

For the one of worst cases, i.e. more pulses before a tracker is started and fewer emitters, the received interleaved constant pulse train that the system processes in the case of 2 emitters is (without noise and jitter) $0, 50\mu s, 75\mu s, 100\mu s, 150\mu s, 150\mu s, 200\mu s, 225\mu s, 250\mu s$. The tracker is initialised with a mean of 8.8 pulses. Therefore pulses 8 and 9 are of interest. The 8^{th} pulse corresponds to $225\mu s$, which is the third pulse of emitter two. The 9^{th} pulse corresponds to $250\mu s$, which is the fifth pulse of emitter one. Therefore the requirement that the system start tracking a jittered PRI sequence by 4 received pulses is met for two emitters but not met for emitter one, where track could be started by the fifth. If we examine the case of 4 emitters (without noise and jitter), then the received pulse train is $0, 31\mu s, 50\mu s, 62\mu s, 75\mu s, 88\mu s, 93\mu s, 100\mu s, 124\mu s$. The pulse of interest is the 9^{th} pulse, which corresponds to $124\mu s$, which is the fourth pulse of emitter 3. Therefore it can be safely concluded that the system meets the requirements.

Table 5.8 shows the performance of the system with varying interference pulses and TMNR.

	2 Emitters											
Missing (M) and Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Mean Number of Trackers Initialised	4.2	4.2	4.2	4.1	3.9	3.9	4.5	4.5	4.4	4.4	4.3	4.3
Average Percentage Pulses Correctly Predicted	86.4	85.8	85.9	85.2	85.4	85.4	87.8	87	87.2	86.9	86.3	86.6
Mean Number of Pulses Processed Before Tracker is Initialised	8.8	9.3	9.3	8.8	9	9	8.8	9.2	9.3	8.6	9.1	9.1

Table 5.8: Interleaved Signal Results - Jittered PRI - Interference

The same conclusions about varying TMNR and interference pulses from section 5.1.1.1 can be reached here as well. Again with the added noise in the form of jitter, the results does not differ much from the constant PRI signals performance seen in section 5.1.2.1.

From the tables in this section, we can see the system performs almost the same in an EW environment with emitters that have added jitter versus an EW environment with emitters with no jitter.

5.1.2.3 Mixed PRI Signals

Next an EW environment with mixed PRI scheme emitters was tested against the system. The PRIs for each emitter were as follows:

Emitter 1: 50 μs (10% jitter), Emitter 2: 75 μs, Emitter 3: 31 μs (10% jitter), Emitter 4: 88 μs

In the case of two emitters being interleaved, the pulses from emitters 1 and 2 are present in the interleaved pulse train. In the case three emitters are interleaved, the pulses from emitters 1 to 3 are present in the interleaved pulse train. Finally, in the case of 4 interleaved emitters, the pulses from emitters 1 to 4 are present in the interleaved pulse train. Table 5.9 shows the performance of the system with varying emitters and TMNR.

	No missing or spurious pulses								
Emitters	**2**			**3**			**4**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Number of Trackers Initialised	4.3	4.1	3.8	4.9	4.9	4.8	5.3	5.3	5.2
Percentage Pulses Correctly Predicted	86.3	86.1	87.2	90	89.8	89.8	92.2	92	92
Mean Number of Pulses Processed Before Tracker is Initialised	8.8	9	9.4	8.6	9.4	9.8	8.3	8.9	9

Table 5.9: Interleaved Signal Results - Mixed PRI - Emitters

The same conclusions of the effect emitters have on performance from section 5.1.2.1 apply here as well. The performance in this EW environment it almost the same as the performance in an EW environment with only constant emitters. The main difference is at high TMNR with 2 emitters there are more trackers initialised. This is because one emitter is a constant PRI scheme and the other is a jittered PRI scheme; this situation was covered in section 3.3.3.1. At high TMNR, with the two different amounts of noise in signals, the SS algorithm gate is too big and accepts more false detections.

Table 5.10 shows the performance of the system with varying interference pulses and TMNR.

	2 Emitters											
Missing (M) and Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Mean Number of Trackers Initialised	4.3	4.1	3.8	4.1	3.8	3.7	4.5	4.4	4.3	4.4	4.3	4.2
Average Percentage Pulses Correctly Predicted	86.3	86.1	87.2	85.2	86	86.5	87.7	87.3	87.5	86.6	86.5	86.7
Mean Number of Pulses Processed Before Tracker is Initialised	8.8	9	9.4	8.9	8.7	8.9	8.7	9	9.3	8.8	8.9	9.1

Table 5.10: Interleaved Signal Results - Mixed PRI - Interference Pulses

The same conclusions drawn about varying TMNR and interference pulses in the discussion in section 5.1.1.1 apply here as well. The results in this table are similar to the results in an EW environment with only constant PRI emitters. The only difference is at high TMNR, and the reason why is already explained above.

5.1.2.4 Mixed PRI Signals - All Types

Finally an EW environment with mixed PRI scheme emitters was tested against the system. The PRIs for each emitter were as follows:

Emitter 1: 50 μs 30 μs 80 μs, Emitter 2: 94 μs, Emitter 3: 67 μs 67 μs 67 μs 42 μs 42 μs 42 μs, Emitter 4: 115 μs (10% Jittered)

In the case of two emitters being interleaved, the pulses from emitters 1 and 2 are present in the interleaved pulse train. In the case three emitters are interleaved, the pulses from emitters 1 to 3 are present in the interleaved pulse train. Finally, in the case of 4 interleaved emitters, the pulses from emitters 1 to 4 are present in the interleaved pulse train. Table 5.11 shows the performance of the system with varying emitters and TMNR.

	No missing or spurious pulses								
Emitters	**2**			**3**			**4**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Mean Number of Trackers Initialised	4.5	4.5	3.2	4.9	4.7	4.8	5.4	5.3	5.3
Average Percentage Pulses Correctly Predicted	86.2	85.6	91.4	90.2	90	89.8	92.1	92	91.8
Mean Number of Pulses Processed Before Tracker is Initialised	9	9.7	9	8.8	9.1	9.1	9	9.7	9

Table 5.11: Interleaved Signal Results - Mixed All PRI - Emitters

The same conclusions drawn on the effect emitters have on performance from section 5.1.2.1 apply here as well. This EW environment with 4 emitters is one of the most complex EW environments tested on the system, and it still manages to correctly predict at least 91.8% of pulses.

Table 5.12 shows the performance of the system with varying interference pulses and TMNR.

	2 Emitters											
Missing (M) and Spurious (S) Pulses	**M=0% S=0%**			**M=10% S=0%**			**M=0% S=10%**			**M=10% S=10%**		
TMNR	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**	**8 dB**	**16 dB**	**26 dB**
Mean Number of Trackers Initialised	4.5	4.5	3.2	4.4	4.2	3.8	4.7	4.8	4.8	4.6	4.6	4.7
Average Percentage Pulses Correctly Predicted	86.2	85.6	91.4	84.5	84.9	87.4	87.5	86.7	86.4	86.2	85.9	85.6
Mean Number of Pulses Processed Before Tracker is Initialised	9	9.7	9	9.5	9.5	9.1	8.9	9.8	9.2	9.4	9.5	9.4

Table 5.12: Interleaved Signal Results - Mixed All PRI - Interference

The same conclusions on varying TMNR and interference pulses from the discussion in section 5.1.2.1 applies here as well.

5.2 Conclusion

In this Chapter, the system was implemented on a CSIR 5^{th} generation DRFM platform. The results of the system were acceptable.

From the analysis of the results, it was determined that an increase in TMNR decreases the amount of trackers need. This means false detections and track loss is minimised at high TMNR. An increase in TMNR also decreases the amount of pulses required to be processed by the system before a tracker can be initialised. With the lower amount of pulses required to start a tracker, the percentage of correctly predicted pulses increased because the pulses before the tracker is initialised is included in the percentage calculation.

As the amount of spurious pulses increase, the percentage of correctly predicted pulses decreases because if a spurious pulse lies in the tracking window of a tracker, it is immediately assigned to that tracker. This is counted as an incorrect prediction. With more pulses, the amount of trackers can increase as the probability of false detections increase. If a spurious pulse lies before a tracker is initialised, it will increase the number of pulses required to start the tracker. Apart from the spurious pulse itself being counted, it will also mess up the TOA difference calculations. More pulses required to initialize a tracker decreases the percentage of correctly predicted pulses.

With an increase of missing pulses, if a missing pulse lies before the tracker is initialised, the amount of pulses required to initialize the tracker will increase. This is because the missing pulses will adversely affect the TOA difference calculations in a similar fashion spurious pulses would. More pulses required to initialize a tracker decreases the percentage of correctly predicted pulses.

From the EW environments that had multiple emitters in them, it was concluded that the PRI values of the emitters in the EW environment determines the number of pulses required to initialise a tracker to some extent. For example, if an emitter transmits pulses more often, it will be deinterleaved before an emitter that has longer intervals between pulses. As there are more emitters in the environment, there need to be more trackers initialised to track them.

From the single emitter EW environments, the requirements of the different PRI schemes were judged. It was required that the system start tracking a staggered PRI sequence after 4 stagger periods which equated to 12 pulses. The system started track after 7.9 pulses, meeting this requirement. The system was also required to start track after two dwell periods of a dwell and switch PRI scheme. The system did it in 1 dwell period, meeting the requirement. A constant PRI scheme needs to be tracked by at least 4 transmitted pulses, but the system started track in a mean of 5 pulses. Jittered PRI sequences had two requirements, the first was to track a jittered PRI sequence with 10% jitter and to start tracking it by 4 received pulses. The system successfully tracks the jittered sequence with a 10% jitter but did not successfully start tracking it in 4 pulses. The system took an average of 7.4 pulses to initialize the track. In the case of one emitter with only one constant PRI (a jittered PRI is equivalent to a constant PRI with added noise), an initial 10 bin CDIF histogram is created with only pulses from that one emitter with one PRI. This means pulses for the same PRI are split over many bins meaning the probability of pulses being grouped together in the same bin and exceeding the threshold is lower. More pulses are therefore required to detect an emitter in this case as pulses with the smallest variations need to be grouped together in one bin. It was proven that the CDIF algorithm could detect emitters faster than CDIF

SS but with a higher false detection rate in section 3.3. However, it would experience the same problem outlined here. Therefore, there may not be any of the required performance gain for a single emitter EW environment with a constant or jittered PRI scheme. Use of the CDIF algorithm could be investigated in the future as an alternate deinterleaving algorithm. Another possible solution is to use less than 10 bins when the measured PRIs has a small range $(range = TOA_{Newest} - TOA_{Oldest})$.

From the multiple emitter EW environments, the requirements of the different PRI schemes were judged again. A constant PRI scheme needs to be tracked by at least 4 transmitted pulses, the system started track in between 3 and 4 pulses. Therefore the requirement can be met with interleaved pulse trains. The other requirement that was not met in the single emitter EW environment was a jittered PRI scheme needs to be tracked by at least 4 transmitted pulses. In the multiple emitter environments, the system started track on one of the emitters after 4 pulses. This requirement is met with interleaved pulse trains.

The performance of system implemented in this Chapter is acceptable. It managed to correctly predict 73.6% of pulses for a worst case scenario in a single emitter EW environment. While in a multiple emitter environment, it managed to successfully predict 84.5% of pulses for a worst case scenario. The system even managed to maintain this high amount of correctly predicted pulses in the presence of missing and spurious pulses, satisfying the relevant requirements.

Chapter 6

Conclusion and Future Work

This problem statement of this research was:

"Study TOA based tracking and deinterleaving algorithms suited to radar emitters in the EW environment for application on the CSIR 5^{th} *generation DRFM platform."*

After analysing the problem statement, the major research objectives were clearly outlined.

The first objective was to research TOA based deinterleaving algorithms. The deinterleaving algorithms researched were a pulse sorting algorithm, a sequence search algorithm, histogramming based algorithms and interleaved pulse train spectrum estimation. The researched histogramming based algorithms included TOA difference histogramming, CDIF histogramming, SDIF histogramming and a two-pass weighted-search algorithm that involved combining the sequence search algorithm with a histogramming based algorithm.

The second research objective was to research TOA based tracking algorithms. The tracking algorithms researched were the Delta-τ histogram, two types of Kalman filters, the alpha-beta filter and the alpha-beta-gamma filter. The algorithms had to operate against emitters having constant, jittered, staggered or dwell and switch PRI schemes.

Objective three was to simulate the researched deinterleaving algorithms and to determine their suitability to be implemented on the CSIR 5^{th} generation DRFM platform. The deinterleaving algorithms were simulated in MATLAB to assess their suitability. The pulse sorting algorithm was not simulated as it was deemed too similar to the sequence search algorithm. The sequence search algorithm was chosen to be simulated over the pulse sorting algorithm as it was more robust. The interleaved pulse train spectrum estimation algorithm results could not be replicated in simulations. Only the original authors of the algorithm managed to obtain the results in literature, Bildøy could also not replicate the results in [29]. The deinterleaving algorithms simulated only extracted constant PRI pulse trains, as an x level stagger sequences are extracted as x constant PRI pulse trains. Algorithms were simulated as if they were already implemented on the DRFM, for example deinterleaving took place using TOAs that were stored in a FIFO ring buffer and all maximum array sizes had to be static. The algorithms were tested in simulated EW environments having varied TOA measurement noise, number of emitters and interference pulses (missing and spurious). General conclusions drawn from the simulations were the success of the algorithms decrease with the increase of emitters in the EW environment, interference pulses increased the success of some algorithms, the success of algorithms increased with TMNR (time measurement

to noise ratio). The CDIF and CDIF with sequence search (CDIF SS) were deemed suitable to be implemented on the DRFM.

The fourth objective was to simulate the researched tracking algorithms and to determine their suitability to be implemented on the CSIR 5^{th} generation DRFM platform. The tracking algorithms were simulated in MATLAB. While simulating the alpha-beta-gamma filter, it was found to become unstable as the number of pulses increased in the case of a constant PRI scheme. This was because the second state variable, which was the PRI, did not change over time such that the filter is suited. The remaining researched algorithms were also simulated as if they were already implemented on the DRFM. They were tested in simulated EW environments with varied emitter PRI schemes, TMNR and interference pulses. General conclusions drawn from the simulations were track loss of the algorithms decrease with increase in TMNR, tracking error decreases with increase in TMNR and interference pulses affect the initial estimates used to initialize the filters. The Delta-τ histogram and alpha-beta filter were deemed suitable to be implemented on the DRFM.

Objectives five and six of the research problem were to implement the suitable algorithms, on the DRFM and compare their simulation results versus their hardware performance. After the algorithms were implemented on the DRFM, they were tested against EW environments similar to the EW environments used in simulations. Since the measured TOA had to be converted from double-precision floating-point to unsigned 32-bit integers, some resolution in measurements were lost. The CDIF SS algorithm had a better deinterleaving success rate in the hardware implementation as compared to its simulation results and the CDIF algorithm hardware implementation. The CDIF algorithm, performed deinterleaving faster at a lower success rate. In hardware, the tracking error was affected due to the change in resolution of the numbers. The alpha-beta filter performed much better with a jittered PRI scheme than Delta-τ histogram in hardware.

The seventh and final objective of the research problem was to implement a system on the CSIR 5^{th} generation DRFM platform that could deinterleave and then track radar emitters in an EW environment. In order to make the system operate as fast as possible, the number of trackers running concurrently needed to be minimised. Therefore, the CDIF SS algorithm was the selected deinterleaving algorithm for the system as it experienced less false detections. The alpha-beta filter performed much better with noise and was selected as the tracking algorithm for the system. The deinterleaving and tracking algorithms were integrated into a single system that deinterleaved and tracked radar emitters based on their TOA characteristics.

Some requirements of the system were outlined in section 1.2.2, in order to verify the system. Table 6.1 provides a checklist of the project requirements, and indicates if these have been met in the study carried out and presented in this dissertation.

Requirement	Met?
1. The system shall utilize PDWs of pulses in the EW environment as inputs.	YES
2. The system shall process pulses that are indistinguishable from one another except in time of arrival.	YES
3. The system shall utilize PDWs consisting of only time of arrival information.	YES
4. The system shall be able to deinterleave pulsed radar emitters by using their associated PDWs as they are input into the system.	YES
5. The system shall deinterleave a minimum of four pulsed radar emitters.	YES
6. The system shall be able to track deinterleaved pulsed radar emitters by using their associated PDWs as they are input into the system.	YES
7. The system shall predict the TOA of the next radar pulse for each emitter.	YES
8. The system shall track a minimum of four pulsed radar emitters.	YES
9. The system shall operate in an EM environment where a maximum of 10% of transmitted radar pulses are not sensed at the receiver of the system.	YES
10. The system shall operate in an EM environment where no more than 10% of radar pulses are spurious.	YES
11. The system shall operate in an EM environment where the minimum time measurement to noise ratio is 8 dB.	YES
12. The system shall operate in an EM environment where the maximum time measurement to noise ratio is 26 dB.	YES
13. The system shall deinterleave the PRI schemes types of constant, staggered, jittered and dwell and switch.	YES
14. The system shall track the PRI scheme types of constant, staggered, jittered and dwell and switch.	YES
15. The system shall begin tracking pulsed radar emitters by 4 received pulses if they utilize a constant PRI scheme.	YES
16. The system shall begin tracking pulsed radar emitters by 4 received stagger PRI sequence repetitions if they utilize a staggered PRI scheme.	YES
17. The system shall track jittered pulsed radars with 10% jitter relative to their PRI.	YES
18. The system shall begin tracking jittered pulsed radars by 4 received pulses if they utilize a jittered PRI scheme.	YES
19. The system shall begin tracking dwell and switch PRI sequences by 2 received dwell periods if they utilize a dwell and switch PRI scheme.	YES
20. The system shall output the predicted time of arrival of the next pulse for each radar that is being tracked.	YES
21. The system shall store the radars it is currently tracking in the EM environment.	YES
22. The system shall store the PRI sequence, time of last received pulse and predicted time of next pulse of each radar it is currently tracking in the EM environment..	YES
23.The system shall operate on a CSIR 5^{th} generation DRFM platform.	YES

Table 6.1: Requirements Checklist

The solution system has met all the requirements. However, when the system is exposed to an EW environment with a single emitter with a constant or jitter PRI scheme it takes slightly longer to initialise track. To rectify this the system should initialise track quicker by using less than the currently used 10 bins in the CDIF part of the deinterleaver when the measured PRI range $(range = TOA_{Newest} - TOA_{Oldest})$ is small.

The next step in this research would be to test the performance of the system against an actual radar. A track

manager that combines trackers with the same PRI, as x level stagger sequences are extracted as x constant PRI pulse trains should also be implemented. The track manager should also decide which pulse belongs to a tracker when more than one pulse lies in the tracking window. As the system tracks an emitter more precisely, a dynamic gate size could be implemented.

The following points should be considered in the future to improve the current research. If the deinterleaver was changed from CDIF SS to just CDIF, how will it affect the performance of the system? When the number system changed while working on the DRFM, the performance of the sequence search in the CDIF SS algorithm improved. It would be interesting to investigate the performance of the sequence search algorithm on the DRFM. The simulations in MATLAB can be updated to include the resolution loss when working with the DRFM to help bridge results between simulations and hardware. Neural networks and their suitability as a deinterleaving algorithm on the system can be investigated. The suitability of the multiple hypothesis tracking method as a deinterleaver on the system can also be investigated. The Kalman filter is complex and requires many processing cycles due to matrix multiplication and inversion. Since the time domain Kalman filter consists of at most 2×2 matrices, further research can be carried out to simplify the Kalman filter into a series of linear equations to reduce complexity. Modern radar pulses increase the bandwidth of the transmitted pulses to achieve better range resolution performance, as discussed in section 2.2.2. The pulse modulation types and the angle of arrival information of pulses can be added to the system to improve deinterleaving in the future. As pulses are processed coherently, the system will also have to track the pulse modulation.

In conclusion, all major research objectives were met. The system designed in objective seven met all requirements and performed well.

Bibliography

[1] Roland Minihold and Dieter Bues, “Introduction to Radar system and Component tests”, ROHDE & SCHWARZ, 2012.

[2] Richard G. Wiley, “Electronic Intelligence: The Analysis of Radar Signals”, 2nd edition, ARTECH HOUSE, ISBN: 0-89006-592-6, 1993.

[3] Soner Avcu, “Radar Pulse Repetition Interval Tracking With Kalman Filter”, THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES OF MIDDLE EAST TECHNICAL UNIVERSITY, 2006.

[4] Keysight Technologies, “Edge Jitter Types”, http://rfmw.em.keysight.com/wireless/helpfiles/n7620a/Content/Main/Edge_Jitter_Type.htm, Accessed: 05/05/2015.

[5] Jan Matuszewski , “The Specific Radar Signature in Electronic Recognition System”, PRZEGLĄD ELEKTROTECHNICZNY, ISSN: 0033-2097, 2013.

[6] Richard G. Wiley, “PRI Analysis and Deinterleaving”, Research Associates of Syracuse, Inc, 2013.

[7] Francis X. Grovers The Third, “Kalman Filter Tutorial, Unmanned Ground Vehicle Fundamentals”, Unmanned Vehicles University, 2013.

[8] Melinda Hock, “Kalman Filter Predictor And Initialization Algorithm For PRI Tracking”, UNITED STATES NAVAL RESEARCH LABORATORY, 1998.

[9] Samuel Lin, Michael Thompson, Stephen Davezac and John C. Sciortino Jr., “Comparison Of Time Of Arrival versus Multiple Parameter Based Radar Pulse Train Deinterleavers”, Naval Research Laboratory, Innolog, Inc., 2014.

[10] D.J. Milojevic and B.M. Popovik , “Improved algorithm for the deinterleaving of radar pulses”, I E E PROCEEDINGS-F, Vol. 139, 1992.

[11] RAFAEL Advanced Defense Systems LTD., “C-Pearl-DV(S): Compact, high performance Digital ESM system for submarine” Brochure, “http://www.rafael.co.il/marketing/SIP_STORAGE/FILES/0/930.pdf ”, Accessed: 01/06/2015.

[12] RAFAEL Advanced Defense Systems LTD., “Sky Shield: EW Support Jamming System” Brochure, “http://www.rafael.co.il/marketing/SIP_STORAGE/FILES/8/958.pdf ”, Accessed: 01/06/2015.

[13] RAFAEL Advanced Defense Systems LTD., “Lite Shield: Electronic Attack Pod For Close Protection and Escort Jamming” Brochure, “http://www.rafael.co.il/marketing/SIP_STORAGE/FILES/7/1107.pdf”, Accessed: 01/06/2015.

[14] RAFAEL Advanced Defense Systems LTD., “Jam –Air: A Directional IR Countermeasure (DIRCM) System” Brochure, “http://www.rafael.co.il/marketing/SIP_STORAGE/FILES/9/959.pdf”, Accessed: 01/06/2015.

[15] Remi Gauvin, “Achieving real-time pulse-to-pulse PRI prediction”, MC Countermeasures INC., January 2004.

[16] Israel Aerospace Industries, ELTA Systems LTD., “Advanced Naval ESM/ System: ELL-8385N” Brochure, “http://www.iai.co.il/Sip_Storage//FILES/7/38047.pdf”, Accessed: 01/06/2015.

[17] Israel Aerospace Industries, ELTA Systems LTD., “ELTA’s Integral Compact Self-Protection Electronic Warfare Suites: EL/L-8247/8” Brochure, “http://www.iai.co.il/Sip_Storage//FILES/0/40790.pdf”, Accessed: 01/06/2015.

[18] Rockwell Collins, “CS-3045 Airborne ELINT/ESM Subsystem” Brochure, “http://www.marlboroughcomms.com/media/9376/CS-3045-Data-Sheet-1009.pdf”, Accessed: 01/06/2015.

[19] SAAB Technologies, “Electronic Surveillance Payload For UAV Applications” Brochure, “http://ftp.combitech.se/Global/Documents%20and%20Images/Air/ Electronic%20Warfare%20Solutions/ESP/ESP%20product%20sheet.pdf”, Version 4, August 2010.

[20] RAFAEL Advanced Defense Systems LTD., “C-Pearl-DV: Compact, high performance Digital ESM system” Brochure, “http://www.rafael.co.il/marketing/SIP_STORAGE/FILES/8/1248.pdf”, Accessed: 01/06/2015.

[21] National Aeronautics And Space Administration, “Systems Engineering Handbook”, Revision 1, December 2007.

[22] International Council On Systems Engineering (INCOSE), “Systems Engineering Handbook: A Guide For System Life Cycle Processes And Activities”, Version 3.2.2, October 2011.

[23] National Imagery and Mapping Agency, “Radar Navigation and Maneuvering Board Manual”, Seventh Edition, Lighthouse Press, ProStar Publications, Inc., 2001.

[24] Christian Wolff, “Radartutorial, Book 2: Radar Sets”, SMSgt. G.A.F. (Rtd.), 2009.

[25] Yujun Kuang, Qingbo Shi, Qianbin Chen, Li Yun and Keping Long, “A Novel SDIF-based PRI Estimation Approach to Deinterleave Repetitive Pulse Sequences”, Special Research Centre for Optical Internet and Wireless Information Networks (COIWIN), Chongqing University of Posts & Telecommunications, 400065, July 2005.

[26] Mehmet Kadir Aslan, “Emitter Identification Techniques In Electronic Warfare”, The Graduate School Of Natural And Applied Sciences, Middle East Technical University, September 2006.

[27] H.K. Mardia, “New Techniques For The Deinterleaving Of Repetitive Sequences”, IEE Proceedings, Vol. 136, Pt. F, No. 4, August 1989.

[28] Shruti Sridharan, Dr. NNSSRK Prasad, Riya George and M. Brindha, “Improved Pulse Repetition Interval (PRI) Deinterleaving For Electronic Support Measure (ESM) Receiver ”, MVJ College Of Engineering, Bangalore, ISSN(PRINT):2394-3408,(ONLINE):2394-3416,VOLUME-2,ISSUE-3, 2015.

[29] Bent Einar Stenersen Bildøy, “Satellite Cluster Concepts - A System Evaluation With Emphasis On Deinterleaving And Emitter Recognition ”, Norwegian University of Science and Technology, June 2006.

[30] Robert J. Orsi, John B. Moore and Robert E. Mahony, “Spectrum Estimation of Interleaved Pulse Trains”, IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 47, NO. 6, JUNE 1999.

[31] Ken’ichi Nishiguchi, “Time-period Analysis for Pulse Train Deinterleaving”, Trans. of the Society of Instrument and Control Engineers Vol.E-4, No.1, 68/78, 2005.

[32] Stephen Eikenberry, “Pulsars: Observation & Timing”, Lecture 5 - Pulsar Timing Notes,January 2014.

[33] Robert J. Orsi, John B. Moore and Robert E. Mahony, “Interleaved Pulse Train Spectrum Estimation”, International Symposium on Signal Processing and its Applications, ISSPA, Gold Coast, Australia , August 1996.

[34] Chris Baker, “An Introduction to Radar Systems”, Dept. Electronic & Computer Engineering, Ohio State University, 2014.

[35] Christian Wolff, “Radartutorial, Book 1: Radar Basics”, SMSgt. G.A.F. (Rtd.), 2009.

[36] Professor David Jenn, “Radar Fundamentals”, Department of Electrical & Computer Engineering, NAVAL POSTGRADUATE SCHOOL, 2014.

[37] Mark A. Richards, James A. Scheer And William A. Holm, “Principles of Modern Radar, Vol. I: Basic Principles”, SciTech Publishing, 2010.

[38] United States Navy, “Navy Electricity and Electronics Training Series, Module 18 - Radar Principles”,Lulu Press, Inc., September 1998.

[39] Mark A. Richards, “Fundamentals of Radar Signal Processing”, Second Edition, McGraw-Hill, New York, 2014.

[40] Dr. Sheng-Chou Lin, “Course Notes: Radar System Design”, Fu Jen Catholic University, 2016.

[41] Merrill I. Skolnik, “Radar Handbook”, Third Edition, McGraw-Hill, 2008.

[42] J.C. Toomay and Paul J. Hannen, “Radar Principles for the Non-Specialist”, Third Edition, Scitech Publishing, Inc., 2004.

[43] Microwaves101.com, "http://www.microwaves101.com/encyclopedias/frequency-letter-bands", Accessed: 12/08/2016.

[44] Iroegbu Chibuisi and Okonba J. Brown, "Application of Radar Technology in Combating Insurgency in Nigeria", International Journal for Research in Applied Science and Engineering Technology (IJRASET), Vol. 2 Issue VIII, ISSN: 2321-9653, August 2014.

[45] Marco Lanzagorta, "Quantum Radar", Morgan & Claypool Publishers, 2012.

[46] A.W. Ata'a and S.N. Abdullah, "Deinterleaving of radar signals and PRF identification algorithms", University of Baghdad, 2002.

[47] Mehmet Burak Kocamis, Hakan Abac, Safak Bilgi Akdemir, Sertan Varma and Alper Yildirim, "Deinterleaving for Radar Warning Receivers with Missed Pulse Consideration",13th European Radar Conference, The Scientific and Technological Research Council of Turkey , October 2016.

[48] Navid Daryasafar and Hamid Dehghani, "Studying an Improved Interval-Only Algorithm for the De-Interleaving of Radar Pulses", Journal of World's Electrical Engineering and Technology, ISSN: 2322-5114, April 2015.

[49] Robert Penoyer, "The Alpha-Beta Filter", The C User's Journal, Vol. 11, Issue 7, R & D Publications, Inc, July 1993.

[50] Macharla Vinaykumar and Ravi Kumar Jatoth, "Performance Evaluation of Alpha-Beta and Kalman Filter for Object Tracking", IEEE ICACCT, ISBN No. 978-1-4799-3914-5, 2014.

[51] G. R. Deeba Lakshmi , R. Gopalakrishnan and Manjunath R. Kounte, "Detection and Extraction of Radio Frequency and Pulse Parameters in Radar Warning Receivers", ERCICA 2013, Elsevier Publications, ISBN: 9789351071020, 2013.

[52] Paul R. Kalata, "The Tracking Index: A Generalized Parameter for $\alpha-\beta$ and $\alpha-\beta-\gamma$ Target Trackers", IEEE Transactions On aerospace And Electronic Systems Vol. AES-20, No. 2, 1984.

[53] Eli Brookner, "Tracking and Kalman Filtering Made Easy", John Wiley & Sons, Inc., ISBN: 0-471-22419-7, 1998.

[54] Dr. Robert M. O'Donnell, "Radar Systems Engineering", Course Lectures, IEEE Aerospace and Electronic Systems Society, 2009.

[55] Roland Minihold and Dieter Bues, "Introduction to Radar system and Component tests", ROHDE & SCHWARZ, Edition 2, 2013.

[56] Christo Cloete, "RF EW EDERI Applications & Expansion: EW Reference Document", CSIR, Document No 5865-AEKB-00001 RPT Rev A, January 2011.

[57] David L. Adamy, "EW 102: A Second Course in Electronic Warfare", Artech House, ISBN No. 1-58053-686-7, 2004.

[58] He Gang, Liu Hao, Li Xin and Huang Qin-huang, “Radar Signal Sorting Method in Dual Airborne-platform Cooperative Reconnaissance”, Atlantis Press, ISBN No. 978-94-6252-150-6, 2016.

[59] Ken’ichi Nishiguchi and Masaaki Kobayashi, “Improved Algorithm for Estimating Pulse Repetition Intervals”, Mitsubishi Electric Corporation, IEEE Transactions on Aerospace and Electronic Systems VOL. 36, NO. 2, 2000.

[60] Defence Acquisition University Press, “Systems Engineering Fundamentals”, Defense Acquisition University Press, Fort Belvoir, Virginia, January 2001.

[61] Justin Darrell Pelan, “Modular Multi-Signal Tracking Pulse Descriptor Word (PDW) Generator with Field Programmable Gate Array (FPGA) Implementation”, Wright State University, 2016.

[62] Steven W. Smith, “The Scientist and Engineer’s Guide to Digital Signal Processing”, California Technical Pub., ISBN No. 0966017633 (ISBN13: 9780966017632) , 1997.

[63] Bassem R. Mahafza, “Radar Systems Analysis and Design Using MATLAB”, Chapman & Hall/CRC, ISBN No. 1-58488-182-8, 2000.

[64] D. C. Jenn, "Microwave Devices & Radar Notes, vol. I”, Naval Postgraduate School, Monterey, California, 2009.

[65] D. C. Jenn, "Microwave Devices & Radar Notes, vol. II”, Naval Postgraduate School, Monterey, California, 2009.

[66] Rodger A. Brown and Vincent T. Wood, “A Guide for Interpreting Doppler Velocity Patterns: Northern Hemisphere Edition”, NOAA/National Severe Storms Laboratory, Norman, Oklahoma, Second Edition, June 2007.

[67] Ronald E. Rinehart, “RADAR for Meteorologists”, 3rd Edition, ISBN No. 0965800202, 1997.

[68] J.J. Strydom, J.E. Cilliers, M. Gouws, D. Naicker and K. Olivier, “Hardware in the loop radar environment simulation on wideband DRFM platforms”, IET International Conference on Radar Systems (Radar 2012), 2012.

[69] William Metz, “Electronic Warfare Receiver Resource Management and Optimization”, Walden University ScholarWorks, 2016.

[70] Xie Guo-liang, Wang Hong-xun, Xu Zhongwei and Wang Chao, “A Fast Sorting Method for Modulated and Jitter PRI Radar Signals”, International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE), 2011.

[71] National Aeronautics And Space Administration, “The Electromagnetic Spectrum”, Astronomer’s Toolbox, “https://imagine.gsfc.nasa.gov/science/toolbox/emspectrum1.html”, Edited: March 2013.

[72] I. Vaughan and L. Clarkson, “Approximate Maximum-Likelihood Period Estimation From Sparse, Noisy Timing Data”, IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 56, NO. 5, MAY 2008.

[73] I. Vaughan, L. Clarkson, Stephen D. Howard and Iven. M. Y. Mareels, “ESTIMATING THE PERIOD OF A PULSE TRAIN FROM A SET OF SPARSE, NOISY MEASUREMENTS”, Fourth International Symposium on Signal Processing and Its Applications , vol. 2 ,1996.

[74] Eric W Weisstein, “Autocorrelation”, MathWorld–A Wolfram Web Resource, “http://mathworld.wolfram.com/Autocorrelation.html”, 2005.

[75] MathWorks, “Find Periodicity Using Autocorrelation”, Signal Processing Toolbox Documentation, “https://www.mathworks.com/help/signal/ug/find-periodicity-using-autocorrelation.html”, Acessed: 26/08/2015.

[76] B. J. Slocumb and E. W. Kamen, “The pulse train PDA analysis and deinterleaving filter”, Proc. SPIE Vol. 3068, ISBN No. 0-8194-2483-8, 1997.

[77] Stephen D. Elton and Benjamin J. Slocumb, “A Robust Kalman Filter For Estimation And Tracking of A Class of Periodic Discrete Event Processes”, IEEE Xplore, Fourth International Symposium on Signal Processing and Its Applications , vol. 1, 1996.

[78] Gary Alan Sipe, “Target Detection Via Kalman Filtering”, Naval Postgraduate School, Monterey, California, 1993.

[79] Greg Welch and Gary Bishop, “An Introduction to the Kalman Filter”, Department of Computer Science, University of North Carolina, Chapel Hill, 24 July 2006.

[80] John A. Lawton, Robert J. Jesionowski and Paul Zarchan, “COMPARISON OF FOUR FILTERING OPTIONS FOR A RADAR TRACKING PROBLEM”, Journal of Guidance, Control, and Dynamics, Vol. 21, No. 4, 1998.

[81] Niklas Pääkkönen, “Speed Measurement and Filtering for Rotating Machinery ”, Department of Information Technologies, Åbo Akademi University, 2010.

[82] Dirk Tenne and Tarunraj Singh, “Characterizing Performance of α-β-γ Filters”, IEEE Transactions On Aerospace And Electronic Systems, Vol. 38, No.3, July 2002.

[83] Wikus Beetge and Klasie Olivier, “5th Generation DRFM”, Rev 1, CSIR, Document No. 5963-DRFM5G-10000, 2012.

[84] CSIR, “5th Generation Wide Band DRFM” Brochure, “https://www.csir.co.za/sites/default/files/Documents/5th%20Generation%20Wide%20Band%20DRFM.pdf”, 2012.

Appendix A

Radar Emission Frequency Bands

Table adapted from [1, 43, 44]

Band	Meaning/Origin	Frequency	Application
HF	High Frequency	3 to 30 MHz	Coastal and over-the-horizon (OTH) radars.
P or VHF	P for "previous", the British used the band for their earliest radars, but later switched to higher frequencies. Very High Frequency	30 to 300 MHz	Used in early radar systems.
UHF	Ultra High Frequency	300 to 1000 MHz	Very long range (eg. ballistic missile early warning), ground penetrating and foliage penetrating radars.
L	Long wave	1 to 2 GHz	Long range air traffic control and surveillance radars.
S	Short wave	2 to 4 GHz	Terminal air traffic control, long range weather and marine radars.
C	C for "compromise" between S and X band.	4 to 8 GHz	Satellite transponders and weather radars.
X	X for cross (as in crosshair)	8 to 12 GHz	Missile guidance, marine radar, weather, medium-resolution mapping and ground surveillance radar. Also used in WWII for fire control.
Ku	Ku for "kurz-under".	12 to 18 GHz	High-resolution mapping and satellite altimetry.
K	German "kurz" means short.	18 to 27 GHz	Used by meteorologists for detecting clouds and by police for detecting speeding motorists.
Ka	Ka for "kurz-above".	27 to 40 GHz	Mapping, short range, airport surveillance and to trigger cameras that take pictures of license plates of cars running red lights.
V	V for "very" high frequency band (not to be confused with VHF).	40 to 75 GHz	Used for military communications.
W	W follows V in the alphabet.	75 to 110 GHz	Automotive radar, high-resolution meteorological observation and imaging.

Appendix B

The DRFM Setup

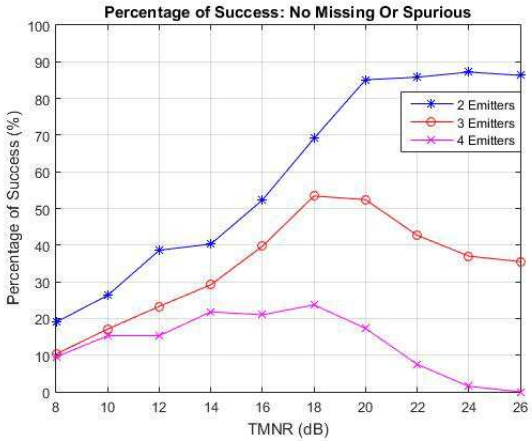
The DRFM Connected to a laptop running MATLAB.

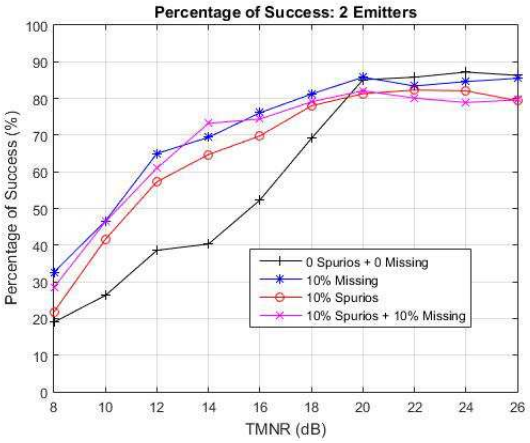
Appendix C

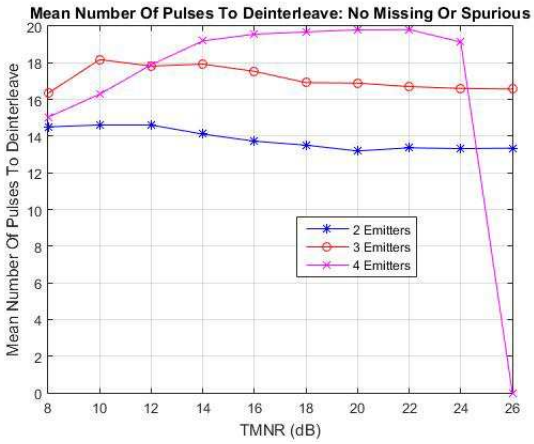
Emitter Deinterleaving - Simulation Results

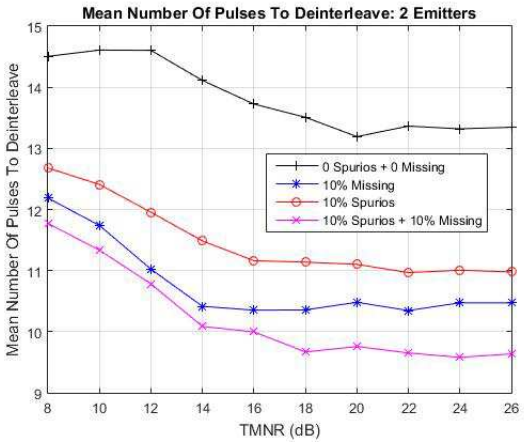
Constant PRI Signals

CDIF

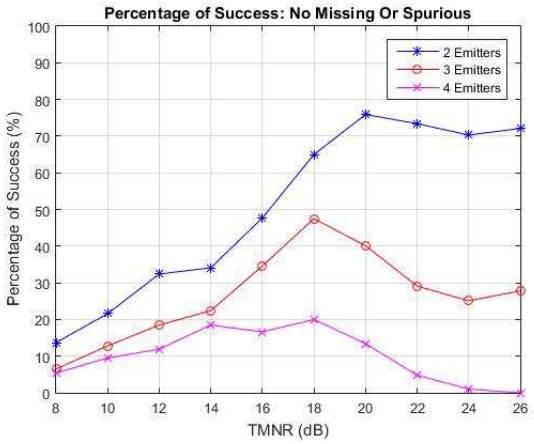


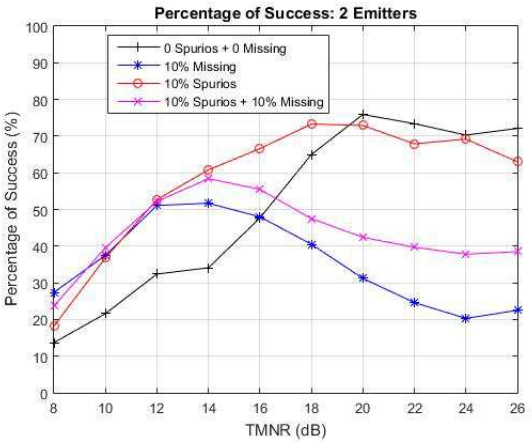


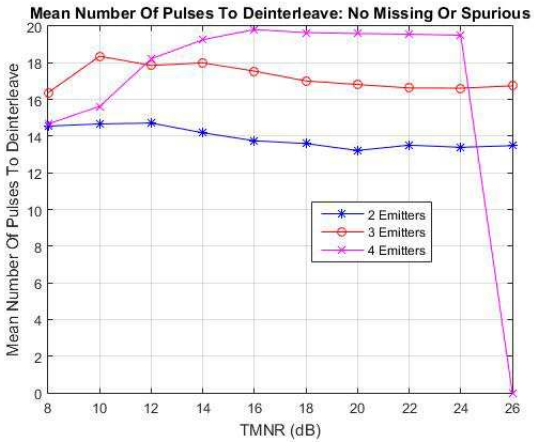


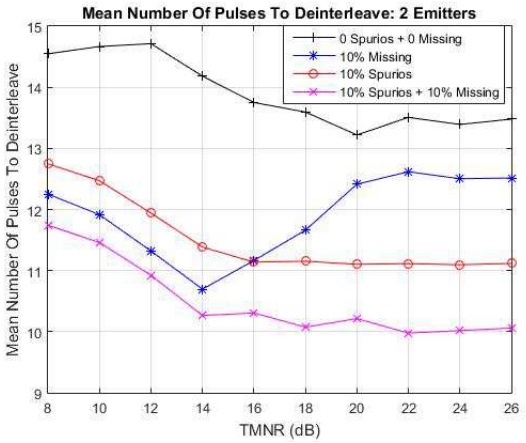


CDIF SS

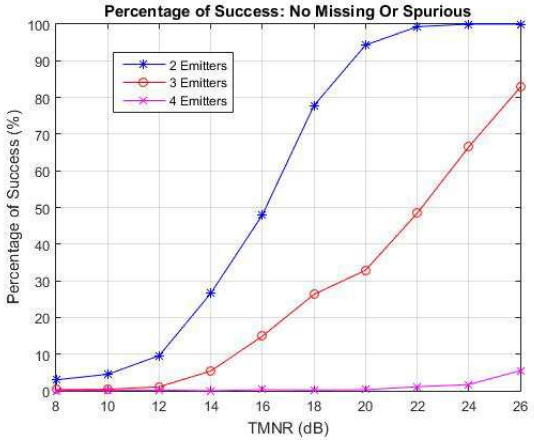


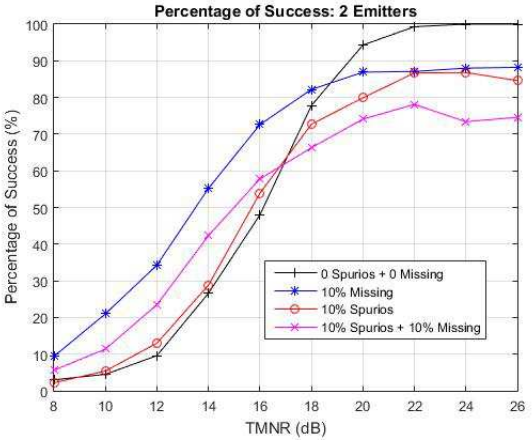


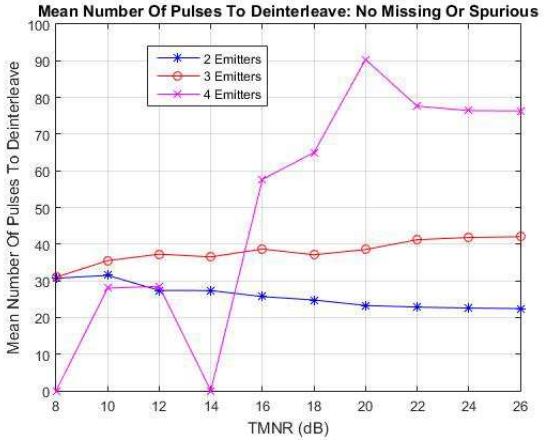


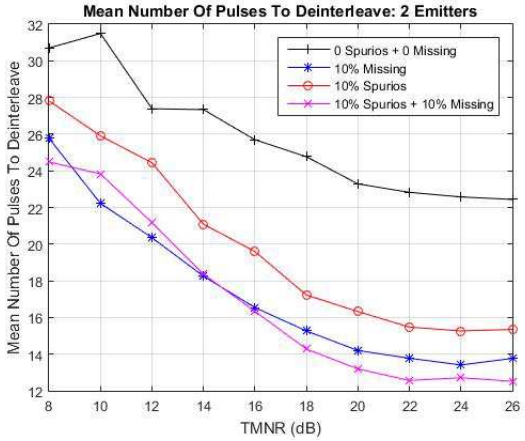


TOA Difference Histogram

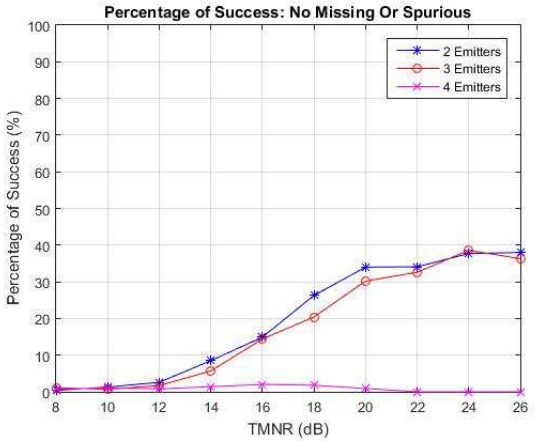


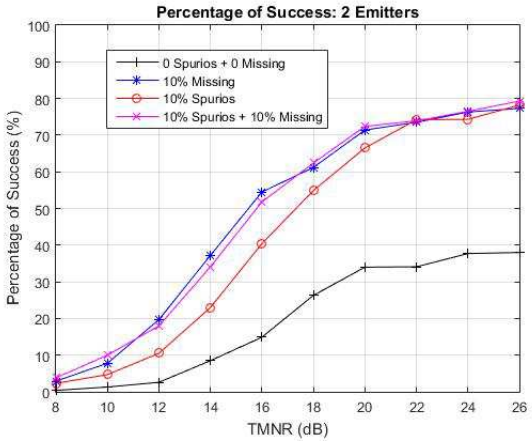


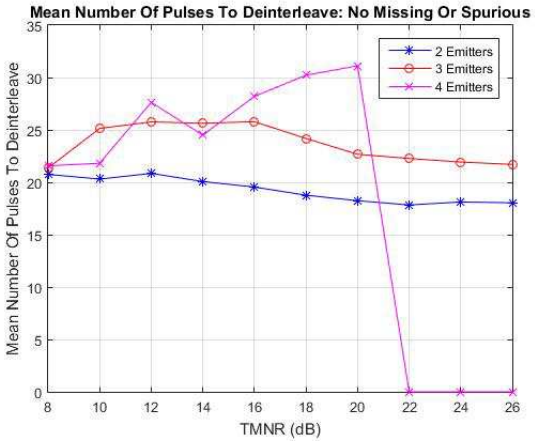

Mean Number Of Pulses To Deinterleave: No Missing Or Spurious
2 Emitters
3 Emitters
4 Emitters
Mean Number Of Pulses To Deinterleave
TMNR (dB)

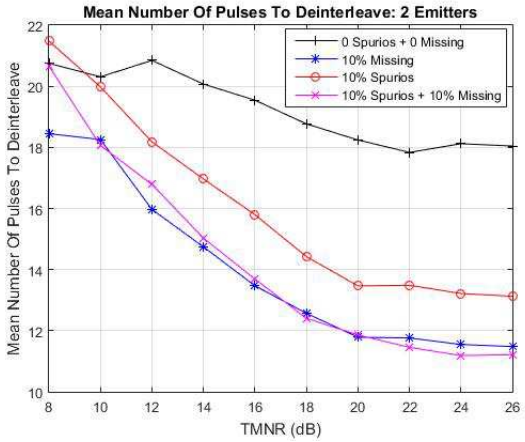

Mean Number Of Pulses To Deinterleave: 2 Emitters
0 Spurios + 0 Missing
10% Missing
10% Spurios
10% Spurios + 10% Missing
Mean Number Of Pulses To Deinterleave
TMNR (dB)

SDIF

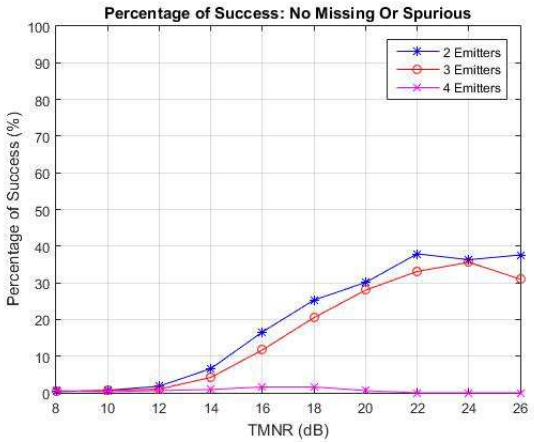

Percentage of Success: No Missing Or Spurious
2 Emitters
3 Emitters
4 Emitters
Percentage of Success (%)
TMNR (dB)

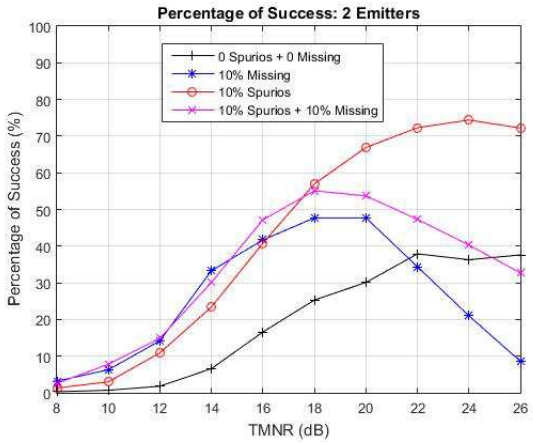

Percentage of Success: 2 Emitters
0 Spurios + 0 Missing
10% Missing
10% Spurios
10% Spurios + 10% Missing
Percentage of Success (%)
TMNR (dB)

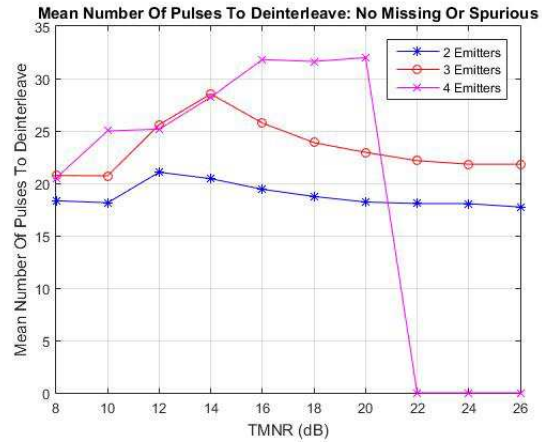

Mean Number Of Pulses To Deinterleave: No Missing Or Spurious
2 Emitters
3 Emitters
4 Emitters
Mean Number Of Pulses To Deinterleave
TMNR (dB)

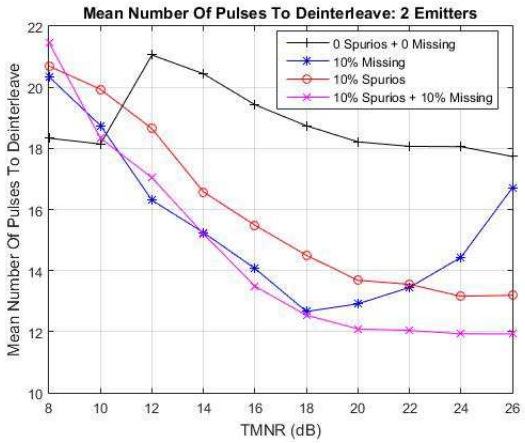

Mean Number Of Pulses To Deinterleave: 2 Emitters
0 Spurios + 0 Missing
10% Missing
10% Spurios
10% Spurios + 10% Missing
Mean Number Of Pulses To Deinterleave
TMNR (dB)

SDIF SS

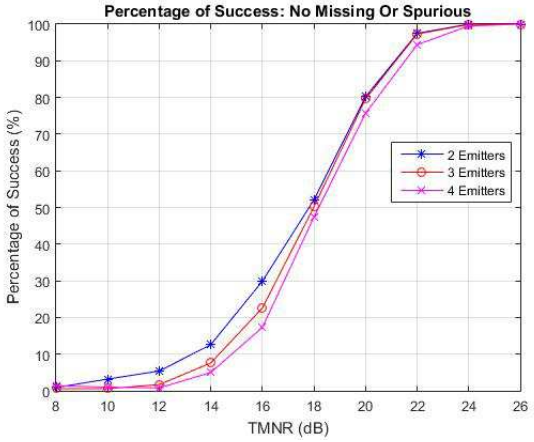


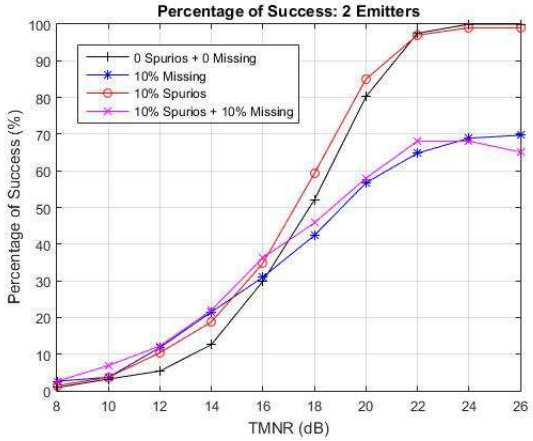


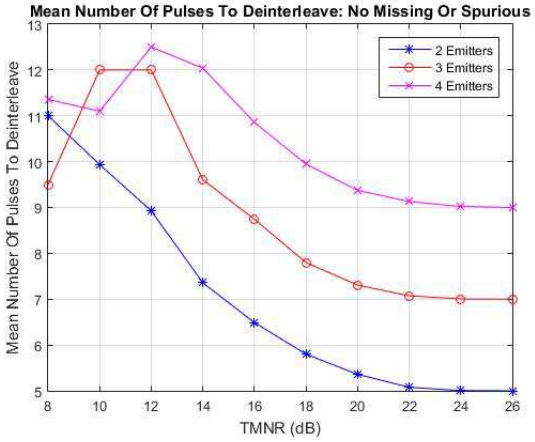


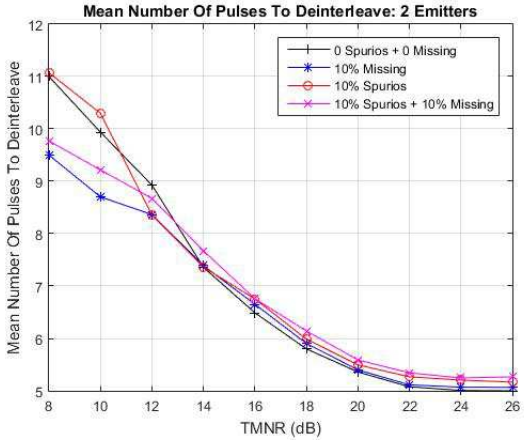


Sequence Search



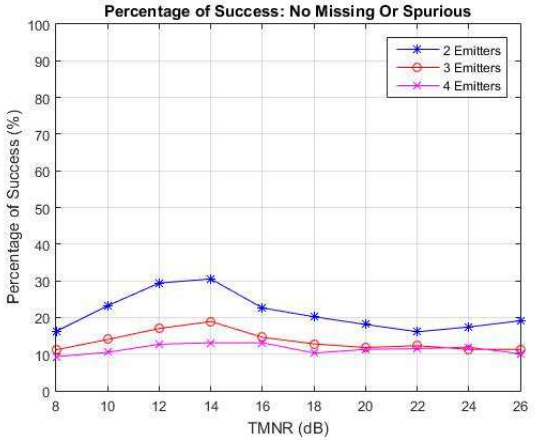


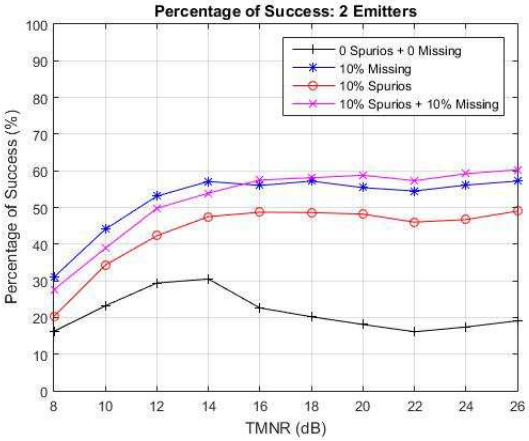


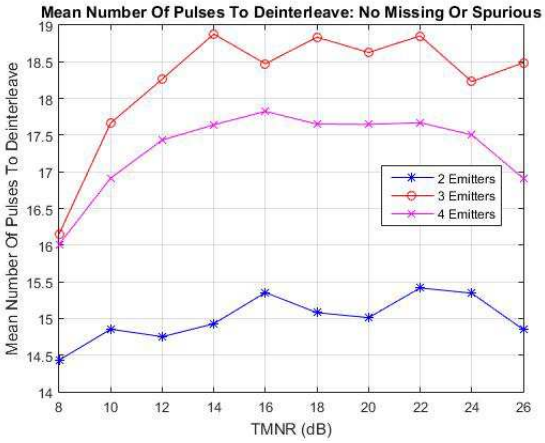


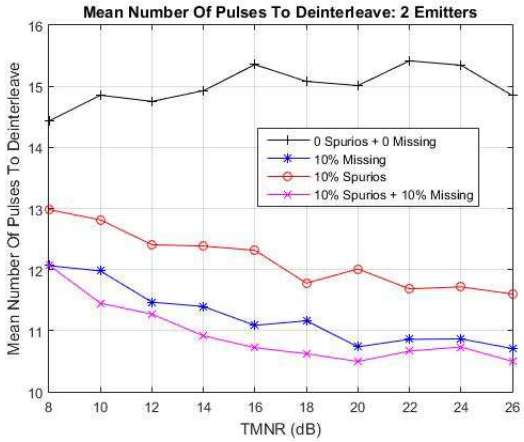
Jittered PRI Signals

CDIF

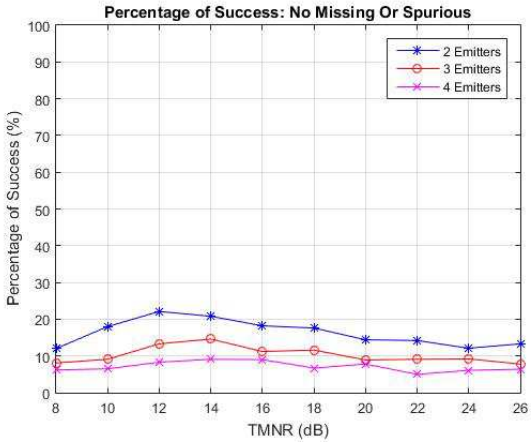


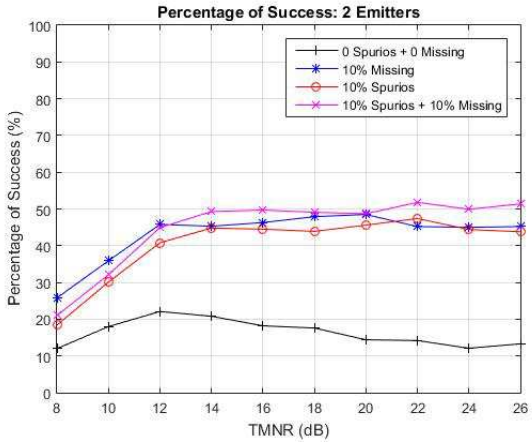


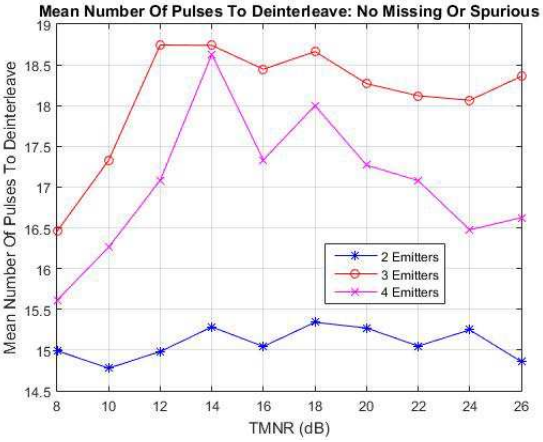


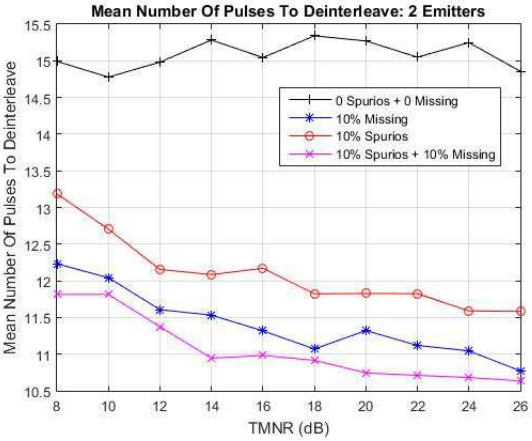


CDIF SS

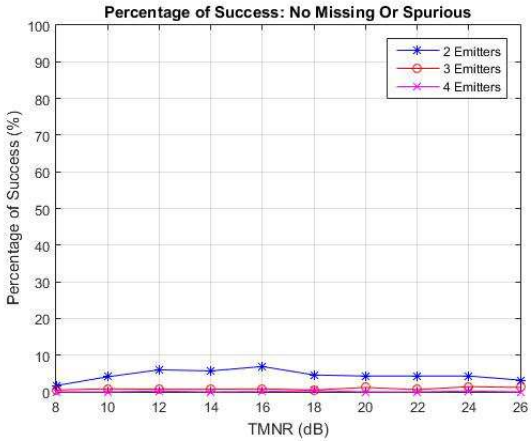


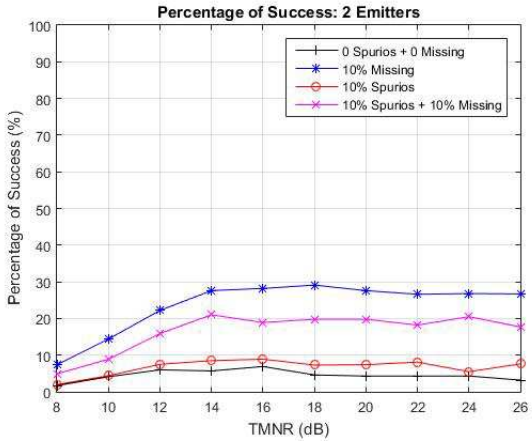


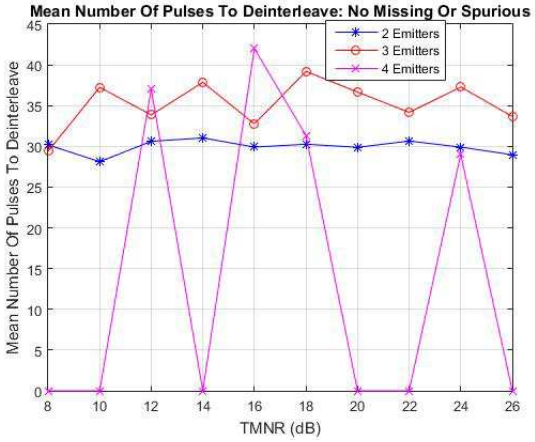


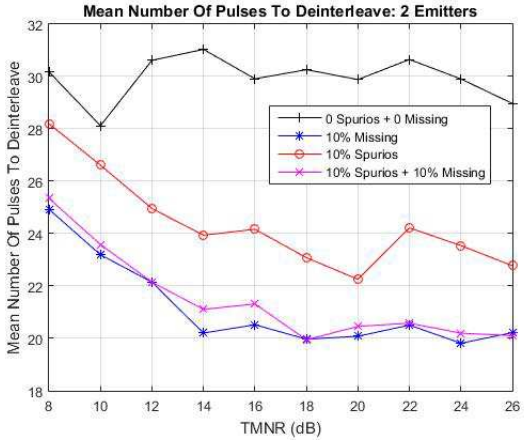


TOA Difference Histogram

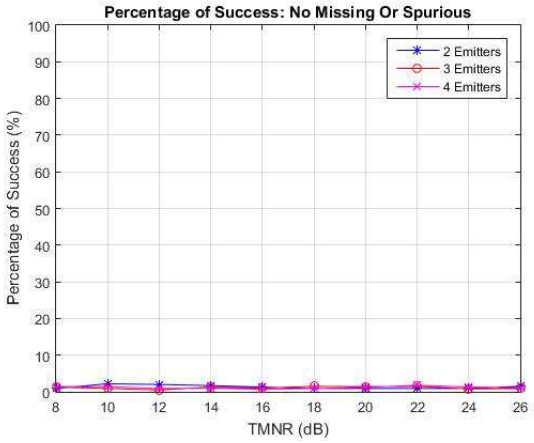


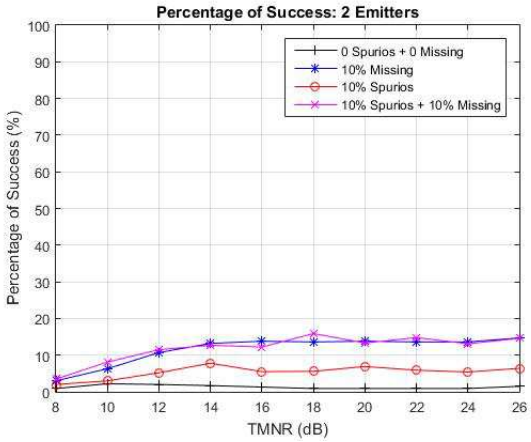


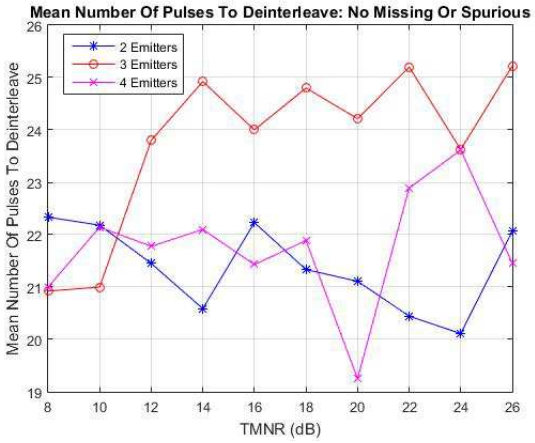

Mean Number Of Pulses To Deinterleave: No Missing Or Spurious
2 Emitters
3 Emitters
4 Emitters
Mean Number Of Pulses To Deinterleave
TMNR (dB)

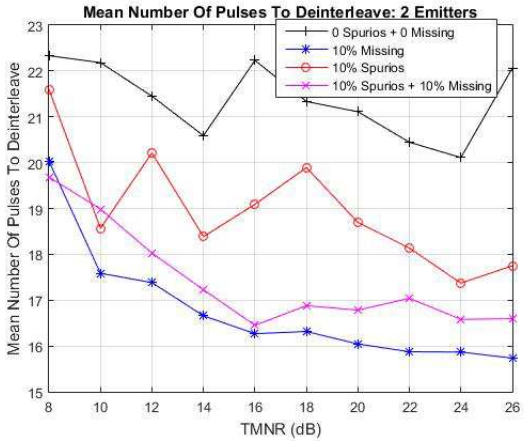

Mean Number Of Pulses To Deinterleave: 2 Emitters
0 Spurios + 0 Missing
10% Missing
10% Spurios
10% Spurios + 10% Missing
Mean Number Of Pulses To Deinterleave
TMNR (dB)

SDIF

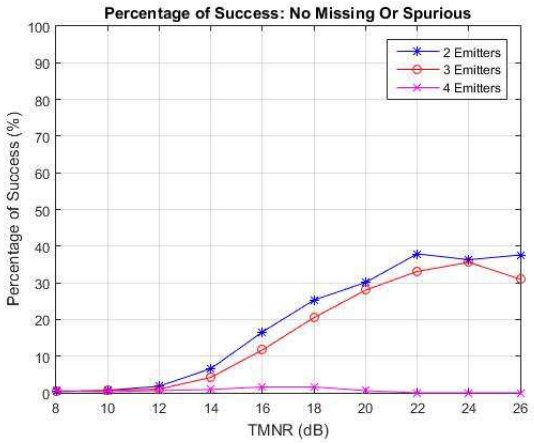

Percentage of Success: No Missing Or Spurious
2 Emitters
3 Emitters
4 Emitters
Percentage of Success (%)
TMNR (dB)

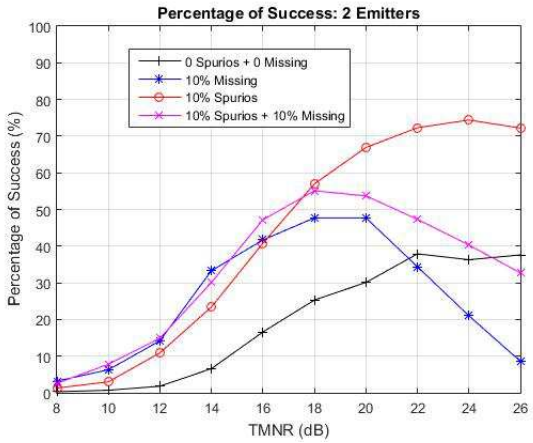

Percentage of Success: 2 Emitters
0 Spurios + 0 Missing
10% Missing
10% Spurios
10% Spurios + 10% Missing
Percentage of Success (%)
TMNR (dB)

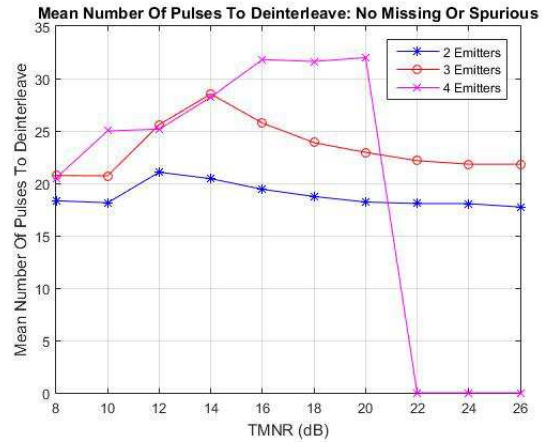

Mean Number Of Pulses To Deinterleave: No Missing Or Spurious
2 Emitters
3 Emitters
4 Emitters
Mean Number Of Pulses To Deinterleave
TMNR (dB)

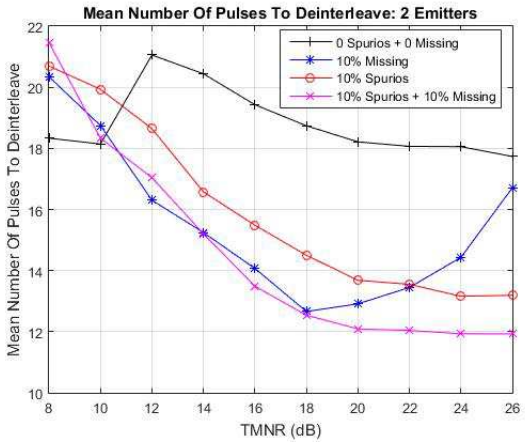

Mean Number Of Pulses To Deinterleave: 2 Emitters
0 Spurios + 0 Missing
10% Missing
10% Spurios
10% Spurios + 10% Missing
Mean Number Of Pulses To Deinterleave
TMNR (dB)

SDIF SS

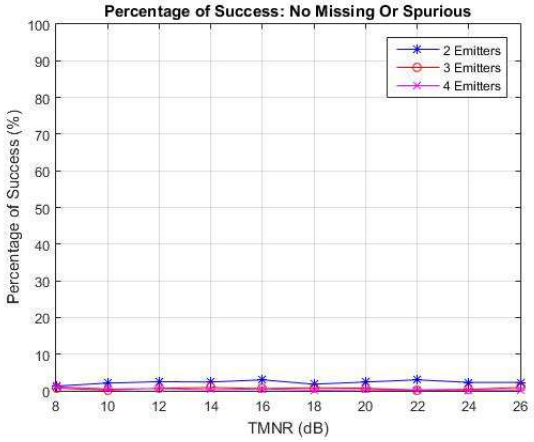


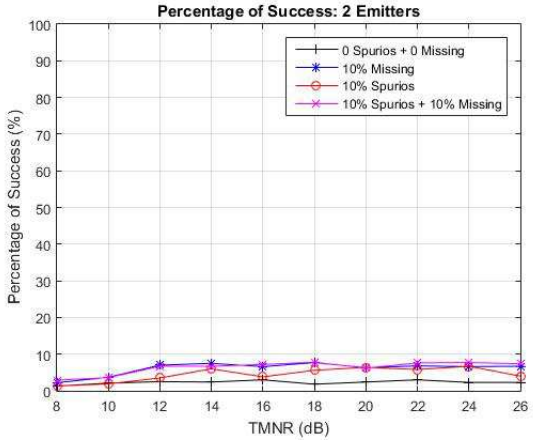


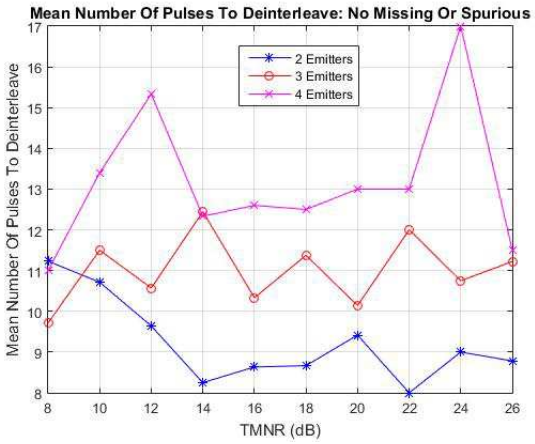


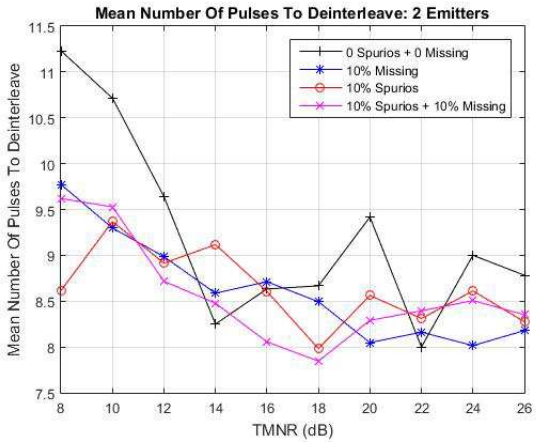


Sequence Search



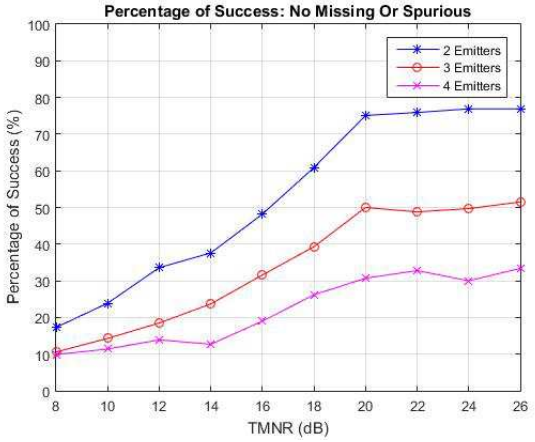


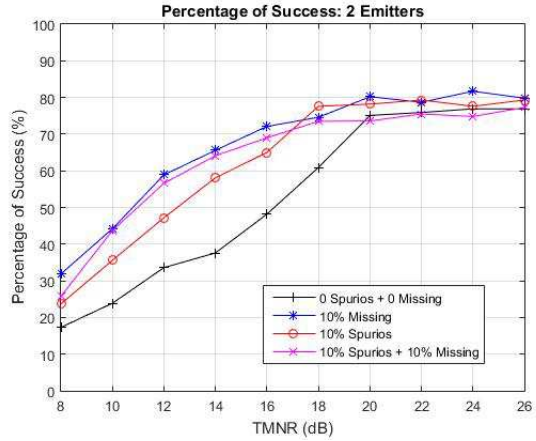


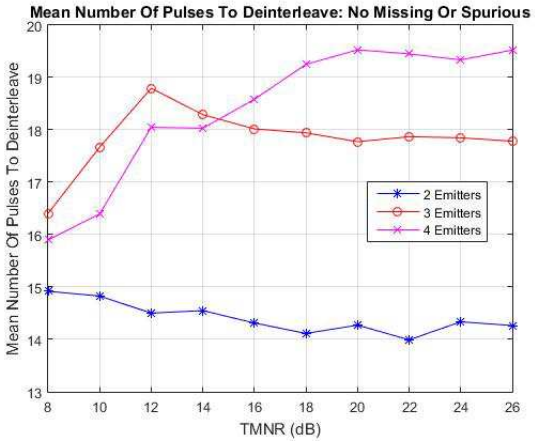


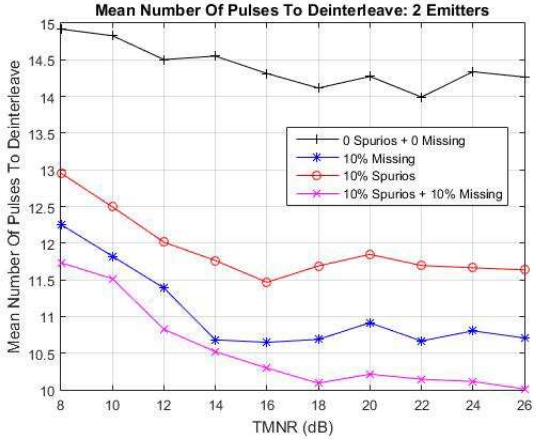
Mixed PRI Signals

CDIF

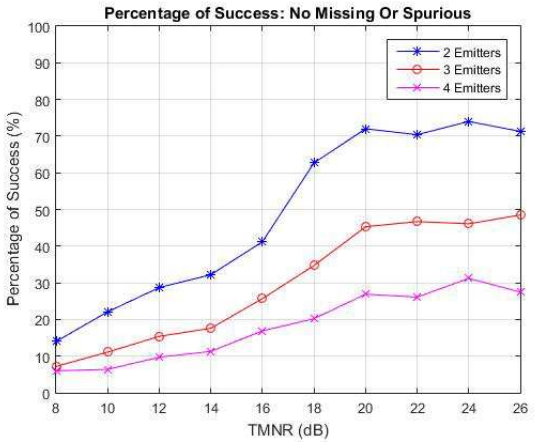


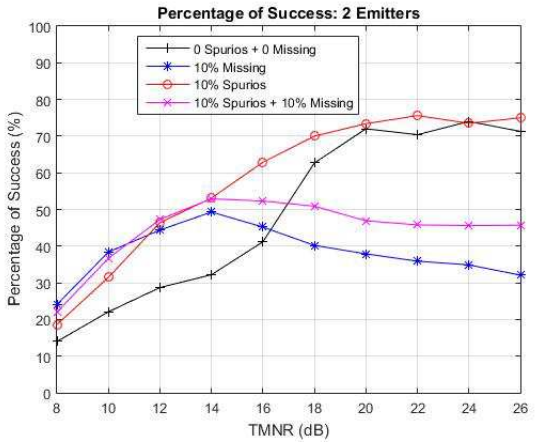


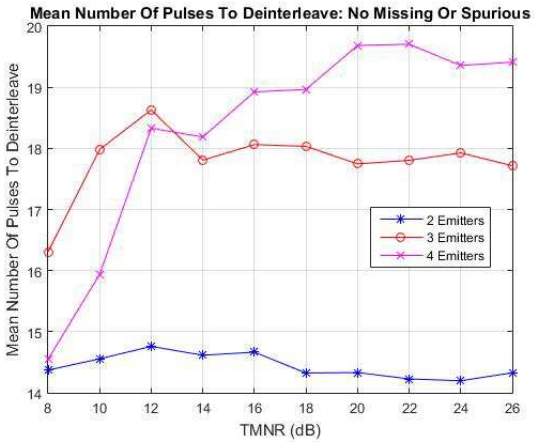


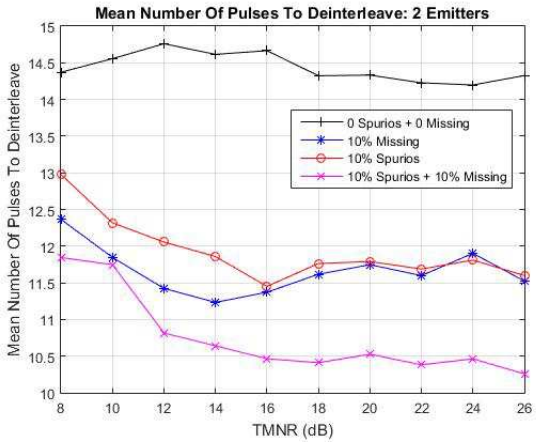


CDIF SS

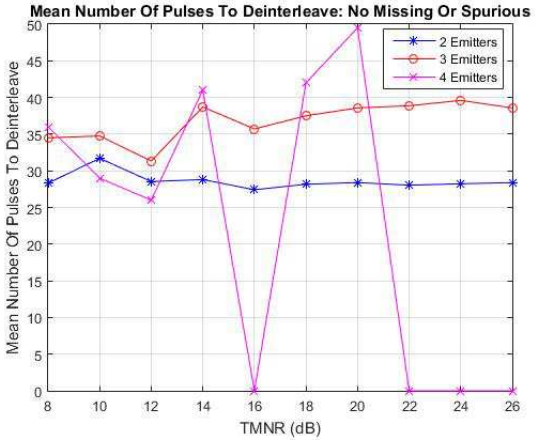


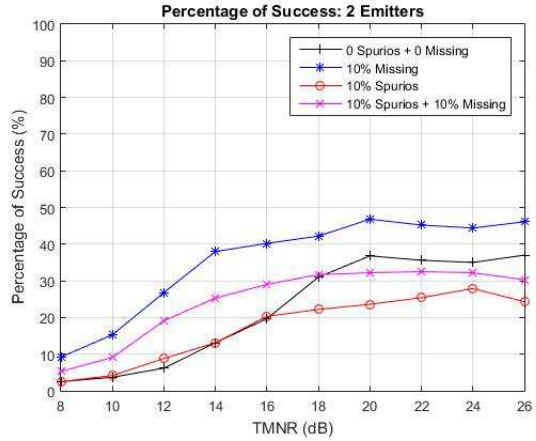


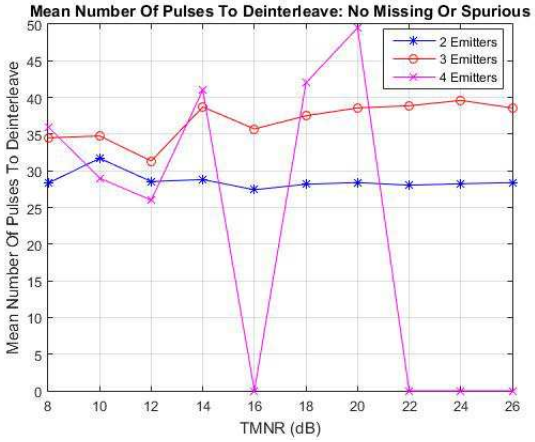


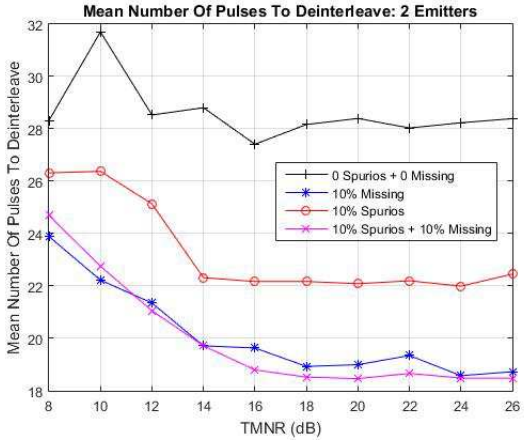


TOA Difference Histogram

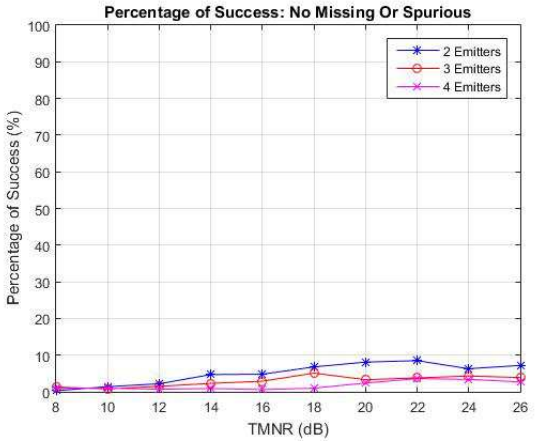


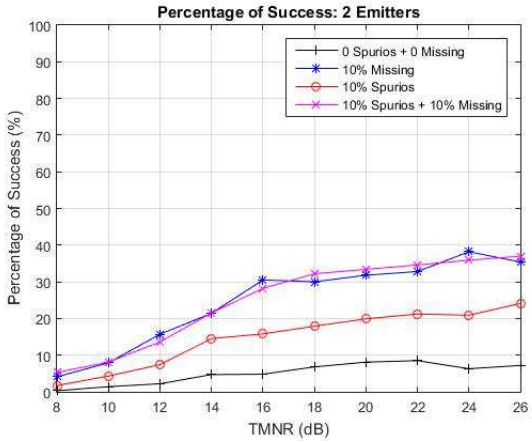


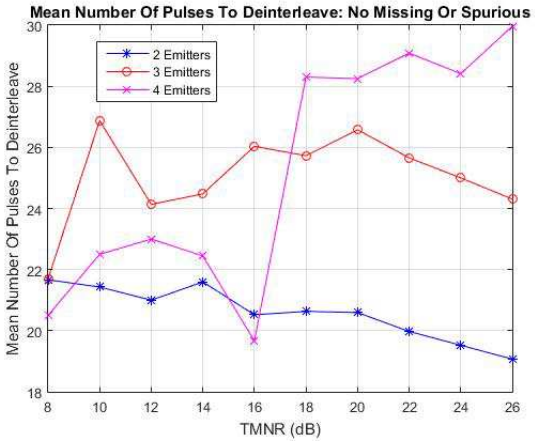

Mean Number Of Pulses To Deinterleave: No Missing Or Spurious
Mean Number Of Pulses To Deinterleave
TMNR (dB)
2 Emitters
3 Emitters
4 Emitters

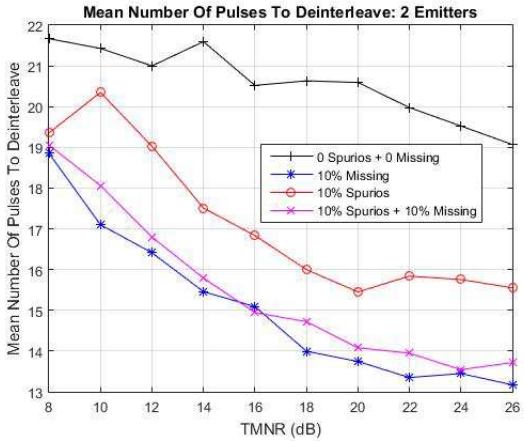

Mean Number Of Pulses To Deinterleave: 2 Emitters
Mean Number Of Pulses To Deinterleave
TMNR (dB)
0 Spurios + 0 Missing
10% Missing
10% Spurios
10% Spurios + 10% Missing

SDIF

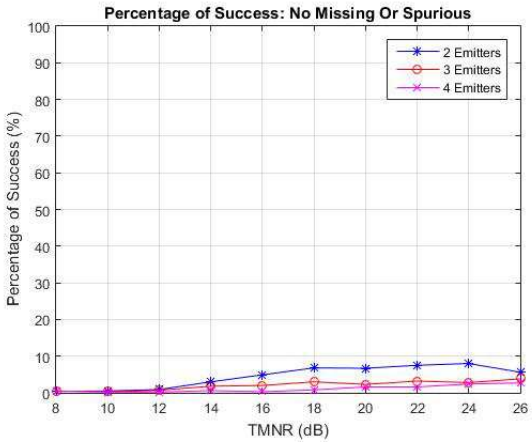

Percentage of Success: No Missing Or Spurious
Percentage of Success (%)
TMNR (dB)
2 Emitters
3 Emitters
4 Emitters

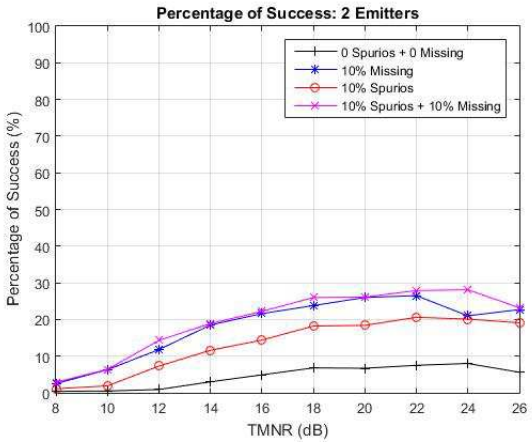

Percentage of Success: 2 Emitters
Percentage of Success (%)
TMNR (dB)
0 Spurios + 0 Missing
10% Missing
10% Spurios
10% Spurios + 10% Missing

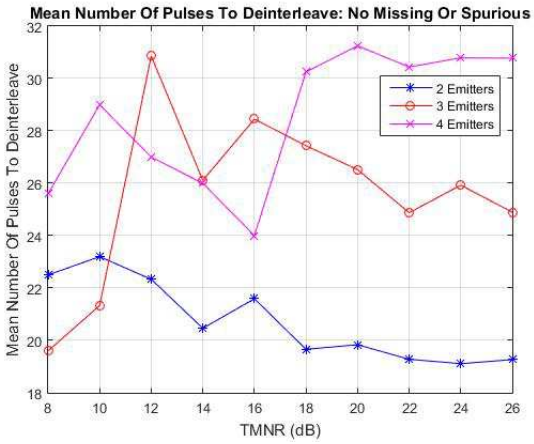

Mean Number Of Pulses To Deinterleave: No Missing Or Spurious
Mean Number Of Pulses To Deinterleave
TMNR (dB)
2 Emitters
3 Emitters
4 Emitters

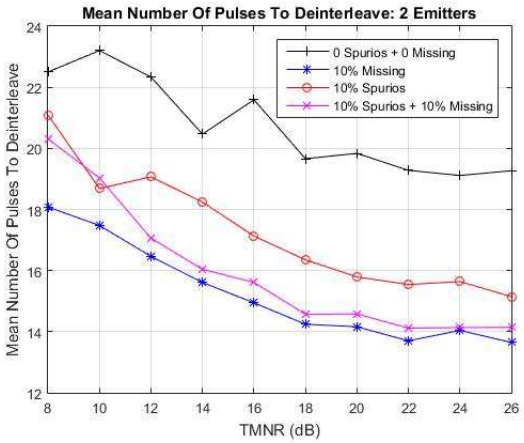

Mean Number Of Pulses To Deinterleave: 2 Emitters
Mean Number Of Pulses To Deinterleave
TMNR (dB)
0 Spurios + 0 Missing
10% Missing
10% Spurios
10% Spurios + 10% Missing

SDIF SS

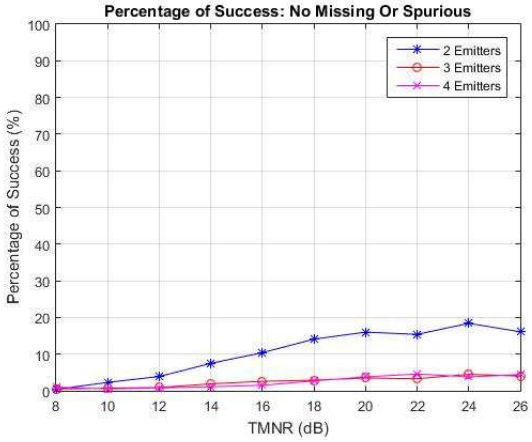


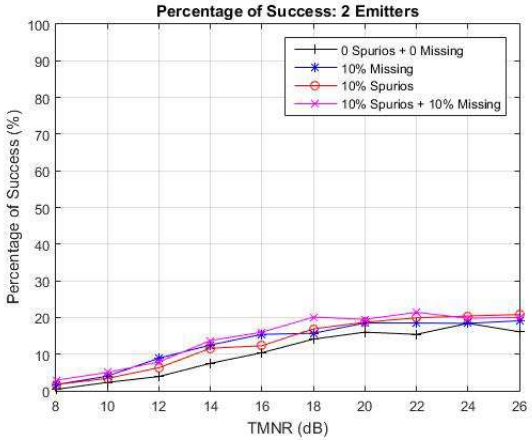


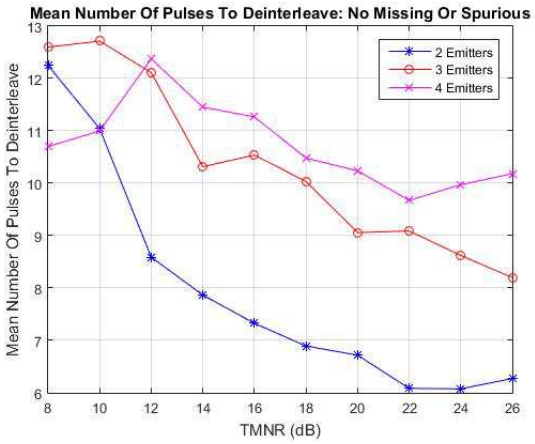


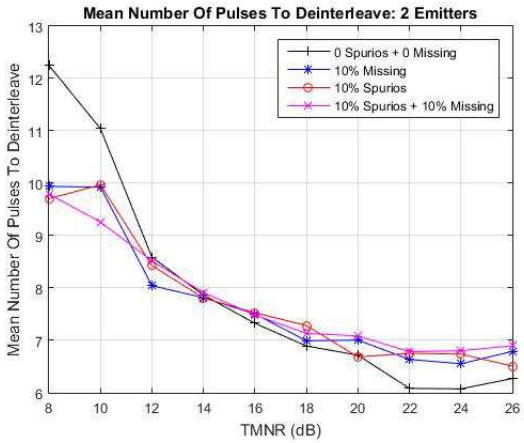


Sequence Search






Mean Number Of Pulses To Deinterleave: No Missing Or Spurious
2 Emitters
3 Emitters
4 Emitters
Mean Number Of Pulses To Deinterleave
TMNR (dB)

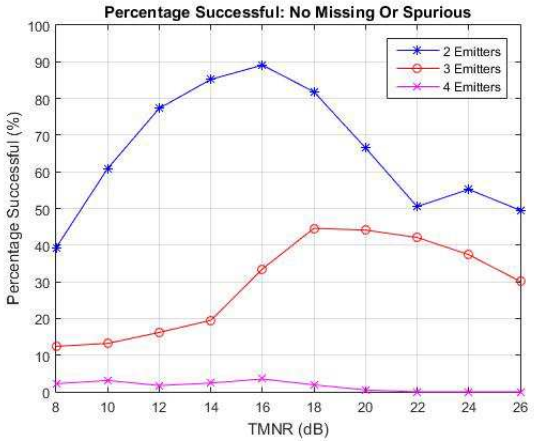

Mean Number Of Pulses To Deinterleave: 2 Emitters
0 Spurios + 0 Missing
10% Missing
10% Spurios
10% Spurios + 10% Missing
Mean Number Of Pulses To Deinterleave
TMNR (dB)

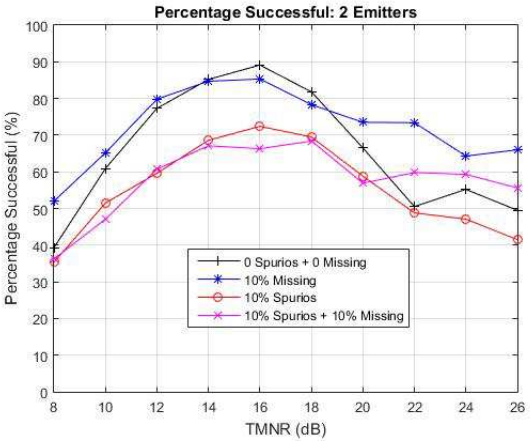
Appendix D

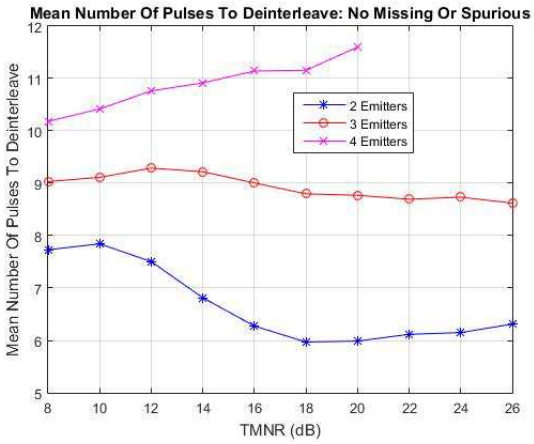
Emitter Deinterleaving - Hardware Results

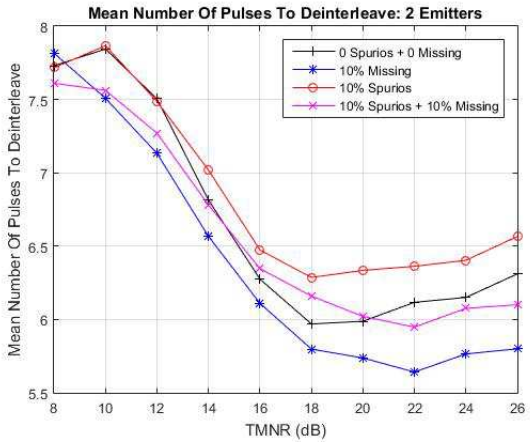
Constant PRI Signals

CDIF

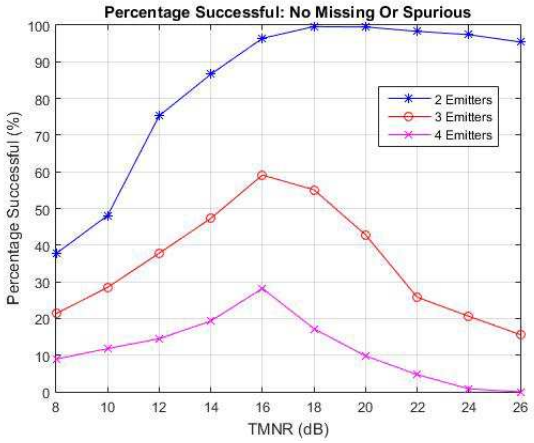


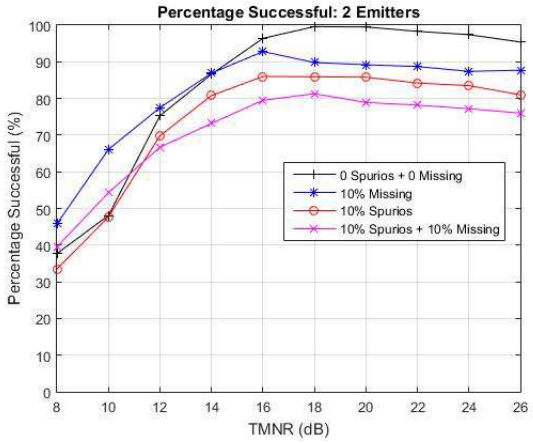


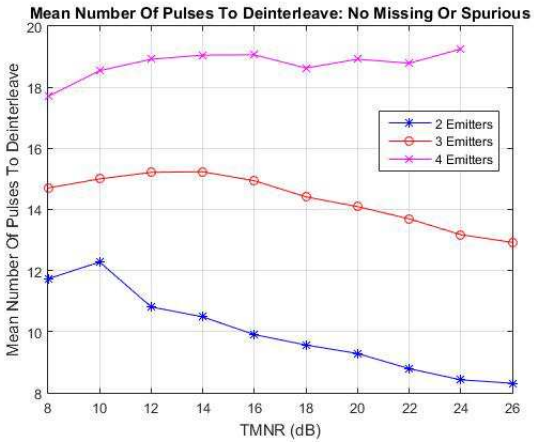


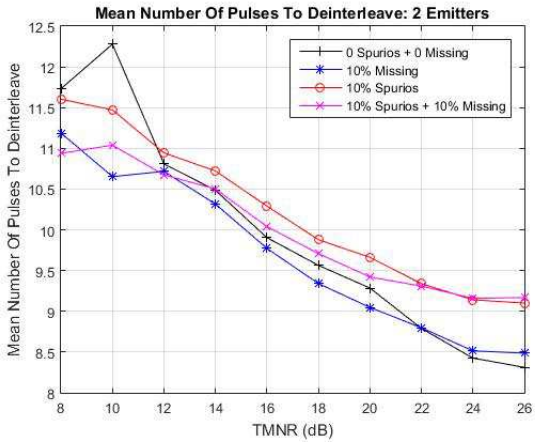


CDIF SS



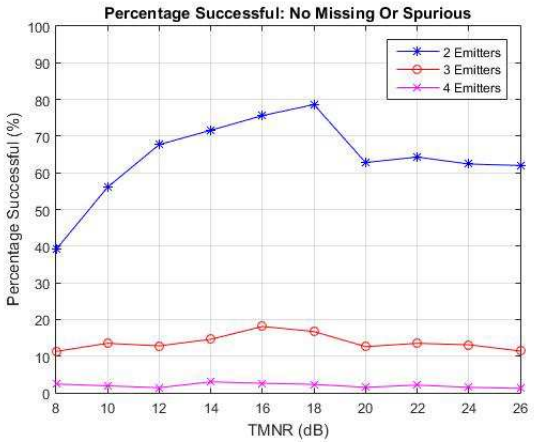


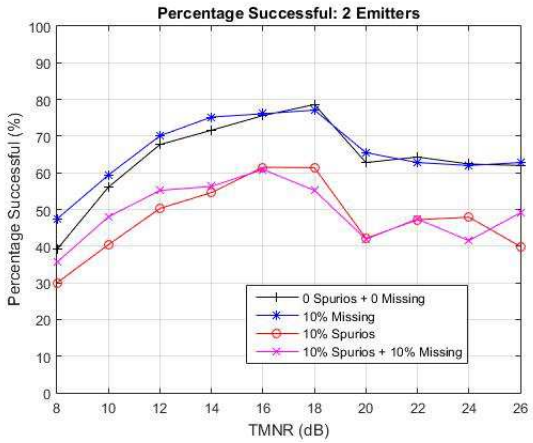


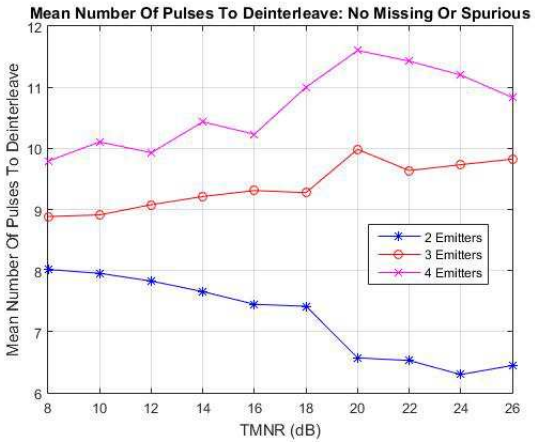


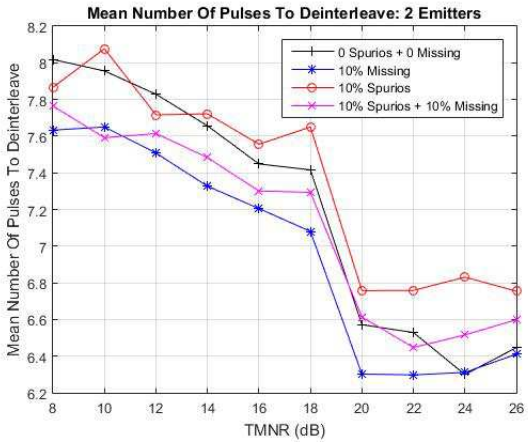
Jittered PRI Signals

CDIF

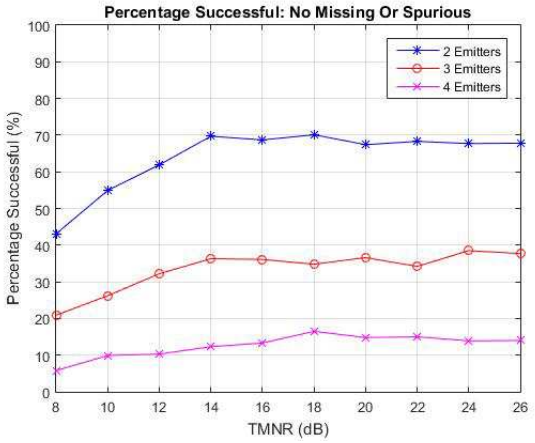


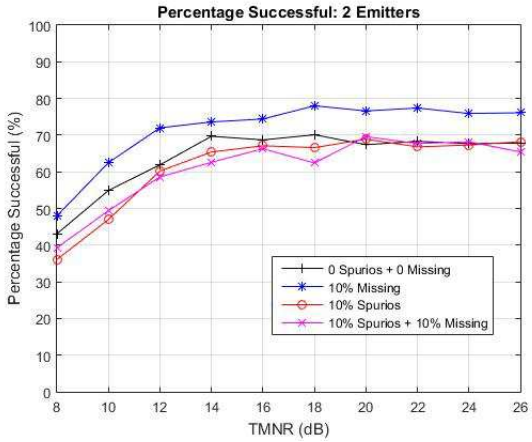


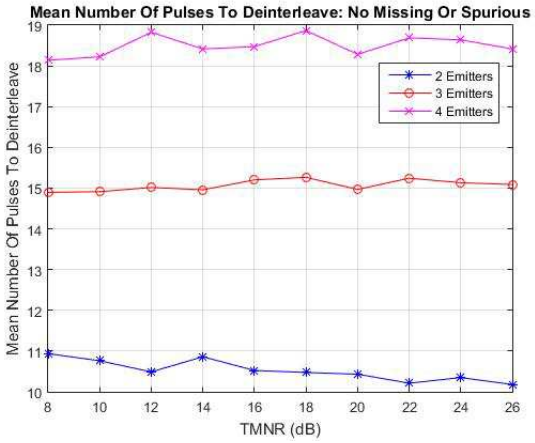

Mean Number Of Pulses To Deinterleave: No Missing Or Spurious
Mean Number Of Pulses To Deinterleave
TMNR (dB)
2 Emitters
3 Emitters
4 Emitters

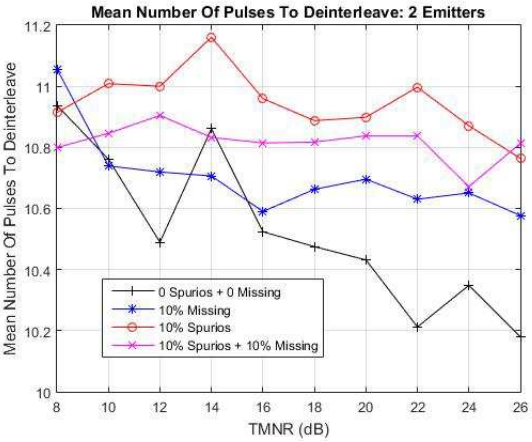

Mean Number Of Pulses To Deinterleave: 2 Emitters
Mean Number Of Pulses To Deinterleave
TMNR (dB)
0 Spurios + 0 Missing
10% Missing
10% Spurios
10% Spurios + 10% Missing

CDIF SS


Percentage Successful: No Missing Or Spurious
Percentage Successful (%)
TMNR (dB)
2 Emitters
3 Emitters
4 Emitters

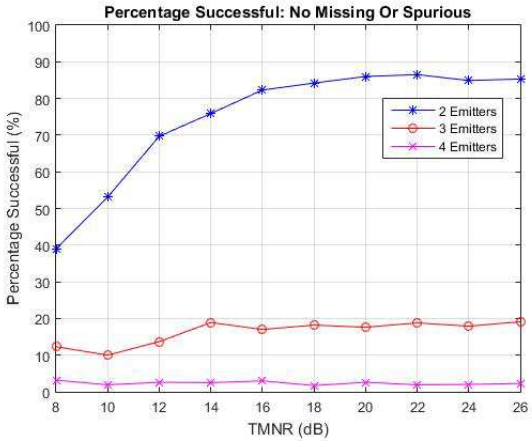

Percentage Successful: 2 Emitters
Percentage Successful (%)
TMNR (dB)
0 Spurios + 0 Missing
10% Missing
10% Spurios
10% Spurios + 10% Missing

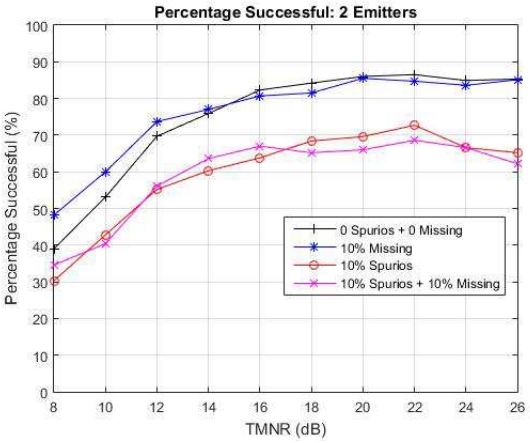

Mean Number Of Pulses To Deinterleave: No Missing Or Spurious
Mean Number Of Pulses To Deinterleave
TMNR (dB)
2 Emitters
3 Emitters
4 Emitters

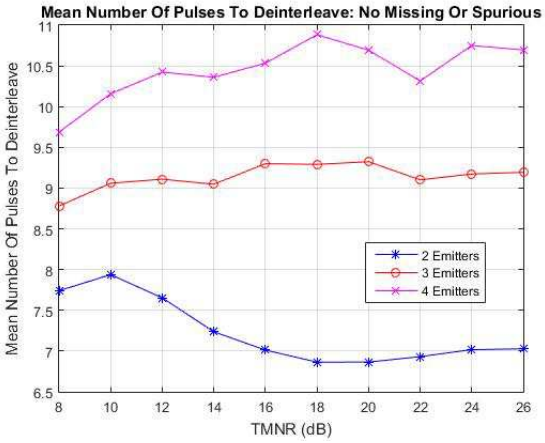

Mean Number Of Pulses To Deinterleave: 2 Emitters
Mean Number Of Pulses To Deinterleave
TMNR (dB)
0 Spurios + 0 Missing
10% Missing
10% Spurios
10% Spurios + 10% Missing

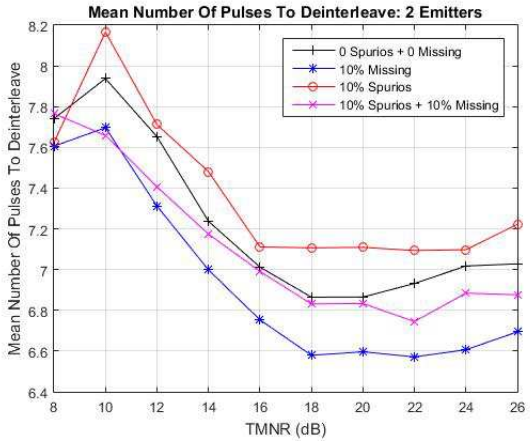
Mixed PRI Signals

CDIF

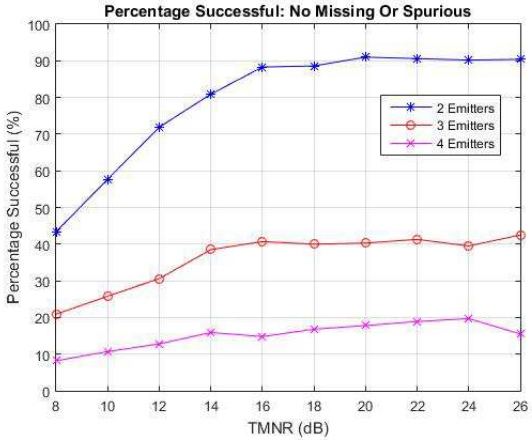


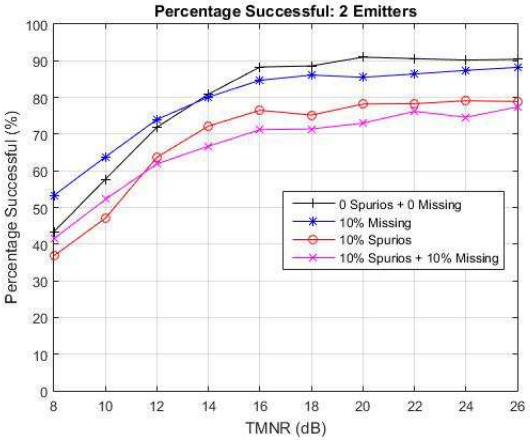


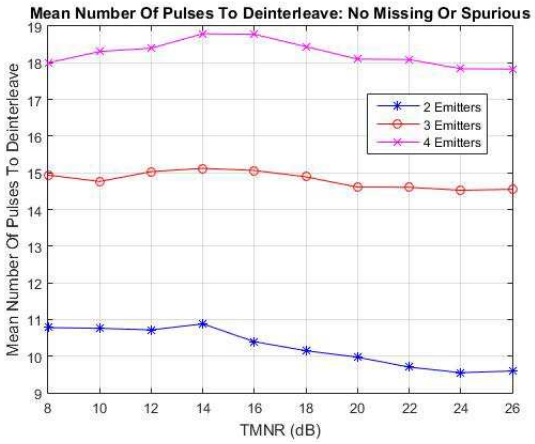


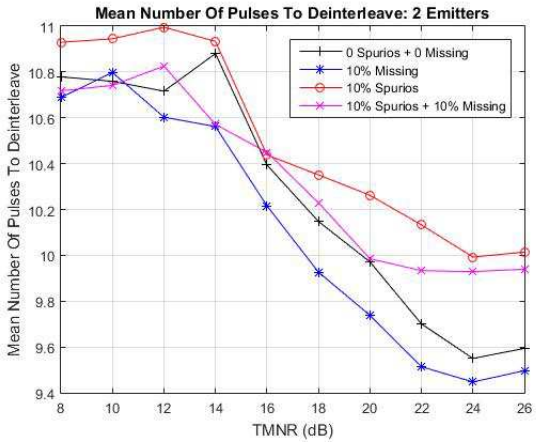


CDIF SS






Mean Number Of Pulses To Deinterleave: No Missing Or Spurious
Mean Number Of Pulses To Deinterleave
TMNR (dB)
2 Emitters
3 Emitters
4 Emitters

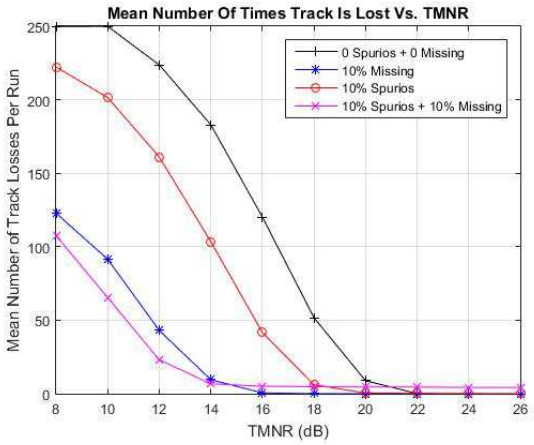

Mean Number Of Pulses To Deinterleave: 2 Emitters
Mean Number Of Pulses To Deinterleave
TMNR (dB)
0 Spurios + 0 Missing
10% Missing
10% Spurios
10% Spurios + 10% Missing

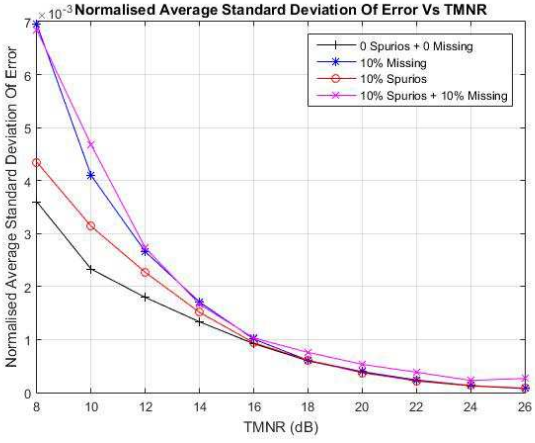
Appendix E

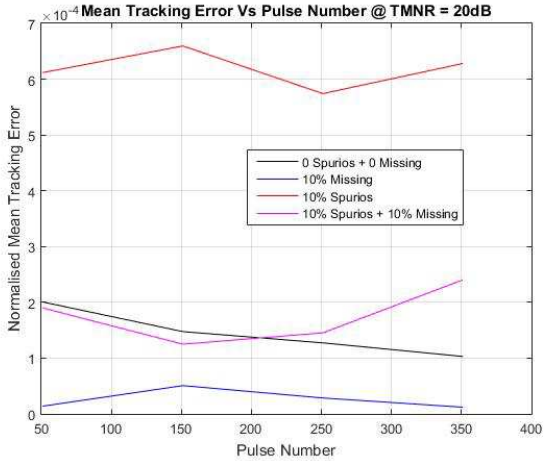
Emitter TOA Tracking - Simulation Results

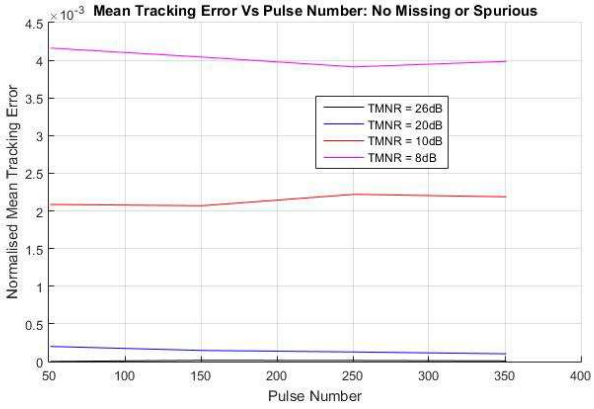
Constant PRI Scheme

Delta-τ Histogram

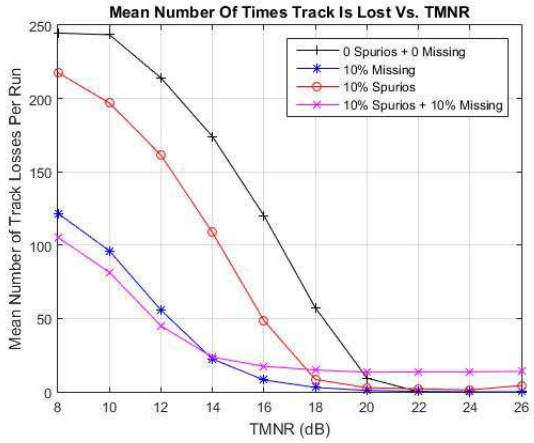


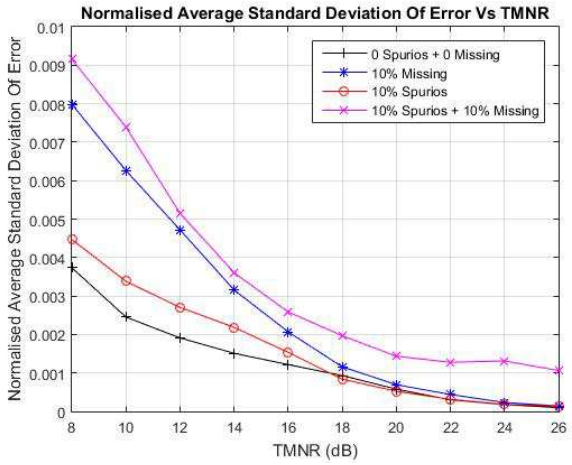


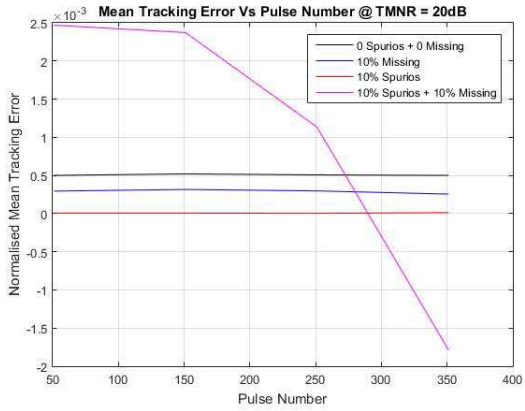


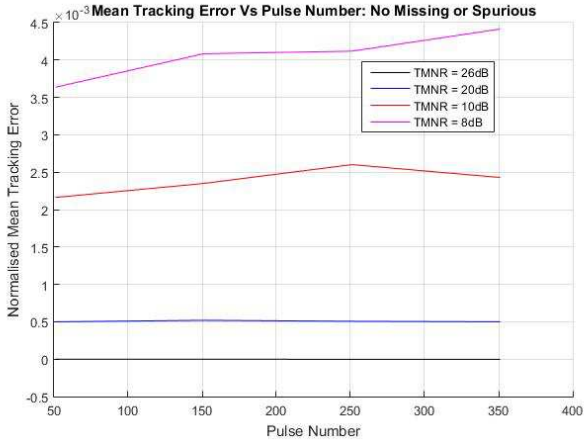


Alpha-Beta Filter

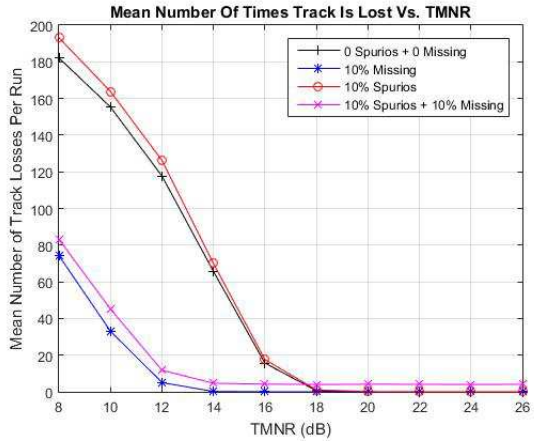


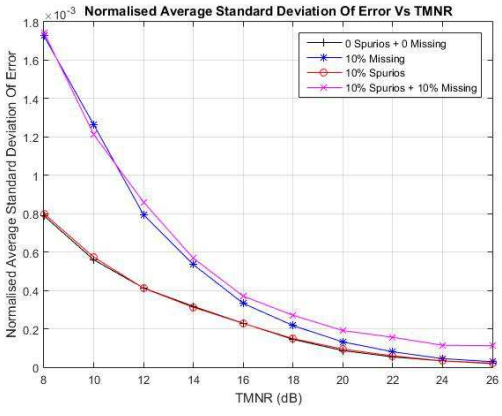


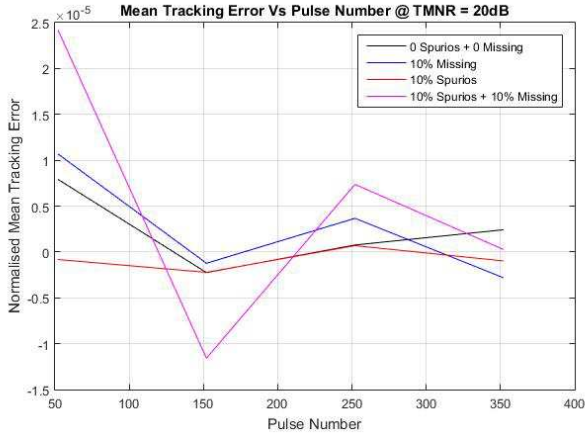


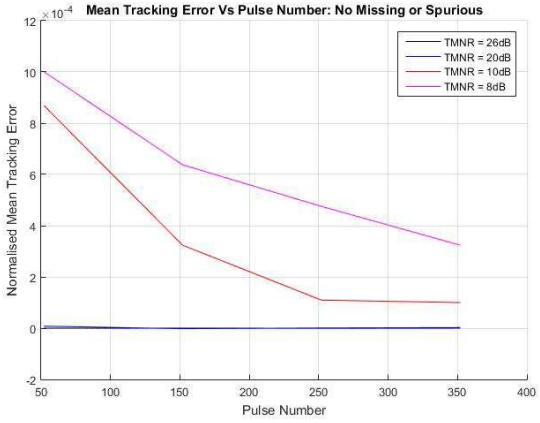


Kalman Filter - Time Domain

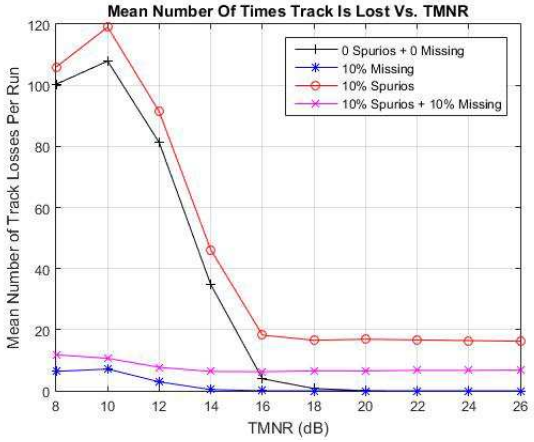


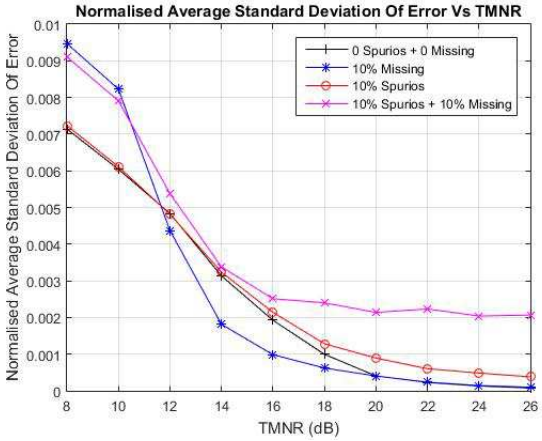


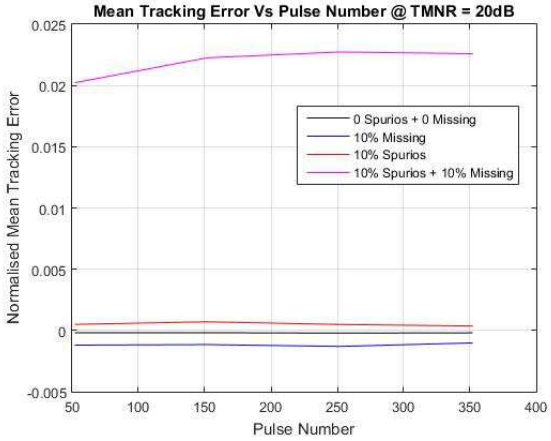


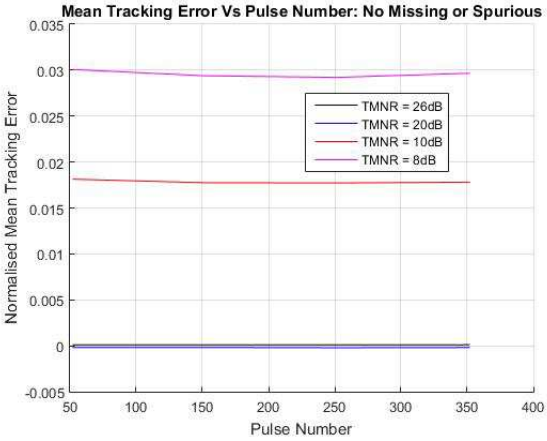


Kalman Filter - Fourier Domain



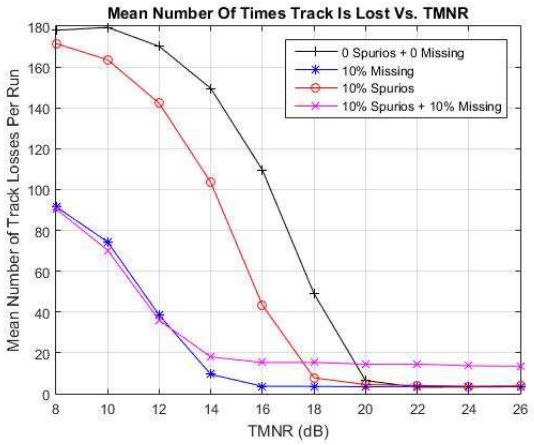


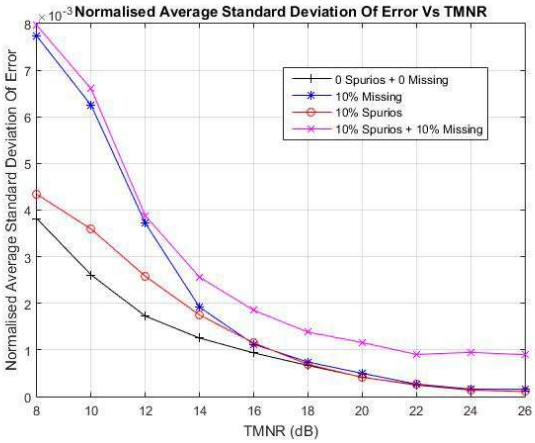


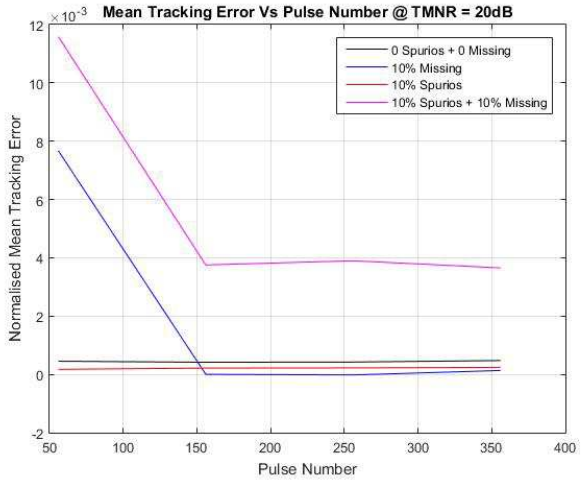


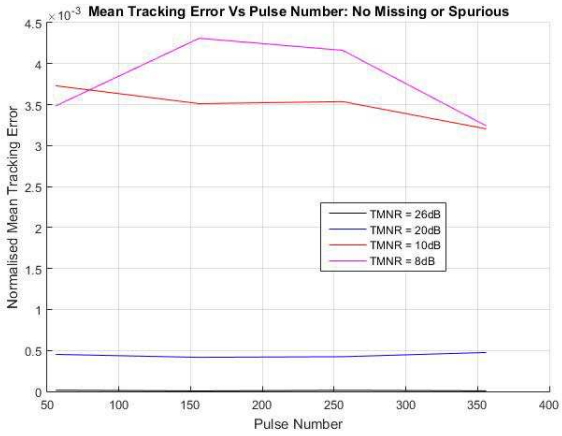
Dwell & Switch PRI Scheme

Delta-τ Histogram

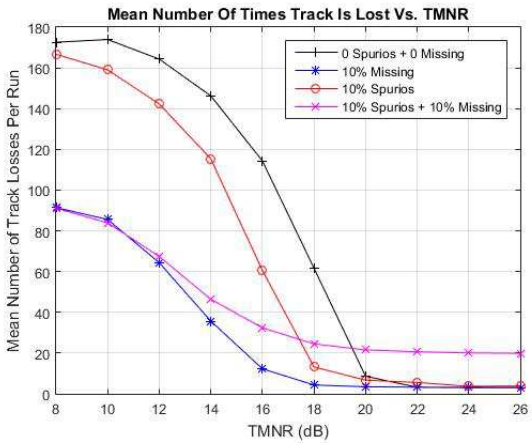


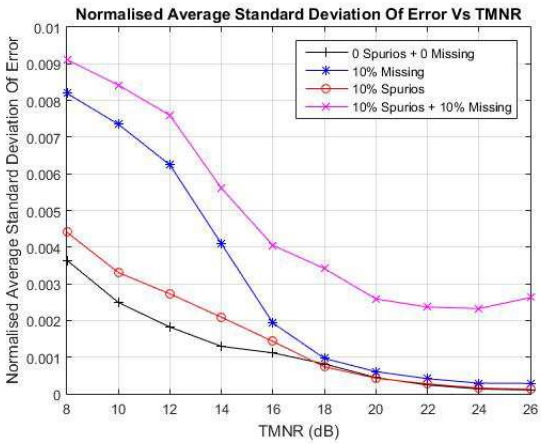


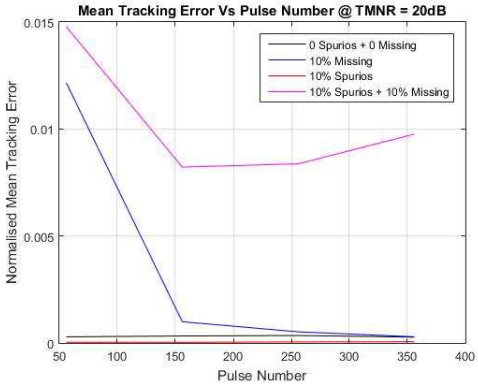


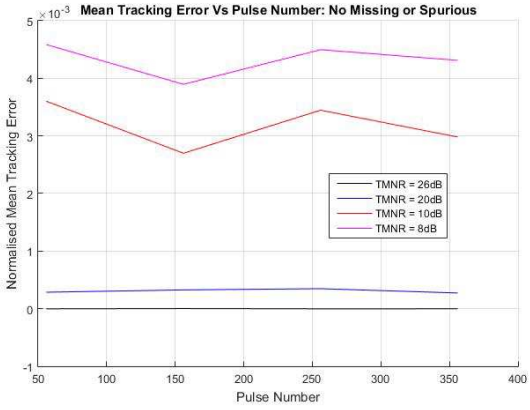


Alpha-Beta Filter

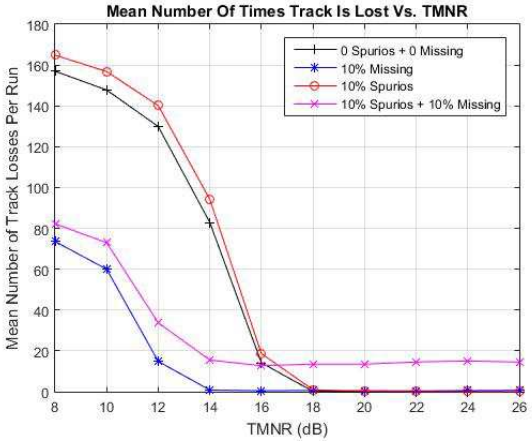


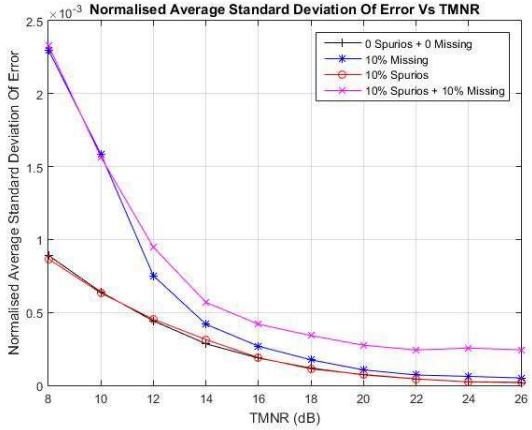


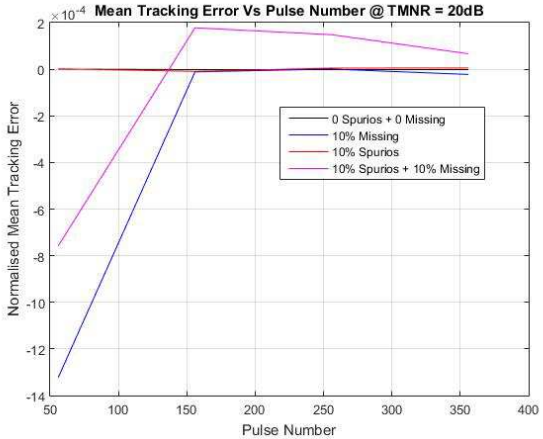


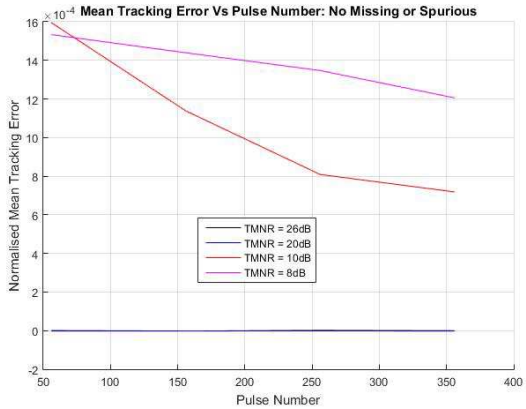


Kalman Filter - Time Domain

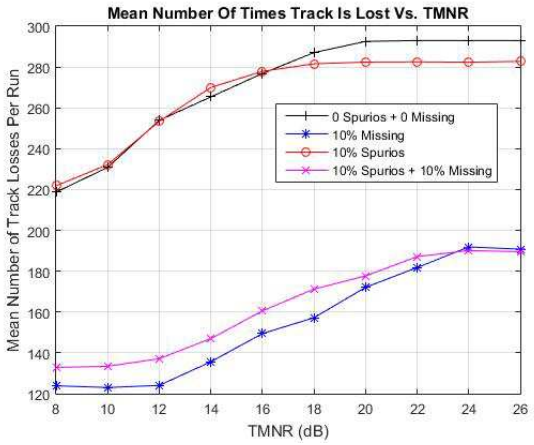


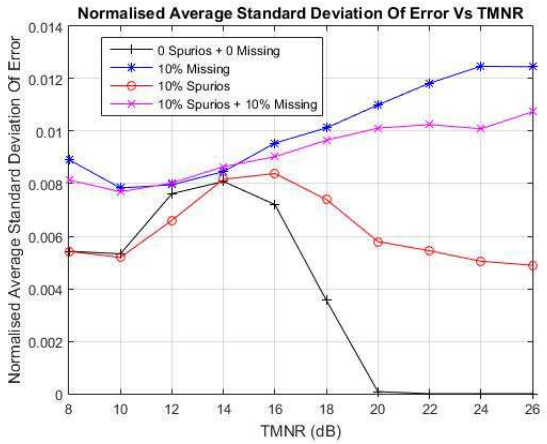


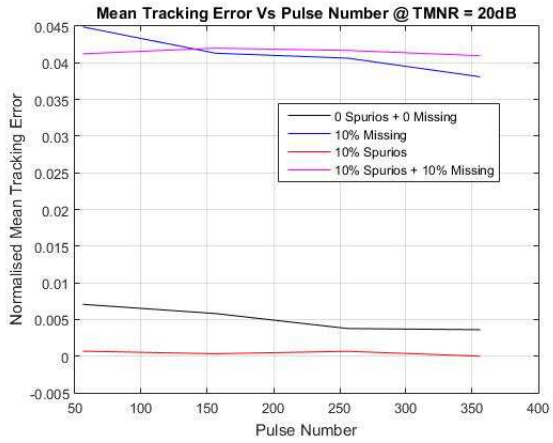


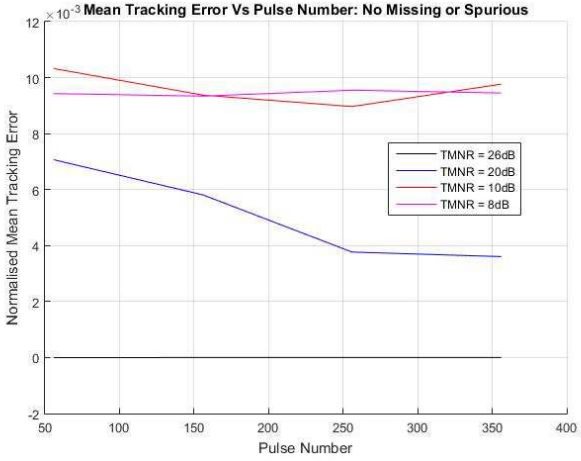


Kalman Filter - Fourier Domain



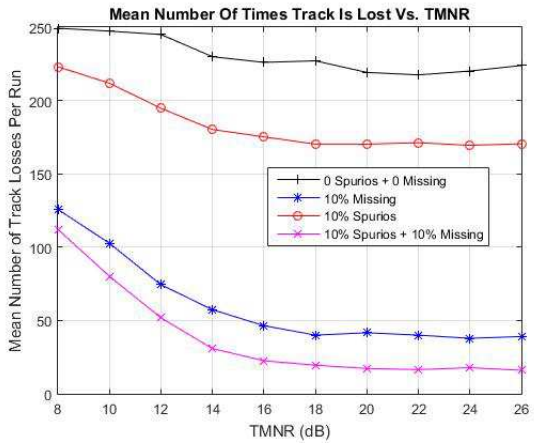


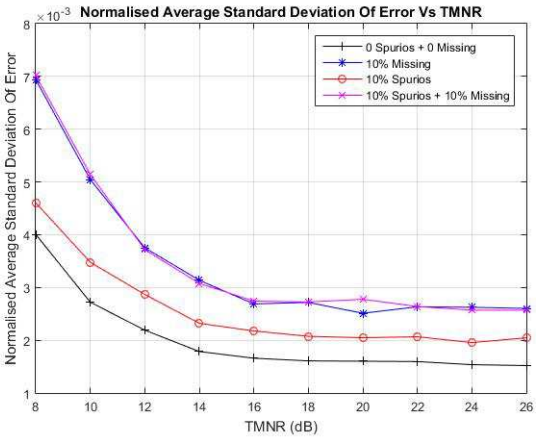


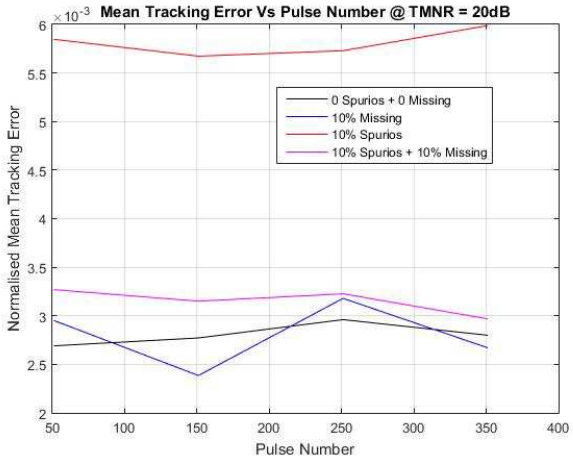


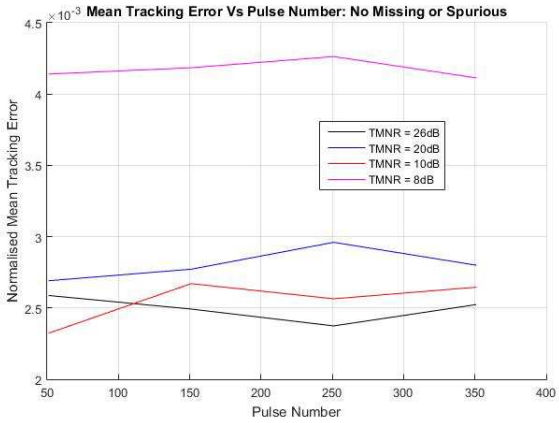
Jittered PRI Scheme

Delta-τ Histogram

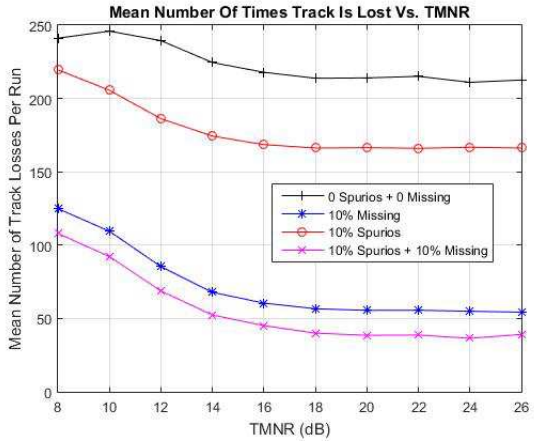


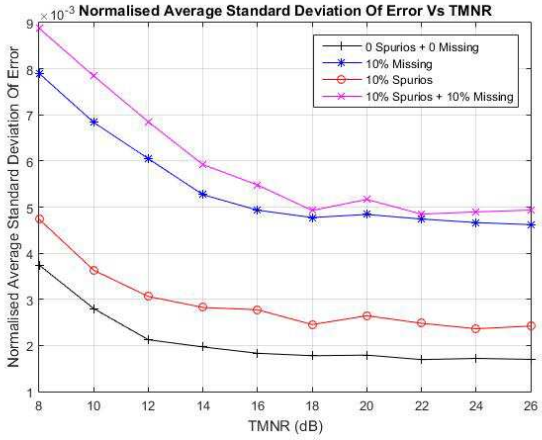


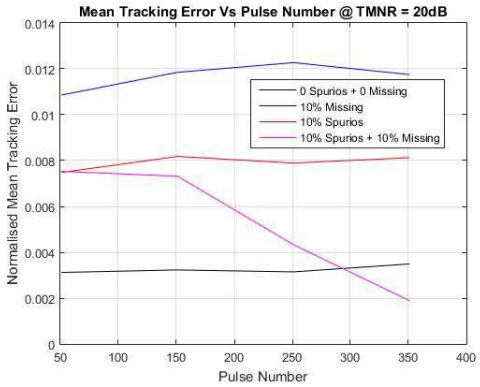


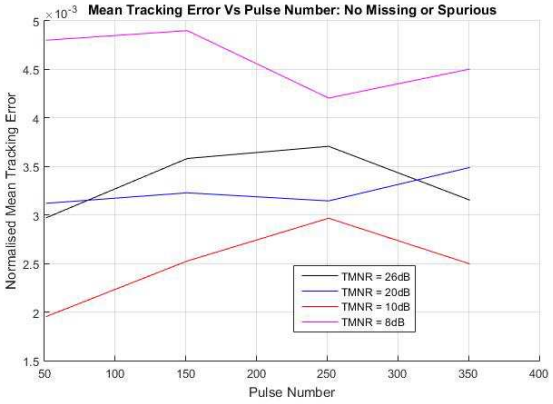


Alpha-Beta Filter

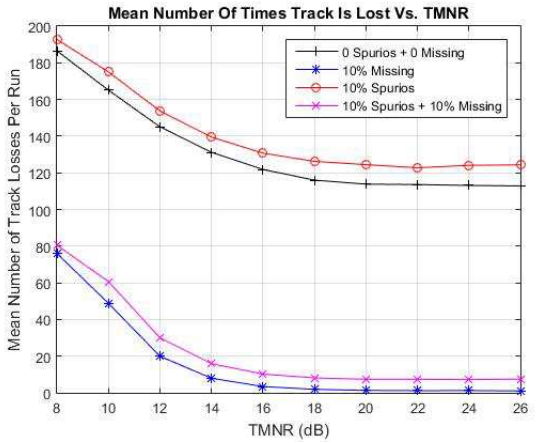


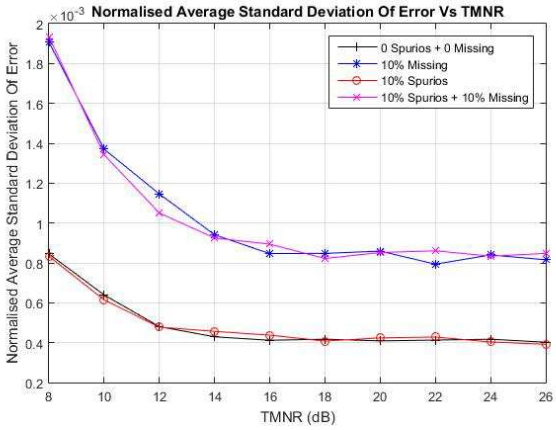


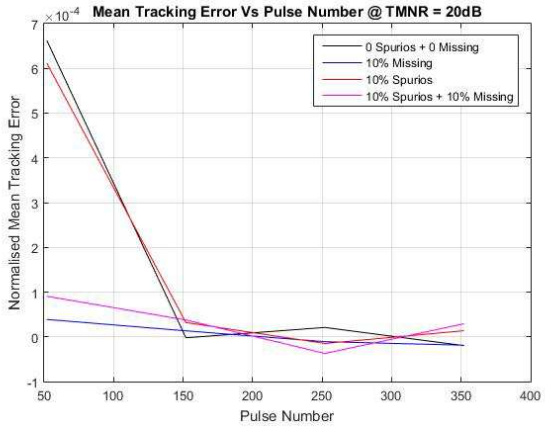


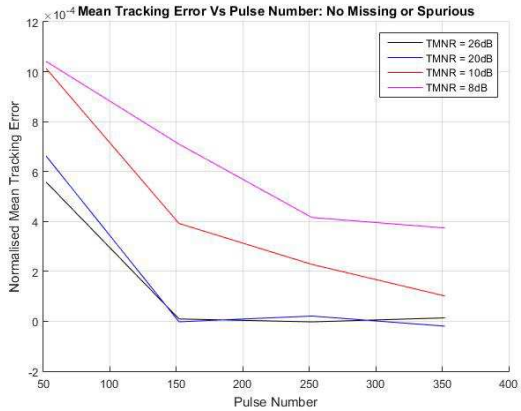


Kalman Filter - Time Domain

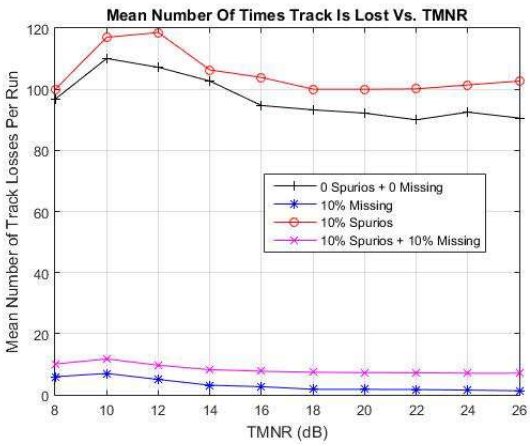


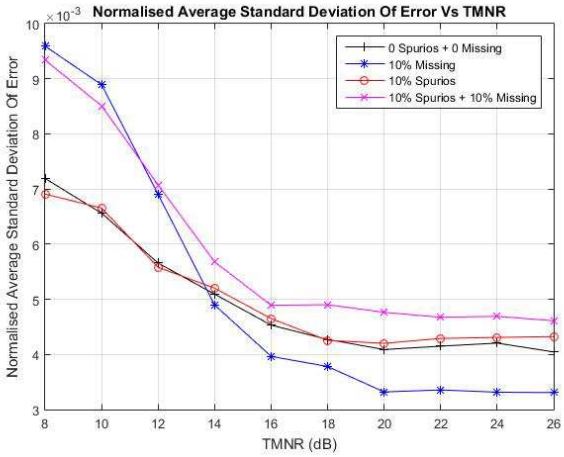


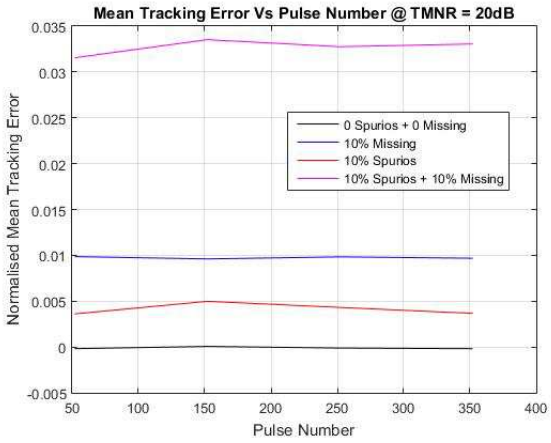


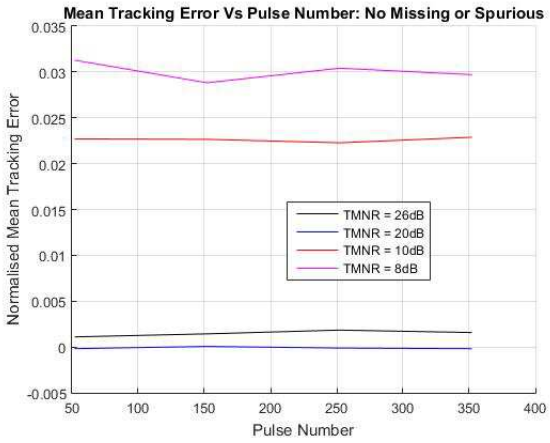


Kalman Filter - Fourier Domain



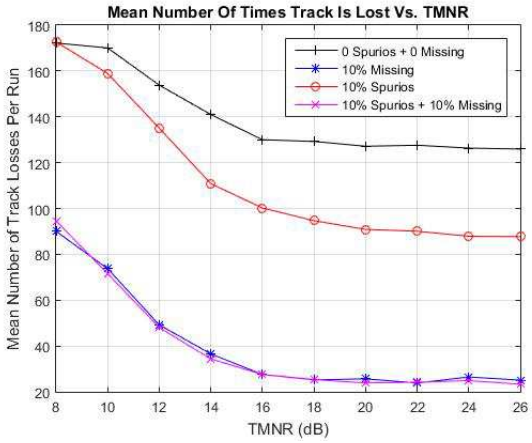


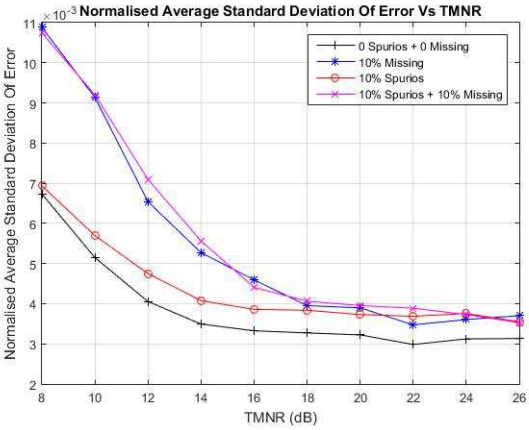


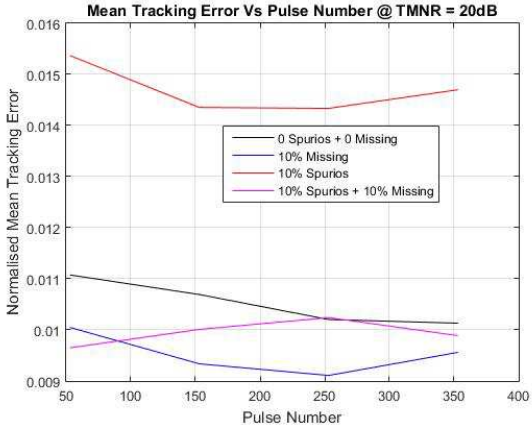


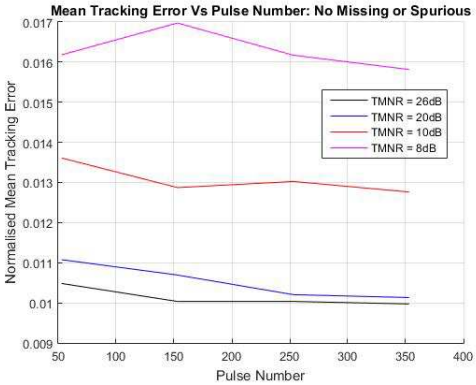
Staggered PRI Scheme

Delta-τ Histogram

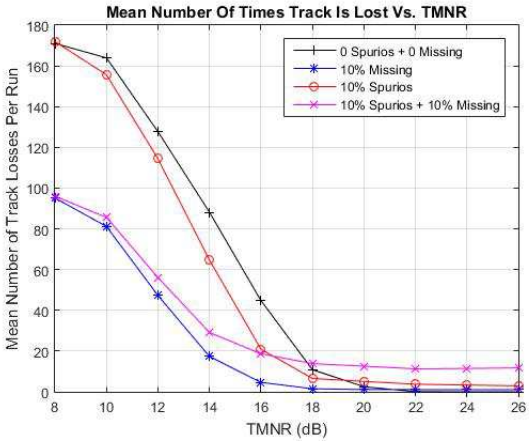


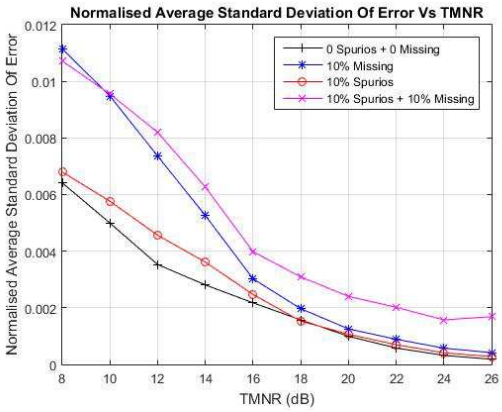


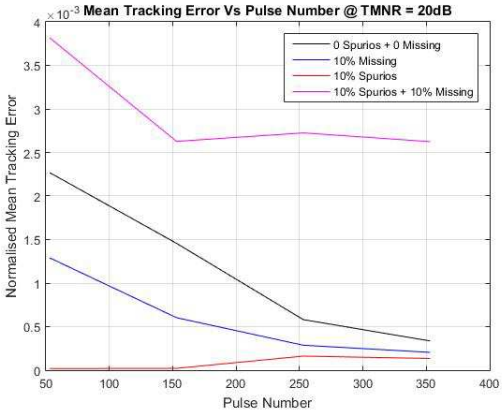


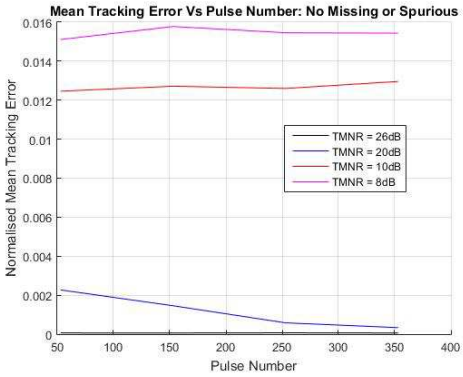


Alpha-Beta Filter

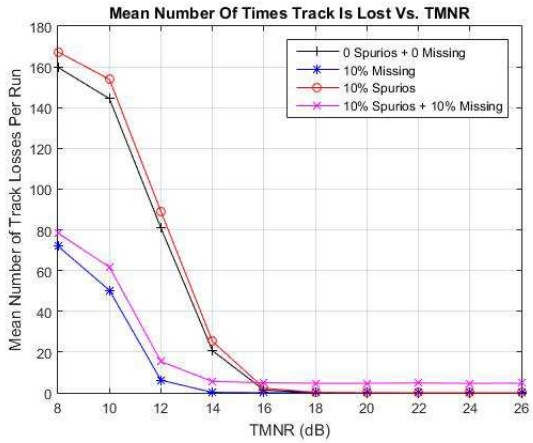


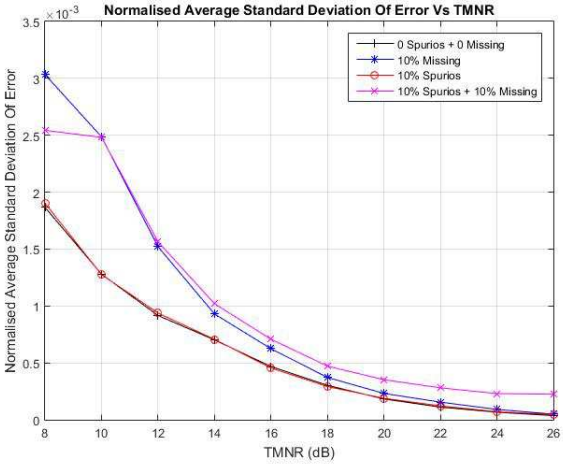


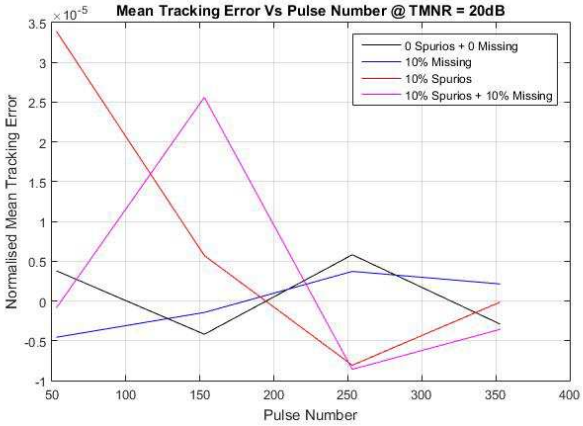


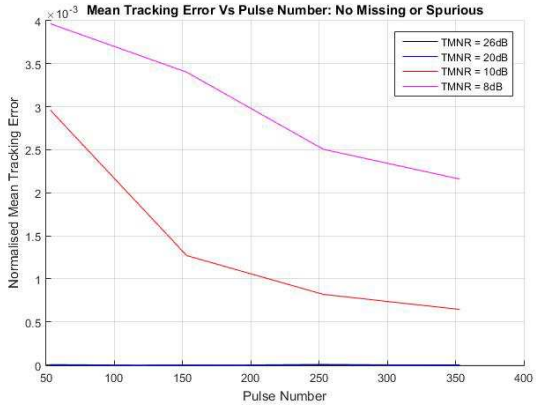


Kalman Filter - Time Domain

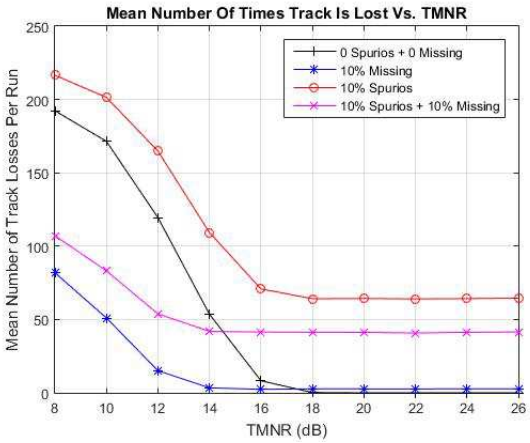


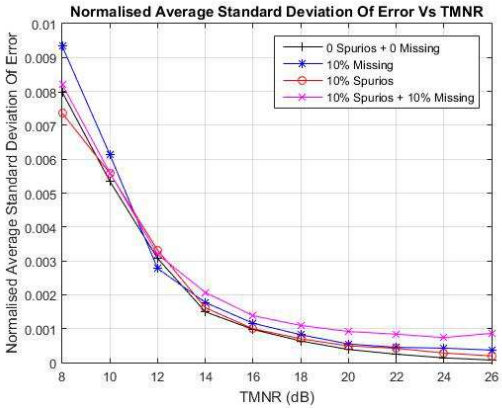


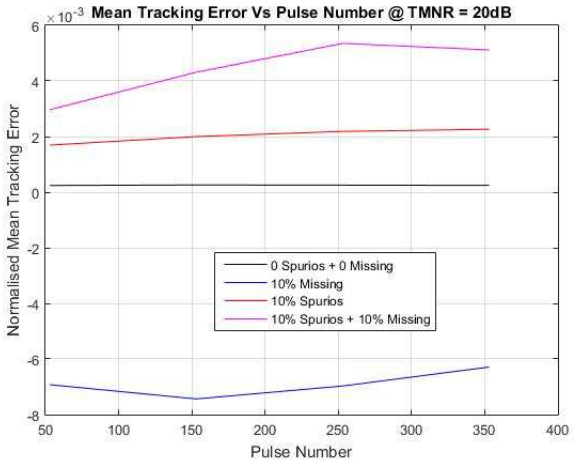


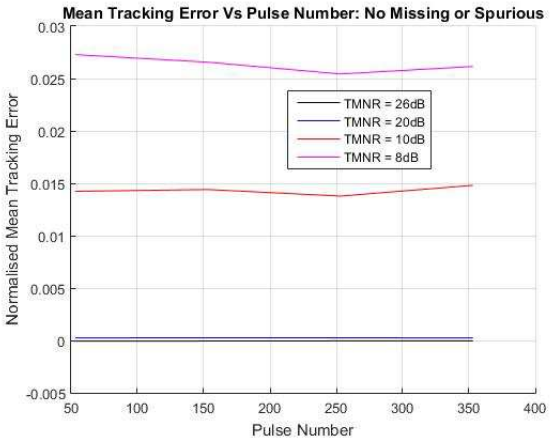


Kalman Filter - Fourier Domain







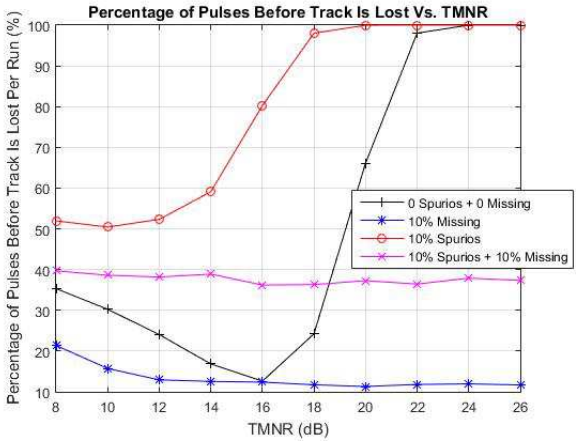


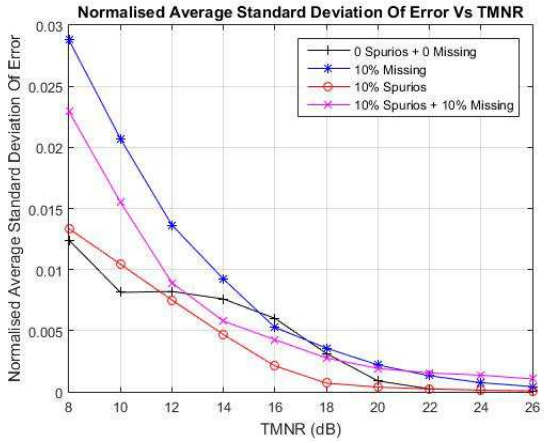
Appendix F

Emitter TOA Tracking - Hardware Results

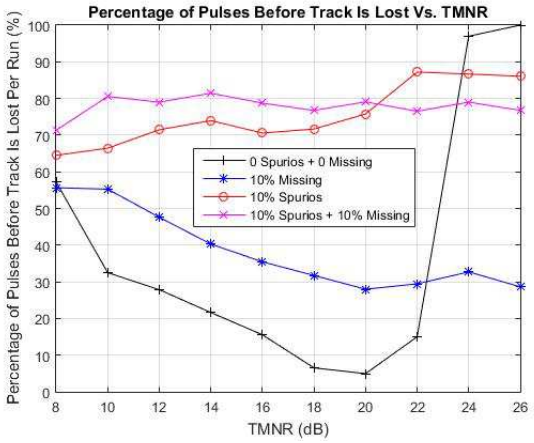
Constant PRI Scheme

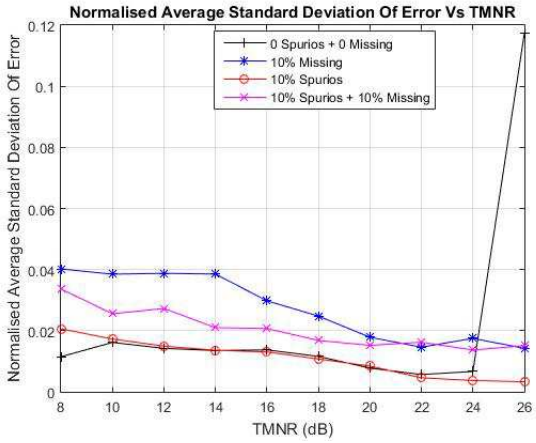
Delta-τ Histogram





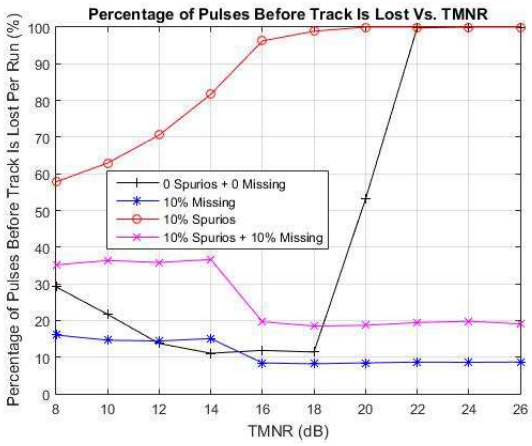
Alpha-Beta Filter

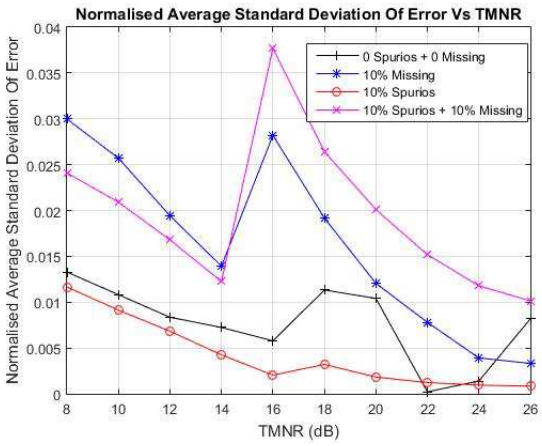




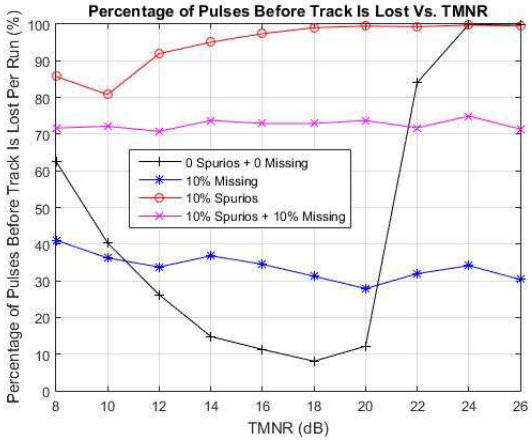
Dwell & Switch PRI Scheme

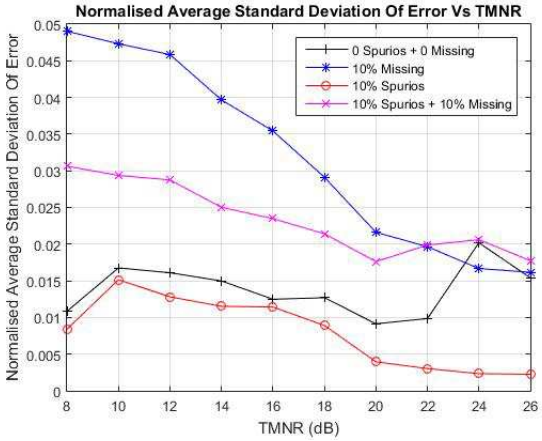
Delta-τ Histogram





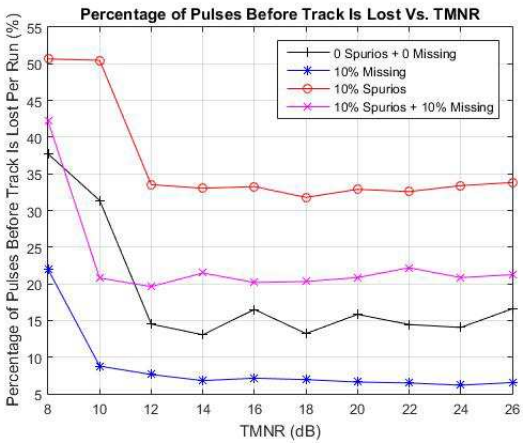
Alpha-Beta Filter

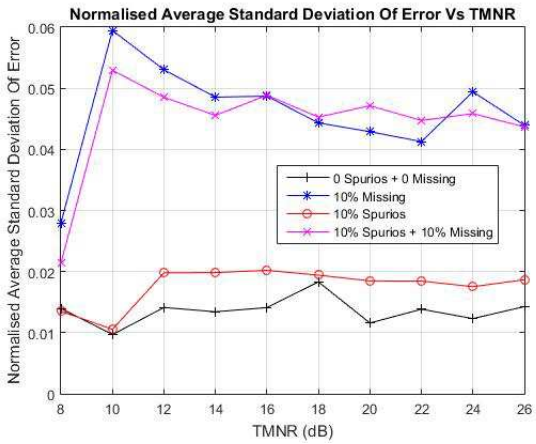




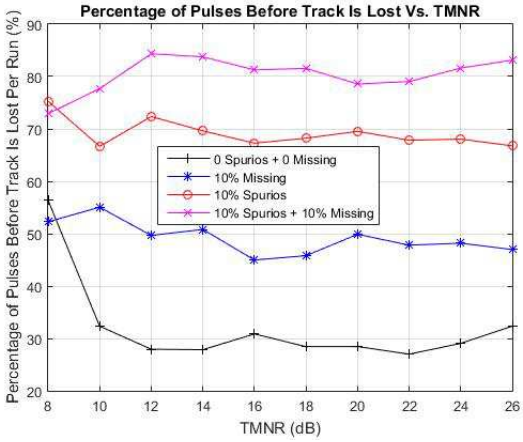
Jittered PRI Scheme

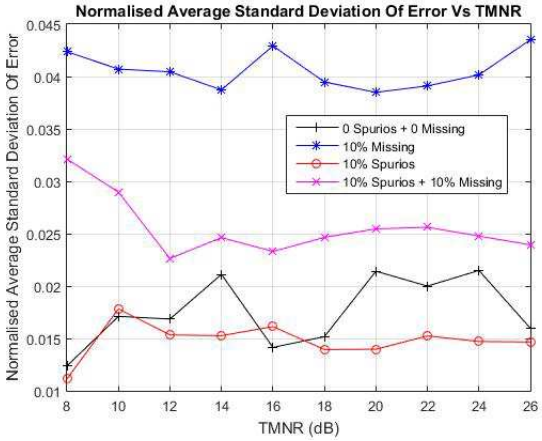
Delta-τ Histogram





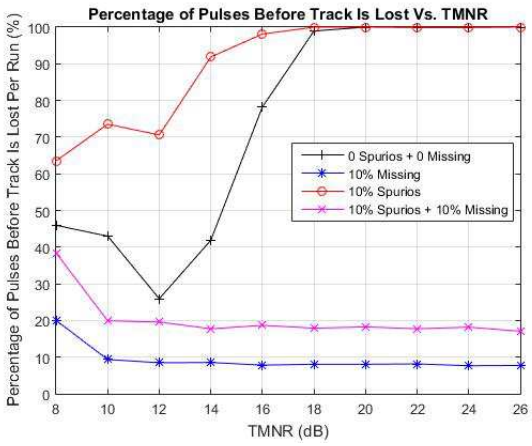
Alpha-Beta Filter

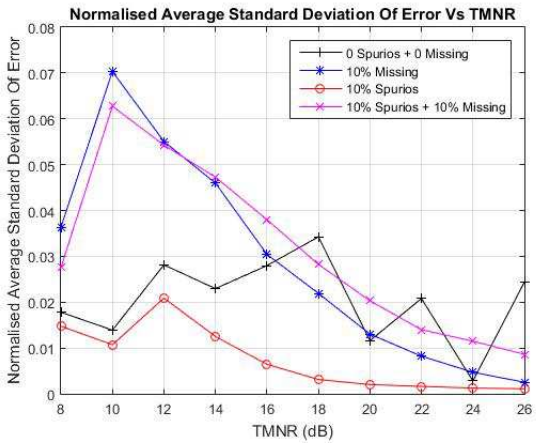




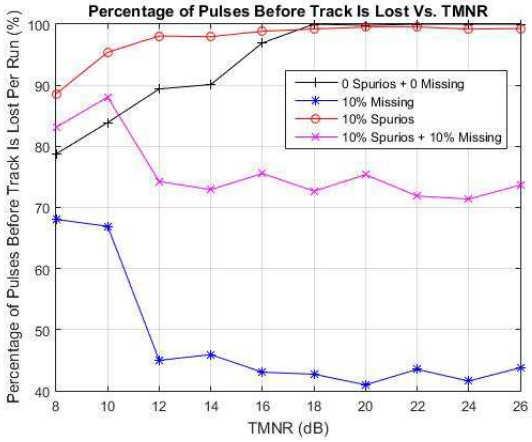
Staggered PRI Scheme

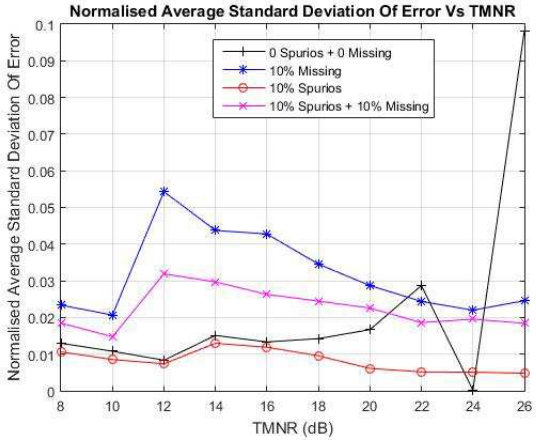
Delta-τ Histogram





Alpha-Beta Filter

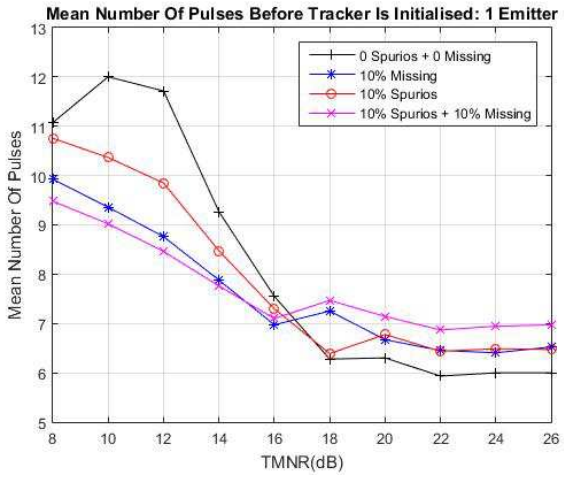


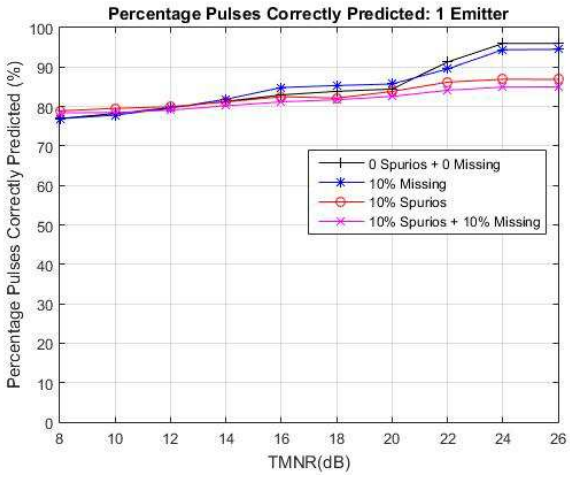


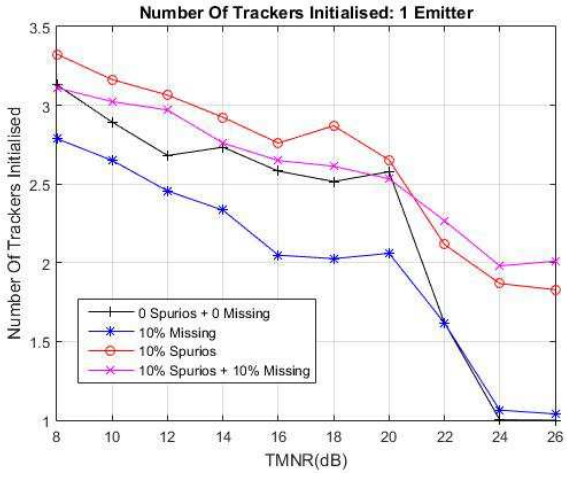
Appendix G

System - Single Emitter Results

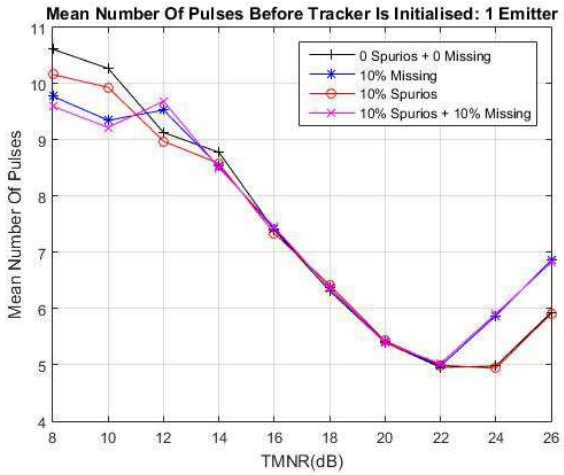
Constant Emitter Results

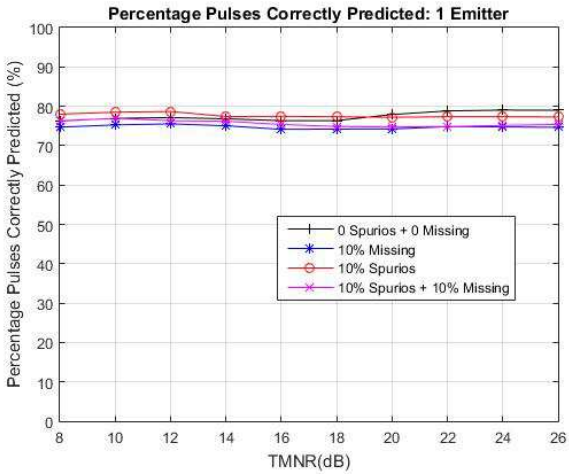


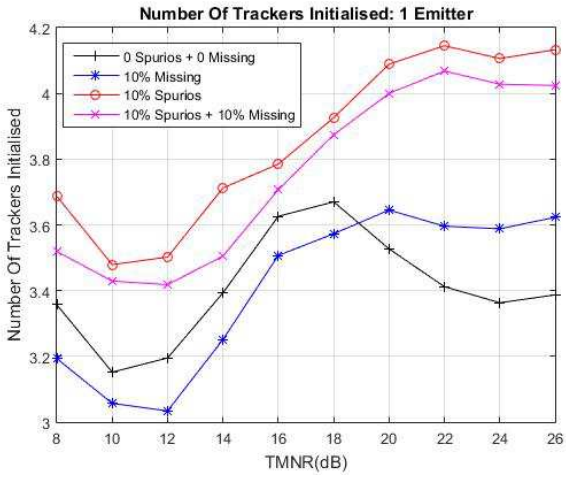




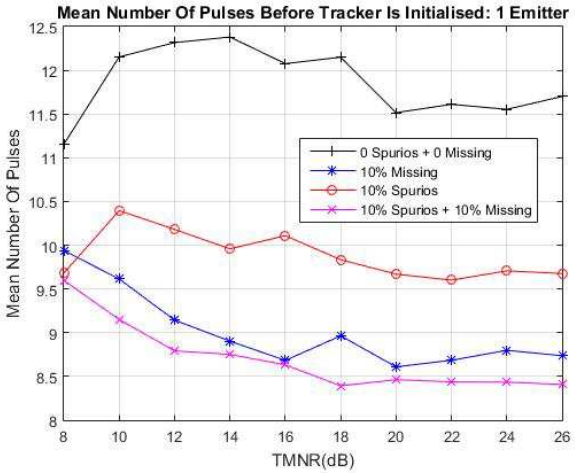
Dwell And Switch Emitter Results

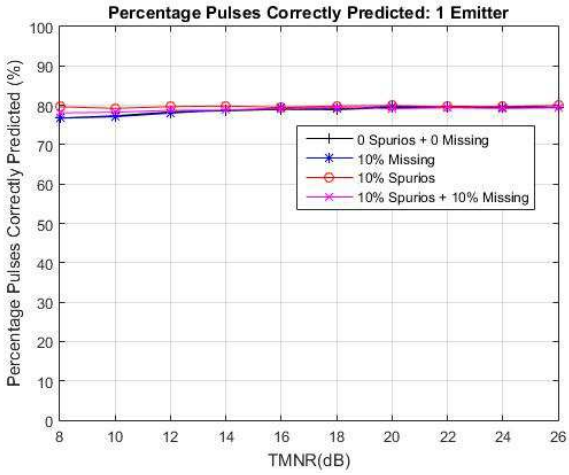


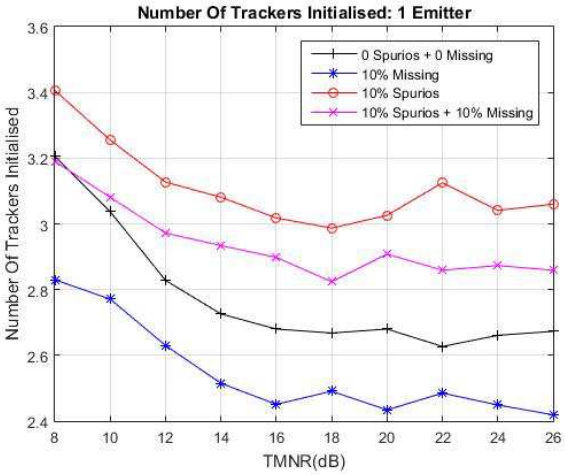




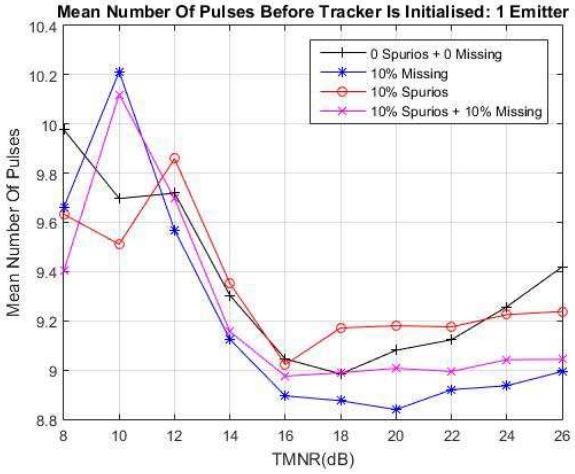
Jittered Emitter Results

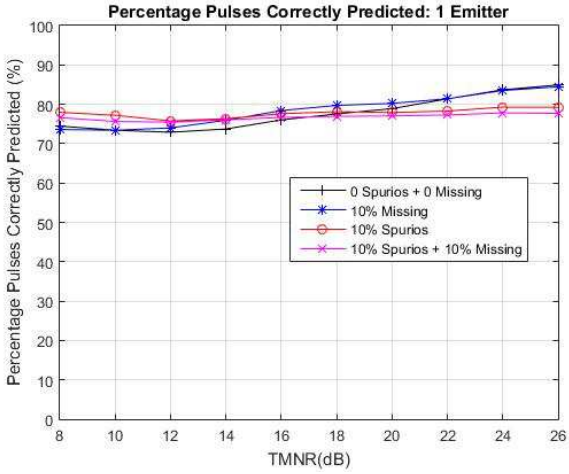


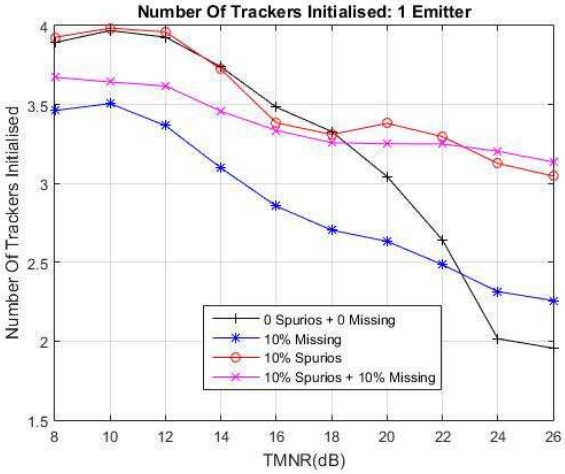




Staggered Emitter Results



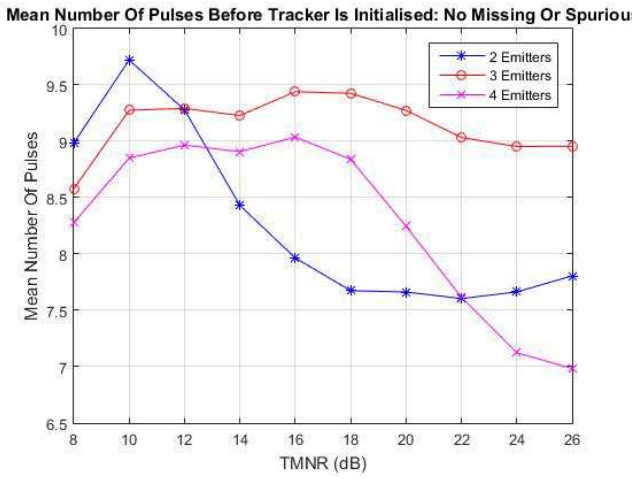


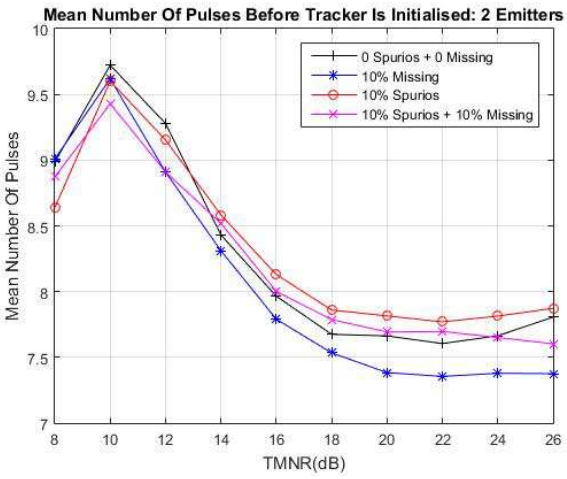


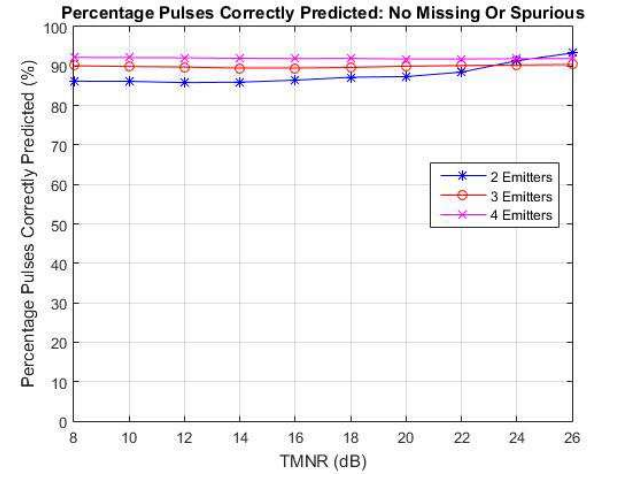
Appendix H

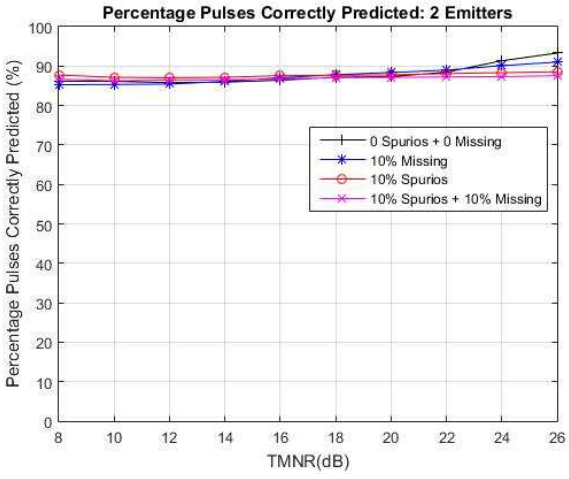
System - Multiple Emitters Results

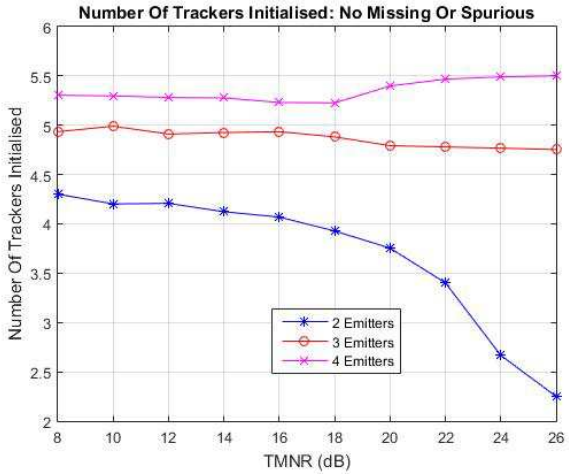
Constant Signals Results

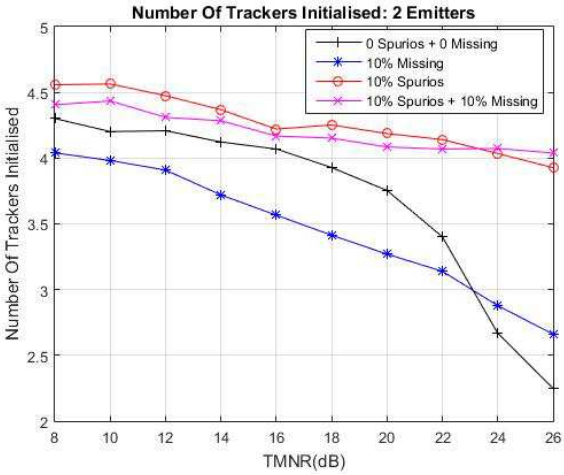




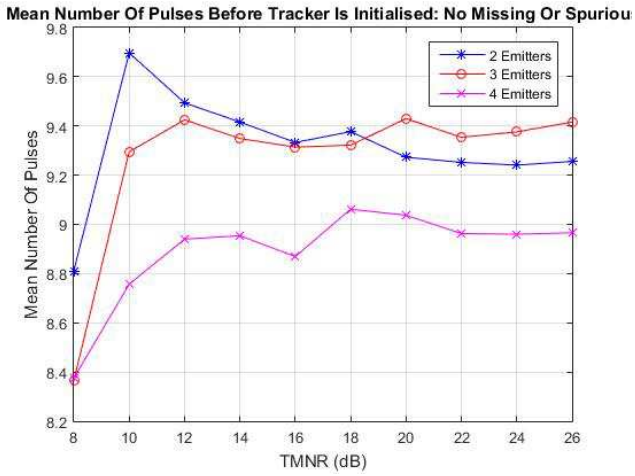


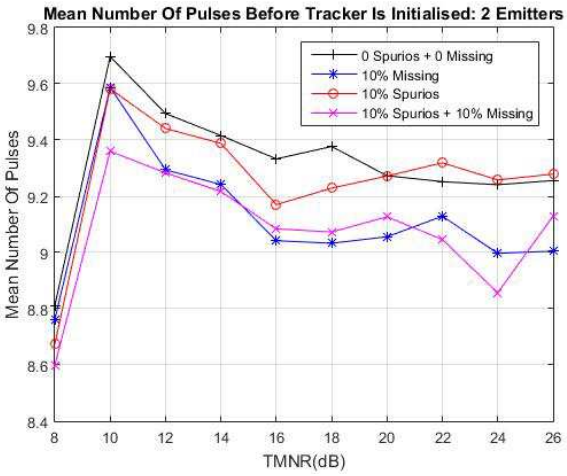


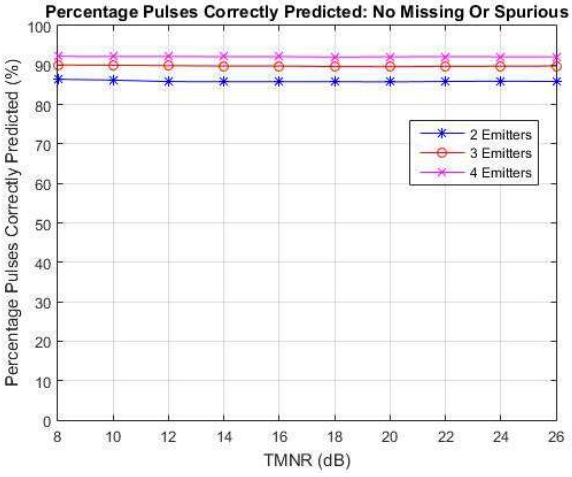


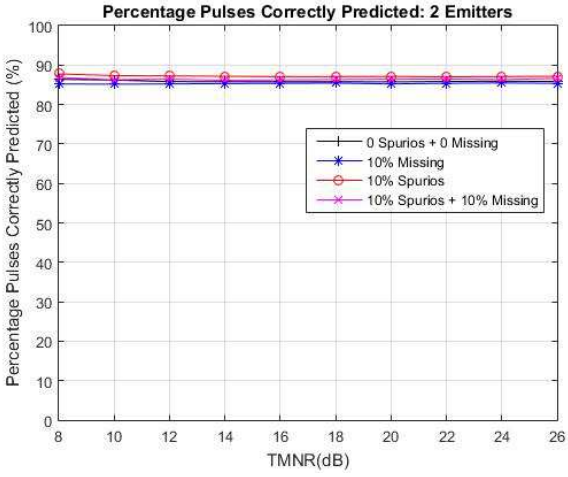


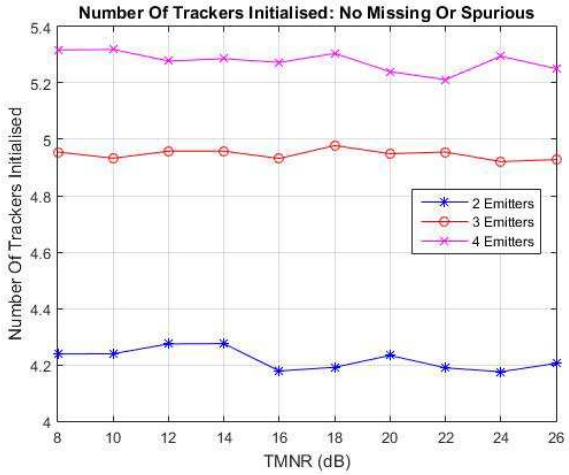
Jittered Signals Results

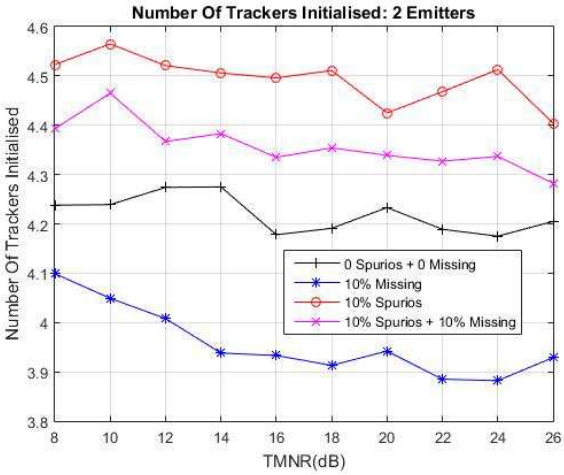




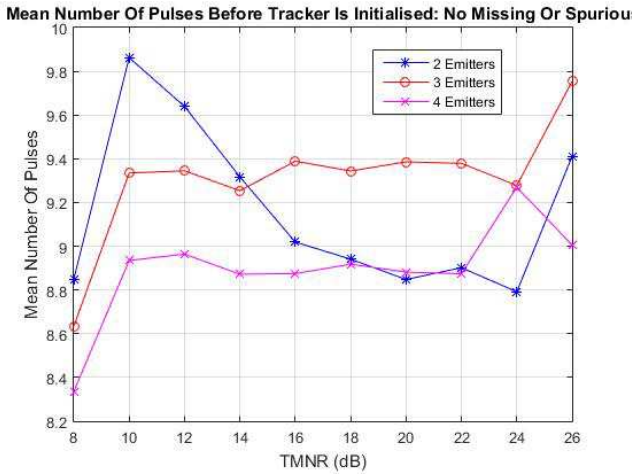


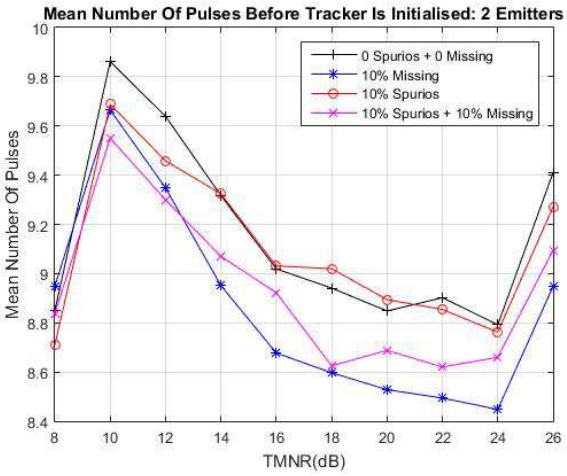


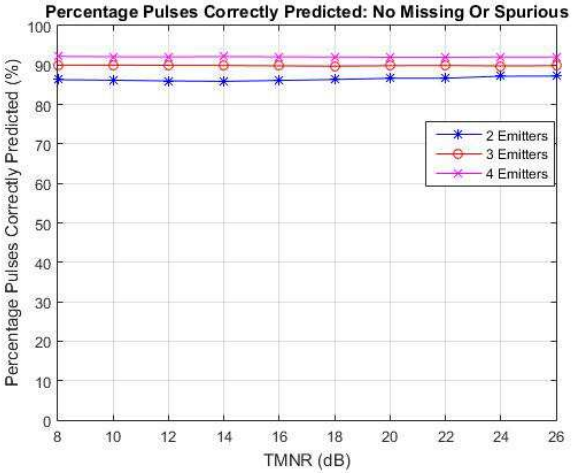


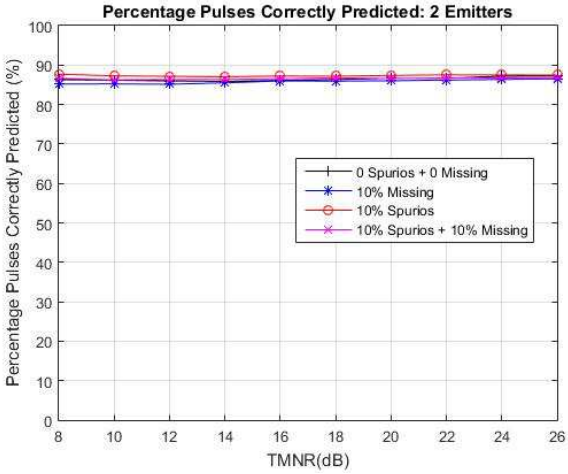


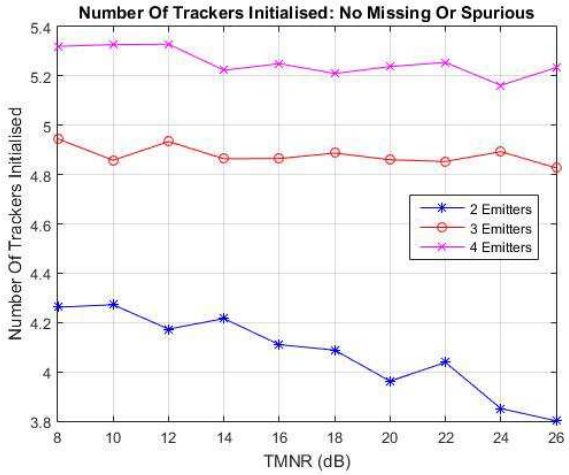
Mixed Signals Results

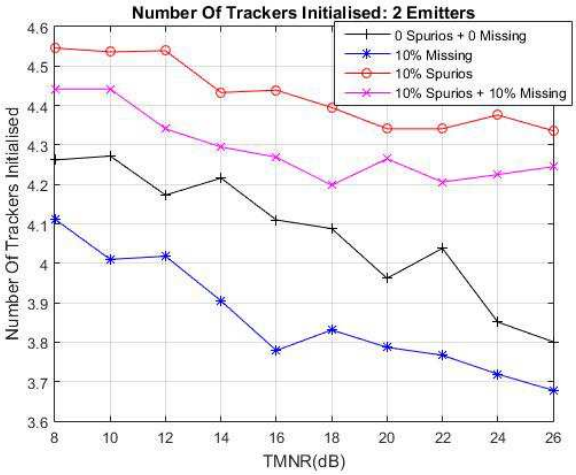




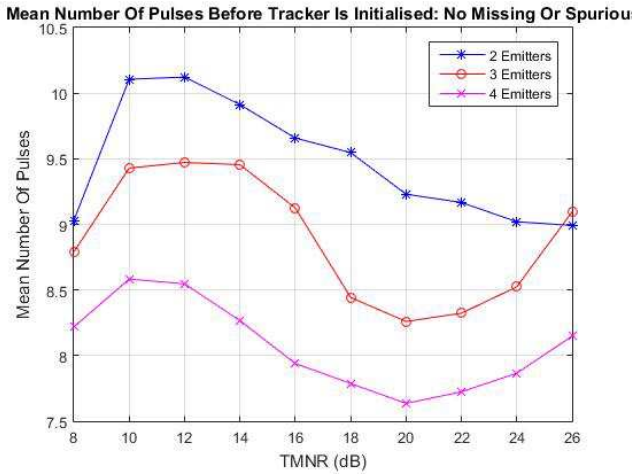


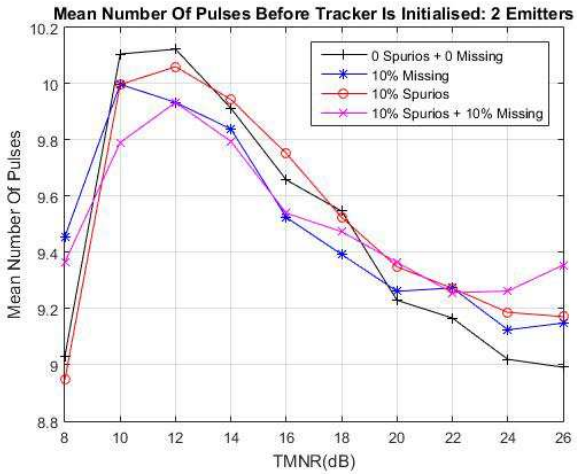


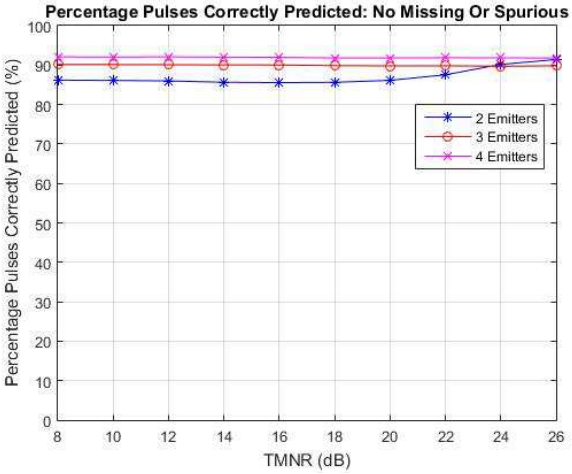


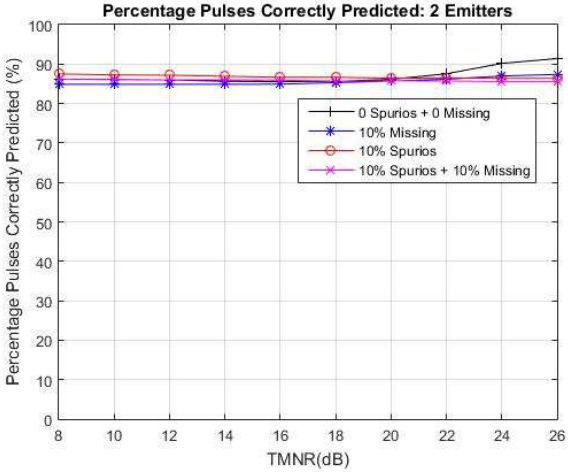


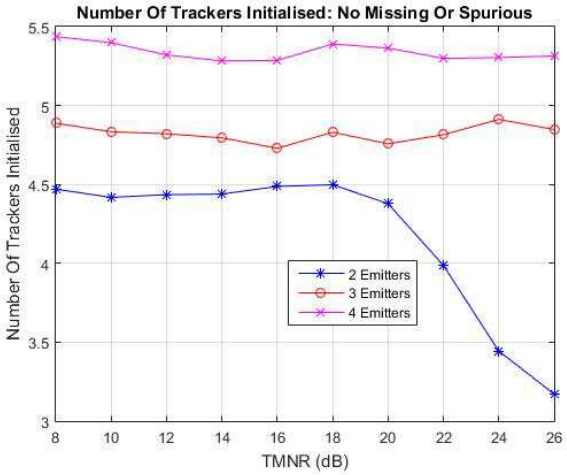
Mixed Signals - All Types Results

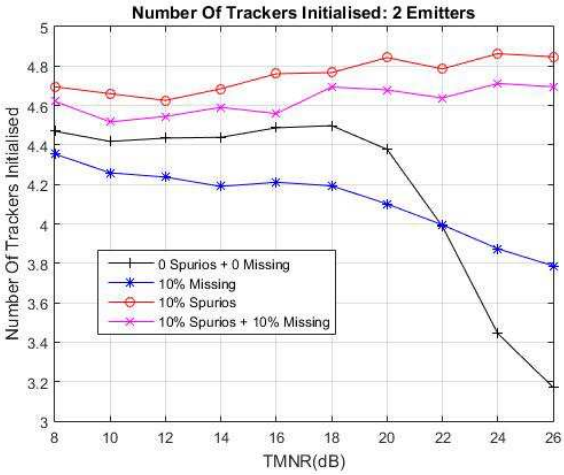









Number Of Trackers Initialised: No Missing Or Spurious
Number Of Trackers Initialised
TMNR (dB)
2 Emitters
3 Emitters
4 Emitters


Number Of Trackers Initialised: 2 Emitters
Number Of Trackers Initialised
TMNR(dB)
0 Spurios + 0 Missing
10% Missing
10% Spurios
10% Spurios + 10% Missing